Supplementary Material for
“Generative Video Transformer: Can Objects be the Words?”

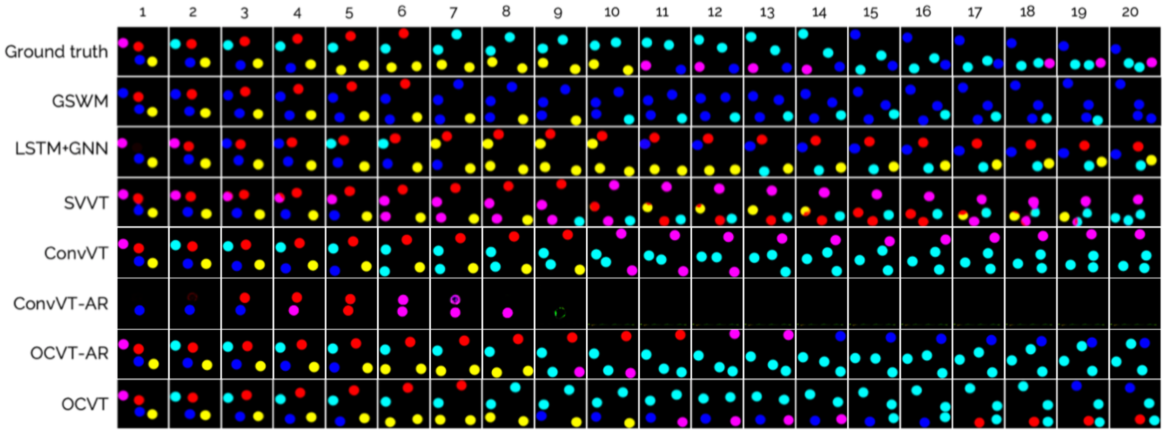

Figure A1. Qualitative generation results for Mod2 dataset. Given the first 50 frames of the video, each model predicts the next 50 frames. The first 20 predicted frames are shown.

A. Additional Results

A.1. Qualitative Generation Results

Figures A1, A2, A3 show qualitative generation results for the Mod2, Mod3, and Mod1234 datasets.

A.2. Attention Analysis

Figure A4 shows an example of the attention weights in the transformer when predicting the last timestep of a sequence in the Mod1 dataset. The right hand side shows the balls at timestep t and the left hand side shows the balls for the 6 timesteps prior. The darker the shade of gray, the stronger the weight. At this particular timestep, the color of the top right ball changes from red to cyan. We see that the strongest attention weights are to the same ball in the previous frames as well as the violet ball several frames prior, which is the ball that last interacted with this ball. This makes intuitive sense because the positions of the same ball in the last few frames are important in predicting to updated location of the ball at the next timestep. In order to correctly predict the color change of the ball, it must also attend to the ball that it most recently interacted with.

A.3. End-to-End Training

We also evaluated OCVT in a setting where we train the entire model end-to-end instead of freezing the parameters of the encoder and decoder while training the transformer. This is done under two settings: (a) training the model completely from scratch end-to-end and (b) using a pre-trained encoder and fine-tuning the model end-to-end. We achieve a next-step change accuracy of 82.21% for (a) and 76.95% for (b) for the Mod1 dataset. While this end-to-end training does not outperform our best pre-trained model, end-to-end training may be beneficial in certain scenarios since the encoder can incorporate temporal information from the scene. We leave this investigation for future work.

B. Implementation Details

B.1. Model Architecture

For the foreground image encoder, we use a ResNet18 (He et al., 2016). For $(H, W) = (4, 4)$, we apply an extra pair of 3×3 convolutions with stride 1 to get the appropriate dimensions per grid cell (see Hyperparameters in the next section). For $(H, W) = (8, 8)$, we remove the last ResNet block and then apply the pair of convolutions. To obtain $\mathbf{z}_t$, each cell is run through a 3-layer fully convolutional network with ReLU activation and group normalization (Wu & He, 2018). After the final layer, we apply softplus to compute standard deviations of the Gaussian distributions for $\mathbf{z}_t^{\text{where}}$, $\mathbf{z}_t^{\text{what}}$, $\mathbf{z}_t^{\text{depth}}$. For $\mathbf{z}_t^{\text{pres}}$, we apply the sigmoid function and use the Gumbel-Softmax (Jang et al., 2016) relaxation to model a Bernoulli random variable. For the foreground

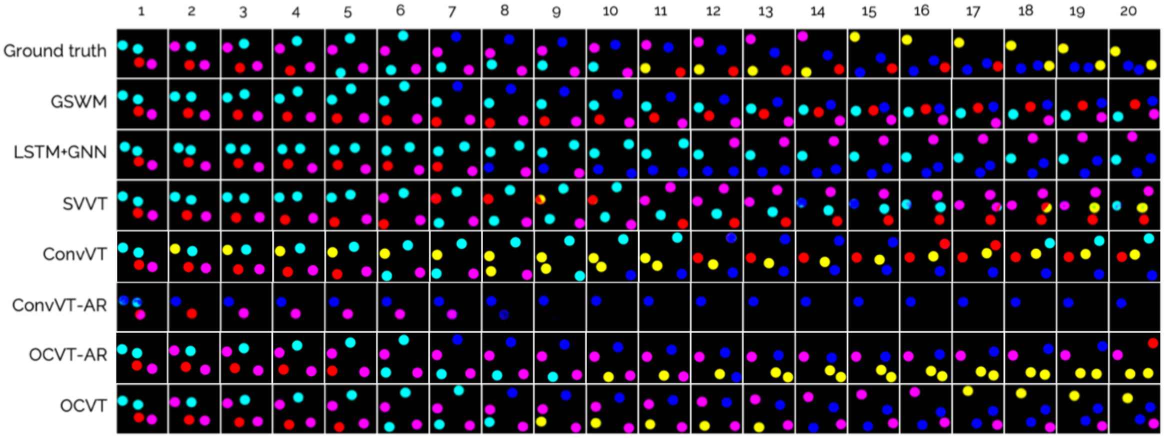

Figure A2. Qualitative generation results for Mod3 dataset. Given the first 50 frames of the video, each model predicts the next 50 frames. The first 20 predicted frames are shown.

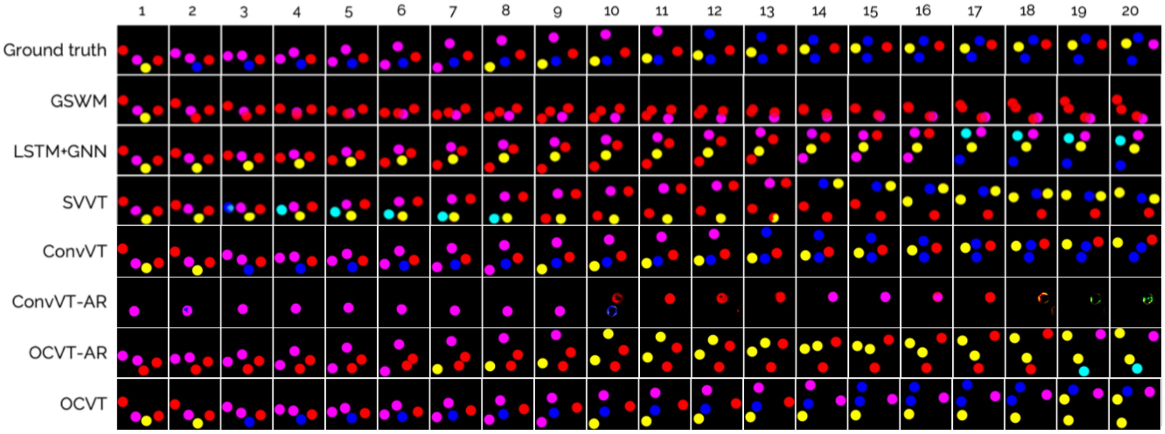

Figure A3. Qualitative generation results for Mod1234 dataset. Given the first 70 frames of the video, each model predicts the next 80 frames. The first 20 predicted frames are shown.

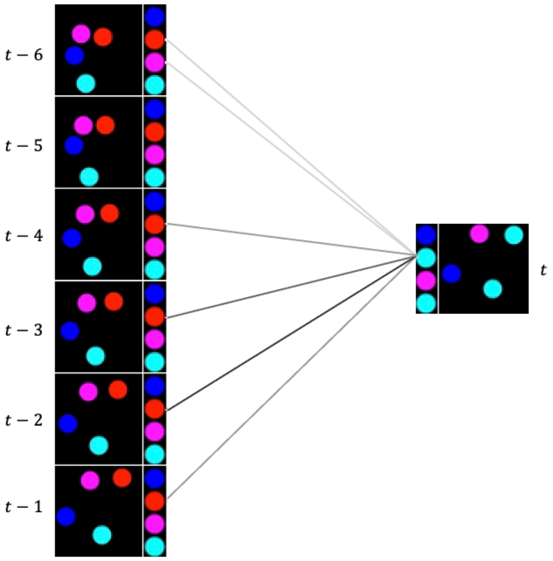

Figure A4. Attention strength at 7th layer during color change.

image decoder, we use a 6-layer sub-pixel convolutional network (Shi et al., 2016) with group normalization in the intermediate layers.

For the background image encoder, we use a 4-layer convolutional network with CELU activation (Barron, 2017) and group normalization followed by a final linear layer. For the background image decoder, we use a 6-layer convolutional network, each consisting of 2D bilinear upsampling followed by a convolution with leaky ReLU activation.

For the transformer, we use a linear layer to obtain the desired dimensions for the transformer input (see Hyperparameters). The output of the transformer runs through a single hidden layer MLP with ReLU activation to obtain the next step predictions.

B.2. Hyperparameters

We provide the hyperparameters used in our experiments in Tables A1 and A2.

Table A1. List of Hyperparameters for Bouncing Ball Datasets

Description	Value
Image Size	(64,64)
Grid size (H, W)	(4,4)
Dimension per grid cell	128
Dimension of $\mathbf{z}^{\text{what}}$	16
Dimension of $\mathbf{z}^{bg}$	None
Foreground Variance	0.2
Background Variance	None
Gumbel-Softmax Temp. for $\mathbf{z}_t^{\text{pres}}$	0.01
β_{where}	20
β_{depth}	0
β_{pres}	1
β_{what}	4
Dimension of transformer input	360
Feedforward dimension in transformer	256
Number of heads	8
Number of transformer layers	15

Table A2. List of Hyperparameters for CATER Datasets

Description	Value
Image Size	(64,64)
Grid size (H, W)	(8,8)
Dimension per grid cell	128
Dimension of $\mathbf{z}^{\text{what}}$	64
Dimension of $\mathbf{z}^{bg}$	64
Foreground Variance	0.05
Background Variance	0.2
Gumbel-Softmax Temp. for $\mathbf{z}_t^{\text{pres}}$	0.01
β_{where}	50
β_{depth}	1
β_{pres}	1
β_{what}	1
Dimension of transformer input	360
Feedforward dimension in transformer	256
Number of heads	6
Number of transformer layers	15

C. Dataset and Experiment Details

C.1. Bouncing Balls

In all the bouncing ball datasets, we have 20,000 videos for training, 200 videos for validation, and 200 videos for testing. For the Mod1, Mod2, and Mod3 datasets, each video has an episode length of 100 frames. For the Mod1234 dataset, each video has an episode length of 150 frames. This longer episode length is to allow for a sufficient number of interactions (up to 4 for this dataset) in the videos. We choose the best model based on the change accuracy on the validation set and then use this model on the test set for evaluation. All models are trained to convergence measured by the plateauing of the change accuracy on the validation set.

C.2. CATER

This dataset consists of 3,080 videos for the training set, 770 videos for the validation set, and 1650 videos for the test set. Each video frame is reshaped to 64x64 pixels. Each video originally has 300 frames and we randomly sample 50 frames for training. For validation and testing, we take every sixth frame for a total of 50 frames. We choose the best model based on the best Top 5 Accuracy for the snitch localization task on the validation set and then use this model on the test set for evaluation.

References

Barron, J. T. Continuously differentiable exponential linear units. *CoRR*, abs/1704.07483, 2017. URL http://arxiv.org/abs/1704.07483.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 1874–1883. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.207. URL https://doi.org/10.1109/CVPR.2016.207.

Wu, Y. and He, K. Group normalization. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, volume 11217 of *Lecture Notes in Computer Science*, pp. 3–19. Springer, 2018. doi: 10.1007/978-3-030-01261-8_1. URL https://doi.org/10.1007/978-3-030-01261-8_1.