A Symmetric and Object-Centric World Model for Stochastic Environments

Patrick Emami **Pan He** **Anand Rangarajan** **Sanjay Ranka**
University of Florida
{pemami,pan.he}@ufl.edu, {anand,ranka}@cise.ufl.edu

Abstract

Object-centric world models learn useful representations for planning and control but have so far only been applied to synthetic and deterministic environments. We introduce a perceptual-grouping-based world model for the dual task of extracting object-centric representations and modeling stochastic dynamics in visually complex and noisy video environments. The world model is built upon a novel latent state space model that learns the variance for object discovery and dynamics separately. This design is motivated by the disparity in available information that exists between the discovery component, which takes a provided video frame and decomposes it into objects, and the dynamics component, which predicts representations for future video frames conditioned only on past frames. To learn the dynamics variance, we introduce a best-of-many-rollouts objective. We show that the world model successfully learns accurate and diverse rollouts in a real-world robotic manipulation environment with noisy actions while learning interpretable object-centric representations.

1 Introduction

Object-centric world models aim to learn structured representations of environments that can be re-used to solve a variety of downstream tasks. For example, various studies have shown that object-centric representations can improve sample efficiency and generalization of visual model-based reinforcement learning agents [39; 22; 37; 19; 21; 3; 35; 8; 1]. However, the environments have largely been synthetic and deterministic. Extending these world models for real-world environments is therefore a promising line of investigation.

The goal of this work is to integrate unsupervised object discovery [13; 14; 15; 12; 5; 11] into a latent state space model (SSM) for realistic video environments. These environments have multiple rigid and non-rigid objects undergoing stochastic motion, occlusion, and variability in scale and illumination. To handle this visual complexity we adopt perceptual grouping (i.e., segmentation) for object discovery following a recent line of work [14; 34; 5; 11; 15; 35; 25]. These models represent images as a symmetric mixture of object-centric image components. A subset of these are *fully symmetric* latent variable generative models, in that any permutation applied to the order of the latent object representations (i.e., object slots) similarly permutes the output of the inference and generation networks. The fully symmetric inductive bias is known to be important when learning dynamics so that a single model of the environment physics can be shared by all object representations [4; 35].

In this work, we address the problem of jointly learning perceptual-grouping-based object discovery and *stochastic* dynamics, which we illustrate with the following example. Suppose that we have a sample from an object discovery distribution over slots, which corresponds to a segmentation of an image o_t of a block undergoing stochastic motion. Assume the block's pixels were assigned to a single slot and the discovery posterior distribution placed low variance on its inferred attributes since it fully observed o_t. However, the dynamics model, which only has access to observations

NeurIPS 2020 Workshop on Object Representations for Learning and Reasoning, Vancouver, Canada.

up to but not including time t, predicts high variance for the block's spatiotemporal attributes. We found that a SSM can respect this distinction between discovery and dynamics by learning their variances with separate objectives. Our key contributions are: (1) a world model with a novel SSM that separates learning the variance for unsupervised object discovery and stochastic dynamics, (2) a novel best-of-many-rollouts objective for training the dynamics variance and (3) experiments on a real-world robotic manipulation benchmark that outperforms the most relevant prior work [35].

2 Related Work

The most closely related world model is OP3 [35], which extends IODINE [15]—a fully symmetric object-centric generative model for images—to support videos by combining it with the RSSM [16], a *joint deterministic and stochastic* SSM. While OP3 claims to learn stochastic dynamics, we found that in practice it fails to do so. This is because OP3 uses a single shared dynamics model both as a prior for object discovery and for rolling out future frames, ignoring the discrepancy between discovery and stochastic dynamics we described in Section 1. As a result, the dynamics model struggles to learn spatiotemporal stochasticity and [35] reports convergence issues, which the deterministic path in the RSSM helps to address. This determinism manifests as blurry rollouts in stochastic environments.

While stochastic video prediction [2; 9; 23; 26] may appear similar to the considered problem, it is less challenging since those models only try to learn a single entangled scene representation. Moreover, those representations are known to be less effective for multi-object downstream tasks such as robotic manipulation [35].

3 Method

The SSM for the proposed object-centric world model, depicted in Figure 1b, receives high-dimensional observations o_t at discrete time steps $0 \leq t < T$ of the K object slots $\mathbf{s}_t := s_t^{1:K} \in \mathbb{R}^{\boldsymbol{M}}$.[1] It receives actions $a_t \in \mathbb{R}^N$ (e.g., a pick-and-place instruction) at each step. The generative process for a video clip of length H conditioned on T previous frames is:

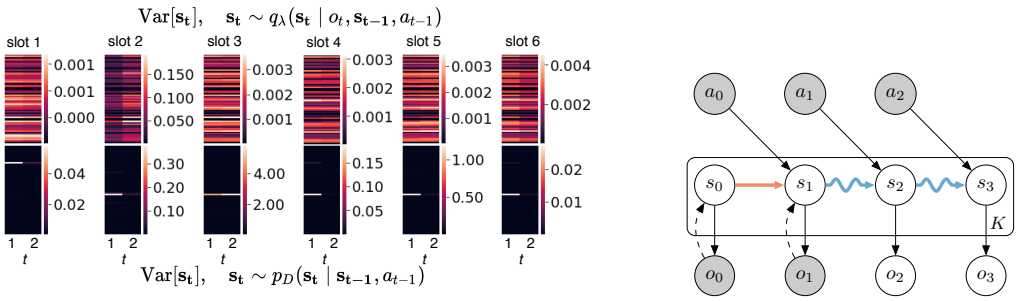
$$\begin{aligned} &p(o_{T \leq t \leq H}, \mathbf{s}_{\leq T+H} \mid o_{<T}, a_{\leq T+H-1}) \\ &= p_O(\mathbf{s}_0 \mid o_0) \prod_{t=1}^{T-1} p_O(\mathbf{s}_t \mid o_t, \mathbf{s}_{t-1}, a_{t-1}) \prod_{t=T}^{T+H} p(o_t \mid \mathbf{s}_t) p_D(\mathbf{s}_t \mid \mathbf{s}_{t-1}, a_{t-1}). \end{aligned} \tag{1}$$

The state space model consists of object discovery posteriors $p_O(\mathbf{s}_0 \mid o_0)$ and $p_O(\mathbf{s}_t \mid o_t, \mathbf{s}_{t-1}, a_{t-1})$, an observation model $p(o_t \mid \mathbf{s}_t)$, and a latent dynamics model for future rollouts $p_D(\mathbf{s}_t \mid \mathbf{s}_{t-1}, a_{t-1})$. The object discovery prior $p_O(\mathbf{s}_t \mid \mathbf{s}_{t-1}, a_{t-1})$ is not shown in Eq. 1 since it is just used to initialize the discovery posterior parameters during iterative inference. Each distribution is defined symmetrically over the K object slots. We use a Gaussian mixture model with fixed global variance to render images: $p(o_t \mid \mathbf{s}_t) = \sum_{k=1}^{K} \pi_t^k \, \mathcal{N}(\mu_t^k, \sigma^2)$ where an object-centric decoder [38] maps $\mathbf{s}_t$ to $(\boldsymbol{\pi}_t, \boldsymbol{\mu}_t)$. Following standard practice for variational inference [18; 31], we approximate the intractable object discovery posterior distribution with $q_O(\mathbf{s}_{\leq T-1} \mid o_{\leq T-1}, \mathbf{s}_{\leq T-2}, a_{\leq T-2})$.

Object discovery prior We define $p_O(\mathbf{s}_t \mid \mathbf{s}_{t-1}, a_{t-1})$ to be a symmetric product of K Gaussians. The K means $\boldsymbol{\mu}_{t,O}$ for this distribution are output by an object-centric interaction network $f(\mathbf{s}_{t-1}, a_{t-1})$ (see [34; 35]). In line with our argument that the discovery distributions do not model spatiotemporal stochasticity, we do not use past information to predict the K variances. Instead, we learn the variance σ_O^2 as a static model parameter shared across time steps $t < T$ and K slots.

Object discovery posterior Like the discovery prior, the discovery posterior is a symmetric product of K Gaussians. Instead of using iterative amortized inference [28; 15] we adopt its sequential extension, amortized variational filtering EM inference [27]. At each step $0 \leq t < T$ we make an initial guess $\boldsymbol{\lambda}_t^{(1)}$ for the parameters of the variational posterior which a refinement network [15] iteratively updates I times. Randomly sampling from the initial posterior guess during step $i = 1$ breaks the symmetry amongst the slots and establishes the object-slot assignment. The refinement

[1] We use boldface to indicate random variables that are repeated K times and $s_{\leq T} := s_{0:T}$.



(a) (b)

Figure 1: (a) The variance for each latent unit for $K = 6$ 64-dim slots at steps $t = 1, 2$ of a video. The dynamics model p_D (bottom) learns to only predict high variance (white) for latent attributes that may change over time. The object discovery posterior variance (top) is uniform and has low magnitude across latent units and slots. We are able to successfully fit dynamics variance (e.g., caused by action noise) by learning the discovery and dynamics variances separately by design. (b) The object-centric state space model for two provided frames and two rollout steps. The object discovery prior links $\mathbf{s_0}$ to $\mathbf{s_1}$ and the dynamics model is shown with snake arrows. Circles represent stochastic variables. Solid lines denote the generative process and dashed lines the inference model.

network takes in several inputs, the most important being $\mathcal{L}_{t,\text{o.d.}}$ (o.d. := object discovery), an estimate of the quality of the current posterior parameters at step i of inference:

$$\mathcal{L}^{(i)}_{t,\text{o.d.}} = -\mathbb{E}_{\mathbf{s}_t^{(i)} \sim \mathcal{N}(\boldsymbol{\lambda}_t^{(i)})}[\log p(o_t \mid \mathbf{s}_t^{(i)})] + D_{KL}\big(\mathcal{N}(\boldsymbol{\lambda}_t^{(i)}) \parallel p_O(\mathbf{s}_t \mid \mathbf{s}_{t-1}, a_{t-1})\big). \tag{2}$$

Note that OP3 puts the dynamics prior in the KL term of Eq. 2 whereas we use the discovery prior. The initial guesses for the posterior are $\boldsymbol{\lambda}_0^{(1)} = \{\boldsymbol{\mu}_0, \boldsymbol{\sigma}_O^2\}$ where $\boldsymbol{\mu}_0$ is learned as a model parameter and $\boldsymbol{\lambda}_t^{(1)} = \{\boldsymbol{\mu}_{t,O}, \boldsymbol{\sigma}_O^2\}$, the discovery prior parameters. The discovery prior means $\boldsymbol{\mu}_{t,O}$ help encourage the discovery posterior to maintain the object-slot assignment from the previous time step [35]. Although this introduces an *implicit* dependence on past information for the posterior mean, we find that the posterior variance is still chiefly influenced by o_t during inference, while the dynamics variance depends entirely on the past (Figure 1a).

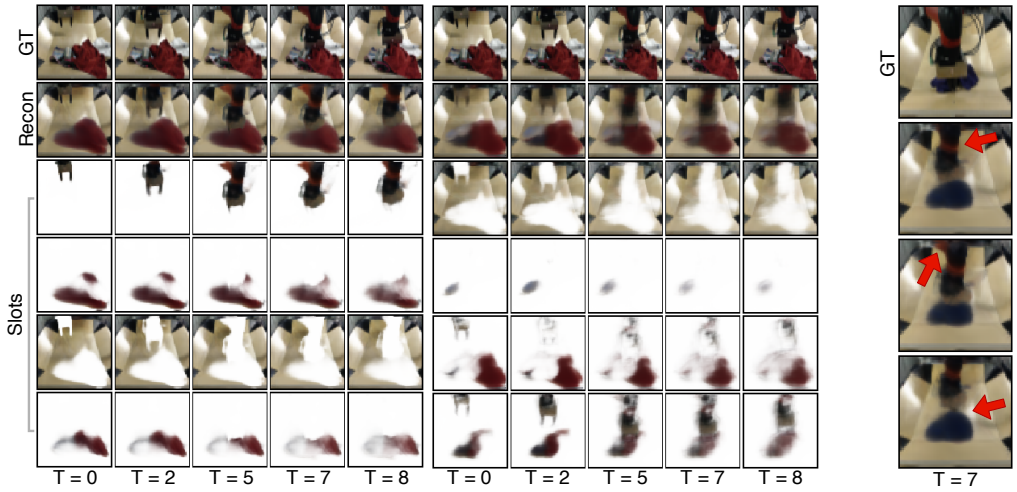
Dynamics model The latent dynamics model $p_D(\mathbf{s}_t \mid \mathbf{s}_{t-1}, a_{t-1})$ is likewise a product of K Gaussians. We use the same spatiotemporal interaction network from the discovery prior to compute the K means of the dynamics distribution, sharing the parameters. We then extend the interaction network to also output K variances $\boldsymbol{\sigma}_{t,D}^2$ so that the dynamics distribution is parameterized as $\{\boldsymbol{\mu}_{t,O}, \boldsymbol{\sigma}_{t,D}^2\}$. We found that sharing the parameters for the means improves sample efficiency and rollout sharpness. The KL term in Eq. 2 helps improve the visual quality of rollouts by pushing the means of the dynamics distribution to match the discovery posterior.

3.1 Training

The dynamics variances do not show up in the KL term in Eq. 2, so we need to devise an additional loss term for them. While a simple approach to fit the dynamics variances could be to rollout the dynamics model for H steps and compute reconstruction losses, we observe that in stochastic environments minimizing average reconstruction error collapses the dynamics variance resulting in blurry rollouts. We can prevent the dynamics variance from collapsing with a best-of-many-rollouts (BMR) objective over H future steps:

$$\mathcal{L}_{\text{BMR}} = \sum_{t=0}^{T-1}\Big(\sum_{i=1}^{I}\frac{i}{I}\mathcal{L}^{(i)}_{t,\text{o.d.}}\Big) - \max_j\Big\{\sum_{t=T}^{T+H}\mathbb{E}[\log p(o_t \mid \mathbf{s}_t^{(j)})]\Big\}_{j=1}^{J}, \tag{3}$$

with $\mathbf{s}_t^{(j)}$ sampled from the object discovery posterior for $t < T$ and $\mathbf{s}_t^{(j)}$ sampled from the dynamics model for $t \geq T$. The `max` over J encourages fitting the dynamics variance so as to increase the chance of drawing a sample that achieves the best possible reconstruction loss. We approximate the objective by having each of the J future rollouts share a single sample through steps $t < T$ to speed



(a) Ours (b) OP3 (c) Diversity

Figure 2: (a-b) Ours is less blurry and achieves a cleaner decomposition while action noise causes the visual quality of OP3's predictions to degrade. We only show 4 / 6 slots to conserve space. (c) Ours can predict multiple physically plausible robot arm configurations. Best viewed in color.

Table 1: BAIR (Realism / Diversity / Accuracy)

Model	FVD (↓)	$(\text{Best - Worst})_{100}$ SSIM (↑)	SSIM / PSNR (↑)
VRNN†	472.5 ± 15.2	0.089	0.72 / 19.72
OP3	642.3 ± 27.2	0.002	0.76 / 21.61
Ours	$\mathbf{564.8 \pm 24.3}$	**0.053**	**0.79 / 22.39**

† No object discovery

up training. We note that the objective proposed in [29] evaluates a max over samples *after each step*, which only fits the distribution over a single step while we take the `max` over entire rollouts. The i/I term down-weights the importance of early steps of iterative inference [15].

4 Experiments

We evaluate on the BAIR `towel_pick_30k` [10; 35; 36] stochastic video prediction benchmark with added $\mathcal{N}(0, 0.05^2)$ action noise and condition on two frames and rollout eight future frames. The main baseline is OP3 [35], the state-of-the-art object-centric world model for realistic videos. Both models use $K = 6$ slots and we use $J = 5$ rollouts for the BMR objective (see appendix for full details and extra qualitative examples). Following similar studies [20; 21] we also use a conditional VRNN [6] as a baseline with comparable capacity to the considered models—but it does *not* do object discovery. In Table 1, we compare the ability to fit spatiotemporal stochasticity by computing the median per-frame SSIM and PSNR of the best out of 100 future rollouts averaged over time steps (**accuracy**) [2; 23; 36], the difference between best and worst SSIM out of 100 rollouts (**diversity**), and the Fréchet Video Distance (FVD) [33] averaged over five populations with sample size 256 (**realism**). Ours outperforms OP3 in all metrics—notice that OP3's rollouts are all virtually identical (diversity = 0.002). Our model does not suffer from training instabilities, unlike OP3 [35]. We compare object decomposition quality by visualizing the object slots (Figure 2) instead of computing ARI scores since ground truth masks are unavailable.

5 Discussion

We have introduced a perceptual-grouping-based world model that uses a latent dynamics model for stochastic rollouts. In a more complete version of this work we will add model ablations, more environments, and comparisons to other relevant object-centric world models (e.g., G-SWM [24]). In future work we will consider more sophisticated dynamics models to handle multi-modal prediction.

References

[1] Antonova, R., Devlin, S., Hofmann, K., and Kragic, D. Benchmarking unsupervised representation learning for continuous control. 2020. URL https://openreview.net/forum?id=UKqIsgNeah7.

[2] Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[3] Bapst, V., Sanchez-Gonzalez, A., Doersch, C., Stachenfeld, K. L., Kohli, P., Battaglia, P. W., and Hamrick, J. B. Structured agents for physical construction. *arXiv preprint arXiv:1904.03177*, 2019.

[4] Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pp. 4502–4510, 2016.

[5] Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., and Lerchner, A. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[6] Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

[7] Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[8] Davidson, G. and Lake, B. M. Investigating simple object representations in model-free deep reinforcement learning. feb 2020. URL https://arxiv.org/abs/2002.06703.

[9] Denton, E. and Fergus, R. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, 2018.

[10] Ebert, F., Dasari, S., Lee, A. X., Levine, S., and Finn, C. Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning. *arXiv preprint arXiv:1810.03043*, 2018.

[11] Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019.

[12] Eslami, S. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Hinton, G. E., et al. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pp. 3225–3233, 2016.

[13] Greff, K., Rasmus, A., Berglund, M., Hao, T., Valpola, H., and Schmidhuber, J. Tagger: Deep unsupervised perceptual grouping. In *Advances in Neural Information Processing Systems*, pp. 4484–4492, 2016.

[14] Greff, K., van Steenkiste, S., and Schmidhuber, J. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pp. 6691–6701, 2017.

[15] Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-object representation learning with iterative variational inference. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2424–2433, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/greff19a.html.

[16] Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/hafner19a.html.

[17] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Kipf, T., van der Pol, E., and Welling, M. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

[20] Kosiorek, A., Kim, H., Teh, Y. W., and Posner, I. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pp. 8606–8616, 2018.

[21] Kossen, J., Stelzner, K., Hussing, M., Voelcker, C., and Kersting, K. Structured object-aware physics prediction for video modeling and planning. *arXiv preprint arXiv:1910.02425*, 2019.

[22] Kulkarni, T. D., Gupta, A., Ionescu, C., Borgeaud, S., Reynolds, M., Zisserman, A., and Mnih, V. Unsupervised learning of object keypoints for perception and control. In *Advances in neural information processing systems*, pp. 10724–10734, 2019.

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

[24] Lin, Z., Wu, Y.-F., Peri, S., Fu, B., Jiang, J., and Ahn, S. Improving generative imagination in object-centric world models, 2020.

[25] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055*, 2020.

[26] Luc, P., Clark, A., Dieleman, S., Casas, D. d. L., Doron, Y., Cassirer, A., and Simonyan, K. Transformation-based adversarial video prediction on large-scale data. *arXiv preprint arXiv:2003.04035*, 2020.

[27] Marino, J., Cvitkovic, M., and Yue, Y. A general method for amortizing variational filtering. In *Advances in Neural Information Processing Systems*, pp. 7857–7868, 2018.

[28] Marino, J., Yue, Y., and Mandt, S. Iterative amortized inference. *35th International Conference on Machine Learning, ICML 2018*, 8:5444–5462, 2018.

[29] Minderer, M., Sun, C., Villegas, R., Cole, F., Murphy, K. P., and Lee, H. Unsupervised learning of object structure and dynamics from videos. In *Advances in Neural Information Processing Systems*, pp. 92–102, 2019.

[30] Rezende, D. J. and Viola, F. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

[31] Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *31st International Conference on Machine Learning, ICML 2014*, 4:3057–3070, 2014.

[32] Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pp. 3483–3491, 2015.

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

[34] Van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. *arXiv preprint arXiv:1802.10353*, 2018.

[35] Veerapaneni, R., Co-Reyes, J. D., Chang, M., Janner, M., Finn, C., Wu, J., Tenenbaum, J. B., and Levine, S. Entity abstraction in visual model-based reinforcement learning. *arXiv preprint arXiv:1910.12827*, 2019.

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

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

[38] Watters, N., Matthey, L., Burgess, C. P., and Lerchner, A. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

[39] Zhu, G., Huang, Z., and Zhang, C. Object-oriented dynamics predictor. In *Advances in Neural Information Processing Systems*, pp. 9804–9815, 2018.

Table 2: State space model details

Distribution	Time steps	Implementation
Object discovery prior	$0 < t < T$	$p_O(\mathbf{s}_t \mid \mathbf{s}_{t-1}, a_{t-1}) := \{\boldsymbol{\mu}_{t,O}, \boldsymbol{\sigma}_O^2\}$
Object discovery posterior	$t < T$	$q_O(\mathbf{s}_t \mid o_t, \mathbf{s}_{t-1}, a_{t-1}) := \prod_k q_O(s_t^k \mid o_t, \mathbf{s}_{t-1}, a_{t-1})$
Object discovery posterior guess	$t = 0$	$\boldsymbol{\lambda}_0^{(1)} := \{\boldsymbol{\mu}_0, \boldsymbol{\sigma}_O^2\}$
Object discovery posterior guess	$0 < t < T$	$\boldsymbol{\lambda}_t^{(1)} := \{\boldsymbol{\mu}_{t,O}, \boldsymbol{\sigma}_O^2\}$
Observation model	$t \leq T + H$	$p(o_t \mid \mathbf{s_t}) := \sum_k \pi_t^k \mathcal{N}(\mu_t^k, \sigma^2)$
Dynamics model	$t \geq T$	$p_D(\mathbf{s_t} \mid \mathbf{s_{t-1}}, a_{t-1}) := \{\boldsymbol{\mu}_{t,O}, \boldsymbol{\sigma}_{t,D}^2\}$

A Implementation details

The proposed state space model is summarized in Table 2 as a reference. In the remainder of this section, we present implementation details and highlight the differences (if any) from prior works. We refer the reader to the appendix of [35] for OP3's implementation details.

All models use the ELU activation function [7] and convolutional layers use a stride equal to 1 and padding equal to 2 unless otherwise noted. We use $|s_t^k| = 64$.

Refinement network The output of the refinement network is $\boldsymbol{\delta}_t^{(i)}$ which is used to make an additive update to $\boldsymbol{\lambda}_t^{(i)}$.

Refinement Network

Type	Size/*Ch.*	Act. Func.	Comment
64×64 inputs	*13*		
Conv 3×3	*32*	ELU	
Conv 3×3	*32*	ELU	
Conv 3×3	*32*	ELU	
Avg. Pool 2d	4×4		
MLP	$8192 \rightarrow 128$	ELU	Flattened input
$[\boldsymbol{\lambda}_t^{(i)}, \nabla_{\boldsymbol{\lambda}_t} \mathcal{L}]$	128		Concat
MLP	$384 \rightarrow 128$	ELU	
LSTM	$128 \rightarrow 128$	Tanh	
MLP	$128 \rightarrow 128$	None	$\boldsymbol{\delta}_t^{(i)}$

We use the following inputs to the refinement network, where LN means Layernorm and SG means stop gradients (the LN and SG is to help stabilize training [15]).

Convolutional Inputs

Description	Formula	LN	SG	Ch.
image	x_t			3
means	$\boldsymbol{\mu_t}$			3
mask	$\boldsymbol{\pi_t}$			1
gradient of means	$\nabla_{\boldsymbol{\mu}_t} \mathcal{L}$	✓	✓	3
gradient of mask	$\nabla_{\boldsymbol{\pi}_t} \mathcal{L}$	✓	✓	1
coordinate channels				2
			total:	13

Vector Inputs

Description	Formula	LN	SG
posterior	$\boldsymbol{\lambda}_t^{(i)}$		
gradient of posterior	$\nabla_{\boldsymbol{\lambda}_t} \mathcal{L}$	✓	✓

The posterior parameters $\boldsymbol{\lambda}_t^{(i)}$ and their gradients are flat vectors, and we concatenate them to the output of the convolutional part of the refinement network, then project the result to match the input dimension of the refinement LSTM with an MLP. Note that in this work, we do not use the mask logits, mask posterior, gradient of mask posterior, or the pixelwise-likelihood from [15] as auxiliary inputs. Based on the ablation studies conducted in [15], these have little impact on performance and unnecessarily increase the number of parameters in the refinement network.

Spatial Broadcast Decoder

Type	Size/*Ch.*	Act. Func.	Comment
Input: $\mathbf{s}_t$	64		
Broadcast	64+*2*		+ coordinates
Conv 5×5	*32*	ELU	
Conv 5×5	*32*	ELU	
Conv 5×5	*32*	ELU	
Conv 5×5	*32*	ELU	
Conv 5×5	*4*	None	RGB + Mask

Interaction network The implementation of the action-conditional interaction network differs from OP3's [35] since we do not split the hidden state into stochastic and deterministic components. Moreover, we use a 128-dim GRU to embed the history $\{s_0^k, \dots, s_{t-2}^k, s_{t-1}^k\}$ to help retain information over multiple stochastic transitions [16]. In future work, we will include an ablation study to demonstrate its impact.

Concretely, we have:

$$\tilde{s}_{\leq t-1}^k = f_o(s_{\leq t-1}^k) \qquad \tilde{s}_{t-1}^k = \text{GRU}(\tilde{s}_{\leq t-1}^k) \qquad \tilde{a}_{t-1} = f_a(a_{t-1}) \qquad \tilde{s}_{t-1,\text{act}}^k = f_{ao}(\tilde{s}_{t-1}^k, \tilde{a}_{t-1})$$

$$s_{t-1,\text{interact}}^k = \sum_{i \neq k}^{K} f_{oo}(\tilde{s}_{t-1,\text{act}}^i, \tilde{s}_{t-1,\text{act}}^k) \qquad \hat{s}_{t-1}^k = f_{\text{comb}}(\tilde{s}_{t-1,\text{act}}^k, s_{t-1,\text{interact}}^k),$$

where $f_{ao}(\cdot,\cdot) := f_{\text{act-att}} \cdot f_{\text{act-eff}}$ computes how and to what extent the action effects a particular object. Likewise, $f_{oo}(\cdot,\cdot) := f_{\text{obj-att}} \cdot f_{\text{obj-eff}}$ computes how and to what extent each object effects the others. All functions are parameterized by single layer MLPs.

Interaction Network

Function	Output	Act. Func.	MLP Size
$f_o(s_{\leq t-1}^k)$	$\tilde{s}_{\leq t-1}^k$	ELU	128
$\text{GRU}(\tilde{s}_{\leq t-1}^k)$	$\tilde{s}_{t-1}^k$	Tanh	128
$f_a(a_t)$	$\tilde{a}_{t-1}$	ELU	32
$f_{\text{act-eff}}(\tilde{s}_{t-1}^k, \tilde{a}_{t-1})$	$\tilde{s}_{t-1,\text{eff}}^k$	ELU	128
$f_{\text{act-att}}(\tilde{s}_{t-1,\text{eff}}^k)$		Sigmoid	128
$f_{\text{obj-eff}}(\tilde{s}_{t-1,\text{act}}^i, \tilde{s}_{t-1,\text{act}}^j)$	$\tilde{s}_{t-1,\text{eff}}^i$	ELU	256
$f_{\text{obj-att}}(\tilde{s}_{t-1,\text{eff}}^i)$		Sigmoid	256
$f_{\text{comb}}(\tilde{s}_{t-1,\text{act}}^k, s_{t-1,\text{interact}}^k)$		ELU	256
MLP	$\boldsymbol{\mu}_t^{\text{obj}}$	None	64
MLP	$\boldsymbol{\sigma}_{t,D}^2$	None	64

A.1 VRNN Baseline

We implement the VRNN [6] so that it has similar model capacity. The resultant sequential VAE resembles the SVG-LP model [9]. The convolutional image encoder $\varphi^{\mathbf{x}}$ is based on the component encoder from GENESIS [11]:

Component Encoder

Type	Size/*Ch.*	Act. Func.	Comment
Input: x_t	3		
Conv 5×5	*32*	ELU	stride 1
Conv 5×5	*32*	ELU	stride 2
Conv 5×5	*64*	ELU	stride 1
Conv 5×5	*64*	ELU	stride 2
Conv 5×5	*64*	ELU	stride 1
MLP	$16384 \rightarrow 256$	ELU	

The decoder φ^{dec} is the same spatial broadcast decoder described above. The image likelihood is a Gaussian with standard deviation fixed at 0.3, and the latent variable encoder φ^z, the encoder φ^{enc}, and the prior network φ^{prior} are 2-layer MLPs. We use an LSTM for the deterministic recurrent backbone. Actions are concatenated to the inputs for φ^z and φ^{prior}. All networks except $\varphi^{\mathbf{x}}$ use 256 hidden nodes and ELU nonlinearities. We chose the architecture hyperparameters based on the VRNN baseline from SQAIR [20].

B Training details

B.1 GECO

We adaptively balance the reconstruction and KL terms with GECO [30], which reformulates the objective as a minimization of KL terms subject to a constraint on the reconstruction error. The full objective is modified for GECO [30] as:

$$\mathcal{L}_{\text{full}} = \sum_{t=1}^{T} \Bigg(\sum_{i=1}^{I} \frac{i}{I} D_{KL}\big(\mathcal{N}(\boldsymbol{\lambda}_t^{(i)}) \parallel p(\mathbf{s}_t \mid \mathbf{s}_{t-1}, a_{t-1})\big) - \zeta\big(C + \mathbb{E}_{\mathbf{s}_t^{(i)} \sim \mathcal{N}(\boldsymbol{\lambda}_t^{(i)})}[\log p(o_t \mid \mathbf{s}_t^{(i)})]\big) \Bigg)$$
$$- \max_j \Bigg(\sum_{t=T+1}^{T+d} \zeta(C + \mathbb{E}[\log p(o_t \mid \mathbf{s}_t^{(j)})]) \Bigg\}_{j=1}^{J} \Bigg),$$

where ζ is a Lagrange parameter that penalizes the model when the reconstruction error is higher than a manually-specified threshold C. We use an exponential moving average C_{EMA} with parameter $\alpha = 0.99$ to keep track of the difference between the reconstruction error of the mini-batch and C [30]. The Lagrange parameter is updated at every step with $\zeta' = \zeta -$ 1e-6 C_{EMA}. For numerical stability, we use $\texttt{softplus}(\zeta)$ when computing the GECO update and constrain $\zeta \geq 0.55$ so that $\texttt{softplus}(\zeta)$ is always greater than or equal to 1.

B.2 Hyperparameters

All models are trained with the ADAM optimizer [17] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay, and a learning rate of 3e-4. We use gradient clipping where if the norm of the global gradient exceeds 5.0, the gradient is scaled down to that norm [15].

The standard deviation used by the Gaussian mixture model image likelihood is set asymmetrically following MONet [5] and GENESIS [11]. We observed that this encouraged the model to only use a single object slot for the background. We set σ_1 for the first object slot to 0.09 and $\sigma_{2:K}$ to 0.11. Note that the model is still fully object equivariant, and therefore the background is not guaranteed to be assigned to the first object slot at time step 0.

To choose the GECO reconstruction threshold C we first conducted a single training run with a guess for C, then adjusted the guess so that the model achieves and maintains reasonable reconstruction quality. All models use the same C.

We use a curriculum where models are first trained to rollout one step conditioned on two frames for 200K train steps, then are trained to rollout three, five, and then eight steps for 50K train steps each. We use a mini-batch size of 16 for the first curriculum stage then 10 for the remaining stages. We set $C = -25500$ for the first stage and increase it to -26000 afterwards. Note that during the last 25K train steps we increase the number of sampled rollouts from $J = 5$ to $J = 10$, which helps address multi-modality.

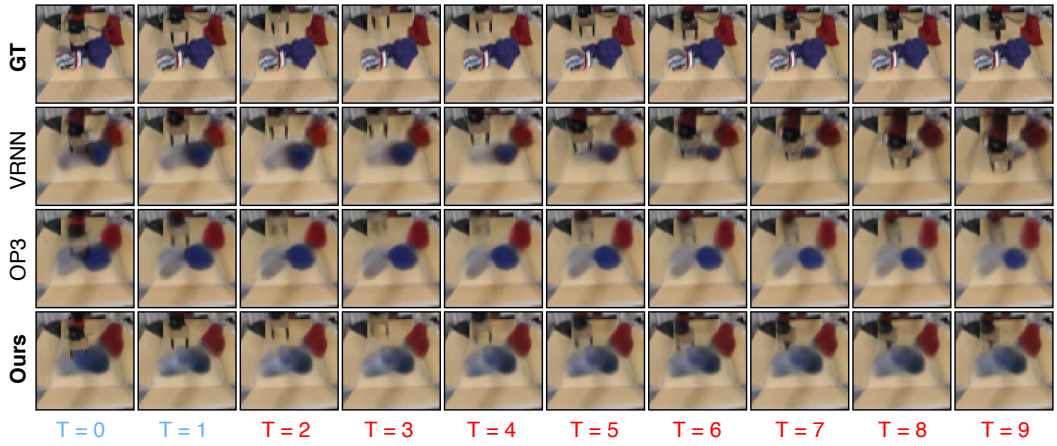


Figure 3: The VRNN's prior struggles to maintain the shape and color of the clothing items for the duration of the rollout and has difficulty with learning the robot arm dynamics. However, it is able to produce the most realistic rollouts (lowest FVD) by virtue of training with the standard CVAE objective [32]. While OP3 is able to handle clothing items better than the VRNN, the robot arm blurs out due to OP3's inability to handle stochastic dynamics. Ours successfully learns the noisy robot arm dynamics and maintains the appearance of the static clothing items over many time steps.

C Additional results

We show a side-by-side comparison of rollouts from the VRNN baseline, OP3, and our model in Figure 3. Figure 4 depicts additional temporal object slot decompositions and Figure 5 shows more examples of multi-future rollouts given a single set of context frames. The VRNN is able to produce rollouts which are more *realistic* yet *less accurate* than the proposal model (Figure 3,Table 1). We can attribute the lower realism to a shortcoming of the BMR objective; by only relying on reconstruction losses for fitting the distribution over future rollouts, the realism of the generation quality will be inferior to the VRNN which uses the standard CVAE objective [32]. The low accuracy of the VRNN can be attributed to difficulty with maintaining the color and shape of the cloth items over time and with predicting arm motion given the provided noisy action. Since our model and OP3 do not have the same problems, we believe this can be explained by the VRNN's use of a single entangled scene representation.

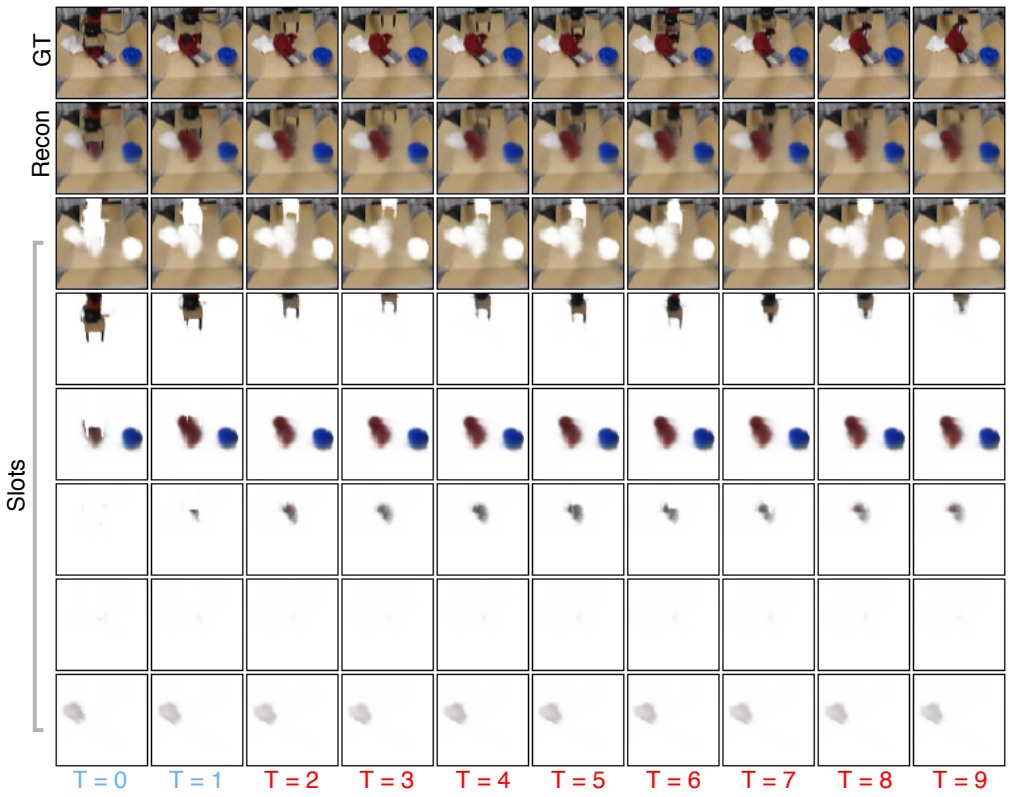


Figure 4: The white and multi-colored clothing items, as well as the robot arm and background, are each assigned to a unique slot. Two cloth items (the red and blue ones) are assigned to a single slot; explicitly enforcing that only one object is assigned to a slot is one potential direction for improvement. Ours maintains the object-slot assignment over time in addition to capturing the spatiotemporal uncertainty of the robot arm dynamics.

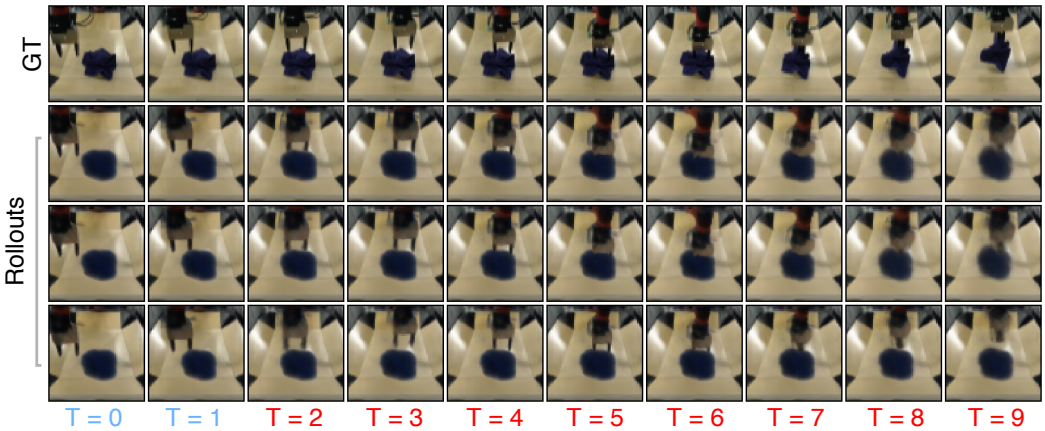


Figure 5: Three random rollouts. The rollouts shown in the second and third rows are highly similar with minor spatial variation, whereas the fourth row shows a distinct physically plausible future.