ModularAgent: A Task-Aware Modular Framework for Joint Optimization of Multimodal Large Language Models and World Models

Yu-Wei Zhan[1] Xin Wang[1*] Pengzhe Mao[2] Tongtong Feng[1*] Ren Wang[1] Wenwu Zhu[1*]
[1]Department of Computer Science and Technology, BNRIST, Tsinghua University
[2]School of Software, Shandong University

1. DMC Setup.

For the DeepMind Control Suite (DMC), we employ the offline dataset released by GenRL [1]. The dataset contains two types of trajectories: (1) trajectories collected by executing the Plan2Explore exploration policy [2]; and (2) trajectories sampled from the replay buffers of task-specific RL agents. Since the original release does not include the Quadruped domain, we additionally collect Quadruped trajectories following the same data-collection protocol. A detailed summary of dataset statistics is provided in Table 1.

For text prompts, we adopt the prompts listed in Table 1, keeping them as close as possible to those in GenRL to ensure fair comparison. Unless otherwise stated, all experiments use 64×64 image observations with an action repeat of 2. Moreover, we report the expert score and random policy score for each task, which are used to compute normalized rewards. The normalized mean episodic reward is defined as:

$$\text{mean episode reward} = \frac{\text{episode reward} - \text{random score}}{\text{expert score} - \text{random score}}.$$

This metric serves as the primary performance measure throughout our experiments.

2. Cross-domain Task Solving

We further evaluate the cross-domain generalization ability of our method on three representative tasks: Run, Stand, and Walk. In each experiment, one environment is used as the source domain for training, and the remaining two environments are used as target domains for transfer evaluation. The results are shown in Fig. 1. Since the official implementation of FOUNDER is not available, we compare our method with two accessible model-based baselines, WM-CLIP and GenRL.

Across almost all settings, our method achieves the best performance. In challenging transfer scenarios, such as transferring the fast run behavior from the humanoid-like Stickman domain to the Quadruped domain, WM-CLIP and GenRL often collapse to near-random behavior and occasionally produce negative rewards. In contrast, our method consistently maintains strong performance in these difficult settings. These results demonstrate the effectiveness of our task-aware adaptive modular architecture. By allowing related tasks to share transferable structures across embodiments, the proposed method successfully reuses knowledge learned in previous environments and achieves robust cross-domain generalization.

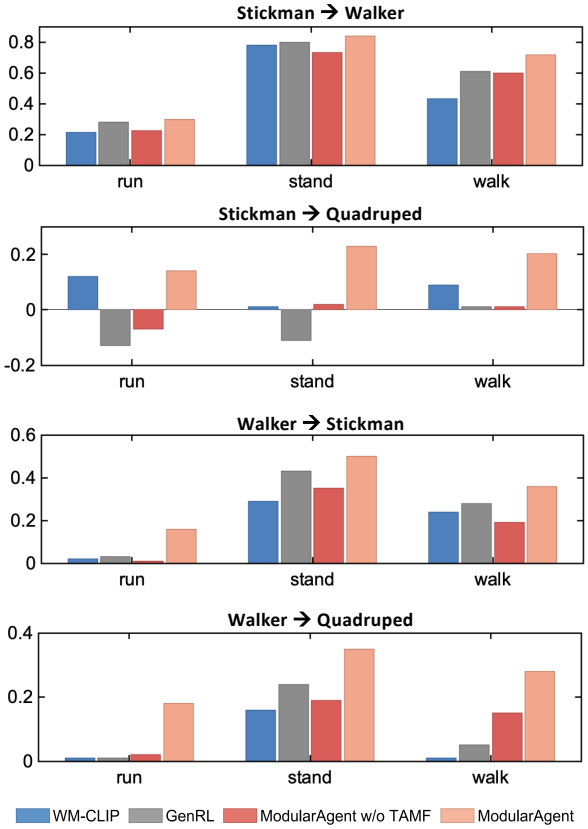

Figure 1. Cross-environment generalization of agents.

*Corresponding authors

Domain	Task	Prompt	Expert Score	Random Score	Dataset Size
Cheetah	run	running like a quadruped	890	9	1.8M
	p2e	running like a quadruped	-	-	1.8M
Walker	stand	standing up straight like a human	970	150	500K
	walk	walk fast clean like a human	960	45	500K
	run	run fast clean like a human	770	30	500K
	p2e	doing backflips	-	-	1M
Stickman	stand	standing like a human	970	70	500K
	walk	robot walk fast clean like a human	960	35	500K
	run	robot run fast clean like a human	830	25	500K
	p2e	doing flips like a human	-	-	1M
Quadruped	stand	spider standing	990	15	500K
	walk	walking fast clean like a quadruped spider	960	10	500K
	run	running fast clean like a quadruped spider	930	10	500K
	p2e	doing flips like a human	-	-	1M

Table 1. **Combined task prompts, expert/random score, and offline dataset statistics** across DMC.

Table 2. Key hyperparameters used in all experiments.

Component	Setting
World Model (RSSM)	
Deterministic size	1024
Stochastic size	32 (32-category discrete)
Posterior type	Single-observation posterior
KL balance coefficient	0.85
Free KL	1.0
Optimization	
Optimizer	Adam
WM learning rate	1×10^{-4}
Actor/Critic learning rate	3×10^{-5}
Gradient clip	100 (actor/critic), 1000 (WM)
Behavior Learning	
Discount factor	0.99
GAE λ	0.95
Imagination horizon	16
Training Setup	
Observation size	64×64
Batch size	64
Batch length	32
Precision	FP16
Action repeat	2
Modular Fusion (MLLM–WM)	
Fusion layers	5
Loss weights	$\lambda_{\text{WM}} = \lambda_{\text{MLLM}} = \lambda_{\text{JBO}} = 1$

Table 3. Comparison with DreamerV3 (Q: Quadruped, W: Walker).

Method	Q-Run	Q-Walk	W-Run	W-Walk	**average**
DreamerV3	0.66	0.84	0.88	1.00	0.845
ModularAgent	0.98	1.04	0.94	1.04	1.00

3. Additional Training Details and Parameters

In all experiments, we adopt a Dreamer-style latent world model and use InternVideo2 as the MLLM, with a unified training configuration across all settings. In our experimental setup, we pretrain the world model and its associated components for 100K gradient steps, followed by another 50K updates during the behavior learning phase. The behavior learning module is the only component updated during downstream fine-tuning; the world model and MLLM parameters remain frozen. Specifically, we adopt a two-stage training paradigm, where each loss is applied to a clearly defined subset of parameters to avoid gradient interference. In the first stage, $\mathcal{L}_{\text{WM}}$ and $\mathcal{L}_{\text{MLLM}}$ jointly train the world model encoder (p_θ, q_ϕ), MLLM encoder, and TAMF. In the second stage, we freeze these components and optimize only the policy network π_ψ using $\mathcal{L}_{\text{JBO}}$.

For cross-domain transfer, we introduce a lightweight Action Adapter that aligns the action spaces of different morphologies (e.g., 6-D Walker $\rightarrow$ 10-D Quadruped). The adapter is trained jointly with the behavior policy and adds negligible overhead to the downstream optimization.

All agents are trained on 64×64 visual observations with a batch size of 64, a sequence length of 32, and an imagination horizon of 16. The model is optimized using the Adam

Table 4. Training cost comparison across baselines. "Ours" refers to our proposed method.

Method	TD3	GenRL	FOUNDER	Ours
Performance	Low	Medium	High	High
Multi-Task Adaptation	Scratch (∼7h)	Finetune (∼5h)	Finetune (∼5h)	Finetune (∼5h)
Pretraining Overhead	N/A	∼120h	∼72h	∼**20h**
Overall Cost	Low	High	Medium	**Low-Medium**

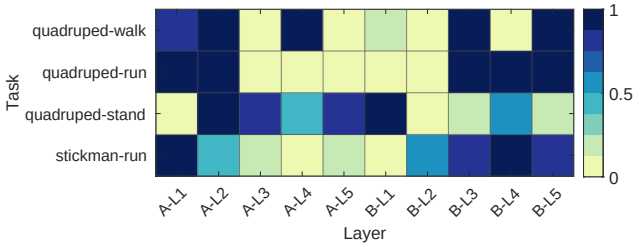

Figure 2. The selection probabilities of the Semantic Expert (A) and Dynamic Expert (B) for four representative tasks within the five-layer modular fusion architecture.

optimizer. For the semantic–dynamic fusion module, the number of modular layers is fixed to 5, and the loss weights λ_{WM}, λ_{MLLM}, and λ_{JBO} are all set to 1 to ensure balanced contributions during joint optimization. Each task is evaluated across 5 seeds. A complete list of key hyperparameters is provided in Table 2.

4. Compared with DreamerV3

We compare our method with DreamerV3. Compared to DreamerV3 trained with 1M steps per task, our method uses only 50K steps yet achieves better performance, demonstrating that incorporating an MLLM effectively improves decision-making efficiency and performance.

5. Visualization of the Task-Aware Expert Activation

To better characterize the task-level dynamic adaptation enabled by our method, we analyze the expert routing behavior across different tasks and layers. The Fig. 2 visualizes the selection probabilities of the Semantic Expert (A) and Dynamic Expert (B) for four representative tasks within the five-layer modular fusion architecture.

For the three tasks originating from the same environment Quadruped, we observe clear task-specific routing patterns, reflecting differences in semantic demands and dynamic complexity of different tasks. We found that Stickman-run and Quadruped-run exhibit highly similar semantic–dynamic routing profiles in several layers, suggesting that the model can transfer shared task structure across different embodiments and thus achieves strong cross-environment generalization. In addition, across all tasks, a consistent trend emerges: shallow layers favor the Semantic Expert to extract high-level semantics and contextual cues, whereas deeper layers increasingly activate the Dynamic Expert to capture fine-grained physical variations and motion dynamics.

6. Training Time Analysis

Our method achieves significantly improved training efficiency compared with existing model-based baselines. As summarized in Table 4, our world model can be pretrained on a single A100 GPU within **20 hours**, which is substantially lower than the pretraining time required by GenRL (∼120 hours) and FOUNDER (∼72 hours). During the behavior learning phase, our method requires only **5 hours** of finetuning to obtain high-performance policies, matching the efficiency of existing finetune-based approaches.

The major efficiency gain comes from performing imagination rollouts and reward computation at the task level. By generating task-conditioned imagined trajectories and computing dense rewards directly in latent space, our approach achieves fast convergence while maintaining strong adaptability and generalization, resulting in an overall training cost that is considerably lower than prior work.

References

[1] Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron Courville, and Sai Rajeswar. Genrl: Multimodal-foundation world models for generalization in embodied agents. *Advances in neural information processing systems*, 37:27529–27555, 2024. 1

[2] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *Proceedings of the International Conference on Machine Learning*, pages 8583–8592. PMLR, 2020. 1