HUMAN-INSPIRED MODULAR REINFORCEMENT LEARNING OF MOVEMENT AND POSTURE FOR MUSCULOSKELETAL CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Controlling upper-body movement while maintaining bipedal balance is a long-standing challenge in high-dimensional musculoskeletal models. Although recent reinforcement learning (RL) algorithms are capable of handling large action spaces, they still struggle here because balancing requires the rest of the body to continuously compensate for arm motion. In human physiology, upright posture is maintained not by a single rigid block but by sets of muscles that link body modules from the feet to the head, counteracting internal perturbations and keeping the center of mass within a narrow base of support. Inspired by human motor control, we propose a modular RL framework that coordinates movement, posture, and equilibrium across body segments. Specifically, we divide the body into arm, torso, and leg modules and train them jointly, with posture imposed as a constraint on the whole-body controller rather than as a reward. This lets the whole-body policy maximize task reward while staying within a stable postural region. In experiments on MyoSuite and MS-Human-700, with control expanded from the arm to the whole body, the resulting policy performs standing and manipulation tasks that demand both stability and mobility. These findings suggest that structuring control along biomechanical lines makes learning in neuromuscular models substantially more tractable.

1 INTRODUCTION

When a person draws on a board, the arm moves along the intended trajectory while the trunk appears to remain still. However, the trunk is not at rest, and the way humans integrate mobility and stability is more complex than it appears. This is because muscles outside the arm continuously compensate for the internal perturbations produced by the arm's movement (Pozzo et al., 2001). These internal perturbations make reinforcement learning (RL) on whole-body musculoskeletal models (Caggiano et al., 2022; Zuo et al., 2024; Li et al., 2026) difficult. Standard approaches control hundreds of muscles with a single policy and encode balance as a reward term alongside the task, so that stability is traded against task performance. As a result, methods that solve a variety of tasks on a single body part fail to maintain stability when applied to whole-body models.

To address this difficulty, we draw on how the human neuromuscular system coordinates movement, posture, and equilibrium (Massion, 1992; Horak, 2006), following three principles. First, movement and posture are controlled by distinct but coordinated processes, organized along the segments of the body (Massion, 1992; Pozzo et al., 2001; Hasegawa & Dubousset, 2022). Second, the postural module not only reacts to disturbances caused by movement but also prepares for the movement in advance and assists it (Bouisset & Zattara, 1981; Le Mouel & Brette, 2017; Stamenkovic et al., 2021). Third, whole-body balance does not require maintaining a specific posture, as long as the body remains within an allowed region (McCollum & Leen, 1989; Hof et al., 2005).

In this work, we integrate these principles into **MM-PoCo** (**M**odular **M**ovement and **Po**stural **Co**ntrol). MM-PoCo divides the policy into modules responsible for movement (arm) and posture (torso and legs). The postural modules observe the modules they support and are trained on the same task reward, so that they anticipate and assist the movement. On top of this, posture is imposed as a constraint through early termination: an episode ends as soon as the body leaves the allowed

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.

region, so that the agent maximizes the task reward alone while remaining inside it. Because MM-PoCo changes only the policy structure and the termination condition, it can be applied to existing RL algorithms without modification.

We apply MM-PoCo to several RL algorithms and evaluate it on four whole-body tasks in MyoSuite (Caggiano et al., 2022) and MS-Human-700 (Zuo et al., 2024). MM-PoCo improves performance in 10 of the 12 combinations of three RL algorithms (Haarnoja et al., 2018; He et al., 2024; Wei et al., 2026) and four whole-body tasks. Averaged over the three algorithms, it raises the success rate on whole-body drawing from 0.7% to 61.9% and the dwell rate on object relocation from 6.2% to 58.7%, and extends how long the body stays upright on both standing tasks. With MM-PoCo, even SAC (Haarnoja et al., 2018), a standard RL algorithm, outperforms algorithms designed for high-dimensional action spaces on the manipulation tasks, raising its success rate on whole-body drawing from 0.1% to 92.5%. Comparing alternative designs for imposing the postural constraint, we find that early termination maintains stability while performing the task better than reward penalties or a Lagrangian method (Altman, 1998; Ha et al., 2020). Finally, without additional training, policies trained with MM-PoCo remain robust to perturbations such as a moving support surface, impacts, and weakened muscles.

Our contributions are as follows.

- We translate principles of human movement and postural coordination into design choices for RL in whole-body musculoskeletal control: functional decomposition, postural adjustments that anticipate and assist the movement, and posture as a feasible region.
- We propose MM-PoCo, which decomposes the whole-body agent into modules over disjoint action subspaces and imposes posture as a constraint through early termination.
- We show that MM-PoCo improves baseline RL algorithms on whole-body tasks across musculoskeletal models, and that the resulting policies remain robust to perturbations unseen in training.

2 Related Work

Reinforcement Learning for Musculoskeletal Control. Musculoskeletal control under RL grew from a single limb to the whole body. Early work posed manipulation on arm and hand models (Berg et al., 2023; He et al., 2024), and recent benchmarks pair an arm or the legs with a torso (Wang et al., 2026a). Locomotion advanced alongside: bipedal walking was learned from biologically plausible objectives (Song et al., 2021; Schumacher et al., 2025), and standing was made possible by a hierarchical control pipeline over a whole-body musculoskeletal model (Ma et al., 2025). Motion imitation drove whole-body models from motion capture (Simos et al., 2025; Li et al., 2026). Exploration is the main obstacle: the action space is high-dimensional and overactuated, so undirected noise rarely finds anything worth learning from. Against it, muscle synergies group actuators that move together (Berg et al., 2023; He et al., 2024), latent and disentangled representations search a lower-dimensional space (Chiappa et al., 2023; Wang et al., 2026b), and embodied or value-guided proposals reshape the exploration noise itself (Schumacher et al., 2023; Wei et al., 2026). We extend manipulation and standing tasks to the whole body, and learn to coordinate mobility and stability through a human-inspired modular control that accounts for posture.

Constrained Reinforcement Learning. RL optimizes a policy to maximize the reward. In a musculoskeletal model, however, movement and posture play different roles: voluntary movement should maximize a task reward, whereas posture must stay within a feasible region to remain stable. To incorporate posture into the RL formulation as a constraint, we consider the constrained Markov decision process (CMDP), which maximizes the return under a budget on the expected discounted cost (Altman, 1999). Trust-region methods keep the policy feasible at every update (Achiam et al., 2017), while classic Lagrangian methods fold the constraint into the objective (Altman, 1998) and have been extended with a PID-controlled multiplier (Stooke et al., 2020) and a worst-case cost constraint (Yang et al., 2021). Early termination with a sufficiently large penalty yields the same optimal policy as the CMDP (Sun et al., 2022), and we adopt this approach. In robotics, constraints replace reward terms for fall prevention (Ha et al., 2020) and constraints define a feasible region for legged locomotion by limiting the probability of undesirable events, such as the trunk touching the ground (Kim et al., 2024). We keep the constraint for upright posture by early termination and a large penalty, holding the body within a narrow region around the desired posture.

3 Problem Formulation and Biological Motivation

In this section, we formulate musculoskeletal control as a goal-conditioned POMDP (Section 3.1) and review how humans coordinate movement and posture while standing (Section 3.2). We then connect this background to RL training and motivate our approach (Section 3.3).

3.1 Problem Setup

We formulate musculoskeletal control as a goal-conditioned partially observable Markov decision process (POMDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{G}, T, R, \gamma \rangle$ (Kaelbling et al., 1998). At timestep t, the model is in state $s_t \in \mathcal{S}$, and the agent receives an observation $o_t \in \mathcal{O}$, a partial view of s_t. An action $a_t \in \mathcal{A} = [0, 1]^m$ is the vector of activations of the m muscles of the model, each actuator controlling a single muscle. A goal $g \in \mathcal{G}$ is a target position in the world frame, and the policy $\pi(a_t \mid o_t, g)$ is conditioned on it. After the action, the model transitions to $s_{t+1} \sim T(\cdot \mid s_t, a_t)$ and returns a reward $r_t = R(s_t, g)$. The objective is the expected discounted return $J(\pi) = \mathbb{E}\left[\sum_{t \geq 0} \gamma^t r_t\right]$ with discount factor $\gamma \in [0, 1)$.

3.2 Background: Human Motor Control

We review how humans coordinate voluntary movement with posture and balance while standing. The review is organized around three questions: what must be maintained while moving, when postural support is prepared, and in which frame posture is held.

Movement, posture, and equilibrium. A voluntary movement performed while standing involves three components (Massion, 1992): the focal *movement* itself; *posture*, the alignment of the trunk and head and the orientation of the body segments with respect to external and internal references; and *equilibrium*, the mechanical condition that keeps the center of mass over the base of support (Horak, 2006). The three are not independent. The trunk supports the arm, and the legs support the trunk, so a rapid arm movement pushes back on the segments that keep the body upright (Pozzo et al., 2001). Equilibrium, moreover, is a condition on a region rather than on a point (McCollum & Leen, 1989; Hof et al., 2005): every state inside the stability region is acceptable, so posture can vary within it while the body remains balanced.

Anticipatory postural adjustments. Posture is prepared before the movement, not only repaired after it. When the arm is raised from standing, trunk and leg muscles become active before the arm muscles contract (Bouisset & Zattara, 1981). These anticipatory postural adjustments (APAs) are distinct from compensatory adjustments, which are feedback responses triggered only after a perturbation has been sensed (Bechet et al., 2025). Nor do APAs merely cancel the perturbation the movement creates: they also help produce the movement itself, so postural activity serves the movement objective rather than stability alone (Le Mouel & Brette, 2017; Stamenkovic et al., 2021).

Reference frames and segmental organization. Posture is held relative to references rather than in absolute coordinates, and these references are multiple: the trunk and head are aligned with respect to gravity, the support surface, the visual surround, and internal references (Horak, 2006), and gaze itself can serve as a reference frame for spatial orientation (Ivanenko & Gurfinkel, 2018). The references are applied segment by segment rather than to the body as a whole: the feet, pelvis, spinal segments, and head are aligned in a single chain, and balance is maintained within a small region around it (Hasegawa & Dubousset, 2022). Movement and posture are thus controlled by distinct but coordinated processes, organized along the segments of the body.

3.3 Motivation

A standard RL approach to the problem in Section 3.1 treats π as a single policy over all m muscles and encodes balance as a reward term alongside the task. We attribute the difficulty of this approach on whole-body agents to three mismatches with how humans coordinate movement and posture in Section 3.2.

Divide the body by function. Humans handle movement and posture with distinct processes organized along the segmental chain. This suggests dividing the policy by body segment, with an arm module responsible for movement and torso (trunk) and leg modules responsible for posture.

Let posture serve the movement. Postural activity precedes the movement and contributes to the task, rather than only reacting to the disturbance afterward. This suggests that each postural module should observe the modules it supports, together with the task goal, and be trained on the same task reward instead of acting as a separate stabilizer.

Treat posture and equilibrium as a feasible region. Human stability does not require a specific posture, as long as the body remains within an allowed region. This suggests imposing posture and equilibrium as a constraint on a feasible region rather than as a reward that pays more the closer the body stays to a reference. When the region of stable upright postures lies well inside the region of equilibrium enabling standing, keeping the posture within its region satisfies equilibrium automatically. We represent this region with a single upper-body landmark as a simple proxy; the multiple, segment-wise references described above suggest richer constraints, which we leave to future work.

4 Modular Movement and Postural Control

Achieving both mobility and stability is challenging because movement continuously perturbs the posture that supports it. Guided by the principles in Section 3.3, we present **MM-PoCo** (Modular Movement and Postural Control), an RL framework that divides the body into segments with distinct functions (i.e., movement and posture) and treats posture as a constraint. In Section 4.1, we first introduce a modular formulation that divides the body into arm, torso, and leg modules and trains them jointly: the arm module produces movement, while the torso and leg modules maintain posture. In Section 4.2, we then formulate the postural requirement as a constraint, so that the policy maximizes task reward while remaining within a stable postural region.

4.1 Action-Subspace Modular Training

Action-Subspace Decomposition. We divide the whole-body agent into three body modules, each with a distinct functional role: an arm module responsible for movement, and torso and leg modules responsible for posture. This division follows the functional organization of posture and movement in the human body (Massion, 1992): the arm carries the end effector, the torso supports the arm and the head, and the legs support the modules above them through the contact of the feet with the ground. Module $k \in \mathcal{K} = \{\mathrm{arm}, \mathrm{torso}, \mathrm{leg}\}$ produces the actions in $\mathcal{A}_k$ (Algorithm 1). The action spaces are disjoint and together form the whole-body action space,

$$\mathcal{A} = \mathcal{A}_{\mathrm{arm}} \times \mathcal{A}_{\mathrm{torso}} \times \mathcal{A}_{\mathrm{leg}}, \qquad a_t = \left(a_t^{\mathrm{arm}}, a_t^{\mathrm{torso}}, a_t^{\mathrm{leg}}\right) \in \mathcal{A}, \tag{1}$$

so every actuator belongs to exactly one module.[1]

Let $o_t^k \in \mathcal{O}_k$ be module k's view of the whole-body state s_t,

$$o_t^{\mathrm{arm}} \subseteq o_t^{\mathrm{torso}} \subseteq o_t^{\mathrm{leg}}, \tag{2}$$

so the arm module sees only its own observation, the torso module sees the arm module's observation together with its own, and the leg module sees all three. We write $o_t = o_t^{\mathrm{leg}}$ for the whole-body observation. This nesting reflects the body's chain from the end effector down to the ground: a module can support the modules above it only if it knows where they are, including how far the arm module has progressed on the task.

The whole-body policy factorizes into the module policies,

$$\pi(a_t \mid o_t, g) = \prod_{k \in \mathcal{K}} \pi_k\left(a_t^k \mid o_t^k, g\right), \tag{3}$$

where $g \in \mathcal{G}$ is the task goal shared by all modules. Each module carries its own actor π_k and critic $Q_k : \mathcal{O}_k \times \mathcal{G} \times \mathcal{A}_k \to \mathbb{R}$, both taking the same inputs o_t^k and g. No module conditions on another module's action, so the effect of the other modules' motion reaches module k only indirectly,

[1] In the standing tasks, the arm module is inactive, and only the torso and leg modules act, so these tasks exercise posture alone.

Algorithm 1 MM-PoCo: Modular Movement and Postural Control

Input: modules $\{\text{arm}, \text{torso}, \text{leg}\}$; state $s_t \in \mathcal{S}$; disjoint action spaces $\mathcal{A}_k$, $k \in \{\text{arm}, \text{torso}, \text{leg}\}$, so that the whole-body action is $a_t = (a_t^{\text{arm}}, a_t^{\text{torso}}, a_t^{\text{leg}})$; nested observations $o_t^{\text{arm}} \subseteq o_t^{\text{torso}} \subseteq o_t^{\text{leg}}$, with $o_t^k \in \mathcal{O}_k$ module k's view of s_t; goal $g \in \mathcal{G}$; dynamics T; reward R; reference landmark $p(s_t)$, its reference position p_0, and threshold ϵ; terminal penalty r_e, charged to the torso and leg modules

Output: controller $\pi = (\pi_{\text{arm}}, \pi_{\text{torso}}, \pi_{\text{leg}})$

initialize π_k, Q_k on $(\mathcal{O}_k, \mathcal{A}_k)$, all $k \in \{\text{arm}, \text{torso}, \text{leg}\}$
while not converged **do**
  $a_t^k \sim \pi_k(\cdot \mid o_t^k, g)$, all $k \in \{\text{arm}, \text{torso}, \text{leg}\}$
  $s_{t+1} \sim T(\cdot \mid s_t, a_t)$, $r_t = R(s_t, a_t, s_{t+1}, g)$, $r_t^k = r_t$, all $k \in \{\text{arm}, \text{torso}, \text{leg}\}$
  if $\|p(s_t) - p_0\| > \epsilon$ **then**
    $r_t^{\text{torso}} \mathrel{+}= r_e$, $r_t^{\text{leg}} \mathrel{+}= r_e$; end episode
  end if
  store $(o_t^k, a_t^k, r_t^k, o_{t+1}^k, g)$ in buffer k
  update Q_k, π_k, all $k \in \{\text{arm}, \text{torso}, \text{leg}\}$
end while
execution: $a_t \sim \pi(\cdot \mid o_t, g)$ at every control step

through the next observation o_{t+1}^k. Module k receives a reward r_t^k built on the shared task reward r_t, with a postural penalty added for the torso and leg modules (Section 4.2), and maximizes its return

$$J_k(\pi) = \mathbb{E}\left[\sum_t \gamma^t r_t^k\right]. \tag{4}$$

The modules are updated together on the same transitions, each changing only its own π_k with the others held fixed, so every module learns to compensate for the other modules as they actually move.

4.2 Constrained Reinforcement Learning for Postural Feasibility

Posture as a Constraint. To maintain postural stability during movement (Section 3.3), we impose the postural requirement as a constraint. This leads to the following constrained problem,

$$\pi^* = \arg\max_{\pi} \; J(\pi) \quad \text{s.t.} \quad J_C(\pi) \le d, \tag{5}$$

where $\pi = (\pi_{\text{arm}}, \pi_{\text{torso}}, \pi_{\text{leg}})$ is the whole-body agent of Section 4.1, $J(\pi) = \mathbb{E}[\sum_t \gamma^t r_t]$ is the task return, $J_C(\pi) = \mathbb{E}[\sum_t \gamma^t c_t]$ is the return of the postural cost c_t, and d is its budget. The cost indicates whether a landmark $p(s_t)$ on the upper body has moved farther than a tolerance ϵ from its reference position p_0,

$$c_t = \mathbf{1}[\|p(s_t) - p_0\| > \epsilon], \tag{6}$$

We refer to the event $c_t = 1$ as a fall. Implementation details, including the landmark, reference position, and tolerance for each task, are given in Appendices B and C.

Constraint via Early Termination. We set $d = 0$, so that no violation is allowed. Because $c_t \ge 0$, $J_C(\pi) \le 0$ holds if and only if the landmark never leaves its tolerance ball, which is exactly the condition enforced by terminating the episode at the first violation. Following Sun et al. (2022), we therefore end the episode at the first step with $c_t = 1$ and assign a terminal reward $r_e < 0$ at that step. Provided a feasible policy exists and r_e is sufficiently negative, this turns (5) into the unconstrained problem

$$\max_{\pi} \; \mathbb{E}\left[\sum_t \gamma^t \left(r_t + r_e\, c_t\right)\right], \tag{7}$$

where $c_t = 1$ only at the terminal step. We charge this terminal penalty only to the torso and leg modules, which maintain posture,

$$r_t^{\text{arm}} = r_t, \qquad r_t^{\text{torso}} = r_t^{\text{leg}} = r_t + r_e\, c_t. \tag{8}$$

Under this assignment, the arm module optimizes the task reward alone and leaves keeping the landmark within its tolerance ball to the torso and leg modules. The arm therefore does not sacrifice its movement for posture. Instead, posture adjusts to support the movement, mirroring the division of labor described in Section 3.2.

5 EXPERIMENTS

We evaluate MM-PoCo on four whole-body tasks built on two musculoskeletal models, MyoSuite and MS-Human-700 (Section 5.1), covering standing, which requires stability alone, and arm manipulation performed while standing, which requires both mobility and stability. We aim to answer three questions:

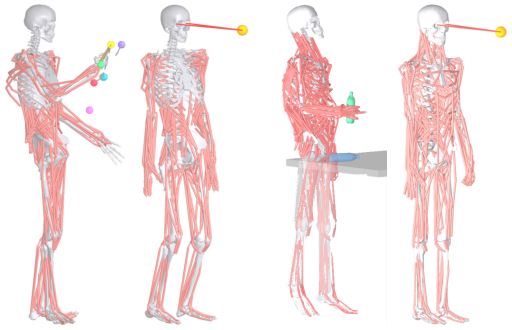
(1) Does modular training outperform a single policy that controls all muscles at once, and does this gain hold across RL algorithms (Section 5.2)?

(2) Which way of imposing the postural requirement is effective (Section 5.3)?

(3) Without further training, does the agent remain stable and keep performing the task under perturbations unseen during training (Section 5.4)?

5.1 EXPERIMENTAL SETUP

Musculoskeletal model. We use two MuJoCo-based musculoskeletal models of the human body: MyoSuite (Caggiano et al., 2022) and MS-Human-700 (Zuo et al., 2024). Both are high-dimensional, overactuated models of the human musculoskeletal system driven by muscle activations rather than joint torques. Like the human body, they have a very large number of degrees of freedom (Bernstein, 1967), and because force is produced by muscle contraction, the same joint torques can arise from many different activation patterns. Their physiological dynamics are also delayed and nonlinear, which, together with the size of the action space, makes motor control particularly difficult. We build the whole-body tasks from the arm, torso, and leg assets provided by the two benchmarks.



	MyoSuite		MS-Human-700	
Module	draw	stand	manipulate	stand
arm	32	—	81	—
torso	210	210	478	478
legs	80	80	100	100
horizon	1050	1000	200	1000

Figure 1: Visualization of four tasks, along with the number of muscles each module controls and the episode horizon.

Tasks. To measure the ability to maintain mobility and stability together, we extend the tasks provided with each musculoskeletal model to the whole body, yielding two tasks on each model. The number of muscles controlled by each module is given in Figure 1, and in all tasks an episode ends when the body falls (Section 4.2) or when the horizon is reached.

In *whole-body standing*, the arm module is omitted, and the body must remain upright.

- **myosuite-stand.** The torso and leg modules control 290 muscles. The evaluation metric is the episode length before a fall.
- **ms-human-stand.** The same task on MS-Human-700, with 578 muscles on a body of a different build and a different distribution of muscles over the segments. The evaluation metric is the episode length, as in **myosuite-stand**.

In *whole-body manipulation*, the body must additionally use the arm to bring an end effector to a target or to relocate an object, while counteracting the perturbations its own arm motion produces.

- **myosuite-draw.** We extend the MyoSuite reaching task (Caggiano et al., 2022) to the whole body, with 322 muscles controlling a body that draws with a stick. Seven targets are sampled from a wide 3D range and presented in sequence, and the task is completed once all seven are reached, which demands rapid arm motion and precise reaching at once. The evaluation metric is the success rate, the fraction of episodes in which all seven targets are reached.

Table 1: Task performance under each RL algorithm, reported as the mean and one standard deviation over three independent runs. Each algorithm alone trains a single policy over all muscles, and the policy trained with MM-PoCo decomposes it into arm, torso, and leg modules. The policy trained with MM-PoCo shows improvement over the single policy.

Method	myosuite-draw success rate (%)	myosuite-stand episode length	ms-human-manipulation dwell rate (%)	ms-human-stand episode length
SAC	0.1 ± 0.0	103 ± 6	0.0 ± 0.0	206 ± 16
SAC + MM-PoCo	$\mathbf{92.5 \pm 1.9}$	$\mathbf{459 \pm 32}$	$\mathbf{62.3 \pm 7.3}$	$\mathbf{808 \pm 102}$
DynSyn	2.1 ± 0.4	436 ± 7	18.5 ± 5.7	848 ± 20
DynSyn + MM-PoCo	$\mathbf{88.4 \pm 2.1}$	$\mathbf{728 \pm 52}$	$\mathbf{69.2 \pm 3.0}$	$\mathbf{893 \pm 18}$
QFLEX	0.0 ± 0.0	126 ± 145	0.0 ± 0.0	11 ± 1
QFLEX + MM-PoCo	$\mathbf{4.8 \pm 4.3}$	11 ± 3	$\mathbf{44.7 \pm 4.3}$	5 ± 0

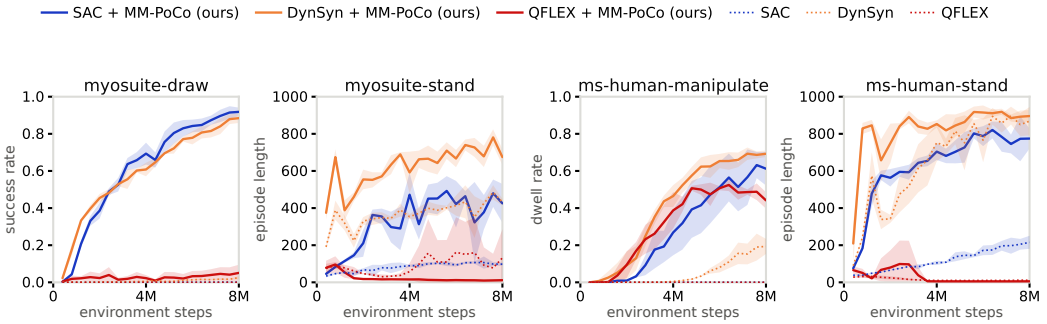


Figure 2: Learning curves on the four whole-body tasks. Solid lines denote the policy trained with MM-PoCo and dotted lines the single policy trained with the algorithm alone. Curves show the mean and one standard deviation over three independent runs. The policy trained with MM-PoCo generally reaches higher performance than the single policy, even on tasks where the single policy rarely succeeds.

- **ms-human-manipulation.** We extend the relocate task of the MS-Human-700 benchmark (Zuo et al., 2024) from the arm alone to the whole body, with 659 muscles. The agent brings an object to a single target and must hold it there for the rest of the episode. The evaluation metric is the dwell rate, the fraction of the episode during which the object stays at the target.

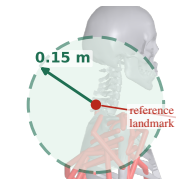
Baselines. We integrate MM-PoCo into the following RL algorithms. For each algorithm, we compare the **policy trained with MM-PoCo** against a **single policy** trained over all muscles with the same task reward and early termination. MM-PoCo replaces this policy with the arm, torso, and leg modules of Section 4.1, each with its own actor and critic, so that the comparison isolates the effect of the modular structure. The algorithms are specified below.

- **Soft Actor-Critic** (SAC; Haarnoja et al., 2018) is an off-policy actor-critic algorithm based on the maximum entropy RL framework. We adapt SAC with a critic ensemble (Chen et al., 2021), the network architecture of BRO (Nauman et al., 2024), and gSDE exploration (Raffin et al., 2022). The details are given in Appendix A.

- **DynSyn** (He et al., 2024) generates synergistic representations of actuators from the dynamical structures of overactuated systems. We build DynSyn on the same SAC configuration without gSDE, following the original paper.

- **QFLEX** (Wei et al., 2026) is the latest method for effective exploration via value-guided flow in the high-dimensional action space. Unlike SAC and DynSyn, QFLEX is built on CrossQ (Bhatt et al., 2024) in JAX (Bradbury et al., 2018), and we follow its implementation.

(a)



(b)

Condition	ET	Reward
task only		$r = R_{\text{task}}$
+ balance		$r = R_{\text{task}} + R_{\text{bal}}$
+ penalty		$r = R_{\text{task}} - 10\,c$
+ Lagrangian		$r = R_{\text{task}} - \lambda\,c$
+ ET	✓	$r = R_{\text{task}}$
+ ET + penalty *(ours)*	✓	$r = R_{\text{task}} - 10\,\mathbf{1}[c > 0]$

(c)

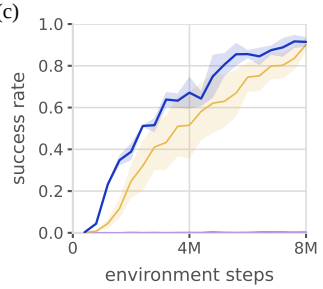


Figure 3: Comparison of six specifications for imposing the postural requirement on *myosuite-draw* task. (a) the postural criterion, a sphere of radius 0.15 m about the reference landmark. (b) the six specifications under a common task reward and criterion; the color chip on each row matches its curve in (c). (c) success rate of the six specifications during training. ET denotes early termination, where the episode ends at the first violation ($c_t = 1$). Reward terms and the cost are defined in Appendix C.

SAC and DynSyn are implemented with Stable-Baselines3 (Raffin et al., 2021), and QFLEX follows its official JAX implementation. Hyperparameters are detailed in Appendix C and all results are averaged across three random seeds.

5.2 Effect of Modular Training

Table 1 and Figure 2 compare a single policy with a policy trained with MM-PoCo. Modular training is incorporated effectively on top of the algorithms and yields consistent improvement. MM-PoCo improves performance in 10 of the 12 combinations of algorithm and task. The gain of modular training is more pronounced in the manipulation task, where a single policy barely performs the task. With a single policy, all three algorithms stay at or below 2.1% on myosuite-draw and 18.5% on ms-human-manipulation. MM-PoCo raises SAC from 0.1% to 92.5% and DynSyn from 2.1% to 88.4% on myosuite-draw. On ms-human-manipulation, it raises SAC from 0.0% to 62.3%, DynSyn from 18.5% to 69.2%, and QFLEX from 0.0% to 44.7%. This implies that modular training effectively handles the difficulty of learning when the rest of the body must continuously compensate for arm motion.

With MM-PoCo, even the standard SAC outperforms the single policies of DynSyn and QFLEX, which are designed for high-dimensional action spaces. On the standing tasks, it is comparable to DynSyn (459 vs. 436, 808 vs. 848). This suggests that the difficulty of tasks requiring manipulation and stability at once lies in the structure of the policy rather than in the learning algorithm.

5.3 Effect of Postural Constraint

To identify an effective way of imposing the postural requirement, we compare six specifications under the same task reward and the same postural criterion. Specifically, we compare no postural term, a dense uprightness bonus, a fixed-coefficient penalty, the same penalty with the coefficient learned by a Lagrangian formulation, early termination, and early termination with a terminal penalty. Figure 3(c) shows that only the two conditions with early termination perform the task. The condition that adds a terminal penalty to early termination reaches a success rate of 92.5%, and the condition with early termination alone reaches 86.2%. The remaining four conditions stay at or below 0.2% throughout training, even when the penalty coefficient is learned by the Lagrangian formulation. This indicates that the postural requirement is effectively imposed by early termination. Adding the terminal penalty makes learning faster. At 2M training steps, its success rate is 41.2%, higher than the 24.0% of the specification with early termination alone. We use this combination as the default setting in all other experiments.

5.4 Robustness to Unseen Perturbations

To evaluate the robustness of a whole-body agent trained with MM-PoCo against perturbations not applied during training, we test the SAC + MM-PoCo agent trained for 12M steps on myosuite-draw under three perturbation scenarios (Figure 4(a)). The scenarios are as follows.

- **Muscle injury.** To model an impaired body, we weaken muscles of the left leg. Following prior work that restricts the left rectus femoris to replicate injury (Ma et al., 2025), we restrict three muscles (rectus femoris, peroneus longus, and adductor magnus) in two modes, giving six variants.
- **Ball strike.** We evaluate an external impact by striking the body with a standard soccer ball at 25 m/s from eight directions.
- **Moving platform.** We evaluate a change in support dynamics by rocking the support surface. The surface tilts by 1.7° in roll and pitch and heaves by 2 cm.

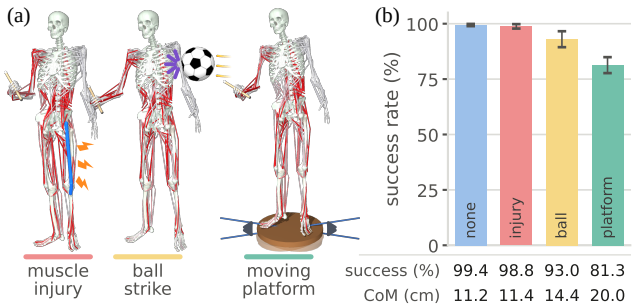


Figure 4: Robustness of SAC + MM-PoCo to perturbations not applied during training. (a) the three perturbation scenarios. (b) success rate and horizontal CoM excursion from the first frame, with and without perturbation, over 200 episodes per condition. Error bars show the spread across 3 seeds.

The details of each scenario are given in Appendix B.

Figure 4(b) reports the success rate and the horizontal excursion of the center of mass (CoM) from the first frame under each scenario, together with the unperturbed case. Each condition is evaluated over 200 episodes and 3 seeds. The CoM excursion grows from the muscle injury to the ball strike to the moving platform, reaching nearly twice the unperturbed value on the moving platform, and the success rate decreases gradually with it. Even on the moving platform, the most disruptive scenario, the agent keeps 82% of its unperturbed success rate, showing that the whole-body agent trained with MM-PoCo maintains stable performance in new environments without further training.

6 Conclusion

In this paper, we introduce MM-PoCo, a modular RL framework for whole-body musculoskeletal control that requires both mobility and stability. Our method divides the actuators by function into an arm module for movement and torso and leg modules for posture, and imposes posture as a constraint that keeps a reference landmark within a stability region, yielding substantial improvement over a single policy on whole-body manipulation tasks across three RL algorithms. MM-PoCo demonstrates whole-body drawing and object relocation while standing on musculoskeletal models, maintaining its performance zero-shot under muscle injury, ball strike, and a moving platform.

The partition into arm, torso, and leg modules is given by hand rather than discovered, and we assume only settings in which the postural modules perform a static task, holding the body upright without a movement of their own. We also represent the stability region with a single upper-body landmark as a simple proxy, and leave the multiple, segment-wise references reviewed in Section 3.2 to future work.

AI use statement. We used large language models for writing assistance and for implementing the experiment and figure-generation code. The authors reviewed and verified all AI-assisted text and code and take full responsibility for the content of this work.

Ethics statement. This work follows the ICLR Code of Ethics. All experiments run in simulation on publicly available musculoskeletal models, with no human or animal subjects and no personal data. The simulated muscle injury is not a clinical model, and the learned controllers are not validated for use on humans.

Reproducibility statement. Appendix A describes the base algorithm, Appendix B the tasks, metrics, and evaluation protocol, and Appendix C the hyperparameters and reward specifications. All results are averaged over three random seeds, and the code will be made publicly available in the future.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, 2017.

Eitan Altman. Constrained Markov decision processes with total cost criteria: Lagrangian approach and dual linear program. *Mathematical Methods of Operations Research*, 1998.

Eitan Altman. *Constrained Markov decision processes*. CRC Press, 1999.

Romain Bechet, Romain Tisserand, Laetitia Fradet, and Floren Colloud. Anticipatory muscle activations to coordinate balance and movement during motor transitions: A narrative review. *Gait & Posture*, 118:130–140, 2025.

Cameron Berg, Vittorio Caggiano, and Vikash Kumar. Sar: Generalization of physiological agility and dexterity via synergistic action representation. *arXiv preprint arXiv:2307.03716*, 2023.

Nikolai A. Bernstein. *The co-ordination and regulation of movements*. Pergamon Press, 1967.

Aditya Bhatt, Daniel Palenicek, Boris Belousov, Max Argus, Artemij Amiranashvili, Thomas Brox, and Jan Peters. CrossQ: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. In *International Conference on Learning Representations*, 2024.

S. Bouisset and M. Zattara. A sequence of postural movements precedes voluntary movement. *Neuroscience Letters*, 22(3):263–270, 1981.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable transformations of Python+NumPy programs. http://github.com/jax-ml/jax, 2018.

Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. Myosuite: A contact-rich simulation suite for musculoskeletal motor control. In *Proceedings of the 4th Annual Learning for Dynamics and Control Conference (L4DC)*, pp. 492–507, 2022.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021.

Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Zixiang Huang, and Alexander Mathis. Latent exploration for reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018.

Sehoon Ha, Peng Xu, Zhenyu Tan, Sergey Levine, and Jie Tan. Learning to walk in the real world with minimal human effort. In *Conference on Robot Learning*, 2020.

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*, 2018.

Kazuhiro Hasegawa and Jean Felix Dubousset. Cone of economy with the chain of balance—historical perspective and proof of concept. *Spine Surgery and Related Research*, 6(4): 337–349, 2022.

Kaibo He, Chenhui Zuo, Chengtian Ma, and Yanan Sui. Dynsyn: Dynamical synergistic representation for efficient learning and control in overactuated embodied systems. In *International Conference on Machine Learning*, 2024.

A. L. Hof, M. G. J. Gazendam, and W. E. Sinke. The condition for dynamic stability. *Journal of Biomechanics*, 38(1):1–8, 2005.

Fay B. Horak. Postural orientation and equilibrium: what do we need to know about neural control of balance to prevent falls? *Age and Ageing*, 35(Suppl 2):ii7–ii11, 2006.

Yury Ivanenko and Victor S. Gurfinkel. Human postural control. *Frontiers in Neuroscience*, 12:171, 2018.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 1998.

Yunho Kim, Hyunsik Oh, Jeonghyun Lee, Jinhyeok Choi, Gwanghyeon Ji, Moonkyu Jung, Donghoon Youm, and Jemin Hwangbo. Not only rewards but also constraints: Applications on legged robot locomotion. *IEEE Transactions on Robotics*, 2024.

Charlotte Le Mouel and Romain Brette. Mobility as the purpose of postural control. *Frontiers in Computational Neuroscience*, 11:67, 2017.

Chengkun Li, Cheryl Wang, Bianca Ziliotto, Merkourios Simos, Jozsef Kovecses, Guillaume Durandau, and Alexander Mathis. Towards embodied ai with musclemimic: Unlocking full-body musculoskeletal motor learning at scale. *arXiv preprint arXiv:2603.25544*, 2026.

Chengtian Ma, Yunyue Wei, Chenhui Zuo, Chen Zhang, and Yanan Sui. Bipedal balance control with whole-body musculoskeletal standing and falling simulations. In *Conference on Robot Learning*, 2025.

Jean Massion. Movement, posture and equilibrium: interaction and coordination. *Progress in Neurobiology*, 38(1):35–56, 1992.

Gin McCollum and Todd K. Leen. Form and exploration of mechanical stability limits in erect stance. *Journal of Motor Behavior*, 21(3):225–244, 1989.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: Scaling for compute and sample-efficient continuous control. In *Advances in Neural Information Processing Systems*, 2024.

Thierry Pozzo, Mourad Ouamer, and Christian Gentil. Simulating mechanical consequences of voluntary movement upon whole-body equilibrium: the arm-raising paradigm revisited. *Biological Cybernetics*, 85(1):39–49, 2001.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Antonin Raffin, Jens Kober, and Freek Stulp. Smooth exploration for robotic reinforcement learning. In *Conference on Robot Learning*, 2022.

Pierre Schumacher, Daniel F. B. Haeufle, Dieter Büchler, Syn Schmitt, and Georg Martius. Dep-rl: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems. In *International Conference on Learning Representations*, 2023.

Pierre Schumacher, Thomas Geijtenbeek, Vittorio Caggiano, Vikash Kumar, Syn Schmitt, Georg Martius, and Daniel F. B. Haeufle. Emergence of natural and robust bipedal walking by learning from biologically plausible objectives. *iScience*, 28(4):112203, 2025.

Merkourios Simos, Alberto Silvio Chiappa, and Alexander Mathis. Reinforcement learning-based motion imitation for physiologically plausible musculoskeletal motor control. *arXiv preprint arXiv:2503.14637*, 2025.

Seungmoon Song, Łukasz Kidziński, Xue Bin Peng, Carmichael Ong, Jennifer Hicks, Sergey Levine, Christopher G. Atkeson, and Scott L. Delp. Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation. *Journal of NeuroEngineering and Rehabilitation*, 18(1):126, 2021.

Alexander Stamenkovic, Lena H. Ting, and Paul J. Stapley. Evidence for constancy in the modularity of trunk muscle activity preceding reaching: implications for the role of preparatory postural activity. *Journal of Neurophysiology*, 126(5):1465–1477, 2021.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, 2020.

Hao Sun, Ziping Xu, Zhenghao Peng, Meng Fang, Taiyi Wang, Bo Dai, and Bolei Zhou. Constrained mdps can be solved by early-termination with recurrent models. In *Advances in Neural Information Processing Systems*, 2022.

Cheryl Wang, Chun Kwang Tan, Balint K. Hodossy, Eric Lyu, Jun Guo, Wentao Zhao, Huaping Liu, Chengkun Li, Merkourios Simos, Bianca Ziliotto, Alexander Mathis, Siyuan Liu, Jiahao Chen, Shanlin Zhong, Bo Jiang, Ci Song, Yaoye Zhu, Chenhui Zuo, Yanan Sui, Mohamed Irfan Refai, Massimo Sartori, Guillaume Durandau, Vikash Kumar, and Vittorio Caggiano. Myochallenge 2025: A new benchmark for human athletic intelligence. *arXiv preprint arXiv:2605.15650*, 2026a.

Yiming Wang, Kaiyan Zhao, Xu Li, Yan Li, Jiayu Chen, Steven Morad, and Leong Hou U. Explore to learn: Latent exploration through disentangled synergy patterns for reinforcement learning in overactuated control. In *AAAI Conference on Artificial Intelligence*, 2026b.

Yunyue Wei, Chenhui Zuo, and Yanan Sui. Scalable exploration for high-dimensional continuous control via value-guided flow. In *International Conference on Learning Representations*, 2026.

Qisong Yang, Thiago D. Simão, Simon H. Tindemans, and Matthijs T. J. Spaan. Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2021.

Chenhui Zuo, Kaibo He, Jing Shao, and Yanan Sui. Self model for embodied intelligence: Modeling full-body human musculoskeletal system and locomotion control with hierarchical low-dimensional representation. In *IEEE International Conference on Robotics and Automation*, pp. 13062–13069, 2024.

A Base Algorithm

SAC and DynSyn share one base algorithm, a SAC variant described below; QFLEX keeps its own. Table 2 compares the three.

SAC variant.

- **Objective.** An off-policy maximum-entropy actor-critic, which augments the return with the policy entropy and is the standard choice for continuous control (Haarnoja et al., 2018).
- **Critic target.** The clipped double-Q target, which suppresses the overestimation caused by maximizing over a noisy value estimate (Fujimoto et al., 2018).
- **Critic ensemble.** Enlarged beyond two critics and aggregated by its minimum, which permits far more gradient updates per environment step without divergence (Chen et al., 2021). Table 9 gives the ensemble size of every network.

- **Architecture.** BroNet for both the actor and the critic; scaling the critic's capacity, rather than the actor's, is what converts the extra updates into sample efficiency (Nauman et al., 2024).
- **Exploration.** Generalized state-dependent exploration (gSDE) (Raffin et al., 2022), with the exploration noise resampled once per episode.
- **Implementation.** Stable-Baselines3 2.7 (Raffin et al., 2021).
- **With MM-PoCo.** Each module carries its own actor and critic of this algorithm, and the modules are trained together as in Algorithm 1.

Table 2: The three RL algorithms. Hyperparameters are given in Appendix C.

	SAC	DynSyn (He et al., 2024)	QFLEX (Wei et al., 2026)
Base algorithm	SAC variant (above)	SAC variant (above)	CrossQ (Bhatt et al., 2024)
Synergy groups	—	Table 7	—
Exploration	gSDE	gSDE disabled, as in He et al. (2024)	value-guided flow
Implementation	Stable-Baselines3	Stable-Baselines3	JAX
Hyperparameters	Table 6	Table 6	Table 8
With MM-PoCo	one agent per module		

B Environments, Metrics, and Protocol

Tasks. Table 3 defines the four tasks. Figure 1 gives the number of muscles each module controls, and Table 9 gives the observation and action dimensions of each module.

- All environments run in MuJoCo with a physics timestep of 2 ms and Hill-type muscle actuators.
- In the standing tasks, the arm module is inactive and the arms hang passively.
- In myosuite-draw and ms-human-manipulation, the left arm is inactive and hangs passively.
- MS-Human-700 manipulation keeps the control period of its original configuration, which DynSyn also uses (He et al., 2024).

Table 3: Details of the four whole-body tasks.

	draw	*stand*	*manipulation*	*stand*
Model	MyoSuite	MyoSuite	MS-Human-700	MS-Human-700
Active modules	arm, torso, leg	torso, leg	arm, torso, leg	torso, leg
Control period	20 ms (frame skip 10)	20 ms (frame skip 10)	10 ms (frame skip 5)	20 ms (frame skip 10)
Scored point	tip of a 20 cm stick	focal point, 0.5 m in front of the eyes	the object	focal point, 0.5 m in front of the eyes
Horizon	at most 1050 steps	1000 steps	200 steps	1000 steps
Other termination	none	none	object dropped or out of bounds; pelvis or head below 0.7 of its standing height	pelvis or head below 0.7 of its standing height
Metric	success rate	episode length	dwell rate	episode length
Landmark	C7	C7	mastoid	mastoid
r_e	-10	-10	-100	-10

Postural criterion.

- **Landmark.** The C7 site on MyoSuite and the mastoid site on MS-Human-700.
- **Reference position.** p_0 is the landmark position at reset. Both $p(s_t)$ and p_0 are measured relative to the pelvis position, so that moving the whole body is not counted as a postural failure.
- **Drift and cost.** The drift is $\Delta_t = \|p(s_t) - p_0\|$, and the cost of Equation (6) is $c_t = \mathbf{1}[\Delta_t > \epsilon]$.
- **Tolerance.** $\epsilon = 0.15$ m on every task.
- **Terminal penalty.** $r_e = -10$ on every task except MS-Human-700 manipulation, where $r_e = -100$ (to match the scale of its own task reward).

Metrics. The four metrics are not comparable to one another.

- **Success rate** (MyoSuite *draw*): the fraction of the sequential targets reached.
- **Dwell rate** (MS-Human-700 manipulation): the fraction of the episode during which the object is within $0.02\,\mathrm{m}$ of the target position and 0.1 of the target orientation.
- **Episode length** (standing tasks): the number of steps before the episode terminates, out of 1000; it measures how long the posture is held.

Training and reporting.

- Every condition is trained for 8×10^6 environment steps with three seeds.
- Table 1 and Figures 2 and 3 report the metric of the episodes collected during training.
- A reported value is the mean of that metric over the last 200k steps of training within a seed, then the mean and standard deviation across the three seeds.
- Deterministic evaluation (Appendix C) is used only to select the checkpoint for Section 5.4.

Zero-shot perturbations. Section 5.4 evaluates the best checkpoint of SAC + MM-PoCo trained on *draw* for 12×10^6 steps, without further training. Muscle injury weakens one of three left-leg muscles (rectus femoris, peroneus longus, or adductor magnus) in one of two modes, limiting its excitation to below 0.007 or halving it, which gives six variants. Ball strike launches a $0.45\,\mathrm{kg}$ ball at $25\,\mathrm{m/s}$ toward the torso from one of eight directions. Moving platform rocks the support surface in roll and pitch by 1.7° and heaves it by $2\,\mathrm{cm}$.

C Experimental Details

Hyperparameters.

- **SAC and DynSyn.** Table 6. Both use the base algorithm of Appendix A.
- **DynSyn only.** Table 7. The synergy groups of MS-Human-700 are those released with DynSyn (He et al., 2024); on MyoSuite, the torso and arm groups are extracted for this work with the procedure of DynSyn.
- **QFLEX.** Table 8. The network sizes and the remaining hyperparameters follow Tables 1 and 2 of Wei et al. (2026), and the replay ratio and the observation normalization match those of SAC on each benchmark.
- **Network sizes.** Table 9. On MyoSuite, the actor and the critic are 512 wide, except for the *draw* arm module at 64.

Task rewards. Table 4 gives the task reward of each task.

- The arm module receives the task reward alone; the torso and leg modules receive the task reward together with the postural terms, as in Equation (8).
- **Standing tasks.** The goal is to stand for as long as possible, which the postural criterion of Appendix B enforces. A focal target is added as a guide: the focal point, fixed to the head $0.5\,\mathrm{m}$ in front of the eyes, is rewarded for staying on a target that switches every 100 steps.
- **Manipulation.** The reward is that of the original MS-Human-700 manipulation environment, with the weights of its released configuration, unchanged (Zuo et al., 2024).

$$r = 50\,e^{-10\|\Delta p\|}\cdot 10\,e^{-2\|\Delta\theta\|} + 30\,e^{-10\,\bar{d}} + 2\cdot\mathbf{1}[\,z - z_0 \geq 0.02\,\mathrm{m} \wedge \neg\,\mathrm{drop}\,] - \frac{\|a\|}{n_a} - 100\cdot\mathbf{1}[\mathrm{done}] \quad (9)$$

- Δp and $\Delta\theta$: the position and the tilt (R_x, R_y) of the object relative to the target.
- $\bar{d}$: the mean distance from the 14 finger segments to the object; $\mathrm{drop} = \mathbf{1}[\bar{d} > 0.3\,\mathrm{m}]$.
- $z - z_0$: the lift of the object above its initial height.
- n_a: the number of actuators; done: the object is dropped or leaves the workspace.

Table 4: Task rewards, paid every step. d is the distance from the scored point of Table 3 to the current target.

Task	Reward	Targets
draw	$-0.1\,d + 10 \cdot \mathbf{1}[d \le 0.0125\,\mathrm{m}]$, the bonus paid once per target	7 targets, uniform in the box $x \in [-0.40, 0]$, $y \in [-0.40, -0.20]$, $z \in [1.10, 1.50]$ m; the next target is shown when the current one is reached or after 150 steps
stand (both models)	$-0.1\,d + 0.1 \cdot \mathbf{1}[d \le 0.0125\,\mathrm{m}]$	7 targets, uniform in a 0.05 m ball about the focal point in the nominal pose; the target switches every 100 steps, and the last one is held until step 1000
manipulation	Equation (9)	one position and orientation of the object, sampled about its initial position

The six postural specifications. Table 5 gives the six conditions of Section 5.3 on *draw*.

- All six share the task reward of *draw* and the 0.15 m threshold on the C7 drift Δ_t, and they differ in how the postural requirement enters.
- The per-step conditions use the violation depth $v_t = \max(0,\ \Delta_t - 0.15)$, which is zero exactly when $c_t = 0$.
- The uprightness bonus uses θ_f, θ_s, the sagittal and coronal tilt of the torso relative to its orientation at reset.

Table 5: The six postural specifications on *draw*. The postural terms go to the torso and leg modules only.

Condition	Postural term	Early termination
task only	none	no
+ balance	$+0.1 \exp\left(-3(\theta_f^2 + \theta_s^2)\right)$ every step	no
+ penalty	$-10\,v_t$ every step	no
+ Lagrangian	$-\lambda\,v_t$ through the cost critic, λ learned	no
+ ET	none	when $c_t = 1$
+ ET + penalty (ours)	$r_e = -10$ once, on the terminating step	when $c_t = 1$

Table 6: Hyperparameters of SAC, shared by DynSyn. Rows above the first rule are shared by both benchmarks, and the rest follow each benchmark's own published configuration. Updates and the replay ratio are per module. DynSyn runs with gSDE disabled.

	MyoSuite	MS-Human-700
Algorithm	SAC with a BroNet-style actor and critic	
Network	two BroNet-style blocks, three Linear + LayerNorm each	
Activation	SiLU, actor *and* critic	
Critic target	minimum over the critic ensemble	
Discount / τ	0.99 / 0.005	
Entropy coefficient	learned, target entropy $= -\lvert A\rvert$	
Critic weight decay	0	
Total steps	8×10^6	
Learning rate	10^{-3} constant	linear decay from 10^{-3} to 2.5×10^{-4}
Batch / replay buffer	512 / 10^6	512 / 10^6
Uniform-action steps	10 000	10 000
Parallel envs	1	8
Env steps : updates	2 : 1	16 : 4
Replay ratio (UTD)	0.5	0.25
gSDE	on, one draw/episode	on, one draw/episode
Obs. normalization	none	running mean/var, clip 10
Reward normalization	none	none
Evaluation	every 25 000 steps, 5 episodes, deterministic	
Checkpoint selection	return, or episode length on the standing tasks	

Table 7: DynSyn settings. A module uses the groups of its own muscles, and a single policy uses the groups of all modules together. The other hyperparameters are those of Table 6.

	MyoSuite	MS-Human-700
Groups / muscles	torso 25 / 210 and arm 16 / 32, extracted for this work; leg 40 / 80, released with DynSyn	arm 40 / 81, torso 48 / 478, leg 44 / 100, all released with DynSyn
Initial log std	0.5	
Weight amplitude	0 until 10^6 steps, then linear to 0.1 at $3{\times}10^6$ steps	
Target entropy	minus the number of groups	
gSDE	disabled	

Table 8: Hyperparameters of QFLEX. Rows above the first rule follow Tables 1 and 2 of Wei et al. (2026) and are shared by both benchmarks, and the rest match the replay ratio and observation normalization of SAC on each benchmark. With MM-PoCo, each module carries its own QFLEX agent, and updates and the replay ratio are per module.

	MyoSuite	MS-Human-700
Implementation	official JAX implementation, commit `a028cfa`, unmodified	
Hidden layers	3 for the policy, the critic, and the flow	
Hidden width	1024, except 256 for the arm module of MM-PoCo	
Discount	0.99	
Optimizer / learning rate	Adam / 3×10^{-4}	
Batch / replay buffer	256 / 10^6	
Flow steps	20	
Gradient ascent steps / step size	20 / 0.01	
Batch normalization decay / ϵ	0.99 / 10^{-5}	
Warm-up steps	30 000	
Total steps	8×10^6	
Parallel envs	1	8
Env steps : updates	2 : 1	16 : 4
Replay ratio (UTD)	0.5	0.25
Obs. normalization	none	running mean/var, clip 10

Table 9: Network sizes of SAC and DynSyn against the dimensions they cover.

Task	Policy	obs	act	critic input	actor	critic	n_Q	width /input
MyoSuite whole body								
draw	single policy	212	322	534	512	512	3	0.96
draw	arm	47	32	79	64	64	2	0.81
draw	torso	123	210	333	512	512	3	1.54
draw	leg	212	80	292	512	512	3	1.75
stand	single policy	290	290	580	512	512	5	0.88
stand	torso	201	210	411	512	512	5	1.25
stand	leg	290	80	370	512	512	5	1.38
MS-Human-700								
manipulation	single policy	2526	659	3185	1024	1024	3	0.32
manipulation	arm	394	81	475	512	512	3	1.08
manipulation	torso	2098	478	2576	1024	1024	3	0.40
manipulation	leg	2526	100	2626	1024	1024	3	0.39
stand	single policy	2514	578	3092	1024	1024	5	0.33
stand	torso	2086	478	2564	1024	1024	5	0.40
stand	leg	2514	100	2614	1024	1024	5	0.39

Every network is two BroNet-style blocks deep, so only the width varies. “act” is what the policy writes; the critic sees the observation and that action, and the last column is the critic’s width over the size of what it reads.