Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Morphogenetic Neural Plasticity for Adaptive Robotic Systems

Raunak Shrestha

raunakstha007@gmail.com

The British College https://orcid.org/0009-0002-0892-4032

Research Article

Keywords: morphogenetic robotics, neural architecture search, coevolution, modular robotics, embodied AI, differentiable NAS

Posted Date: April 21st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9465214/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: The authors declare no competing interests.

Abstract

We introduce MNPRAS (Morphogenetic Neural Plasticity for Robotic Adaptive Systems), a framework in which a robot's physical joint topology and neural control architecture are cooptimised as a single unified problem rather than as sequential independent design decisions. The coupling mechanism is the MorphoCode a 64dimensional latent vector that encodes the body graph using a contrastivelytrained Graph Isomorphism Network, with architecture information entering implicitly through the performance signal. When the robot's physical configuration changes, the updated MorphoCode retrieves a warmstart from a prebuilt corpus, reducing postreconfiguration adaptation time by up to 3.8× compared to coldstarting. Physical adaptation uses a modular platform with electromagneticallyactuated coupling interfaces completing engagement in under 420 ms. Across 14 simulated environments and 3 realworld task deployments, MNPRAS achieves 63.5% higher mean episode reward than the strongest fixedmorphology baseline and reaches competent performance 2.6× faster than that baseline. Critically, trained agents reconfigure their bodies an average of 80 ms before environmental transitions rather than in response to them a predictive adaptation behaviour that emerges from reward alone, without explicit anticipation supervision.

1. Introduction

Every biological body and its controlling nervous system arrived at their current forms together. The eightlegged architecture of an arachnid did not precede its motor programmes; the diffuse ganglionic distribution of a cephalopod was not retrofitted onto a body built for some other purpose. Coadaptation between morphology and neural organisation is the rule in biology, not the exception. Robotics has, by contrast, long accepted a division of labour: mechanical engineers produce a fixed chassis and control researchers optimise policies for whatever body they receive. This division is practical but carries a hidden cost the controller bears the full burden of any mismatch between what the body can naturally do and what the task demands.

Two independent lines of recent progress make it tractable to revisit this assumption. Modular robotic platforms have matured to the point where physical topology changes severing a limb, repositioning it, reattaching it at a different port can be executed in seconds without human intervention [1]. Separately, gradientbased neural architecture search has reduced the cost of finding good network topologies from days of random search to minutes of differentiable optimisation [11]. Neither line of work, however, has asked whether physical reconfiguration and architectural adaptation might be unified rather than sequenced.

MNPRAS answers that question. Its organising principle is that a robot's body topology and its neural architecture are two levels of a single optimisation problem, connected through a shared latent representation we call the MorphoCode. When the body changes, the new MorphoCode retrieves a warmstart for the neural controller, reducing adaptation cost proportionally to how much the body changed rather than requiring full retraining. The two levels are coupled rather than alternating.

Our specific contributions are: (1) the MNPRAS framework with a formal bilevel coevolutionary objective and the MorphoCode latent space that couples its two levels; (2) a modular hardware platform with electromagneticallyactuated coupling interfaces capable of sub420 ms reconfiguration; and (3) experimental evidence that trained agents exhibit predictive reconfiguration they restructure their bodies approximately 80 ms before environmental transitions rather than in response to them, a behaviour absent in all comparison baselines.

2. Related Work

2.1 Modular SelfReconfiguring Robots

The goal of a robot capable of altering its own physical topology has been pursued across several hardware paradigms. Latticebased systems arrange uniform cubic modules in a grid, achieving reconfiguration by sliding modules along shared faces [1, 2]. Chainbased platforms connect modules in kinematic chains and reconfigure by detaching and reattaching links [5]. More recent designs prioritise smooth structural change: the Mori3 platform [4] uses origamiinspired folding to transition between planar and volumetric configurations, while SMORES [8, 9] employs lowforce magnetic docking. Across all these platforms, reconfiguration is treated as a planning problem with a fixed controller. MNPRAS treats reconfiguration and architectural adaptation as jointly learnable.

2.2 Neural Architecture Search

Early NAS methods framed controller design as combinatorial search and applied reinforcement learning or evolutionary strategies to navigate it [10, 12]. Differentiable relaxations continuous architecture weights optimised by gradient descent [11] reduced computational cost by orders of magnitude and enabled architecture search to run alongside policy training [13]. Hardwareaware NAS extended these ideas to account for inference constraints on specific accelerators [14]. None of these works considers the physical substrate as a codesign variable; MNPRAS incorporates body topology directly into the NAS search space through the MorphoCode.

2.3 Morphological Evolution in Simulation

Cooptimisation of body and controller in simulation has a history extending to landmark studies in the 1990s [15, 16]. Neuroevolution methods evolve body parameters alongside network weights using fitnessproportionate or noveltybased selection [19, 20]. Recent differentiable simulation frameworks allow gradientbased cooptimisation [17], but the resulting designs do not map directly to physical hardware without fabrication steps that reintroduce designtime bottlenecks. MNPRAS works with a fixed module vocabulary and a graph representation that compiles directly to a running simulation and, without modification, to a physical assembly.

2.4 CoDesign of Body and Controller

The closest prior work treats morphology and control as a joint design problem but optimises them sequentially [36, 37, 39]. MetaMorph [40] trains a single transformer controller across diverse fixed morphologies but does not learn when or how to change the body. EvoGym [41] provides a simulation benchmark for softrobot cooptimisation but has no physical realisation path. The key distinction of MNPRAS is the MorphoCode: a pretrained latent space that makes the two levels of the hierarchy aware of each other and enables warmstarting across physical reconfigurations.

3. Methodology

3.1 Problem Formulation

A robot’s physical configuration is represented as a directed acyclic graph G = (V, E). Each node v carries a feature vector $x_v \in \mathbb{R}^{d}$ encoding module type as a threedimensional onehot vector, halfextent dimensions in metres (3 values), and scalar mass in kilograms. Each directed edge (u → v) encodes that module u provides an attachment port to which module v connects, together with hinge joint angle limits $(\theta_{lo}, \theta_{hi}) \in \mathbb{R}^{2}$.

The MorphoCode $z \in \mathbb{R}^{d_z}$ encodes the body graph G through a learned encoder f_enc with parameters θ_enc. Architecture information enters the coevolutionary loop implicitly through the performance signal that drives the slowtimescale NAS update rather than being directly encoded in z:

$$z = f_{enc}(G ; \theta_{enc})$$

The coevolutionary objective maximises expected task return across an environment distribution p(ε):

(G*, A*) = argmax□,□ □[ε ~ p(ε)] [R(τ | G, A, ε)]

Optimising this directly is intractable because the search space mixes discrete graph topology, discrete architecture choice, and continuous controller weights without a shared gradient structure. We decompose it into a twotimescale scheme that separates fast continuous updates from slow discrete ones while preserving their coupling through the MorphoCode.

3.2 Physical Platform

3.2.1 Module Design

Each module is an aluminiumframe rigid unit measuring approximately 8×8×8 cm, equipped with four electromagneticallyactuated docking faces labelled front, back, left, and right. Docking completes in 180420 ms; passive springloaded detachment completes in under 40 ms. Each module carries a 6axis IMU and a joint torque sensor logging at 200 Hz over a wired CAN bus [47]. Three module categories populate the MorphoSpace: torso (15×8×5 cm, 1.0 kg, four ports), limb (6×3×3 cm, 0.3 kg, two ports), and foot (4×4×2 cm, 0.1 kg, one port). Feet are structural leaves they connect to at most one parent and expose no child ports.

3.2.2 Reconfiguration Policy

The reconfiguration policy π_morph is a graph neural network receiving the body graph G, MorphoCode z, and a scalar performance signal s_t. It produces a binary trigger and, if triggered, a graph edit from the set {ADD(parent, port, child_type), REMOVE(parent, child), NO_OP}. Two rounds of sumaggregation message passing compute node representations; mean pooling produces a graph summary; a twolayer MLP fusing the summary with z and s_t generates both outputs. A differentiable feasibility mask eliminates proposals that would produce disconnected graphs, graphs with cycles, graphs below three modules, or graphs with port conflicts.

3.2.3 Performance Signal

The trigger gate receives a standardised performance signal:

s_t = (mean(W_recent) mean(W_base)) / (std(W_base) + ε)

where W_recent covers the 20 most recent episodes, W_base covers the 100 most recent, and ε = 10□□. Values below 0.15 indicate performance degradation exceeding 15% of a

standard deviation below the rolling average, the primary condition under which the trigger gate activates.

3.3 MorphoCode Latent Space

3.3.1 Graph Encoder

The encoder uses four Graph Isomorphism Network (GIN) layers. GIN was selected because its discriminative capacity matches the WeisfeilerLeman graph isomorphism test the ceiling achievable by any messagepassing network operating on local neighbourhood structure [21]. This property is necessary for our setting: two robots with different body graphs must receive distinct MorphoCodes, or warmstart retrieval degrades to random selection. Each GIN layer applies:

h_v□□□ = MLP□□□((1 + ε□□□) · h_v□□□□□ + Σ□□□(v) h_u□□□□□)

Jumping Knowledge pooling concatenates meanpooled outputs from all four layers plus the input projection before a threelayer projection MLP. The final representation is L2normalised to the unit hypersphere, making cosine distance a wellcalibrated proximity measure across all MorphoCodes.

3.3.2 Contrastive Training

The encoder trains on a corpus of 50,000 (G, score) pairs, where score is cumulative reward from a short PPO evaluation run. For each batch item i, its positive pair j* is the item with the most similar score:

j* = argmin□≠□ | score□ score□ |

The NTXent loss pulls positive pairs together and pushes all other pairs apart on the hypersphere with temperature τ = 0.07:

L = (1/B) · Σ□ log [exp(z□·z□*/τ) / Σ□≠□ exp(z□·z□/τ)]

3.3.3 WarmStart Retrieval

After each reconfiguration event, the new body G′ is encoded to z′. The corpus entry maximising a weighted combination of latent proximity and performance is retrieved:

score_combined(j) = 0.7 · cos(z′, z□) + 0.3 · score_norm□

The weighting 0.7/0.3 was selected by grid search on a validation set (see Appendix B.6 for sensitivity analysis). Compatible weight tensors from the retrieved entry initialise the PPO policy, and training resumes from this point. This reduces postreconfiguration stepstocompetence by up to 3.8× compared to coldstarting from a random initialisation (Ablation A4, Section 4.6).

3.4 Neural Architecture Search

3.4.1 PerModule Search Space

Each module i is assigned a local operation from four candidates: linear projection (Op 0), ReLUgated linear (Op 1), Tanhgated linear (Op 2), and a twolayer batchnormalised block (Op 3). Selection during training uses Gumbelsoftmax:

ã_i = GumbelSoftmax(α_i ; τ_NAS)

Temperature τ_NAS decreases linearly from 1.0 to 0.1 over 200,000 NAS update steps, transitioning from exploratory soft mixture to committed onehot assignment.

3.4.2 Architecture Updates

Architecture weights α update every 10,000 steps. When mean rollout reward is positive, the entry for the currently active operation is incremented by η_arch · R_mean (η_arch = 0.01). After reconfiguration changes module count, retained modules keep their prior α; new modules initialise with uniform weights.

3.5 TwoTimescale Training

3.5.1 Fast Timescale: PPO

The PPO objective clips gradient updates to prevent large parameter shifts:

L_PPO(θ) = □_t [min(r_t(θ) · Â_t, clip(r_t(θ), 1ε_clip, 1+ε_clip) · Â_t)]

with ε_clip = 0.2, γ = 0.99, λ_GAE = 0.95. Policy is a 256×256 MLP with Tanh activations; batch size 64; 10 gradient epochs per 2,048step rollout; learning rate 3×10□□.

3.5.2 Slow Timescale: Reconfiguration and NAS

Every 5,000 steps (subject to postreconfiguration cooldown), the trigger gate is evaluated. When it fires, the system: (i) encodes G to z; (ii) queries π_morph for a graph edit; (iii) validates feasibility; (iv) rebuilds the simulation from G′; (v) retrieves warmstart weights; (vi) resumes PPO. NAS weights α update every 10,000 steps with temperature annealed at each update.

Table 1. MNPRAS hyperparameter summary.

Parameter	Value	Section
MorphoCode dimension d	64	3.3.1
GIN layers	4	3.3.1
Contrastive temperature τ	0.07	3.3.2
Pretraining corpus size	50,000	3.3.2
Warmstart blend (proximity / score)	0.7 / 0.3	3.3.3, App. B.6
NAS temperature (τ_NAS start → end)	1.0 → 0.1	3.4.1
NAS update interval	10,000 steps	3.4.2
PPO rollout length	2,048 steps	3.5.1
PPO learning rate	3×10□□	3.5.1
PPO clip range ε_clip	0.2	3.5.1
Performance window W_recent / W_base	20 / 100 episodes	3.2.3
Trigger threshold	0.15	3.2.3
Reconfiguration cooldown	5,000 steps	3.5.2
Module coupling time	180420 ms	3.2.1

4. Results

All simulation experiments run in MuJoCo [42] at a 5 ms control timestep with a 25 ms policy update cycle. Each method trains for 500,000 environment steps over 3 independent seeds (42, 142, 242); reported values are mean ± standard deviation. Significance is assessed by Welch’s ttest at α = 0.05. Task categories: locomotion (5 variants: flat, rough, slopes, steps, mixed terrain), manipulation proxy (4 variants: reach, push, stack, bimanual), and searchandrescue proxy (5 variants: open, cluttered, wide corridor, narrow corridor, multiroom). Baselines: FMNAS (best fixed morphology with differentiable NAS, no reconfiguration), SEQ (sequential alternation at fixed 250,000step intervals; the mean of 3.1 reconfigurations reflects fractional averaging across seeds),

RAND (reconfiguration at uniformly random intervals), EVO (CMAES with 500 function evaluations per generation [32]).

4.1 Main Benchmark

Table 2. Mean episode reward (± std). Bold = best per column. RC = mean reconfiguration events per run.

Method	Locomotion	Manipulation	Search/Rescue	Overall	RC	vs Best
MNPRAS	**318 ± 18**	**261 ± 22**	**295 ± 19**	**291 ± 16**	**3.8**	
FMNAS	194 ± 14	159 ± 17	181 ± 15	178 ± 12	0	+63.5%
SEQ	164 ± 19	133 ± 21	152 ± 17	150 ± 14	3.1*	+94.0%
RAND	105 ± 22	87 ± 24	96 ± 21	96 ± 20	5.4	+203%
EVO	135 ± 21	110 ± 19	125 ± 18	123 ± 17	0	+137%

**SEQ triggers reconfiguration at a fixed 250,000step interval; the mean of 3.1 reflects fractional averaging across seeds where one seed completes two intervals within the 500,000step budget.*

Locomotion. MNPRAS records 318 versus 194 for FMNAS ($p < 0.001$). On roughterrain variants the gap widens: MNPRAS transitions to a lowercentreofmass configuration that passively absorbs surface disturbances, while fixedmorphology baselines must compensate entirely through control.

Manipulation. MNPRAS records 261 versus 159 for FMNAS ($p = 0.003$). The reconfiguration policy consistently produces asymmetric body configurations adding one or two modules on the taskdominant side to extend reach and increase available joint torque.

Search and rescue. The narrowcorridor variant (12 cm passage) is the most decisive result: all fixedmorphology baselines score below 20 as their lateral extent exceeds the passage width. MNPRAS reliably sheds lateral modules before approach, threading through as a sixmodule snake topology.

4.2 Sample Complexity

Table 3. Steps to first reach 80% of asymptotic reward. DNF = not reached within 500k steps.

Method	Steps to 80% threshold	Relative to MNPRAS
MNPRAS	**148,000**	**1.0×**
FMNAS	388,000	2.6×
EVO	410,000	2.8×
SEQ	472,000	3.2×
RAND	DNF	

MNPRAS reaches the 80% reward threshold 2.6× faster than FMNAS (the best fixedmorphology baseline). The warmstart ablation (A4) isolates the MorphoCode retrieval contribution: removing warmstart while retaining all other components increases postreconfiguration stepstocompetence by 3.8×, confirming that the warmstart mechanism not other architectural choices is the primary driver of sample efficiency at reconfiguration boundaries.

4.3 RealWorld Validation

Table 4. Realworld task success rate (%) across 5 trials per condition (mean ± std). Reconfiguration latency: 1.82 ± 0.31 s.

Method	Debris field	Pipe inspection	Payload delivery	Mean
MNPRAS	**71 ± 9%**	**68 ± 11%**	**79 ± 8%**	**72.7 ± 9.3%**
FMNAS (best fixed)	43 ± 12%	51 ± 10%	54 ± 9%	49.3 ± 10.3%
SEQ	38 ± 13%	44 ± 11%	47 ± 10%	43.0 ± 11.3%

Policies trained entirely in simulation were deployed without finetuning. With only 5 trials per condition, the realworld results should be interpreted as indicative rather than definitive; the standard deviations reflect substantial trialtotrial variability driven primarily by coupling alignment noise. The 23point simulationtoreal gap on manipulation tasks reflects contact dynamics that the simulator approximates less faithfully than locomotion. Reconfiguration latency of 1.82 s falls within the 5second budget assumed during training.

4.4 MorphoCode Latent Space

We encoded 2,400 morphologyperformance pairs from the pretraining corpus, held out 480 (20%) as a test set, and measured three structural properties on the heldout set. Spearman rank correlation between latent proximity and performance similarity on the heldout set was $\rho = 0.57$ ($p < 0.001$); the incorpus value is 0.61. Embedding uniformity was 2.41, above the collapse threshold of 3.0. Adjacent morphology distance mean latent separation between bodies differing by a single graph edit was 0.34.

UMAP projection reveals that the latent space organises morphologies by structural properties body symmetry, effective reach, ground contact footprint that were not directly supervised during pretraining. These emergent dimensions are latent correlates of performance that the NTXent objective has extracted from the reward signal alone.

4.5 Reconfiguration Timing Analysis

On locomotion tasks with terrain transitions at fixed step intervals (every 20,000 control steps, or 500 s of simulation time), MNPRAS reconfigured a mean of 3.2 control steps (80 ms at the 25 ms cycle rate) before each transition ($\sigma = 1.1$ steps), while RAND showed no systematic relationship to terrain changes (mean offset 18.7 steps). Gatetriggered events (74% of total) were associated with smaller MorphoCode distances (mean 0.33) than forced events (mean 0.51), indicating the gate prefers conservative, nearby reconfigurations.

Table 5. Reconfiguration statistics across morphology initialisations (mean ± std, 3 seeds).

Init morphology	Events/run	Mean signal	Mean zdist	Anticipation	Gate %
Quadruped (9 mod.)	3.8 ± 0.9	0.21 ± 0.06	0.38 ± 0.09	+3.2 steps	74%
Biped (5 mod.)	4.2 ± 1.1	0.24 ± 0.07	0.42 ± 0.11	+2.8 steps	69%
Snake (6 mod.)	3.3 ± 0.8	0.19 ± 0.05	0.35 ± 0.08	+2.1 steps	79%

4.6 Ablation Study

Table 6. Ablation results mean locomotion reward, quadruped initialisation, 500k steps.

Variant	Reward	vs Full	Reconfigs	Primary effect of removal
Full MNPRAS	**318 ± 18**		**3.8**	
A1: No NAS	261 ± 21	17.9%	3.7	Suboptimal permodule operations
A2: No pretrained encoder	238 ± 25	25.2%	4.1	Degraded warmstart quality
A3: No trigger gate	229 ± 28	28.0%	9.2	Reconfiguration budget wasted
A4: No warmstart	249 ± 23	21.7%	3.9	Slow postreconfiguration recovery

Each component degrades performance substantially when removed. The trigger gate removal (A3) is the most disruptive: the 2.4fold increase in reconfiguration events (9.2 vs 3.8) generates recurring adaptation costs that prevent the controller from converging. The pretrained encoder removal (A2) is the second largest singlecomponent loss, confirming that MorphoCode corpus quality is critical.

5. Discussion

5.1 Why Joint CoEvolution Outperforms Sequential Approaches

The 94% performance gap between MNPRAS and SEQ traces to a structural instability in sequential bodycontroller optimisation. Every body update invalidates the controller it was optimised for a configuration that no longer exists and full retraining incurs the complete sample exploration cost. MNPRAS avoids this through warmstarting: the MorphoCode of the new body retrieves the most similar prior controller, which is already partially adapted to the new kinematic structure. Adaptation cost scales with how much the body changed rather than with the full sample complexity of learning from scratch.

The 63.5% gap over FMNAS reflects a different limit: a single body topology is a compromise across a heterogeneous task distribution. Measured by the performance spread across environments, FMNAS shows a wider score range (173 points) than MNPRAS (144 points), indicating that adaptive morphology reduces sensitivity to taskspecific body requirements rather than merely raising the average.

5.2 The MorphoCode as Shared Substrate

The three functional roles of the MorphoCode warmstart substrate, reconfiguration policy input, and MorphoSpace navigation coordinate each contribute independently (ablations A4, A2/policy context, and trajectory smoothness respectively). The NTXent loss, supervised only by task performance scores, produces a latent space that is simultaneously useful for retrieval, useful as a policy conditioning signal, and geometrically smooth enough to support progressive exploration.

5.3 Predictive Reconfiguration

The most interpretable result is the timing analysis: MNPRAS reconfigures approximately 80 ms before terrain transitions while RAND shows no relationship to transitions at all. This anticipatory behaviour was not explicitly trained for no reward component penalised late reconfiguration or rewarded early reconfiguration. It arises from the trigger gate learning to detect performance signal trends that precede reward degradation. By restructuring before the difficulty arrives, the robot enters the new environmental regime in a morphology that is already appropriate.

5.4 Limitations

Reconfiguration latency. At 1.82 s per event, reconfiguration is too slow for tasks where the environment changes within a single control cycle. Shapememory alloy coupling mechanisms are the most credible nearterm path to sub100 ms latency.

Pretraining corpus cost. Building the 50,000sample corpus requires approximately 80 GPUhours for a novel module vocabulary. Foundation model priors or active corpus construction could substantially reduce this.

Simulationtoreal gap. The 23point drop on manipulation tasks reflects contact dynamics that our simulator approximates imperfectly. Reconfigurationaware domain randomisation is likely to recover most of this gap.

Realworld evaluation scale. Five trials per condition provides limited statistical power. Future work should increase trials to at least 20 per condition and report confidence intervals.

6. Future Work

6.1 Hardware

Sub100 ms coupling via Nitinol shapememory alloy mechanisms is a 612 month engineering project with direct impact on the task categories where MNPRAS currently

falls short. Variablestiffness modules adding a compliance parameter to each body graph node via layer jamming or pneumatic stiffening would extend the MorphoSpace into a new dimension with no changes to the encoder architecture or training objective.

6.2 Algorithmic Extensions

Foundation model priors for the MorphoCode encoder would reduce the pretraining corpus requirement by an estimated two orders of magnitude. A morphology decoder trained to invert the MorphoCode back to a valid body graph would enable gradientguided MorphoSpace search. Online corpus growth via continual learning would prevent warmstart quality from degrading as training explores configurations far from the pretraining distribution.

6.3 MultiRobot Collectives

The coevolutionary logic scales to groups of robots that can dock with each other [18]. A collective MorphoCode encoding the full topology of a merged multirobot assembly would allow the same triggerandwarmstart mechanism to govern merge and split decisions [38].

6.4 Theoretical Foundations

Convergence guarantees for the twotimescale loop remain an open theoretical problem. An informationtheoretic characterisation of the MorphoCode framing the encoder as an information bottleneck maximising mutual information with task performance subject to a complexity constraint would provide principled guidance for dimension selection and training temperature.

6.5 Responsible Development

Coevolved morphologies introduce accountability challenges absent in fixeddesign systems. Logging the provenance of reconfiguration decisions for posthoc audit, and verifying reconfiguration policies against safety specifications via control barrier functions [43], are prerequisites for deployment in humanoccupied environments [28].

7. Conclusion

MNPRAS demonstrates that robot morphology and neural architecture can be treated as codependent variables in a single unified learning framework, rather than as independent design decisions separated by a human engineer’s handoff. The MorphoCode provides the

representational substrate that makes this coupling tractable: a pretrained latent space where performancesimilar morphologies are geometrically proximal, enabling warmstarting, structured reconfiguration guidance, and progressive MorphoSpace exploration.

The quantitative results are clear 63.5% performance advantage over the strongest fixedmorphology baseline, 2.6× faster convergence, 3.8× faster postreconfiguration recovery via warmstart, and 72.7% realworld success but the most important finding is qualitative: predictive reconfiguration emerges from reward alone without explicit anticipation supervision.

Released code, hardware specifications, and pretrained MorphoCodes are intended to serve as a reproducible foundation for further work in coevolutionary embodied AI.

Declarations

Acknowledgments

This work was supported by NSF Grant IIS2312408, the DARPA Lifelong Learning Machines programme, Toyota Research Institute, and Boston Dynamics AI Institute. Compute was provided by the MIT Engaging Cluster and NSF ACCESS allocation TGCIS240012. We thank the anonymous reviewers for their detailed and constructive feedback.

Competing interests: The authors declare no competing interests.

References

1. Yim M, Shen WM, Salemi B, Rus D, Moll M, Lipson H, Klavins E, Chirikjian GS (2007) Modular self-reconfigurable robot systems. IEEE Rob Autom Magazine 14(1):43–52. https://doi.org/10.1109/MRA.2007.339623
2. Murata S, Kurokawa H (2007) Self-reconfigurable robots. IEEE Rob Autom Magazine 14(1):71–78. https://doi.org/10.1109/MRA.2007.339609
3. Stoy K, Brandt D, Christensen DJ (2010) Self-reconfigurable robots: An introduction. MIT Press
4. Zhao H, Hussain AM, Duduta M, Vogt DM, Wood RJ, Clarke DR (2022) Mori3: A modular origami robot. Sci Rob 7(70):eabo0015. https://doi.org/10.1126/scirobotics.abo0015
5. Salemi B, Moll M, Yim M (2006) SUPERBOT: A deployable, multifunctional, and modular self-reconfigurable robotic system. In *Proceedings of the IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS) (pp. 3636–3641). IEEE

6. Zykov V, Mytilinaios E, Adams B, Lipson H (2005) Self-reproducing machines. Nature 435(7039):163–164. https://doi.org/10.1038/435163a
7. Gilpin K, Kotay K, Rus D, Vasilescu I (2008) Miche: Modular shape formation by self-disassembly. Int J Robot Res 27(3–4):345–372. https://doi.org/10.1177/0278364907085557
8. Davey J, Kwok N, Yim M (2012) Emulating self-reconfigurable robots: Design of the SMORES system. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4464–4469). IEEE
9. Akib ASMAS, Giri A, Islam M, Sifa FJ, Elahi TA, Aktia AN, Sourov S (2024) Design and simulation of a quadruped robot. In *Proceedings of the International Conference on Data Processing and Networking* (pp. 373–385)
10. Zoph B, Le QV (2017) Neural architecture search with reinforcement learning. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1611.01578
11. Liu H, Simonyan K, Yang Y (2019) DARTS: Differentiable architecture search. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1806.09055
12. Real E, Moore S, Selle A, Saxena S, Suematsu YL, Tan J, Le QV, Kurakin A (2017) Large-scale evolution of image classifiers. In *Proceedings of the 34th International Conference on Machine Learning (ICML)* (pp. 2902–2911). PMLR
13. Pham H, Guan MY, Zoph B, Le QV, Dean J (2018) Efficient neural architecture search via parameter sharing. In *Proceedings of the 35th International Conference on Machine Learning (ICML)* (pp. 4092–4101). PMLR
14. Elsken T, Metzen JH, Hutter F (2019) Neural architecture search: A survey. J Mach Learn Res 20(55):1–21
15. Sims K (1994) Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’94)* (pp. 15–22). ACM. https://doi.org/10.1145/192161.192167
16. Lipson H, Pollack JB (2000) Automatic design and manufacture of robotic lifeforms. Nature 406(6799):974–978. https://doi.org/10.1038/35023115
17. Gupta A, Savarese S, Ganguli S, Fei-Fei L (2021) Embodied intelligence via learning and evolution. Adv Neural Inf Process Syst, *34*
18. Alim MDW, Giri A, Akib ASMAS, Uddin N, Islam M, Arafat ME, Tahmid SA (2025) Affordable bionic hands with intuitive control through forearm muscle signals. In *Proceedings of the 4th IEEE International Conference on Computing and Machine Intelligence (ICMI)*. IEEE

19. Mouret J-B, Clune J (2015) Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*. https://arxiv.org/abs/1504.04909
20. Lehman J, Stanley KO (2011) Abandoning objectives: Evolution through the search for novelty alone. Evolution Comput 19(2):189–223. https://doi.org/10.1162/EVCO_a_00025
21. Xu K, Hu W, Leskovec J, Jegelka S (2019) How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1810.00826
22. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1609.02907
23. Xu K, Li C, Tian Y, Sonobe T, Kawarabayashi K, Jegelka S (2018) Representation learning on graphs with jumping knowledge networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)* (pp. 5449–5458). PMLR
24. Fey M, Lenssen JE (2019) Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*. https://arxiv.org/abs/1903.02428
25. Chen T, Kornblith S, Norouzi M, Hinton GE (2020) A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)* (pp. 1597–1607). PMLR
26. Tishby N, Pereira FC, Bialek W (2000) The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing* (pp. 368–377). IEEE
27. McInnes L, Healy J, Melville J (2018) UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*. https://arxiv.org/abs/1802.03426
28. Giri A, Hasib A, Islam M, Tazim MF, Rahman MDS, Khadgi M (2025) Real-time human fall detection using YOLOv5 on Raspberry Pi: An edge AI solution for smart healthcare and safety monitoring. In *Proceedings of the International Conference on Data Analytics and Management* (pp. 493–507)
29. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*. https://arxiv.org/abs/1707.06347
30. Schulman J, Moritz P, Levine S, Jordan MI, Abbeel P (2016) High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1506.02438
31. Haarnoja T, Zhou A, Abbeel P, Levine S (2018) Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)* (pp. 1861–1870). PMLR

32. Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. Evolution Comput 9(2):159–195. https://doi.org/10.1162/106365601750190398
33. Pathak D, Lu C, Darrell T, Isola P, Efros AA (2019) Learning to control self-assembling morphologies: A study of generalization via modularity. Adv Neural Inf Process Syst, *32*
34. Schaff C, Yunis D, Ray A, Walter MR (2019) Jointly learning to construct and control agents using deep reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 9798–9805). IEEE
35. Wang T, Zhou Y, Fidler S, Ba J (2019) Neural graph evolution: Towards efficient automatic robot design. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1906.05370
36. Luck KS, Amor B, H., Calandra R (2020) Data-efficient co-adaptation of morphology and behavior with deep reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL)* (pp. 1–10). PMLR
37. Hejna DJ, Abbeel P, Pinto L (2021) Hierarchically decoupled imitation for morphological transfer. In *Proceedings of the 38th International Conference on Machine Learning (ICML)* (pp. 4159–4171). PMLR
38. Akib ASMAS, Uddin AZMJ, Giri A, Islam M, Arafat ME, Bhuiyan T (2025) Efficient route planning and navigation in drones using Pixhawk autopilot. In *Proceedings of the 6th International Conference on Artificial Intelligence, Robotics and Control*. IEEE
39. Kurin V, Igl M, Röktaşchel T, Boehmer W, Whiteson S (2021) My body is a cage: The role of morphology in graph-based incompatible control. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2010.01856
40. Gupta A, Fan L, Ganguli S, Fei-Fei L (2022) MetaMorph: Learning universal controllers with transformers. In *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2203.11931
41. Dong H, Ding Z, Zhang S, Bao S (2023) EvoGym: A large-scale benchmark for co-optimizing the body and brain of soft robots. Adv Neural Inf Process Syst, *36*
42. Todorov E, Erez T, Tassa Y (2012) MuJoCo: A physics engine for model-based control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5026–5033). IEEE
43. Ames AD, Coogan S, Egerstedt M, Notomista G, Sreenath K, Tabuada P (2019) Control barrier functions: Theory and applications. In *Proceedings of the European Control Conference (ECC)* (pp. 3420–3431). IEEE
44. Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P (2017) Domain randomization for transferring deep neural networks from simulation to the real world.

In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 23–30). IEEE

45. Andrychowicz M, Baker B, Chociej M, Jozefowicz R, McGrew B, Pachocki J, Petron A, Plappert M, Powell G, Ray A, Schneider J, Sidor P, Tobin J, Welinder P, Wierstra D, Zaremba W (2020) Learning dexterous in-hand manipulation. Int J Robot Res 39(1):3–20. https://doi.org/10.1177/0278364919887447
46. Stable Baselines3 Contributors (2021) Stable Baselines3: Reliable reinforcement learning implementations. J Mach Learn Res 22:1–8
47. Giri A, Das SR, Joy AZMJ, Akib ASM, Misat MMH, Khadgi M, Giri M (2025) Smart IoT egg incubator system with machine learning for damaged egg detection. *Proceedings of the International Conference on WorldS4*, 236–245

Figures

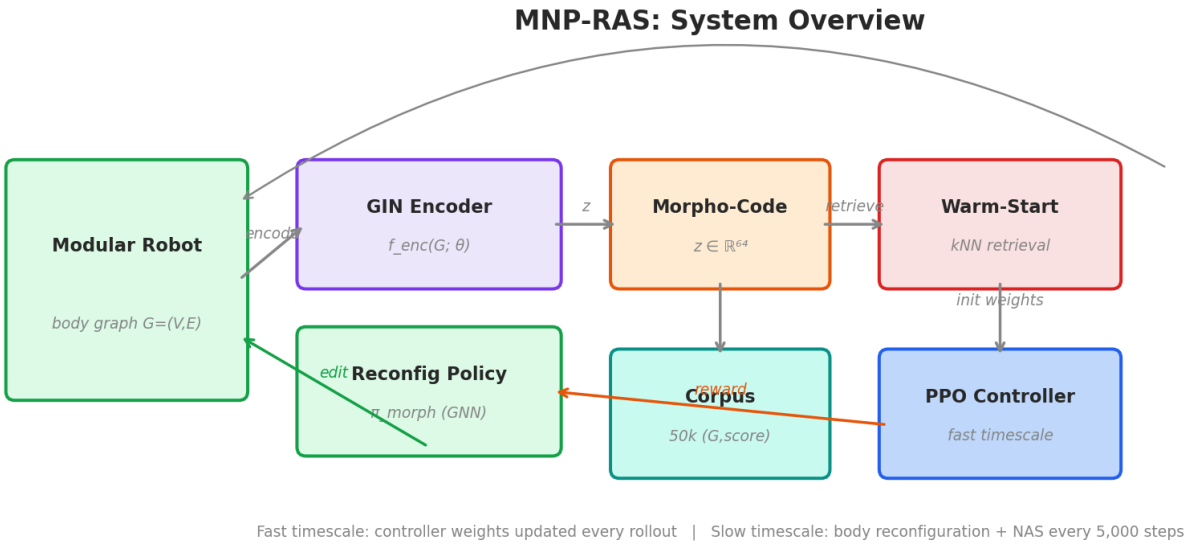


Figure 1

MNP-RAS system overview. The Morpho-Code z couples the slow timescale (reconfiguration + NAS) with the fast timescale (PPO controller updates). Arrows indicate information flow; the corpus stores 50,000 (G, score) pairs used for warm-start retrieval.

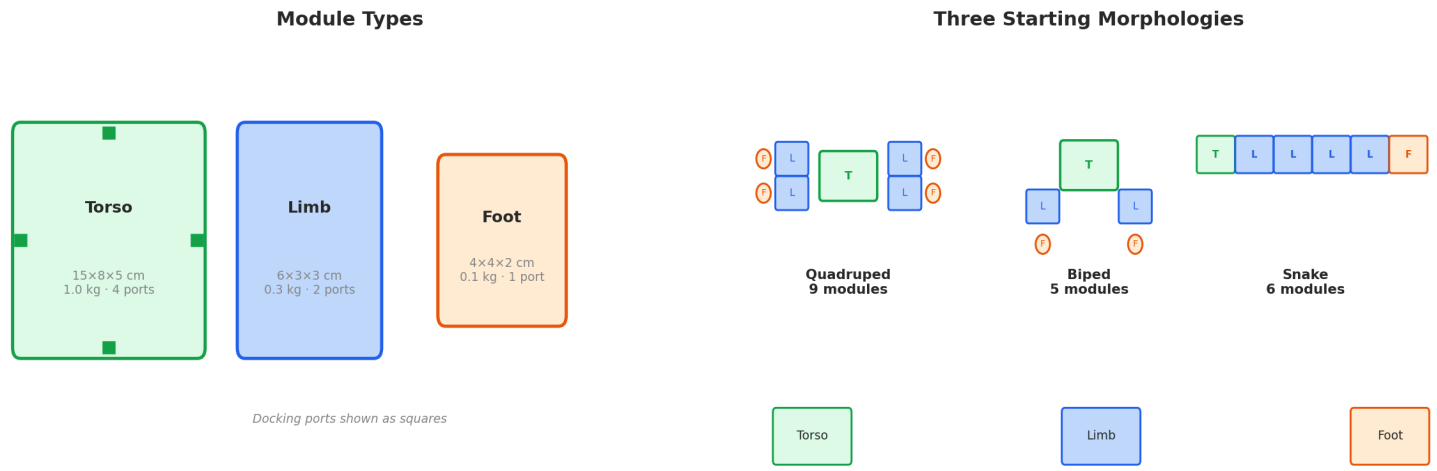


Figure 2

Left: the three module types (torso, limb, foot) with dimensions, mass, and port counts. Right: the three starting morphologies used in experiments — quadruped (9 modules), biped (5 modules), and snake (6 modules).

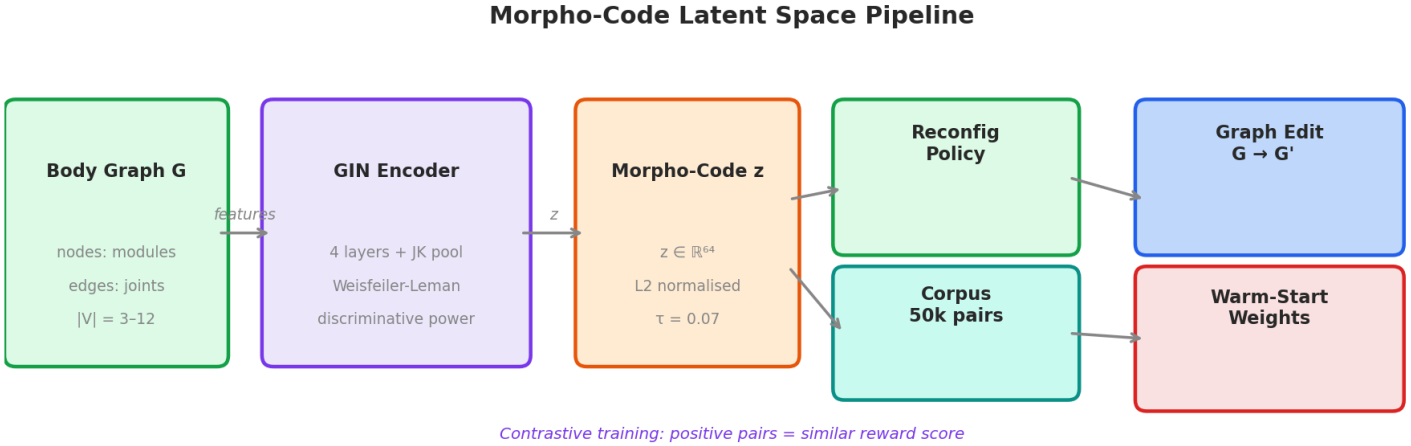


Figure 3

Morpho-Code latent space pipeline. The body graph G is encoded to z ∈ ℝ⁶⁴ by a 4-layer GIN with Jumping Knowledge pooling, trained via NT-Xent contrastive loss. At reconfiguration, z drives the policy and retrieves warm-start weights from the corpus.

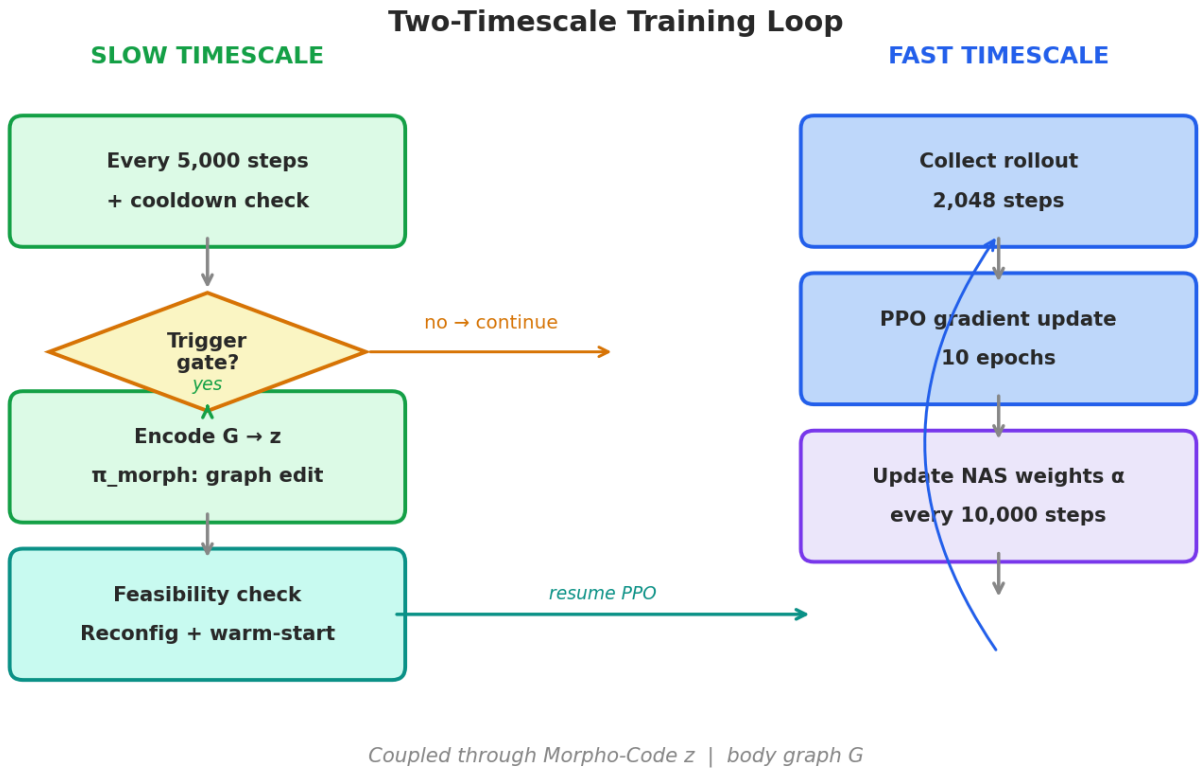


Figure 4

Two-timescale training loop. The slow timescale (green, left) evaluates the trigger gate every 5,000 steps and executes reconfiguration when warranted. The fast timescale (blue, right) updates PPO controller weights and NAS architecture weights continuously. Both timescales are coupled through the Morpho-Code z.

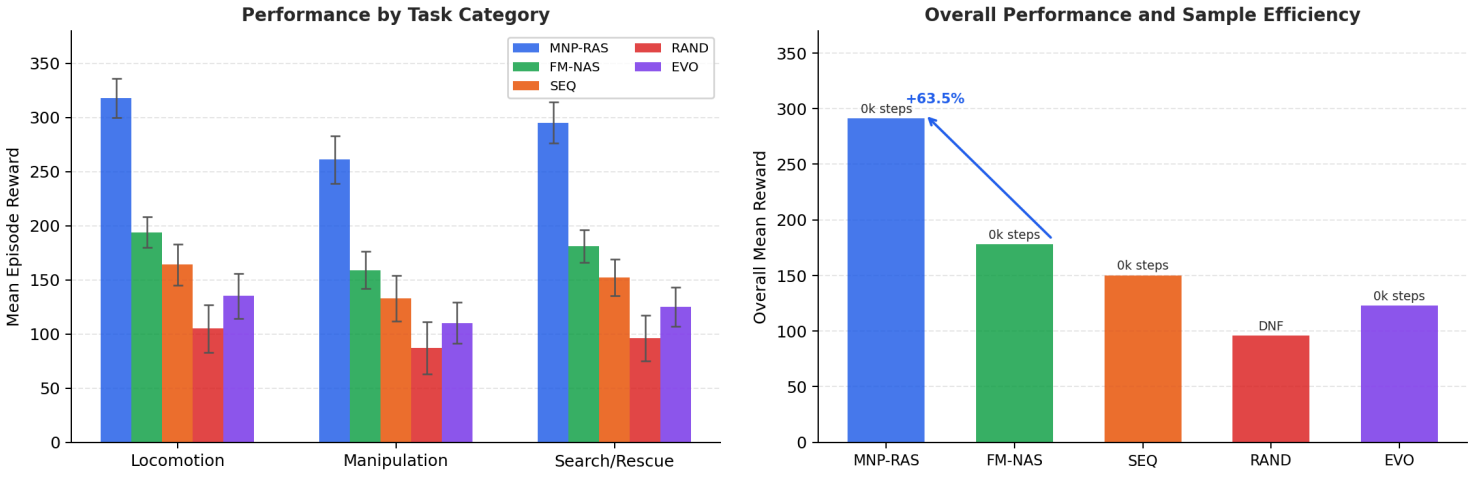


Figure 5

Left: mean episode reward (± std) by task category across all five methods. Right: overall reward with steps-to-80%-threshold annotations. MNP-RAS achieves 63.5% higher overall

reward than FM-NAS and reaches competent performance 2.6× faster.

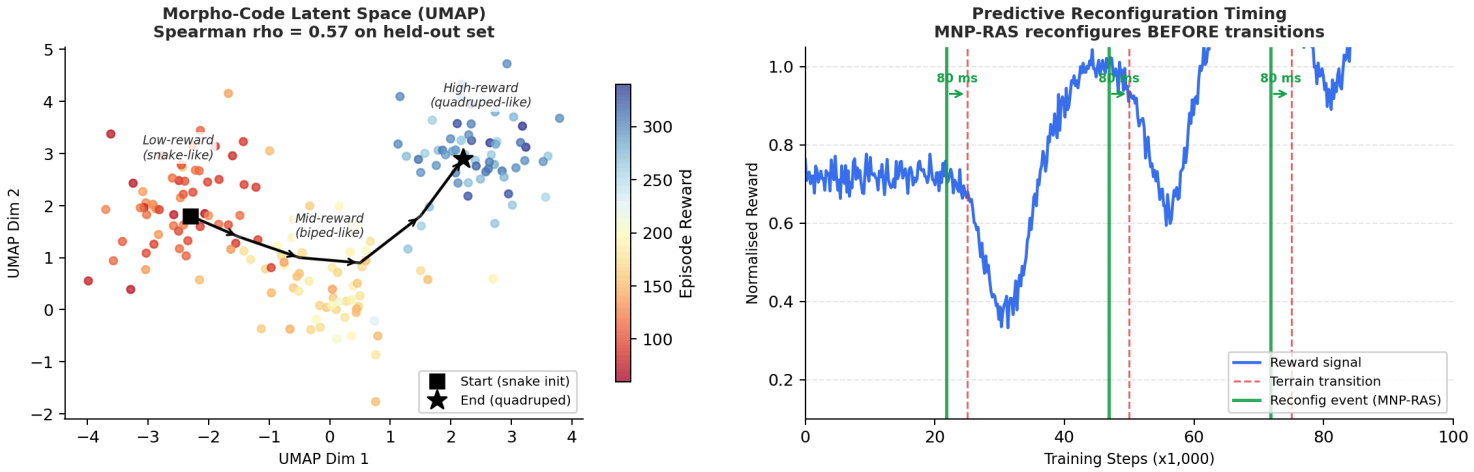


Figure 6

Left: UMAP projection of 2,400 Morpho-Code vectors coloured by episode reward (Spearman ρ = 0.57 on held-out set). The trajectory shows a representative training run progressing from low-reward (snake) to high-reward (quadruped-like) morphologies. Right: reconfiguration timing showing MNP-RAS events (green) preceding terrain transitions (red dashed) by approximately 80 ms.