Contents

A. Method

A.1. Memory Efficiency

A.1.1 GHVAE

Because GHVAEs optimize each module with regard to image reconstruction, we must include in memory both the current module and some of the prior modules. Here, we briefly describe the memory savings of GHVAEs. GHVAEs save GPU or TPU memory allocation by avoiding the need to store gradient information in previous modules during back-propagation. Specifically, for the encoder, intermediate activations and all gradients from the frozen modules no longer need to be stored in memory. For the decoder, the gradients of the activations will still need to be stored for backpropagation into the currently trained module. Table 8 quantifies the amount of GPU or TPU memory saved for 1 to 6-module GHVAE models. This table indicates that the memory savings of a GHVAE model increases as the number of modules increases.

A.1.2 Other Methods

While GHVAEs alleviate the challenge of training large-scale video prediction models in the face of GPU or TPU memory constraints, there are other ways of addressing this challenge, such as increasing the number of GPUs or TPUs (as opposed to increasing the memory capacity per GPU or TPU), having different examples on different GPUs, and allocating model weights across more than one GPUs. Our method is orthogonal and complementary

Model Parameter	Value					
Number of Modules K	1	2	3	4	5	6
End-to-End Training Memory Usage (GB)	3.44	4.63	5.79	13.57	19.99	28.34
Greedy Training Memory Usage (GB)	3.44	3.46	4.23	9.20	13.60	17.05
Memory Saved (% Greedy Training Memory)	0%	33.8%	36.9%	47.5%	47.0%	66.2%

Table 8: **GPU or TPU Memory Usage of GHVAE Models.** All numbers are computed on a batch size of 1 per GPU, a rollout horizon of 10, two context frames, and $64 \times 64 \times 3$ image observations.

to such directions. Also, while increasing the number of GPUs or TPUs can increase the training batch size, our method can still allow larger models to be trained even after batch size per GPU lowers to 1.

It is also important to note that greedy training leads to higher optimization stability for GHVAEs in particular, as revealed in Ablation 2 of Table 4 in the main paper. Ablation 2 indicates that when GHVAEs are trained end-to-end from scratch, the model was unable to converge to any good performance in any single run compared to the greedy setting. GPU or TPU memory saving is only one of the benefits of performing greedy training.

A.2. Architecture

A.2.1 Context Frames

To incorporate context frames into the architecture of GHVAE, all context frames are first encoded individually using GHVAEs' encoders $\mathcal{W}_{enc}^{1:K}$. These embeddings are then concatenated channel-wise to the hidden variable h_t^k as input into the prior network, and the hidden variable h_{t+1}^k as input into the posterior inference network.

Concretely, let t denote the number of context frames available, then for any timestep t', the prior variable $h_{t'}^k$ and the posterior variable $h_{t'+1}^k$ are calculated as:

$$h_{t'}^k = \{\mathcal{W}_{enc}^{1:K}(x_{t'}), \mathcal{W}_{enc}^{1:K}(x_1), \ldots, \mathcal{W}_{enc}^{1:K}(x_t)\} \tag{5}$$

A.2.2 Action Conditioning

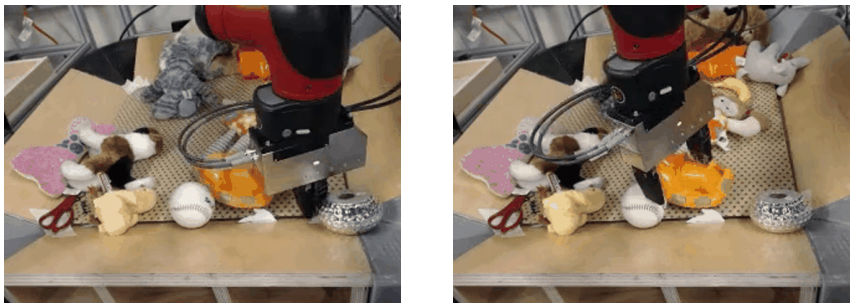
To incorporate action into the inputs of the prior network, the action vector is tiled spatially along both the height and width dimensions, such that the vector becomes a three-dimensional feature map per image. This action feature map is then concatenated to h_t^k similar to the context frames.

A.2.3 Prior vs. Posterior Variables

This section is to further clarify that given the current timestep t, all variables in the paper with subscript t are prior variables, such as h_t^k and z_t^k, since these variables are encoded from the current observation x_t. In contrast, all variables in the paper with subscript $t+1$ are posterior variables, such as h_{t+1}^k and z_{t+1}^k, because these variables are encoded from the future observation x_{t+1}.

A.3. Intuition

In this section, we elaborate on the main paper's intuition on why it is important to capture the multi-level stochasticity of future observations in video prediction. Shown in Fig. 4 is an example of a current and next image observation from RoboNet. In action-conditioned video prediction for RoboNet, the video prediction model is given a four-dimensional vector $[dx, dy, dz, gripper]$, in which dx, dy, dz denote the future end-effector translation from the current position, and $gripper$ is a binary integer for opening ($gripper = 0$) or closing ($gripper = 1$) the gripper. To accurately predict the next image observation, the video prediction model needs to precisely capture the end-effector position from the current monocular image, so that given the expected end-effector translation, the model can predict the new end-effector position and reconstruct all pixels that belong to the robot in the next image accordingly. The current end-effector position is considered a high-level visual feature that has inherent stochasticity because it is difficult to measure how long an inch is in this monocular image and therefore challenging to predict the precise pixel location of the robot in the next timestep. In addition, as the robot moves to a new position, the pixels currently occluded by the robot's arm will be revealed, and yet it is highly uncertain what is behind the robot's arm, let alone to predict these pixels for the next timestep. Concretely, there could be one or more objects



Current Image Observation Next Image Observation

Figure 4: **Example pair of current and next image in robotic manipulation.**

behind the robot arm or zero objects. In the case where there are one or more objects, the ground truth texture and orientation of these objects are almost entirely occluded and unknown. These are the uncertainties around the low-level features in the image. In summary, multiple levels of uncertainty exist in the current image (from the high-level representation of end-effector position to the lower-level texture of the occluded objects and table background), therefore demanding the video prediction model to accurately model such multi-level stochasticity of future observations with hierarchical architectures.

As a side note, in the main paper, we posit that "VAEs treat each dimension of a stochastic latent variable as independent". Here, this statement refers to the case where the VAE uses a diagonal multivariate Gaussian distribution to model the latent variable distribution, which applies to GHVAEs as well.

B. Experiments

B.1. Video Prediction

In this section, we visualize qualitative results and discuss how we calculate each performance metric for video prediction, how we perform human evaluation using Amazon Mechanical Turk, and additional ablation studies.

B.1.1 Visualizations

Figure 5, 6, 7, 8 exhibits example rollouts from video prediction methods reported in the main paper. Figure 9 and 10 are the example rollouts from real-robot experiments: Pick&Sweep and Pick&Wipe tasks.

B.1.2 Performance Evaluation

Methodologies for calculating performance metrics are available at Table 9. Note that these methodologies match those reported in prior works so that experiments conducted in this paper provide fair comparisons.

Dataset	Action-free / Action-conditioned	Batch Size	# of Context Frames	Rollout Horizon	GPU Memory Usage (GB)	# of GPUs	FVD Batch Size
RoboNet	Action-conditioned	32	2	10	24	4	256
KITTI	Action-free	32	5	25	24	4	148
Human3.6M	Action-free	32	5	25	24	4	256
CityScapes	Action-free	128	2	28	16	8	256
Real-Robot Experiments	Action-conditioned	140	2	10	24	4	256

Table 9: GPU Memory Usage for All Experiments in Table 1 and Table 2. All Convolutional Layers in the 6-Module GHVAE model for CityScapes are Downsized by 40% to fit into 16GB GPU Memory for Fair Comparison.

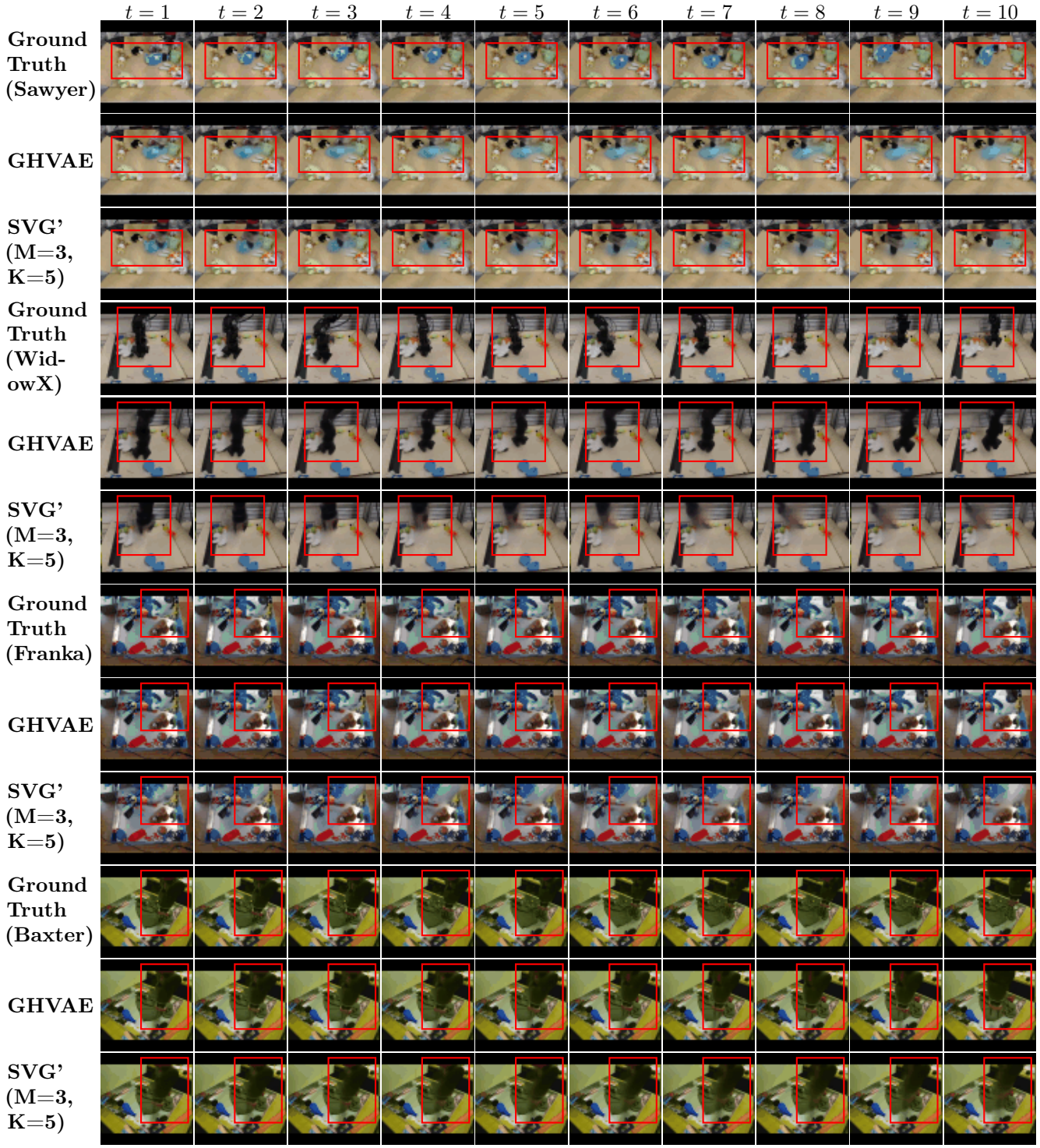


Figure 5: RoboNet Video Prediction. Specifically, we provide examples for various physical robots in RoboNet: Sawyer, WidowX, Franka, and Baxter. Both GHVAE and SVG' (M=3, K=5) are given the same two context images. Here, a 6-module GHVAE model exhibits visible performance superiority over SVG' (M=3, K=5) on generating realistic object (Sawyer) and robot movements (WidowX, Franka, Baxter). The red boxes highlight the differences.

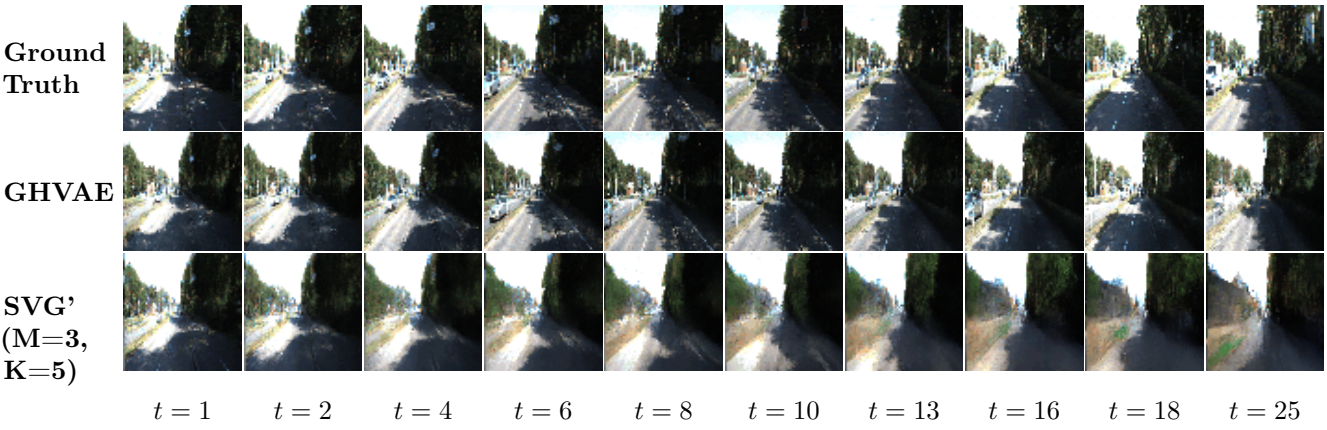


Figure 6: KITTI Driving Video Prediction. Both GHVAE and SVG' (M=3, K=5) are given the same five context images. Here, a 6-module GHVAE model exhibits performance advantage over SVG' (M=3, K=5).

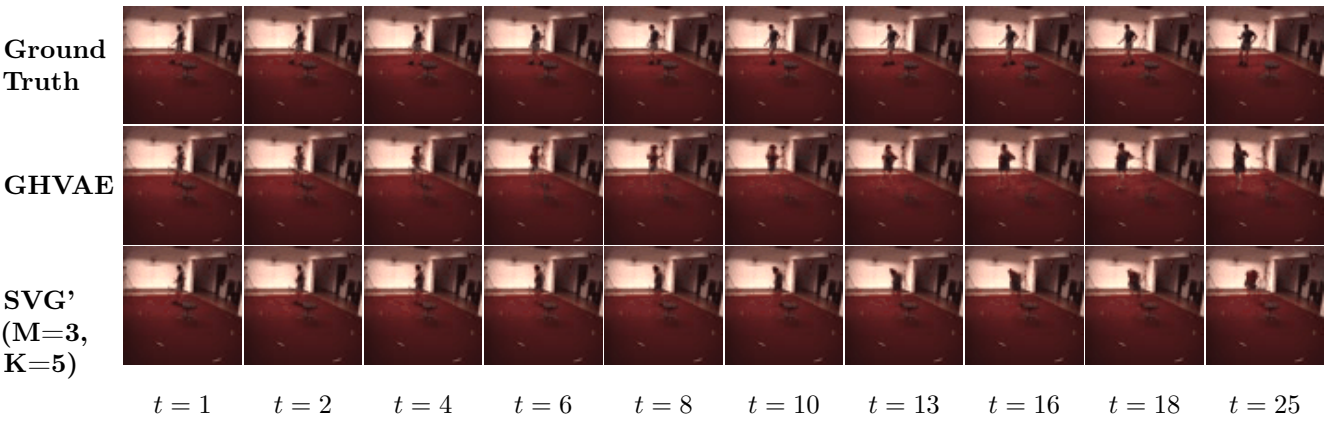


Figure 7: Human3.6M Video Prediction. Both GHVAE and SVG' (M=3, K=5) are given the same five context images. Here, a 6-module GHVAE model exhibits performance advantage over SVG' (M=3, K=5).

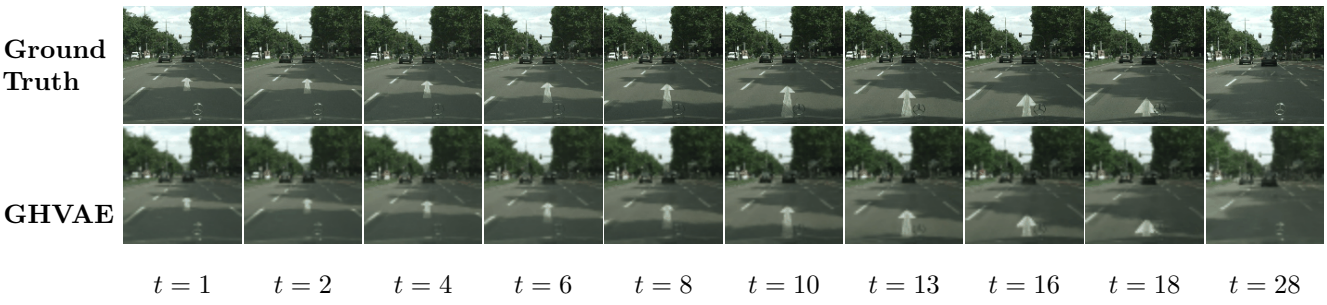


Figure 8: Cityscapes Driving Video Prediction. Both GHVAE and Hier-VRNN are given the same two context images. Here, a 6-module GHVAE model exhibits performance advantage over Hier-VRNN. Note that this paper directly compares to Hier-VRNN results reported in Castrejon et al. [11] and does not re-implement the Hier-VRNN algorithm.

B.1.3 Human Evaluation

For human evaluation, we provide 300 videos from both GHVAE and SVG' to Amazon Mechanical Turk workers in the form of 300 tasks. In each task, the workers are presented with three videos: a video generated by GHVAE,

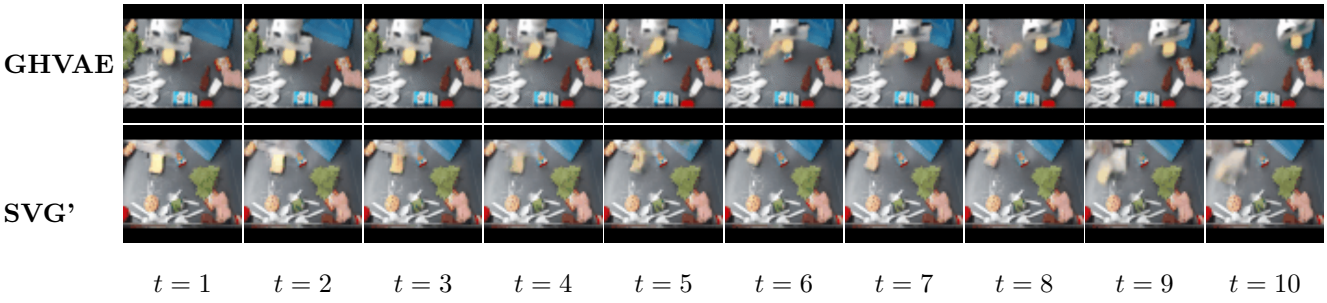


Figure 9: Video Prediction in Real-Robot Pick&Sweep Tasks. Both GHVAE and SVG' are given the same two context images. Here, GHVAE exhibits performance advantage over SVG'. Note that due to our random shooting planning strategy, the rollout length of each method is variable and different in every trial. Kindly see Appendix B.2.5 for more details.

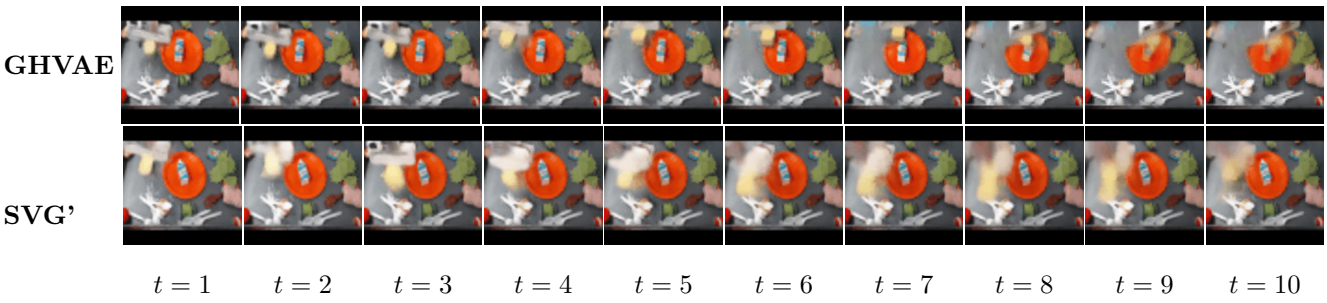


Figure 10: Video Prediction in Real-Robot Pick&Wipe Tasks. Both GHVAE and SVG' are given the same two context images. Here, GHVAE exhibits performance advantage over SVG'. Note that due to our random shooting planning strategy, the rollout length of each method is variable and different in every trial. Kindly see Appendix B.2.5 for more details.

a video generated by SVG', and the ground truth video. The worker does not know which video is generated by GHVAE or SVG', but do know which one is the ground truth video. In each task, the workers are asked to select the video that is more realistically similar to the ground truth video. These selections count as preferences. We then average all preferences and report results in Table 1 and 2.

B.1.4 Ablation for Encoder-Decoder Architectures

In the early stages of our research, we have experimented with an alternative encoder-decoder architecture that expands or keeps the spatial dimension constant while reducing the channel dimension instead. The empirical performance of doing so significantly underperforms the current GHVAE architecture, which reduces spatial dimensions iteratively and compensates this dimensionality reduction by expanding the channel dimension. As mentioned in the paper, we hypothesize that reducing the spatial dimensions allows GHVAEs to perform better mean-field approximation in the deepest latent space.

B.1.5 Ablation for Single vs. Multiple Latents

In this section, we provide further intuition for the tradeoff between using single vs. multiple latent variables in a K-module GHVAE. Using multiple latent variables for GHVAE is an obvious option that we have empirically experimented with without satisfying results. Experimentally, when the GHVAE model uses all K latent variables, the earlier latent variables provide suboptimal information and undesirably noisy signals to the overall network because of their inability to perform high-fidelity mean-field approximation when the spatial dimensions are large. This empirical phenomenon motivated us to only use the deepest latent variable in a GHVAE model. It is however important to note that using a single latent variable does not prevent GHVAEs from learning to accurately represent

the multi-level stochasticity of future pixels. One can model such multi-level stochasticity using a single latent variable, provided that the decoders learn to appropriately project stochasticity from a succeeding layer to a preceding layer via non-linear transformation. In summary, we designed the GHVAE model to contain a single level of stochastic prediction, which is propagated through earlier deterministic layers to model multi-level stochasticity of future observations.

B.2. Real Robot

In this section, we elaborate on real-robot experimental setup and training data, visualizations of real-robot task execution and environments, and the random shooting planner we use to control the Franka robot.

B.2.1 Setup

In the first **Pick&Wipe** task, the robot needs to pick a wiping tool (e.g. sponge, table cloth, etc.) up and wipe all objects off the plate for cleaning using the wiping tool. Each of the 20 trials contains different plates, objects, and wiping tools all unseen during training, and there could be at most two objects on the plate. The initial physical locations of the plate, the objects on the plate, and the robot itself are all randomized except that the robot is above the wiping tool. At the beginning of each trial, the wiping tool is *not yet* in the robot's gripper, which makes the task more difficult. The task is considered successful if the robot picks the wiping tool up successfully and all objects are entirely wiped off the plate using the wiping tool within 50 timesteps.

In the second **Pick&Sweep** task, the robot is required to pick a sweeping tool (e.g. dustpan sweeper, table cloth, or dry sponge, etc.) up and sweep an object into the dustpan that is randomly placed in the bin. At the beginning of each trial, the sweeping tool is *not yet* in the robot's gripper, which makes the task difficult. When a sweeping tool is not present in the scene, the robot then needs to sweep the object into the dustpan using its gripper. Each of the 20 trials contains different dustpans, objects, and sweeping tools all unseen during training. The physical location of the dustpan is uniformly random, and the object and the robot are also arbitrarily placed except that the robot is above the sweeping tool. The task is determined successful if the target object is swept into the dustpan within 50 timesteps. When a sweeping tool is indeed present, pushing the object into the dustpan using the robot's gripper will be considered a failure. Only pushing the object using the sweeping tool will be considered successful. This requires the video prediction methods to detect whether a tool was used for sweeping in the goal image and act accordingly in the physical task.

B.2.2 Training Data

The video prediction models used for the real-robot experiments in this paper are not trained using the RoboNet dataset directly, but instead first pre-trained on RoboNet and then fine-tuned on a self-collected dataset of 5000 videos using the target Franka robot. Yet, this paper is about fitting video prediction models to large-scale datasets and this training scheme might seem to be contradicting with the main message. While the models can be trained directly on RoboNet, without fine-tuning on the 5000-video Franka dataset, the empirical task success rate is much lower for both GHVAE and SVG' on the target Franka environment due to unseen lighting conditions and camera viewpoint. On the other hand, if the models are only trained on the 5000-video dataset, the models easily overfit and fail to generalize to novel objects and tools. The purpose of large-scale video prediction is not to overfit a large dataset, but to learn powerful generalization such that the model can perform few-shot learning on the target environment using a small amount of data. Such a training scheme works in favor of learning large-scale video prediction, as opposed to defeating its purpose. Example environments for self-supervised training data collection are available at Fig. 14.

The collection of training data is entirely self-supervised. Concretely, the robot randomly interacts with the training objects in the bin for 2-3 minutes in episodes of 20 timesteps, before pushing the objects from the corners to the center of the bin, so that object interaction remains frequent.

B.2.3 Task Execution

Figure 11 and 12 exhibit example Pick&Sweep and Pick&Wipe trials of real-robot task execution using the GHVAE and SVG' methods. Real-robot execution videos are at https://sites.google.com/view/ghvae.

Figure 11: Real-Robot Task Execution in Pick&Sweep Experiments. Here, a 6-module GHVAE model exhibits more frequent successes than SVG'.

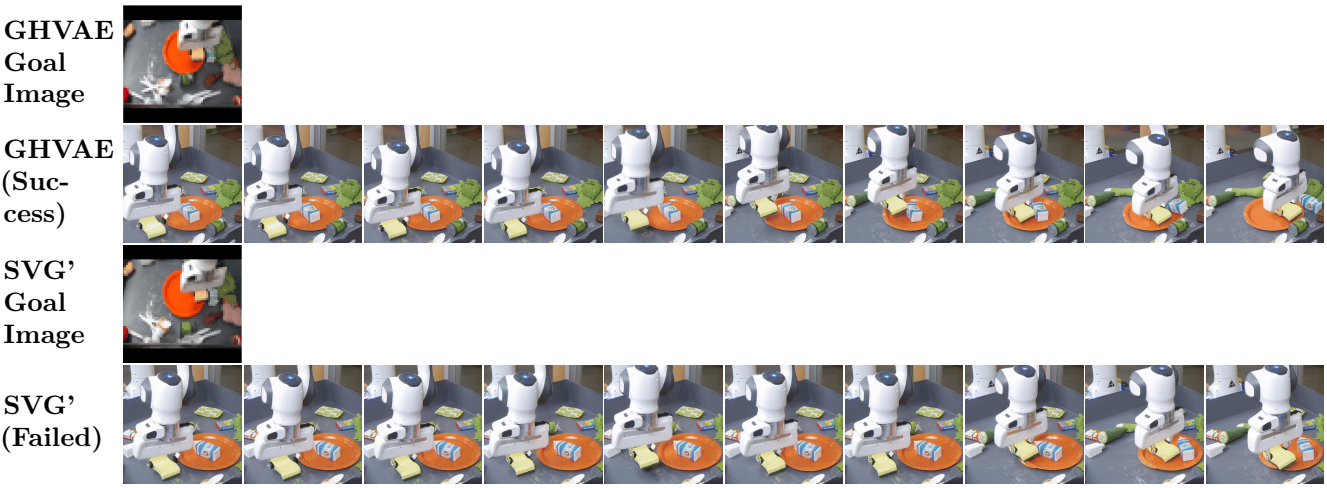


Figure 12: Real-Robot Task Execution in Pick&Wipe Experiments. Here, a 6-module GHVAE model exhibits more frequent successes than SVG'.

B.2.4 Task Diversity

In Figure 13, we visualize more environments and tools used for real-robot tasks to reveal the diversity of the evaluation tasks. All objects used for evaluation are unseen during training.

B.2.5 Planning

For simplicity, all real-robot experiments in this paper use a random shooting planner to optimize actions in visual foresight. Concretely, given a video prediction model and a goal image, we randomly sample a batch of 140 trajectories from the model and select the action sub-sequence for which the predicted images lead to the lowest L1 loss to the provided goal image. The robot replans after each execution of action sequences until the horizon of 50 timesteps is reached.

Concretely, the action space for the Franka robot has a dimension of 4 ($\mathcal{A} = \mathbb{R}^4$), which contains three scalars for the $[x, y, z]$ end-effector translation and one binary scalar for opening vs. closing its parallel-jaw gripper. Given the current image x_t, a goal image g, a sequence of t context images $x_{1:t}$ and a sampled action sequence $a_{t:t+T-1}$, the

Pick&Sweep Tasks

Pick&Wipe Tasks

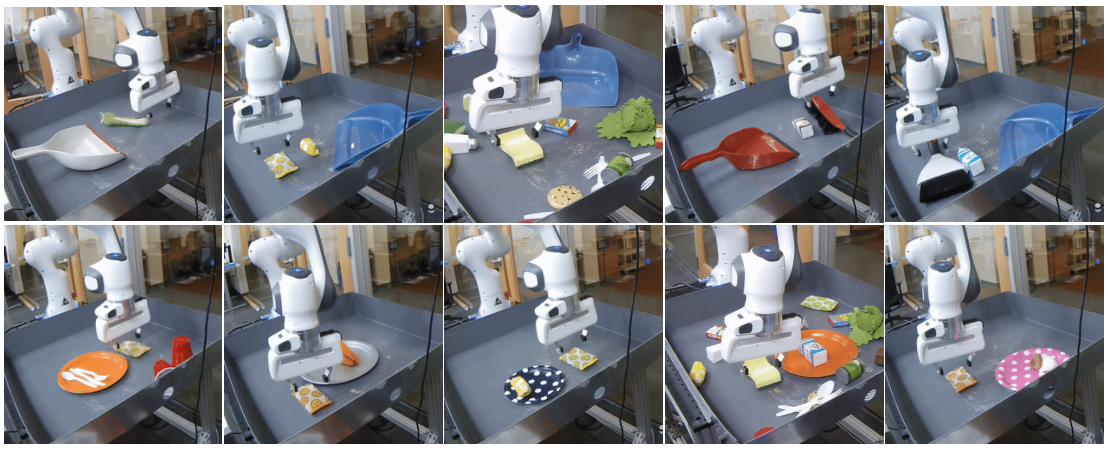


Figure 13: Sample Real-Robot Evaluation Tasks

Sample Training Environment

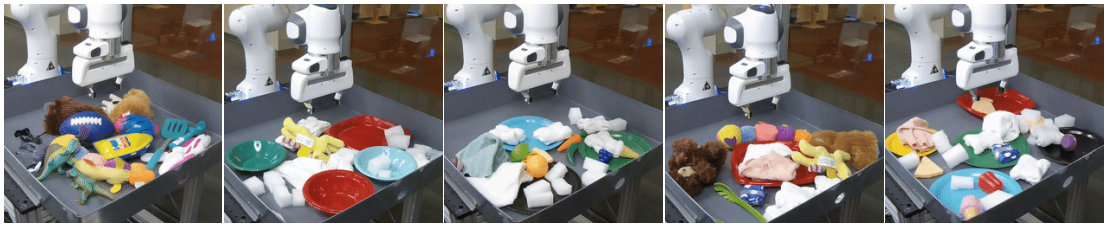


Figure 14: **Representative Real-Robot Training Environment.** Note that all objects used during training are excluded from evaluation. The 5000-video training data for both the Pick&Sweep and the Pick&Wipe tasks are the same.

sequence of frames predicted by the video prediction model f is:

$$\hat{x}_{t'+1} = f(\hat{x}_{t'}, a_{t'}, x_{1:t}) \tag{6}$$

where $t' \in [t, t+T-1]$, $\hat{x}_t = x_t$.

In practice, $T = 10$ for the Franka robot, and we sample a batch of 140 action sequences $\{a^1_{t:t+T-1}, \ldots, a^{140}_{t:t+T-1}\}$ and predicted frames $\{\hat{x}^1_{t+1:t+T}, \ldots, \hat{x}^{140}_{t+1:t+T}\}$.

Next, we calculate the optimal length of action sequence $T^* \in [1, T]$, and the best action sequence index $b^* \in [1, 140]$ using the following equation:

$$b^*, T^* = \underset{b\in[1,140], T'\in[1,T]}{\arg\min} |\hat{x}^b_{t+T'} - g| \tag{7}$$

Finally, the best action sequence is then calculated as: $a_{1:T^*} = a^{b^*}_{1:T^*}$. The robot then executes this T^*-timestep action sequence and repeats this planning procedure.

C. Mathematical Proofs

C.1. Proof of Theorem 1

Theorem 1 (ELBO Validity) *For any $k \in \mathbb{Z}^+$ and any set of frozen, greedily or end-to-end trained weights $\mathcal{W}^{1^*\ldots k-1^*}$,*

$$\begin{aligned} \log p(x_{t+1}) &\geq \max_{\mathcal{W}^{1\ldots k-1,k}} \mathcal{L}^k_{e2e}(x_{t+1}) \\ &\geq \max_{\mathcal{W}^k} \mathcal{L}^k_{greedy}(x_{t+1}) \end{aligned} \tag{3}$$

where $\mathcal{L}^k_{e2e}(x_{t+1})$ is GHVAE's ELBO for timestep $t+1$ when optimized end-to-end. More formally, $\mathcal{L}^k_{e2e}(x_{t+1})$ is $\mathcal{L}^k_{greedy}(x_{t+1})$ in Eq. 2, except that the VAE model $p^k \equiv p_{\mathcal{W}^{1\dots k-1,k}_{enc,dec,prior}}$ and the variational distribution $q^k \equiv q_{\mathcal{W}^{1\dots k-1,k}_{enc,post}}$.

Proof. Suppose $\mathcal{W}^{k^*}$ is the optimal parameters of the last module of a k-module GHVAE model:

$$\mathcal{W}^{k^*} = \arg\max_{\mathcal{W}^k} \mathcal{L}^k_{greedy}(x_{t+1}) \tag{8}$$

In other words:

$$\max_{\mathcal{W}^k} \mathcal{L}^k_{greedy}(x_{t+1}) = \mathcal{L}^k_{greedy}(x_{t+1}; \mathcal{W}^{k^*}) \tag{9}$$

Therefore:

$$\log p(x_{t+1}) \geq \max_{\mathcal{W}^{1\dots k}} \mathcal{L}^k_{e2e}(x_{t+1}) \geq \mathcal{L}^k_{greedy}(x_{t+1}; \mathcal{W}^{k^*}) = \max_{\mathcal{W}^k} \mathcal{L}^k_{greedy}(x_{t+1}) \tag{10}$$

C.2. Proof of Theorem 2

Theorem 2 (Monotonic Improvement) *For any $k \in \mathbb{Z}^+$ and any set of frozen, greedily or end-to-end trained weights $\mathcal{W}^{1^*\dots k-1^*}$,*

$$\begin{aligned} \log p(x_{t+1}) &\geq \mathcal{L}^k_{greedy}(x_{t+1}; \mathcal{W}^{1^*\dots k-1^*}) \\ &\geq \mathcal{L}^{k-1}(x_{t+1}; \mathcal{W}^{1^*\dots k-1^*}) \end{aligned} \tag{4}$$

Recall that:

$$\mathcal{L}^k_{greedy}(x_{t+1}) = \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\log p^k(x_{t+1} \mid x_t, z^k_{t+1})\right] - D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1}|x_t)\Big) \tag{11}$$

Steps in the following derivation that don't change from the previous step are in gray, while annotations are in blue.

$$\log p(x_{t+1}) \geq \mathcal{L}^k_{greedy}(x_{t+1}) \qquad \text{[Variation Lower-Bound]} \quad (12)$$

$$= \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\log p^k(x_{t+1} \mid x_t, z^k_{t+1})\right] - D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big) \qquad \text{[Eq. 11]} \quad (13)$$

$$= \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\log \int_{z^{k-1}_{t+1}} p^k(x_{t+1} \mid z^{k-1}_{t+1}, x_t, z^k_{t+1}) p^k(z^{k-1}_{t+1} \mid x_t, z^k_{t+1}) \frac{q^{k-1}(z^{k-1}_{t+1} \mid x_{t+1})}{q^{k-1}(z^{k-1}_{t+1} \mid x_{t+1})}\right]$$
$$- D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big) \qquad \text{[Algebra]} \quad (14)$$

$$= \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\log \mathbb{E}_{q^{k-1}(z^{k-1}_{t+1}|x_{t+1})}\left[\frac{p^k(x_{t+1} \mid z^{k-1}_{t+1}, x_t, z^k_{t+1}) p^k(z^{k-1}_{t+1} \mid x_t, z^k_{t+1})}{q^{k-1}(z^{k-1}_{t+1} \mid x_{t+1})}\right]\right]$$
$$- D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big) \qquad \text{[Algebra]} \quad (15)$$

$$\geq \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\mathbb{E}_{q^{k-1}(z^{k-1}_{t+1}|x_{t+1})}\left[\log p^k(x_{t+1} \mid z^{k-1}_{t+1}, x_t, z^k_{t+1}) + \log \frac{p^k(z^k_{t+1} \mid x_t, z^{k-1}_{t+1}) p^k(z^{k-1}_{t+1} \mid x_t)}{p^k(z^k_{t+1} \mid x_t)}\right.\right.$$
$$\left.\left. - \log q^{k-1}(z^{k-1}_{t+1} \mid x_{t+1})\right]\right] - D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big) \qquad \text{[Jensen's Inequality, Bayes' Rule]} \quad (16)$$

$$= \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\mathbb{E}_{q^{k-1}(z^{k-1}_{t+1}|x_{t+1})}\left[\log \frac{p^k(x_{t+1} \mid z^{k-1}_{t+1}, x_t) q^k(z^k_{t+1} \mid x_{t+1}, z^{k-1}_{t+1}, x_t)}{p^k(z^k_{t+1} \mid z^{k-1}_{t+1}, x_t)} + \log \frac{p^k(z^k_{t+1} \mid x_t, z^{k-1}_{t+1}) p^k(z^{k-1}_{t+1} \mid x_t)}{p^k(z^k_{t+1} \mid x_t)}\right.\right.$$
$$\left.\left. - \log q^{k-1}(z^{k-1}_{t+1} \mid x_{t+1})\right]\right] - D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big) \qquad \text{[Bayes' Rule]} \quad (17)$$

$$= \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\mathbb{E}_{q^{k-1}(z^{k-1}_{t+1}|x_{t+1})}\left[\log \frac{p^k(x_{t+1} \mid z^{k-1}_{t+1}, x_t) q^k(z^k_{t+1} \mid x_{t+1}, z^{k-1}_{t+1}, x_t) p^k(z^{k-1}_{t+1} \mid x_t)}{p^k(z^k_{t+1} \mid x_t)}\right.\right.$$
$$\left.\left. - \log q^{k-1}(z^{k-1}_{t+1} \mid x_{t+1})\right]\right] - D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big) \qquad \text{[Algebra]} \quad (18)$$

$$= \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\mathbb{E}_{q^{k-1}(z^{k-1}_{t+1}|x_{t+1})}\left[\log p^k(x_{t+1} \mid z^{k-1}_{t+1}, x_t) + \log p^k(z^{k-1}_{t+1} \mid x_t) - \log q^{k-1}(z^{k-1}_{t+1} \mid x_{t+1})\right.\right.$$
$$\left.\left. + \log q^k(z^k_{t+1} \mid x_{t+1}, z^{k-1}_{t+1}, x_t) - \log p^k(z^k_{t+1} \mid x_t)\right]\right] - D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big) \qquad \text{[Algebra]} \quad (19)$$

$$= \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\mathbb{E}_{q^{k-1}(z^{k-1}_{t+1}|x_{t+1})}\left[\log p^{k-1}(x_{t+1} \mid z^{k-1}_{t+1}, x_t) + \log p^{k-1}(z^{k-1}_{t+1} \mid x_t) - \log q^{k-1}(z^{k-1}_{t+1} \mid x_{t+1})\right.\right.$$
$$\left.\left. + \log q^k(z^k_{t+1} \mid x_{t+1}, z^{k-1}_{t+1}, x_t) - \log p^k(z^k_{t+1} \mid x_t)\right]\right] - D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big)$$
$$\text{[}z^{k-1}_{t+1}\text{ is independent of }p^k\text{ given }p^{k-1}\text{]} \quad (20)$$

$$= \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\mathcal{L}^{k-1}(x_{t+1}) + \mathbb{E}_{q^{k-1}(z^{k-1}_{t+1}|x_{t+1})}\left[\log q^k(z^k_{t+1} \mid x_{t+1}, z^{k-1}_{t+1}, x_t) - \log p^k(z^k_{t+1} \mid x_t)\right]\right]$$
$$- D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big) \qquad \text{[Eq. 11]} \quad (21)$$

$$= \mathcal{L}^{k-1}(x_{t+1}) + \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\log q^k(z^k_{t+1} \mid x_{t+1}) - \log p^k(z^k_{t+1} \mid x_t)\right] - D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1} \mid x_t)\Big)$$
$$\text{[Remove conditionally independent variables, Algebra]} \quad (22)$$

$$= \mathcal{L}^{k-1}(x_{t+1}) \qquad \text{[Algebra]} \quad (23)$$

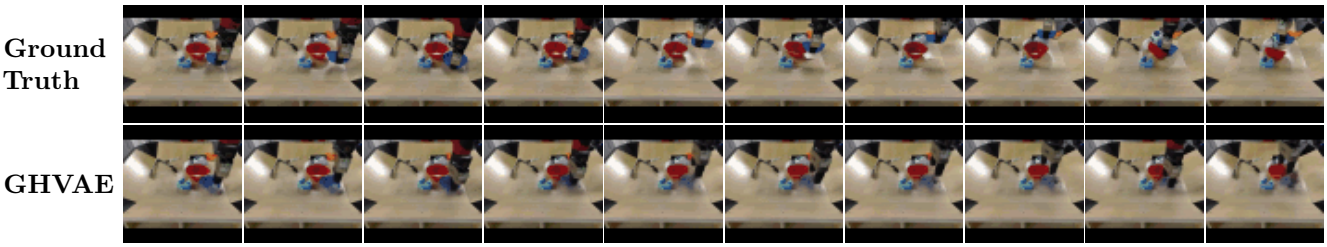


Figure 15: Failure case for a 6-module GHVAE model on RoboNet. In this case, the GHVAE model failed to accurately track the movement of the blue bowl. This indicates that the GHVAE model is still slightly underfitting on RoboNet. We hypothesize that training an 8-module, 10-module or 12-module GHVAE model will resolve such failure case.

where $\mathcal{L}^{k-1} \in \{\mathcal{L}^{k-1}_{greedy}, \mathcal{L}^{k-1}_{e2e}\}$ and $\mathcal{L}^{k}_{greedy}$ is initialized with the weights $\mathcal{W}^{1^*\ldots k-1^*}$. Notice that the proof above assumes action-free video prediction. The proof for action-conditioned video prediction is the same with every conditional variable x_t in the proof above expanding into two joint conditional variables x_t and a_t. For example, the term $p^k(x_{t+1} \mid x_t, z^k_{t+1})$ would be $p^k(x_{t+1} \mid x_t, a_t, z^k_{t+1})$ instead.

C.3. Clarification for Equation 2

Note that while Eq. 2 in the paper is an accurate mathematical form of GHVAE's ELBO, we have omitted a_t in the term $\log p^k(x_{t+1} \mid x_t, z^k_{t+1})$ in this equation since GHVAE in practice only uses a_t in the prior network. In other words, a more general form for Eq. 2 is the following:

$$\mathcal{L}^k_{greedy}(x_{t+1}) = \mathbb{E}_{q^k(z^k_{t+1}|x_{t+1})}\left[\log p^k(x_{t+1} \mid x_t, a_t, z^k_{t+1})\right] - D_{KL}\Big(q^k(z^k_{t+1} \mid x_{t+1}) \parallel p^k(z^k_{t+1}|x_t, a_t)\Big) \tag{24}$$

D. Failure Case Analysis

While a 6-Module GHVAE outperforms SVG' and Hier-VRNN, the model is still slightly underfitting RoboNet. We provide visualizations of failure examples in Figure 15. In this figure, the GHVAE model failed to accurately track the movement of the blue bowl. This indicates that the GHVAE model is still slightly underfitting on RoboNet. Given that such failure to track graspable object does not occur frequently for RoboNet, we hypothesize that this failure case is due to underfitting, and that training an 8-module, 10-module or 12-module GHVAE model can potentially tackle such failure case.

In addition, we hypothesize that a monocular image can cause partial observability to the video prediction problem. In Figure 15 for example, without visually capturing the precise 3D locations of the robot and the blue bowl, it is difficult to tell whether the robot has successfully grasped the blue bowl and to predict the future motions of the blue bowl accordingly. Therefore, adding an $[x, y, z]$ state end-effector position vector or a second camera image from a different viewpoint (both are readily available information) to the GHVAE model can potentially resolve such a failure case.