UNIVERSAL EMBODIED INTELLIGENCE: LEARNING FROM CROWD, RECOGNIZING THE WORLD, AND REINFORCED WITH EXPERIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

The interactive artificial intelligence in the motion control field is an interesting topic, especially when universal knowledge adaptive to multiple task and universal environments is wanted. Although there are increasing efforts on Reinforcement learning (RL) studies with the assistance of transformers, it might subject to the limitation of the offline training pipeline, in which the exploration and generalization ability is prohibited. Motivated by the cognitive and behavioral psychology, such agent should have the ability to learn from others, recognize the world, and practice itself based its own experience. In this study, we propose the framework of Online Decision MetaMorphFormer (ODM) which attempts to achieve the above learning modes, with a unified model architecture to both highlight its own body perception and produce action and observation predictions. ODM can be applied on any arbitrary agent with a multi-joint body, located in different environments, trained with different type of tasks. Large-scale pretrained dataset are used to warmup ODM while the targeted environment continues to reinforce the universal policy. Substantial interactive experiments as well as few-shot and zero-shot tests in unseen environments and never-experienced tasks verify ODM's performance, and generalization ability. Our study shed some lights on research of general artificial intelligence on the embodied and cognitive field studies.

1 INTRODUCTION

Research of embodied intelligence focus on the learning of control policy given the agent with some morphology (joints, limbs, motion capabilities), while it has always been a topic whether the control policy should be more general or specific. As the improvement of large-scale data technology and cloud computing ability, the idea of artificial general intelligence (AGI) has received substantial interest (Reed et al., 2022). Accordingly, a natural motivation is to develop a universal control policy for different morphological agents and easy adaptive to different scenes. It is argued that such a smart agent could be able to identify its 'active self' by recognizing the egocentric, proprioceptive perception, react with exteroceptive observations and have the perception of world forward model (Hoffmann & Pfeifer, 2012). However, there is seldom such machine learning framework by so far although some previous studies have similar attempts in one or several aspects.

In the aspect of model architecture, Multi-Layer Perceptron (MLP) is traditionally employed for Reinforcement Learning(RL) policies which could have reasonable performance when learing within a specific environment. However, upon recent progress on large-scale NLP research, transformer (Vaswani et al., 2017) has received substantial attention which provides a uniformed solution of sequence modeling with great accuracy and generalization ability. Correspondingly, transformer-based RL (Chen et al., 2021; Lee et al., 2022; Janner et al., 2021; Zheng et al., 2022; Xu et al., 2022) has been proposed by transforming RL trajectories as a large sequence model, and obtain comparable performance with expert agents trained by traditional RL. Such methodologies capture the time dependency underlying the sequential modeling, but lacks of agent embodiment consideration. In contrast, there are some other studies which focus on the agent's morphology information, including GNN-based RL which modeling agent joints as a kinematics tree graph (Huang et al., 2020), Amorpheus which encodes a transformer for each body joint, and MetaMorph (Gupta et al., 2022) which intuitively use transformer to encode the body morphology as a sequence of joint properties,

instead of time sequence dependency. Based on these efforts, a time-morphology transformer might be a better architecture which take advantage of both time and morphology dependency for a general purpose policy.

It is also interesting to perceive the training paradigm of such generalist agent when considering the natural world mechanism. Most transformer-based methods belong to offline learning since training transformer require enormous data. In the RL field, this can be analogous to learning from others, including teachers, peers or even someone with worse skills. Such methods are closely connected with imitation learning or behavior cloning. On the other hand, on-policy RL such as Proximal Policy Optimization (PPO (Schulman et al., 2017) is able to learn the policy therefore has good adaptive ability to environment, but might subject to large performance variations. Such methodology is an example of behavioral psychology in which agent improve its skill by actual practice. There are also model-based RL studies in which the agent makes decision based on the perception of how the real world evolves (Ha & Schmidhuber, 2018; Wu et al., 2022). It is reasonable to believe that an embodied intelligence agent should have the above three learning paradigm simultaneously.

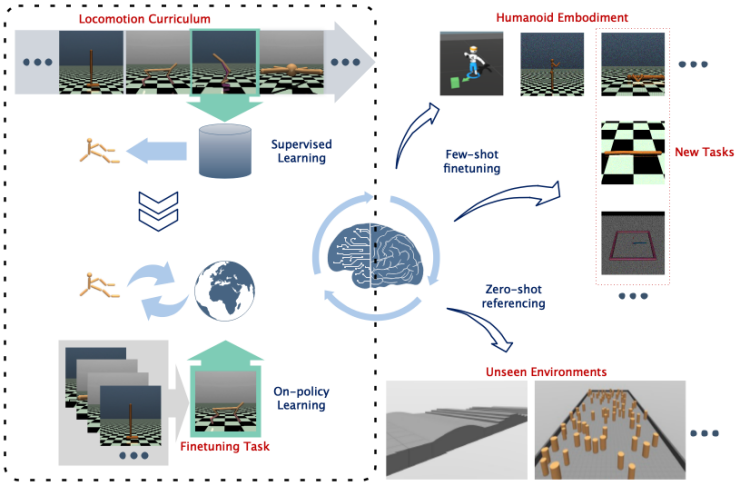

Figure 1: Application pipeline of ODM.

In this work, we attempt to capture the above considerations by a framework called **O**nline **D**ecision **M**etamorphformer (ODM). ODM aims to study the general knowledge of embodied control across different body shapes, environments and tasks, as indicated in Figure 1. The model architecture contains the universal backbone and the task-specific modules. The morphology-aware auto-encoder is employed to encode states and actions and map to the same latent space. A casual transformer is then applied on this sequence problem to learn the general control knowledge, but with a morphological prompt to capture each body shape's characteristic. We first pretrain this model with a curriculum learning, by learning the demonstration from the easiest to the hardest task, from the expert to low-level players. The world model prediction is added as an auxiliary loss. The same architecture can then be applied in the online finetuning with the very last action trained with on-policy reinforce, and another value prediction head is added and learned by bootstrap method. During the test, we are able to test ODM with all training environments, transfer the policy to different body shapes, adaptive to unseen environments and accommodate with new types of tasks (e.g. from locomotion to reaching, target capturing or escaping from obstacles.). Code, result and video examples can be found on the website `https://baimaxishi.github.io/`.

Main contributions of this paper include:

- We design a unified model architecture to encode time and morphology dependency simultaneously which bridges sequential decision making with embodiment intelligence.
- We propose a training paradigm which mimic the process of natural intelligence emerging, including learning from others, boost with practices, and recognize the world.
- We train and test our framework with agent in eight different body shapes, different environment terrain and different task types. These comprehensive analysis verifies the general knowledge of motion control learned by us.

2 PRELIMINARIES

2.1 REINFORCEMENT LEARNING

Reinforcement Learning (RL) generally studies on *Markov Decision Process* (MDP). At time step t, the system state $s_t \in \mathcal{R}^{n_s}$ is affected by an action $a_t \in \mathcal{R}^{n_a}$ from the agent, generating a scalar reward $r_t \in \mathcal{R}^1$, as well as the next state $s_{t+1} \in \mathcal{P}(s_t, a_t)$. The agent generate its action from its policy $\pi(a_t|s_t)$ and push the environment stage forward. This interactive process produces the episode sequence:

$$\tau_{0:T} = \{s_0, a_0, r_0, s_1, a_1, r_1, \cdots, s_T, a_T, r_T\} \tag{1}$$

in which T means the episode ends or reaching the maximum time length. RL then solves the sequential decision making problem by finding π such that $\max_\pi \mathbb{E}_{\tau\sim\pi}(R)$ in which the episode return defined as

$$R := \sum_{t=0}^{T} \gamma^t r_t \tag{2}$$

with $\gamma \in [0, 1)$ as the discounted factor.

2.2 PROXIMAL POLICY OPTIMIZATION (PPO)

With the a critic which estimate a state value function $V(s)$, the advantage can be estimated by generalized advantage estimation by traversing the episode backward (Schulman et al., 2016)

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{3}$$

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{4}$$

Then (an actor-critic style) Proximal Policy Optimization (PPO) algorithm learns the policy by maximizing the following surrogate objective in the policy-gradient way:

$$L^{\text{Actor}} = \text{CLIP}(\text{E}_t \frac{\pi_{\theta_{\text{old}}}(a_t|s_t)}{\pi_\theta(a_t|s_t)} \hat{A}_t - \beta \text{KL}[\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)]) \tag{5}$$

in which the CLIP function means clipping the object by $[1-\epsilon, 1+\epsilon]\hat{A}_t$, and KL denotes the famous K-L divergence. The critic loss can calculated the classical Bellman function

$$L^{\text{Critic}} = \mathbb{E}_{s\sim d^\pi}[(r_t + \gamma(V_{\theta_{\text{old}}}(s_{t+1}) - V_\theta(s_t))^2]$$

A PPO policy update is then conducted by minimizing the objective during each iteration:

$$L^{\text{PPO}} = -\eta_A L^{\text{Actor}} + \eta_C L^{\text{Critic}}$$

2.3 ATTENTION-BASED ENCODING

The attention mechanism (Vaswani et al., 2017) has the follow form:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{6}$$

in which Q, K, V denote query, key and value. Given a stacked time sequence variable $x \in \mathbb{R}^{T,e}$ with T as the time length and e as the embedding dimension, a simple version of time sequence encoder is then formulated by applying Eq 6 by feeding all Q, K, V with x, as an implementation of self-attention

$$\text{Encode}_T(x) = \text{Attention}(\text{Q=}x, \text{K=}x, \text{V=}x) \in \mathbb{R}^{T,e} \tag{7}$$

in which e is the embedding dimension. Similarly, Given a stacked 'pose' embedding variable $p \in \mathbb{R}^{K,e}$ with K as the number of joints, a morphology-aware auto-encoder instead applies the self-attention mechanism on this joint sequence

$$\text{Encode}_M(x) = \text{Attention}(\text{Q=}p, \text{K=}p, \text{V=}p) \in \mathbb{R}^{K,e} \tag{8}$$

Within each type of encoder, timestep or joint position info can be considered by a lookup embedding and a adding-on operation to the latent vector. More details can be referred to the MetaMorph work (Gupta et al., 2022).

3 METHOD

3.1 EMBODIED VIEWPOINT OF STATES AND ACTIONS

Following the idea of Gupta et al. (2022), here we redefine some conventional RL notations in a more 'embodied way', for any arbitrary agent with a multi-joint body. For an agent body with K joints and $K+1$ bodyparts (or limbs), we assume each joint has m degree of freedom (DOF) to execute actions (e.g. torque); and each joint could feel itself status, i.e. proprioceptive observations (e.g. angular, angular velocity), with n dimensions. The global sensory of agent also provide x-dimensional exteroceptive observations (e.g. position, velocity), independent of any specific joint. Masks of observations and actions might be necessarily to allow joint variation (e.g. DOF of human forearm should be smaller than upper-arm due to their physical connection). State and action of our RL system are then re-defined as:

State: the state $s := [o^{pro}, o^{ext}]$ in which $o^{pro} \in \mathbb{R}^{K,n}$ is the proprioceptive observations and $o^{ext} \in \mathbb{R}^{x}$ is the exteroceptive observation. A binary mask $\mathcal{M}_s^{pro} \in \mathbb{R}^{K,n}$ is employed to mask out some impractical observations from o^{pro}. [1].

Action: the action $a \in \mathbb{R}^{K,m}$, with a binary mask $\mathcal{M}_a \in \mathbb{R}^{K,m}$ employed to mask out some impractical motions from a. [2].

3.2 MODEL ARCHITECTURE

Our ODM model structure contains a unified backbone and task-specific modules. Since our model deals with state and action variables in continuous spaces, the task-specific modules manage to map these original variables into a uniform-dimensional latent variable space, as the embedding; and map the predicted latent variable back to their original space, i.e., the projectors. The backbone has a two-directional, morphology-time transformer structure, including auto-encoders which encodes the pose embedding information, and a casual transformer which deal with the time-dependent decision making process. Architecture details are specified in Fig 3.2.

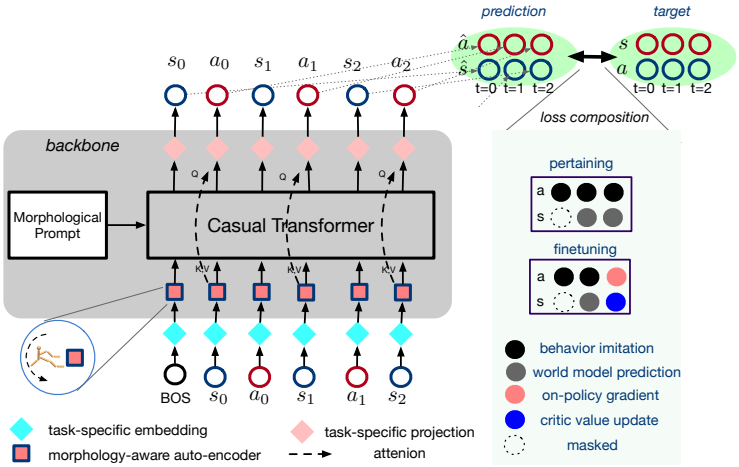

Figure 2: Model structure of ODM and its training paradigm.

Embedding: At each time t, observationss and actions are first embedded to the latent space with dimension e

$$o_t^e = \text{MLP}_o(o_t^{pro}) \in \mathbb{R}^{K,e}, \quad x_t^e = \text{MLP}_x(o_t^{ext}) \in \mathbb{R}^{e}, \quad a_t^e = \text{MLP}_a(a_t) \in \mathbb{R}^{K,e} \tag{9}$$

Morphology-aware Auto-Encoder: At each time t, we get the pose embedding vector p of agent by traversing and encoding the agent morphology by Eq 8:

$$o_t^p = \text{Encode}_M(o_t^e), \quad s_t^p = \text{MLP}_s([s_t^p, x_t^e]) \in \mathbb{R}^{e}, \quad a_t^p = \text{Encode}_M(a_t^e) \tag{10}$$

[1]Dimension of the original state defined in Section 2.1 satisfies $n_a = K * n - \sum \mathcal{M}_s + x$

[2]Dimension of the original action defined in Section 2.1 satisfies $n_a = K * m - \sum \mathcal{M}_a$

Casual Transformer: To capture the agent morphology difference, we apply the prompt technique as in (Xu et al., 2022), but embedding the morphology specification parameters instead of imitations

$$\text{Prompt} = \text{Embed}(K, n, m, x) \tag{11}$$

The casual transformer then translates the prompt and the input sequence into the output sequence

$$\text{output} = \text{Encode}_T(\text{Prompt}, \text{input}) \tag{12}$$

$$\text{input} := \{\text{BOS}, s_0^p, \quad a_0^p, s_1^p, \quad \cdots, \cdots, \quad a_{t-1}^p, s_t^p\} \tag{13}$$

$$\text{output} := \{\hat{s}_0^p, \hat{a}_0^p, \quad \hat{s}_1^p, \hat{a}_2^p, \quad \cdots, \cdots, \quad \hat{s}_t^p, \hat{a}_t^p\} \tag{14}$$

with an additional forward casual time mask. Detailed structure is inherited from GPT2, a decoder-only structure as in (Chen et al., 2021; Janner et al., 2021; Zheng et al., 2022). For practical consideration, input and output sequences are truncated by a window length T_w, with padding time mask for episodes shorter than T_w. To emphasize the instant impact, we further conduct a multi-head attention by querying the target variable and marking the input variable as key and value:

$$\hat{a}_t^p += \text{Attention}(\text{Q}=\hat{a}_t^p, \text{K}=s_t^p, \text{V}=s_t^p), \quad \hat{s}_t^p += \text{Attention}(\text{Q}=\hat{s}_t^p, \text{K}=a_{t-1}^p, \text{V}=a_{t-1}^p) \tag{15}$$

Projector: The task-specific projectors map the embedded variables to their original spaces:

$$\hat{a}_t = \text{Proj}_a(\hat{a}_t^p), \quad \hat{s}_t = \text{Proj}_s(\hat{s}_t^p), \quad \hat{V}_t = \text{Proj}_V(\hat{s}_t^p) \tag{16}$$

$\text{MLP}_o, \text{MLP}_x, \text{MLP}_a, \text{MLP}_s, \text{Proj}_a, \text{Proj}_s, \text{Proj}_V$ are all modeled as MLPs with LayerNorm and Relu between layers. More detailed configuration can be found in Appendix.

3.3 Training paradigm

In Section 3.2 we define a uniformed model structure which can be learned by both offline supervised training and on-policy RL. Correspondingly, our training paradigm can be divided into two phases, as specified in Algorithm 1.

Pretraining: To mimic the learning process of human children, we design a curriculum-based learning mechanism in which the training dataset transverses from the easiest task to the most complicated one. During each epoch, the current dataset is trained in a self-regressive manner with the loss containing three components:

$$L^{\text{prediction}} = \text{MSE}(\hat{s}_t, s_t^p), \quad L^{\text{imitation}} = \text{MSE}(\hat{a}_t, a_t^p), \quad L^{\text{pretrain}} = \eta_p L^{\text{prediction}} + \eta_i L^{\text{imitation}} \tag{17}$$

in which MSE denotes the mean-square-error between two continuous variables. $L^{\text{imitation}}$ are such named since it corresponding to the action imitation from others, while $L^{\text{prediction}}$ guild the agent to have a better scope of world model. Note the regression loss of the very first state is zero-out as specified in Figure 3.2, since it is meaningless to predict the very first state which is randomly decided.

Finetuning: Actor and critic can be obtained with the same model by using the estimated action and value as outputs:

$$\hat{a}_t \to \pi(s_t), \quad \hat{V}_t \to V(s_t) \tag{18}$$

with shared bottom parameters. This model can then be reinforced with PPO introduced in Section 2.2. By including some extent of L^{pretrain} as auxiliary tasks

$$L^{\text{finetune}} = \eta_1 L^{\text{PPO}} + \eta_2 L^{\text{pretrain}} \tag{19}$$

this finetuning can also be viewed as self-supervised or model-based RL.

4 EXPERIMENTS

4.1 Experiment Configurations

4.1.1 Environments and Tasks

We practice with enormous agents, environments and tasks, to validate the general knowledge studied by ODM. These scenes include:

Algorithm 1 ODM

Initialize θ
Pretraining:
    set num of epoch = 0
    SWITCH between 6 environments :
        activate grad of env-specific module of current env, freeze others
        REPEAT learning from pioneers of different expert levels
            self-regressive training on L^{pre} of a mini-batch
        increment num of epoch
    save the model checkpoint
Finetuning:
    load the model in the target environment
    REPEAT iterations:
        for each actor **do**
            run current policy π for T timesteps
            compute advantage estimates $\hat{A}_0, \cdots, \hat{A}_T$
        end for
        update θ with the surrogate L^{PPO} on a mini-batch
        stop when converges

- *Body shape*: including swimmer (3-joints, no foot), reacher (1-joint and one-side fixed), hopper (1 foot), halfcheetah (2-foot), walker2d (2-foot), ant (4-foot), and humanoid on the gym-mujoco platform [3]; walker (the agent called ragdoll has a realistic humanoid body) [4] on the unity platform (Juliani et al., 2018); and finally unimal, a mujoco-based enviroment which contain 100 different morphological agents (Gupta et al., 2021). Detailed morphology parameters are in Table B.
- *Environment*: flat terrain (FT), variable terrain (VT) or escaping from obstacles.
- *Task*: pure locomotion, standing-up attempt (humanoid), target reaching (reacher) or target reaching (walker).

4.1.2 BASELINES

- *Metamorph*: a morphology-aware encoding method based on the instant state information, trained with PPO Gupta et al. (2022), which represents the state-of-the-art of online learning algorithms on universal control policy.
- *DT*: As a state-of-the-art offline learning baseline, we implement the decision transformer with the expert action inference as in Lee et al. (2022) and deal with continuous space as in (Chen et al., 2021). We name it DT in the following sections for abbreviation.
- *PPO*: The classical on-policy RL algorithm Schulman et al. (2017). Code is cloned from stable-baseline3 in which PPO is in the actor-critic style.
- *Random*: The random control policy by sampling each action from uniform distribution from its bounds. This indicates the performance without any prior knowledge especially for the zero-shot case.

4.1.3 DEMONSTRATING PIONEERS

For purpose of pretraining, we collect offline data samples of hopper, halfcheetah, walker2d and ant from D4RL (Fu et al., 2020), as sources of pioneer demonstrations. For these environments, D4RL provide datasets sampled from agents of different skill levels, which corresponds to different learning pioneers in our framework, including *expert*, *medium-expert*, *medium*, *medium-replay* and *random*. We also train some baseline expert agents and using them to sample offline dataset on walker and unimal. These dataset contains more than 25 million data samples, with statistics details shown in Appendix, Table 6. Within each curriculum, we also rotate demonstrations from the above pioneers for training, as indicated in Algorithm 1.

[3] https://www.gymlibrary.dev/environments/mujoco/
[4] https://github.com/Unity-Technologies/ml-agents

4.2 EXPERIMENT RESULTS

Pretraining: Model is trained with datasets of hopper, halfcheetah, walker2d, ant, walker and unimal, from the easiest to the most complex. Figure 4.2 shows the loss plot. One can observe that the training loss successfully converges within each curriculum course; although its absolute value occasionally jumps to different levels because of the environment (and the teacher) switching. Validation set accuracy is also improved with walker and unimal as exhibition examples in the figure.

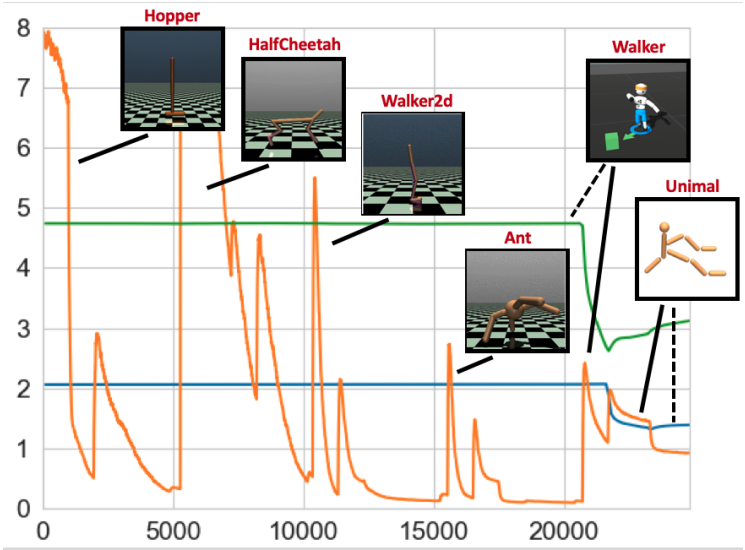

Figure 3: Time plot of pretraining performance. Orange: training loss. Green: validation mse of walker; Blue: validation mse of unimal.

Online Experiments: To make the online learning faster, we use 32 independent agents to sample the trajectory in parallel, with 1000 as the maximum episode steps. Experiment continues more than 1500 iterations after the performance converges. Figure 4.2 provides a quick snapshot of online performances. Comparing with ODM without pretraining, returns of ODM are higher at the very beginning, indicating knowledge from pretraining is helpful. As online learning continuous, the performance degrades slightly until finally grows up again, indicating the training criteria of offline supervising and the online RL still have some contradictions.

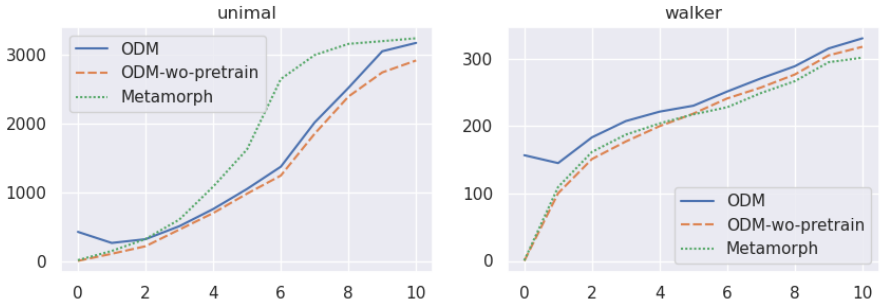

Figure 4: Comparison of averaged episode returns during online experiments. Left: unimal. Right: walker. Curves are smoothed and values are rescaled for better visualization.

During online testing, 100 independent episodes are sampled, and each is fed with a unique random seed. We examine the averaged episode return and episode length as our evaluation metrics . Table 1 shows the online testing performance. One can observe that our ODM outperforms or very closes to other baselines. It is worth noting that DT does not work for unimal, indicating the limitation of pure

offline method with changing agent body shapes. Performance in walker has substantial deviations since there is noise within the state forward.

Table 1: Averaged Episodic Performance in online locomotion environments.

Metric	Env.	ODM	Metamorph	DT	PPO	Random
return	walker	**331.88**±280.96	303.933±279.16	252.74±281.18	265.23±275.32	0.55±0.83
	unimal	3197.22±228.04	**3251.68**±192.61	-0.09±3.75	2507.32±260.71	-3.54±4.97
length	walker	**133.29**±35.88	128.42±33.52	126.37±33.73	128.56±34.06	10.34±1.55
	unimal	917.85±40.84	**931.92**±33.21	347.36±66.47	884.39±50.34	321.98±71.49

Few-shot Experiments: We examine the policy transferability by providing several few-shot experiment. Pretrained ODM is loaded in several unseen environments or tasks, which are listed in Table 2. To check the adaptation ability, online training only lasts for 500 steps before testing. Again, ODM obtains the best performance except humanoid on flat terrain.

Table 2: Performance in few-shot experiments.

Metric	Env.	Task	ODM	Metamorph	PPO	Random
return	unimal	obstacle	**1611.86**±179.38	1288.15±127.48	932.34±79.45	-2.08±4.85
	unimal	VT	**580.10**±41.23	499.58±35.21	310.02±22.93	-4.87±8.19
	swimmer	FT	**145.58**±16.97	143.01±11.31	142.36±13.31	0.14±2.00
	reacher	target reaching	**-32.97**±5.18	-33.28±4.58	-34.28±4.56	-42.96±0.15
	humanoid	FT	359.90±54.84	360.87±51.28	**360.70**±50.28	108.13±0.83
	humanoid	standup	**76388.12**±906.01	75750.41±897.01	75033.74±895.20	38921.67±451.25
length	unimal	obstacle	**827.85**±53.25	771.38±50.18	619.82±68.35	322.05±65.81
	unimal	VT	**780.23**±89.02	764.48±77.73	524.10±59.92	542.70±88.76
	swimmer	FT	1000	1000	1000	1000
	reacher	reaching	50^*	50^*	50^*	50^*
	humanoid	FT	**68.10**±3.15	66.85±3.94	67.59±2.34	22.15±0.17
	humanoid	standup	1000	1000	1000	1000

$*$: the official reacher environment has a maximum episode length limit of 50.

Zero-shot Experiments: Zero-shot experiments can be conducted by inferencing the model directly without any online finetuning. The unimal environment allows such experiment in which the flat terrain can be replaced by variable terrain (VT) or obstacles. Results are shown in Table 3. It can be observed that ODM could perform well for zero-shot tests, indicating that our framework could capture general high-level knowledge even without any prior experience.

Table 3: Performance in zero-shot experiments.

Metric	Env.	Task	ODM	Metamorph	DT	Random
return	unimal	obstacle	**1271.70**±182.34	1137.52±178.60	-0.55±3.06	-2.08±4.85
	unimal	VT	**521.08**±34.48	480.29±23.21	8.22±6.70	-4.87±8.19
length	unimal	obstacle	**750.99**±86.23	736.90±75.16	228.20±55.74	322.05±65.81
	unimal	VT	**698.80**±69.49	664.63±72.25	585.13±83.54	542.70±88.76

4.3 Typical Visualizations

General knowledge learning not only aims to improve the mathematical metrics, but also the motion reasonability from human's viewpoint. It is difficult for traditional RL to work on this issue which only solve the mathematical optimization problem. By jointly learning other agent's imitation and bridge with the agent's self-experiences, we assume ODM could obtain more universal intelligence about body control by solving many different types of problems. Here we provide some quick visualizations about generated motions of ODM, comparing with the original versions [5].

By examining the agent motion's rationality and smoothness, we first visualize motions of trained models on the walker environment. Since the walker agent has a human-like body (the 'ragdoll') such that reader could easily evaluate the motion reasonability based on real life experiences. Figure 4.2 exhibit key frames of videos on the same time points. In this experiment, we force the agent starts

[5]Full version of videos can be found on the website `https://baimaxishi.github.io/`

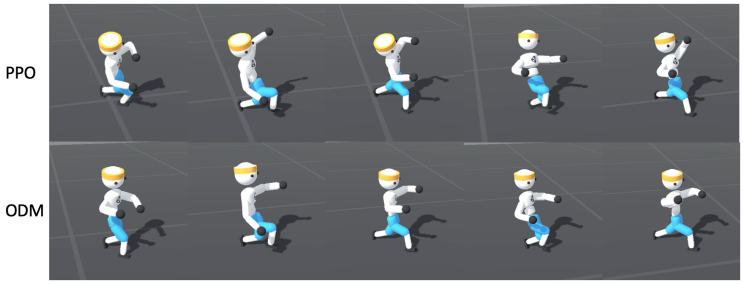

Figure 5: ODM improve motion fluency and coherence in the walker environment. Key frames are screened on Second 1, 2, 3, 4, 5, respectively (See the video on `https://baimaxishi.github.io/`).

from exact the same state and remove the process noise. By comparing ODM (the bottom line) with PPO (the upper line), one can see the ODM behaves more like human while PPO keep swaying forwards and backwards and side to side, with unnatural movements such as lowering shoulders and twisting waist.

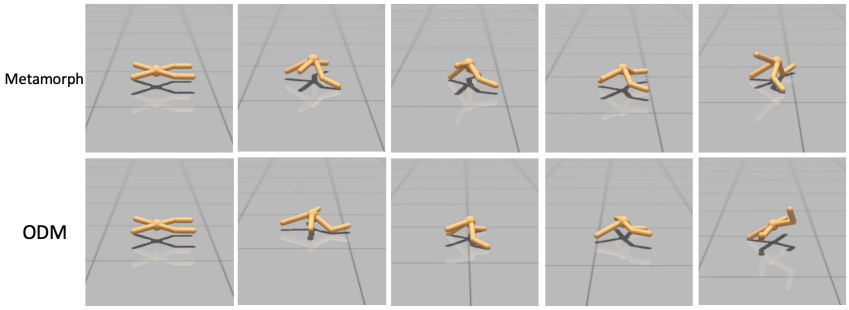

Figure 6: ODM improve motion agility in the unimal environment. Key frames are screened evenly from 30-second videos (See the video on `https://baimaxishi.github.io/`).

We compare the motion agility by visualizing on the unimal environment, in which the agent tries to walk toward arbitrary direction. Figure 4.3 comparing ODM with Metamorph on one of its typical agents. Although it is more obvious by viewing the original video, the grids could help reader to find that the ODM agent walks more distances than the Metamorph agent, within the same time interval. The Metamorph agent spends some time to shake foots, fluiding and gliding, therefore slower to walk the same distance.

5 CONCLUSION

In this paper, motivated by the intelligence development process in the natural world, we propose a learning framework to learn a universal body control policy in arbitrary body shapes, environments and tasks. We combine ideas of learning from others, reinforcing with self-experiences, as well as the world model recognition. To achieve this, we design a two-dimensional transformer structure which first encode the morphological information of agent states and actions at each time step, then encode the time sequential decision process to formulate the policy function. A two-phase training paradigm is designed accordingly in which the agent first learns from demonstrations of pioneers on different skill levels, and from different tasks. After that, the agent interact with its own environment and further reinforce its skill by on-policy RL. Online, few-shot and zero-shot experiments show that our methodology is able to learn some general knowledge for embodied motion control. We believe this work could shed some light on the embodied intelligence study when some kind of generalist intelligent is wanted.

REFERENCES

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 15084–15097. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/7f489f642a0ddb10272b5c31057f0663-Paper.pdf.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL https://arxiv.org/abs/2004.07219.

Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature Communications*, 12, 2021. URL https://doi.org/10.1038/s41467-021-25874-z.

Agrim Gupta, Linxi Fan, Surya Ganguli, and Li Fei-Fei. Metamorph: Learning universal controllers with transformers. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Matej Hoffmann and Rolf Pfeifer. The implications of embodiment for behavior and cognition: animal and robotic case studies. *CoRR*, abs/1202.0440, 2012. URL http://arxiv.org/abs/1202.0440.

Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *ICML*, 2020.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 1273–1286. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/099fe6b0b444c23836c4a5d07346082b-Paper.pdf.

Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *CoRR*, abs/1809.02627, 2018. URL http://arxiv.org/abs/1809.02627.

Kuang-Huei Lee, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Multi-game decision transformers. *Arxiv*, abs/2205.15241v1, 2022. URL https://arxiv.org/abs/2205.15241v1.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Gabriel Barth-Maron Alexander Novikov, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *CoRR*, abs/2205.06175, 2022. URL https://arxiv.org/abs/2205.06175.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS '17, 2017.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *CoRL*, 2022.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24631–24645. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/xu22g.html.

Qinqing Zheng, Amy Zhang, and Aditya Grover. Online decision transformer. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27042–27059. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zheng22c.html.