Video-to-Video Synthesis

Ting-Chun Wang[1], **Ming-Yu Liu**[1], **Jun-Yan Zhu**[2], **Guilin Liu**[1],
Andrew Tao[1], **Jan Kautz**[1], **Bryan Catanzaro**[1]
[1]NVIDIA, [2]MIT CSAIL
{tingchunw,mingyul,guilinl,atao,jkautz,bcatanzaro}@nvidia.com,
junyanz@mit.edu

Abstract

We study the problem of video-to-video synthesis, whose goal is to learn a mapping function from an input source video (e.g., a sequence of semantic segmentation masks) to an output photorealistic video that precisely depicts the content of the source video. While its image counterpart, the image-to-image translation problem, is a popular topic, the video-to-video synthesis problem is less explored in the literature. Without modeling temporal dynamics, directly applying existing image synthesis approaches to an input video often results in temporally incoherent videos of low visual quality. In this paper, we propose a novel video-to-video synthesis approach under the generative adversarial learning framework. Through carefully-designed generator and discriminator architectures, coupled with a spatio-temporal adversarial objective, we achieve high-resolution, photorealistic, temporally coherent video results on a diverse set of input formats including segmentation masks, sketches, and poses. Experiments on multiple benchmarks show the advantage of our method compared to strong baselines. In particular, our model is capable of synthesizing 2K resolution videos of street scenes up to 30 seconds long, which significantly advances the state-of-the-art of video synthesis. Finally, we apply our approach to future video prediction, outperforming several state-of-the-art competing systems. Code, models, and more results are available at our website.

1 Introduction

The capability to model and recreate the dynamics of our visual world is essential to building intelligent agents. Apart from purely scientific interests, learning to synthesize continuous visual experiences has a wide range of applications in computer vision, robotics, and computer graphics. For example, in model-based reinforcement learning [1, 22], a video synthesis model finds use in approximating visual dynamics of the world for training the agent with less amount of real experience data. Using a learned video synthesis model, one can generate realistic videos without explicitly specifying scene geometry, materials, lighting, and their dynamics, which would be cumbersome but necessary when using standard graphics rendering techniques [70].

The video synthesis problem exists in various forms, including future video prediction [62, 47, 16, 42, 74, 68, 13, 65, 39] and unconditional video synthesis [66, 56, 64]. In this paper, we study a new form: video-to-video synthesis. At the core, we aim to learn a mapping function that can convert an input video to an output video. To the best of our knowledge, a general-purpose solution to video-to-video synthesis has not yet been explored in the prior work, although its image counterpart, the image-to-image translation problem, is a popular research topic [31, 63, 4, 60, 77, 40, 41, 29, 78, 70]. Our method is inspired by previous application-specific video synthesis methods [57, 58, 72, 55].

We cast the video-to-video synthesis problem as a distribution matching problem, where the goal is to train a model such that the conditional distribution of the synthesized videos given input videos resembles that of real videos. To this end, we leverage the generative adversarial learning

Preprint. Work in progress.

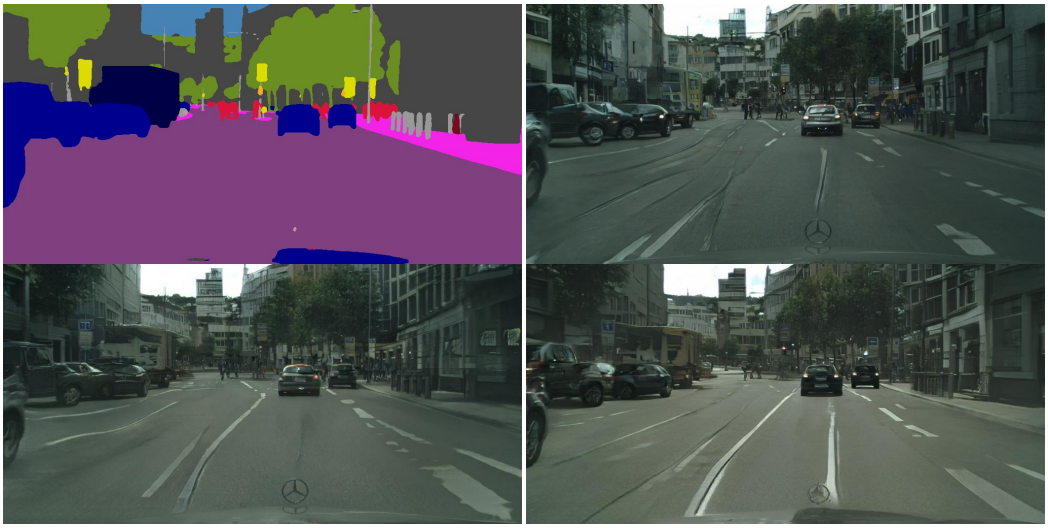

Figure 1: Generating a photorealistic video from an input segmentation map video on Cityscapes. Top left: input. Top right: `pix2pixHD`. Bottom left: `COVST`. Bottom right: `vid2vid` (ours). *The figure is best viewed with Acrobat Reader. Click the image to play the video clip.*

framework [18]. Given aligned input and output videos, we learn to map input videos to the output domain at the test time. With carefully-designed generator and discriminator networks and a novel spatio-temporal learning objective function, the proposed approach can learn to synthesize high-resolution, photorealistic, and temporally coherent videos. Moreover, we later extend our method to multimodal video synthesis. Conditioning on the same input, our model can produce videos with diverse appearances.

We conduct extensive experimental validation on several datasets on the task of converting a sequence of segmentation masks to photorealistic videos. Both quantitative and qualitative results indicate that our synthesized footage looks more photorealistic than those from strong baselines. See Figure 1 for an example. We further demonstrate that the proposed approach is capable of generating photorealistic 2K resolution videos, up to 30 seconds long. Our method also grants users flexible high-level control over the video generation results. For example, a user can easily replace all the buildings with trees in a street view video. In addition, we extend our approach to future prediction and show that our method can outperform existing systems. Code, models, and more results are available at our website.

2 Related Work

Generative Adversarial Networks (GANs). We build our model on GANs [18]. During GAN training, a generator and a discriminator are set up to play a zero-sum game. The generator aims to produce realistic synthetic data so that the discriminator cannot differentiate between real and synthetic data. In addition to random samples from a noise distribution [18, 52, 12], various forms of data can also be used as input to the generator, including images [31, 77, 40], categorical labels [50, 49], and textual descriptions [53, 76]. Such conditional models are called conditional GANs, and allow flexible control over the output of the model. Our method belongs to the category of conditional video generation with GANs. However, instead of predicting future videos conditioning on the current observed images [47, 38, 66], our method synthesizes photorealistic videos conditioning on manipulable semantic representations, such as segmentation masks, sketches, or poses.

Image-to-image translation algorithms transfer an input image from one domain to a corresponding image in another domain. There exists a large body of work for this problem [31, 63, 4, 60, 77, 40, 41, 29, 78, 70]. Our approach is their video counterpart. In addition to ensuring that each video frame looks photorealistic, a video synthesis model also has to produce temporally coherent frames, which is a challenging task, especially for a long duration video.

Unconditional video synthesis. Several works [66, 56, 64] extend the GAN framework for unconditional video synthesis, which learns a generator for converting a random vector to a video.

VGAN [66] uses a spatio-temporal convolutional network. TGAN [56] projects a latent code to a set of latent image codes, and uses an image generator to convert those latent image codes to frames. MoCoGAN [64] disentangles the latent space to motion and content subspaces and uses a recurrent neural network to generate a sequence of motion codes. Due to the unconditional setting, these methods often produce low-resolution and short-length videos.

Future video prediction. Conditioning on the current observed images, video prediction models are trained to predict future frames [62, 33, 16, 47, 42, 74, 68, 69, 13, 65, 39, 38]. These models are often trained with image reconstruction losses. As a result, they tend to generate blurry videos due to the classic regress-to-the-mean problem. Also, they fail to generate long duration videos even with adversarial training [47, 39]. The video-to-video synthesis problem is substantially different because it does not attempt to predict object motions or any other aspect of the future video frames. Instead, our approach is conditional on an existing video, and is capable of generating high-resolution and long-length videos in a different domain.

Video-to-video synthesis. While video super-resolution [58, 59], video matting and blending [2, 10], and video inpainting [71] can be considered as special cases of the video-to-video synthesis problem, existing approaches rely on problem-specific constraints in their designs. Hence, these methods cannot be easily applied to other video-to-video applications. Video style transfer, transferring the style of a reference image (e.g., painting) to a video (e.g., a natural scene video), is another related problem. Although existing methods [8, 20, 26, 55] cannot directly work for the video-to-video synthesis problem, we compare our method with a strong baseline that combines a state-of-the-art video style transfer algorithm with a state-of-the-art image-to-image translation approach.

3 Video-to-Video Synthesis

Let $\mathbf{s}_1^T \equiv \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_T\}$ be a sequence of source images for video synthesis. For example, it can be a sequence of semantic segmentation masks or boundary maps. Let $\mathbf{x}_1^T \equiv \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ be the sequence of corresponding real images. The goal of video-to-video synthesis is to learn a mapping function that can convert $\mathbf{s}_1^T$ to a sequence of output images, $\tilde{\mathbf{x}}_1^T \equiv \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ..., \tilde{\mathbf{x}}_T\}$, so that the conditional distribution of $\tilde{\mathbf{x}}_1^T$ given $\mathbf{s}_1^T$ is identical to the conditional distribution of $\mathbf{x}_1^T$ given $\mathbf{s}_1^T$.

$$p(\tilde{\mathbf{x}}_1^T|\mathbf{s}_1^T) = p(\mathbf{x}_1^T|\mathbf{s}_1^T). \tag{1}$$

Through matching the conditional video distributions, the model learns to generate photorealistic and temporally coherent output sequences as if they were captured by a video camera.

We propose a conditional GAN framework for the conditional video distribution matching task. Let G be a generator that maps an input source sequence to a corresponding output image sequence: $\mathbf{x}_1^T = G(\mathbf{s}_1^T)$. We train the generator by solving the minimax optimization problem given by

$$\max_D \min_G E_{(\mathbf{x}_1^T, \mathbf{s}_1^T)}[\log D(\mathbf{x}_1^T, \mathbf{s}_1^T)] + E_{\mathbf{s}_1^T}[\log(1 - D(G(\mathbf{s}_1^T), \mathbf{s}_1^T))], \tag{2}$$

where D is the discriminator. We note that as solving (2), we minimize the Jensen-Shannon divergence between $p(\tilde{\mathbf{x}}_1^T|\mathbf{s}_1^T)$ and $p(\mathbf{x}_1^T|\mathbf{s}_1^T)$ as shown by Goodfellow et al. [18].

Solving the minimax optimization problem in (2) is a well-known, challenging task. Careful design of the network architecture and the objective function are required to achieve good performance as shown in the literature [12, 52, 28, 76, 19, 46, 34, 70, 48]. We follow the same spirit and propose new network designs and a spatio-temporal objective for video-to-video synthesis as detailed below.

Sequential generator. To simplify the video-to-video synthesis problem, we make a Markov assumption where we factorize the conditional distribution $p(\tilde{\mathbf{x}}_1^T|\mathbf{s}_1^T)$ to a product form given by

$$p(\tilde{\mathbf{x}}_1^T|\mathbf{s}_1^T) = \prod_{t=1}^{T} p(\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^t). \tag{3}$$

In other words, we assume the video frames can be generated sequentially, and the generation of the t-th frame $\tilde{\mathbf{x}}_t$ only depends on three things: 1) the current source image $\mathbf{s}_t$, 2) the past L source images $\mathbf{s}_{t-L}^{t-1}$, and 3) the past L generated images $\tilde{\mathbf{x}}_{t-L}^{t-1}$. We train a feed-forward network F to model the conditional distribution $p(\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^t)$ as $\tilde{\mathbf{x}}_t = F(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^t)$. We obtain the final output $\tilde{\mathbf{x}}_1^T$ by applying the function F in a recursive manner. We found that a small L (e.g., $L = 1$) causes the

training to be less stable, while a large L increases training time and GPU memory but with minimal quality improvement. In our experiments, we set $L = 2$.

Video signals contain a large amount of redundant information in consecutive frames. If the optical flow [43] from the current frame to the next frame is known, we can use it to warp the current frame to generate an estimation of the next frame [67, 51]. This estimation would be largely correct except for the occluding areas. Based on this observation, we model F as

$$F(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t}) = (\mathbf{1} - \tilde{\mathbf{m}}_t) \odot \tilde{\mathbf{w}}_{t-1}(\tilde{\mathbf{x}}_{t-1}) + \tilde{\mathbf{m}}_t \odot \tilde{\mathbf{h}}_t, \tag{4}$$

where $\odot$ is the element-wise product operator and $\mathbf{1}$ is an image of all ones. The first part corresponds to pixels warped from the previous frame, while the second part aims to hallucinate new pixels. The definitions of the other terms in Equation 4 are given below.

- $\tilde{\mathbf{w}}_{t-1} = W(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t})$ is the estimated optical flow from $\tilde{\mathbf{x}}_{t-1}$ to $\tilde{\mathbf{x}}_t$, and W is the optical flow prediction function. We estimate the optical flow using both input source images $\mathbf{s}_{t-L}^{t}$ and previously synthesized images $\tilde{\mathbf{x}}_{t-L}^{t-1}$. By $\tilde{\mathbf{w}}_{t-1}(\tilde{\mathbf{x}}_{t-1})$, we warp $\tilde{\mathbf{x}}_{t-1}$ based on $\tilde{\mathbf{w}}_{t-1}$.
- $\tilde{\mathbf{h}}_t = H(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t})$ is the hallucinated image, an image generated from scratch.
- $\tilde{\mathbf{m}}_t = M(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^{t})$ is the occlusion mask with continuous values between 0 and 1. The function M is the mask prediction function. Note that our occlusion mask is soft instead of binary to better handle the "zoom in" scenario. For example when an object is moving closer to our camera, the object will become blurrier over time if we only warp previous frames. To increase the resolution of the object, we need to synthesize new texture details. By using a soft mask, we can add details by gradually blending the warped pixels and the newly synthesized pixels.

We implement the functions M, W, and H using a residual network architecture [24]. To generate high-resolution videos, we adopt a coarse-to-fine generator design proposed by Wang et. al. [70].

Using multiple discriminators has been shown beneficial in mitigating the model collapse problem in GAN training [17, 64, 70]. We use two types of discriminators in our approach.

Conditional image discriminator D_I. The purpose of D_I is to ensure that each output frame resembles a real image given the same source image. This conditional discriminator should output 1 for a true pair $(\mathbf{x}_t, \mathbf{s}_t)$ and 0 for a fake one $(\tilde{\mathbf{x}}_t, \mathbf{s}_t)$.

Conditional video discriminator D_V. The purpose of D_V is to ensure that consecutive output frames resemble the temporal dynamics of a real video given the same optical flow. It is also a conditional discriminator; While D_I conditions on the source image, D_V conditions on the flow. Let $\mathbf{w}_{t-K}^{t-2}$ be $K-1$ optical flow for the K consecutive real images $\mathbf{x}_{t-K}^{t-1}$. This conditional discriminator D_V should output 1 for a true pair $(\mathbf{x}_{t-K}^{t-1}, \mathbf{w}_{t-K}^{t-2})$ and 0 for a fake one $(\tilde{\mathbf{x}}_{t-K}^{t-1}, \mathbf{w}_{t-K}^{t-2})$.

We introduce two sampling operators to facilitate the discussion. First, let ϕ_I be a random image sampling operator such that $\phi_I(\mathbf{x}_1^T, \mathbf{s}_1^T) = (\mathbf{x}_i, \mathbf{s}_i)$ where i is an integer uniformly sampled from 1 to T. In other words, ϕ_I randomly samples a pair of images from $(\mathbf{x}_1^T, \mathbf{s}_1^T)$. On the other hand, we define ϕ_V as a sampling operator that randomly retrieve K consecutive frames. Specifically, $\phi_V(\mathbf{w}_1^{T-1}, \mathbf{x}_1^T, \mathbf{s}_1^T) = (\mathbf{w}_{i-K}^{i-2}, \mathbf{x}_{i-K}^{i-1}, \mathbf{s}_{i-K}^{i-1})$ where i is an integer uniformly sampled from $K+1$ to $T+1$. This operator retrieves K consecutive frames and the corresponding $K-1$ optical flow images. With ϕ_I and ϕ_V, we are ready to present our learning objective function.

Learning objective function. We train the sequential video synthesis function F by solving

$$\min_F \big(\max_{D_I} \mathcal{L}_I(F, D_I) + \max_{D_V} \mathcal{L}_V(F, D_V) \big) + \lambda_W \mathcal{L}_W(F), \tag{5}$$

where $\mathcal{L}_I$ is the GAN loss on images defined by the conditional image discriminator D_I, $\mathcal{L}_V$ is the GAN loss on K consecutive frames defined by D_V, and $\mathcal{L}_W(F)$ is the flow estimation loss. The weight λ_W is set to 10 throughout the experiments. In addition to the loss terms in Equation 5, we also use the discriminator feature matching loss [37, 70] and VGG feature matching loss [32, 14, 70] as they improve the convergence speed and training stability [70]. Please see the appendix for more details.

The image GAN loss $\mathcal{L}_I$ is an image-conditional GAN loss [31] by utilizing the operator ϕ_I

$$E_{\phi_I(\mathbf{x}_1^T, \mathbf{s}_1^T)}[\log D_I(\mathbf{x}_i, \mathbf{s}_i)] + E_{\phi_I(\tilde{\mathbf{x}}_1^T, \mathbf{s}_1^T)}[\log(1 - D_I(\tilde{\mathbf{x}}_i, \mathbf{s}_i))]. \tag{6}$$

Similarly, the video GAN loss $\mathcal{L}_V$ is given by

$$E_{\phi_V(\mathbf{w}_1^{T-1},\mathbf{x}_1^T,\mathbf{s}_1^T)}[\log D_V(\mathbf{x}_{i-K}^{i-1},\mathbf{w}_{i-K}^{i-2})] + E_{\phi_V(\mathbf{w}_1^{T-1},\tilde{\mathbf{x}}_1^T,\mathbf{s}_1^T)}[\log(1 - D_V(\tilde{\mathbf{x}}_{i-K}^{i-1},\mathbf{w}_{i-K}^{i-2}))]. \quad (7)$$

Recall that we synthesize a video $\tilde{\mathbf{x}}_1^T$ by recursively applying F.

The flow loss $\mathcal{L}_W$ includes two terms. The first is the endpoint error between the ground truth and the estimated flow, and the second is the warping loss when the flow is used to warp the previous frame to the next frame. Let $\mathbf{w}_t$ be the ground truth flow from $\mathbf{x}_t$ to $\mathbf{x}_{t+1}$. The flow loss $\mathcal{L}_W$ is given by

$$\mathcal{L}_W = \frac{1}{T-1}\sum_{t=1}^{T-1}\big(||\tilde{\mathbf{w}}_t - \mathbf{w}_t||_1 + \|\tilde{\mathbf{w}}_t(\mathbf{x}_t) - \mathbf{x}_{t+1}\|_1\big). \quad (8)$$

,

Foreground-background prior. When using semantic segmentation masks as the source video, we can divide an image into foreground and background areas based on the semantics. For example, buildings and roads belong to the background, while cars and pedestrians are considered as the foreground. We leverage this strong foreground-background prior in the generator design to further improve our performance.

In particular, we decompose the image hallucination function H into a foreground model $\tilde{\mathbf{h}}_{F,t} = H_F(\mathbf{s}_{t-L}^t)$ and a background model $\tilde{\mathbf{h}}_{B,t} = H_B(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^t)$. We note that background motion is a global transformation in general, where optical flow can be estimated quite accurately. As a result, background image synthesis can be generated accurately via warping. The background hallucination function H_B aims to synthesize background content in the occluded areas. On the other hand, a foreground object often has a large motion and only occupies a small portion of the image, which makes optical flow estimation difficult. The function H_F has to synthesize most of the foreground content from scratch. With this foreground–background prior, F is then given by

$$F(\tilde{\mathbf{x}}_{t-L}^{t-1}, \mathbf{s}_{t-L}^t) = (\mathbf{1} - \tilde{\mathbf{m}}_t) \odot \tilde{\mathbf{w}}_{t-1}(\tilde{\mathbf{x}}_{t-1}) + \tilde{\mathbf{m}}_t \odot \big((\mathbf{1} - \mathbf{m}_{B,t}) \odot \tilde{\mathbf{h}}_{F,t} + \mathbf{m}_{B,t} \odot \tilde{\mathbf{h}}_{B,t}\big), \quad (9)$$

where $\mathbf{m}_{B,t}$ is the ground truth background mask derived from $\mathbf{s}_t$. In practice, we find that the model converges much faster when the foreground areas are set as occluded in the occlusion mask.

Multimodal synthesis. The synthesis function F is a unimodal mapping function. Given an input source video, it can only generate one output video. To achieve multimodal synthesis [78, 70, 17], we adopt a feature embedding scheme [70] for the source video that consists of instance-level semantic segmentation masks. Specifically, at training time, we train an image encoder E to encode the ground truth real image $\mathbf{x}_t$ into a d-dimensional feature map ($d = 3$ in our experiments). We then apply an instance feature averaging operation to the map so that all the pixels within the same object share the same feature vectors. We then feed both the instance-averaged feature map $\mathbf{z}_t$ and the input semantic segmentation mask $\mathbf{s}_t$ to the generator F. Once training is done, we fit a mixture of Gaussian distribution to the feature vectors that belong to the same class. At test time, we sample a feature vector for each object instance using the estimated distribution of that object class. Given different feature vectors, the generator F can synthesize videos with different visual appearances.

4 Experiments

Implementation details. We train our network in a spatio-temporally progressive manner. In particular, we start with generating low-resolution and few frames, and all the way up to generating full resolution and 30 (or more) frames. Our coarse-to-fine generator consists of three scales, which operates on 512×256, 1024×512, and 2048×1024 resolutions, respectively. The mask prediction function M and flow prediction function W share all the weights except for the output layer. We use the multi-scale PatchGAN discriminator architecture [31, 70] for D_I. In addition to multi-scale in the spatial resolution, our multi-scale D_V also focuses on different frame rates of the video to ensure both short-term and long-term consistency. Please see the appendix for more details.

We train our model for 40 epochs using the ADAM [36] optimizer with lr $= 0.0002$ and $(\beta_1, \beta_2) = (0.5, 0.999)$ on an NVIDIA DGX1 machine. We use the LSGAN loss [46]. Due to the high image resolution, even with one short video per batch, we have to use all the GPUs in DGX1 (8 V100 GPUs, each with 16GB memory) for training. We distribute the generator computation task to 4 GPUs and the discriminator computation task to the other 4 GPUs. Training takes ~ 10 days for 2K resolution.

Datasets. We evaluate the proposed approach on several datasets.

Table 1: Comparison between competing video-to-video synthesis approaches on Cityscapes.

Fréchet Inception Distance	I3D	ResNeXt
pix2pixHD	5.57	0.18
COVST	5.55	0.18
vid2vid (ours)	**4.66**	**0.15**

Human Preference Score	short seq.	long seq.
vid2vid (ours) / pix2pixHD	**0.87** / 0.13	**0.83** / 0.17
vid2vid (ours) / COVST	**0.84** / 0.16	**0.80** / 0.20

Table 2: Ablation study. We compare the proposed approach to its three variants.

Human Preference Score	
vid2vid (ours) / no background-foreground prior	**0.80** / 0.20
vid2vid (ours) / no conditional video discriminator	**0.84** / 0.16
vid2vid (ours) / no flow warping	**0.67** / 0.33

Table 3: Comparison between future video prediction methods on Cityscapes.

Fréchet Inception Distance	I3D	ResNeXt
PredNet	11.18	0.59
MCNet	10.00	0.43
vid2vid (ours)	**3.44**	**0.18**

Human Preference Score	
vid2vid (ours) / PredNet	**0.92** / 0.08
vid2vid (ours) / MCNet	**0.98** / 0.02

- **Cityscapes** [11]. The dataset consists of 2048×1024 street scene videos captured in several German cities. Only a subset of images in the videos contains ground truth semantic segmentation masks. To obtain the input source videos, we use those images to train a DeepLabV3 semantic segmentation network [9], and apply the trained network to segment all the videos. We use the optical flow extracted by FlowNet2 [30] as the ground truth flow $\mathbf{w}$ to train F. We treat the instance segmentation masks computed by the Mask R-CNN [23] as our instance-level ground truth. In summary, the training set contains 2975 videos, each with 30 frames. The validation set consists of 500 videos, each with 30 frames. Finally, we test our method on 3 long sequences from the Cityscapes demo videos, with 600, 1100, and 1200 frames, respectively. We will show that although we train our model using short videos, it is capable of synthesizing long videos.
- **Apolloscape** [27] consists of 73 street scene videos captured in Beijing, where the video length varies from 100 to 1000 frames. Similar to Cityscapes, Apolloscape is constructed for the image/video semantic segmentation task. But we use it for synthesizing videos using the semantic segmentation mask. We split the dataset into half for training and validation.
- **Face video dataset** [54]. We use the real videos in the FaceForensics dataset, which contains 854 videos of news briefing from different reporters. We use this dataset for the sketch video to face video synthesis task. To extract a sequence of sketches from a video, we first apply a face alignment algorithm [35] to localize facial landmarks in each frame. The facial landmarks are then connected to create the face sketch. For background, we extract Canny edges outside the face regions. We split the dataset into 704 videos for training and 150 videos for validation.
- **Dance video dataset.** We download YouTube dance videos for the pose to human motion synthesis task. Each video is about $3 \sim 4$ minutes at 1280×720 resolution, and we crop the central 512×720 regions. We extract human poses with the DensePose [21] and the OpenPose [5] algorithms, and directly concatenate the results together. The training set includes a dance video from a single dancer, while the test set contains videos of other dance motions or from other dancers.

Baselines. We compare our approach to two baselines trained on the same data.

- pix2pixHD [70] is the state-of-the-art image-to-image translation approach. When applying the approach to the video-to-video synthesis task, we process input videos frame-by-frame.
- COVST is built on the coherent video style transfer [8] by replacing the stylization network with pix2pixHD. The key idea in COVST is to warp high-level deep features using optical flow for achieving temporally coherent outputs. No additional adversarial training is applied. We feed in ground truth optical flow to COVST, which is impractical for real applications. In contrast, our model estimates optical flow from source videos.

Figure 2: Apolloscape results. Left: `pix2pixHD`. Center: `COVST`. Right: proposed. The input semantic segmentation mask video is shown in the left video. *The figure is best viewed with Acrobat Reader. Click the image to play the video clip.*

Figure 3: Example multi-modal video synthesis results. These synthesized videos contain different road surfaces. *The figure is best viewed with Acrobat Reader. Click the image to play the video clip.*

Figure 4: Example results of changing input semantic segmentation masks to generate diverse videos. Left: tree→building. Right: building→tree. The original video is shown in Figure 3. *The figure is best viewed with Acrobat Reader. Click the image to play the video clip.*

Evaluation metrics. We use both subjective and objective metrics for performance evaluation.

- **Human preference score.** We perform a human subjective test for evaluating the visual quality of synthesized videos. We use the Amazon Mechanical Turk (AMT) platform. Each question is an AB test, where an AMT worker is first shown two videos at a time (results synthesized by two different algorithms) and then asked which one looks more like a video captured by a real camera. We specifically ask the worker to check for both temporal coherence and image quality. A worker must have a life-time task approval rate greater than 98% to participate in the evaluation. For each question, we gather answers from 10 different workers. We calculate the human preference score of an algorithm as the ratio that the algorithm outputs are preferred.

- **Fréchet Inception Distance (FID)** [25] is a widely used metric for implicit generative models, as it correlates well with the visual quality of generated samples. The FID was originally developed for evaluating image generation. We propose a variant for video evaluation, which measures both visual quality and temporal consistency. Specifically, we use a pre-trained video recognition CNN as a feature extractor after removing the last few layers from the network. This feature extractor will be our "inception" network. For each video, we extract a spatio-temporal feature map with this CNN. We compute the mean $\tilde{\mu}$ and covariance matrix $\tilde{\Sigma}$ for the feature vectors from all the synthesized videos. We also calculate the same quantities μ and Σ for the ground truth videos. The FID is then calculated as $\|\mu - \tilde{\mu}\|^2 + \mathrm{Tr}\left(\Sigma + \tilde{\Sigma} - 2\sqrt{\Sigma\tilde{\Sigma}}\right)$. We use two different pre-trained video recognition CNNs in our evaluation, I3D [6] and ResNeXt [73].

Figure 5: Example sketch-to-face video results. Our method can generate realistic expressions given the edge maps. *The figure is best viewed with Acrobat Reader. Click the image to play the video clip.*

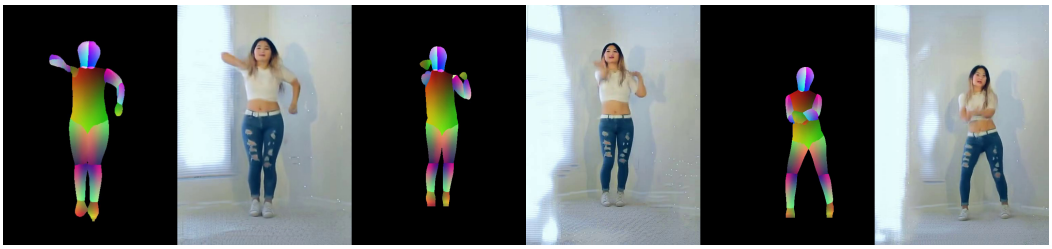

Figure 6: Example pose-to-dance results. The left image pair shows the result on the same dancer (with different clothing) doing different motions, while the other two pairs are results on different dancers. *The figure is best viewed with Acrobat Reader. Click the image to play the video clip.*

Main results. We compare the proposed approach to the baselines on the Cityscapes benchmark, where we apply the learned models to synthesize 500 short video clips in the validation set. Quantitative and qualitative evaluation results are reported in Table 1, which show that the our results have a smaller FID and are often favored by the human subjects. We also report the human preference scores on the three long test videos. Again, the videos rendered by our approach are considered more realistic by the human subjects. The performance scores for the Apolloscape dataset are given in the appendix.

Figures 1 and 2 show the video synthesis results. Although each frame rendered by `pix2pixHD` is photorealistic, the resulting video lacks temporal coherence. The road lane markings and building appearances are inconsistent across frames. While improving upon `pix2pixHD`, `COVST` still suffers from temporal inconsistency. On the contrary, our approach produces high-resolution, photorealistic, and temporally consistent video output. We can generate 30-second long videos, showing that our approach synthesizes convincing videos with longer lengths.

We conduct an ablation study to analyze several design choices of our method. Specifically, we create three variants. In one variant, we do not use the foreground-background prior, which is termed `no background-foreground prior`. That is, instead of using Equation 9, we use Equation 4. The second variant is `no conditional video discriminator` where we do not use D_V for training. In the last variant, we remove W and M from F in Equation 4 and only use H for synthesis. This variant is referred to as `no flow warping`. We use the human preference score on Cityscapes for this ablation study. Table 2 shows that the visual quality of output videos degrades significantly without the ablated components.

Multimodal results. Figure 3 shows example multimodal synthesis results. In this example, we keep the sampled feature vectors of all the object instances in the video the same except for the road instance. The figure shows temporally smooth videos with different road surface appearances.

Semantic manipulation. Our approach also allows the user to manipulate the semantics of source videos. In Figure 4, we show an example of changing the semantic labels. In the left video, we replace all trees with buildings in the original segmentation masks and synthesize a new video. On the right, we show the result of replacing buildings with trees.

Sketch-to-video synthesis. We train a sketch-to-face synthesis video model using the real face videos in the FaceForensics dataset [54]. As shown in Figure 5, our model can convert sequences of sketches to photorealistic output videos.

Pose-to-video synthesis. We also apply our method to the task of converting sequences of human poses to photorealistic output videos. We note that the image counterpart was studied in recent works [44, 15, 3, 45], and similar tasks for generating human motions were also explored in [7, 75]. As shown in Figure 6, our model learns to synthesize high-resolution photorealistic output dance videos that contain unseen body shapes and motions.

Figure 7: Future video prediction results. Top left: ground truth. Top right: PredNet [42]. Bottom left: MCNet [65]. Bottom right: ours. *The figure is best viewed with Acrobat Reader. Click the image to play the video clip.*

Future video prediction. We show an extension of our approach to the future video prediction task: learning to predict the future video given a few observed frames. We decompose the task into two sub-tasks: 1) synthesizing future semantic segmentation masks using the observed frames, and 2) converting the synthesized segmentation masks into videos. After extracting the segmentation masks from the observed frames, we train a generator similar to Equation 7 to predict future semantic masks via solving a variant of Equation 5. We then use the proposed video-to-video synthesis approach to convert the predicted segmentation masks to a future video.

We conduct both quantitative and qualitative evaluations with comparisons to two start-of-the-art approaches: PredNet [42] and MCNet [65]. We follow the prior works [67, 38] and report the human preference score. We also include the FID scores. As shown in Table 3, our model produces smaller FIDs, and the human subjects favor our resulting videos. In Figure 7, we visualize the future video synthesis results. While the image quality of the results from the competing algorithms degrades significantly over time, ours remains consistent.

5 Conclusion

We present a general video-to-video synthesis framework based on conditional GANs. Through carefully-designed generator and discriminator networks as well as a spatio-temporal adversarial objective, we can synthesize high-resolution, photorealistic, and temporally consistent videos. Extensive experiments demonstrate that our results are significantly better than the results by state-of-the-art methods. Its extension to the future video prediction task also compares favorably against the competing approaches.

Limitations and future work. Although our approach outperforms previous methods, our model still fails in a couple of situations. For example, our model struggles in synthesizing turning cars due to insufficient information in label maps. We speculate that this could be potentially addressed by adding additional 3D cues, such as depth maps. Furthermore, our model still can not guarantee that an object has a consistent appearance across the whole video. Occasionally, a car may change its color gradually. This issue might be alleviated if object tracking information is used to enforce that the same object shares the same appearance throughout the entire video. Finally, when we perform semantic manipulations such as turning trees into buildings, visible artifacts occasionally appear as building and trees have different label shapes. This might be resolved if we train our model with coarser semantic labels, as the trained model would be less sensitive to label shapes.

Acknowledgements We thank Karan Sapra, Fitsum Reda, and Matthieu Le for generating the segmentation maps for us. We also thank Lisa Rhee for allowing us to use her dance videos for training. We thank William S. Peebles for proofreading the paper.

References

[1] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[2] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *ACM Transactions on Graphics (TOG)*, volume 28, page 70, 2009.

[3] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag. Synthesizing images of humans in unseen poses. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. Everybody dance now. In *European Conference on Computer Vision (ECCV) Workshop*, 2018.

[8] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua. Coherent online video style transfer. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[9] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[10] T. Chen, J.-Y. Zhu, A. Shamir, and S.-M. Hu. Motion-aware gradient domain video composition. *IEEE Trans. Image Processing*, 22(7):2532–2544, 2013.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[13] E. L. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[14] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[15] P. Esser, E. Sutter, and B. Ommer. A variational u-net for conditional appearance and shape generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[17] A. Ghosh, V. Kulharia, V. Namboodiri, P. H. Torr, and P. K. Dokania. Multi-agent diverse generative adversarial networks. *arXiv preprint arXiv:1704.02906*, 2017.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[19] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[20] A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei. Characterizing and improving stability in neural style transfer. *arXiv preprint arXiv:1705.02092*, 2017.

[21] R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[22] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[26] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu. Real-time neural style transfer for videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The ApolloScape dataset for autonomous driving. *arXiv preprint arXiv:1803.06184*, 2018.

[28] X. Huang, Y. Li, O. Poursaeed, J. E. Hopcroft, and S. J. Belongie. Stacked generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. *arXiv preprint arXiv:1804.04732*, 2018.

[30] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016.

[33] N. Kalchbrenner, A. v. d. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016.

[34] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.

[35] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 2009.

[36] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[37] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning (ICML)*, 2016.

[38] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[39] X. Liang, L. Lee, W. Dai, and E. P. Xing. Dual motion GAN for future-flow embedded video prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[40] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[41] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[42] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[43] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[44] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[45] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz. Disentangled person image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 99–108, 2018.

[46] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[47] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations (ICLR)*, 2016.

[48] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[49] T. Miyato and M. Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations (ICLR)*, 2018.

[50] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. In *International Conference on Machine Learning (ICML)*, 2017.

[51] K. Ohnishi, S. Yamamoto, Y. Ushiku, and T. Harada. Hierarchical video generation from orthogonal information: Optical flow and texture. *arXiv preprint arXiv:1711.09618*, 2017.

[52] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2015.

[53] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning (ICML)*, 2016.

[54] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*, 2018.

[55] M. Ruder, A. Dosovitskiy, and T. Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, 2016.

[56] M. Saito, E. Matsumoto, and S. Saito. Temporal generative adversarial nets with singular value clipping. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[57] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *ACM Transactions on Graphics (TOG)*, 2000.

[58] E. Shechtman, Y. Caspi, and M. Irani. Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(4):531–545, 2005.

[59] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[60] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[61] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[62] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning (ICML)*, 2015.

[63] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *International Conference on Learning Representations (ICLR)*, 2017.

[64] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. MoCoGAN: Decomposing motion and content for video generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[65] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations (ICLR)*, 2017.

[66] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[67] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[68] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision (ECCV)*, 2016.

[69] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[70] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[71] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[72] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(3), 2007.

[73] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[74] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[75] C. Yang, Z. Wang, X. Zhu, C. Huang, J. Shi, and D. Lin. Pose guided human video generation. *arXiv preprint arXiv:1807.11152*, 2018.

[76] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

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

[78] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

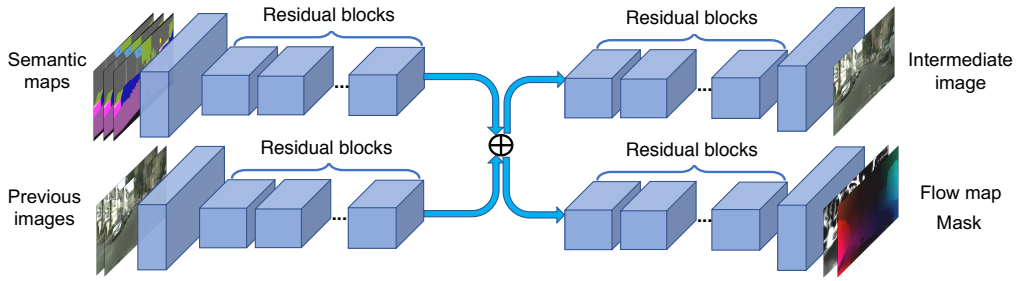

Figure 8: The network architecture (G_1) for low-res videos. Our network takes in a number of semantic label maps and previously generated images, and outputs the intermediate frame as well as the flow map and the mask.

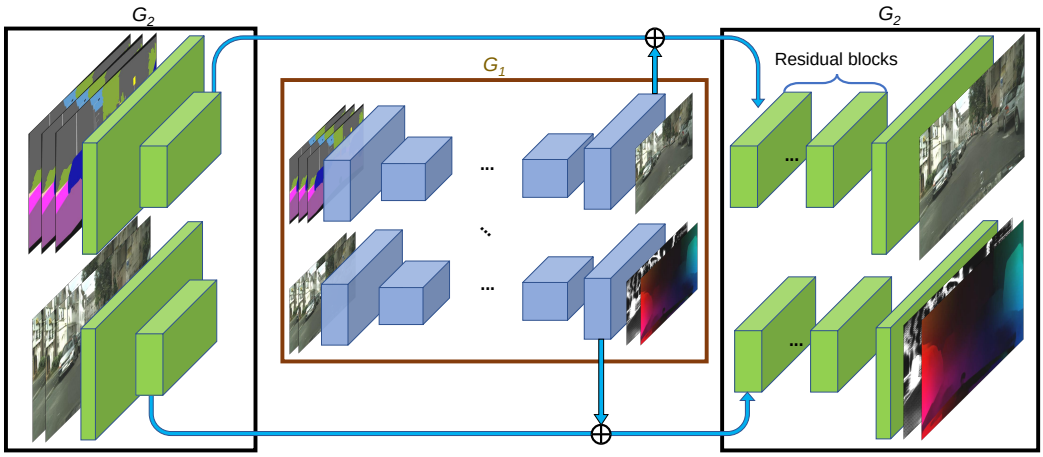

Figure 9: The network architecture (G_2) for higher resolution videos. The label maps and previous frames are downsampled and fed into the low-res network G_1. Then, the features from the high-res network and the last layer of the low-res network are summed and fed into another series of residual blocks to output the final images.

A Network Architecture

A.1 Generators

Our network adopts a coarse-to-fine architecture. For the lowest resolution, the network takes in a number of semantic label maps $\mathbf{s}_{t-L}^{t}$ and previously generated frames $\tilde{\mathbf{x}}_{t-L}^{t-1}$ as input. The label maps are concatenated together and undergo several residual blocks to form intermediate high-level features. We apply the same processing for the previously generated images. Then, these two intermediate layers are added and fed into two separate residual networks to output the hallucinated image $\tilde{\mathbf{h}}_t$ as well as the flow map $\tilde{\mathbf{w}}_t$ and the mask $\tilde{\mathbf{m}}_t$ (Figure 8).

Next, to build from low-res results to higher-res results, we use another network G_2 on top of the low-res network G_1 (Figure 9). In particular, we first downsample the inputs and fed them into G_1. Then, we extract features from the last feature layer of G_1 and add them to the intermediate feature layer of G_2. These summed features are then fed into another series of residual blocks to output the higher resolution images.

A.2 Discriminators

For our image discriminator D_I, we adopt the multi-scale PatchGAN architecture [31, 70]. We also design a temporally multi-scale video discriminator D_V by downsampling the frame rates of the real/generated videos. In the finest scale, the discriminator takes K consecutive frames in the original sequence as input. In the next scale, we subsample the video by a factor of K (i.e., skipping every

$K-1$ intermediate frames), and the discriminator takes consecutive K frames in this new sequence as input. We do this for up to three scales in our implementation and find that this helps us ensure both short-term and long-term consistency. Note that D_V is also multi-scale in the spatial domain as D_I.

A.3 Feature matching loss

In our learning objective function, we also add VGG feature matching loss and discriminator feature matching loss to improve the training stability. For VGG feature matching loss, we use the VGG network [61] as a feature extractor and minimize L1 losses between the extracted features from the real and the generated images. In particular, we add $\sum_i \frac{1}{P_i}[||\psi^{(i)}(\mathbf{x}) - \psi^{(i)}(G(\mathbf{s}))||_1]$ to our objective, where $\psi^{(i)}$ denotes the i-th layer with P_i elements of the VGG network. Similarly, we adopt the discriminator feature matching loss, to match the statistics of features extracted by the GAN discriminators. We use both the image discriminator D_I and the video discriminator D_V.

B Evaluation for the Apolloscape Dataset

We provide both the FID and the human preference score on the Apolloscape dataset. For both metrics, our method outperforms the other baselines.

Table 4: Comparison between competing video-to-video synthesis approaches on Apolloscape.

Inception Net. of FID	I3D	ResNeXt
`pix2pixHD`	2.33	0.128
`COVST`	2.36	0.128
`vid2vid` (ours)	**2.24**	**0.125**

Human Preference Score	
`vid2vid` (ours) / `pix2pixHD`	**0.61** / 0.39
`vid2vid` (ours) / `COVST`	**0.59** / 0.41