FOCUS: Object-Centric World Models for Robotic Manipulation

Stefano Ferraro **Pietro Mazzaglia** **Tim Verbelen** **Bart Dhoedt**

Ghent University
name.surname@ugent.be

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 a structured world model that allows controlling the agent accurately remains a challenge. To address this, we propose FOCUS, a model-based agent that learns an object-centric world model. The learned representation makes it possible to provide the agent with an object-centric exploration mechanism, which encourages the agent to interact with objects and discover useful interactions. We apply FOCUS in several robotic manipulation settings where we show how our method fosters interactions such as reaching, moving, and rotating the objects in the environment. We further show how this ability to autonomously interact with objects can be used to quickly solve a given task using reinforcement learning with sparse rewards. *Project website:*

1 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 [29, 36, 24, 32, 28, 11]. Among RL algorithms, model-based approaches aspire to provide greater data efficiency, compared to the model-free counterparts [14, 18]. Adopting world models [17, 20], 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 [20, 42, 21], including real-world applications, such as robotic manipulation and locomotion [51].

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 small objects [47], or they can waste most of the model capacity on rich, but potentially irrelevant features, such as static backgrounds [7]. In the case of robotic manipulation, this is problematic because the agent strongly needs to acquire information about the objects to manipulate in order to solve a given task.

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 [2, 6, 27, 40]. 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 such rewards in the environment and thus require appropriate exploration strategies, for which previous work has resorted to artificial curiosity mechanisms [37, 43] or entropy maximization strategies [34, 30].

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

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 [23, 13]. 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 [50, 49, 12].

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 to exploit such representation to explore object-oriented interactions.

Contributions Our contributions can be summarized as:

- an object-centric world model, which learns a latent dynamics of the environment where the information about objects is discriminated into distinct latent vectors;
- an object-centric exploration strategy, which encourages interactions with the objects, by maximizing the entropy of the latent object's representation;
- empirical evaluation of the approach, showing how object-centric exploration can foster interaction with the objects and consequent ability to solve robotic manipulation tasks, in several settings and tasks, across ManiSkill2 [16], robosuite [56] and Metaworld [54] environments;

2 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*. We first describe the overall structure and loss of the world model (left in Fig. 1) before delving into more details about the novel object-centric components of FOCUS (center in Fig. 1).

World model. Overall, the learned world model is composed of the following components:

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

which are trained end-to-end by minimizing the following loss:

$$\mathcal{L}_{\text{wm}} = \mathcal{L}_{\text{dyn}} + \mathcal{L}_{\text{proprio}} + \mathcal{L}_{\text{obj}}. \tag{2}$$

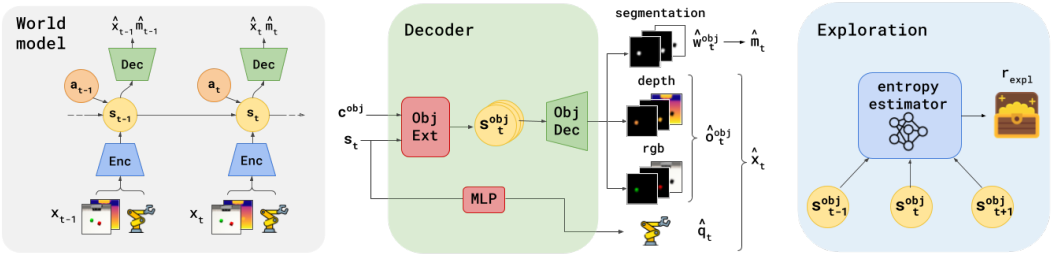

Figure 1: **FOCUS.** The agent learns a structured world model (left) that disentangles information in the environment by learning to reconstruct masked information about each observed object, thanks to an object-centric decoder (center). The learned object-centric state representation is used to incentivize object-centric exploration (right), maximizing the entropy of the object representation as a form of intrinsic reward.

For the dynamics component, i.e. prior and posterior, we adopt a recurrent state-space model (RSSM) [19], which extracts a latent state s_t made of a deterministic and a stochastic component. The dynamics minimizes the Kullback–Leibler (KL) divergence between posterior and prior:

$$\mathcal{L}_{\text{dyn}} = D_{\text{KL}}[p_\phi(s_{t+1}|s_t, a_t, e_{t+1})||p_\phi(s_{t+1}|s_t, a_t)]. \tag{3}$$

Proprioceptive information $\hat{q}_t$ is decoded out of the latent state s_t, using an MLP. The proprioceptive decoder learns to reconstruct proprio states, by minimizing a negative log-likelihood (NLL) loss:

$$\mathcal{L}_{\text{proprio}} = -\log p_\theta(\hat{q}_t|s_t) \tag{4}$$

Object-centric modules. The latent state of the world model tends to compress all the information from the environment in a unique latent structure. Our intention in FOCUS is to disentangle such information into separate latent structures, learning an object-centric world model.

For each object in the scene, the *object latent extractor* receives the model latent state s_t and a (one-hot) vector identifying the object c^{obj}, and extracts an object-centric latent s_t^{obj}. Given such an object latent, the *object decoder* reconstructs object-related observation information by outputting two kinds of information: one-dimensional "object weights" w_t^{obj}, which are used to build a segmentation mask of the scene, and object-specific observation $\hat{o}_t^{obj}$, where the information that is irrelevant to the object is masked out through the segmentation. Details about the segmentation loss function are provided in the appendix.

Object-centric Exploration. State maximum entropy approaches for RL [34, 46, 30] 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 well-directed exploration towards object interactions and the discovery of novel object views, by having the agent maximize the entropy over the object latent state representation.

In order to estimate the entropy value over batches, we apply a K-NN particle-based estimator [48] 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:

$$r_{\text{expl}} = \sum_{obj=0}^{N} r_{\text{expl}}^{obj} \qquad \text{where} \quad r_{\text{expl}}^{obj}(s) \propto \sum_{i=1}^{K} \log \left\| s^{obj} - s_i^{obj} \right\|_2 \tag{5}$$

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

Crucially, as we learn a (object-centric) world model we can use it to optimize actions by learning actor and critic in imagination [20], so that the latent states in Equation 5 are states of imaginary trajectories, generated by the world model by following the actor's predicted actions.

3 Experiments

We argue that FOCUS object-centric world model and exploration strategy can be used to improve control in robotic manipulation, where interactions with objects are essential. The experiments

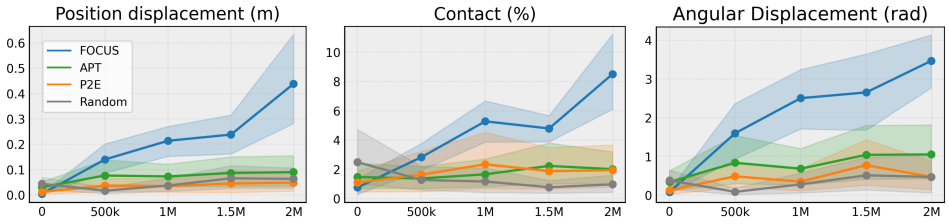

Figure 2: **Exploration performance.** Comparing exploration metrics across 10 tasks from ManiSkill2, robosuite and Metaworld. Experiments are run with 3+ seeds per task and aggregated in a statistically sound way using RLiable [1].

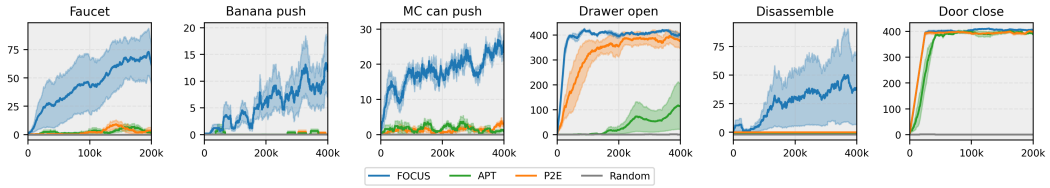

Figure 3: **Task fine-tuning performance.** Comparing fine-tuning performance across tasks from ManiSkill2 (Faucet, Banana, Master-Chef Can) and Metaworld (Drawer, Disassemble, Door). Experiments are run with 3+ seeds.

aim to empirically validate our argument by evaluating (i) the exploration performance of FOCUS compared to the state-of-the-art in world models and exploration, (ii) performance on sparse reward manipulation tasks, after an exploration stage.

We adopt 10 tasks from three robotic manipulation benchmarks: ManiSkill2 [16], robosuite [56] and Metaworld [54] (see Appendix). Both ManiSkill and robosuite provide segmentation masks as an (optional) input for the agent, while Metaworld does not. Thus, we adopted fastSAM (refer to appendix for details) to extract segmentation masks in those tasks, an evaluation setting that serves us the purpose of a test field for real-world experiments.

Exploration. To evaluate the performance of the strategy of FOCUS, we chose contact with object, and both position and angular displacement of the object. Details in appendix.

In Figure 2, we compare FOCUS against three exploration strategies: Plan2Explore (P2E) [45], Active Pre-training (APT) [30] and Random actions. [1]

As shown by all metrics, FOCUS interacts with objects much more assiduously than the other approaches, with the exploration performance consistently increasing over time. APT and P2E perform similarly and they only slightly perform better than Random, showing the importance of focussing on the objects when exploring a robotic manipulation environment.

Sparse reward tasks fine-tuning. As the agent explores the environment, it may encounter important information that may be a source of (sparse) reward, such as opening a drawer or closing a door handle. In order to exploit such information, while we keep exploring, we concurrently train an exploitative *task* actor-critic, which can be used for solving tasks after exploring the environment, in a zero-shot or few-shot fashion. The task actor-critic is defined as follows:

$$\text{Task actor:} \quad \pi_{\text{task}}(a_t|s_t), \qquad \text{Task critic:} \quad v_{\text{task}}(s_t). \tag{6}$$

and thanks to the world model, these can be learned in imagination, while the agent keeps exploring the environment [45]. After exploring the environment for 2M environment steps, we adapt the task actor-critic, allowing an additional (smaller) number of environment interactions for fine-tuning the agent and perfecting the task. The adaptation curves, showing episode rewards over time, are presented in Figure 3.

4 Conclusion

We presented FOCUS, an object-centric model-based agent that eagerly discovers interactions with objects, enabling one to learn manipulation tasks more efficiently. We extensively evaluated and compared our approach to state-of-the-art baselines, showcasing the opportunities that our method unveils, especially for learning how to interact with objects and how to solve tasks from sparse rewards. One major limitation of our method is reliance upon the segmentation information, used in the object decoder's loss. While this works well in controlled environments, we found that in some settings, inaccurate masks can lead to lower-quality representations. In order to overcome such limitations, we aim to investigate unsupervised strategies for scene decomposition [52, 31], which we could use to refine the masks that are provided by the pretrained segmentation model. We also aim to extend our work to disentangle object features at a finer level, e.g. position, shape, different parts, and investigate how these could be used to further facilitate robotic manipulation control.

[1] For fairness with P2E and FOCUS, both APT and Random are implemented on top of a DreamerV2 world-model-based agent, following [42].

5 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). We thank ServiceNow Research for the computational resources provided during this work.

References

[1] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, and M. G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 2021.

[2] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in ai safety, 2016.

[3] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros. Large-scale study of curiosity-driven learning, 2018.

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

[5] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014. URL `https://arxiv.org/abs/1409.1259`.

[6] J. Clark and D. Amodei. Faulty reward functions in the wild. `https://openai.com/blog/faulty-reward-functions/`, 2016. Accessed: 2022-04-19.

[7] F. Deng, I. Jang, and S. Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations, 2021.

[8] A. Dittadi, S. Papa, M. D. Vita, B. Schölkopf, O. Winther, and F. Locatello. Generalization and robustness implications in object-centric learning, 2022.

[9] C. Diuk, A. Cohen, and M. 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] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[11] S. Ferraro, T. Van de Maele, P. Mazzaglia, T. Verbelen, and B. Dhoedt. Computational optimization of image-based reinforcement learning for robotics. *Sensors*, 22(19):7382, 2022.

[12] S. Ferraro, T. Van de Maele, P. Mazzaglia, T. Verbelen, and B. Dhoedt. Disentangling shape and pose for object-centric deep active inference models. *arXiv preprint arXiv:2209.09097*, 2022.

[13] S. Ferraro, T. Van de Maele, T. Verbelen, and B. Dhoedt. Symmetry and complexity in object-centric deep active inference models. *Interface Focus*, 13(3):20220077, 2023.

[14] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods, 2018.

[15] 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, 2020.

[16] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su. Maniskill2: A unified benchmark for generalizable manipulation skills, 2023.

[17] D. Ha and J. Schmidhuber. World models. 2018. doi: 10.5281/ZENODO.1207631. URL `https://zenodo.org/record/1207631`.

[18] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

[19] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *ICML*, pages 2555–2565, 2019.

[20] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. In *ICLR*, 2021.

[21] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[22] N. Hansen, X. Wang, and H. Su. Temporal difference learning for model predictive control, 2022.

[23] J. Hawkins, S. Ahmad, and Y. 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.

[24] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *ArXiv*, abs/1806.10293, 2018.

[25] T. Kipf, E. van der Pol, and M. Welling. Contrastive learning of structured world models, 2020.

[26] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything, 2023.

[27] V. 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.

[28] A. X. Lee, C. Devin, Y. Zhou, T. Lampe, K. Bousmalis, J. T. Springenberg, A. Byravan, A. Abdolmaleki, N. Gileadi, D. Khosid, C. Fantacci, J. E. Chen, A. S. Raju, R. Jeong, M. Neunert, A. Laurens, S. Saliceti, F. Casarini, M. A. Riedmiller, R. Hadsell, and F. Nori. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. *ArXiv*, abs/2110.06192, 2021.

[29] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 2016.

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

[31] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention, 2020.

[32] Y. Lu, K. Hausman, Y. Chebotar, M. Yan, E. Jang, A. Herzog, T. Xiao, A. Irpan, M. Khansari, D. Kalashnikov, and S. Levine. AW-opt: Learning robotic skills with imitation andreinforcement at scale. In *5th Annual Conference on Robot Learning (CoRL)*, 2021.

[33] P. Mazzaglia, O. Catal, T. Verbelen, and B. Dhoedt. Curiosity-driven exploration via latent bayesian surprise, 2022.

[34] M. Mutti, L. Pratissoli, and M. Restelli. Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate, 2021.

[35] A. Nakano, M. Suzuki, and Y. 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.

[36] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. A. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. M. Zhang. Solving rubik's cube with a robot hand. *ArXiv*, abs/1910.07113, 2019.

[37] 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.

[38] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction, 2017.

[39] D. Pathak, D. Gandhi, and A. Gupta. Self-supervised exploration via disagreement, 2019.

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

[41] S. Rajeswar, C. Ibrahim, N. Surya, F. Golemo, D. Vazquez, A. Courville, and P. O. Pinheiro. Touch-based curiosity for sparse-reward tasks, 2021.

[42] S. Rajeswar, P. Mazzaglia, T. Verbelen, A. Piché, B. Dhoedt, A. Courville, and A. Lacoste. Mastering the unsupervised reinforcement learning benchmark from pixels. 2023.

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

[44] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. 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.

[45] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models. In *ICML*, 2020.

[46] Y. Seo, L. Chen, J. Shin, H. Lee, P. Abbeel, and K. Lee. State entropy maximization with random encoders for efficient exploration, 2021.

[47] Y. Seo, D. Hafner, H. Liu, F. Liu, S. James, K. Lee, and P. Abbeel. Masked world models for visual control, 2022.

[48] H. Singh, N. Misra, V. Hnizdo, A. Fedorowicz, and E. Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.

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

[50] L. B. Smith, S. Jayaraman, E. Clerkin, and C. 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.

[51] P. Wu, A. Escontrela, D. Hafner, K. Goldberg, and P. Abbeel. Daydreamer: World models for physical robot learning, 2022.

[52] Z. Wu, N. Dvornik, K. Greff, T. Kipf, and A. Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models, 2023.

[53] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng. Track anything: Segment anything meets videos, 2023.

[54] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019. URL https://arxiv.org/abs/1910.10897.

[55] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang. Fast segment anything. 2023.

[56] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

Appendix

Background

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 $q(x, a)$ [18, 14]. World models [17] 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 [20, 21] or to actively plan at each action [44, 22, 42], which can lead to higher data efficiency in solving the task.

Exploration. Solving sparse-reward tasks is a hard problem in RL because of the difficulty of exploring the environment and identifying rewarding states. Inspired by artificial curiosity theories [43, 37], several works have designed exploration strategies for RL [38, 33, 41]. Other exploration strategies that have shown great success are based upon the ideas of maximizing uncertainty [39, 45], or the entropy of the agent's state representation [30, 46, 34]. One issue with exploration in visual environments is that these approaches can be particularly attracted by easy-to-reach states that strongly change the visual appearance of the environment [3]. For robotic manipulation this can cause undesirable behaviors, e.g. a robot arm exploring different poses in the proximity of the camera but ignoring interactions with the objects in the workspace [42].

Object-centric representations. 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 recently analyzed [31, 15, 4, 35]. 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 [25] that we show could strongly aid robotic manipulation settings.

Methods details

How is the segmentation mask learned? The object decoder outputs one-dimensional "object weights" w_t^{obj}, which represent object-specific per-pixel logits. These logits are aggregated in a scene by applying a softmax among all object weights. The overall segmentation mask is obtained as:

$$\hat{m_t} = \text{softmax}(w_t^1, ..., w_t^N) \tag{7}$$

with N being the object instances. Object-specific masks can be obtained by taking the corresponding object's channel mask in the segmentation. Defining object-specific masks as m_t^{obj}, we can multiply the observation by these masks, to obtain object-specific observations $\hat{o_t^{obj}}$ that focus only on the obj-th object information. [2]

The object decoder loss is defined as follows:

$$\mathcal{L}_{\text{obj}} = -\log \underbrace{p(\hat{m_t})}_{\text{mask}} - \log \sum_{\text{obj}=0}^{N} \underbrace{m_t^{\text{obj}} p_\theta(\hat{x_t^{\text{obj}}}|s_t^{\text{obj}})}_{\text{masked reconstruction}} \tag{8}$$

By minimizing the NLL of the masked reconstruction term, 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 to occupy their correct space in the scene (in pixel space), through the $mask$ loss.

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

How are the segmentation mask targets for the mask loss obtained? In order to discriminate object information into different latent vectors, the object-centric components leverage an object discrimination process that entails learning to segment the scene observations. While obtaining segmentation masks in some simulation environments is easy, thanks to the available ground truth knowledge of the simulator.

The increasing availability of large pre-trained models for segmentation offers an opportunity to remedy the problem. In our experiments, when the segmentation information is not available, we chose to adopt an implementation of the Segment Anything Model (fastSAM; [26, 55]). At the beginning of each episode, per object segmentation instances are generated with fastSAM. Prompt modalities for the selection of objects of interest are either text prompts or box prompts. Modalities are chosen according to the consistency of the segmentation obtained. For subsequent frames, segmentation maps are produced by a tracking model, for which we ground on the XMem model [53].

DreamerV2 The architecture adopted for DreamerV2 is relevant to the one documented by [20]. Model states have both a deterministic and a stochastic component: the deterministic component is the 200-dimensional output of a GRU ([5], 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 [20], 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 [21] 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.

We adopt the same set of hyperparameters across all methods, including MWM in the dense reward tasks.

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.

Experiments details

Tasks The tasks we adopted are part of three robotic manipulation benchmarks: ManiSkill2 (MS), Metaworld (MW) and robosuite (RS). Following the scheme in Figure 4, the 10 tasks adopted are: (a) Red cube (RS), (b) RG cubes (RS), (c) Faucet (MS), (d) Banana (MS), (e) Master Chef Can (MS), (f) Door Open (MW), (g) Door Close (MW), (h) Disassemble (MW), (i) Drawer Open (MW), (j) Peg Insert (MW).

Exploration 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.

Dense reward experiments. FOCUS is made of two main novel components: the object-centric world model and the object-centric exploration. After showing that the exploration originating from the object-centric representation is beneficial, it is interesting to see how the object-centric world model impacts performance for general RL settings.

In Figure 5, we compare the final normalized performance (in terms of episode rewards) between FOCUS and the world-model-based agents DreamerV2 [20] and MWM [47] across 6 dense-reward

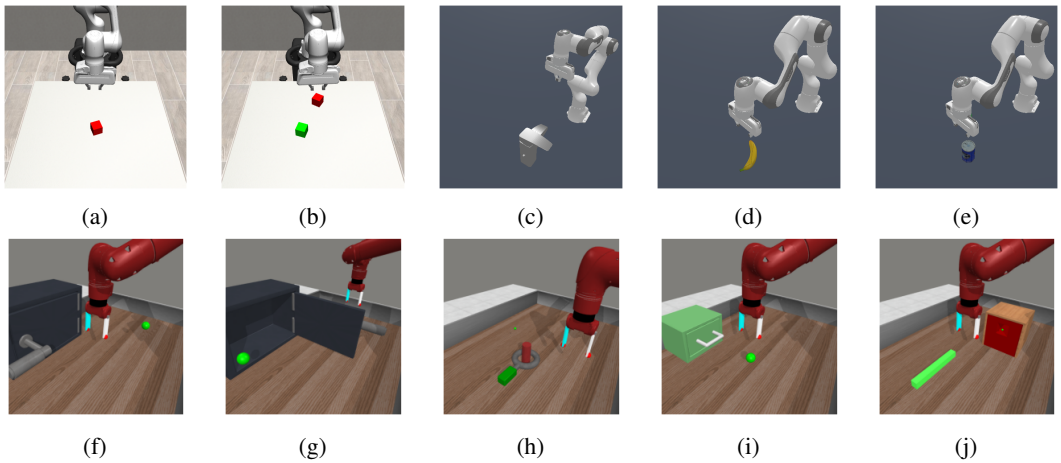

Figure 4: **Simulation environments.** Visualization of the simulation environments that have been used for our experiments.

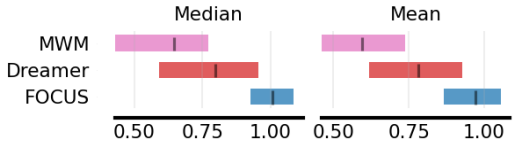

Figure 5: **Dense reward performance** Results of dense reward experiments across 6 tasks. Experiments are run for 2M steps with 3 seeds per task and aggregated using RLiable [1].

tasks: Drawer Open, Door Open, Door Close, Lift Cube, Stack Cube, and Turn Faucet. The only difference between the methods is the world model, so the study compares the quality of the model to learn control policies.

We observe that FOCUS obtains the highest median and mean performance, with DreamerV2 following and MWM lagging behind. This shows that object-centric world models can be beneficial also when working with dense rewards. While MWM should also perform similarly or better than Dreamer in these environments, we observed that MWM's model requires a higher frequency of updates to perform better, potentially due to the use of visual transformers [10].