Greedy Hierarchical Variational Autoencoders for Large-Scale Video Prediction

Bohan Wu, Suraj Nair, Roberto Martín-Martín, Li Fei-Fei,[†] Chelsea Finn[†]
Department of Computer Science
Stanford University
Stanford, CA 94305
`{bohanwu, surajn, robertom, feifeili, cbfinn}@cs.stanford.edu`

1 Introduction

A core aspect of intelligence is the ability to predict the future. Indeed, if equipped with an accurate video prediction model, an intelligent agent such as a robot will be able to perform a variety of tasks using raw pixel inputs. For example, algorithms such as visual foresight [1] can leverage an action-conditioned video prediction model to plan a sequence of actions that accomplish a desired task objective. Importantly, such video prediction models can in principle be trained with broad, unlabeled datasets, and building methods that can scale to large and diverse offline data is a recipe that has seen substantial success in visual [2] and language [3] understanding. However, learning an accurate video prediction model across diverse environments remains a significant challenge. The future visual observations of the world are hierarchical [4], high-dimensional, and uncertain, demanding the model to accurately represent the multi-level stochasticity of future pixels, which can include both low-level features (e.g. the texture of a table as it becomes unoccluded by an object) and high-level attributes (e.g. how an object will move when touched).

To capture the stochasticity of the future, prior works have proposed a variety of stochastic latent variable models [5, 6, 7]. While these methods have generated reasonable predictions for small-scale video prediction datasets such as the BAIR robot pushing dataset [8], they suffer from severe underfitting in large-scale datasets in the face of practical GPU or TPU memory constraints [9]. In parallel, while hierarchical variational autoencoders (VAEs) can in principle capture multiple levels of stochasticity, they become substantially more difficult to optimize as the number of hierarchical latent variables in the network increases [10]– an optimization dilemma unsolved in hierarchical VAEs [11].

The key insight of this work is that greedy and modular optimization of hierarchical autoencoders can simultaneously address both the memory constraints and the optimization challenges of learning accurate large-scale video prediction. On one hand, by circumventing end-to-end training, greedy machine learning allows local training of sub-modules of the entire video prediction model, enabling much larger models to be learned within the same limit of GPU or TPU memory. On the other hand, optimizing hierarchical VAEs in a greedy and modular fashion breaks the bidirectional dependency among individual stochastic latent variables. As a result, these variables can remain stable and useful throughout the entire training process, resolving the typical instability of training deep hierarchical VAEs.

With this key insight, this paper introduces Greedy Hierarchical VAEs ("GHVAEs" hereafter) (Fig. 1 in Appendix)– a set of local latent VAE modules that can be sequentially stacked and trained in a greedy, module-by-module fashion, leading to a deep hierarchical video prediction model that in practice admits a stable optimization and in principle can scale to large-scale video datasets. As evaluated in Section 3, GHVAEs outperform state-of-the-art video prediction models by 17-55% in FVD score [12] on four different datasets. In addition, our empirical and theoretical analyses also find

[†] Equal advising and ordered alphabetically. 34th Conference on Neural Information Processing Systems (NeurIPS 2020) Workshop: Self-Supervised Learning - Theory and Practice, Vancouver, Canada.

that GHVAE's performance can improve monotonically as the number of GHVAE modules in the network increases. In summary, the core contributions of this work is the use of greedy optimization to improve both the optimization stability and the memory efficiency of hierarchical VAEs, leading to significant gains in both large-scale video prediction accuracy and real robotic task success rates.

2 Greedy Hierarchical VAEs (GHVAEs)

Overview. To develop an expressive yet stably optimized video prediction model, we introduce Greedy Hierarchical VAEs (Fig. 1 in Appendix), which are locally optimized VAE modules that can be stacked together sequentially to incrementally add capacity to the model. In order to train a stack of modules without needing to fit the entire model into memory, each module is optimized locally using cached outputs from the previous trained module. Concretely, a GHVAE model has multiple GHVAE modules. Each GHVAE module has four convolutional sub-networks: an encoder, a decoder, a prior inference network, and a posterior inference network. In the remainder of this section, we overview mathematical notation in Appendix B, describe each of these model components in detail, derive the training objective for each module as a variational lower bound, and theoretically analyze the implications of greedy training in Appendix C.

Encoder. Shown as grey downward arrows in Fig. 1, the encoders incrementally map from x_t to h_t^K and serves as part of both the VAE model and the variational distribution. For the encoder design, it is important to recall that VAEs treat each dimension of a stochastic latent variable as independent (i.e. the mean-field approximation). However, convolutional embeddings of images contain significant spatial correlations due to the low frequency of natural images, violating this approximation. To mitigate this challenge, we design the encoder architecture to incrementally compress the spatial dimensions of the embeddings while simultaneously significantly expanding the channel dimensions of the embeddings. This allows the model, at its deepest layer, to store plenty of information (including spatial information) without strongly-correlated dimensions. Concretely, the k^{th} encoder $\mathcal{W}_{enc}^k$ maps from h_t^{k-1} to h_t^k (except for the first encoder $\mathcal{W}_{enc}^0$, which maps x_t to h_t^1), and incrementally compresses the height and width, $H^k < H^{k-1}$, $W^k < W^{k-1}$, while expanding the channels $C_{\mathcal{H}}^k > C_{\mathcal{H}}^{k-1}$.

Decoder. Shown as blue arrows in Fig. 1, the decoders incrementally map from the deepest stochastic latent variable z_{t+1}^K back to x_{t+1} to predict the next image. To alleviate the burden of latent space prior inference for the prior inference network, it is desirable to ensure that the stochastic latent variable z_{t+1}^k contains mostly new information about the future that is absent from the past. In other words, any partial information of the future that exists in h_t^k does not need to be predicted and thus should not be contained in z_{t+1}^k. Hence, the decoder in the deepest latent space, $\mathcal{W}_{dec}^K$, takes as input both h_t^K and the posterior latent variable z_{t+1}^K, so that the network can borrow information directly from the past. Similarly, each decoder $\mathcal{W}_{dec}^k \in \{\mathcal{W}_{dec}^1 \dots \mathcal{W}_{dec}^{K-1}\}$ takes as input both h_t^k and h_{t+1}^{k+1} and predicts h_{t+1}^k (except for $\mathcal{W}_{dec}^1$, which predicts x_{t+1}), and incrementally expands the height and width, $H^k > H^{k+1}$, $W^k > W^{k+1}$, while compressing the channels $C_{\mathcal{H}}^k < C_{\mathcal{H}}^{k+1}$.

Prior Inference Network. Shown as green arrows in Fig. 1, the prior inference network $\mathcal{W}_{prior}^k$ maps h_t^k and a_t to the mean and variance of a diagonal Gaussian distribution for z_{t+1}^k to model stochasticity of future observations. The prior inference network is recurrent-convolutional and used both at train and test time. Empirically, using all K stochastic latent variables $z_{t+1}^1 \dots z_{t+1}^K$ leads to excessive stochasticity and degrades performance as the number of GHVAE modules scales up. Therefore, one key design choice is that while a K-module GHVAE uses all K stochastic latent variables during training (i.e., $z_{t+1}^{1 \dots K}$, one for each module) to sequentially learn the multi-level stochasticity of future observations, only the latent variable at the deepest level, z_{t+1}^K, is used during inference and requires prediction from the prior inference network. Training each decoder $\mathcal{W}_{dec}^k$ greedily with a separate stochastic latent variable z_{t+1}^k enables every decoder to project stochasticity in the succeeding latent space into meaningful stochasticity in the current latent space sequentially, until the stochasticity in the deepest latent space is projected all the way back to the pixel space. This allows GHVAEs to implicitly model the multi-level stochasticity of future observations without

Table 1: GHVAE vs. SVG' Video Prediction Test Performance (Mean ± Standard Error, Best Performance in Bold). GHVAE outperforms the largest SVG', "SVG' (M=3, K=5)", on all datasets across all metrics.

Dataset	Method	# of Modules	End-to-End	Video Prediction Test Performance				
				FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Human ↑
RoboNet	GHVAEs	6	No	**95.2±2.6**	**24.7±0.2**	**89.1±0.4**	**0.036±0.001**	92.0%
	SVG' (M=3, K=5) [9]	N/A	Yes	123.2±2.6	23.9±0.1	87.8±0.3	0.060±0.008	8.0%
KITTI	GHVAEs	6	No	**552.9±21.2**	**15.8±0.1**	**51.2±2.4**	**0.286±0.015**	93.3%
	SVG' (M=3, K=5) [9]	N/A	Yes	1217.3 [9]	15.0 [9]	41.9 [9]	0.327±0.003	6.7%
Human3.6M	GHVAEs	6	No	**355.2±2.9**	**26.7±0.2**	**94.6±0.5**	**0.018±0.002**	86.6%
	SVG' (M=3, K=5) [9]	N/A	Yes	429.9 [9]	23.8 [9]	88.9 [9]	0.028±0.006	13.4%

explicitly using multiple stochastic latent variables *during inference*, and maximally compress the latent space spatially module-by-module such that h_t^K contains as few spatial dimensions as possible. Because the deepest encoder will have the fewest spatial dimensions (and hence the least spatial correlation), we choose to include only one stochastic latent variable in the model at this deepest part of the network.

Posterior Inference Network. Although the encoder and decoder have minimized spatial dimensions in the deepest hidden layer h^K, the encoding process has produced a high channel dimension $C_{\mathcal{H}}^K$ for h^K. To improve quality of future prediction, the channels in h^K may need be further downsized to reduce the required output dimensions of the prior inference network. Hence, shown as brown arrows in Fig. 1, the posterior inference network maps the current module's hidden variable h_{t+1}^k to the mean and variance of a diagonal Gaussian distribution over the stochastic latent variable z_{t+1}^k. When a new module is added, a new posterior network for the current latent space is trained based on the latest module's representation. Both h_{t+1}^k and z_{t+1}^k are posterior latent variables because they contain information about the future observation x_{t+1}. For this same reason, the recurrent-convolutional posterior inference network is only available at train time and not used for inference at test time.

Optimization. The training process of a K-module GHVAE model is split into K training phases, and only the k^{th} GHVAE module is trained during phase k, where $k \in [1, K]$. GHVAE's training objective for the k^{th} module is:

$$\max_{\mathcal{W}^k} \sum_{t=0}^{T-1} \mathcal{L}_{greedy}^k(x_{t+1}, z_{t+1}^k) \tag{1}$$

where $\mathcal{W}^k = \{\mathcal{W}_{enc}^k, \mathcal{W}_{dec}^k, \mathcal{W}_{prior}^k, \mathcal{W}_{post}^k\}$, $\mathcal{W}^{1^* \dots k-1^*}$ are the frozen, greedily trained weights of all preceding GHVAE modules, and $\mathcal{L}_{greedy}^k(x_{t+1}, z_{t+1}^k)$ is GHVAE's Evidence Lower-Bound (ELBO) with respect to the current module $\mathcal{W}^k$ at timestep t:

$$\begin{aligned}&\mathcal{L}_{greedy}^k(x_{t+1}, z_{t+1}^k)\\ &\quad = \mathbb{E}_{q^k(z_{t+1}^k \mid x_{t+1})}[\log p^k(x_{t+1} \mid x_{t+1}, z_{t+1}^k)] - D_{KL}\big(q^k(z_{t+1}^k \mid x_{t+1}) \parallel p^k(z_{t+1}^k | x_{t+1}, a_t)\big)\end{aligned} \tag{2}$$

where $p^k \equiv p_{\mathcal{W}_{enc,dec,prior}^{1* \dots k-1*, k}}$ and $q^k \equiv q_{\mathcal{W}_{enc,post}^{1* \dots k-1*, k}}$

Here, p^k is the VAE model, while q^k is the variational distribution. The encoder, decoder, and the prior are all part of the model p^k, and the encoder and the posterior are both part of q^k. To improve training stability, we use a fixed standard deviation for the posterior latent variable distribution $q^k(z_{t+1}^k \mid x_{t+1})$ in the KL divergence term in Eq. 2.

3 Experimental Evaluation and Analysis

We conduct video prediction and real-robot experiments to answer six key questions about GHVAEs: **1)** How do GHVAEs compare to state-of-the-art models in video prediction? **2)** Can GHVAEs achieve monotonic improvement in video prediction accuracy by simply adding more modules, as Theorem 2 suggests? **3)** Does training a GHVAE model end-to-end outperform training greedily per module, as Theorem 1 suggests? **4)** Does the high expressivity of GHVAEs cause overfitting during training? **5)** How important is the prior inference network to GHVAEs' performance? **6)** Does the high expressivity of GHVAEs improve real-robot performance? Visualizations and videos are at `https://sites.google.com/view/ghvae`, and more experimental results are in Appendix E.

Video Prediction Performance. To answer the first question, this paper evaluates video prediction methods across five metrics: Fréchet Video Distance (FVD) [12], Structural Similarity Index Measure (SSIM), Peak Signal-to-noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS) [13], and human preference. FVD and human preference both measure overall visual quality and temporal coherence without reference to the ground truth video. PSNR, SSIM, and LPIPS measure similarity to the ground-truth in different spaces, with LPIPS most accurately representing human perceptual similarity. To stress-test each method's ability to learn from large and diverse offline video datasets, we use four datasets: RoboNet [14] to measure prediction of object interactions, KITTI [15] and CityScapes [16] to evaluate the ability to handle partial observability, and Human3.6M [17] to assess prediction of structured motion. This paper compares GHVAEs against SVG' [7, 9] and Hier-VRNN [10], which are two state-of-the-art prior methods that use non-hierarchical and hierarchical VAEs respectively. While SAVP [6] is another prior method, we empirically found that SAVP underperforms SVG' on these datasets, and therefore omitted SAVP results for simplicity. All metrics are summarized statistically (mean and standard error).

Table 2: GHVAE vs. Hier-VRNN Video Prediction Test Performance on CityScapes (Mean±Standard Error, Best Performance in Bold). All Convolutional Layers in the 6-Module GHVAE model are Downsized by 40% to fit into 16GB GPU Memory for Fair Comparison.

Method	# of Modules	End-to-End	FVD ↓	SSIM ↑	LPIPS ↓
GHVAEs	6	No	**418.0±5.0**	**78.5±0.4**	**0.193±0.014**
Hier-VRNN [10]	N/A	Yes	567.5 [10]	62.8 [10]	0.264 [10]

For SVG' in particular, this paper compares to "SVG' (M=3, K=5)" [9], which is the *largest* version of SVG' that can fit into 24GB GPUs empirically while maintaining a batch size of 32, and also the largest and best-performing SVG' model that Villegas et al. [9] evaluate. "SVG' (M=3, K=5)" has 3x larger convolutional LSTMs and 5x larger encoder and decoder CNNs compared to the original SVG' [7] and significantly outperforms the original SVG' by 40-60% in FVD scores [9]. Since Villegas et al. [9] reported the FVD, SSIM and PSNR performance of "SVG' (M=3, K=5)" on KITTI and Human3.6M, we directly compare to their results using the same evaluation methodology. For RoboNet and for evaluating LPIPS and human preference, we re-implement "SVG' (M=3, K=5)" and report the corresponding performance. In Table 1, 6-module GHVAEs outperform "SVG' (M=3, K=5)" across all three datasets across all metrics. Most saliently, we see a 17-55% improvement in FVD score and a 30-45% improvement in LPIPS. Further, we see that humans prefer predictions from GHVAEs more than 85% of the time.

To compare to Hier-VRNN [10], we use the CityScapes driving dataset [16]. Since Castrejon et al. [10] already reports FVD, SSIM and LPIPS performance on CityScapes, we directly compare to these results using the same evaluation setting. Table 2 indicates that GHVAEs outperform Hier-VRNN [10] for CityScapes across all three metrics when the number of modules reaches six.

These results indicate that given the same amount of GPU or TPU memory, GHVAEs significantly outperform state-of-the-art hierarchical and non-hierarchical variational video prediction models. The performance superiority of GHVAEs mainly originates from the capacity to learn larger models with a stable optimization within the same amount of GPU or TPU memory. In Appendix E, we perform several ablations to better understand the good performance of GHVAEs.

4 Conclusion

This paper introduces Greedy Hierarchical VAEs (GHVAEs), which are local VAE modules that can be stacked sequentially and optimized greedily to construct an expressive yet stably optimized hierarchical video prediction model. This method significantly outperforms state-of-the-art hierarchical and non-hierarchical video prediction methods by 17-55% in FVD score across four video datasets. In addition, GHVAE achieves monotonic improvement by simply stacking more modules. By addressing the underfitting challenge of large-scale video prediction, this work makes it possible for intelligent agents such as robots to learn from large-scale offline video datasets and generalize across a wide range of complex visuomotor tasks through accurate visual foresight.

References

[1] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, “Visual foresight: Model-based deep reinforcement learning for vision-based robotic control,” *arXiv preprint arXiv:1812.00568*, 2018.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” 2020.

[4] S. E. Palmer, “Hierarchical structure in perceptual representation,” *Cognitive psychology*, vol. 9, no. 4, pp. 441–474, 1977.

[5] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, “Stochastic variational video prediction,” *ICLR*, 2018.

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

[7] E. Denton and R. Fergus, “Stochastic video generation with a learned prior,” ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 1174–1183. [Online]. Available: http://proceedings.mlr.press/v80/denton18a.html

[8] F. Ebert, S. Dasari, A. X. Lee, S. Levine, and C. Finn, “Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning,” in *Conference on Robot Learning (CORL)*, 2018.

[9] R. Villegas, A. Pathak, H. Kannan, D. Erhan, Q. V. Le, and H. Lee, “High fidelity video prediction with large stochastic recurrent neural networks,” in *Advances in Neural Information Processing Systems*, 2019, pp. 81–91.

[10] L. Castrejon, N. Ballas, and A. Courville, “Improved conditional vrnns for video prediction,” in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[11] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, “How to train deep variational autoencoders and probabilistic ladder networks,” in *33rd International Conference on Machine Learning (ICML)*, 2016.

[12] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, “Towards accurate generative models of video: A new metric & challenges,” *arXiv preprint arXiv:1812.01717*, 2018.

[13] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018.

[14] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, “Robonet: Large-scale multi-robot learning,” in *CoRL 2019: Volume 100 Proceedings of Machine Learning Research*, 2019.

[15] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *International Journal of Robotics Research (IJRR)*, 2013.

[16] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, jul 2014.

[18] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” in *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[19] X. Liang, L. Lee, W. Dai, and E. P. Xing, “Dual motion gan for future-flow embedded video prediction,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1744–1752.

[20] J. Xu, B. Ni, and X. Yang, “Video prediction via selective sampling,” in *Advances in Neural Information Processing Systems*, 2018, pp. 1705–1715.

[21] J.-T. Hsieh, B. Liu, D.-A. Huang, L. F. Fei-Fei, and J. C. Niebles, “Learning to decompose and disentangle representations for video prediction,” in *Advances in Neural Information Processing Systems*, 2018, pp. 517–526.

[22] F. A. Reda, G. Liu, K. J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro, “Sdc-net: Video prediction using spatially-displaced convolution,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 718–733.

[23] J. Xu, B. Ni, Z. Li, S. Cheng, and X. Yang, “Structure preserving video prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1460–1469.

[24] Y. Ye, M. Singh, A. Gupta, and S. Tulsiani, “Compositional video prediction,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 353–10 362.

[25] B. Boots, A. Byravan, and D. Fox, “Learning predictive models of a depth camera & manipulator from raw execution traces,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 4021–4028.

[26] C. Finn and S. Levine, “Deep visual foresight for planning robot motion,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2786–2793.

[27] N. Kalchbrenner, A. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, “Video pixel networks,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 1771–1779.

[28] F. Ebert, C. Finn, A. X. Lee, and S. Levine, “Self-supervised visual planning with temporal skip connections,” in *Conference on Robot Learning (CORL)*, 2017.

[29] A. Xie, F. Ebert, S. Levine, and C. Finn, “Improvisation through physical understanding: Using novel objects as tools with visual foresight,” in *Robotics: Science and Systems (RSS)*, 2019.

[30] C. Paxton, Y. Barnoy, K. Katyal, R. Arora, and G. D. Hager, “Visual robot task planning,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8832–8838.

[31] S. Nair and C. Finn, “Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation,” *ICLR*, 2020.

[32] S. Nair, M. Babaeizadeh, C. Finn, S. Levine, and V. Kumar, “Trass: Time reversal as self-supervision,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 115–121.

[33] A. S. Polydoros and L. Nalpantidis, “Survey of model-based reinforcement learning: Applications on robotics,” *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, 2017.

[34] A. massoud Farahmand, A. Shademan, M. Jagersand, and C. Szepesvári, “Model-based and model-free reinforcement learning for visual servoing,” in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2917–2924.

[35] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. Johnson, and S. Levine, “Solar: Deep structured representations for model-based reinforcement learning,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 7444–7453.

[36] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, “Action-conditional video prediction using deep networks in atari games,” in *Advances in neural information processing systems*, 2015, pp. 2863–2871.

[37] M. S. Nunes, A. Dehban, P. Moreno, and J. Santos-Victor, “Action-conditioned benchmarking of robotic video prediction models: a comparative study,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8316–8322.

[38] J. Walker, A. Gupta, and M. Hebert, “Dense optical flow prediction from a static image,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2443–2451.

[39] C. Finn, I. Goodfellow, and S. Levine, “Unsupervised learning for physical interaction through video prediction,” in *Advances in neural information processing systems*, 2016, pp. 64–72.

[40] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, “Dynamic filter networks,” in *Advances in neural information processing systems*, 2016, pp. 667–675.

[41] 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*, 2016, pp. 91–99.

[42] 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*. Springer, 2016, pp. 835–851.

[43] A. Byravan and D. Fox, “Se3-nets: Learning rigid body motion using deep neural networks,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 173–180.

[44] C. Vondrick and A. Torralba, “Generating the future with adversarial transformers,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1020–1028.

[45] J. Van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala, “Transformation-based models of video sequences,” *arXiv preprint arXiv:1701.08435*, 2017.

[46] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, “Video frame synthesis using deep voxel flow,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4463–4471.

[47] B. Chen, W. Wang, and J. Wang, “Video imagination from a single image with transformation generation,” in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017, pp. 358–366.

[48] C. Lu, M. Hirsch, and B. Scholkopf, “Flexible spatio-temporal networks for video prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6523–6531.

[49] R. Shu, J. Brofos, F. Zhang, H. H. Bui, M. Ghavamzadeh, and M. Kochenderfer, “Stochastic video prediction with conditional density estimation,” in *ECCV Workshop on Action and Anticipation for Visual Learning*, vol. 2, 2016.

[50] N. Wichers, R. Villegas, D. Erhan, and H. Lee, “Hierarchical long-term video prediction without supervision,” *International Conference on Machine Learning (ICML)*, 2018.

[51] J.-Y. Franceschi, E. Delasalles, M. Chen, S. Lamprier, and P. Gallinari, “Stochastic latent residual video prediction,” *arXiv preprint arXiv:2002.09219*, 2020.

[52] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, “Roboturk: A crowdsourcing platform for robotic skill learning through imitation,” in *Conference on Robot Learning*, 2018.

[53] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, “Conditional image generation with pixelcnn decoders,” in *Advances in neural information processing systems*, 2016, pp. 4790–4798.

[54] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, “Pixelcnn++: A pixelcnn implementation with discretized logistic mixture likelihood and other modifications,” in *ICLR*, 2017.

[55] M. Janner, S. Levine, W. T. Freeman, J. B. Tenenbaum, C. Finn, and J. Wu, “Reasoning about physical interactions with object-oriented prediction and planning,” in *International Conference on Learning Representations*, 2018.

[56] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner, “Multi-object representation learning with iterative variational inference,” in *International Conference on Machine Learning*, 2019, pp. 2424–2433.

[57] N. Watters, L. Matthey, M. Bosnjak, C. P. Burgess, and A. Lerchner, “Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration,” *arXiv preprint arXiv:1905.09275*, 2019.

[58] R. Veerapaneni, J. D. Co-Reyes, M. Chang, M. Janner, C. Finn, J. Wu, J. Tenenbaum, and S. Levine, “Entity abstraction in visual model-based reinforcement learning,” in *Conference on Robot Learning*, 2019, pp. 1439–1456.

[59] T. Kipf, E. van der Pol, and M. Welling, “Contrastive learning of structured world models,” in *International Conference on Learning Representations*, 2019.

[60] M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner, “Genesis: Generative scene inference and sampling with object-centric latent representations,” in *International Conference on Learning Representations*, 2019.

[61] S. Nair, S. Savarese, and C. Finn, “Goal-aware prediction: Learning to model what matters,” *International Conference on Machine Learning (ICML)*, 2020.

[62] J. J. Verbeek, N. Vlassis, and B. Kröse, “Efficient greedy learning of gaussian mixture models,” *Neural computation*, vol. 15, no. 2, pp. 469–485, 2003.

[63] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[64] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in neural information processing systems*, 2007, pp. 153–160.

[65] T. Haarnoja, K. Hartikainen, P. Abbeel, and S. Levine, "Latent space policies for hierarchical reinforcement learning," *arXiv preprint arXiv:1804.02808*, 2018.

[66] E. Belilovsky, M. Eickenberg, and E. Oyallon, "Greedy layerwise learning can scale to imagenet," in *International conference on machine learning*. PMLR, 2019, pp. 583–593.

[67] M. Malinowski, G. Swirszcz, J. Carreira, and V. Patraucean, "Sideways: Depth-parallel training of video models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[68] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.

[69] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International conference on artificial neural networks*. Springer, 2011, pp. 52–59.

[70] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *European conference on computer vision*. Springer, 2014, pp. 1–16.

[71] V. Kumar, G. C. Nandi, and R. Kala, "Static hand gesture recognition using stacked denoising sparse autoencoders," in *2014 Seventh International Conference on Contemporary Computing (IC3)*. IEEE, 2014, pp. 99–104.

[72] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *arXiv preprint arXiv:1605.08104*, 2016.

[73] E. P. Ijjina *et al.*, "Classification of human actions using pose-based features and stacked auto encoder," *Pattern Recognition Letters*, vol. 83, pp. 268–277, 2016.

[74] D. Singh and C. K. Mohan, "Deep spatio-temporal representation for detection of road accidents using stacked autoencoder," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 879–887, 2018.

[75] Y. Qi, Y. Wang, X. Zheng, and Z. Wu, "Robust feature learning by stacked autoencoder with maximum correntropy criterion," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6716–6720.

[76] S. Löwe, P. O'Connor, and B. Veeling, "Putting an end to end-to-end: Gradient-isolated learning of representations," in *Advances in Neural Information Processing Systems*, 2019, pp. 3039–3051.

[77] S. Löwe, P. O'Connor, and B. S. Veeling, "Greedy infomax for self-supervised representation learning," 2019.

[78] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[79] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder variational autoencoders," in *Advances in neural information processing systems*, 2016, pp. 3738–3746.

[80] S. Zhao, J. Song, and S. Ermon, "Towards deeper understanding of variational autoencoding models," *arXiv preprint arXiv:1702.08658*, 2017.

[81] A. Vahdat and J. Kautz, "Nvae: A deep hierarchical variational autoencoder," *arXiv preprint arXiv:2007.03898*, 2020.

[82] S. Zhao, J. Song, and S. Ermon, "Learning hierarchical features from generative models," in *33rd International Conference on Machine Learning (ICML)*, 2016.

[83] L. Maaløe, M. Fraccaro, V. Liévin, and O. Winther, "Biva: A very deep hierarchy of latent variables for generative modeling," in *Advances in neural information processing systems*, 2019, pp. 6551–6562.

[84] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv preprint arxiv:2006.11239*, 2020.

[85] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv:2010.02502*, October 2020. [Online]. Available: https://arxiv.org/abs/2010.02502

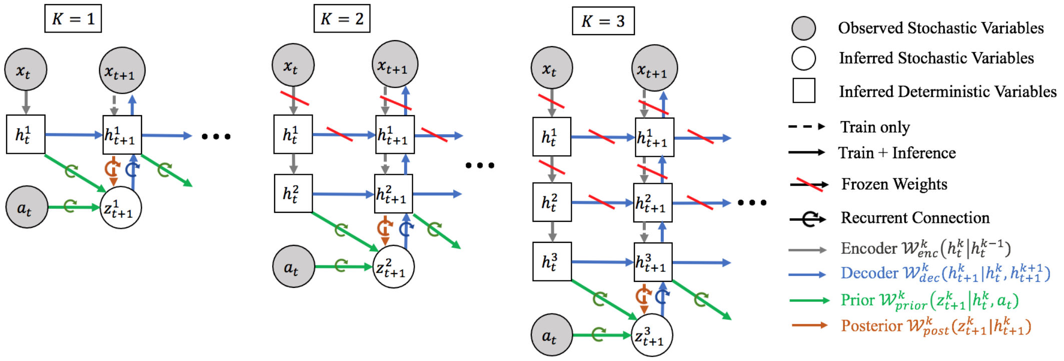

K = 1
K = 2
K = 3
x_t
x_{t+1}
h_t^1
h_{t+1}^1
h_t^2
h_{t+1}^2
h_t^3
h_{t+1}^3
a_t
z_{t+1}^1
z_{t+1}^2
z_{t+1}^3
Observed Stochastic Variables
Inferred Stochastic Variables
Inferred Deterministic Variables
Train only
Train + Inference
Frozen Weights
Recurrent Connection
Encoder $\mathcal{W}_{enc}^k(h_t^k|h_t^{k-1})$
Decoder $\mathcal{W}_{dec}^k(h_{t+1}^k|h_t^k, h_{t+1}^{k+1})$
Prior $\mathcal{W}_{prior}^k(z_{t+1}^k|h_t^k, a_t)$
Posterior $\mathcal{W}_{post}^k(z_{t+1}^k|h_{t+1}^k)$

Figure 1: **Training Procedure for One, Two, and Three-Module GHVAE models.** Let K denote the total number of GHVAE modules in the entire network. When $K = 1$, all network weights are trained end-to-end. When $K = 2$, all weights from the first module are frozen and the second module is trained. When $K = 3$, all first and second-module weights are frozen, and only the third module is trained, so on and so forth. The legends in the figure denote the four components in each GHVAE module (encoder, decoder, prior, and posterior) and whether each component is frozen (tilted red bars) or used only for training and not for inference at test time (dashed as opposed to solid lines). In order to limit the number of spatial dimensions that requires prior inference, only the prior and posterior in the final, K^{th} GHVAE module are used. The action a_t is included in action-conditioned video prediction and excluded in action-free video prediction.

A Acknowledgement

We thank Danfei Xu for brainstorming and discussions at the early stages of this research.

B Mathematical Notations

This paper uses K to denote the total number of GHVAE modules in the network, $\mathcal{W}^k, k \in [1, K]$ to denote the k^{th} GHVAE module, $\mathcal{W}^k = \{\mathcal{W}_{enc}^k, \mathcal{W}_{dec}^k, \mathcal{W}_{prior}^k, \mathcal{W}_{post}^k\}$ to denote the k^{th} module's encoder, decoder, prior inference network, and posterior inference network respectively, $x_t \in \mathcal{X} = \mathbb{R}^{H^0 \times W^0 \times C^0}$ to represent the robot's RGB image observation (height H^0, width W^0, channel $C^0 = 3$) at the current timestep t, $h_t^k \in \mathcal{H}^k = \mathbb{R}^{H^k \times W^k \times C_{\mathcal{H}}^k}$ to denote the hidden variable encoded by the k^{th} module for the current timestep t, $z_{t+1}^k \in \mathcal{Z}^k = \mathbb{R}^{H^k \times W^k \times C_{\mathcal{Z}}^k}$ to denote the k^{th} stochastic latent variable used to explicitly model the stochasticity of the future observation at timestep $t + 1$, $a_t \in \mathcal{A}$ to denote the robot's action at the current timestep t in the case of action-conditioned video prediction, and T to denote the the model's rollout horizon during training.

C Theoretical Guarantees

GHVAE's ELBO manifests two theoretical guarantees. **1) ELBO Validity:** sequentially optimizing each GHVAE module in the network is equivalent to maximizing a *lower-bound* of the ELBO for training all GHVAE modules end-to-end. This suggests that GHVAE's ELBO is *valid*:

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

$$\log p(x_{t+1}) \geq \max_{\mathcal{W}^{1 \ldots k}} \mathcal{L}_{e2e}^k(x_{t+1}, z_{t+1}^k) \geq \max_{\mathcal{W}^k} \mathcal{L}_{greedy}^k(x_{t+1}, z_{t+1}^k) \tag{3}$$

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

Proof. Suppose $\mathcal{W}^{k^*}$ is the optimal parameters from greedily training the last module of a model comprised of k GHVAEs, and $\mathcal{W}^{1^* \ldots k-1^*}$ are all previous modules' frozen weights previously trained in a sequential, greedy fashion module by module:

$$\mathcal{W}^{k^*} = \arg\max_{\mathcal{W}^k} \mathcal{L}_{greedy}^k(x_{t+1}, z_{t+1}^k) \tag{4}$$

In other words:

$$\max_{\mathcal{W}^k} \mathcal{L}^k_{greedy}(x_{t+1}, z^k_{t+1}) = \mathcal{L}^k_{greedy}(x_{t+1}, z^k_{t+1}; \mathcal{W}^{k^*}) \tag{5}$$

Therefore:

$$\log p(x_{t+1}) \geq \max_{\mathcal{W}^{1...k}} \mathcal{L}^k_{e2e}(x_{t+1}, z^k_{t+1}) \geq \mathcal{L}^k_{greedy}(x_{t+1}, z^k_{t+1}; \mathcal{W}^{k^*}) = \max_{\mathcal{W}^k} \mathcal{L}^k_{greedy}(x_{t+1}, z^k_{t+1}) \tag{6}$$

2) Monotonic Improvement: adding more modules can only raise (as opposed to lower) GHVAE's ELBO, which justifies and motivates maximizing the number of modules in a GHVAE model:

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

$$\log p(x_{t+1}) \geq \mathcal{L}^k_{greedy}(x_{t+1}, z^k_{t+1}) \geq \mathcal{L}^{k-1}(x_{t+1}, z^{k-1}_{t+1}) \tag{7}$$

where $\mathcal{L}^{k-1} \in \{\mathcal{L}^{k-1}_{greedy}, \mathcal{L}^{k-1}_{e2e}\}$.

Full proof is available at `https://sites.google.com/view/ghvae`.

D Related Work

The Underfitting Challenge of Large-Scale Video Prediction. Resolving the underfitting challenge of large-scale video prediction [18, 19, 20, 21, 22, 23, 10, 24] can lead to powerful generalization such as visual foresight [25, 26, 27, 28, 8, 1, 29, 30, 31, 32], which performs model-based robotic control [33, 34, 35] via action-conditioned video prediction [36, 37]. Initially, video prediction is tackled by a deterministic model [38, 39, 40, 41, 42, 19, 43, 44, 45, 46, 47, 48]. VAEs were later adopted to model the stochasticity of future visual observations [49, 5, 50, 51]. Nevertheless, modeling the stochasticity of the real world using a trajectory-based latent variable model leads to blurry predictions inadvertently. This problem is then addressed by two lines of orthogonal work– VAE-GANs [6] and timestep-based latent variable models [7]. While these methods resolve blurry predictions in small-scale video datasets such as the BAIR robot pushing dataset [8], they suffer from severe underfitting in large-scale, multi-domain, or multi-robot datasets, such as RoboNet [14] and RoboTurk [52]. In parallel, Villegas et al. [9] validates that higher model capacity leads to greater prediction fidelity. This raises the question of how to learn larger models to meet the underfitting challenge of large-scale video prediction, especially in the face of practical GPU or TPU memory constraints. In parallel, Castrejon et al. [10] applied dense connections to hierarchical VAEs to address the optimization challenge of fitting deep video prediction models. While this work outperforms the state-of-the-art in relatively small video datasets, it was unable to scale its hierarchical VAE up substantially due to deep optimization problems [11]. Other works have also attempted to address the underfitting challenge of large-scale video prediction through other angles. For example, one line of work attempts to represent pixels as discrete as opposed to continuous distributions [53, 54]. Other works have proposed to predict forward alternative quantities such as object-centric representations [55, 56, 57, 58, 59, 60] and goal-centric representations [61]. Unlike these approaches, we find that GHVAEs scale to large real-world video datasets without requiring additional inductive biases such as detailed segmentation annotations.

Greedy Machine Learning. Greedy machine learning [62, 63, 64, 65, 66, 67] was first introduced to provide a good weight initialization for deep networks to escape bad local optima during end-to-end back-propagation. As originally proposed, each greedy module of a deep network is stacked on top of the preceding greedy module and trained locally based on the features extracted from the preceding module. Subsequently, greedy machine learning has been applied to pre-training good feature extractors and stacked autoencoders [68, 69, 70, 71, 72, 73, 74] for downstream tasks in vision, sound, and language [75, 76, 77]. Trained via self-supervised learning, these feature extractors and autoencoders excelled at capturing and preserving time-invariant information in sequential data such as videos. In contrast, we propose a video prediction method that uses a hierarchy of latent variables to explicitly model time-variant information about the future. Finally, greedy training of generative adversarial networks (GANs) is proposed to generate high-quality, high-resolution single-images [78]. Unlike these prior works, we propose a greedy approach to training large-scale video prediction models that simultaneously addresses the memory constraints and the optimization challenges of hierarchical VAEs.

Hierarchical Variational Autoencoders. Hierarchical [79] and sequential VAEs [80] were recently introduced to improve generative modeling in various vision tasks such as video prediction [10] and image generation [81].

Table 3: GHVAE Ablation Studies. Note that GHVAEs exhibit monotonic improvement as the number of modules increases from 2 to 4 and finally to 6.

Dataset	Method	# of Modules	End-to-End	Train / Test	Video Prediction Performance			
					FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓
KITTI	GHVAEs (Train Performance, Abl. 2)	6	No	Train	453.5±12.5	19.4±0.2	61.4±1.6	0.209±0.006
	GHVAEs (Original)	6	No	Test	552.9±21.2	15.8±0.1	51.2±2.4	0.286±0.015
Human3.6M	GHVAEs (Train Performance, Abl. 2)	6	No	Train	258.9±6.8	28.6±0.3	96.4±0.1	0.015±0.002
	GHVAEs (Original)	6	No	Test	355.2±2.9	26.7±0.2	94.6±0.5	0.018±0.002
Cityscapes	GHVAEs (Train Performance, Abl. 2)	6	No	Train	401.8±5.4	N/A	82.9±0.1	0.194±0.006
	GHVAEs (Original)	6	No	Test	418.0±5.0		78.5±0.4	0.193±0.014
RoboNet	GHVAEs (Train Performance, Abl. 2)	6	No	Train	94.4±3.9	24.9±0.3	89.3±0.7	0.036±0.002
	GHVAEs (Original)	6	No	Test	95.2±2.6	24.7±0.2	89.1±0.4	0.036±0.001
	GHVAEs (Abl. 1)	4	No	Test	151.2±2.3	24.2±0.1	87.5±0.4	0.059±0.006
	GHVAEs (Abl. 1)	2	No		292.4±11.1	23.5±0.2	86.4±0.2	0.106±0.010
	GHVAEs (Uniform Prior, Abl. 3)	6	No		281.4±1.6	22.1±0.3	85.0±0.4	0.098±0.007
	GHVAEs (End-to-End Fine-Tuning, Abl. 4)	6	No		91.1±3.1	25.0±0.2	89.5±0.5	0.032±0.003
	GHVAEs (End-to-End Training, Abl. 4)	6	Yes		509.9±6.2	21.2±0.3	83.5±1.0	0.148±0.004

They are known to have optimization challenges [11], mainly due to the hierarchical dependencies among the individual variables. When optimized end-to-end, the hierarchical VAE needs to keep each latent variable useful for the generative task at hand throughout the entire training process, while preserving the dependent relationships among these variables simultaneously. To this end, previous works introduced a variety of inductive biases such as dense connections [10], ladder structures [82], bidirectional inference [83], progressive lossy compression [84, 85], and spectral regularization [81] to alleviate such optimization difficulties specific to hierarchical VAEs. These approaches have largely been successful in the context of image generation, while we study the more difficult video prediction problem. Unlike these approaches, we propose a greedy training scheme that significant alleviates the optimization challenges of conditional hierarchical VAEs.

E Experiments

Ablation 1: Monotonic Improvement and Scalabiliy of GHVAEs. Given that GHVAEs can be stacked sequentially, it becomes important to determine whether GHVAEs can achieve monotonic improvement by simply adding more GHVAE modules, as suggested by Theorem 2. We observe in Table 3 that increasing the number of GHVAE modules from 2, to 4, to eventually 6 improves performance across all metrics. These results validate Theorem 2, and suggest that greedily adding more modules monotonically increases performance in practice and enables GHVAEs to scale to large datasets.

Ablation 2: Train-Test Performance Comparison for GHVAEs. Since GHVAEs tackle the underfitting challenge of large-scale video prediction, a natural question is whether GHVAEs have instead overfitted to the training dataset. We observe in Table 3 that for RoboNet, a 6-module GHVAE's training performance in rows "GHVAEs (Train Performance, Abl. 2)" is statistically similar to its test performance in rows "GHVAEs (Original)" across all four metrics, implying little overfitting. For KITTI, Human3.6M and Cityscapes, we observe that train performance is statistically better than test performance across most metrics, indicating some overfitting. We hypothesize that this is due to the smaller sizes of these three datasets compared to RoboNet. In addition, the fact that the test set for Human3.6M is composed of two unseen human subjects implies that the train-test distribution gap is higher for Human3.6M.

Ablation 3: Performance Contribution of Prior Inference in the Deepest Latent Space. One of GHVAEs' insights is to perform prior inference only in the deepest latent space. Therefore, it may be important to quantify the contribution of the learned prior inference network in the deepest latent space to the overall performance of the model. We observe in Table 3 that using a learned prior in row "GHVAEs (Original)" significantly outperforms using a uniform diagonal Gaussian prior in row "GHVAEs (Uniform Prior, Abl. 3)" across all metrics. This indicates that prior inference in the deepest latent space is important to GHVAEs' empirical video prediction performance.

Ablation 4: Greedy vs. End-to-End Optimization of GHVAEs. End-to-end learning is conventionally preferred over greedy training when GPU or TPU memory constraints are loose. To examine whether this pattern also holds for GHVAEs, we trained a 6-module GHVAE model end-to-end in two 48GB GPUs (since the end-to-end model does not fit in 24GB GPUs) across five separate trials. In addition, we conducted a second experiment in which we fine-tune the greedily trained GHVAE model end-to-end using two 48GB GPUs. We found in row "GHVAEs (End-to-End Training, Abl. 4)" in Table 3 that the model was unable to converge to any good performance in any single run compared to the greedy setting. Qualitatively, when optimized end-to-end, GHVAE models need to update each module to improve video prediction quality, while preserving the interdependency among individual hidden variables simultaneously, which can lead to optimization difficulties [11]. Even if GHVAEs can be optimized end-to-end, limited GPU or TPU memory capacity will still make it infeasible to train as the number of modules grows beyond six. However, end-to-end

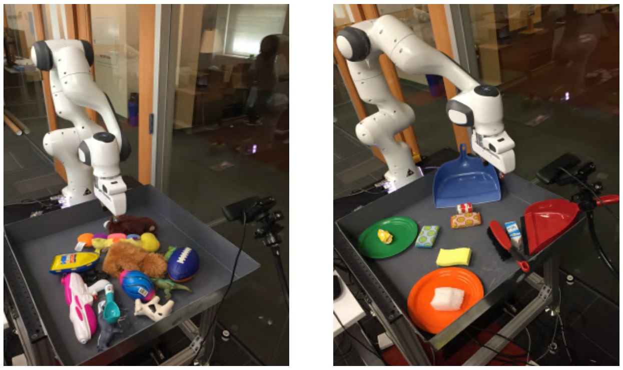
Table 4: GHVAE vs. SVG' Real-Robot Performance

Method	# of Modules	End-to-End	Test Task Success Rate	
			Pick&Wipe	Pick&Sweep
GHVAEs	6	No	**90.0%**	**85.0%**
SVG'	N/A	Yes	50.0%	50.0%

fine-tuning does lead to minor performance gains as indicated by row "GHVAEs (End-to-End Fine-Tuning, Abl. 4)". These two experiments imply that greedy training of GHVAEs leads to higher optimization stability than end-to-end training from scratch. They also indicate that end-to-end training of GHVAE can outperform greedy training as suggested by Theorem 1, so long as the GHVAE model is first pre-trained greedily.

E.1 Real-Robot Performance

Finally, we seek to evaluate how improved video prediction performance translates to greater success on downstream tasks. In particular, we consider two real robotic manipulation tasks: **Pick&Wipe** and **Pick&Sweep**. Concretely, each method is given a small, *autonomously collected* training dataset of 5000 videos of random interactions between the Franka robot and diverse objects such as those in the dark-grey tabletop bin in Fig. 2a. At test time, to measure each method's ability to generalize across novel objects, all objects, tools, and containers used at test time are *never seen* during training. Empirically, training directly on this small 5000-video dataset leads to poor generalization to novel objects at test time for both methods. Therefore, to enable better generalization, all networks are first pretrained on RoboNet [14] and subsequently fine-tuned on this 5000-video dataset. All methods are evaluated across two visuomotor tasks (Fig. 2) with horizons of 50 timesteps, with results reported and analyzed individually below. In both tasks, the robot is given by a single 64×64 RGB goal image, with no hand-designed rewards provided. The model rollout horizon for each video prediction method is 10, with two prior context frames and a sequence of 10 future actions provided as input. All real-robot results are evaluated across 20 trials. For planning, we perform random shooting) on a random policy with an action space of dimension 4 ($\mathcal{A} = \mathbb{R}^4$) for the Franka robot, which contains three scalars for the $[x, y, z]$ end-effector translation and one binary scalar for opening vs. closing its parallel-jaw gripper.



(a) Train: Random Interaction (b) Test: Unseen Objects

Figure 2: **Real-Robot Experimental Setup.** The white Franka robot is equipped with a 45° black camera capturing a 64×64 RGB image. Training GHVAEs comprises of pre-training on RoboNet and fine-tuning on a *fully-autonomously* collected dataset of 5000 videos of the robot's random interactions with objects in the bin (Fig. 2a), which empirically takes 120 hours with no human supervision. Using the trained GHVAE video prediction model, the Franka robot is tested across two tasks: Pick&Wipe (top and bottom left of bin in Fig. 2b) and Pick&Sweep (top and bottom right of bin in Fig. 2b). All tasks are evaluated on objects, tools and containers *never seen* during training (Fig. 2b).

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. The task is successful if the robot picks the wiping tool up and wipe all object 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 sponge, etc.) up and sweep one or two objects into the dustpan. The task is successful if the target object is swept into the dustpan within 50 timesteps. At the beginning of each task, the wiping or sweeping tool is *not yet* in the robot's gripper, which makes the tasks more difficult. Table 4 reveals that a 6-Module GHVAE model outperforms SVG' by 40% and 35% in success rate for Pick&Wipe and Pick&Sweep respectively. For Pick&Wipe, SVG' produces blurry predictions especially when the robot and the plate overlap in the image. This reduces SVG's ability to predict the best action sequence for wiping objects off the plate. In contrast, GHVAE empirically produces accurate predictions of the robot's motion and the position of the wiping tool and the objects. For Pick&Sweep, SVG' has difficulty predicting the movement of the object during the robot's sweeping motion, leading to more frequent task failures. In contrast, GHVAE predicts plausible robot sweep motions and object movements, reaching an 85.0% success rate. These results indicate that GHVAEs not only lead to better video prediction performance, but that they lead to better downstream performance on real robotic manipulation tasks.