FOCUS: Object-Centric World Models for Robotics Manipulation

Stefano Ferraro[1,*], Pietro Mazzaglia[1,*], Tim Verbelen[2], Bart Dhoedt[1]

***Abstract*—Understanding the world in terms of objects and the possible interactions with them is an important cognition ability, especially in robotic manipulation. However, learning such a structured world model, which also allows to accurately control the agent, remains a challenge. To address this, we propose FOCUS, a model-based agent that learns an object-centric world model, thanks to a novel exploration bonus that stems from the object-centric representation. Evaluating our approach on manipulation tasks across different settings, we show that object-centric world models allow the agent to solve tasks more efficiently and enable consistent exploration of robot-object interactions. Using a Franka Emika robot arm, we also showcase how FOCUS could be adopted in real-world settings. *Project website:* https://focus-manipulation.github.io/**

I. Introduction

For robot manipulators, the tasks we perform as humans are extremely challenging due to the high level of complexity in the interaction between the agent and the environment. In recent years, deep reinforcement learning (RL) has shown to be a promising approach for dealing with these challenging scenarios [28, 35, 23, 31, 27, 10].

Among RL algorithms, model-based approaches aspire to provide greater data efficiency, compared to the model-free counterparts [13, 17]. Adopting world models [16, 19], i.e. generative models that learn the environment dynamics by reconstructing the agent's observations, model-based agents have shown impressive performance across several domains [19, 41, 20], including real-world applications, such as robotics manipulation and locomotion [50].

However, world models that indistinctly reconstruct all information in the environment can suffer from several failure modes. For instance, in visual tasks, they can ignore small, but important features for predicting the future, such as little objects [46], or they can waste most of the model capacity on rich, but potentially irrelevant features, such as static backgrounds [7]. In the case of robot manipulation, this is problematic because the agent strongly needs to acquire information about the objects to manipulate in order to solve tasks.

Another challenge in RL for manipulation is engineering reward functions, able to drive the agent's learning toward task completion, as attempting to design dense reward feedback easily leads to faulty reward designs [1, 5, 26, 39]. One solution is to adopt sparse reward feedback, providing a positive reward only for successful task completion. However, these functions are challenging to optimize with RL, due to the difficulty of finding rewards in the environment and thus require appropriate exploration strategies, for which previous work has resorted to artificial curiosity mechanisms [36, 42].

Humans, on the other hand, tend to develop a structured mental model of the world by interacting with objects, registering specific features associated with objects, such as shape, color, etc [22, 12]. Since infancy, toddlers learn this by actively engaging with objects and manipulating them with their hands, discovering object-centric views that allow them to build an accurate mental model [49, 48, 11].

Inspired by the principle that objects should be of primary importance in the agent's world model, we present **FOCUS**, a model-based RL agent that learns an object-centric representation of the world and is able to explore object interactions.

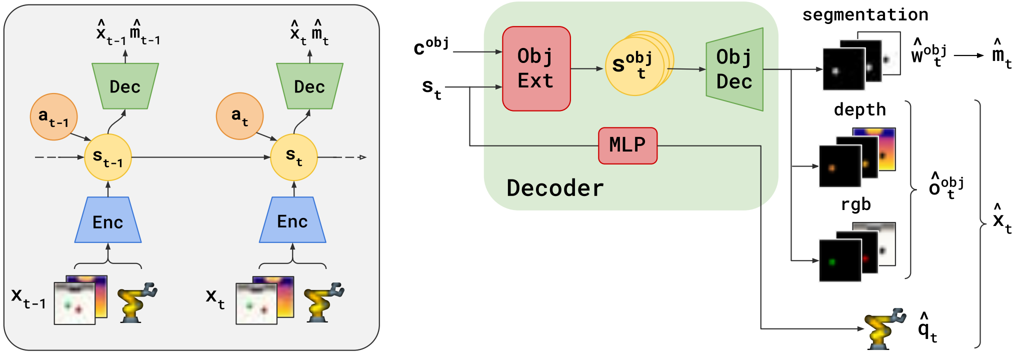


Fig. 1: **FOCUS.** Overview of the components of the object-centric world model learned by FOCUS. *(Left)* Overall composition of the model. *(Right)* Detailed view of the decoder, including the object latent extractor and object decoder.

[*] Equal contribution
[1] Ghent University - name.surname@ugent.be
[2] VERSES - name.surname@verses.io

Contributions Our contributions can be summarized as:

- A new object-centric latent dynamics model, which learns the dynamics of the environment, while discriminating object representations into distinct latent vectors.
- An object-centric exploration strategy, which encourages interactions with the objects, by maximizing the entropy of the latent object representation, that facilitates finding sparsely rewarded goals in the environment.
- We show the effectiveness of our object-centric model, qualitatively showing that FOCUS captures crucial information about objects and that the representation allows faster learning of manipulation tasks.

Background material relative to the work can be found in Appendix A.

II. Method

Reinforcement Learning. In RL, the agent receives inputs x from the environment and can interact through actions a. The objective of the agent is to maximize the discounted sum of rewards $\sum_t \gamma^t r_t$, where t indicates discrete timesteps. To do so, RL agents learn an optimal policy $\pi(a|x)$ outputting actions that maximize the expected cumulative discounted reward over time, generally estimated using a critic function, which can be either a state-value function $v(x)$ or an action-value function $q(x, a)$ [17, 13]. World models [16] additionally learn a generative model of the environment, capturing the environment dynamics into a latent space, which can be used to learn the actor and critic functions using imaginary rollouts [19, 20] or to actively plan at each action [43, 21, 41], which can lead to higher data efficiency in solving the task.

Object-centric World Model. The agent observes the environment through the inputs $x_t = \{o_t, q_t\}$ it receives at each interaction, where we can distinguish the (visual) observations o_t, e.g. camera RGB and depth, from the proprioceptive information q_t, e.g. the robot joint states and velocities. This information is processed by the agent through an encoder model $e_t = f(x_t)$, which can be instantiated as the concatenation of the outputs of a CNN for high-dimensional observations and an MLP for low-dimensional proprioception.

The world model aims to capture the dynamics of the inputs into a latent state s_t. In previous work, this is achieved by reconstructing the inputs using an observation decoder. With FOCUS, we are interested in separating object-specific information into separate latent representations s_t^{obj}. For this reason, we instantiate two object-conditioned components: an *object latent extractor* and an *object decoder*.

Models. Overall, the learned world model, shown in Fig. 1 is composed of the following components:

$$\begin{aligned}
\text{Encoder:} \quad & e_t = f(x_t), \\
\text{Posterior:} \quad & p_\phi(s_{t+1}|s_t, a_t, e_{t+1}), \\
\text{Prior:} \quad & p_\phi(s_{t+1}|s_t, a_t), \\
\text{Proprio decoder:} \quad & p_\theta(\hat{q}_t|s_t), \\
\text{Object latent extractor:} \quad & p_\theta(s_t^{obj}|s_t, c^{obj}), \\
\text{Object decoder:} \quad & p_\theta(\hat{o}_t^{obj}, w_t^{obj}|s_t^{obj}).
\end{aligned}$$

We adopt a recurrent state-space model (RSSM) [18] for the dynamics components, i.e. prior and posterior, which extracts a latent state s_t made of a deterministic and a stochastic component. Proprioceptive information $\hat{q}_t$ is decoded out of the latent state s_t, using an MLP.

For each object in the scene, the *object latent extractor* receives the world model latent state s_t and a vector identifying the object c^{obj}, and extracts an object-centric latent s_t^{obj}. Given such an object latent, the *object decoder* reconstructs the object-related observation information $\hat{o_t^{obj}}$, where the information that is irrelevant to the object is masked out. The object decoder also outputs one-dimensional "unnormalized weights" w_t^{obj}, which represent object-specific per-pixel logits for segmenting the objects. The overall scene segmentation is obtained by applying a softmax among all object weights, where each pixel is assigned to the object that outputs the highest weight (see following Objective paragraph)[1]. For the object-conditioning vector c^{obj} we adopt a one-hot vector identifying the object instance.

Objective. The model is trained end-to-end by minimizing the following loss:

$$\mathcal{L}_{\text{wm}} = \mathcal{L}_{\text{dyn}} + \mathcal{L}_{\text{proprio}} + \mathcal{L}_{\text{obj}}. \tag{1}$$

The dynamics minimizes the Kullback–Leibler (KL) divergence between posterior and prior:

$$\mathcal{L}_{\text{dyn}} = D_{\text{KL}}[p_\phi(s_{t+1}|s_t, a_t, e_{t+1})||p_\phi(s_{t+1}|s_t, a_t)]. \tag{2}$$

The proprioceptive decoder learns to reconstruct proprio states, by minimizing a negative log-likelihood (NLL) loss:

$$\mathcal{L}_{\text{proprio}} = -\log p_\theta(\hat{q}_t|s_t) \tag{3}$$

The object decoder learns to reconstruct object-centric information, outputting "object weights" for the segmentation mask and reconstructing observations. The observation is masked via an object-specific mask m_t^i, to focus only on the i-th object information in the loss. The object decoder loss is:

$$\mathcal{L}_{\text{obj}} = -\log \underbrace{p(\hat{m_t})}_{\text{mask}} - \log \sum_{i=0}^{N} \underbrace{m_t^i p_\theta(\hat{x_t^i}|s_t^i)}_{\text{masked reconstruction}} \tag{4}$$

where the overall segmentation mask is obtained as:

$$\hat{m_t} = \text{softmax}(w_t^1, ..., w_t^N) \tag{5}$$

with N being the object instances. By minimizing the NLL of $masked\ reconstruction$, the object-decoder ensures that each object latent s^i focuses on capturing only its relevant information, as the reconstructions obtained from the latent are masked per object. Furthermore, objects compete for occupying their correct space in the scene (in pixel space), through the $mask$ loss.

Further details about the model architecture, the objective, and the optimization process are provided in Appendix.

[1] The scene, with objects masked out, is also considered a "special object".

Object-centric Exploration. State maximum entropy approaches for RL [33, 45, 29] learn an environment representation, on top of which they compute an entropy estimate that is maximized by the agent's actor to foster exploration. Given our object-centric representation, we can incentivize exploration towards object interactions and discovery of novel object views, by having the agent maximize the entropy over the object latents.

In order to estimate the entropy value over batches, we apply a K-NN particle-based estimator [47] on top of the object latent representation. By maximizing the overall entropy, with respect to all objects in the scene, we derive the following reward for object-centric exploration:

$$\begin{aligned} r_{\text{expl}} &= \sum_{obj=0}^{N} r_{\text{expl}}^{obj} \\ \text{where} \quad r_{\text{expl}}^{obj}(s) &\propto \sum_{i=1}^{K} \log \left\| s^{obj} - s_i^{obj} \right\|_2 \end{aligned} \tag{6}$$

where s^{obj} is extracted from s using the object latent extractor, s_i^{obj} is the i-th nearest neighbor to s^{obj}.

Crucially, as we use the world model to optimize actor and critic networks in imagination [19], the latent states in Equation 6 are states of imaginary trajectories, generated by the world model by following the actor's predicted actions.

III. Experiments

FOCUS object-centric world model can be used to improve task performance in robotics manipulation tasks and to meaningfully explore the environment using our object-driven exploration bonus. Our experiments aim to assess these qualities both qualitatively and quantitatively. Details about both the simulation and the real world setup are provided in appendix.

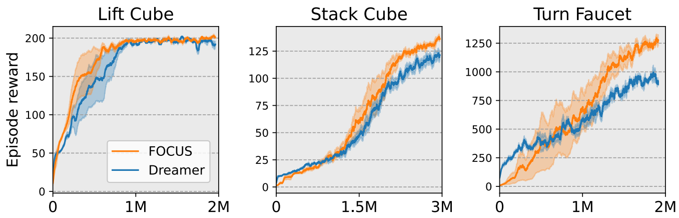


Fig. 2: **Manipulation tasks.** Comparing FOCUS and DreamerV2 over dense supervised manipulation tasks. (3 seeds)

Supervised tasks. In Figure 2, we compare the performance in terms of task episodic rewards that are obtained by FOCUS compared to DreamerV2 on supervised dense-reward versions of the tasks. The only difference between the two methods is the world model, so this study compares the quality of the world model to learn control policies. We observe that FOCUS converges faster across all tasks, reaching equal or better performance compared to DreamerV2.

Exploring object interactions. To evaluate the performance of FOCUS exploration approach, we picked exploration metrics that are related to the interaction with the object and the ability to find sparsely rewarded areas in the environment. Further details about those are provided in appendix.

In Table I, we compare FOCUS against two exploration strategies: Plan2Explore (P2E) [44] and a combination of the DreamerV2 algorithm and Active Pre-training (APT) [29, 41]. We also test a DreamerV2 agent, which aims to maximize sparse rewards in the environment [2].

We find that FOCUS shines across all simulation environments, showing strong object interaction performance, and outperforming all other approaches in most metrics. DreamerV2 manages to find sparse rewards, and thus explore object

[2] DreamerV2 is still able to explore, with limited capacity, thanks to the actor entropy term in the actor's objective [19].

		Contact	**Pos Disp**	**Ang Disp**	**Placement**					**Reward**		
					Up	↑	↓	←	→	P	L	T
Red cube	FOCUS	**0.04**	**0.15**	**1.86**	**3.5**	**37.45**	**0.06**	**0.15**	**0.01**	**0.67**	-	-
	Dreamer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-	-
	P2E	0.0	0.0	0.01	0.0	0.15	0.0	0.0	0.0	0.0	-	-
	APT	0.0	0.01	0.15	0.01	0.11	0.0	0.0	0.0	0.0	-	-
RG cubes	FOCUS	**0.1**	**0.4**	**4.46**	0.0	0.0	0.53	0.0	0.0	-	0.0	-
	Dreamer	0.0	0.01	0.01	0.0	0.0	0.16	0.0	0.0	-	0.0	-
	P2E	0.0	0.01	0.01	0.0	0.0	0.3	0.0	0.0	-	0.0	-
	APT	0.01	0.03	0.25	0.06	0.0	**5.05**	0.0	**0.03**	-	**2.0**	-
Faucet	FOCUS	**0.12**	-	1.22	-	-	-	-	-	-	-	**101.67**
	Dreamer	0.0	-	0.01	-	-	-	-	-	-	-	0.0
	P2E	0.1	-	**1.23**	-	-	-	-	-	-	-	35.33
	APT	0.08	-	0.98	-	-	-	-	-	-	-	34.33
Banana	FOCUS	**0.08**	**0.38**	**3.5**	**1.31**	**25.28**	0.72	**6.08**	**7.12**	**38.67**	**27.0**	-
	Dreamer	0.0	0.0	0.06	0.04	0.22	0.02	0.16	0.29	2.67	0.67	-
	P2E	0.02	0.11	0.97	0.64	8.48	0.22	4.78	4.14	19.67	14.67	-
	APT	0.01	0.04	0.4	0.24	2.0	**0.98**	1.3	2.1	6.67	5.33	-
Master Chef Can	FOCUS	**0.05**	**0.49**	**5.78**	**0.17**	**145.66**	**28.57**	**61.51**	**62.8**	507.67	**2.67**	-
	Dreamer	0.02	0.26	3.04	0.08	85.68	24.02	24.34	46.35	**628.67**	0.33	-
	P2E	0.03	0.29	3.26	0.12	69.95	30.63	29.72	36.0	207.67	2.33	-
	APT	0.01	0.14	1.56	0.06	23.64	23.74	15.88	13.52	68.33	1.67	-

TABLE I: **Object exploration.** Comparing exploration metrics across multiple environments and tasks. Dreamer is the only agent that aims to maximize the task's sparse rewards. All other approaches only perform unsupervised exploration. On the *right* sparse rewards obtained during the exploration phase. (3 seeds)

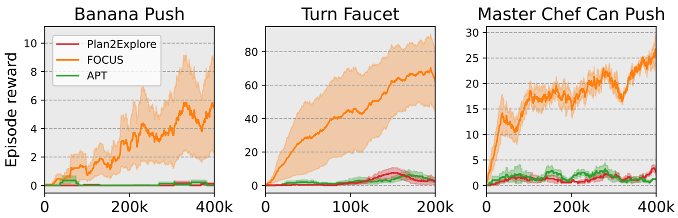


Fig. 3: **Finetuning.** Comparing adaptation performance of FOCUS and other exploration strategies in sparse-reward manipulation tasks. (3 seeds)

interactions consistently, only in the master chef can 'push' environment, but struggles elsewhere. APT and P2E tend to explore object interactions more consistently than DreamerV2, but generally move objects less than FOCUS and tend to find sparse rewards, i.e. achieving placement or task rewards, more rarely.

Model finetuning. After exploring the environment for 2M environment steps, we adapt the exploration approaches, allowing them an additional number of environment interactions. During this stage, if the exploration stage was fruitful, the agent may already know where the rewards can be found in the environment, despite not having actively attempted to maximize them.

In order to speed up the finetuning stage, task-driven actor and critic networks are pre-trained in imagination, during the exploration stage, using the reward predictor trained on the exploration stage rewards. These actor and critic networks are then fine-tuned during the adaptation stage. This setup is similar to the adaptation experiments presented in [44].

To provide an idea of how frequently the agents were able to find sparse rewards during the exploration stage, we provide aggregate metrics on the right of Table I. The adaptation curves, showing episode rewards over time, are presented in Figure 3. Results clearly show that FOCUS is the only method that makes significant progress across all tasks. As the figures in the Table show, this is mainly due to the fact that FOCUS was also the approach that more frequently found rewards during the exploration stage, making finetuning to downstream tasks easier.

Reconstructions. In order to qualitatively assess the capacity of FOCUS to better capture information about the objects, we compare reconstructions from FOCUS to the non-object centric world model of DreamerV2, on random trajectories, in Figure 4. We observe that, while DreamerV2 can reconstruct the robot and the background, it struggles to accurately reconstruct objects. The banana is reduced to a yellow spot, while the can is almost invisible against the background. FOCUS, instead, reconstructs the objects more accurately, by capturing the object information in the object latents. FOCUS reconstructions shown are the result of the merging of all object and background RGB images into a single one.

Real-world. FOCUS can also be deployed to real-world settings. The main issue with applying FOCUS in the real world comes from the absence of segmentation masks. However, thanks to recent progress in the study of large-scale vision models, the issue can be easily circumvented using a pre-trained segmentation model. For these experiments, we adopted the Segment Anything Model (SAM) [25], to recognize the bricks in the scene, after providing a single labeled sample [3]. In Figure 4, we show that our method effectively captures object information, though sometimes sacrificing information about the gripper. When compared to DreamerV2, it can be noticed how the decoding of objects is of higher precision.

IV. Conclusion

We presented FOCUS, an object-centric model-based agent that can learn manipulation tasks more efficiently and easily discovers interactions with objects. One major limitation of our method is reliance upon the segmentation information, used in the object decoder's loss, which is not easily available to a real robot. In order to overcome such limitations, we aim to investigate unsupervised strategies for scene decomposition [51, 30]. We also aim to extend the evaluation further, with more extensive real-world experimentation and benchmarking.

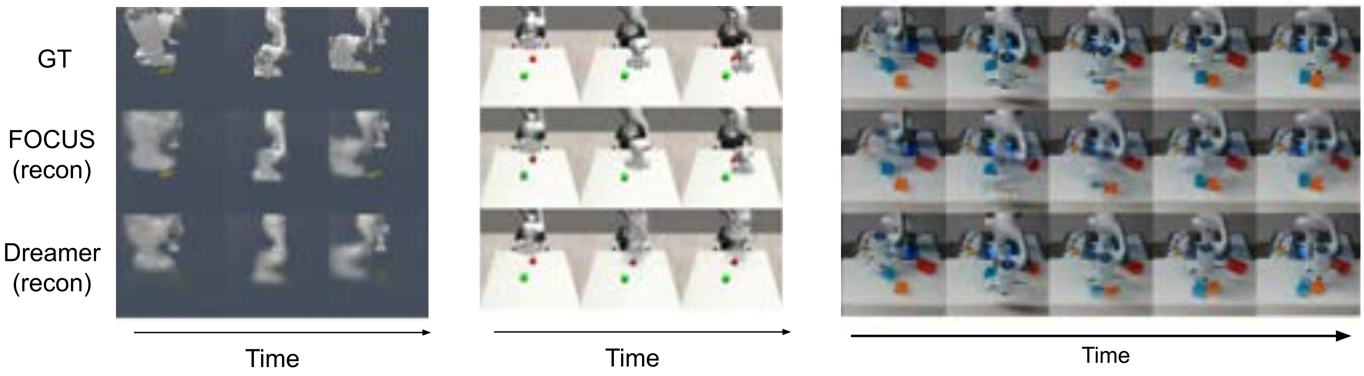


Fig. 4: **Reconstructions.** Comparing reconstructions between FOCUS's object-centric model and DreamerV2's world model. *(Left)* Maniskill banana environment. *(Middle)* Robosuite red and green cube environment. *(Right)* Real-world setup with 3 colored blocks.

Acknowledgments

This research received funding from the Flemish Government (AI Research Program). Pietro Mazzaglia is funded by a Ph.D. grant of the Flanders Research Foundation (FWO).

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016.

[2] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation, 2019.

[3] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023.

[4] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014. URL https://arxiv.org/abs/1409.1259.

[5] Jack Clark and Dario Amodei. Faulty reward functions in the wild. https://openai.com/blog/faulty-reward-functions/, 2016. Accessed: 2022-04-19.

[6] Toon Van de Maele, Tim Verbelen, Ozan Catal, and Bart Dhoedt. Disentangling what and where for 3d object-centric representations through active inference. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 701–714. Springer, 2021.

[7] Fei Deng, Ingook Jang, and Sungjin Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations, 2021.

[8] Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning, 2022.

[9] Carlos Diuk, Andre Cohen, and Michael L Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 240–247, 2008.

[10] Stefano Ferraro, Toon Van de Maele, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Computational optimization of image-based reinforcement learning for robotics. *Sensors*, 22(19):7382, 2022.

[11] Stefano Ferraro, Toon Van de Maele, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Disentangling shape and pose for object-centric deep active inference models. *arXiv preprint arXiv:2209.09097*, 2022.

[12] Stefano Ferraro, Toon Van de Maele, Tim Verbelen, and Bart Dhoedt. Symmetry and complexity in object-centric deep active inference models. *Interface Focus*, 13(3): 20220077, 2023.

[13] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods, 2018.

[14] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference, 2020.

[15] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills, 2023.

[16] David Ha and Jürgen Schmidhuber. World models. 2018. doi: 10.5281/ZENODO.1207631. URL https://zenodo.org/record/1207631.

[17] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

[18] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, pages 2555–2565, 2019.

[19] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *ICLR*, 2021.

[20] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[21] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control, 2022.

[22] Jeff Hawkins, Subutai Ahmad, and Yuwei Cui. A theory of how columns in the neocortex enable learning the structure of the world. *Frontiers in Neural Circuits*, 11, 2017. ISSN 1662-5110. doi: 10.3389/fncir.2017.00081. URL https://www.frontiersin.org/articles/10.3389/fncir.2017.00081.

[23] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *ArXiv*, abs/1806.10293, 2018.

[24] Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models, 2020.

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[26] Victoria Krakovna et al. Specification gaming: the flip side of ai ingenuity. https://www.deepmind.com/blog/specification-gaming-the-flip-side-of-ai-ingenuity, 2020. Accessed: 2022-04-19.

[27] Alex X. Lee, Coline Devin, Yuxiang Zhou, Thomas Lampe, Konstantinos Bousmalis, Jost Tobias Springenberg, Arunkumar Byravan, Abbas Abdolmaleki, Nimrod Gileadi, David Khosid, Claudio Fantacci, José Enrique

Chen, Akhil S. Raju, Rae Jeong, Michael Neunert, Antoine Laurens, Stefano Saliceti, Federico Casarini, Martin A. Riedmiller, Raia Hadsell, and Francesco Nori. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. *ArXiv*, abs/2110.06192, 2021.

[28] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 2016.

[29] Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18459–18473. Curran Associates, Inc., 2021.

[30] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention, 2020.

[31] Yao Lu, Karol Hausman, Yevgen Chebotar, Mengyuan Yan, Eric Jang, Alexander Herzog, Ted Xiao, Alex Irpan, Mohi Khansari, Dmitry Kalashnikov, and Sergey Levine. AW-opt: Learning robotic skills with imitation andreinforcement at scale. In *5th Annual Conference on Robot Learning (CoRL)*, 2021.

[32] Pietro Mazzaglia, Ozan Catal, Tim Verbelen, and Bart Dhoedt. Curiosity-driven exploration via latent bayesian surprise, 2022.

[33] Mirco Mutti, Lorenzo Pratissoli, and Marcello Restelli. Task-agnostic exploration via policy gradient of a nonparametric state entropy estimate, 2021.

[34] Akihiro Nakano, Masahiro Suzuki, and Yutaka Matsuo. Interaction-based disentanglement of entities for object-centric world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JQc2VowqCzz.

[35] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas A. Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei M. Zhang. Solving rubik's cube with a robot hand. *ArXiv*, abs/1910.07113, 2019.

[36] P. Oudeyer, F. Kaplan, and V. V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2): 265–286, 2007.

[37] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction, 2017.

[38] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement, 2019.

[39] Ivaylo Popov et al. Data-efficient deep reinforcement learning for dexterous manipulation, 2017.

[40] Sai Rajeswar, Cyril Ibrahim, Nitin Surya, Florian Golemo, David Vazquez, Aaron Courville, and Pedro O. Pinheiro. Touch-based curiosity for sparse-reward tasks, 2021.

[41] Sai Rajeswar, Pietro Mazzaglia, Tim Verbelen, Alexandre Piché, Bart Dhoedt, Aaron Courville, and Alexandre Lacoste. Mastering the unsupervised reinforcement learning benchmark from pixels. 2023.

[42] J. Schmidhuber. Curious model-building control systems. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, pages 1458–1463 vol.2, 1991.

[43] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, dec 2020. doi: 10.1038/s41586-020-03051-4. URL https://doi.org/10.1038%2Fs41586-020-03051-4.

[44] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *ICML*, 2020.

[45] Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration, 2021.

[46] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control, 2022.

[47] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.

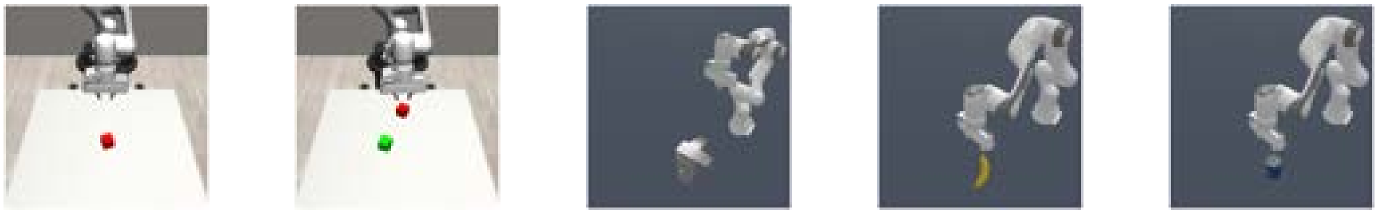
[48] Lauren Slone, Linda Smith, and Chen Yu. Self-generated variability in object images predicts vocabulary growth. *Developmental Science*, 22, 02 2019. doi: 10.1111/desc.12816.

[49] Linda B. Smith, Swapnaa Jayaraman, Elizabeth Clerkin, and Chen Yu. The developing infant creates a curriculum for statistical learning. *Trends in Cognitive Sciences*, 22(4):325–336, 2018. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2018.02.004. URL https://www.sciencedirect.com/science/article/pii/S1364661318300275.

[50] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning, 2022.

[51] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models, 2023.

[52] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.



(a) Red cube (b) RG cubes (c) Faucet (d) Banana (e) Master Chef Can

Fig. 5: **Simulation environments.** Visualization of the environments we applied FOCUS on. *(a-b)* from robosuite. *(c-e)* from ManiSkill2.

Appendix

A. Background material

Exploration. Solving sparse-reward tasks is a hard problem in RL. Inspired by artificial curiosity theories [42, 36], several works have designed exploration strategies for RL [37, 32, 40]. Other explorations strategies that have shown great success over time are based upon maximizing uncertainty [38, 44], or the entropy of the agent's state representation [29, 45, 33].

Object-centric. Decomposing scenes into objects can enable efficient reasoning over high-level building blocks and ensure the agent hones in on the most relevant concepts [8]. Several 2D object-centric representations, based on the principle of representing objects separately in the model, have been studied in the last years[30, 14, 2, 34, 6]. Inspired by the idea that such representations could help exploit the underlying structure of our control problem [9, 8], we propose a world model with an object-centric structured representation [24] that we show could strongly aid robotics manipulation settings.

B. Model details

DreamerV2 The architecture adopted for DreamerV2 is relevant to the one documented by [19]. Model states have both a deterministic and a stochastic component: the deterministic component is the 200-dimensional output of a GRU ([4], with a 200-dimensional hidden layer; the stochastic component consists of 32 categorical distributions with 32 classes each. States-based inputs such as the proprioception, the encoder, and the decoder are 4-layer MLP with a dimensionality of 400. For pixels-based inputs, the encoder and decoder follow the architecture of DreamerV2 [19], taking 64×64 RGBD images as inputs. Both encoder and decoder networks have a depth factor of 48. To ensure stable training during the initial phases, we adopt a technique from [20] where the weights of the output layer in the critic network are initialized to zero. This approach contributes to the stabilization of the training process especially in the early stages of training.

Networks are updated by sampling batches of 32 sequences of 32 timesteps, using Adam with learning rate $3\mathrm{e}^{-4}$ for the updates, and clipping gradients norm to 100.

FOCUS The architecture proposed is based on the implementation of DreamerV2 described above. The encoding network and the state-based decoding unit have the same structure mentioned in Dreamer. We introduced an object latent extractor unit consisting of a 3-layer MLP with a dimensionality of 512. The object-decoder network resembles the structure of the Dreamer's decoder, the depth factor for the CNN is set to 72. 64x64 RGBD images along with a ”segmentation weights” image are generated per each object.

Object-Centric Exploration The K-NN filter adopted for the entropy approximation consist of averaging the distance from the K-nearest neighbors (K = 30).

Objective In the objective of FOCUS we describe object-specific masks m_t^i. These masks are binary images, obtained from the entire scene segmentation mask. In practice, $m_t = \sum_i m_t^i \cdot \arg\max c^i$, which means the overall segmentation mask is obtained by summing the object-specific masks, multiplied by their object index, which can be extracted from c^i through an $\arg\max$, being c^i a one-hot vector.

C. Simulation environments

We opted for two simulation environments: ManiSkill2 [15] and robosuite [52]. For the robosuite environment, we considered a single (red) cube and a two-cube (red and green) setup. For Maniskill2, we opted for two single YCB objects setups (banana and master chef can), and a faucet setup (model 5007). Segmentation masks for training FOCUS' decoder are provided by the simulator.

In both environments, the robotic agent is a Franka arm with 7-DoF, controlled in cartesian end-effector space with fixed gripper orientation. Gripper state (open/close) adds a degree of freedom. For all the mentioned environments objects are spawned at a fixed pose. Visual observations are rendered at 64x64 resolution for all the environments.

D. Real-world environment

The setup for the real-world sperimentation consist of a Franka arm with 7-DoF, controlled in cartesian end-effector space with fixes gripper orientation and width. In the workspace in front of the robot, 3 colored (*blue, orange, red*) blocks are present. Pixel states of the environment are acquired through a Realsense D435. Both RGB and depth information are collected at 64x64 resolution. Observations are captured at 10Hz, for a total episode length of 500 frames.

E. Metrics and task rewards

To evaluate the quality of the exploration with the respect to the objects in the scene we considered the following metrics:

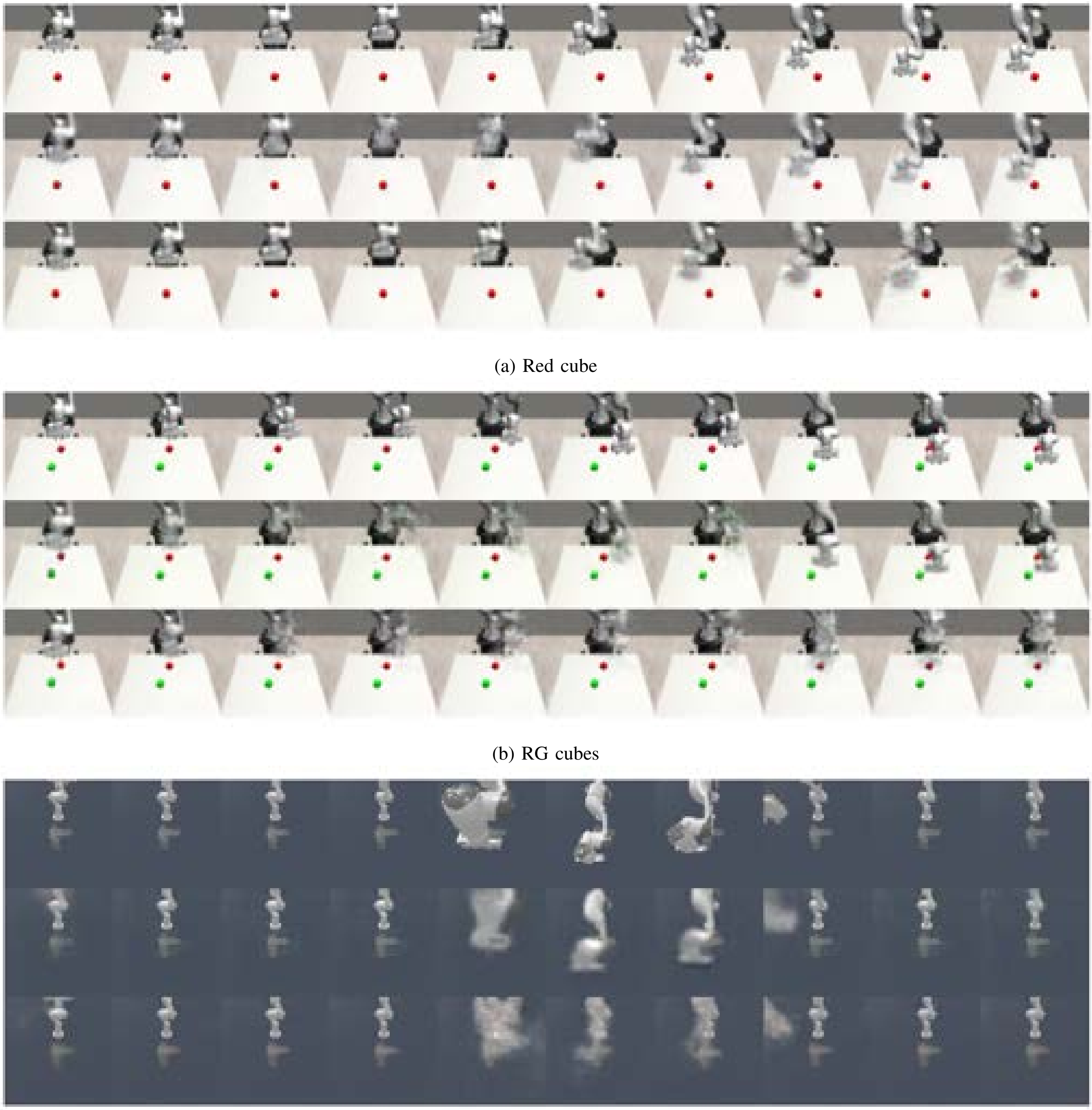
- *Contact (%):* average percentage of contact interactions between the gripper and the objects over an episode.
- *Positional displacement (m):* cumulative position displacement of all the objects over an entire episode.
- *Angular displacement (rad):* cumulative angular displacement of all the objects over an entire episode.
- *Up, far, close, left, right placement:* number of times the object is moved in the relative area of the working space.
- *Reward:* whether the agent was able to obtain sparse reward in the environment. The tasks analyzed are either “lift” (L) or “push” (P) tasks, for cube and YCB objects environments, and “turn” (T) for the faucet.

To evaluate the placement metric, we divide the workspace in front of the robot into 5 areas, with respect to the 5 cited directions: right, left, far, close, and up. For the Robosuite environment, with respect to the origin of the environment (coinciding with the center of the table) we consider a successful placement if the object is placed at a minimum of 0.25m up to a maximum of 0.4m along the directions considered. For the ”up” placement the minimum threshold is reduced to 0.05m.

For the Maniskill2 environment, in the same logic, we considered a minimum threshold of 0.4m up to a maximum of 0.5m. For the ”up” placement a minimum threshold is reduced to 0.1m.

The definition of the task rewards for the exploration tasks is linked to the placement definition given. Any reward is provided till the object has been placed in the desired area. For the lift tasks, before assigning any reward we check if the object is grasped. The definition of the task rewards for dense tasks is the one defined by the default implementation.

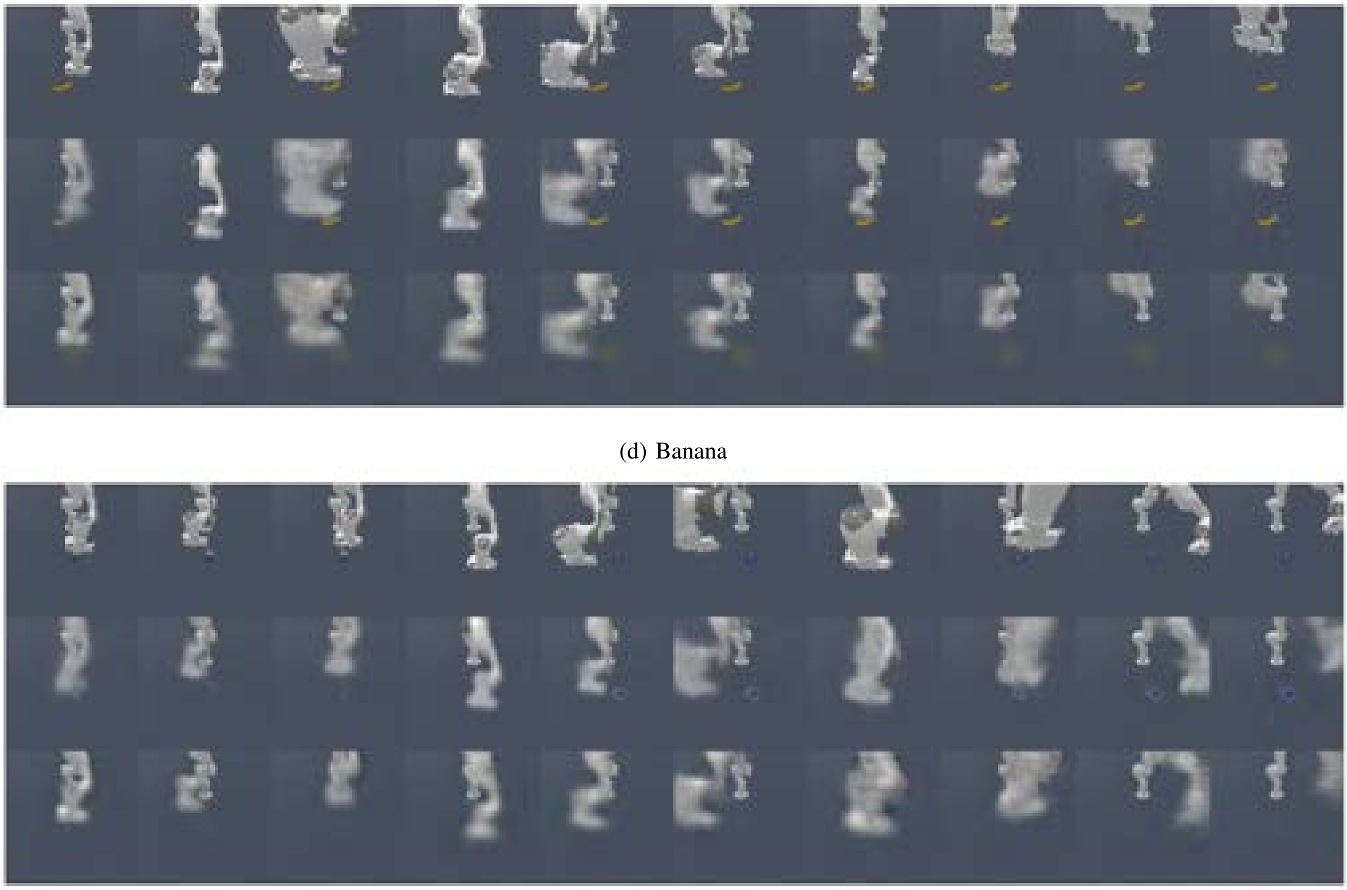
F. Extended results visualizations



(a) Red cube

(b) RG cubes

(c) Faucet



(d) Banana

(e) Master Chef Can

Fig. 6: Model reconstructions over random actions for considered environments. First row is ground truth. Second row is FOCUS reconstruction. Last row shows DreamerV2 reconstruction.

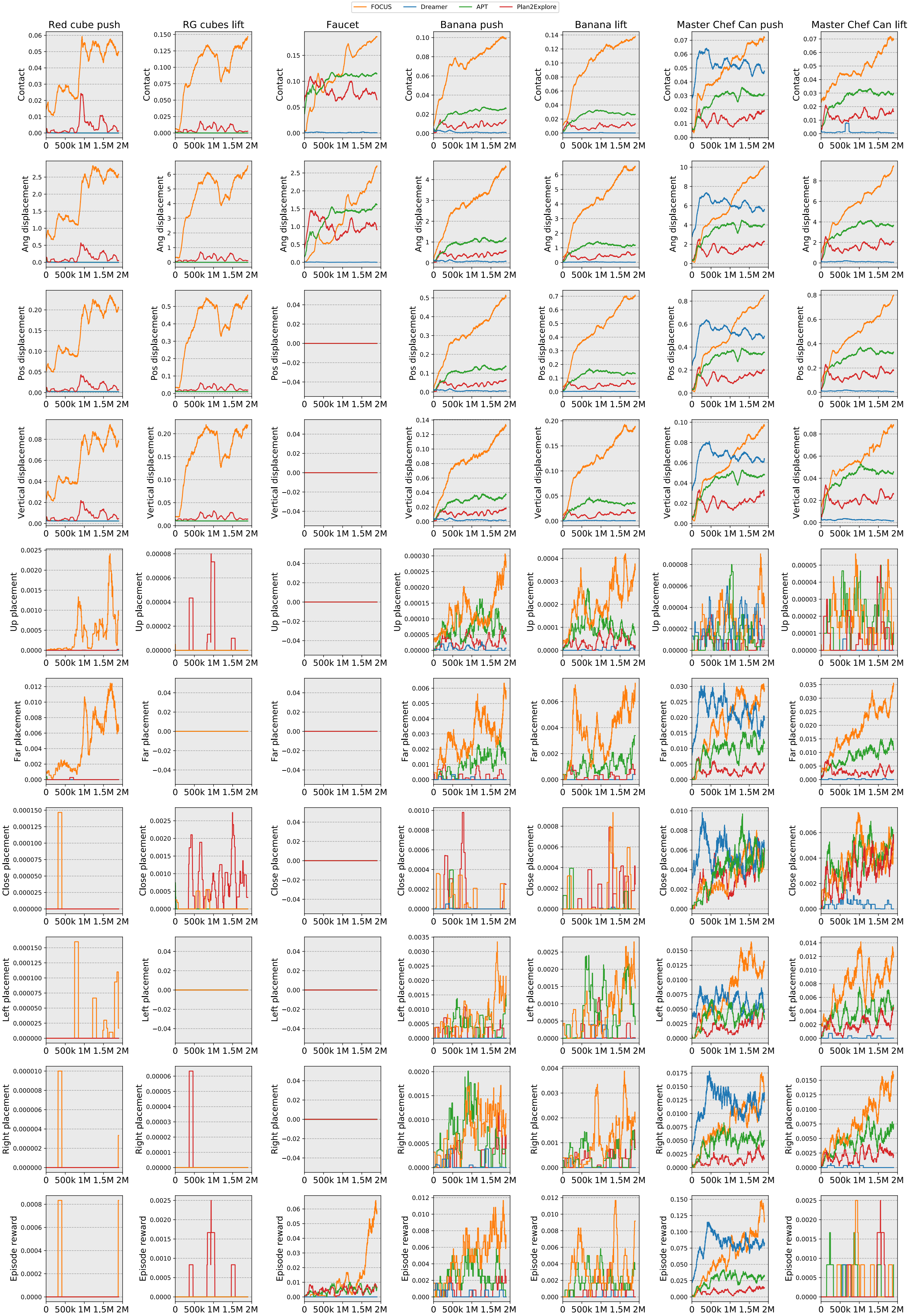


Fig. 7: Displaying exploration metrics over time for the object exploration experiments.