GenRL: Multimodal-foundation world models for generalization in embodied agents

Pietro Mazzaglia*
IDLab, Ghent University

Tim Verbelen
VERSES AI Research Lab

Bart Dhoedt
IDLab, Ghent University

Aaron Courville
Mila, University of Montreal

Sai Rajeswar
ServiceNow Research

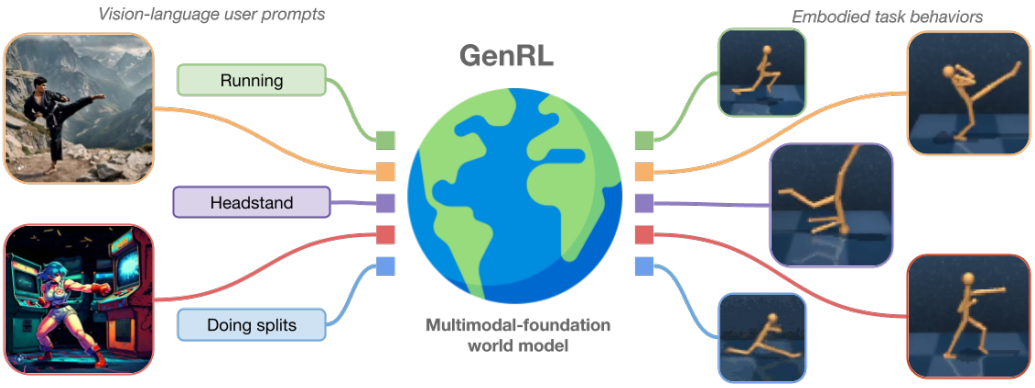


Figure 1: *Multimodal-foundation world models* connect and align the video-language space of a foundation model with the latent space of a generative world model for reinforcement learning, requiring vision-only data. Our *GenRL* framework turns visual and/or language prompts into latent targets and learns to realize the corresponding behaviors by training in the world model's imagination.

Abstract

Learning generalist embodied agents, able to solve multitudes of tasks in different domains is a long-standing problem. Reinforcement learning (RL) is hard to scale up as it requires a complex reward design for each task. In contrast, language can specify tasks in a more natural way. Current foundation vision-language models (VLMs) generally require fine-tuning or other adaptations to be adopted in embodied contexts, due to the significant domain gap. However, the lack of multimodal data in such domains represents an obstacle to developing foundation models for embodied applications. In this work, we overcome these problems by presenting multimodal-foundation world models, able to connect and align the representation of foundation VLMs with the latent space of generative world models for RL, without any language annotations. The resulting agent learning framework, GenRL, allows one to specify tasks through vision and/or language prompts, ground them in the embodied domain's dynamics, and learn the corresponding behaviors in imagination. As assessed through large-scale multi-task benchmarking in locomotion and manipulation domains, GenRL enables multi-task generalization from language and visual prompts. Furthermore, by introducing a data-free policy learning strategy, our approach lays the groundwork for foundational policy learning using generative world models.

Website, code and data: `mazpie.github.io/genrl`

*Work done while interning at Mila/ServiceNow Research. Email: `pietro.mazzaglia@ugent.be`

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-0865

1 Introduction

Foundation models are large pre-trained models endowed with extensive knowledge of the world, which can be readily adapted for a given task [44]. These models have demonstrated extraordinary generalization capabilities in a wide range of vision [28, 45, 67] and language tasks [43, 19, 53, 11]. As we aim to extend this paradigm to embodied applications, where agents physically interact with objects and other agents in their environment, we require generalist agents that are capable of reasoning about these interactions and executing action sequences within these settings [61].

Reinforcement learning (RL) allows agents to learn complex behaviors from visual and/or proprioceptive inputs [18, 26, 27] by maximizing a specified reward function. Scaling up RL to multiple tasks and embodied environments remains challenging as designing reward functions is a complicated process, requiring expert knowledge and prone to errors which can lead to undesired behaviors [1]. Recent work has proposed the adoption of visual-language models (VLMs) to specify rewards for visual environments using language [4, 48, 39], e.g. using the similarity score computed by CLIP [45] between an agent's input images and text prompts. However, these approaches mostly require fine-tuning of the VLM [38], otherwise, they tend to work reliably only in a few visual settings [48].

In most RL settings, we lack multimodal data to train or fine-tune domain-specific foundation models, due to the costs of labelling agents' interactions and/or due to the intrinsic unsuitability of some embodied contexts to be converted into language. For instance, in robotics, it's non-trivial to convert a language description of a task to the agent's actions which are hardware-level controls, such as motor currents or joint torques. These difficulties make it hard to scale current techniques to large-scale generalization settings, leaving open the question:

How does one effectively leverage foundation models for generalization in embodied domains?

In this work, we present GenRL, a novel approach requiring no language annotations that allows training agents to solve multiple tasks from visual or language prompts. GenRL learns multimodal-foundation world models (MFWMs), where the joint embedding space of a foundation video-language model [57] is connected and aligned with the representation of a generative world model for RL [23], using only vision data. The MFWM allows the specification of tasks by grounding language or visual prompts into the embodied domain's dynamics. Then, we introduce an RL objective that enables learning to accomplish the specified tasks in imagination [24], by matching the prompts in latent space.

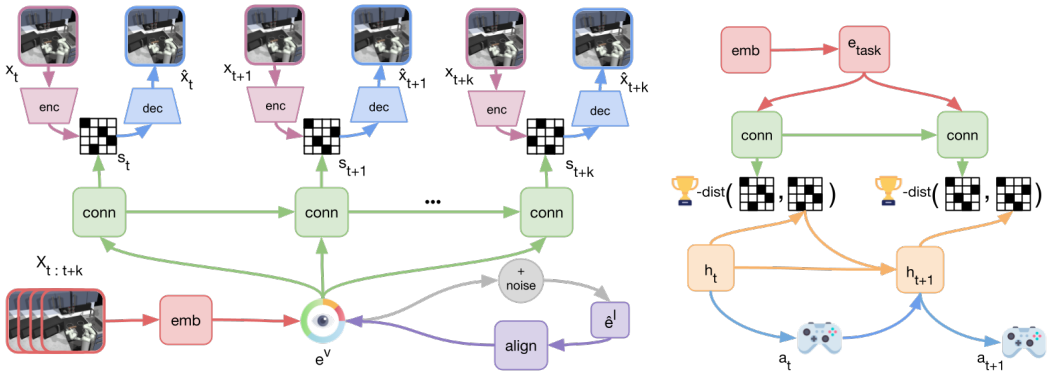
Compared to previous work in world models and VLMs for RL, one emergent property of GenRL is the possibility to generalize to new tasks in a completely data-free manner. After training the MFWM, it possesses both strong priors over the dynamics of the environment, and large-scale multimodal knowledge. This combination enables the agent to interpret a large variety of task specifications and learn the corresponding behaviors. Thus, analogously to foundation models for vision and language, GenRL allows generalization to new tasks without additional data and lays the groundwork for foundation models in embodied RL domains [44].

2 Preliminaries and background

Additional related works can be found in Appendix A.

Problem setting. The agent receives from the environment observations $x \in \mathcal{X}$ and interacts with it through actions $a \in \mathcal{A}$. In this work, we focus on visual reinforcement learning, so observations are images of the environment. The objective of the agent is to accomplish a certain task τ, which can be specified either in the observation space x_τ, e.g. through images or videos, or in language space y_τ, where $\mathcal{Y}$ represents the space of all possible sentences. Crucially, compared to a standard RL setting, we do not assume that a reward signal is available to solve the task. When a reward function exists, it is instead used to evaluate the agent's performance.

Generative world models for RL. In model-based RL, the optimization of the agent's actions is done efficiently, by rolling out and scoring imaginary trajectories using a (learned) model of the environment's dynamics. In recent years, this paradigm has grown successful thanks to the adoption of generative world models, which learn latent dynamics by self-predicting the agent's inputs [23]. World models have shown impressive performance in vision-based environments [24],



(a) Connecting and aligning (Section 3.2) (b) Learning task behavior (Section 3.3)

Figure 2: *Overview of GenRL.* The agent learns a multimodal-foundation world model that connects and aligns (a) the representation of a foundation VLM with the latent states of a generative world model. Given a certain task prompt, (b) the model allows embedding the task and translating into targets in the latent dynamics space, which the agent can learn to achieve by using RL in imagination.

improving our ability to solve complex and open-ended tasks [26]. Generative world models have been successfully extended to many applications, such as exploration [51], skill learning [42], solving long-term memory tasks [50], and robotics [58, 16].

Foundations models for RL. Large language models (LLMs) have been used for specifying behaviors using language [41, 29, 56, 59], but this generally assumes the availability of a textual interface with the environment or that observations and/or actions can be translated to the language domain. The adoption of vision-language models (VLMs) reduces these assumptions, as it allows the evaluation of behaviors in the visual space. However. this approach has yet to show robust performance, as it generally requires fine-tuning of the VLM [4, 15], prompt hacking techniques [9] or visual modifications to the environment [4].

Vision-language generative modelling. Given the large success of image-language generative models [49], recent efforts in the community have focused on replicating and extending such success to the video domain, where the temporal dimension introduces new challenges, such as temporal consistency and increased computational costs [30, 3]. Video generative models are similar to world models for RL, with the difference that generation models outputs are typically not conditioned on actions, but rather conditioned on language [30] or on nothing at all (i.e. an unconditional model).

3 GenRL

3.1 World models for RL

GenRL learns a task-agnostic world model representation by modelling the sequential dynamics of the environment in a compact discrete latent space S [24, 26]. Latent states $s \in S$ are sampled from independent categorical distributions. The gradients for training the model are propagated through the sampling process with straight-through estimation [5].

The world model is made of the following components:

$$\begin{aligned} &\text{Encoder:} && q_\phi(s_t|x_t), && \text{Sequence model:} && h_t = f_\phi(s_{t-1}, a_{t-1}, h_{t-1}), \\ &\text{Decoder:} && p_\phi(x_t|s_t), && \text{Dynamics predictor:} && p_\phi(s_t|h_t), \end{aligned}$$

trained with the loss:

$$\mathcal{L}_\phi = \sum_t \underbrace{D_{\mathrm{KL}}\left[q_\phi(s_t|x_t) \| p_\phi(s_t|s_{t-1}, a_{t-1})\right]}_{\text{dyn loss}} - \underbrace{\mathbb{E}_{q_\phi(s_t|x_t)}[\log p_\phi(x_t|s_t)]}_{\text{recon loss}}, \tag{1}$$

where $p_\phi(s_t|s_{t-1}, a_{t-1})$ is a shorthand for $p_\phi(s_t|f_\phi(s_{t-1}, a_{t-1}, h_{t-1}))$. The sequence model is implemented as a linear GRU cell [8]. Differently from recurrent state space models (RSSM; [25]),

https://doi.org/10.52202/079017-0865

for our framework, encoder and decoder models are not conditioned on the information present in the sequence model. This ensures that the latent states only contain information about a single observation, while temporal information is stored in the hidden state of the sequence model. Given the simpler encoder-decoder strategy of our model, the encoder can be seen as a probabilistic visual tokenizer, which is grounded in the target embodied environment [64].

3.2 Multimodal-foundation world models

Multimodal VLMs are large pre-trained models that have the following components:

$$\text{Vision embedder:} \quad e^{(v)} = f_{\text{PT}}^{(v)}(x_{t:t+k}), \qquad \text{Language embedder:} \quad e^{(l)} = f_{\text{PT}}^{(l)}(y),$$

where $x_{t:t+k}$ is a sequence of visual observations and y is a text prompt. For video-language models, k is generally a constant number of frames (e.g. $k \in \{4, 8, 16\}$ frames). Image-language models are a special case where $k = 1$ as the vision embedder takes a single frame as an input. For our implementation, we adopt the InternVideo2 video-language model [57] (with k=8).

To connect the representation of the multimodal foundation VLM with the world model latent space, we instantiate two modules: a *latent connector* and a *representation aligner*:

$$\text{Connector:} \quad p_\psi(s_{t:t+k}|e), \qquad \mathcal{L}_{\text{conn}} = \sum_t D_{\text{KL}}\big[p_\psi(s_t|s_{t-1}, e) \| \text{sg}(q_\phi(s_t|x_t))\big],$$

$$\text{Aligner:} \quad e^{(v)} = f_\psi(e^{(l)}), \qquad \mathcal{L}_{\text{align}} = \|e^{(v)} - f_\psi(e^{(l)})\|_2^2,$$

where $\text{sg}(\cdot)$ indicates to stop gradients propagating.

The connector learns to predict the latent states of the world model from embeddings in the VLM's representation space. The connector's objective consists of minimizing the KL divergence between its predictions and the world model's encoder distribution. While more expressive architectures, such as transformers [55] or state-space models [21] could be adopted, we opt for a simpler GRU-based architecture for video modelling. This way, we keep the method simple and the architecture of the connector is symmetric with respect to the world model's components.

Aligning multimodal representations. The connector learns to map visual embeddings from the pretrained VLM to latent states of the world model. When learning the connector from visual embeddings $e^{(v)}$, we assume it can generalize to the (theoretical) corresponding language embedding $e^{(l)}$ if the angle θ between the two embeddings is small enough, as shown in Fig. 3a. This can be expressed as $\cos\theta > c$ or $\theta < \arccos c$, with c a small positive constant [68].

Multimodal VLMs trained with contrastive learning exhibit a *multimodality gap* [34], where the spherical embeddings of different modalities are not aligned. Given a dataset of vision-language data, this projective function can be learned. However, in embodied domains vision-language data is typically unavailable. Thus, we have to find a way to align the representations using *no language annotations*.

Previous methods inject the noise into the vision embeddings during training [66, 68]. This leads to the situation shown in Figure 3b, where c grows larger with the noise. This allows language embeddings to be close enough to their visual counterparts.

In our work, we instead learn an aligner network, which maps points surrounding $e^{(v)}$ closer to $e^{(v)}$. As represented in Fig. 3c, this way, c is unaltered but the aligner will map $e^{(l)}$ close enough to $e^{(v)}$. Since we use noise to sample points around $e^{(v)}$ the aligner model can be trained using vision-only data and thus, no language annotations.

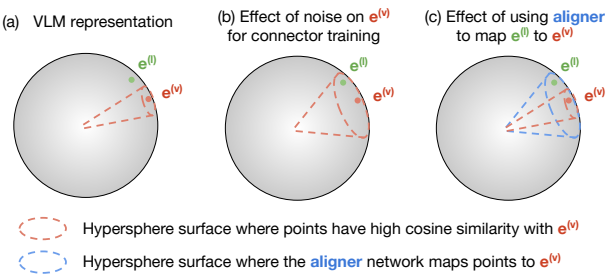


Figure 3: When training the connector on (a) the VLM's representation we can address the multimodality gap in multiple ways: (b) prior works adopt noise during the training of the connector, (c) we adopt an aligner network that learns to map points in proximity of the visual embedding close the corresponding embedding.

https://doi.org/10.52202/079017-0865

The aligner allows us to train a noise-free connector, which has two main advantages: (i) it yields higher prediction accuracy for visual embedding inputs while maintaining a similar alignment for language embedding inputs; and (ii) it is more flexible; it's easier to re-train/adapt for different noise levels, as it only requires re-training the aligner module, and its use can be avoided if unnecessary.

3.3 Specifying and learning tasks in imagination

World models can be used to imagine trajectories in latent space, using the sequential and dynamics models. This allows us to train behavior policies in a model-based RL fashion [24]. Given a task specified through a visual or language prompt, our MFWM can generate the corresponding latent states by turning the embedder's output, e_{task}, into sequences of latent states $s_{t:t+k}$ (decoded examples are shown in Figure 1). The objective of the policy model π_θ is then to match the goals specified by the user by performing trajectory matching.

The trajectory matching problem can be solved as a divergence minimization problem [13], between the distribution of the states visited by the policy π_θ and the trajectory generated using the aligner-connector networks from the user-specified prompt:

$$\theta = \arg\min_{\theta} \mathbb{E}_{a_t \sim \pi_\theta(s_t)} \Big[\sum_t \gamma^t \texttt{distance}\big(p_\phi(s_{t+1}|s_t, a_t) \| p_\psi(s_{t+1}|e_{\text{task}})\big) \Big], \quad \text{with} \quad e_{\text{task}} = f_{\text{PT}}(\cdot). \tag{2}$$

The KL divergence is a natural candidate for the distance function [13]. However, in practice, we found that using the cosine distance between linear projections of the latent states notably speeds up learning and enhances stability. We can then turn the objective in Eq. 2 into a reward for RL:

$$r_{\text{GenRL}} = \cos\big(g_\phi(s^{\text{dyn}}_{t+1}), g_\phi(s^{\text{task}}_{t+1})\big), \quad \text{with} \quad s^{\text{dyn}}_{t+1} \sim p_\phi(s_{t+1}|s_t, a_t), s^{\text{task}}_{t+1} \sim p_\psi(s_{t+1}|e_{\text{task}}), \tag{3}$$

where g_ϕ represents the first linear layer of the world model's decoder. We train an actor-critic model to maximize this reward and achieve the tasks specified by the user [26]. Additional implementation details are provided in Appendix B.

Temporal alignment. One issue with trajectory matching is that it assumes that the distribution of states visited by the agent starts from the same state as the target distribution. However, the initial state generated by the connector may differ from the initial state where the policy is currently in. For example, consider the Stickman agent on the right side of Figure 1. If the agent is lying on the ground and tasked to run, the number of steps to get up and reach running states may surpass the temporal span recognized by the VLM (e.g. in our case 8 frames), causing disalignment in the reward.

To address this initial condition alignment issue, we propose a *best matching trajectory* technique, inspired by best path decoding in speech recognition [20]. Our technique involves two steps:

1. We compare the first b states of the target trajectory with b states obtained from the trajectories imagined by the agent by sliding along the time axis. This allows one to find at which timestep t_a the trajectories are best aligned (the comparison provides the highest reward).
2. We align the temporal sequences in the two possible contexts: (a) if a state from the agent sequence comes before t_a, the reward uses the target sequence's initial state; and (b) if the state comes k steps after t_a, it's compared to the s_{t+k} state from the target sequence.

In all experiments, we fix $b = 8$ (number of frames of the VLM we use [57]), which we found to strike a good compromise between comparing only the initial state ($b = 1$) and performing no alignment ($b =$ imagination horizon). An ablation study can be found in Appendix E.

4 Experiments

Overall, we employ a set of 4 locomotion environments (Walker, Cheetah, Quadruped, and a newly introduced Stickman environment) [54] and one manipulation environment (Kitchen) [22], for a total of 35 tasks where the agent is trained without rewards, using only visual or language prompts. Details about the datasets, tasks, and prompts used can be found in the Appendix C.

https://doi.org/10.52202/079017-0865

Table 1: *Language-to-action in-distribution.* Offline RL from language prompts on tasks that are included in the agent's training dataset. Scores are episodic rewards averaged over 10 seeds (± standard error) rescaled using min-max scaling with $(\min = \text{random policy}, \max = \text{expert policy})$.

	Image-language VLM				Video-language VLM				
	IQL	TD3+BC	TD3	WM-CLIP	IQL	TD3+BC	TD3	WM-CLIP	GenRL
walker stand	0.67 ± 0.03	0.92 ± 0.02	0.93 ± 0.03	1.01 ± 0.0	0.66 ± 0.05	0.64 ± 0.03	1.01 ± 0.0	0.94 ± 0.01	1.02 ± 0.0
walker run	0.24 ± 0.03	0.27 ± 0.01	0.09 ± 0.02	0.05 ± 0.02	0.29 ± 0.02	0.24 ± 0.02	0.35 ± 0.01	0.7 ± 0.01	0.77 ± 0.02
walker walk	0.41 ± 0.05	0.34 ± 0.05	0.14 ± 0.0	0.21 ± 0.01	0.4 ± 0.03	0.44 ± 0.03	0.88 ± 0.02	0.91 ± 0.02	1.01 ± 0.0
cheetah run	0.41 ± 0.05	0.0 ± 0.01	-0.01 ± 0.0	-0.0 ± 0.0	0.15 ± 0.02	-0.01 ± 0.0	0.37 ± 0.01	0.56 ± 0.03	0.74 ± 0.01
quadruped stand	0.56 ± 0.02	0.64 ± 0.04	0.65 ± 0.04	0.97 ± 0.0	0.52 ± 0.06	0.43 ± 0.05	0.61 ± 0.05	0.97 ± 0.0	0.97 ± 0.0
quadruped run	0.3 ± 0.03	0.28 ± 0.02	0.24 ± 0.02	0.27 ± 0.0	0.38 ± 0.03	0.25 ± 0.02	0.26 ± 0.01	0.61 ± 0.02	0.86 ± 0.02
quadruped walk	0.26 ± 0.02	0.31 ± 0.02	0.28 ± 0.01	0.47 ± 0.02	0.32 ± 0.02	0.28 ± 0.04	0.28 ± 0.02	0.92 ± 0.01	0.93 ± 0.01
stickman stand	0.45 ± 0.06	0.58 ± 0.04	0.06 ± 0.04	0.71 ± 0.02	0.43 ± 0.04	0.45 ± 0.05	0.08 ± 0.02	0.32 ± 0.01	0.7 ± 0.02
stickman walk	0.4 ± 0.04	0.48 ± 0.04	0.18 ± 0.01	0.23 ± 0.01	0.51 ± 0.02	0.46 ± 0.03	0.41 ± 0.02	0.65 ± 0.05	0.83 ± 0.01
stickman run	0.2 ± 0.01	0.22 ± 0.02	0.03 ± 0.0	0.19 ± 0.01	0.23 ± 0.02	0.19 ± 0.02	0.21 ± 0.0	0.35 ± 0.01	0.35 ± 0.01
kitchen microwave	0.06 ± 0.04	0.22 ± 0.11	0.0 ± 0.0	0.0 ± 0.0	0.01 ± 0.01	0.0 ± 0.0	0.11 ± 0.08	0.9 ± 0.09	0.97 ± 0.02
kitchen light	0.14 ± 0.04	0.11 ± 0.11	0.59 ± 0.16	0.1 ± 0.09	0.02 ± 0.01	0.0 ± 0.0	0.18 ± 0.11	0.26 ± 0.13	0.46 ± 0.09
kitchen burner	0.21 ± 0.05	0.18 ± 0.05	0.09 ± 0.05	0.03 ± 0.03	0.05 ± 0.02	0.02 ± 0.01	0.31 ± 0.1	0.78 ± 0.06	0.62 ± 0.07
kitchen slide	0.02 ± 0.01	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.04 ± 0.02	0.02 ± 0.02	0.7 ± 0.14	0.88 ± 0.04	1.0 ± 0.0
overall	0.31 ± 0.03	0.33 ± 0.04	0.23 ± 0.04	0.30 ± 0.02	0.29 ± 0.02	0.24 ± 0.02	0.41 ± 0.05	0.70 ± 0.04	0.80 ± 0.02

4.1 Offline RL

In offline RL, the objective of the agent is to learn to extract a certain task behavior from a given fixed dataset [33]. The performance of the agent generally depends on its ability to 'retrieve' the correct behaviors in the dataset and interpolate among them. Popular techniques for offline RL include off-policy RL methods, such as TD3 [18], advantage-weighted behavior cloning, such as IQL [31], and behavior-regularized approaches, such as CQL [32] or TD3+BC [17].

We aim to assess the multi-task capabilities of different approaches for designing rewards using VLMs. We collected large datasets for each of the domains evaluated, containing a mix of structured data (i.e. the replay buffer of an agent [26] learning to perform some tasks) and unstructured data (i.e. exploration data collected using [51]). The datasets contain **no reward information** and **no text annotations** of the trajectories. The rewards for training for a given task must be inferred by the agent, i.e. using the cosine similarity between observations and the given prompt or, in the case of GenRL, using our reward formulation (Eq. 3).

We compare GenRL to two main categories of approaches:

- *Image-language rewards*: following [48], the cosine similarity between the embedding for the language prompt and the embedding for the agent's visual observation is used as a reward. For the VLM, we adopt the SigLIP-B [65] model as it's reported to have superior performance than the original CLIP [45].
- *Video-language rewards*: similar to the image-language rewards, with the difference that the vision embedding is computed from a video of the history of the last k frames, as done in [15]. For the VLM, we use the InternVideo2 model [57], the same used for GenRL.

The evaluation compares GenRL to various offline RL methods from the literature, including IQL, TD3+BC, and TD3. We also introduce a model-based baseline, *WM-CLIP*. This baseline is the antithesis of GenRL as, rather than learning a connector and an aligner, it learns a "reversed connector". This module learns to predict VLM embeddings from the world model states (GenRL does the opposite). This makes it possible to compute rewards in imagination in a similar way to the model-free baselines, by computing the cosine similarity between the visual embeddings predicted from imagined states and the task's language embeddings.

All methods are trained for 500k gradient steps, and evaluated on 20 episodes. For each task, model-free agents require training the agent from scratch, including the visual encoder, actor, and critic networks on the entire dataset. Model-based agents require training the model once for each domain and then training an actor-critic for each task. Other training and baseline details are reported in Appendix D.

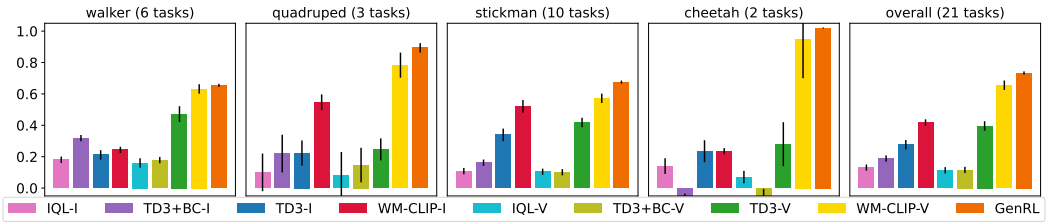


Figure 4: *Language-to-action generalization.* Offline RL from language prompts on tasks that are not deliberately included in the training dataset. Performance averaged over 10 seeds and standard error was reported with black lines. Detailed results per task in Appendix K.

Language-to-action in-distribution. We want to verify whether the methods can retrieve the task behaviors that are certainly present in the training data, when specifying the task only through language. We present results in Table 1, with episodic rewards rescaled so that 0 represents the performance of a random agent, while 1 represents the performance of an expert agent.

GenRL excels in overall performance across all domains and tasks, outperforming other methods particularly in dynamic tasks like walking and running in the quadruped and cheetah domains. However, in some static tasks of the kitchen domain, other methods occasionally outperform GenRL. This can be explained by the fact that the target sequences that GenRL infers from the prompt are often slightly in motion, even in static cases. To address this, we could set the target sequence length to 1 for static prompts, but we opted to maintain the method's simplicity and generality, acknowledging this as a minor limitation.

As expected, video-language rewards tend to perform better than image-language rewards for dynamic tasks. The less conservative approach, TD3, performs better than the other model-free baselines in most tasks, similarly to what is shown in [62]. The model-based baseline's performance, WM-CLIP-V, is the closest to GenRL's.

Language-to-action generalization. To assess multi-task generalization, we defined a set of tasks not included in the training data. Although we don't anticipate agents matching the performance of expert models, higher scores in this benchmark help gauge the generalization abilities of different methods. We averaged the performance across various tasks for each domain and summarized the findings in Figure 4, with detailed task results in Appendix K.

Overall, we observe a similar trend as for the in-distribution results. GenRL significantly outperforms all model-free approaches, especially in the quadruped and cheetah domains, where the performance is close to the specialized agents' performance. Both for image-language (-I in the Figure) and video-language (-V in the Figure) more conservative approaches, such as IQL and TD3+BC tend to perform worse. This could be associated with the fact that imitating segments of trajectories is less likely to lead to high-rewarding trajectories, as the tasks are not present in the training data.

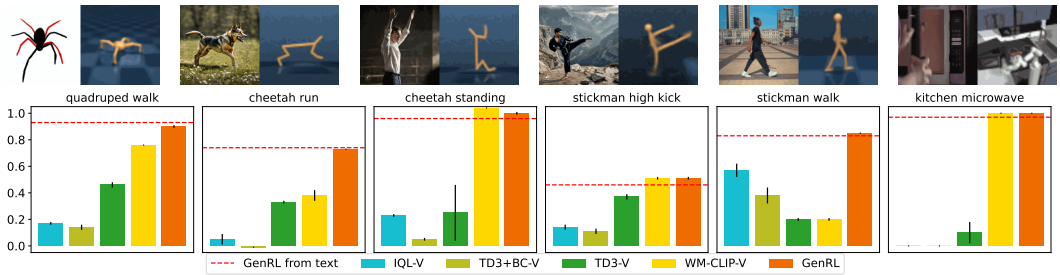


Figure 5: *Video-to-action.* GenRL allows grounding video prompts into the target environment's dynamics. It allows visualization of the model's interpretation of the prompts, using the decoder (top row), and it allows turning prompts into behaviors, leading to generally higher performance than other approaches. 10 seeds. Additional visualizations on the project website.

https://doi.org/10.52202/079017-0865

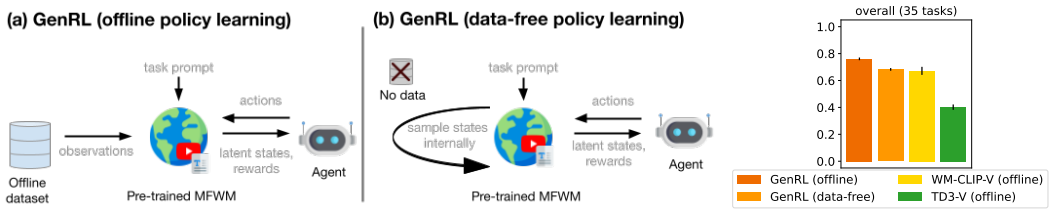


Figure 6: By removing data dependencies on actions (on-policy learning in the model's imagination), rewards (computed using only the prompt and latent states, using Eq. 3), and observations (by sampling latent states within the model), GenRL agents can be adapted for new tasks in a data-free fashion. Performance is averaged over 10 seeds and standard error is reported with black lines. Detailed results per task in Appendix K.

Video-to-action. While language strongly simplifies the specification of a task, in some cases providing visual examples of the task might be easier. Similarly as for language prompts, GenRL allows grounding visual prompts (short videos) into the embodied domain's dynamics and then learning of the corresponding behaviors.

In Figure 5, we provide behavior learning results from video prompts. The tasks included are of static and dynamic nature and span across 4 different domains. Visualizations of the videos used as prompts are available on the project website, where we also present a set of "grounded videos" generated by the model using the prompts (see snapshots at the top of Fig. 5). These can be obtained by inferring the latent targets corresponding to the vision prompts (left images, in the Figure) and then using the decoder model to decode reconstructed images (right images, in the Figure).

The results show a similar trend to the language prompts experiments and the performance when using video prompts is aligned to the language-to-action performance, for the same tasks. In general, we found it interesting that the VLM allows us to generalize to very different visual styles (drawings, realistic, AI-generated), very different camera viewpoints (quadruped, microwave), and different morphologies (cheetah tasks).

Summary. The experiments presented allow us to establish more clearly the main ingredients that contribute to the stronger performance of GenRL: (i) the video-language model helps in dynamic tasks, (ii) model-based algorithms lead to higher performance, (iii) the connection-alignment system presented generally outperforms the "reversed" way of connecting the two representations.

4.2 Data-free policy learning

In the previous section, we evaluated several approaches for designing reward using foundation VLMs. Clearly, model-free RL approaches require continuous access to a dataset, to train the actor-critic and generalize across new tasks. Model-based RL can learn the actor-critic in imagination. However, in previous work [26, 24], imagining sequences for learning behaviors first requires processing actual data sequences. The data is used to initialize the dynamics model, and obtain latent states that represent the starting states to rollout the policy in imagination. Furthermore, in order to learn new tasks, reward-labelled data is necessary to learn a reward model, which provides rewards to the agent during the task learning process.

Foundation models [44] are generally trained on enormous datasets in order to generalize to new tasks. The datasets used for the model pretraining are not necessary for the downstream applications, and sometimes these datasets are not even publicly available [43, 19]. In this section, we aim to establish a new paradigm for foundation models in RL, which follows the same principle of foundation models for vision and language. We call this paradigm *data-free policy learning* and we define it as the ability to generalize to new tasks, after pre-training, by learning a policy completely in imagination, with no access to data (not even to the pre-training dataset).

GenRL enables data-free policy learning thanks to two main reasons: the agent learns a task-agnostic MFWM on a large varied dataset during pre-training, and the MFWM enables the possibility of specifying tasks directly in latent space, without requiring any data. Thus, in order to learn behaviors in imagination, the agent can: `(i)` sample random latent states in the world model's representation,

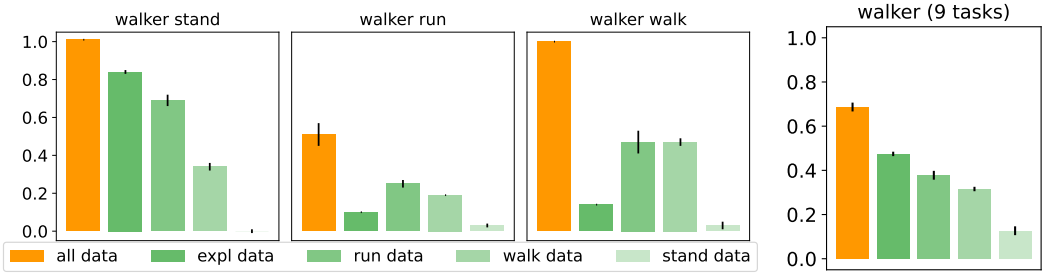


Figure 7: *Training data distribution*. Analysing the impact of the training data distribution on the generalization performance of GenRL. Performance is obtained by training behaviors in data-free mode, after training the MFWM on different subsets of the training dataset. Performance averaged over 10 seeds (black lines indicate standard error). Full results in Appendix K.

`(ii)` rollout sequences in imagination, following the policy's actions, and `(iii)` compute rewards, using the targets obtained by processing the given prompts with the connector-aligner networks.

In Figure 6, we provide a diagram that further clarifies the differences between training GenRL in an offline and in a data-free policy learning fashion. Then, we present results that compare data-free policy learning with offline RL baselines, as discussed in Section 4.1. While data-free policy learning shows a slight decrease in overall performance, its performance remains close to the original GenRL's performance and still outperforms other approaches. In Appendix K, we further show that the difference in performance is minimal across most domains, and data-free policy learning even performs better in the kitchen domain.

By employing data-free learning, after pre-training, agents can master new tasks without data. By requiring no CPU-GPU memory transfers of the data, data-free policy learning also reduces the training time of the policy, often allowing convergence within only 30 minutes of training. As we scale up foundation models for behavior learning, the ability to learn data-free will become crucial. Although very large datasets will be employed to train future foundation models, GenRL adapts well without direct access to original data, offering flexibility where data may be proprietary, licensed or unavailable.

4.3 Analysis of the training data distribution

As demonstrated in Sections 4.1 and 4.2, after training on a large dataset, a GenRL agent can adapt to multiple new tasks without additional data. The nature of the training data, detailed in Appendix C, combines exploration and task-specific data. Thus, we ask ourselves what subsets of the data are the most important ones for GenRL's training.

To identify critical data types for GenRL, we trained different MFWMs on various dataset subsets. Then, we employ data-free behavior learning to train task behaviors for all tasks. We present an analysis over subsets of the walker dataset in Figure 7.

The results confirm that a diverse data distribution is crucial for task success, with the best performance achieved by using the complete dataset, followed by the varied exploration data. Task-specific data effectiveness depends on task complexity, for instance, 'run data' proves more useful and generalizable than 'walk data' or 'stand data' across tasks. Crucially, 'stand data', which shows minimal variation, limits learning for a general agent but can still manage simpler tasks like 'lying down' and 'sitting on knees' as detailed in Appendix K.

Moving forward with training foundation models in RL, it will be essential to develop methods that extract multiple behaviors from unstructured data and accurately handle complex behaviors from large datasets. Thus, the ability of GenRL to primarily leverage unstructured data is a significant advantage for scalability.

https://doi.org/10.52202/079017-0865

5 Discussion

We introduced GenRL, a world-model based approach for grounding vision-language prompts into embodied domains and learning the corresponding behaviors in imagination. The multimodal-foundation world models of GenRL can be trained using unimodal data, overcoming the lack of multimodal data in embodied RL domains. The data-free behavior learning capacity of GenRL lays the groundwork for foundation models in RL that can generalize to new tasks without any data.

A framework for behavior generation. A common challenge with using LLMs and VLMs involves the need for prompt tuning to achieve specific tasks. As GenRL relies on a foundation VLM, similar to previous approaches [4, 48] it is not immune from this issue. However, GenRL uniquely allows for the visualization of targets obtained from specific prompts. By decoding the latent targets, using the MFWM decoder, we can visualize the interpreted prompt before training the corresponding behavior. This enables a much more explainable framework, which allows fast iteration for prompt tuning, compared to previous (model-free) approaches which often require training the agent to identify which behaviors are rewarded given a certain prompt.

Limitations. Despite its strengths, GenRL presents some limitations, largely due to inherent weaknesses in its components. From the VLMs, GenRL inherits the issue related to the multimodality gap [34, 66] and the reliance on prompt tuning. We proposed a connection-alignment mechanism to mitigate the former. For the latter, we presented an explainable framework, which facilitates prompt tuning by allowing decoding of the latent targets corresponding to the prompts. From the world model, GenRL inherits a dependency on reconstructions, which offers advantages such as explainability but also drawbacks, such as failure modes with complex observations. We further investigate this limitation in Appendix I and present other potential limitations in Appendix J.

Future work. As we strive to develop foundation models for generalist embodied agents, our framework opens up numerous research opportunities. One such possibility is to learn multiple behaviors and have another module, e.g. an LLM, compose them to solve long-horizon tasks. Another promising area of research is investigating the temporal flexibility of the GenRL framework. We witnessed that for static tasks, greater temporal awareness could enhance performance. This concept could also apply to actions that extend beyond the time comprehension of the VLM. Developing general solutions to these challenges could lead to significant advancements in the framework.

Acknowledgments and Disclosure of Funding

Pietro Mazzaglia is funded by a Ph.D. grant of the Flanders Research Foundation (FWO). This research was supported by a Mitacs Accelerate Grant.

References

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

[2] B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos, 2022.

[3] O. Bar-Tal, H. Chefer, O. Tov, C. Herrmann, R. Paiss, S. Zada, A. Ephrat, J. Hur, G. Liu, A. Raj, Y. Li, M. Rubinstein, T. Michaeli, O. Wang, D. Sun, T. Dekel, and I. Mosseri. Lumiere: A space-time diffusion model for video generation, 2024.

[4] K. Baumli, S. Baveja, F. Behbahani, H. Chan, G. Comanici, S. Flennerhag, M. Gazeau, K. Holsheimer, D. Horgan, M. Laskin, et al. Vision-language models as a source of rewards. *arXiv preprint arXiv:2312.09187*, 2023.

[5] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[6] J. Bruce, M. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, Y. Aytar, S. Bechtle, F. Behbahani, S. Chan, N. Heess, L. Gonzalez, S. Osindero, S. Ozair, S. Reed, J. Zhang, K. Zolna, J. Clune, N. de Freitas, S. Singh, and T. Rocktäschel. Genie: Generative interactive environments, 2024.

[7] E. Cetin, A. Tirinzoni, M. Pirotta, A. Lazaric, Y. Ollivier, and A. Touati. Simple ingredients for offline reinforcement learning, 2024.

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[9] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation?, 2022.

[10] F. Deng, I. Jang, and S. Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 4956–4975. PMLR, 2022.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[12] Embodiment Collaboration et al. Open x-embodiment: Robotic learning datasets and rt-x models, 2024.

[13] P. Englert, A. Paraschos, J. Peters, and M. P. Deisenroth. Model-based imitation learning by probabilistic trajectory matching. In *2013 IEEE international conference on robotics and automation*, pages 1922–1927. IEEE, 2013.

[14] A. Escontrela, A. Adeniji, W. Yan, A. Jain, X. B. Peng, K. Goldberg, Y. Lee, D. Hafner, and P. Abbeel. Video prediction models as rewards for reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[15] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

[16] S. Ferraro, P. Mazzaglia, T. Verbelen, and B. Dhoedt. Focus: Object-centric world models for robotics manipulation, 2023.

[17] S. Fujimoto and S. S. Gu. A minimalist approach to offline reinforcement learning, 2021.

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

[19] Gemini Team et al. Gemini: A family of highly capable multimodal models, 2024.

[20] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[21] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

[22] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning, 2019.

[23] D. Ha and J. Schmidhuber. World models. 2018.

[24] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination, 2020.

[25] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels, 2019.

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

[27] N. Hansen, H. Su, and X. Wang. Td-mpc2: Scalable, robust world models for continuous control, 2024.

[28] 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.

[29] M. Klissarov, P. D'Oro, S. Sodhani, R. Raileanu, P.-L. Bacon, P. Vincent, A. Zhang, and M. Henaff. Motif: Intrinsic motivation from artificial intelligence feedback, 2023.

 https://doi.org/10.52202/079017-0865

[30] D. Kondratyuk, L. Yu, X. Gu, J. Lezama, J. Huang, R. Hornung, H. Adam, H. Akbari, Y. Alon, V. Birodkar, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

[31] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning, 2021.

[32] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning, 2020.

[33] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

[34] W. Liang, Y. Zhang, Y. Kwon, S. Yeung, and J. Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning, 2022.

[35] S. Lifshitz, K. Paster, H. Chan, J. Ba, and S. McIlraith. Steve-1: A generative model for text-to-behavior in minecraft, 2024.

[36] J. Lin, Y. Du, O. Watkins, D. Hafner, P. Abbeel, D. Klein, and A. Dragan. Learning to model the world with language, 2023.

[37] H. Liu, W. Yan, M. Zaharia, and P. Abbeel. World model on million-length video and language with blockwise ringattention, 2024.

[38] E. S. Lubana, J. Brehmer, P. de Haan, and T. Cohen. Fomo rewards: Can we cast foundation models as reward functions?, 2023.

[39] C. Luo, M. He, Z. Zeng, and C. Sun. Text-aware diffusion for policy learning, 2024.

[40] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman. Liv: Language-image representations and rewards for robotic control, 2023.

[41] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar. Eureka: Human-level reward design via coding large language models, 2024.

[42] P. Mazzaglia, T. Verbelen, B. Dhoedt, A. Lacoste, and S. Rajeswar. Choreographer: Learning and adapting skills in imagination. In *International Conference on Learning Representations*, 2023.

[43] OpenAI et al. Gpt-4 technical report, 2024.

[44] R. Bommasani et al. On the opportunities and risks of foundation models, 2022.

[45] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[46] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[47] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas. A generalist agent, 2022.

[48] J. Rocamonde, V. Montesinos, E. Nava, E. Perez, and D. Lindner. Vision-language models are zero-shot reward models for reinforcement learning. *arXiv preprint arXiv:2310.12921*, 2023.

[49] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[50] M. R. Samsami, A. Zholus, J. Rajendran, and S. Chandar. Mastering memory tasks with world models. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[52] D. Tarasov, V. Kurenkov, A. Nikulin, and S. Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning, 2023.

[53] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.

[54] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

[56] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023.

[57] Y. Wang, K. Li, X. Li, J. Yu, Y. He, G. Chen, B. Pei, R. Zheng, J. Xu, Z. Wang, Y. Shi, T. Jiang, S. Li, H. Zhang, Y. Huang, Y. Qiao, Y. Wang, and L. Wang. Internvideo2: Scaling video foundation models for multimodal video understanding, 2024.

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

[59] T. Xie, S. Zhao, C. H. Wu, Y. Liu, Q. Luo, V. Zhong, Y. Yang, and T. Yu. Text2reward: Reward shaping with language models for reinforcement learning, 2024.

[60] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, L. Kaelbling, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators, 2024.

[61] S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans. Foundation models for decision making: Problems, methods, and opportunities, 2023.

[62] D. Yarats, D. Brandfonbrener, H. Liu, M. Laskin, P. Abbeel, A. Lazaric, and L. Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning, 2022.

[63] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning, 2021.

[64] L. Yu, J. Lezama, N. B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, V. Birodkar, A. Gupta, X. Gu, A. G. Hauptmann, B. Gong, M.-H. Yang, I. Essa, D. A. Ross, and L. Jiang. Language model beats diffusion – tokenizer is key to visual generation, 2024.

[65] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training, 2023.

[66] Y. Zhang, E. Sui, and S. Yeung-Levy. Connect, collapse, corrupt: Learning cross-modal tasks with uni-modal data, 2024.

[67] L. Zhao, N. B. Gundavarapu, L. Yuan, H. Zhou, S. Yan, J. J. Sun, L. Friedman, R. Qian, T. Weyand, Y. Zhao, R. Hornung, F. Schroff, M.-H. Yang, D. A. Ross, H. Wang, H. Adam, M. Sirotenko, T. Liu, and B. Gong. Videoprism: A foundational visual encoder for video understanding, 2024.

[68] Y. Zhou, R. Zhang, C. Chen, C. Li, C. Tensmeyer, T. Yu, J. Gu, J. Xu, and T. Sun. Lafite: Towards language-free training for text-to-image generation, 2022.

 https://doi.org/10.52202/079017-0865

A Extended Related Work

[Linked to Section 2]

World models. Recent research has focused on the question of how to learn world models from large-scale video datasets [37, 60]. In [6], they leverage a latent action representation, but their work is mostly focussed on 2D platform videogames or simple robotic actions. In [14], they use frame-by-frame video prediction as a way to provide rewards for RL. DynaLang [36] studies the incorporation of language prediction as part of the world model, to train multimodal world models also from datasets without actions or rewards. The representation in DynaLang is shared in the world model between vision and language, while for GenRL, the world model representation is trained on vision-only data and connected-aligned to the multimodal foundation representation.

Foundation models for actions. Few cases of foundation models for embodied domains have been developed. Notable mentions are GATO [47], a large-scale behavior cloning agent, trained on 604 tasks. VPT [2] a large-scale model trained on Minecraft data, using human-expert labeled trajectories. The model learns strong behavioral priors by behavior cloning which can be fine-tuned using RL. STEVE-1 [35] connects VPT's behavioral prior with the MineCLIP model representation [15], using the unCLIP approach [46]. RT-X [12] are large-scale transformer models trained on expert robotics dataset, sharing a common action space (end-effector pose) across different embodiments.

B Implementation details

[Linked to Section 3]

Actor-critic. Rewards can be maximized over time in imagination in a RL fashion, using actor-critic models of the form:

$$\text{Actor:} \quad \pi_\theta(a_t|s_t), \qquad \text{Critic:} \quad v_\theta(R_t^\lambda|s_t), \quad \text{where} \quad R_t^\lambda = r_t + \gamma[(1-\lambda v_{t+1}) + \lambda R_{t+1}^\lambda]$$

For the actor-critic, we follow the implementation advances proposed in DreamerV3 [26] (version 1 of the paper, dated January 2023), such as using a two-hot distribution for learning the critic network and scaling returns in the actor loss.

When computing the reward r_{GenRL}, we use the mode of the distribution for the target $s_{t+1}^{\text{task}} \sim p_\psi(s_{t+1}|e_{\text{task}})$ to improve stability.

Hyperparameters. For the hyperparameters, we follow DreamerV3 [26] (version 1 of the paper, dated January 2023). Differences from the default hyperparameters or model size choices are illustrated in Table 2. For instance, a main difference is that we use difference batch sizes/lengths for training the MFWM and the actor-critic as these two stages are now independent from each other.

The connector network uses the same hyperparameters and architecture as the sequential dynamics of the world model. The aligner network employs a small U-Net, with a bottleneck that is half the size of the embedding representation. The embedding representation is 768-dimensional. Further details can be found in our accompanying code implementation.

Name	**Value**
Multimodal Foundation World Model	
Batch size	48
Sequence length	48
GRU recurrent units	1024
CNN multiplier	48
Dense hidden units	1024
MLP layers	4
Actor-Critic	
Batch size	32
Sequence length	32

Table 2: World model and actor-critic hyperparameters.

C Tasks

[Linked to Section 4]

Stickman environment. The Stickman environment is based on the Walker environment from the *dm_control* suite. We designed the Stickman environment to explore tasks that require upper body limbs (e.g. boxing, doing a handstand) without the complexity of training a humanoid (which requires a significantly larger amount of data to be solved [63]). The number of joints is increased by 4: 2 joints per arm, one is for the shoulder, the other for the elbow. The total number of joints is 10. The action space is normalized to be in [-1,1] as all *dm_control* tasks. The robot also presents a head, to resemble a humanoid.

Prompts and scores. We present the list of tasks employed, along with the language prompts used for specifying the task, in Table 3. For the newly introduced tasks, the goal can be easily inferred by reading the task's name or its prompt. For the 'flipping' tasks, we consider flips both in the forward direction and backward direction, as the VLM struggles to distinguish directions. The reward functions used to evaluate the agent's score can be found in our open-source code.

The prompts we use have been fine-tuned for the InternVideo2 model [57]. However, we found that they mostly improved performance for the SigLIP model too [65]. One common observation is that these models are generally biased towards human actions. Thus, specifying the embodiment in the prompt is sometimes helpful, e.g. 'spider running fast' or 'running like a quadruped'. Another observation is that for some behaviors the agent can produce very different styles, e.g. the agent can be walking in a slow or fast way, or in a more or less composed manner. Specifying words like 'fast' or 'clean' helps clarifying what kind of behavior is expected.

Table 3: Task and prompt used for each task

Task	Prompt	Specialized agent score	Random agent score
quadruped run	spider running fast	930	10
quadruped walk	spider walking fast	960	10
quadruped stand	spider standing	990	15
quadruped jump	spider jumping	875	15
quadruped two legs	on two legs	875	14
quadruped lie down	lying down	965	750
cheetah run	running like a quadruped	890	9
cheetah standing	standing like a human	930	5
cheetah lying down	lying down	920	430
stickman walk	robot walk fast clean	960	35
stickman run	robot run fast clean	830	25
stickman stand	standing	970	70
stickman flipping	doing flips	790	45
stickman one foot	stand on one foot	865	20
stickman high kick	stand up and kick	920	55
stickman lying down	lying down horizontally	965	380
stickman sit knees	praying	966	40
stickman lunge pose	lunge pose	950	100
stickman headstand	headstand	955	180
stickman boxing	punch	920	80
stickman hands up	standing with the hands up	830	5
walker walk	walk fast clean	960	45
walker run	run fast clean	770	30
walker stand	standing up straight	970	150
walker flipping	doing backflips	720	20
walker one foot	stand on one foot	955	20
walker high kick	stand up and kick	960	25
walker lying down	lying down horizontally	975	170
walker sit knees	praying	945	100
walker lunge pose	lunge pose	945	150
kitchen microwave	opening the microwave fully open	1	0
kitchen light	activate the light	1	0
kitchen burner	the burner becomes red	1	0
kitchen slide	slide cabinet above the knobs	1	0

https://doi.org/10.52202/079017-0865

D Experiments settings

[Linked to Section 4]

Baselines. In order to implement performant model-free offline RL baselines we adopt the findings of [52] and [7], adopting larger deeper networks and layer normalization. Inputs are 64x64x3 RGB images. We use a frame stack of 3. The encoder architecture is adapted from the DrQ-v2 encoder [63]. We did find augmentations on the images, e.g. random shifts, to hurt performance.

The WM-CLIP baseline learns a “reversed connector" from the world model representation to the VLM representation (GenRL does the opposite). The “reversed connector", given the latent state corresponding to a certain observation, predicts the corresponding embedding. Formally:

$$\text{Reversed connector:} \quad \hat{e}_t^{(v)} = f_\psi(s_t, h_t) \qquad \mathcal{L}_{\text{rev\_conn}} = \|e^{(v)} - \hat{e}_t^{(v)}\|_2^2$$

After training the reversed connector, visual embeddings can be inferred from latent states. For policy learning, rewards are computed using the cosine similarity between embeddings inferred from imagined latent states and the prompts' embedding.

The reversed connector is implemented as a 4-layer MLP, with hidden size 1024. For fair comparison, we adopt the same world model for WM-CLIP and GenRL. For WM-CLIP we pre-train the additional reversed connector, while for GenRL the connector and aligner.

Offline RL. For each task, training model-free agents (IQL, TD3, TD3+BC) requires re-training the full agent (visual encoder, actor, critic) on the entire dataset, from scratch, while training model-based agents (GenRL, WM-CLIP) requires training the model once for each domain and then training an actor-critic for each task. Moreover, for training the actor-critic in GenRL, we only use 50k gradient steps, as the policy converges significantly faster than for the other methods.

Datasets composition. We present the datasets' composition in Table 4.

Table 4: Datasets composition.

Domain	$\sim$ num of observations	Subset	Subcount
walker	2.5M	walker run	500k
		walker walk	500k
		walker stand	500k
		walker expl	1M
cheetah	1.8M	cheetah run	1M
		cheetah expl	820k
quadruped	2.5M	quadruped expl	1M
		quadruped run	500k
		quadruped stand	500k
		quadruped walk	500k
kitchen	3.6M	kitchen slide	700k
		kitchen light	700k
		kitchen bottom burner	700k
		kitchen microwave	700k
		kitchen expl	800k
stickman	2.5M	stickman stand	500k
		stickman walk	500k
		stickman expl	1M
		stickman run	500k
minecraft	4M	-	-

Compute resources. We use a cluster of V100 with 16GB of VRAM for all our experiments. To enable efficient training, image and video-CLIP embeddings are computed in advance and stored with the datasets. Training the MFWM for 500k gradient steps takes $\sim$ 5 days. After pre-training the MFWM, training the actor-critic for a prompt for 50k gradient steps takes less than 5 hours. In data-free mode, it takes less than 3 hours. In both cases, convergence normally arrives after 10k gradient steps, but we keep training. Model-free baselines take around 7 hours to train for 500k gradient steps.

On a single GPU, model-free RL is faster to train for a small number of runs. GenRL starts becoming advantageous when using the world model for training for more than 60 runs (which is often the case, considering the number of runs = N seeds x M tasks per domain). When adopting the data-free policy learning strategy, GenRL doesn't rely on the dataset at all. This halves the time required for training, as there are no data transfers between the CPU (where the dataset is normally loaded) and the GPU for training.

E Temporal alignment ablation

[Linked to Section 3]

In Figure 8, we investigate the impact of our best-matching trajectory strategy for temporally aligning imaginary latent states to target states for GenRL's reward computation.

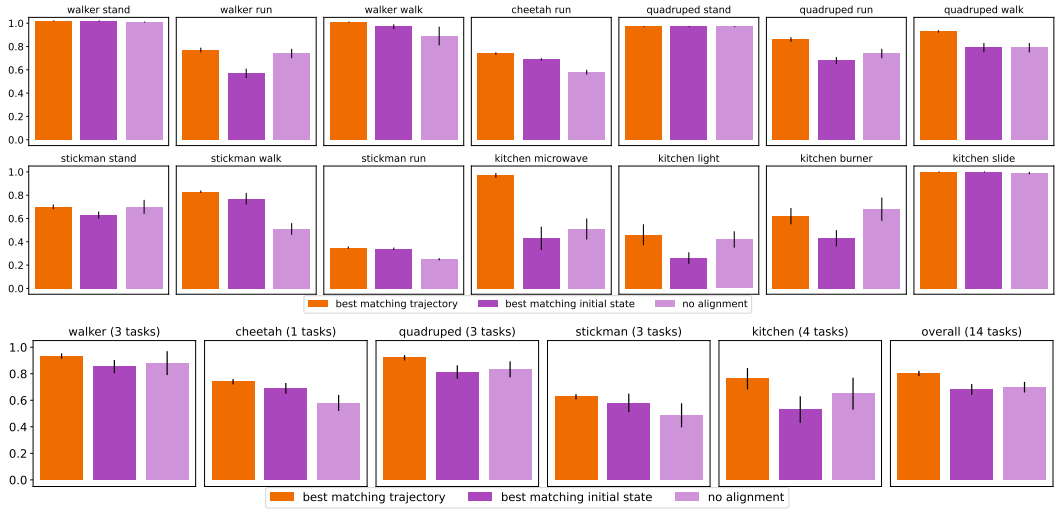


Figure 8: *Temporal alignment ablation.* We analyze the impact of temporal alignment in our proposed RL objective for matching sequential targets. Results averaged over 10 seeds.

F Aligner model ablation study

[Linked to Section 3]

To establish the importance of the aligner network, in Table 5 we report the results of additional ablations:

- **GenRL - no aligner:** this is an ablation of GenRL where the the language prompt's embedding is directly fed into the connector, rather than processing it first with the aligner;
- **TD3-V and WM-CLIP-V + aligner:** for these ablations, we first process the language prompt's embedding using GenRL's pre-trained aligner. Then, we use it to compute the cosine similarity for the reward function, as for the original baselines.

Table 5: Aligner ablation study.

	GenRL - no aligner	GenRL	WM-CLIP-V	WM-CLIP-V + aligner	TD3-V	TD3-V + aligner
cheetah	0.32 ± 0.02	0.93 ± 0.01	0.82 ± 0.17	0.84 ± 0.02	0.31 ± 0.09	0.77 ± 0.04
stickman	0.09 ± 0.01	0.66 ± 0.01	0.54 ± 0.03	0.51 ± 0.03	0.38 ± 0.02	0.38 ± 0.02
walker	0.19 ± 0.01	0.75 ± 0.01	0.70 ± 0.02	0.74 ± 0.01	0.56 ± 0.04	0.48 ± 0.04
kitchen	0.25 ± 0.00	0.76 ± 0.08	0.71 ± 0.14	0.84 ± 0.09	0.32 ± 0.16	0.27 ± 0.10
quadruped	0.17 ± 0.02	0.91 ± 0.02	0.81 ± 0.04	0.76 ± 0.05	0.32 ± 0.04	0.33 ± 0.04
overall	0.17 ± 0.00	0.76 ± 0.01	0.67 ± 0.03	0.68 ± 0.02	0.40 ± 0.02	0.42 ± 0.02

We can observe that: i) the aligner mechanism is crucial in GenRL's functioning. ii) processing the language embedding in the reward function of the WM-CLIP-V and TD3-V baselines changes performance on some tasks (performance per domain varies). However, using the aligner provides no advantage overall.

We believe the aligner is very important in GenRL because its output, the processed language embedding, is fed to another network, the connector. If the language embeddings were not processed by the aligner, they would have been too different from the embeddings used to train the connector, which are the visual embeddings.

Instead, for the baselines, we process the language embedding with the aligner and then use it to compute a similarity score with the visual embeddings. This overall renders very similar performance to no aligner processing, hinting that the aligner network doesn't improve the cosine similarity signal. At the same time, this also suggests that the aligner network doesn't hurt the generality of the VLM's embeddings, as the cosine similarity after processing the embedding provides a similarly useful signal as before processing.

https://doi.org/10.52202/079017-0865

G Comparison with LIV

[Linked to Section 4]

We also tested all baselines with the LIV's representation [40] in the Kitchen tasks. We used LIV's open-source code to download and instantiate the model. Note that the available model is the general pre-trained model, not the one fine-tuned for the Kitchen environment.

LIV's results confirm the original paper's claims (see Appendix G4) that the representation does not work well for vision-language rewards without fine-tuning on domain-specific vision-language pairs (which are unavailable in our settings, as we use no language annotations).

Table 6: Comparison with LIV [40] on kitchen tasks.

	IQL + LIV	TD3+BC + LIV	TD3 + LIV	WM-CLIP + LIV	GenRL
kitchen microwave	0.03 ± 0.03	0.0 ± 0.0	0.2 ± 0.16	0.0 ± 0.0	0.97 ± 0.02
kitchen light	0.53 ± 0.24	0.05 ± 0.04	0.0 ± 0.0	0.0 ± 0.0	0.46 ± 0.09
kitchen burner	0.2 ± 0.08	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.62 ± 0.07
kitchen slide	0.03 ± 0.03	0.03 ± 0.03	0.0 ± 0.0	0.67 ± 0.27	1.0 ± 0.0
overall	0.20 ± 0.11	0.02 ± 0.02	0.05 ± 0.07	0.17 ± 0.12	0.76 ± 0.08

H Extended Discussion on Data-free Policy Learning

[Linked to Section 4]

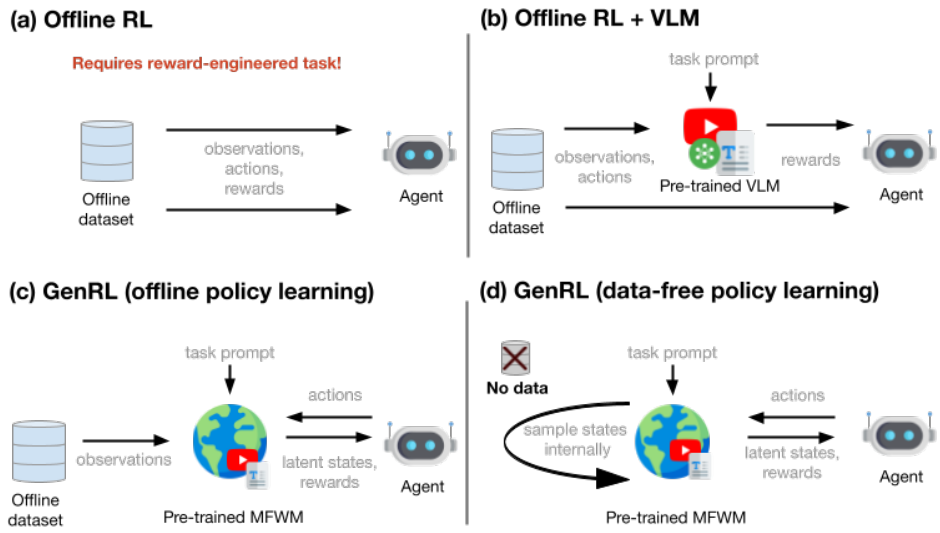


Figure 9: Representing how the different agents process the data for policy learning. By removing data dependencies on actions (on-policy learning in the model's imagination), rewards (computed using only the prompt and latent states, using Eq. 3), and observations (by sampling latent states within the model), GenRL agents can be adapted for new tasks in a data-free fashion.

Offline RL methods (Fig. 9a) learn from offline datasets of observations, actions and rewards.

Offline RL methods (Fig. 9b), combined with VLMs, can learn to perform tasks zero-shot from new prompts, but they need to sample observations and actions from the dataset for computing rewards and for policy learning.

GenRL (Fig. 9c) needs to sample (sequences of) observations from the dataset, to infer the initial latent states for learning in imagination. Afterwards, rewards can be computed on the imagined latent sequences, enabling policy learning.

Data-free GenRL (Fig. 9d), samples the initial latent states internally by combining: (i) random samples of the latent space, (ii) randomly sampled embeddings, which are mapped to "actual embeddings" using the aligner, and turned into latent states, by the connector. Thus, policy learning requires no data sampling at all.

Initial states distribution. Uniform sampling from the latent space of the world model often results in meaningless latent states. Additionally, the sequential dynamics model of the MFWM, using a GRU, requires some 'warmup' steps to discern dynamic environmental attributes, such as velocities.

To address these issues we perform two operations. First, we combine uniformly sampled states from the discrete latent spaces with states generated by randomly sampling the connector model, as sequences generated by the connector tend to have a more coherent structure than random uniform samples. Second, we perform a rollout of five steps using a mix of actions from the trained policy and random actions. This leads to a varied distribution of states, containing dynamic information, which we use as the initial states for the learning in imagination process.

I Scaling to complex observations

[Linked to Section 5]

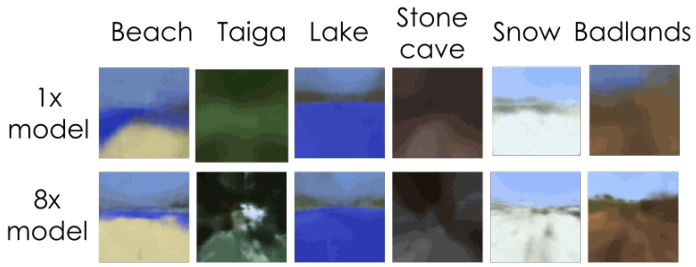


Figure 10: Decoded language prompts in Minecraft

Generalist embodied agents should be able to scale to open-ended learning settings. Using GenRL, we explored this by training an agent in the Minecraft environment using a small dataset collected by a DreamerV3 agent [26]. Note that GenRL is also trained using similar experimental settings as DreamerV3, e.g. setting up the environment in the same way. The primary challenge we found was the model's difficulty in reconstructing complex observations in this open-ended environment.

Reconstructing complex observations is a common issue with world models [10]. To overcome this limitation, while keeping the method unaltered, we attempted to scale up the number of parameters of the MFWM. Qualitative reconstruction results are presented in Figure 10. We observe that the agent is able to identify different biomes from language, even with the smaller size of the model. However, the reconstructions are significantly blurrier compared to the other environments we analyzed. When using a larger model, the reconstructions gain some details but the results still highlight the difficulty of the model in providing accurate targets from prompts.

While this might not be an issue for simple high-level tasks, e.g. 'navigate to a beach', inferring unclear targets might make it difficult to perform more precise actions, e.g. 'attack a zombie'. Future research should aim to address this issue, for instance, by improving our simple GRU-based architecture, leveraging transformers or diffusion models to improve the quality of the representation [30, 3].

J Additional limitations

[Linked to Section 5]

Pre-training data requirements. As we developed our framework, we observed that, in order to solve more complex tasks, the agent requires some expert data/demonstration of the complex behavior. We observed and analyzed this aspect in the experiments of Section 4.3. We believe this limitation is, to some extent, inevitable, as data-driven AI agents need to observe complex behaviors during training in order to be able to replicate them.

In this work, we used some exploration data, obtained using an exploration agent (Plan2Explore), and some task-specific data, collected using an expert RL agent (DreamerV3). Alternatively, one could investigate the adoption of a small set of demonstrations.

Precise manipulation skills. Adding additional tasks to the Kitchen environment, for testing multi-task generalization, showed to be harder than for the other domains. This is due to the difficulties of exploring meaningful behaviors in manipulation environments.

We are able to use GenRL to retrieve the four tasks present in the dataset and we are generally able to achieve similar tasks, such as "reach the microwave". From decoding the language prompts, we also see that GenRL can sensibly decode prompts such as "moving to the left" or "staying still", or more 'unusual' prompts such as

https://doi.org/10.52202/079017-0865

“swan pose" (where the robot arm would imitate the pose of a swan). While these prompts show that the system works well in this environment, none of these tasks would be particularly useful.

When we prompt the system with more interesting tasks that are out of the distribution, the agent generally struggles. For instance, if we ask to open the double door cabinet on the top left, the agent may reach for it but will not open it. The reason behind this is that it probably never opened that cabinet in the training dataset and thus it’s impossible to reproduce such behavior in an offline manner. The same issues are most likely present with all the baselines as well, as they all use the same dataset and foundation models for training.

For the locomotion domains, instead, the exploration data is varied enough that new tasks, outside of the tasks present in the training dataset, are enabled through the exploration data/

K Detailed results

[Linked to Section 4]

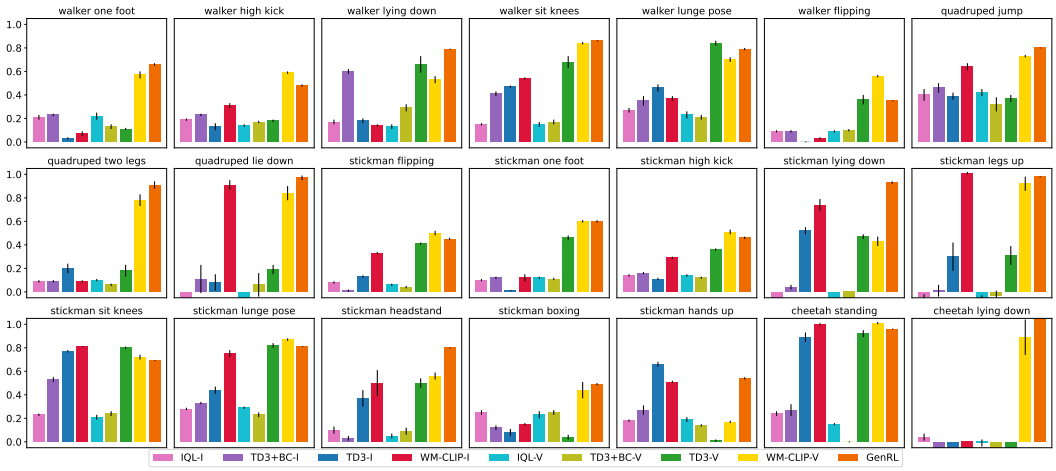


Figure 11: *Multi-task generalization detailed results.* Results averaged over 10 seeds.

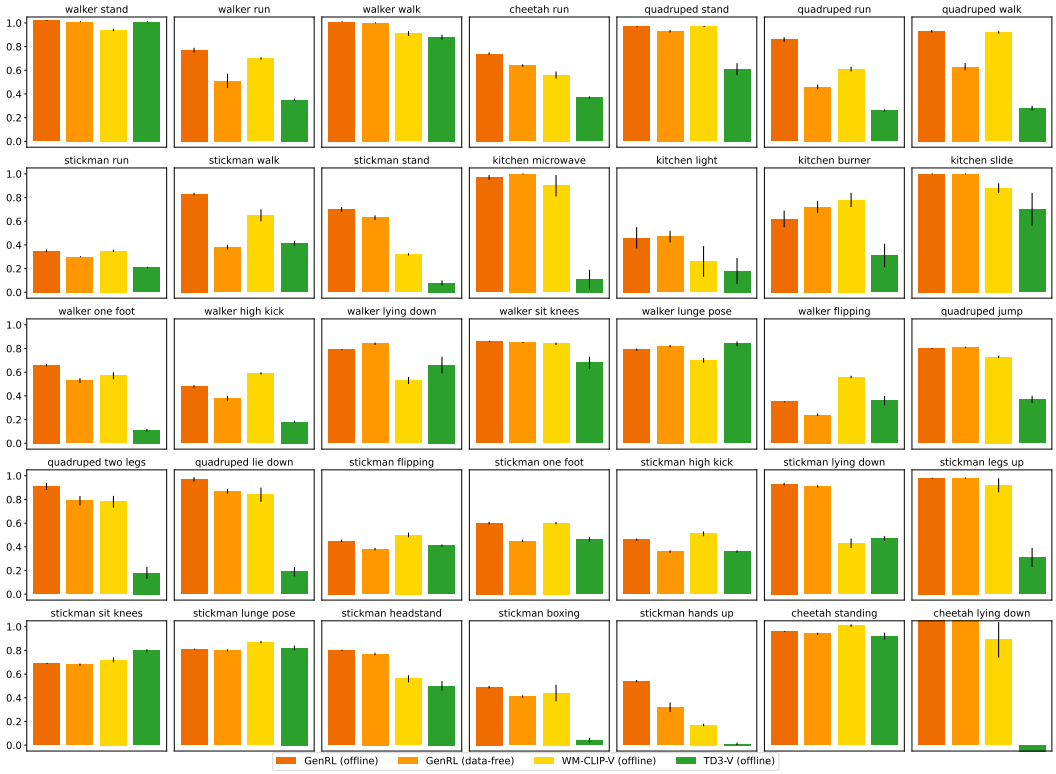


Figure 12: *Data-free RL detailed results*. Results averaged over 10 seeds.

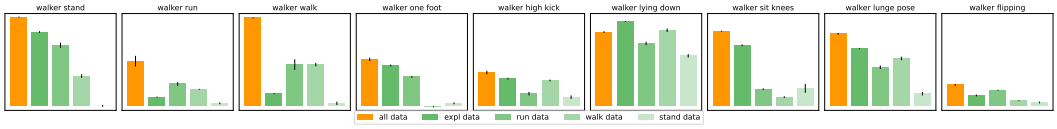


Figure 13: *Training data distribution detailed results*. Results averaged over 10 seeds.

https://doi.org/10.52202/079017-0865

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All the claims presented in the abstract and introduction are further discussed while presenting the method and the results. The experiments support the claims made.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of the approach are discussed in the experiments and analysis sections of the paper. All potential limitations are also restated and summarized in the final discussion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not provide theoretical results. The new objectives and models introduced are evaluated empirically.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The details to reproduce the results are provided in the experiments section and in the supplementary material (appendix). To further facilitate reproduction of our work, the code and data will be made publicly available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: See the project website for access to data and code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

https://doi.org/10.52202/079017-0865

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The details to reproduce the results are provided in the experiments section and in the supplementary material (appendix). To further facilitate reproduction of our work, the code and data will be made publicly available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide standard errors in tables and error bars in the plots.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Information about the compute exploited is provide in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work does not deviate from the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The work has no societal impact at the current stage.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

https://doi.org/10.52202/079017-0865

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the corresponding papers for the models and the environments used in our experiments, e.g. SigLIP, InternVideo2, DmControl.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: New assets will be released and documented at a later date.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-0865