Supplementary material for World-Consistent Video-to-Video Synthesis

Arun Mallya*, Ting-Chun Wang*, Karan Sapra, and Ming-Yu Liu

NVIDIA
{amallya,tingchunw,ksapra,mingyul}@nvidia.com

A Objective functions

Our objective functions contain five losses: an image GAN loss, a video GAN loss, a perceptual loss, a flow-warping loss, and a world-consistency loss. Except for the world-consistency loss, the others are inherited from the vid2vid [5]. Note that we replace the least square losses used in the vid2vid for GAN losses with the hinge losses as used in SPADE [3]. We describe these terms in details in the following.

GAN losses. Let $\mathbf{s}_1^T \equiv \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_T\}$ be a sequence of input semantic frames. Let $\mathbf{x}_1^T \equiv \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ be the sequence of corresponding real video frames, and $\tilde{\mathbf{x}}_1^T \equiv \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, ..., \tilde{\mathbf{x}}_T\}$ be the synthesized frames by our generator. Define $(\mathbf{x}_t, \mathbf{s}_t)$ as one pair of frames at a particular time instance where $\mathbf{x}_t \in \mathbf{x}_1^T$ and $\mathbf{s}_t \in \mathbf{s}_1^T$. The image GAN loss ($\mathcal{L}_I^t$) and the video GAN loss ($\mathcal{L}_V^t$) for time t are then defined as

$$\begin{aligned}
\mathcal{L}_I^t =& E_{(\mathbf{x}_t, \mathbf{s}_t)}[\min(0, -1 + D_I(\mathbf{x}_t, \mathbf{s}_t))] + && (1)\\
& E_{(\tilde{\mathbf{x}}_t, \mathbf{s}_t)}[\min(0, -1 - D_I(\tilde{\mathbf{x}}_t, \mathbf{s}_t)] && (2)\\
\mathcal{L}_V^t =& E_{\mathbf{x}_{t-K+1}^t}[\min(0, -1 + D_V(\mathbf{x}_{t-K+1}^t)] + && (3)\\
& E_{\tilde{\mathbf{x}}_{t-K}^{t-1}}[\min(0, -1 - D_V(\tilde{\mathbf{x}}_{t-K}^{t-1}))] && (4)
\end{aligned}$$

where D_I and D_V are the image and video discriminators, respectively. The video discriminator takes K consecutive frames and concatenates them together for discrimination. For both GAN losses, we also accompany them by the feature matching loss ($\mathcal{L}_{FM}^t$) as in pix2pixHD [6],

$$\mathcal{L}_{FM,I/V}^t = \sum_i \frac{1}{P_i} \left[||D_{\{I/V\}}^{(i)}(\mathbf{x}_t) - D_{\{I/V\}}^{(i)}(\tilde{\mathbf{x}}_t)||_1 \right], \tag{5}$$

where $D_{\{I/V\}}^{(i)}$ denotes the i-th layer with P_i elements of the discriminator network D_I or D_V.

* Equal contribution

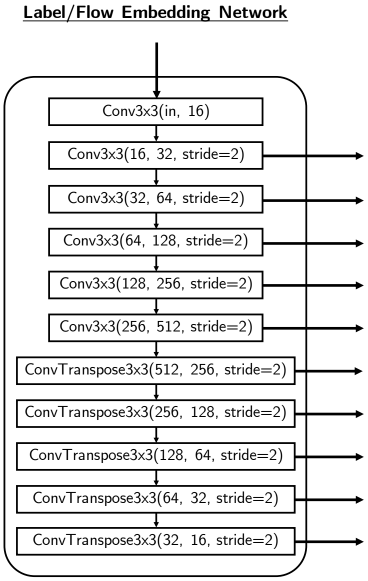
Label/Flow Embedding Network
Conv3x3(in, 16)
Conv3x3(16, 32, stride=2)
Conv3x3(32, 64, stride=2)
Conv3x3(64, 128, stride=2)
Conv3x3(128, 256, stride=2)
Conv3x3(256, 512, stride=2)
ConvTranspose3x3(512, 256, stride=2)
ConvTranspose3x3(256, 128, stride=2)
ConvTranspose3x3(128, 64, stride=2)
ConvTranspose3x3(64, 32, stride=2)
ConvTranspose3x3(32, 16, stride=2)

Fig. A.1: Label / flow-warped image embedding network.

Perceptual loss. We use the VGG-16 network [4] as a feature extractor and minimize L1 losses between the extracted features from the real and the generated images. In particular,

$$\mathcal{L}_P^t = \sum_i \frac{1}{P_i} \left[||\psi^{(i)}(\mathbf{x}_t) - \psi^{(i)}(\tilde{\mathbf{x}}_t)||_1 \right], \tag{6}$$

where $\psi^{(i)}$ denotes the i-th layer of the VGG network.

Flow-warping loss. We first warp the previous frame to the current frame using optical flow. We then encourage the warped frame to be similar to the current frame by using an L1 loss,

$$\mathcal{L}_F^t = ||\tilde{\mathbf{x}}_t - \mathbf{w}_t(\tilde{\mathbf{x}}_{t-1})||_1 \tag{7}$$

where $\mathbf{w}_t$ is the warping function derived from optical flow.

World-consistency loss. Finally, we add the world consistency by enforcing the generated image to be similar to our guidance image. It is achieved by

$$\mathcal{L}_{WC}^t = ||\tilde{\mathbf{x}}_t - \tilde{\mathbf{g}}_t||_1 \tag{8}$$

where $\tilde{\mathbf{g}}_t$ is our estimated guidance image.

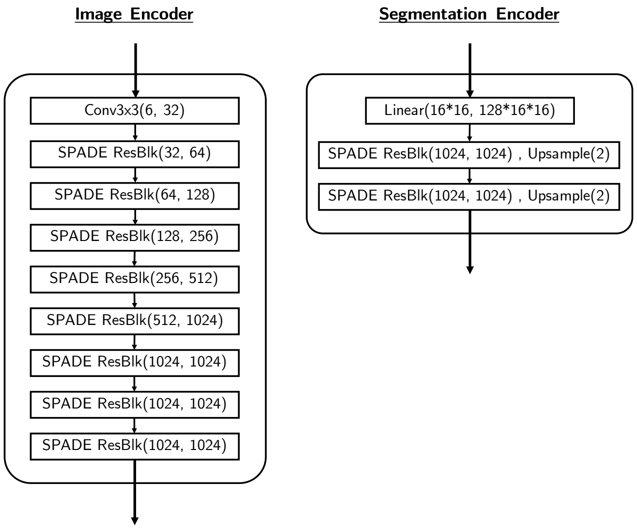

Fig. B.1: Previous image / segmentation encoder.

The overall objective function is then

$$\mathcal{L} = \sum_t \min_G \left(\max_{D_I, D_V} (\lambda_I \mathcal{L}_I^t + \lambda_V \mathcal{L}_V^t) \right) + \qquad (9)$$

$$\min_G \left(\lambda_{FM} \mathcal{L}_{FM}^t + \lambda_P \mathcal{L}_P^t + \lambda_F \mathcal{L}_F^t + \lambda_W \mathcal{L}_{WC}^t \right) \qquad (10)$$

where λ are the weights for each individual terms, which are set to 1, 1, 10, 10, 10, 10 in all of our experiments.

Optimization details. We use the ADAM optimizer [2] with $(\beta_1, \beta_2) = 0, 0.999$ for all experiments and network components. We use a learning rate of 1e-4 for the encoder and generator networks (which are described below) and 4e-4 for the discriminators.

B Network architecture

As described in the main paper, our framework contains four components: a label embedding network (Fig. A.1), an image encoder (Fig. B.1), a flow embedding network (Fig. A.1), and an image generator (Fig. B.2).

Label embedding network (Fig. A.1). We adopt an encoder-decoder style network to embed the input labels into different feature representations, which are then fed to the Multi-SPADE modules in the image generator.

Image / segmentation encoder (Fig. B.1). These networks generate the input to the main image generator. The segmentation encoder is used when

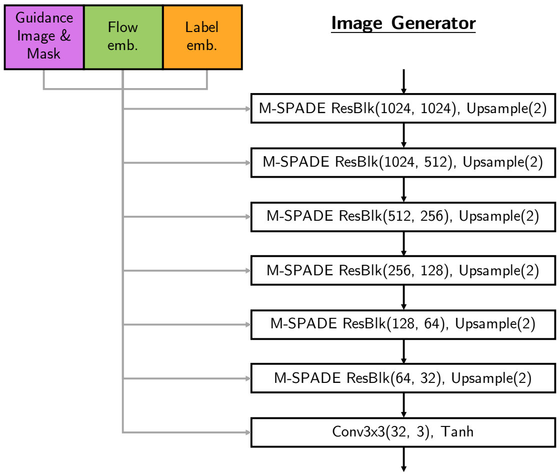
Guidance Image & Mask
Flow emb.
Label emb.
Image Generator
M-SPADE ResBlk(1024, 1024), Upsample(2)
M-SPADE ResBlk(1024, 512), Upsample(2)
M-SPADE ResBlk(512, 256), Upsample(2)
M-SPADE ResBlk(256, 128), Upsample(2)
M-SPADE ResBlk(128, 64), Upsample(2)
M-SPADE ResBlk(64, 32), Upsample(2)
Conv3x3(32, 3), Tanh

Fig. B.2: Image generator.

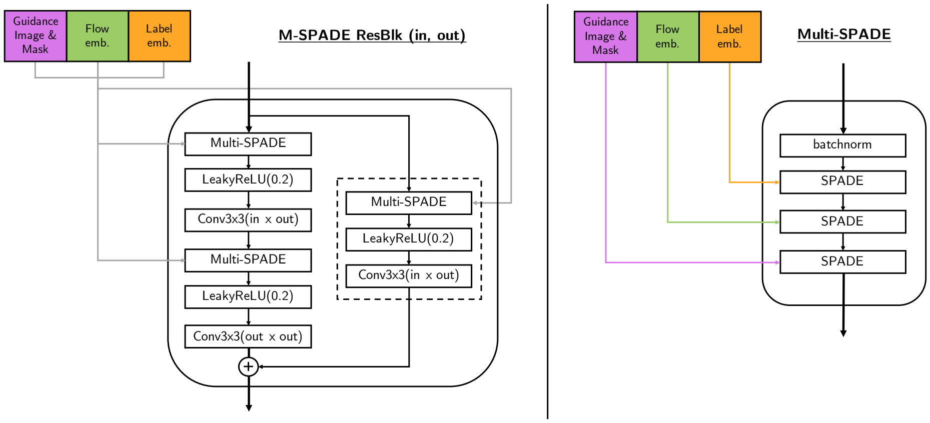

Fig. B.3: Multi-SPADE Residual Block and Multi-SPADE module.

generator the first frame in the sequence, while the image encoder is used when generating the subsequent frames. The segmentation encoder encodes the input semantics of the first frame, while the image encoder encodes the previously generated frame.

Flow embedding network (Fig. A.1). It is used to embed the optical flow-warped previous frame, which adopts the same architecture as the label embedding network except for the number of input channels. The embedded features are again fed to the Multi-SPADE layers in the main image generator.

Image generator (Fig. B.2). The generator consists of a series of Multi-SPADE residual blocks (M-SPADE ResBlks) and upsampling layers. The structure of each M-SPADE Resblk is shown in Fig. B.3, which replaces the SPADE layers in the original SPADE Resblks with Multi-SPADE layers.

Discriminators. We use the same image and video discriminators as vid2vid [5].

C Additional Results

Short-term temporal video consistency. For each sequence, we first take two neighboring frames from the ground truth images to compute the optical flow between them using FlowNet2 [1]. We then use the optical flow to warp the corresponding synthesized images and compute the L1 distance between the warped image and the target image, in RGB space, normalized by the number of pixels and channels. This process is repeated for all pairs of neighboring frames in all sequences and averaged. The result is shown in below in Table 1. As can be seen, Ours w/o World Consistency (W.C.) consistently performs better than vid2vid [5], and Ours (with world consistency) again consistently outperforms Ours w/o W.C.

Table 1: Short-term temporal consistency scores. Lower is better.

Dataset	vid2vid [5]	Ours w/o W.C.	Ours
Cityscapes	0.0036	0.0032	**0.0029**
MannequinChallenge	0.0397	0.0319	**0.0312**
ScanNet	0.0351	0.0278	**0.0192**

References

1. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
2. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
3. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
4. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
5. Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B.: Video-to-video synthesis. In: Conference on Neural Information Processing Systems (NeurIPS) (2018)
6. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)