Explorations in Dynamics Decomposition for Multi-Task RL

Varun Giridhar
Georgia Institute of Technology

Ignat Georgiev
Georgia Institute of Technology

Nicklas Hansen
University of California, San Diego

Animesh Garg
Georgia Institute of Technology

Abstract

Reinforcement Learning (RL) has demonstrated remarkable success in solving single-task problems but continues to face challenges in generalizing to multi-task settings involving diverse tasks and embodiments. While world models provide a scalable approach by learning to simulate environment dynamics, they remain limited in handling Out-of-Distribution (OOD) data effectively. To address this limitation, we propose a novel approach: decomposing world model dynamics into independent "subdynamics." in preliminary tests, this method achieves significant gains in the state-of-the-art in dynamics modeling across a range of control tasks from the DeepMind Control Suite. Furthermore, we extend our exploration to video data by leveraging the attention mechanism and hypothesize scaling this method to the multi-task setting.

1. Introduction

World models have become a popular paradigm in the world of reinforcement learning: world models can be used to generate imaginary data where we can then train a policy on this data. Another line of work focuses on the differentiability of world models, where loss can be propagated through the world model [2]. We can also combine optimal control algorithms (e.g., model predictive control) with learned world models to solve complex tasks [5].

The World Models Framework helps algorithms generalize to out-of-distribution samples [1]. However, in the current stage of robotics, we lack the quality and quantity of data required to generalize between different tasks and different robot embodiments. Concisely, the World Models Framework suffers from two important issues:

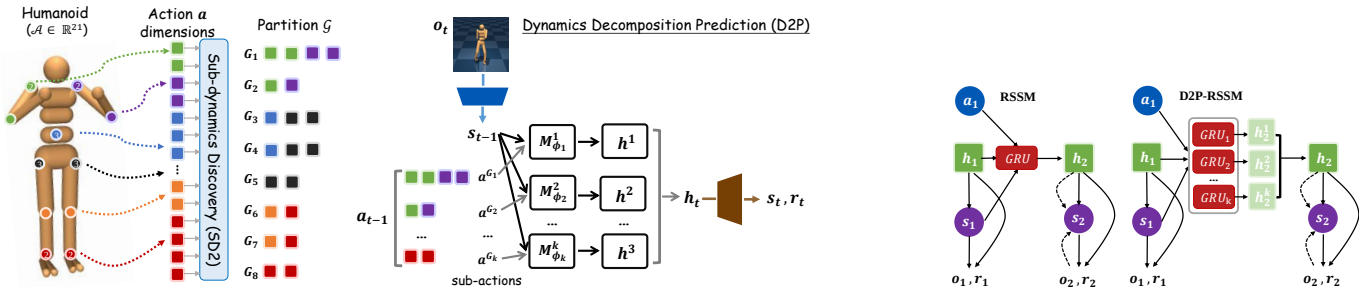
1. Inability to generalize to **out-of-distribution** samples
2. Intractable optimization landscapes with contact-rich and chaotic dynamics tasks [2]

Many of the current works that train a single well-designed network to fit the entire environment dynamics suffer from the aforementioned issues. In light of this, we propose an alternative pathway to scale to the multi-task setting: decomposing entire world model dynamics into subdynamics. Anecdotally, this approach can be compared to breaking down a complex dynamic system into several independent subdynamic components, similar to how mechanical systems or control theory often isolate subsystems to simplify modeling. We see this direction as valuable to both the reinfocement learning and generative modeling community since advances will help solve World Model's primary bottlenecks (1), (2).

2. Related Work

2.1. World Models

HIEROS [10] learns time-abstracted world representations and imagines trajectories at multiple time scales in latent space. Using the framework of multiple world models (functioning at different time scales), HIEROS outperforms SOTA on the Atari 100k benchmark and shows that their proposed world model is able to predict complex dynamics very accurately - a hint of the benefits of using a composition of world models' latent representations. RoboDreamer [14] breaks down language prompts into a set of lower-level language primitives and conditions a diffusion-based video generator to imagine a trajectory. Song *et al*. [12] decouples the complex whole-body simulation, which may exhibit discontinuities due to contact, into two separate continuous domains: a differentiable simulator world model and a non-differentiable simulator that acts as a ground truth to the differentiable simulator, closing the loop. This work, although reaching robust locomotion performance in the real world zero-shot, still requires accurate simulators, which may not always be



(a) An example of a subfigure.

(b) Another example of a subfigure.

Figure 1. **Overview of the ED2 Framework:** The ED2 framework consists of two main components: sub-dynamics discovery (SD2) and dynamics decomposition prediction (D2P). SD2 is responsible for decomposing the dynamics by creating a partition G over the action dimensions. D2P utilizes this partition to split the action a_{t-1} into multiple sub-actions according to G, enabling decomposed predictions based on s_t and each sub-action a_{G_k}. The final prediction h_t is obtained by aggregating the outputs of all sub-dynamics models, which is then used to generate the next state s_{t+1} and the reward r_t. [6]

readily available for diverse task sets. Model ensembling is also widely used in model construction for uncertainty estimation, providing a more reliable prediction [7].

2.2. Dynamics Decomposition

Most world model works are limited to investigating how to better model the dynamics in a black-box manner, but they all ignore the nature of the environmental dynamics themselves. ED2 [6], proved that the breakdown into subdynamics can be beneficial. However, they lacked empirical evidence that supported the technical aspect of the idea. Most notably, their technique for creating subdynamics was application-specific, and their subdynamics were extracted from proprioceptive data which limits the scalability to large internet-scale data training and sim-to-real performance. We hope to introduce a method to improve upon these limitations.

2.3. Object-centric Dynamics

Object-centric representation learning has gained popularity in understanding more complicated scenes with many moving objects. Attention has been found to be very good at understanding object interdependence in an unsupervised manner. Inspired by this, SlotAttention [9] presents an architecture component that interfaces with the output of a convolutional neural network and produces a set of task-dependent abstract representations that they call "slots." Slots bind to an object through a competitive procedure over multiple rounds of attention. SAVi [8] extended this idea to videos, predicting frame reconstructions or optical flow over a sequence of images. SlotFormer [13] utilized Slots to disentangle individual objects from the scene and learn their interactions completely unsupervised.

3. Method

3.1. Background

We focus on discrete-time infinite horizon Reinforcement Learning (RL) scenarios categorized by system states $s \in \mathbb{R}^n = \mathcal{S}$, actions $a \in R^m = \mathcal{A}$, dynamics function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{A}$. Combined, these form the Markov Decision Problem (MDP) described by tuple $(\mathcal{S}, \mathcal{A}, f, r, \gamma)$, where γ is the discount factors. Actions are samples at each timestep t by stochastic policy $a_t \sim \pi_\theta(\cdot|s_t)$, parameterized by θ. The goal of the policy is to maximize the cumulative discounted rewards:

$$\max_\theta J(\theta) := \max_\theta \mathbb{E}_{s_1 \sim \rho(\cdot), a_t \sim \pi_\theta(\cdot|s_t)} \left[\sum_{t=1}^{\infty} \gamma^t r(s_t, a_t) \right]$$

where $\rho(s_1)$ is the initial state distribution. We leave further details of RL policy learning to the reader's own interest, since the core idea of our method focuses on learning the dynamics function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{A}$. In the dynamics decomposition section 2.3, we assume that our states are proprioceptive robot states, as this helps to build intuition. However, the goal in the next section and beyond is to treat states as videos of the robot. This allows our method to scale to more general and more prevalent scenarios in robot learning.

3.2. Dynamics Decomposition

Given an environment with a m-dimensional action space and the index of each action dimension $\Lambda = \{1, 2, \cdots, m\}$, any disjoint partition $\mathcal{G} = G_1, \cdots, G_k$ over Λ corresponds to a particular way of decomposing the action space. For each dimension of action i in Λ, we define the action space A^i, which satisfies $A = A^1 \times \cdots \times A^m$. The decomposition of the action space under partition $\mathcal{G}$ is defined as $A^{\mathcal{G}} = \{A^{G_1}, \cdots, A^{G_k}\}$, where subaction spaces $A^{\mathcal{G}_j} = \prod_{x \in G_j} A^x$. ED2 [6] formalizes a method to create a

partition $\mathcal{G}$. They utilize a feature extraction method where they extract the properties of each action dimension by computing the Pearson correlation coefficient between the action dimensions and state dimensions over episode rollouts of a random agent acting in the environment. They then clusters nearby action dimensions. This yields the "subdynamics" (i.e. $G_1, ..., G_8$ in Figure 1).

With partition $\mathcal{G}$, ED2 partitions the neural network that makes up the dynamics model $f : \mathcal{S} \times \mathcal{A} \to \mathcal{A}$. For example, the recurring state-space model (RSSM) that is used in DreamerV3 [4] would be partitioned into m separate RSSM modules to model each subdynamic independently:

$$s_{t+1} = \frac{1}{k} \sum_{i=1}^{k} f_i(s, a^{G_i}), \forall s, a \in S \times A$$

where f_i denotes our partitioned neural network, with k total partitions. Each f_i has input dimensions $(dim(s) + dim(a^{G_i}))$ and output dimension $dim(s)$, representing a separate dynamics function for each partition.

Empirically, ED2 showed promising results in complex locomotion tasks, namely *DeepMind Control Humanoid*. However, we observe multiple drawbacks with this method and pose potential research questions built on their findings:

Drawbacks

1. ED2 explicitly identified action dimensions via clustering methods, which limits the scope to discrete control tasks and limits the generalizing to different robot modalities.

2. ED2 does not support image data.

Research Questions:

1. Can we implicitly identify subdynamics of any continuous control tasks, i.e. utilizing a parameterized model to decouple subdynamics.

2. Given a diverse dataset of different robot morphologies, can we decompose it into a set of distinct subdynamics and utilize them to generalize to new tasks/modalities. In other words, can we build a physics simulator composed of many independent *unit physics* (subdynamics) modules that allow us to generalize our dynamics model to new scenarios?

3.3. Attention to Decomposing Dynamics

In light of these issues, we explore attention as a means of decomposing dynamics. Attention has long been known to be good at learning object-centric representations in dynamic scenes.

Slot Attention [9] is a mechanism designed to iteratively bind inputs to a fixed number of "slots," which can represent object-centric latent variables. SAVi (Slot Attention for Video) extended this approach to dynamic visual scenes. SAVi uses a recurrent attention mechanism to process video sequences, where each frame is encoded into feature maps $\mathbf{F}_t$. These feature maps are iteratively updated into slots $\mathbf{S}_t$ using the slot attention module.

The Slot Attention module involves computing attention weights using a dot product attention mechanism:

$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right),$$

where $\mathbf{Q}$ (queries) are the slot representations, $\mathbf{K}$ (keys), and $\mathbf{V}$ (values) are derived from the feature map $\mathbf{F}_t$. The updated slot representations $\mathbf{S}_{t+1}$ are computed as:

$$\mathbf{S}_{t+1} = \text{GRU}(\mathbf{S}_t, \mathbf{A}\mathbf{V}),$$

with a GRU that provides iterative refinement. SAVi further integrates temporal consistency by conditioning slots on prior time steps, enabling smooth tracking of objects and their dynamics across frames.

We attempt to utilize SAVi to learn rigid-body subdynamics from videos.

4. Data

We compose a variety of data sets for the various tasks that we want to perform.

The authors of TD-MPC2 open-sourced their episodic training data for continuous control tasks. In particular, they released an 80-task dataset containing 545M transitions (34 GB) and a 30-task dataset with 345M transitions (20 GB), the latter of which we utilize since we stay within the DeepMind Control suite for preliminary tests. The data encompass a wide range of behaviors, from random to expert policies, across multiple embodiments and action spaces. This diversity of data allowed us to sufficiently train robust world models to solve continuous control tasks. For further details on the implementation of the data collection pipeline, we encourage readers to refer to the TD-MPC2 paper itself.

DreamerV3's authors did not open-source their data. Therefore, we collected our own datasets using replay buffers throughout the training process. The datasets comprised of single-task episodic data for various continuous control tasks in the DeepMind Control suite. Since the policy gradually improved throughout the training process, the data encompasses a wide range of behaviors, from random to expert policies.

The replay buffer was configured with a capacity of 1 million episodes. Each episode in the dataset contains transitions that include observations, actions, rewards, and state information. The batch size for training was set to 16, with a batch length of 64 timesteps. This approach allows the agent to learn from a diverse set of experiences, improving

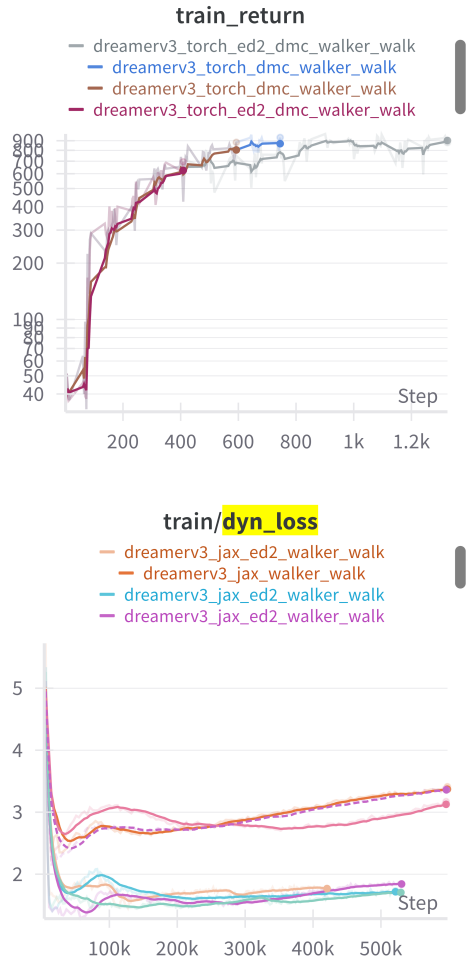


Figure 2. Episode Rewards and Dynamics Loss ($\mathcal{L}_{\text{dynamics}} = \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{D}}\left[\|f_\theta(s_t,a_t) - s_{t+1}\|_2^2\right]$) of DreamerV3 vs DreamerV3+ED2 on walker-walk task

its ability to generalize across different tasks within each modality.

The data collection process was integral to DreamerV3's learning, as it used this information to train its world model, critic, and actor networks. The world model learned to predict future states and rewards, while the critic and actor networks used these predictions to improve the agent's policy.

Lastly, we collect data to explore attention-based dynamics decomposition. The MoCapAct dataset comprises video rollouts of a CMU humanoid performing various tasks using expert policies. This large multi-task set, albeit under a single modality, represents a sufficiently diverse distribution for our intention. The dataset encompasses 2,589 clip snippets derived from 836 original motion capture clips, each 4-6 seconds long with 1-second overlaps. Two versions of the rollout dataset are available: a "large" 600-gigabyte collection with 200 rollouts per snippet, totaling 67 million environment transitions (equivalent to 620 hours in the simulator), and a "small" 50-gigabyte collection with 20 rollouts per snippet, containing 5.5 million environment transitions (51 hours of simulation time). Each video in the dataset contains sequences of the humanoid executing different actions, capturing complex motions and interactions with the environment. We applied a 70-20-10 train, test, validation split to this dataset, ensuring that the test set included completely new tasks to sufficiently evaluate generalization to unseen behaviors. Furthermore, we standardize the frame rate and video length across all samples to enable consistency in downstream tasks. The composition of this dataset, featuring a single embodiment across various tasks, offers a unique opportunity to study how object-centric models can generalize across different behaviors while maintaining consistency in object representation within a fixed morphology.

Task	DreamerV3	DreamerV3 + ED2
Walker-Walk	3.22	1.84
Humanoid-Walk	6.01	1.21
Cheetah-Run	–	–
Hopper-Hop	–	–
Walker-Run	–	–

Table 1. Converged dynamics loss comparison for DreamerV3 and DreamerV3 + ED2 across tasks.

Task	TDMPC2	TDMPC2 + ED2
Walker-Walk	–	–
Humanoid-Walk	–	–
Cheetah-Run	5e-5	4e-5
Hopper-Hop	1e-3	1e-3
Walker-Run	5e-5	3e-5

Table 2. Converged dynamics loss comparison for TDMPC2 and TDMPC2 + ED2 across tasks.

5. Experiments

5.1. ED2 Baseline Tests

The ED2 paper implemented their concept on two well-known algorithms for single task expert learning: MBPO [7] and DreamerV1 [3]. However, to better understand ED2 on more modern control algorithms, we implemented the concept in DreamerV3 [4] and TDMPC2 [5]. Dreamer learn behaviors by propagating analytic gradients of learned state values back through trajectories imagined in the compact state space of a

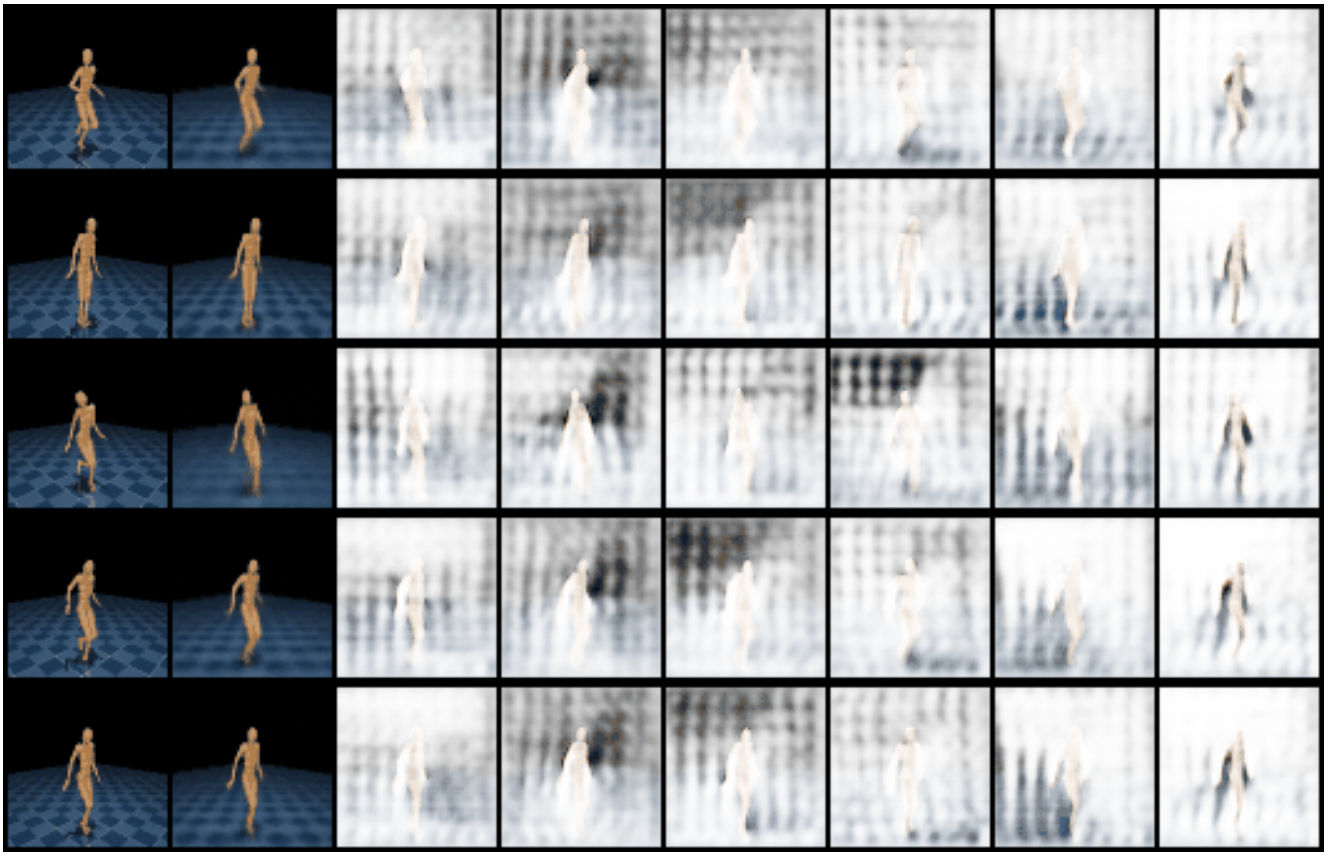


Figure 3. SAVi results on MOCAPACT data after 30k training steps ($\sim$ 8 hours). Along the rows are 5 different humanoid task/actions. The left-most column is our test-sample, and the columns to the right of those are Slots (7 slots), each associated to capturing distinct subdynamics of the whole-body dynamics.

learned world model. TDMPC2 performs local trajectory optimization using an MPC in the latent space of a learned implicit. Both works use the world models approach for imagination in the latent space.

We tested the 4 algorithms - **DreamerV3**, **DreamerV3 + ED2**, **TDMPC2**, **TDMPC2 + ED2** - in a variety of tasks: *DMControl Humanoid Walk, DMControl Walker Walk, DMControl Cheetah Run, DMControl Hopper Hop*. The vanilla **TDMPC2** and **DreamerV3** acted as baselines for our ED2 implementations, respectively. Furthermore, the algorithms were tested on 3+ seed to eliminate any occurrences of off-chance high returns. Both TDMPC2 and DreamerV3 optimized over their respective offline datasets as mentioned in the previous section 4. We omit certain tasks due to their non-convergence, algorithm suppport, or compute constraints.

From Table 1, we observe that environment decomposition improves the loss of the converged dynamics model in both the Walker-walk and Humanoid-walk tasks. This indicates that our world model benefits from the incorporation of dynamics decomposition. However, we note that the episode rewards for both algorithms remained largely comparable. We attribute this outcome to the limitations of the vanilla ED2 approach when applied to more complex architectures such as DreamerV3. Specifically, achieving similar performance gains to those demonstrated in the original ED2 paper would likely require additional architectural optimizations tailored to the advanced structure of DreamerV3.

From Table 2, we observe a marginal improvement in the dynamics loss for the Cheetah-run, Hopper-hop, and Walker-run tasks with environment decomposition. Similarly, the episode rewards of both algorithms remained largely equivalent, which we attribute to the same limitations noted in the context of DreamerV3. Furthermore, we find that the performance gains from ED2 are more pronounced with DreamerV3 compared to TDMPC2. We hypothesize that this discrepancy arises from differences in the world model backbone. While TDMPC2 employs Multi-Layer Perceptrons (MLPs) as its world model backbone, DreamerV3 utilizes an ensemble of GRUCells integrated

into an RSSM. MLP-based world models may lack the representational capacity necessary for effectively modeling decomposed dynamics. Empirical results suggest that the GRUCells in the RSSM architecture of DreamerV3 offer a superior backbone for modeling decomposed dynamics. More significantly, these results demonstrate that decomposing dynamics improves the dynamics loss even on state-of-the-art models, highlighting a pathway to develop more robust and better-performing algorithms.

5.2. SlotAttention for Dynamics Decomposition

The preceding results provide strong evidence that dynamics decomposition effectively reduces dynamics loss. Additionally, a key insight emerged: GRUCells exhibit greater representational capacity compared to the MLPs employed in TD-MPC2. SAVi, which leverages GRUCells to model slot dynamics over multiple frames, aligns naturally with our findings. We train and evaluate SAVi with the hyperparameter *slots* = *7* on the CMU-humanoid dataset described in Section 4, investigating how sub-dynamics are distributed across different slots, as illustrated in Figure 3.

Our analysis reveals that Slot Attention tends to strongly bias learning toward capturing the entire rigid-body dynamics within a single slot, leaving the remaining six slots (left 6 columns in figure 3) to encode sparse and partial representations of the actual rigid-body dynamics. This behavior persisted despite efforts to tune the number of slots. We attribute this limitation to an inherent design constraint of Slot Attention, as the original authors note its sensitivity to closely interacting object dynamics.

A potential solution would be to condition the Slot Attention module on action dimensions. This approach, analogous to the ED2 framework, could guide Slot Attention to implicitly associate actions with corresponding state transitions, enabling the automatic clustering of shared dynamics into distinct partitions. Furthermore, a semi-supervised method for dynamics discovery, such as keypoint detection followed by an attention mechanism, could offer a more efficient alternative, as demonstrated in recent work [11]. However, we acknowledge that exploring such model variants was beyond the scope of this project but are still excited for potential in this direction.

6. Conclusion

Through this work, we discuss some of the ongoing issues in multi-task reinforcement learning and propose a new pathway to scale: decomposing dynamics model into sub-dynamics. Our evaluations demonstrated that decomposing the dynamics through ED2 improves our dynamics prediction loss. We proceed to explore dynamics decomposition in the image space using SlotAttention and reach a bittersweet conclusion: SlotAttention learns rigid-body dynamics well; however, it biases the learning to take place within a single slot. Future directions include finding a way to induce the model to partition the dynamics, composing sub-dynamics for out-of-distribution data, and assessing more thoroughly the generalization of our method on large multi-task datasets.

References

[1] Philip J. Ball, Cong Lu, Jack Parker-Holder, and Stephen Roberts. Augmented world models facilitate zero-shot dynamics generalization from a single offline environment, 2021. 1

[2] Ignat Georgiev, Varun Giridhar, Nicklas Hansen, and Animesh Garg. Pwm: Policy learning with large world models, 2024. 1

[3] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. 4

[4] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. 3, 4

[5] Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 4

[6] Xinyi Hao, Matthew Brown, Naveen Rajasekaran, Sergey Levine, and Chelsea Finn. Ed2: Environment dynamics decomposition world models for continuous control. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. 2

[7] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 2, 4

[8] Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video, 2022. 2

[9] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention, 2020. 2, 3

[10] Tobias Mattes, Kai Zhang, Kuang-Huei Lee, Chelsea Finn, and Sergey Levine. Hieros: Hierarchical imagination on structured state space sequence world models. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. 1

[11] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 6

[12] Jiayuan Song, Kenji Liu, Haozhi Zhu, Jiajun Wu, and Bill Freeman. Learning quadruped locomotion using differentiable simulation. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. 1

[13] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models, 2023. 2

[14] Huihan Zhou, Tongzhou Yu, Nishanth Saxena, and Sergey Levine. Robodreamer: Learning compositional world models for robot imagination. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. 1