Compositional Video Prediction (Supplementary)

Yufei Ye[1] Maneesh Singh[2] Abhinav Gupta[13*] Shubham Tulsiani[3*]
[1]Carnegie Mellon University [2]Verisk Analytics [3]Facebook AI Research
{yufeiy2,abhinavg}@cs.cmu.edu maneesh.singh@verisk.com shubtuls@fb.com

https://judyye.github.io/CVP/

A. Architecture Details

Entity Predictor. Our predictor leverages the graph neural network family, whose learning process can be abstracted to iterative message passing and message aggregation. In each round of message passing, each node (edge) is a parameterized function of their neighboring node and edges, which updates their parameters by back propagation. We introduce the predictor architecture by instantiating the message passing and aggregation operation as following:

For the l-th layer of message passing, it consists of two operations:

$$v \to e : e_{i,j}^{(l)} = f_{v\to e}^{(l)}[v_i^{(l)} \oplus v_j^{(l)}]$$
$$e \to v : v_i^{(l+1)} = f_{e\to v}^{(l)}[\mathrm{POOL}[e_{i,j}^{(l)}|(i,j)$$

We first perform node-to-edge passing $f_{v\to e}^{(l)}$ where edge embeddings are implicitly learned. Then we perform edge-to-node $f_{v\to e}^{(l)}$ operation given the updated edge embeddings. The message passing block can be stacked to arbitrary layers to perform multiple rounds of message passing between edge and node. In our experiment, we stack four blocks of the above module. For each block, $f_{v\to e}^{(l)}, f_{e\to v}^{(l)}$ are both implemented as a single fully connected layer. The aggregation operator is implemented as a average pooling. Note that connection expressed in the edge set can be either from explicitly specified graph, or a fully connected graph when the relationship is not explicitly observed.

Frame Decoder. We use the backbone of Cascaded Refinement Networks. Given feature in shape of (N, D, h_0, w_0) either from entity predictor or background feature, the frame decoder upsamples it at the end of every unit. Each unit comprises of $Conv \to Batch \to LeakyRelu$. When the entity features are warped to image coordinates, the spatial transformation is implemented as a forward transformation to sharpen entities.

Latent Encoder. At training, the encoder takes in the concatenated features of two frames and apply a one layer neural network to obtain mean and variance of u, where we resample with reparameterization trick at training time. The resampled u' is fed into a one-layer LTSM as cell unit to generates a sequence of z^t.

Training Details. We optimize the total loss with Adam optimizer in learning rate $1e-4$. $\lambda_1 = 100, \lambda_2 = 1e-3$. The dimensionality of latent is 8, i.e. $|u| = |z^t| = 8$. Location feature is represented as the center of entities $|b| = 2$, appearance feature $|a| = 32$. The region of each entity is set to a large enough fixed width and height to cover the entity, $d = 70$ in all of our experiment. All generated frame are in resolution of 224×224.

B. Dataset

In Shapestacks, the 'entities' correspond to distinct objects, among which the graph used for interaction is fully connected since no explicit relationships are observed. The videos are generated by simulating the given initial configurations in in mujoco [1] for 16 steps. While the setting is deterministic under perfect state information (precise 3D position and pose, mass, friction, etc), the prediction task is ambiguous given an image input. The subset is split to 1320 clips for training, 281 clips for validation, and 296 clips for testing. When we evaluate the generalization ability, the test set further includes 221 (136 / 93) clips of videos comprised of 4 (5 / 6) blocks.

In Penn Action, 'entities' correspond to joints of human body and the graph is built based on prior knowledge of skeletons. If some joint is missing in the video, we instead link the edge to its parent if possible. We train our model to generate video sequences of 1 second at 8 fps given an initial frame. The categories we used are bench press, clean and jerk, jumping jacks, pull up, push up, sit up, and squat. To reduce overfitting, we augment data on the fly, including randomly selecting the starting frame for each clip, random spatial cropping, etc.

* The last two authors were equally uninvolved.

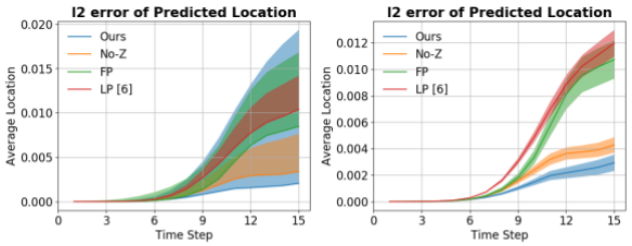

Figure 1. Left: For all 100 samples, σ (shade) and best samples (line at lower boundary of shade) are plotted. Right: For best 5 samples of 100, mean and σ are plotted.

C. Baseline Model

The No-Factor model does not predict a per-entity appearance but simply outputs a global feature that is decoded to foreground appearance and mask. To ensure the use of the same supervision as box locations, the No-Factor model also takes as input (and outputs) the per-entity bounding boxes. Thus, the foreground is represented as the extracted feature of the entire frame concatenated by all locations and they are directly predicted together with fully connected layers. To decode them to pixels, an additional binary mask is applied. However, no mechanism in No-Factor baseline guarantees the generated pixels of entities respect the predicted locations.

In the No-Edge baseline, we remove all but self-link edges between nodes so that all the nodes are predicted independently.

Pose-Knows [3] consists of two models: a Pose-VAE that takes input as the initial frame together with joint location and outputs joint location in the future, a Pose-GAN with skip layers that takes input as the initial frame together with rendered predicted poses and generate frames. The original work uses 3D convolutional [2] network to generate low resolution videos (80×64). However, with progress in GAN techniques in recent years [4], we find that 2D convolution with frame-wise adversarial loss improves performance when generating high resolution videos (224×224) in terms of both qualitative and quantitative evaluation.

D. Standard Deviation

Prediction task is multi-modal, a model that correctly handles uncertainty will predict diverse future states (and therefore should) in the error across samples (as $\sigma = 0$ implies mean prediction). while generate enough good samples. We plot the σ in location error (Shapestacks) over 100 samples in Figure 1 (Left). We also report the σ across top 5 samples in Figure 1 (right)

References

[1] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, 2012. 1

[2] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

[3] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *ICCV*, 2017. 2

[4] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 2