Learning world models by self-supervised exploration

1st Stefan Frisch
Technical University of Munich
Munich, Germany
stefan.frisch@tum.de

2nd Florian Stilz
Technical University of Munich
Munich, Germany
florian.stilz@tum.de

***Abstract*—This project introduces an adapted version of Plan2Explore, where an agent builds a world model trained in a self-supervised manner. We introduce this framework to a new task Stacker from the DeepMind Control Suite. In addition to visual observations, we introduce proprioceptive information to Plan2Explore to enhance the agent's performance. In particular, we added contact sensors information from the fingers to the input of the agent such that it can fulfill subtasks like grabbing and stacking. These subtasks are exploration rewards that should steer the agent towards a more targeted exploration of the environment and improve the performance for the desired stacker task. In addition, we introduce a simplified version of the stacker task called Push2Target, where only the x-axis is considered.**

I. Introduction

The prevailing approach to sensorimotor control involves training an agent on specific tasks using rewards in reinforcement learning or demonstrations in imitation learning. However, this method often proves to be inefficient as each task requires a significant amount of task-specific environment interaction to solve from scratch. As a result, a crucial question arises: How can an agent efficiently generalize to unseen tasks in a zero or few-shot manner? This idea was explored by Hafner et al. in Plan2Explore [11].

One of the tasks not considered in the original Plan2Explore [11] is the stacker 2 task from the DeepMind Control Suite [12]. Here, the agent has to move and stack 2 boxes until they overlap with a given target. The task can be observed in figure 1. It should be noted that the image depicts the setting with 4 boxes, compared to the setting with 2 boxes that we focused on in the following report. Nevertheless, the results are very similar for the 2 box and 4 box setting. The major problem that can be observed when training Plan2Explore and DreamerV2 on it is the lack of interactions between the agent and physical objects i.e. boxes. This lack of interaction makes it impossible to form an accurate world model and solve the stacker task. Additionally, the reward is extremely sparse for this task, which makes it even for agents that get the full reward difficult to tackle the problem adequately.

We built upon the idea of Hafner et al. and propose several solutions to the problems of the stacker task that can be observed for vanilla Plan2Explore. Our contributions are the following:

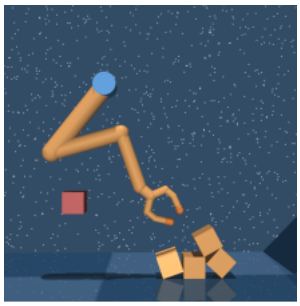

Fig. 1. Stacker 4 task illustration [12]. In the following report, we only focused the setting with 2 boxes.

- Proprioceptive information: we added sensory information for the fingers to the model input.
- Intrinsic motivation: by using some environment specific clues we introduce a much richer exploration signal that should help the agent to explore the environment more efficiently and to built a better world model. For that we introduce the biologically inspired Learn2Grab and Stacking reward. This also has the goal of increasing the agent to box interactions by a great margin.
- Push2Target: to further simplify the task, we introduce an additional dense reward that only tracks the x-coordinate of the target. This way we provide a much richer reward to the agent ideally helping it to correctly identify how to move boxes in the direction of the target. Additionally, it strongly reduces the issue of sparsity within the reward function.

II. Related Work

In unsupervised skill discovery, a model has to use intrinsic rewards to generate new knowledge since no external rewards are available. Most early works focus on the curiosity of an agent [1], [9] and do not build a world model. In more recent projects the notion of model disagreement [2], [10] is found. In this method, an ensemble of models is trained and the disagreement between these models is used to identify the most interesting actions for the agent to pursue. Plan2Explore [11] for instance measures the variance between the predictions of the different models in the latent space. Model-based RL historically brought higher data efficiency over model-

free agents while only working with low dimensional input [4]. Recently, several methods have been proposed which build world models in latent space and thus being able to operate on high dimensional inputs such as images [3], [5], [7]. Plan2Explore is built on top of Dreamer [6] which is a competitive example of a method that builds world models in latent state using images as observations.

III. Method

Our architecture is inspired by Plan2Explore [11]. A more targeted exploration can be achieved by incorporating certain environmental information into the exploration process. In the case of the stacker, we have to encourage a more targeted exploration of hand-box interaction.

A. Learn2Grab

The Learn2Grab approach is a biological-inspired exploration reward signal for efficiently exploring an environment where the unknown final goal is to stack objects. The primary idea of Learn2Grab is heavily influenced by how babies learn. There is a stage in a human baby's life where it tries to grab everything around it, such as the fingers of adults and toys. The Learn2Grab approach involves subgoals that the agent can learn efficiently:

- Hand Close to a Box: The first subgoal is to move the agent's hand close to the box.
- Grab the Box: The agent receives a reward for successfully having contact between the two fingers and one box
- Lift the Box: The agent receives a reward for lifting the box to a certain height threshold.

The agent is trained to switch between these subtasks based on the previous rewards it has received.

B. Stacking

The stacking phase constitutes the second targeted exploration phase, aimed at enhancing the agent's comprehension of box-to-box interactions. Following the successful acquisition of box-grasping skills, stacking represents the next logical step for the agent to master. This phase is designed to reward the agent for correctly placing a box on top of another box, either by deliberately placing it or by allowing it to fall from a higher altitude. The former is considered more challenging and thus more incentivized.

C. Proprioceptive Information

We enhanced the agent's input by incorporating sensory information from the fingers. This approach is preferable to relying solely on visual input, as it is challenging to determine from images whether a finger is merely close to or in contact with the box. Additionally, because Mujoco's physics simulation accurately models the boxes' weight, the agent must exert a specific amount of force to lift them off the ground. In the ablation study IV-A we explore in detail what effect the sensory information has on the grabbing subtask.

D. Push2Target

We also experimented with decreasing the complexity of the task by introducing a Push2Target reward. The primary motivation is the sparsity of the reward for the main task. The agent only receives a reward once a box overlaps with the target, and the agent does not touch the box simultaneously. Therefore, a richer reward function is chosen that considers only the x-axis of the boxes and the target in order to reduce the complexity by only considering the 1D case. Furthermore, the reward function uses a distance field, which measures the distance between the box and the target and, based on that, provides a continuous reward to the agent.

IV. Experiments

A. Ablation Study

Figure 2 displays the Learn2Grab exploration reward for the agent using various feature inputs. As previously mentioned, relying solely on visual inputs is insufficient for learning object grasping and lifting. This is due to limitations in image resolution, as well as a lack of information regarding the amount of force being applied to the object. Interestingly, we observed that even with access to the full set of proprioceptive information, including the agent's position, velocity, and touch information, the agent struggled to master the Learn2Grab reward. We hypothesize that this is due to the superior generalization capabilities of the weight sharing CNN used to process image information, as compared to the MLP used for other input forms. These findings underscore the importance of selecting appropriate input modalities.

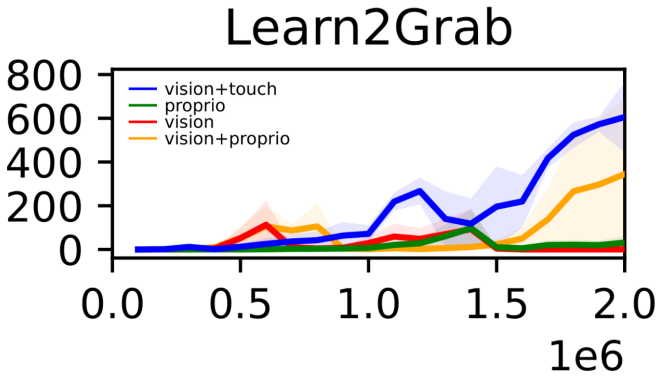

Fig. 2. Comparison of the Learn2Grab exploration reward with different feature inputs for the agent. Proprioceptive information refers to agent's and boxes' position, velocity, and touch information.

B. Environment Interactions

To address the issue of limited interactions between the agent and the boxes in the stacker task, targeted exploration was introduced to emphasize hand-to-box interactions. Figure 3 shows that the Learn2Grab approach has significantly more interactions with the boxes than both the vanilla self-supervised Plan2Explore [11] and the DreamerV2 [8] agent, where the latter is the same agent but trained with constant task

reward. This finding is particularly significant, as it highlights the effectiveness of targeted exploration in enabling the agent to interact more effectively with its environment, thereby overcoming the challenge of rare box-to-hand interactions.

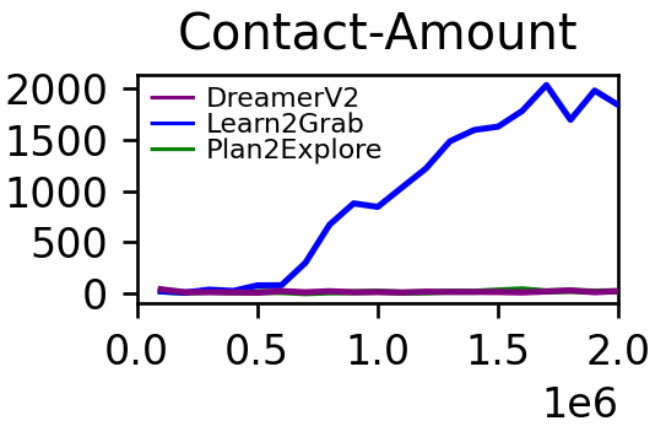

Fig. 3. Comparison of the box-to-hand contacts for baseline models and Plan2Explore trained with the Learn2Grab exploration reward.

C. Final Task Experiments

Figure 4 compares the performance of all pretrained agents from targeted exploration tasks, as well as a simplified version utilizing Push2Target, with the baseline models Plan2Explore and DreamerV2. Notably, none of the models were able to achieve an average reward value above 200, indicating insufficient learning of the task. The peaks and valleys in the figure correspond to the random start positions of the target and boxes, which can either facilitate or hinder task completion, leading to fluctuations in reward. Rarely, the agents were able to solve the task through their own actions. The performance curves of the pretrained agents for Learn2Grab and Stacking were indistinguishable, and for brevity, only Learn2Grab is shown in the figure.

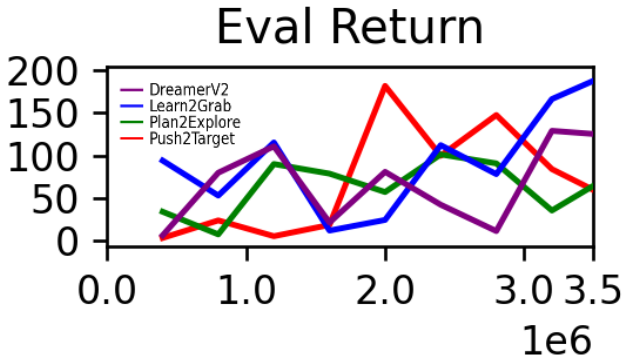

Fig. 4. Comparison of the stacker 2 evaluation reward of baseline models, Learn2Grab (pretrain Plan2Explore with the Learn2Grab exploration reward for 2M step for building the world model and the rest of the steps with DreamerV2), and Push2Target (DreamerV2 but with our simplified reward).

V. Discussion

The reason for the lack of success in solving the stacker task seems to be the lack of representational power of the world model. It struggles to correctly model the movement and location of boxes after interaction with the agent. This makes it nearly impossible to complete the task of moving an object from one position to another. A possible way of fixing this problem is to create an additional latent variable which is specifically designed to model the box locations and thus steers the model to put a stronger emphasis on the correct box locations.

VI. Conclusion

The stacker task poses a formidable challenge for reinforcement learning agents, even when provided with the complete reward signal. Two key challenges were identified: a lack of interactions between the agent and boxes, and sparse rewards. To address these challenges, two solutions were proposed: targeted exploration that emphasizes desired box-to-hand and box-to-box interactions, and a simplified task with a dense reward function using Push2Target. Combining proprioceptive information with vision improved the agent's ability to learn how to grab boxes. However, despite these efforts, the stacker task remained unsolved due to the limited representational capacity of the agent to accurately model box-to-agent interactions.

References

[1] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning, 2018.

[2] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018.

[3] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models, 2018.

[4] Marc Peter Deisenroth and Carl Edward Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 465–472, Madison, WI, USA, 2011. Omnipress.

[5] David Ha and Jürgen Schmidhuber. World models. 2018.

[6] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2019.

[7] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels, 2018.

[8] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

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

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

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

[12] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018.