# FOCUS: SCALING CONTINUOUS CONTROL VIA STRUCTURED EXPLORATION OVER DIVERSIFIED ACTION SUBSPACES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Reinforcement learning in high-dimensional continuous action spaces faces a significant bottleneck: the exponentially expanding search space leads to poor sample efficiency and limited exploration scalability. To address this challenge, we propose **FOCUS**, a framework designed for **structured exploration** that avoids the exhaustive search of the full action space. The core insight of FOCUS lies in **decoupling** control into a hierarchical selection of **diversified action subspaces**, effectively rendering the exploration process more tractable. Specifically, a high-level policy generates **binary** preferences to dynamically activate task-critical control units, while a low-level policy produces precise actions conditioned on these selections. To ensure the effectiveness of this hierarchy, we incorporate a **diversity-consistency tradeoff** in the optimization objective. This objective encourages functional specialization by promoting distinct action distributions between preferences that select a control unit and those that do not, while encouraging consistency across preferences that leave the unit unselected. Extensive evaluations on challenging continuous control benchmarks demonstrate that FOCUS significantly outperforms state-of-the-art methods, particularly as the action dimensionality increases, establishing its superior scalability and exploration efficiency.

## 1 INTRODUCTION

Model-based reinforcement learning (MBRL) provides a sample-efficient framework for decision-making tasks, particularly with high-dimensional observations, by learning transition models that can generate imagined trajectories Hafner et al. (2025). However, its scalability is often hindered in high-dimensional action spaces, where exploration and policy optimization tend to become inefficient and unstable. As the number of controllable degrees of freedom increases, traditional methods suffer from **exploration dilution**, where stochastic search signals are spread across numerous redundant or task-irrelevant dimensions. In many real-world domains, such as humanoid locomotion, power grid control, and multi-agent collaboration, although the action spaces are high-dimensional, only a sparse subset of dimensions is typically crucial for effective decision-making at any given temporal slice.

A natural approach to mitigating the challenges of high-dimensional action spaces is to exploit their underlying structure. In particular, biasing policy learning toward more informative action dimensions can substantially reduce search complexity and improve efficiency. Several prior methods have explored this direction by introducing hierarchical or factorized action representations. However, many of these approaches rely on discrete action assumptions Saito et al. (2024); Chen et al. (2019); Kumar et al. (2017) or predefined base-action sets Kim & Dean (2002); Geißer et al. (2020), thereby limiting their scalability to general continuous control tasks.

To improve exploration and policy optimization in high-dimensional continuous action spaces, we propose **FOCUS** (Figure 1), a scalable MBRL framework. Unlike existing MBRL methods that treat all action dimensions uniformly Hafner et al. (2025); Hansen et al. (2024), FOCUS introduces a control unit switching mechanism which automatically identifies promising action subspaces, adaptively steering behavior optimization toward the most impactful degrees of freedom.

Specifically, FOCUS represents the activation of control units using learnable Bernoulli distributions over all action dimensions. The agent presents a **hierarchical decision-making strategy**: (i) A



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

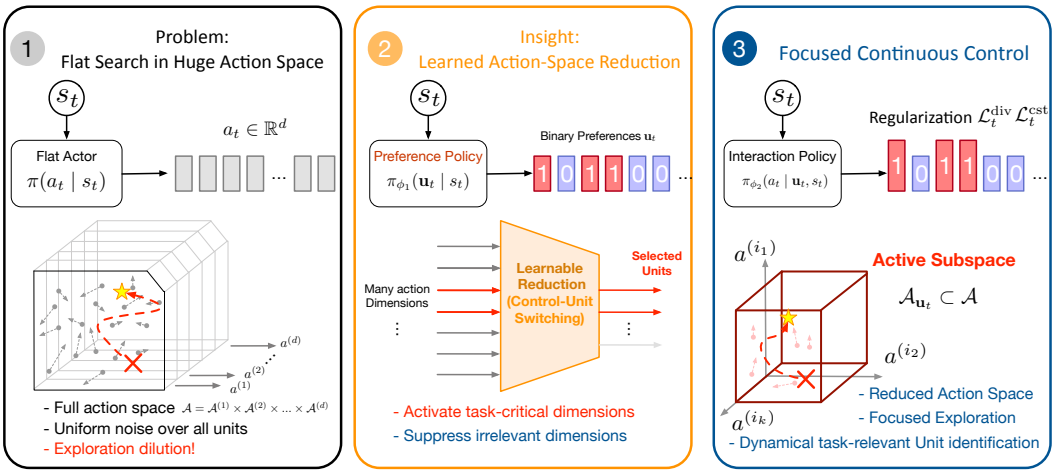


Figure 1: **Introduction to FOCUS framework.** FOCUS addresses the challenges of high-dimensional continuous control by employing a hierarchical control-unit switching mechanism. The high-level policy generates binary preferences to dynamically activate task-relevant action subspaces, concentrating policy optimization and exploration on the most impactful control units.

high-level policy predicts activation probabilities and generates binary signals (the "*control-unit preference prompt*") to gate specific control units; (ii) a low-level interaction policy, conditioned on these signals, specializes its control within the selected action subspaces. Both policies are jointly optimized to maximize expected returns over imagined rollouts produced by the learned state transition model. To encourage diversified structured exploration and alignment between the cascaded decision layers, we introduce a **diversity-driven policy optimization objective** that maximizes behavioral discrepancies across different control-unit preferences. Once trained, the agent carries out the hierarchical policies by performing **preference-based Monte Carlo planning**, which reduces possible control-unit sampling bias and significantly improves the decision quality.

Experiments on benchmarks with high-dimensional action spaces, including DeepMind Control, MyoSuite, HumanoidBench and Grid Preventive Control, demonstrate that FOCUS outperforms model-based and model-free RL approaches in both learning efficiency and final performance.

An overview of FOCUS is illustrated in Figure 1. In summary, our main contributions are as follows:

- We present FOCUS, a high-dimensional MBRL framework that utilizes a structured exploration mechanism to prioritize promising control units for optimization and planning by modeling the importance of individual action dimensions.
- To enable the policy to respond to varying control-unit preferences, we propose diversity-driven policy learning to promote diversified behaviors on prioritized action subspaces and ensure alignment between cascaded policies.
- We introduce preference-based SMC planning, which delivers higher computational efficiency and decision quality than CEM-based planning approaches in high-dimensional RL.

## 2 Related Work

### 2.1 RL with High-Dimensional Action Spaces

One way to handle high-dimensional actions is to exploit known structure. Factored MDP and multi-agent methods use a specified factorization of the decision problem Osband & Roy (2014); Guestrin et al. (2001); Mahajan et al. (2021); Peng et al. (2021), while methods for over-actuated systems use musculoskeletal knowledge to design or reduce the control space Schumacher et al. (2023); Luo et al. (2023). Such structure can be useful, but may require domain knowledge that is unavailable in a new task. Some methods address action redundancy through learned action representations, regularization, or filtering Chandak et al. (2019); Baram et al. (2021); Zhong et al. (2024). Their focus is on redundant actions or effects, rather than state-dependent preferences over native continuous control dimensions. Other approaches focus the action space through masking or action selection Stolz et al. (2024); Zhang & Cai (2025), or organize action components with structured policies Sharma

et al. (2017); Tavakoli et al. (2018). FOCUS instead couples state-dependent preferences over native continuous control units to a full-dimensional actor through diversity and consistency objectives, without hard-masking the executed action.

### 2.2 Hierarchical Reinforcement learning

Prior studies in hierarchical structured action space usually assume a multi-stage policy that divides the search space into smaller parts Kumar et al. (2017); Chen et al. (2019); Tang & Agrawal (2020); Saito et al. (2024). However, these methods are primarily restricted to tasks with discrete action spaces. In a broader context, hierarchical policy structures have traditionally been employed to enable efficient long-horizon reasoning through temporal abstractions Pateria et al. (2021); Vezhnevets et al. (2017); Gumbsch et al. (2024); Gürtler et al. (2021), such as goals or options. For instance, Director is a model-based framework that features a hierarchical structure where a high-level autoencoder generates long-term goals Hafner et al. (2022). In contrast to these temporal-focused approaches, FOCUS leverages hierarchical structures to model the dimensional sparsity inherent in large action spaces. Instead of decomposing a task over time, our method reduces the instantaneous search space by switching between specific control units.

## 3 Problem Formulation

Our work targets continuous control tasks with high-dimensional action spaces, where exploring and optimizing over many control dimensions is challenging. Formally, we formulate the problem as an MDP defined by a 5-tuple $(\mathcal{O}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$, where: $\mathcal{O}$ is the high-dimensional observation space, $\mathcal{A}$ is the high-dimensional continuous action space, $\mathcal{R}$ is the reward function, $\mathcal{T}$ is the state transition dynamics, and $\gamma$ is the discount factor.

In particular, we assume the action space is a Cartesian product of $d$ continuous subspaces:

$$\mathcal{A} = \mathcal{A}^{(1)} \times \mathcal{A}^{(2)} \times ... \times \mathcal{A}^{(d)}, \quad \text{with } \mathcal{A}^{(i)} \subset \mathbb{R}. \tag{1}$$

An action at time step $t$ is thus represented as a $d$-dimensional vector $a_t = [a_t^{(1)}, \ldots, a_t^{(d)}]$. In this framework, we refer to each dimension $a_t^{(i)} \in \mathcal{A}^{(i)}$ (or group of dimensions[1]) as a **Control Unit**. The objective is to learn a policy $\pi : \mathcal{O} \to \mathcal{A}$ that maximizes $\mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t]$.

**Challenges.** The entanglement of high-dimensional observations and high-dimensional continuous actions presents unique and under-explored challenges. First, the state space (*e.g.*, pixel data, grid power-flow state) is inherently noisy and redundant, making it non-trivial to extract a compact latent representation that captures essential dynamics required for stable learning. Moreover, in large continuous action spaces, the volume of the search space grows exponentially with the number of dimensions $d$. Standard exploration strategies typically apply uniform noise across all control units, and suffer from exploration dilution, where the effective learning signal is lost amidst task-irrelevant stochasticity. When $d$ is large, without a mechanism to prioritize relevant action subspaces, the policy optimization process often becomes trapped in suboptimal local minima or fails to converge entirely.

## 4 Method

### 4.1 Overview of FOCUS

To tackle the challenges of exploration dilution caused by high-dimensional action spaces, we propose FOCUS, an MBRL framework guided by a hierarchical switching mechanism. The training and environment interaction pipeline consists of three primary stages:

- **Transition model learning:** We follow DreamerV3 Hafner et al. (2025) to train a Recurrent State Space Model (RSSM) to predict latent state transitions and rewards from observed transitions.
- **Diversity-driven policy learning:** FOCUS employs a hierarchical policy architecture, where a high-level **Preference Policy** outputs an activation distribution over all control units. At each step, it samples the preference prompt from Bernoulli distributions, dynamically activating sparse subsets of control units and guiding the low-level **Interaction Policy** to generate specific actions. Both layers are jointly optimized to maximize expected returns while promoting behavioral diversity.

[1]Each *control unit* corresponds to a specific functional action subset (*e.g.*, a joint in a humanoid robot), potentially spanning multiple dimensions to reflect its degrees of freedom.

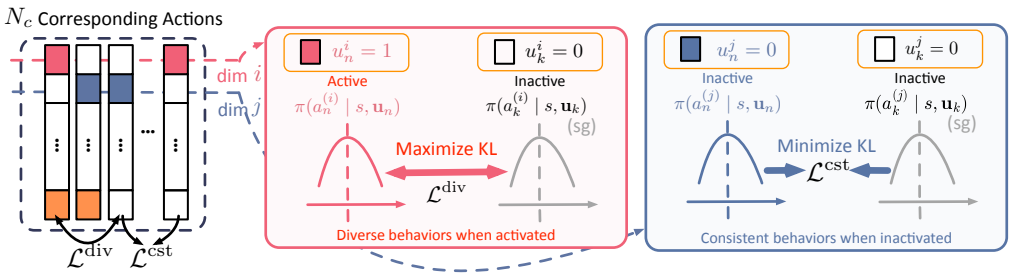


Figure 2: **Diversity-Driven Policy Learning.** Given $N_c$ preference prompts and corresponding actions, we apply $\mathcal{L}^{\text{div}}$ to maximize KL divergence on *activated* dimensions ($u^{(i)} = 1$, pink) for diverse behaviors, and $\mathcal{L}^{\text{cst}}$ to minimize that on *inactive* dimensions ($u^{(j)} = 0$, blue) for consistency.

- **Preference-based Monte Carlo planning:** During environment interaction, the agent samples candidate preference prompts and performs multi-step Monte Carlo planning. Candidate actions are weighted by cumulative advantage estimates along trajectories, ensuring robust decision-making.

### 4.2 Control Unit Switching Mechanism

Our approach is motivated by the observation that high-dimensional control tasks often exhibit inherent sparsity—at any given time, some control units more significantly contribute to task performance than others. For example, in the `HumanoidStand` task, the agent begins by pushing itself upward from the ground using support from its legs and arms. In such moments, joints such as the ankles and wrists play a dominant role, while other control units are secondary.

By selectively switching between these units, exploration becomes significantly more efficient. We introduce a high-level Preference Policy $\pi_{\phi_1}$ that generates a binary switching signal $\mathbf{u}_t \in \{0, 1\}^d$, where $u_t^{(i)} = 1$ activates the $i$-th control unit. This design leads to a hierarchical policy structure, where the Interaction Policy $\pi_{\phi_2}$ is conditioned on both the latent state and the preference prompt:

$$\text{Preference policy:} \quad \mathbf{u}_t \sim \pi_{\phi_1}(s_t), \quad \text{Interaction policy:} \quad a_t \sim \pi_{\phi_2}(\mathbf{u}_t, s_t). \tag{2}$$

The high-level policy predicts control-unit Bernoulli probabilities from the latent state and samples preferences independently. Conditioning the low-level policy on the full prompt and jointly optimizing both policies for return can favor useful co-activation patterns and coordinated continuous actions.

### 4.3 Diversity-Driven Policy Learning

To promote an efficient exploration of potential policies guided by diverse control-unit preferences, we introduce a diversity-driven optimization objective. Without regularization, a hierarchical agent may ignore the switching decisions and collapse into a standard uniform policy.

In our model-based training framework, FOCUS samples a preference prompt $\mathbf{u}_t$ at each RSSM rollout step, and generates a preference-conditioned action $a_t$ based on the current rollout state $\hat{s}_t$. The action $a_t$ is passed through the transition model to update the imagined trajectories, which are then used to jointly optimize the hierarchical policy $\pi_{\phi_{1,2}}$ and the value model $v_\psi$.

However, directly maximizing the expected value function, as in Dreamer-style objectives, is insufficient for inducing low-level behaviors that are responsive to high-level preferences, where the actor may fail to interpret and follow the high-level guidance effectively. To address this, we design a diversity-driven learning objective serving as soft regularization, as illustrated in Figure 2. The key idea is that the low-level actor should produce diverse behaviors for control units selected by different preference prompts, while maintaining consistent behaviors for those not selected.

Specifically, we sample $N_c$ preference prompts $\{\mathbf{u}_{t,n}\}_{n=1}^{N_c}$ from $\pi_{\phi_1}(s_t)$ and corresponding action $\{a_{t,n}\}_{n=1}^{N_c}$, each decomposed as $a_{t,n} = [a_{t,n}^{(1)}, \ldots, a_{t,n}^{(d)}]$ along $d$ dimensions. We encourage the actor to produce diverse behaviors for activated units while maintaining consistency for inactive ones:

**Diversity-promoting loss.** We encourage distinct behaviors for activated units ($u^{(i)} = 1$):

$$\mathcal{L}_t^{\text{div}} = \sum_{i=1}^{d} \left[ -\sum_{n=1}^{N_c} \mathbb{I}(\mathbf{u}_n^{(i)} = 1) \sum_{k=1}^{N_c} \mathbb{I}(\mathbf{u}_k^{(i)} = 0) \cdot \text{KL}\left[\pi(a_n^{(i)} \mid s, \mathbf{u}_n) \,\|\, \text{sg}(\pi(a_k^{(i)} \mid s, \mathbf{u}_k))\right] \right] \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function, $\mathrm{sg}(\cdot)$ denotes stop-gradient. We ignore time step index for clarity.

**Consistency loss.** We penalize divergence for units not selected by the prompt ($u^{(j)} = 0$):

$$\mathcal{L}_t^{\text{cst}} = \sum_{j=1}^{d} \left[ \sum_{n=1}^{N_c} \mathbb{I}(\mathbf{u}_n^{(j)} = 0) \sum_{k=1}^{N_c} \mathbb{I}(\mathbf{u}_k^{(j)} = 0) \cdot \mathrm{KL}\left[ \pi(a_n^{(j)} \mid s, \mathbf{u}_n) \parallel \mathrm{sg}(\pi(a_k^{(j)} \mid s, \mathbf{u}_k)) \right] \right]. \quad (4)$$

These losses impose a contrastive structure in the high-dimensional action space, enhancing exploration and better alignment between low- and high-level policies. The final actor objective is:

$$\mathcal{L}(\phi_{1,2}) = \mathbb{E}_{p_\psi, p_\theta} \left[ \sum_{t=1}^{H-1} \left[ \mathcal{L}_t^{\text{div}} + \mathcal{L}_t^{\text{cst}} - V_t^\lambda - \eta \mathcal{H} \right] \right], \quad (5)$$

where the entropy $\mathcal{H} = \mathcal{H}(a_t \mid \hat{s}_t, \mathrm{sg}(\mathbf{u}_t))$ further encourages exploration conditioned on preference prompts. Notably, both levels of policies are jointly optimized to maximize the imagined values.

Importantly, we do not stop the gradient at $\mathbf{u}_n$ in the above regularization terms, allowing $\pi_{\phi_1}$ to be co-optimized with $\pi_{\phi_2}$. This design further promotes exploration, as $\mathcal{L}_t^{\text{div}}$ rewards diverse control-unit activations and prevents the high-level policy from collapsing to identical but biased activation signals.

The value network is optimized with:

$$\mathcal{L}(\psi) = \mathbb{E}_{p_{\phi_{1,2}}, p_\theta} \left[ \sum_{t=1}^{H-1} \frac{1}{2} (v_\psi(\hat{s}_t) - \mathrm{sg}(V_t^\lambda))^2 \right]. \quad (6)$$

For the sake of efficiency, we take $s_{t+1,n} \sim p_\theta(s_t, a_{t,n})$ with the highest predicted value $v_\psi(s_{t+1,n})$ as $s_{t+1}$ among multiple candidates at each imagination step. In practice, we compute the diversity-driven objectives only at the first step of imagination. Despite this simplification, we observe a significant performance gain over versions without diversity-driven regularization, while incurring little additional training overhead.

### 4.4 Preference-Based Monte Carlo Planning

To reduce variance during environment interaction, we employ an online planning scheme that reasons over multiple future trajectories, each conditioned on a different preference prompt. This Preference-Based Monte Carlo Planning allows the agent to evaluate which "switch configuration" is most promising before executing an action.

At each interaction step, we sample $N_p$ preference prompts $\{\mathbf{u}_n\}_{n=1}^{N_p}$ based on the current state $s_t$, and initialize $N_p$ planning trajectories with uniform weights $w_n = 1$. At each planning step $t + h$, we reuse $\mathbf{u}_n$ to sample low-level actions $\hat{a}_{t+h,n} \sim \pi_{\phi_2}(\hat{s}_{t+h,n}, \mathbf{u}_n)$. This contrasts with model-based policy learning, where $\mathbf{u}_n$ is typically re-sampled at each imagination step. Instead, we preserve temporal consistency in the sampled control-unit prompts during planning. We then use the learned dynamics $p_\theta$ to rollout future states $s_{t+h:t+L,n}$. We evaluate these candidate trajectories via advantage estimation and update their weights accordingly:

$$A_{t+h,n} = \hat{r}_{t+h,n} + \gamma v_\psi(\hat{s}_{t+h+1,n}) - v_\psi(\hat{s}_{t+h,n}), \quad w_{t+h+1,n} = w_{t+h,n} \cdot \exp(A_{t+h,n}). \quad (7)$$

At the end of the planning horizon, we sample one trajectory from the categorical distribution defined by $\{w_{t+L,n}\}_{n=1}^{N_p}$ and execute its first action in the environment. Empirically, this approach significantly outperforms prior sample-based planning methods (*e.g.*, MPPI in TD-MPC2 Hansen et al. (2024)) in both computational efficiency and decision quality. Unlike MPPI, our method avoids online policy refitting, making it more suitable for high-dimensional control tasks.

## 5 Experiments

### 5.1 Experimental Setup

**Environments.** We first evaluate FOCUS on two challenging reinforcement learning benchmarks with visual inputs: DeepMind Control Suite Tassa et al. (2018) and MyoSuite Caggiano et al. (2022). We report empirical results on six specific tasks characterized by high-dimensional continuous action spaces. We then conduct experiments on four tasks from HumanoidBench Sferrazza et al. (2024), which utilize proprioceptive states as input, together with a Grid Preventive Control problem where the agent observes the full grid power-flow state. Detailed information regarding the number of action dimensions is provided in the Appendix.

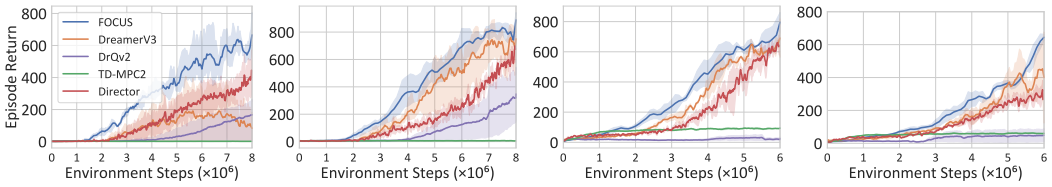


Figure 3: **Results on DeepMind Control Suite.** All results are averaged over three training seeds.

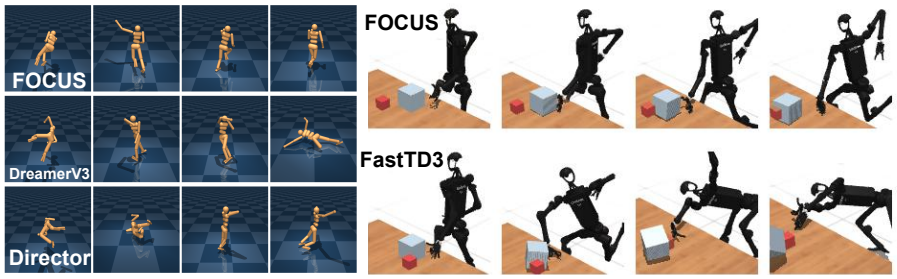


Figure 4: **Policy Visualization.** We compare different models on `HumanoidWalk` (Left) and `H1hand-push` (Right). In `H1hand-push`, the robot's task is to push the object to the red target.

**Baselines.** We compare FOCUS against the following strong baselines to assess its efficiency in high-dimensional domains:

- **DreamerV3** Hafner et al. (2025): A strong MBRL method in high-dimensional observation space.
- **TD-MPC2** Hansen et al. (2024): A decoder-free MBRL baseline with CEM-based planning.
- **DrQv2** Yarats et al. (2022): A prominent model-free visual RL approach with data augmentation.
- **Director** Hafner et al. (2022): A hierarchical MBRL method that optimizes behavior by conditioning a low-level policy on high-level goals (representing a temporal hierarchy).
- **FastTD3** Seo et al. (2025): A model-free TD3 variant that accelerates humanoid control training through parallel simulation, large-batch updates, and a distributional critic. We use it exclusively on HumanoidBench for its high training efficiency. We construct FastTD3-FOCUS by replacing its flat policy with our hierarchical switching architecture and adding diversity-driven regularization.
- **SAC** Haarnoja et al. (2018): Similar to FastTD3, We augment SAC with hierarchical switching and diversity-driven regularization for Grid Preventive Control.

**Training protocol.** All models are trained under a fixed environment interaction budget. For two tasks in the `Humanoid` suite, each model is trained for 8M frames, and for the `Dog` suite, 6M frames. Tasks in `MyoSuite` allow interactions for 1M frames, 500K for tasks in `HumanoidBench` and 10K for grid control benchmark. Average episode returns are computed over 10 episodes per seed.

### 5.2 Domain #1: DeepMind Control Suite

As shown in Figure 3, FOCUS consistently outperforms baseline methods across all four tasks. Notably, it achieves a substantial improvement over DreamerV3 in `HumanoidWalk`. In both `Humanoid` tasks, the learning curves show a prolonged warm-up phase before meaningful improvement, reflecting the challenge of locating sparse high-reward regions within large action spaces. Compared with **DreamerV3**, our model exhibits an earlier performance rise, demonstrating its ability to reduce sample complexity and efficiently explore targeted reward regions.

In `Dog` tasks, rewards are less sparse, and DreamerV3 is already capable of sampling high-reward regions as shown by its steadily increasing learning curves. Despite more action dimensions, the high-reward regions in the action space are sufficiently large for random policies to reach. Thus, the exploration is not the primary bottleneck, limiting potential performance gains. **TD-MPC2** fails to achieve meaningful progress on `humanoid` tasks, highlighting its difficulty in effectively exploring the vast search space, particularly under pixel input. The comparison with **Director** highlights the impact of the diversity-driven objective: while both methods employ hierarchical policies, FOCUS explicitly prioritizes subsets of the action space, yielding higher exploration efficiency.

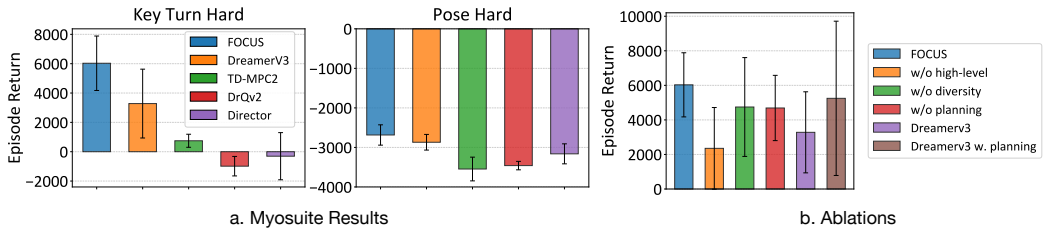


Figure 5: (a). **Results on the MyoSuite hard tasks.** Returns are averaged over three training seeds, with 10 evaluation episodes per seed. FOCUS outperforms all baselines across every task. (b). **Effect of each model component (`KeyTurnHard`)**

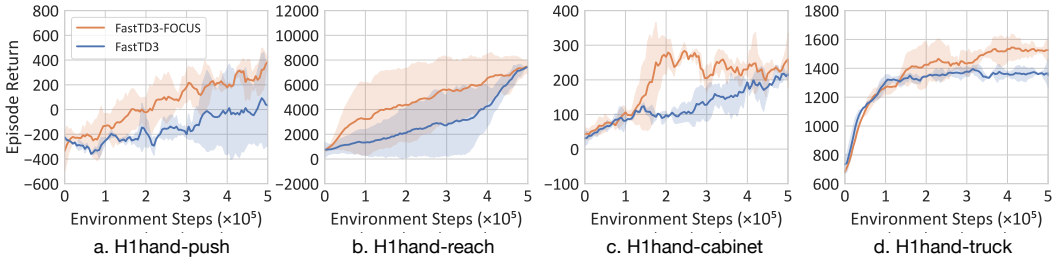


Figure 6: **Results on HumanoidBench.** We integrate the FOCUS switching mechanism into FastTD3 rather than DreamerV3 or TD-MPC2, which fail to reach convergence within the allocated interaction budget. All empirical results are reported as averages across three independent training seeds.

We showcase the policy evaluation of FOCUS, DreamerV3, and Director in Figure 4(left), with full results in the Appendix. As illustrated, FOCUS policy adjusts the humanoid's posture in fewer time steps to start walking ($t = 50$), meanwhile maintaining more stable control than DreamerV3, which falls to the ground ($t = 200$) after standing up. Additionally, **DrQv2** requires roughly 4M warm-up environment steps, suggesting that augmenting failing episodes provides limited benefit.

We provide further results, comparing the planning efficiency of FOCUS with TD-MPC2 in Figure 13, and evaluating its generalization to low-dimensional action spaces in Table 2.

### 5.3 Domain #2: MyoSuite

Performance results on `MyoSuite` are shown in Figure 5a. FOCUS outperforms all baselines on `KeyTurnHard`. Specifically, it improves over DreamerV3, demonstrating higher efficiency in exploring high-reward regions within the same environment step budget. Other baselines struggle to solve the task, consistent with the difficulty of exploring its high-dimensional action space. All methods face challenges on `PoseHard`, though FOCUS achieves slightly better results, thanks in part to $\mathcal{L}_t^{cst}$ preventing overfitting as discussed in Section 4.3.

### 5.4 Domain #3: HumanoidBench

We present the results of four tasks on `HumanoidBench` in Figure 6. To evaluate the impact of our switching mechanism in an environment-efficient setting, we implemented FOCUS on top of FastTD3 rather than DreamerV3 or TD-MPC2. FastTD3 is specifically optimized for high-speed interaction and serves as a robust baseline for these high-dimensional humanoid control tasks. Detailed descriptions of the FastTD3 architecture and our hierarchical modifications are in Appendix.

In all four tasks, **FastTD3-FOCUS** demonstrates superior sample efficiency compared to the vanilla FastTD3. On `H1hand-push` and `H1hand-truck`, FastTD3-FOCUS improves consistently throughout training, indicating that focused exploration helps the agent escape local optima that trap the flat MLP policy. On `H1hand-reach` and `H1hand-cabinet`, while both methods eventually converge to similar return levels, FastTD3-FOCUS reaches these milestones significantly faster. The diversity-driven objective encourages the agent to discover distinct and functional behavior patterns among different control units. This structured exploration enables faster coverage of high-reward regions in the state space, effectively accelerating the acquisition of complex reaching behaviors.

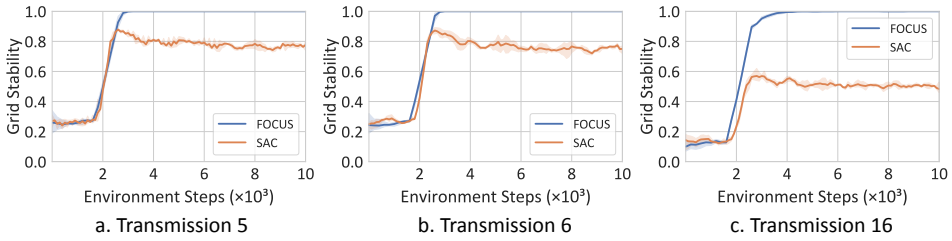


Figure 7: **Results on Grid Preventive Control.** We implement the hierarchical policy structure in FOCUS on top of SAC. All empirical results are reported as averages across three independent seeds.

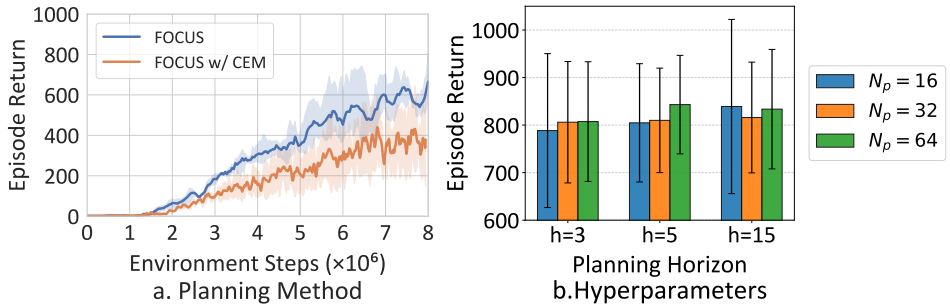


Figure 8: **Evaluation of the planning method:** (a): Comparison of planning efficiency with CEM method. (b): Effect of planning horizon $h$ and candidate trajectories $N_p$ (`HumanoidWalk`).

### 5.5 Domain #4: Grid Preventive Control

Grid preventive control is a typical high-dimensional control problem that requires continuous coordination across massive generators and loads to maintain system stability under disturbances. Specifically, we consider an IEEE 39-bus system where three-phase ground faults may disturb the transmission lines. At each step, the agent observes the full power-flow state of the grid, and is required to continuously adjust active/reactive power outputs of generators and shed a fraction of selected loads, steering the system toward a new pre-fault operating point that is robust to upcoming disturbances. The agent receives a reward of 1 if the post-fault system remains stable (generators do not lose synchronism), and 0 otherwise. With 10 generators and 19 loads, the resulting action space is high-dimensional and tightly coupled through the underlying power-flow equations.

In this benchmark, we build the hierarchical policy structure upon SAC and augment its training objective with $\mathcal{L}_t^{div}$ and $\mathcal{L}_t^{cst}$. As is shown in Figure 7, FOCUS converges to full stability rapidly, whereas SAC remains suboptimal and exhibits persistent oscillations throughout training. This gap highlights the complementary roles of the two components: (1) the hierarchical policy decomposes the high-dimensional joint action space over generators and loads into semantically meaningful control units, substantially reducing the effective exploration dimensionality; (2) the diversity-driven objective further encourages distinct behaviors across activated subsets, improving exploration efficiency over flat policies that explore uniformly across all action dimensions.

### 5.6 Model Analyses

**Ablation studies.** We conduct ablation studies on the `KeyTurnHard` task in MyoSuite to quantify the contribution of each core component. We evaluate three specific variants of our method: (1) **No switching policy**: removes the high-level policy $\pi_{\phi_1}$, (2) **No diversity objectives**: removes $\mathcal{L}_t^{div}$ and $\mathcal{L}_t^{cst}$, and (3) **No planning**: removes the preference-based Monte Carlo planning stage. These variants are compared against the full FOCUS framework, DreamerV3, and DreamerV3-Planning to isolate the impact of our hierarchical switching mechanism.

As shown in Figure 5b, the full FOCUS framework consistently outperforms all other variants, confirming that the synergy between its components is essential for success in high-dimensional tasks. The variant without the high-level policy $\pi_{\phi_1}$ effectively reduces to a version of DreamerV3 that includes planning during both exploration and execution. Interestingly, this variant performs worse than standard DreamerV3. This decline is attributed to exploration bias; without the structured

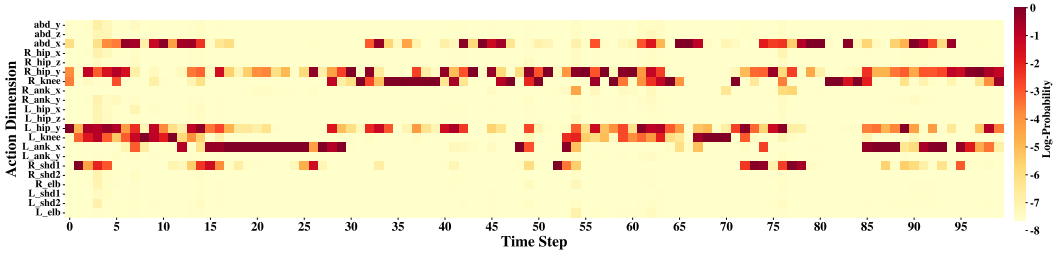


Figure 9: **Temporal evolution of action preferences in Bernoulli distributions over the first** 100 **timesteps.** The high-level preference policy employs a sparse and dynamic selection strategy, with specific action dimensions being selectively activated in response to temporal transitions.

dimension selection provided by the switching policy, the planning process becomes susceptible to noisy value guidance in the expansive, unconstrained action space.

While adding execution-time planning to DreamerV3 improves average performance, it exhibits instability and shows a pronounced vulnerability to local optima when searching directly within flat action spaces. In contrast, FOCUS performs planning at the high-level switching stage. Furthermore, the results demonstrate that the diversity-driven loss and the planning algorithm are both critical for our model. We also compare FOCUS to its Cross Entropy Method (CEM)-based variant in Figure 8a. As illustrated, SMC enables faster and more effective learning compared to CEM. We argue that prior sample-based planning methods which require online distribution refitting are not sufficiently capable of modeling target distributions in a high-dimensional action space and thus limit their effectiveness.

**Hyperparameter analyses.** We examine different hyperparameter combinations for planning horizons $h$ and the number of candidate trajectories $N_p$ in Figure 8b. We find that increasing $N_p$ generally improves overall performance, but does not guarantee a monotonic increase in episode return. We attribute this to omitting the policy log-likelihood term from the advantage estimate, unlike prior Control-as-Inference research Levine (2018); Piché et al. (2018). We find that this term introduces large variance in high-dimensional action spaces, leading to rapid particle degeneration. Removing this term mitigates variance but promotes riskier planning, as actions with larger magnitudes are weighted higher without accounting for their sample probabilities.

### 5.7 Temporal Evolution of Learned Preferences

In Figure 9, we visualize the temporal evolution of the learned preference prompts in Bernoulli distributions. The results show that rather than converging to a static set of actuators, the policy dynamically modulates its focus across different dimensions over time, adapting to time-varying task requirements. This result supports the hypothesis that FOCUS leverages structured sparsity to concentrate exploration on behaviorally relevant subspaces. Furthermore, we visualize the preference prompt selection in Figure 14 and Figure 15 in the Appendix and find it kinematically reasonable.

## 6 Conclusions and Limitations

In this paper, we presented FOCUS (Focused Control Unit Switching), a scalable framework for reinforcement learning in high-dimensional continuous action spaces. FOCUS addresses the critical challenge of exploration dilution by introducing a hierarchical switching mechanism that dynamically prioritizes task-relevant control units. By generating sparse, binary action dimension preference signals, the high-level policy effectively limits the expansive action space into a focused subspace, allowing the low-level policy to optimize behavior with significantly higher sample efficiency. Our extensive empirical evaluations on DeepMind Control, MyoSuite, HumanoidBench, and Grid Preventive Control demonstrate that FOCUS consistently outperforms state-of-the-art flat and hierarchical baselines in both convergence speed and asymptotic performance.

Despite its effectiveness, the quality of the learned preference prompts is closely tied transition model quality, as inaccuracies in imagined rollouts can lead to suboptimal high-level guidance. Furthermore, the hierarchical decision-making mechanism introduces extra parameters and training complexity. Exploring effective preference representations can be a promising direction for future work.

### AI USE STATEMENT

Generative AI tools assisted with language polishing and rephrasing. The authors reviewed the AI-assisted text and take responsibility for the final manuscript, including its claims and citations.

### REPRODUCIBILITY STATEMENT

This work proposes FOCUS for reinforcement learning in high-dimensional continuous action spaces. To facilitate reproducibility, we provide the complete implementation code as part of the supplementary materials. We describe the control-unit switching mechanism, diversity-driven policy learning, and preference-based planning in Sections 4.2–4.4. The experimental setup and evaluation protocol are given in Section 5.1. A complete training algorithm, transition-model details, benchmark descriptions, and additional experimental analyses are provided in Appendices A–D.

## REFERENCES

Nir Baram, Guy Tennenholtz, and Shie Mannor. Action redundancy in reinforcement learning. In *Uncertainty in Artificial Intelligence*, pp. 376–385, 2021.

Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. Myosuite–a contact-rich simulation suite for musculoskeletal motor control. *Proceedings of Machine Learning Research*, 168, 2022.

Yash Chandak, Georgios Theocharous, James Kostas, Scott Jordan, and Philip Thomas. Learning action representations for reinforcement learning. In *ICML*, pp. 941–950, 2019.

Haokun Chen, Xinyi Dai, Han Cai, Weinan Zhang, Xuejian Wang, Ruiming Tang, Yuzhou Zhang, and Yong Yu. Large-scale interactive recommendation with tree-structured policy gradient. In *AAAI*, volume 33, pp. 3312–3320, 2019.

Florian Geißer, David Speck, and Thomas Keller. Trial-based heuristic tree search for MDPs with factored action spaces. In *SoCS*, volume 11, pp. 38–47, 2020.

Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored MDPs. In *NeurIPS*, pp. 1523–1530, 2001.

Christian Gumbsch, Noor Sajid, Georg Martius, and Martin V. Butz. Learning hierarchical world models with adaptive temporal abstractions from discrete latent dynamics. In *ICLR*, 2024. URL https://openreview.net/forum?id=TjCDNssXKU.

Nico Gürtler, Dieter Büchler, and Georg Martius. Hierarchical reinforcement learning with timed subgoals. In *NeurIPS*, volume 34, pp. 21732–21743. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b59c21a078fde074a6750e91ed19fb21-Paper.pdf.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. In *NeurIPS*, volume 35, pp. 26091–26104, 2022.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *Nature*, 640:647–653, 2025.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *ICLR*, 2024.

Kee-Eung Kim and Thomas Dean. Solving factored MDPs with large action space using algebraic decision diagrams. In *PRICAI*, pp. 80–89, 2002.

A Kumar, K Swersky, and G Hinton. Feudal learning for large discrete action spaces with recursive substructure. In *NeurIPS Workshop on Hierarchical Reinforcement Learning*, volume 9, 2017.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Shuzhen Luo, Ghaith Androwis, Sergei Adamovich, Erick Nunez, Hao Su, and Xianlian Zhou. Robust walking control of a lower limb rehabilitation exoskeleton coupled with a musculoskeletal model via deep reinforcement learning. *Journal of neuroengineering and rehabilitation*, 20(1):34, 2023.

Anuj Mahajan, Mikayel Samvelyan, Lei Mao, Viktor Makoviychuk, Animesh Garg, Jean Kossaifi, Shimon Whiteson, Yuke Zhu, and Animashree Anandkumar. Tesseract: Tensorised actors for multi-agent reinforcement learning. In *ICML*, volume 139, pp. 7301–7312, 2021.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored MDPs. In *NeurIPS*, pp. 604–612, 2014.

Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5):1–35, 2021.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gradients. In *NeurIPS*, volume 34, pp. 12208–12221, 2021.

Alexandre Piché, Valentin Thomas, Cyril Ibrahim, Yoshua Bengio, and Chris Pal. Probabilistic planning with sequential monte carlo methods. In *ICLR*, 2018.

Yuta Saito, Jihan Yao, and Thorsten Joachims. Potec: Off-policy learning for large action spaces via two-stage policy decomposition. *arXiv preprint arXiv:2402.06151*, 2024.

Pierre Schumacher, Daniel Haeufle, Dieter Büchler, Syn Schmitt, and Georg Martius. DEP-RL: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems. In *ICLR*, 2023.

Younggyo Seo, Carmelo Sferrazza, Haoran Geng, Michal Nauman, Zhao-Heng Yin, and Pieter Abbeel. FastTD3: Simple, fast, and capable reinforcement learning for humanoid control. *arXiv preprint arXiv:2505.22642*, 2025.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

Sahil Sharma, Aravind Suresh, Rahul Ramesh, and Balaraman Ravindran. Learning to factor policies and action-value functions: Factored action space representations for deep reinforcement learning. *arXiv preprint arXiv:1705.07269*, 2017.

Roland Stolz, Hanna Krasowski, Jakob Thumm, Michael Eichelbeck, Philipp Gassert, and Matthias Althoff. Excluding the irrelevant: Focusing reinforcement learning through continuous action masking. *Advances in Neural Information Processing Systems*, 37:95067–95094, 2024.

Yunhao Tang and Shipra Agrawal. Discretizing continuous action space for on-policy optimization. In *AAAI*, volume 34, pp. 5981–5988, 2020.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Arash Tavakoli, Fabio Pardo, and Petar Kormushev. Action branching architectures for deep reinforcement learning. In *AAAI*, pp. 4131–4138, 2018.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *ICML*, pp. 3540–3549, 2017.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *ICLR*, 2022.

Wenbo Zhang and Hengrui Cai. Where to intervene: Action selection in deep reinforcement learning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=D3au9XkWuy.

Dianyu Zhong, Yiqin Yang, and Qianchuan Zhao. No prior mask: Eliminate redundant action for deep reinforcement learning. In *AAAI*, volume 38, pp. 17078–17086, 2024.

## A Algorithmic Description

We include a complete algorithmic description of the FOCUS training pipeline in Algorithm 1, which integrates RSSM model learning, diversity-driven model-based policy learning, and preference-based SMC planning.

**Algorithm 1** Full training algorithm of FOCUS, with Diversity-driven model-based policy learning, and Preference-based SMC planning

**Initialize:** RSSM $\theta$; hierarchical actor $\{\phi_1, \phi_2\}$; critic $\psi$; replay buffer $\mathcal{B}$ with random episodes.

**while** not converged **do**
  **for** update step $c = 1 \dots C$ **do**
    `// RSSM learning`
    Draw data sequences $\{(o_t, a_t, r_t)\}_{t=1}^{T} \sim \mathcal{B}$.
    Update dynamics model $p_\theta$ and reward model $r_\theta$ using DreamerV3 objectives.
    `// Diversity-driven model-based policy learning`
    Draw a random observation $o_1 \sim \mathcal{B}$.
    **for** time step $t = 1 \dots H-1$ **do**
      Sample $N_c$ preference prompts $\{\mathbf{u}_{t,n}\}_{n=1}^{N_c} \sim \pi_{\phi_1}(s_t)$.
      Generate actions $\{a_{t,n} \sim \pi_{\phi_2}(s_t, \mathbf{u}_{t,n})\}_{n=1}^{N_c}$.
      Predict $\{s_{t+1,n} \sim p_\theta(s_t, a_{t,n})\}$ and $\{r_{t,n} \sim r_\theta(s_{t+1,n})\}$.
      Compute $\mathcal{L}_t^{\text{div}}$ and $\mathcal{L}_t^{\text{cst}}$ via equation 3 and equation 4 in the main text.
      Select $s_{t+1,n}$ with highest $v_\psi(s_{t+1,n})$ as $s_{t+1}$.
    **end for**
    Update $\pi_{\phi_1}$, $\pi_{\phi_2}$, $v_\psi$ via equation 5 in the main text.
  **end for**
  `// Environment interaction with SMC planning`
  $o_1 \leftarrow$ `env.reset()`
  **for** time step $t = 1 \dots T$ **do**
    Compute posterior state $s_t \sim q_\theta(o_t)$.
    Sample $N_p$ preference prompts $\{\mathbf{u}_n\}_{n=1}^{N_p} \sim \pi_{\phi_1}(\cdot \mid s_t)$.
    Initialize trajectories $\{\tau_n\}_{n=1}^{N_p}$ from $\hat{s}_{t,n} = s_t$; weights $w_n = 1$.
    **for** planning steps $h = 0 \dots L-1$ **do**
      Generate $\hat{a}_{t+h,n} \sim \pi_{\phi_2}(\hat{s}_{t+h,n}, \mathbf{u}_n)$.
      Predict $\hat{s}_{t+h+1,n} \sim p_\theta(\hat{s}_{t+h,n}, \hat{a}_{t+h,n})$, $\hat{r}_{t+h,n} \sim p_\theta(\hat{s}_{t+h+1,n})$.
      Compute $A_{t+h,n} = \hat{r}_{t+h,n} + \gamma v_\psi(\hat{s}_{t+h+1,n}) - v_\psi(\hat{s}_{t+h,n})$.
      Update $w_{t+h+1,n} = w_{t+h,n} \cdot \exp(A_{t+h,n})$.
    **end for**
    Sample $\tau_n$ from categorical distribution with weights $\{w_{t+L,n}\}_{n=1}^{N_p}$.
    Select first action $\hat{a}_{t,n}$ in $\tau_n$ as $a_t$.
    $r_t, o_{t+1} \leftarrow$ `env.step`$(a_t)$
  **end for**
  Add $(o_t, a_t, r_t)_{t=1}^{T}$ to replay buffer $\mathcal{B}$.
**end while**

## B Transition Model Details

The DreamerV3 transition model is implemented as a recurrent state-space model (RSSM), which maintains a deterministic hidden state $h_t$ and a stochastic latent variable $z_t$ to encode observations and predict future states. In our main text, we denote the concatenated feature $s_t = [h_t, z_t]$ as the overall state representation. The world model can be formulated as follows:

$$\begin{aligned}
\text{Sequence model:} \quad & h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}), \\
\text{Encoder:} \quad & z_t \sim q_\phi(z_t \mid h_t, x_t), \\
\text{Dynamics predictor:} \quad & \hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t), \\
\text{Reward predictor:} \quad & \hat{r}_t = r_\phi(h_t, z_t), \\
\text{Continue predictor:} \quad & \hat{c}_t = c_\phi(h_t, z_t), \\
\text{Decoder:} \quad & \hat{x}_t = g_\phi(h_t, z_t),
\end{aligned} \tag{8}$$

Table 1: **Overview of action space dimensions for all continuous control tasks in our experiments.**

| Environment | Tasks | Action Space Size ($\mathbb{R}^d$) |
|---|---|---|
| High-dimensional DMC | `Humanoid Stand` | 21 |
| | `Humanoid Walk` | 21 |
| | `Dog Walk` | 38 |
| | `Dog Trot` | 38 |
| MyoSuite | - | 39 |
| Low-dimensional DMC | `Walker Walk` | 6 |
| | `Cheetah Run` | 6 |
| | `Quadruped Walk` | 12 |
| HumanoidBench | - | 56 |
| Grid Preventive Control | IEEE 39-bus | 28 |

where $f_\phi$ is the recurrent transition function modeled by a GRU. $q_\phi$ and $p_\phi$ are implemented with MLPs. We use another MLP with separate heads $r_\phi$ and $c_\phi$ to predict the reward $r_t$ and the episode continuation flag $c_t$, respectively. We use a CNN decoder to reconstruct the observation $x_t$.

The world model is trained by minimizing a variational objective that combines reconstruction losses with KL regularization. In particular, the total loss is written as:

$$\begin{aligned}
\mathcal{L}(\phi) &= \mathbb{E}_{\tau\sim\mathcal{B}}\left[\sum_{t=1}^{T}\left(\mathcal{L}_t^{\text{pred}}+\mathcal{L}_t^{\text{dyn}}+\mathcal{L}_t^{\text{rep}}\right)\right], \\
\mathcal{L}_t^{\text{pred}} &= -\ln p_\phi(x_t \mid h_t, z_t) - \ln p_\phi(r_t \mid h_t, z_t) \\
&\quad - \ln p_\phi(c_t \mid h_t, z_t), \\
\mathcal{L}_t^{\text{dyn}} &= \mathrm{KL}\left(\mathrm{sg}(q_\phi(z_t \mid h_t, x_t)) \,\|\, p_\phi(\hat{z}_t \mid h_t)\right), \\
\mathcal{L}_t^{\text{rep}} &= \mathrm{KL}\left(q_\phi(\hat{z}_t \mid h_t) \,\|\, \mathrm{sg}(p_\phi(z_t \mid h_t, x_t))\right).
\end{aligned} \tag{9}$$

where $\mathcal{L}_t^{\text{pred}}$ is a self-supervised loss that reconstructs $x_t$ and predicts the reward $r_t$ and terminal signal $c_t$; $\mathcal{L}_t^{\text{dyn}}$ is a KL divergence aligning the prior $p_\phi(\hat{z}_t \mid h_t)$ to the posterior $q_\phi(z_t \mid h_t, x_t)$; and $\mathcal{L}_t^{\text{rep}}$ is a reverse KL that regularizes the latent representations.

# C Summary of Experimental Environments

Table 1 summarizes the environmental setups in the DeepMind Control and MyoSuite benchmarks. While FOCUS is designed to tackle high-dimensional problems with large action spaces, its results on low-dimensional tasks highlight its broad applicability.

# D Additional Results

## D.1 Visual Evaluation of Learnt Policies

We evaluate trained policies of different models on the DMC and Myosuite tasks and visualize episode frames for comparison. The full results on `HumanoidWalk`, `DogTrot`, and `KeyTurnHard` are presented in Figure 10, Figure 11 and Figure 12 respectively.

## D.2 Planning Efficiency vs. TD-MPC2

We compare the planning efficiency of FOCUS and TD-MPC2 in Figure 13(Left), measuring the time consumed from receiving an observation to producing an action. The experiments were conducted on a system equipped with an NVIDIA RTX 4090 GPU and a dual-socket Intel Xeon Gold 6240C CPU at 2.60GHz. For a fair comparison, both compared models are implemented in PyTorch 2.4.1, whereas the proposed FOCUS is implemented in JAX for the main text and all other experiments.

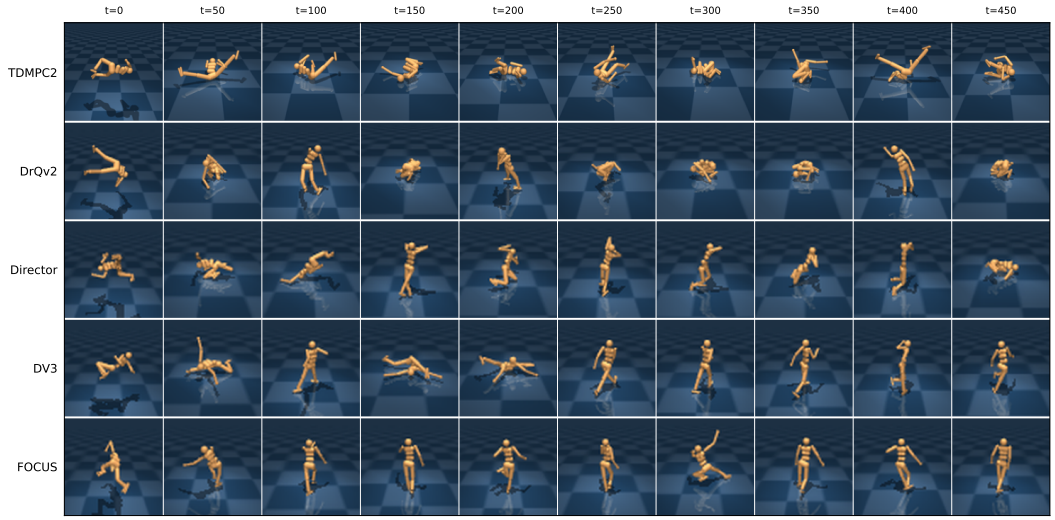


Figure 10: **Demonstration of the learned policy on `HumanoidWalk`.**

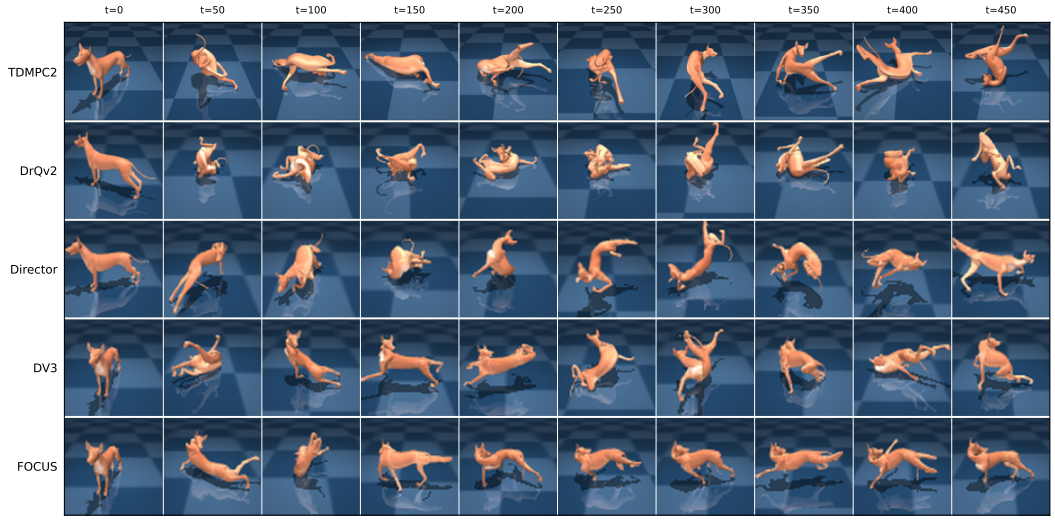


Figure 11: **Demonstration of the learned policy on `DogTrot`.**

Although both methods rely on online planning, TD-MPC2 uses the *Model Predictive Path Integral* (MPPI) algorithm, which is generally more computationally intensive than the proposed Sequential Monte Carlo (SMC) planning employed in FOCUS. As shown, FOCUS achieves significantly lower latency than TD-MPC2, with a notably smaller increase in latency across varying planning horizons. This also highlights the scalability of FOCUS for long-horizon tasks.

### D.3 Applicability to Tasks with Low-Dimensional Action Spaces

A key concern regarding the proposed FOCUS is whether the high-level preference policy might prevent sufficient exploration in environments with small action spaces. To examine this, we evaluate FOCUS on three tasks from the DeepMind Control Suite characterized by low-dimensional action spaces: `Walker Walk`, `Cheetah Run`, and `Quadruped Walk`, while additional task details can be found in Table 1. As shown in Table 2, FOCUS achieves performance comparable to DreamerV3 across these low-dimensional control tasks, demonstrating its broad applicability.

### D.4 Experimental Details on HumanoidBench

**Setup.** The idea of diversity-driven hierarchical policy learning also extends to complex humanoid control tasks that use proprioceptive robot states as input. To validate this insight, we conduct experiments on four tasks from HumanoidBench Sferrazza et al. (2024).

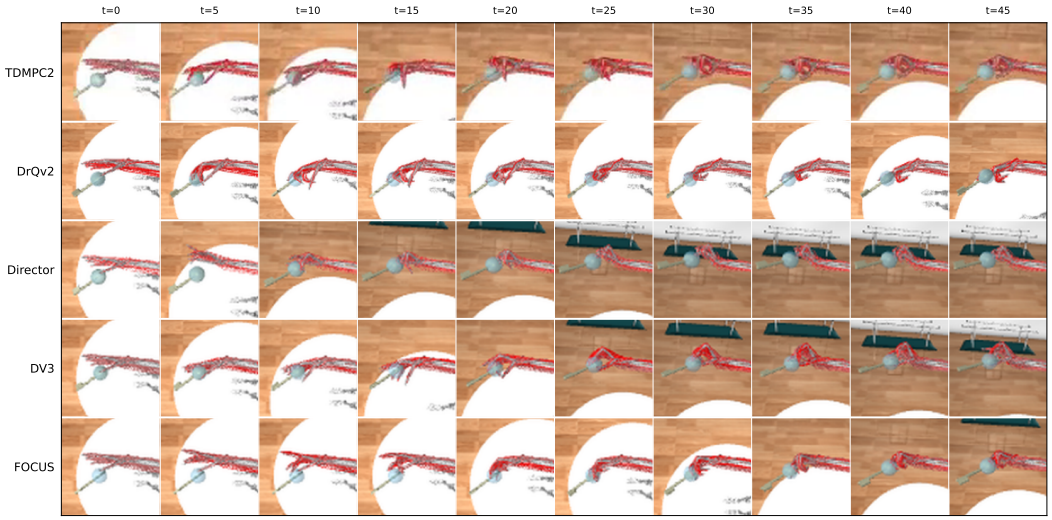


Figure 12: **Demonstration of the learned policy on `KeyTurnHard`.**

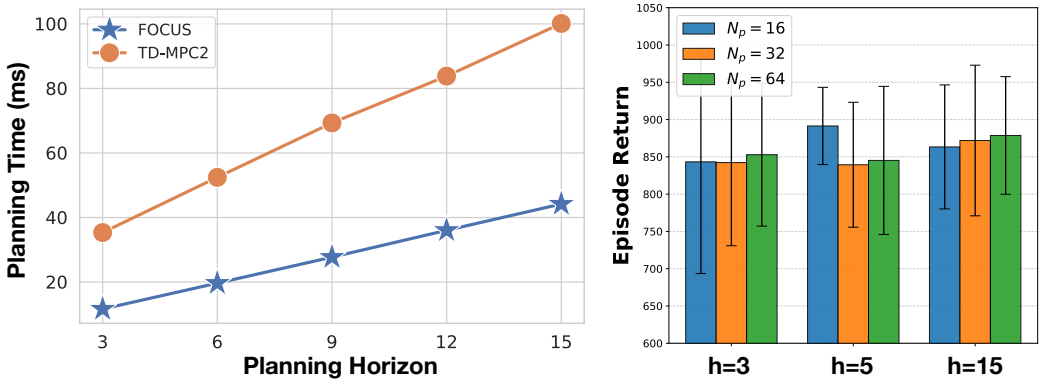


Figure 13: **Left: Comparison of planning efficiency.** We measure the time per interaction step across different planning algorithms. **Right: Further ablation study results on `DogWalk`.** We evaluate the Effect of planning horizon $h$ and candidate trajectories $N_p$ in our planner.

**Model implementation.** On HumanoidBench, standard model-based methods often encounter significant computational overhead. We adopt the training framework of FastTD3 Seo et al. (2025), which has been shown to efficiently solve HumanoidBench tasks through parallel simulation, large-batch updates, a distributional critic, and carefully tuned hyperparameters. We modify FastTD3's policy architecture to incorporate our hierarchical policy model and diversity-driven objectives, resulting in a variant that we refer to as **FastTD3-FOCUS**. Concretely, we implement the high-level preference policy as a 3-layer MLP that outputs Bernoulli logits, while keeping FastTD3's low-level policy architecture unchanged except for the ability to condition on preference prompts. During training, we compute both $\mathcal{L}_t^{\text{div}}$ and $\mathcal{L}_t^{\text{cst}}$ in addition to FastTD3's original policy loss. Since FastTD3 is a model-free algorithm, FastTD3-FOCUS follows the same setting and does not include transition model learning or SMC planning.

### D.5 Hyperparameter Analyses on DogWalk

We further perform a hyperparameter sensitivity analysis on `DogWalk`, complementing the experiments on `HumanoidWalk`, with results shown in Figure 13(Right). As noted earlier, increasing $N_p$ does not necessarily improve performance, since the exclusive computation of the policy log-likelihood limits the advantage of sampling more preference candidates.

Table 2: **Episode return on DMC with low-dimensional action spaces.** The comparable performance to DreamerV3 highlights the applicability of FOCUS across diverse continuous visual control setups.

| Method | Walker Walk | Cheetah Run | Quadruped Walk | Average |
|---|---|---|---|---|
| FOCUS | 973.3±13.5 | 910.1±61.44 | 953.0±8.33 | 945.5 |
| DreamerV3 | 966.4±21.19 | 901.9±26.2 | 943.4±29.1 | 937.2 |
| TD-MPC2 | 332.9±31.9 | 202.7±2.3 | 162.2±49.7 | 232.6 |

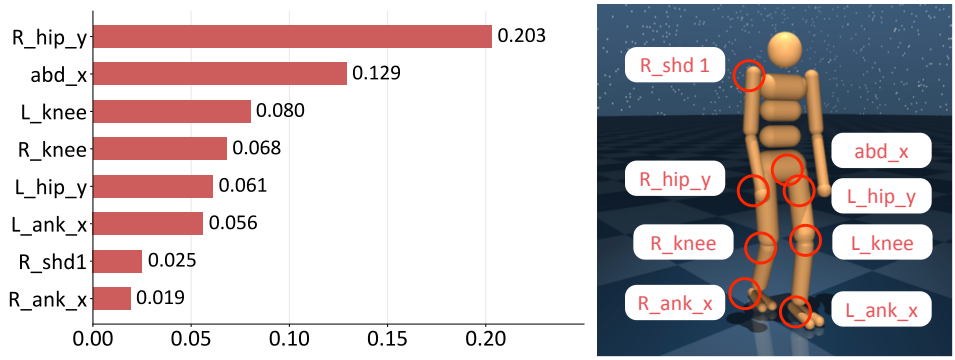


Figure 14: **Visualization of most frequently selected action dimensions.** The subset of action dimensions dynamically switched on by FOCUS aligns closely with the kinematic priorities of the HumanoidStand task. This demonstrates the framework's capacity to autonomously identify and prioritize behaviorally significant control channels within expansive action spaces.

### D.6 Visualization of Preference Frequencies

We further present quantitative and qualitative results regarding the preference prompts sampled on `HumanoidStand`, using the model trained immediately after the rising point (*i.e.*, after 3M environment steps) and recording preference prompts over a single episode. These results serve two purposes: (1) to qualitatively assess whether the learned preference policy captures meaningful, structured sparsity across action dimensions, and (2) to illustrate the temporal evolution of control preferences as the agent attempts to stand.

We visualize the 8 most frequently selected action dimensions on the humanoid embodiment along with their selection frequencies in Figure 14. The full action dimension frequencies are presented in the Appendix, demonstrating that FOCUS successfully captures the structural sparsity within the original action space. Actuators corresponding to these preferred dimensions are highlighted to showcase FOCUS's ability to discover meaningful kinematic correspondences. As illustrated, the learned preferences are strongly concentrated on the lower-body actuators and the core trunk, which are physically critical for the `HumanoidStand` task. Specifically, the hip joints (*e.g.*, *R_hip_y*) and abdomen (*abd_x*) dominate the selection, reflecting their key role in generating upward torque and maintaining core stability necessary for an upright posture. This distribution confirms that FOCUS effectively identifies structural sparsity, facilitating efficient policy learning in a reduced action space.

We also present the full results of action-dimension selection frequency in Figure 15. Actuators beyond the top 8 most frequently selected ones are rarely activated, indicating that the high-level preference policy effectively discovers structural sparsity and reduces the action space.

### D.7 Visualization of Learned Preferences

We visualize the preference prompts sampled by the learned high-level policy in Figure 16, and present the full visualization video in supplementary materials. The results show that our hierarchical policy can selectively prioritize action dimensions corresponding to key joints, such as the legs and arms in the `humanoid` tasks.

For instance, in the second row in Figure 16, the policy focuses on the *left knee* and the *x-actuator at the right ankle* to support the stand-up motion. A closer look at joint control axes reveals that the selected x-actuator at the right ankle is oriented vertically to the ground, making it particularly

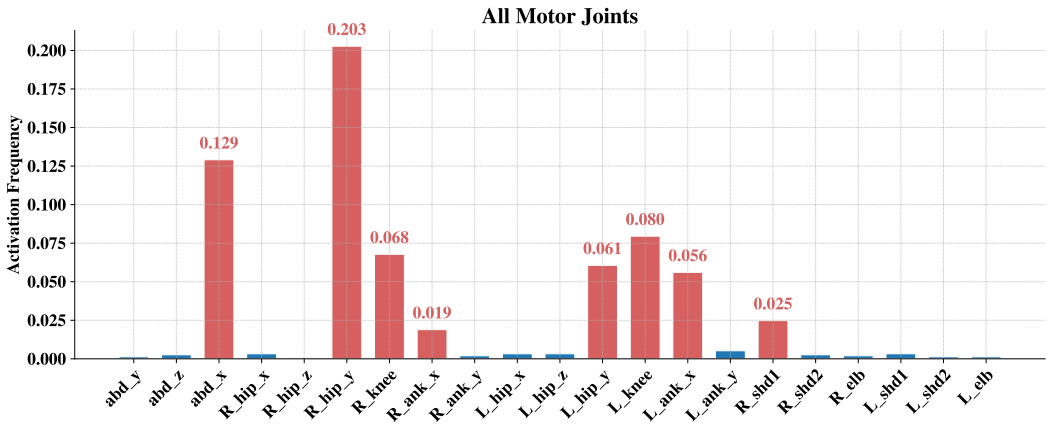


Figure 15: **Action-dimension selection frequencies.** FOCUS effectively leverages structured sparsity to guide focused exploration, supporting our objective of reducing exploration overhead in high-dimensional control tasks.

effective for initiating a standing-up behavior. In the third row, upon detecting that the right lower leg requires adjustment, the policy prioritizes control of the right knee joint to enable stable locomotion.

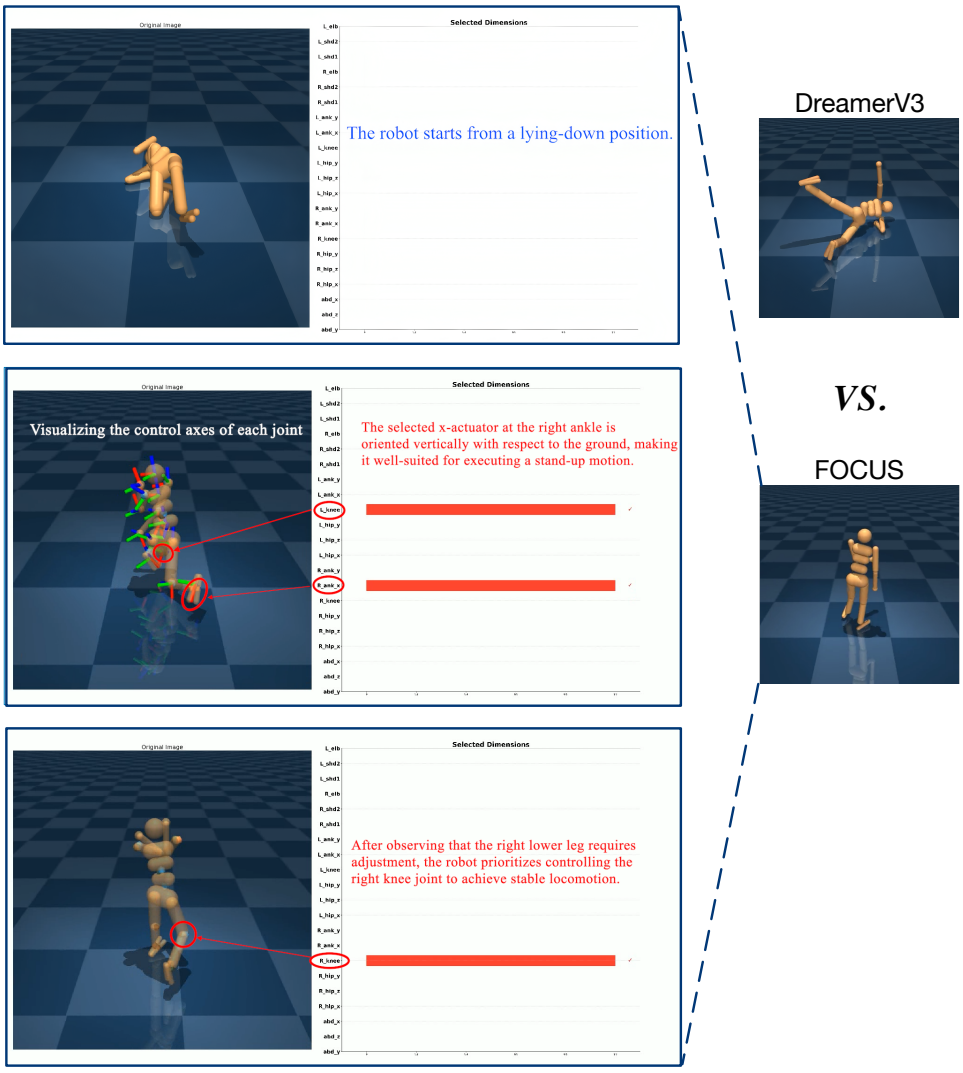


Figure 16: **Visualization of learned action-dimensional preferences.** Red bars indicate the action dimensions selected by the high-level policy. Please refer to the supplementary video for details.