TE-RoboNet: Transfer Enhanced RoboNet for Sample-Efficient Generation of Robot Co-Designs

Kishan Reddy Nagiredla
Applied Artificial Intelligence Initiative
Deakin University
Australia
knagiredla@deakin.edu.au

Arun Kumar Anjanapura Venkatesh
Applied Artificial Intelligence Initiative
Deakin University
Australia

Kevin Sebastian Luck
Computational Intelligence Group
Vrije Universiteit Amsterdam
Netherlands

Thommen George Karimpanal
Schoool of Information Technology
Deakin University
Australia

Santu Rana
Applied Artificial Intelligence Initiative
Deakin University
Australia

Abstract

Robot co-design requires joint optimization of morphology and its control mechanism and is thus associated with a vast, high-dimensional design space. Traditional co-design methods are sample-inefficient, and are thus used for incremental refinement of known designs rather than the discovery of novel, high-performing embodiments by effectively traversing this complex space. Our method extends RoboNet, a novel Generative Flow Network based robot co-design method that excels in generating superior designs in a sample-efficient manner but does that with independent training of control mechanism for each individual robot, resulting in good designs paired with weak controllers. In this paper, we propose a novel policy transfer mechanism to continuously learn a modularized policy, comprising a core network shared across all robot morphologies, and morphology-specific adapters. By effectively disentangling morphology-specific and transferable control components, our framework addresses the critical challenge of knowledge transfer between robot morphologies and their topologies with varying DoFs. Experiments in four distinct co-design environments show that our method, TE-RoboNet, achieves up to 40% improvement in performance compared to the closest co-design baselines under equivalent memory and computational budgets.

1 Introduction

The design of robotic systems is inherently intertwined with policies that control them. A robot's morphology profoundly influences its capabilities, and optimal control strategies often depend on the specific morphological traits of the robot. This mutual dependency gives rise to the problem of robot co-design in which the simultaneous optimization of both a robot's body and its control policy is required. Additionally, as robots become increasingly diverse and are deployed in unstructured environments, co-design becomes essential for achieving high-performance, task-specific solutions. However, robot co-design poses a unique set of challenges. The joint design space encompassing dis-

18th European Workshop on Reinforcement Learning (EWRL 2025).

crete morphological configurations with continuous parametric representations and high-dimensional policy manifolds is vast [Crespo Márquez, 2022], non-differentiable, and often lacks smooth gradients [Levine et al., 2018, Ha, 2019]. This makes conventional optimization techniques ineffective or sample-inefficient. In practice, a new robot design often requires re-training a control policy from scratch, leading to prohibitively high computational costs. Recent advances have attempted to address this problem by leveraging evolutionary algorithms [Wang et al., 2019, Doncieux et al., 2015, Lipson and Pollack, 2000, Gupta et al., 2021], harnessing early-stopping [Nagiredla et al., 2024b], and reinforcement learning [Luck et al., 2020, Yuan et al., 2021, Lu et al., 2025, Fan et al., 2024]. Yet, many of these methods suffer from poor sample efficiency often converging to local optima under reasonable sampling budgets.

A recent promising line of work, RoboNet [Nagiredla et al., 2024a], uses Generative Flow Networks (GFlowNets) [Bengio et al., 2021] to sample robot designs in a reward-aware manner, and formulates robot morphology as a graph to generate designs based on downstream performance. While this method shows impressive capabilities in exploring the design spaces and generates a diverse set of well-performing designs, it still treats each robot-control pair independently, thus producing excellent designs but sometimes with poor controllers, weakening the case of co-designing.

We argue that effective co-design demands not just better exploration of the design space, but also mechanisms for knowledge transfer. Specifically, the ability to transfer control policy components across morphologically distinct robots could significantly accelerate learning and improve sample efficiency. However, this remains a challenging open problem, as it requires identifying and disentangling morphology-invariant and morphology-specific aspects of control.

To address the robot co-design problem, our method, inspired by neurobiological evidence suggesting a separation of conserved motor primitives from body-specific mappings [Bizzi, 2008, Montgomery, 2024], hypothesizes that robotic control policies can similarly benefit from an adapter network to capture input and output mappings while a core network captures behavioral primitives. Our Transfer-Enhanced RoboNet (TE-RoboNet) is a novel framework that builds on RoboNet's GFlowNet-based design generation pipeline and introduces a core-adapter policy architecture. Specifically, the control policy for each robot is decomposed into a core (blue blocks in Fig. 1), which captures transferable behavioral priors, and robot-specific adapters (red blocks in Fig. 1), which encode morphology-dependent nuances. The shared core (in green) represents aggregated knowledge across each individual core after reward-weighted averaging across a population of diverse robots morphologies. To ensure smooth learning, regularization is applied during each update. Through such novel and controlled transfer, our TE-RoboNet framework allows the shared core to propagate useful behaviors between morphologically diverse agents.

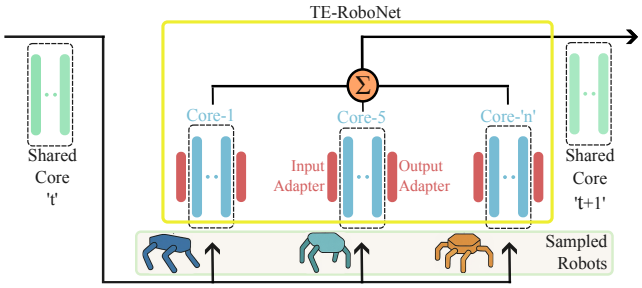

Figure 1: An illustration of the policy transfer across robots of various morphologies. The adapters (red) handle input-output observations, while the core (blue) captures locomotion primitives which are then aggregated into the shared core (green) for warm starting subsequent (blue) cores.

Contributions Our contributions are as follows: (**C1**) We propose TE-RoboNet, a GFlowNet-augmented co-design framework that uses a novel core-adapter policy architecture to generate more effective robot morphology-policy pairs. (**C2**) We demonstrate, across four diverse physics environments, that TE-RoboNet significantly improves sample efficiency—achieving up to 40% higher performance than strong baselines under the same computational budget.

2 Background

Sims [1994] is the first work that identified the need for co-evolving designs with control in virtual robots. Whilst initial focus was more on developing evolutionary approaches due to their closeness to how animal kingdom evolved, more recent approaches took a more pragmatic approach in treating

them as graphs that needs to be optimized with a policy/behavior learning framework. These work are presented in more detail below:

2.1 Evolutionary Approaches

Evolutionary Algorithms (EAs) apply principles like selection and mutation to explore the design space. Pioneering works demonstrated the co-evolution of morphology and control (e.g., Lipson and Pollack [2000]). Neural Graph Evolution (NGE) modeled robots as graphs and used Graph Neural Networks (GNNs) for control policies, enabling some policy transfer. Deep Evolutionary Reinforcement Learning (DERL) co-evolved morphologies and learned complex controllers using deep RL, studying environmental complexity's impact. EAs excel at exploring large, non-differentiable spaces but are sample-inefficient, as controller learning for each candidate is costly. They also generally lack explicit mechanisms for structured knowledge transfer across diverse morphologies.

2.2 Reinforcement Learning for Co-adaptation

Reinforcement Learning (RL) learns optimal control policies by maximizing rewards. In co-design, RL has been used to learn controllers for evolving morphologies or to jointly optimize morphology and control. Luck et al. [2020] used deep RL (Soft Actor-Critic) to co-adapt morphology and behavior, leveraging information from prior evaluations to improve data efficiency. Yuan et al. [2021] proposed Transform2Act, where an RL agent learns transform actions to modify its morphology and then control actions using the adapted body, employing Graph Neural Networks to handle variable joint numbers. RL can learn complex policies but is sample-intensive, especially when morphology changes, often requiring policy retraining. Transferring policies effectively across different morphologies remains a key challenge.

2.3 Generative Models and the Need for Knowledge Transfer

Recently, Generative Flow Networks (GFlowNets) [Bengio et al., 2021] have emerged as a potent method to sample from a distribution in the space of discrete structures. GFlowNets learn to sample discrete objects (like robot morphologies) with probability proportional to a reward function, constructing them step-by-step. RoboNet [Nagiredla et al., 2024a] applied GFlowNets to robot co-design, sampling promising graph-based morphologies based on anticipated task performance, and introduced innovations like rate-based prioritization and cost-aware sampling. However, RoboNet treats each robot-control pair independently, lacking mechanisms to transfer behavioral knowledge across morphologies. This knowledge silo issue motivates the need for explicit knowledge transfer to reduce redundant learning efforts. The morphological modularity hypothesis [Cappelle et al., 2016, Bongard, 2011] suggests optimal policies might share common behavioral foundations adaptable to specific morphologies.

2.4 Policy Transfer for Heterogeneous Robots

Transfer learning in robotics fundamentally seeks to enable knowledge reuse across diverse embodiments, tasks, and environments. A prominent baseline in this domain is MetaMorph [Gupta et al., 2022], which introduced a Transformer-based architecture for learning universal controllers transferable across varying robot morphologies. Subsequent methods, build on this foundation by improving parameter efficiency and specialization [Przystupa et al., 2025, Hao et al., 2024]. However, these approaches largely remain constrained to morphologically similar agents, leaving cross-morphological transfer across structurally heterogeneous robots an open problem.

TE-RoboNet addresses these gaps by integrating a core-adapter policy MLP architecture with GFlowNet-based design generation. The shared 'core' captures transferable primitives, updated via reward-weighted aggregation from evaluated designs, while robot-specific 'adapters' handle morphological nuances across heterogeneous topologies. This facilitates ongoing knowledge distillation and propagation during co-design, to improve sample efficiency through a MLP-based policy transfer.

3 Preliminaries

3.1 Reinforcement Learning

We model Reinforcement Learning (RL) problems as a Markov Decision Process (MDP), formally defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, R \rangle$. Here, $\mathcal{S}$ represents the state space, $\mathcal{A}$ denotes the action space, and $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function indicating the subsequent state upon executing action $a \in \mathcal{A}$ from state $s \in \mathcal{S}$. The function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ provides a scalar reward for taking action a in state s until a terminal state s_T is reached.

The core objective in RL is to discover an optimal policy that maximizes the expected cumulative reward, often referred to as returns, $\mathbb{E}_{\tau \sim \pi}[r(\tau)]$. A trajectory τ is a sequence of states and actions sampled according to policy π, and $r(\tau) = \sum_{(s_i, a_i) \in \tau} R(s, a)\gamma^i$ concisely represents the discounted sum of rewards along trajectory τ, where $\gamma \in [0, 1]$ is the discount factor.

3.2 Generative Flow Networks

Generative Flow Networks (GFlowNets) [Bengio et al., 2021] offer a unique approach to learning a stochastic, step-wise construction policy. The fundamental principle of GFlowNets is to sample terminal state $\bar{s}_T$ with a probability that is directly proportional to a given reward $\bar{R}(\bar{s}_T)$:

$$P(\bar{s}_T) \propto \bar{R}(\bar{s}_T) \tag{1}$$

This inherent design characteristic of GFlowNets, by explicitly distributing probability mass across numerous high-reward states, naturally facilitates the generation of diverse candidates. This capability has proven highly effective in domains demanding both high quality and variety, such as the discovery of novel drug candidates [Jain et al., 2022] and advanced materials [Cipcigan et al., 2024].

MDP Formulation for GFlowNets In the context of GFlowNets, the construction process is framed as an MDP. Beginning from an initial (often empty) state $\bar{s}_0$, a policy $\bar{\pi}_\theta(\bar{a}|\bar{s})$ makes sequential decisions: either to terminate the construction process or to attach a new component. This sequential decision-making generates a trajectory $\bar{\tau} = (\bar{s}_0, \bar{a}_0, \ldots, \bar{s}_T)$. Each state $\bar{s}$ within this construction graph is associated with a non-negative value known as its flow $F(\bar{s})$, which adheres to a crucial conservation rule:

$$F(\bar{s}) = \sum_{\bar{a} \in \bar{\mathcal{A}}} P(\bar{s}' \mid \bar{s}, \bar{a}) F(\bar{s}') \tag{2}$$

where $P(\bar{s}' \mid \bar{s}, \bar{a})$ represents the forward transition probability. When this flow conservation condition is satisfied, the resulting distribution of terminal states effectively matches the target distribution specified in Equation (1). This mechanism ensures two key properties: a) trajectories leading to high-reward terminal states are sampled with greater frequency, and b) the flow is systematically distributed across various construction paths, promoting the discovery of diverse solutions rather than converging on a single optimal one. Throughout training, the policy network iteratively adjusts the flow values in each state, ensuring consistency with both the defined reward function and the flow conservation property. Consequently, the model learns a probability distribution over trajectories that inherently encourages the generation of diverse, high-reward graphs.

Learning Objective To ensure that the sampling model learns a policy capable of generating samples according to a desired target distribution, GFlowNets utilize the Trajectory Balance (TB) objective [Malkin et al., 2022]. This objective is formally expressed as:

$$\mathcal{L}_{\text{TB}}(\bar{\tau}; \theta, \psi) = \left(\log \left(\frac{Z_\psi \prod_{\bar{s} \to \bar{s}' \in \bar{\tau}} P_{F_\theta}(\bar{s}' \mid \bar{s})}{\bar{R}(\bar{s}_T)} \right) \right)^2 \tag{3}$$

Here, θ and ψ denote the learnable parameters of the model, and Z_ψ is the partition function that maintains consistency across state transitions. The fundamental aim of the TB objective is to ensure that the marginal likelihood of a given trajectory becomes directly proportional to the reward $\bar{R}(\bar{s}_T)$ through an efficient credit assignment mechanism.

3.3 Robot Co-Design using GFlowNets

RoboNet [Nagiredla et al., 2024a] leverages GFlowNets and approaches the robot co-design problem as a bi-level optimization, aiming to maximize the expected cumulative reward $\mathbb{E}[R(m, \pi)]$ for morphology-policy pairs $(m, \pi) \in \mathcal{M} \times \Pi$. Here, morphological configurations is represented as $\mathcal{M}$ and control policies as Π and mathematically:

$$\max_{m \in \mathcal{M}} \bar{R}(m, \pi^*), \text{ s.t. } \pi^* = \arg\max_{\pi \in \Pi} R(m, \pi) \tag{4}$$

where m (a robot morphology in graph form $\mathbf{G} \in \mathcal{M}$) is optimized by an outer loop, and π^* is the optimal control policy determined by an inner loop. The outer loop models robot morphology design as a sequential graph construction process, represented as an MDP with a stochastic policy $\bar{\pi}_\theta$ maximizing:

$$\mathbb{E}\left[\sum_{t=0}^{T} \bar{\gamma}^t \bar{r}(\bar{s}_i, \bar{a}_i) \mid \bar{a} \sim \bar{\pi}_\theta(\bar{s}_i), \bar{s}_i = \bar{d}(\bar{a}_{i-1}, \bar{s}_{i-1})\right],$$

Here, $\bar{s}_i$ is the graph $\mathbf{G}_i$ at step i, $\bar{a}_i$ are graph-modifying actions, and $\bar{s}_i = \bar{d}(\bar{a}_{i-1}, \bar{s}_{i-1})$ are deterministic dynamics. RoboNet utilizes GFlowNets to learn $\bar{\pi}_\theta$, sampling terminal graphs $\mathbf{G}_T$ with probability proportional to the non-negative terminal reward $\bar{R}(\mathbf{G}_T)$, with $\bar{\gamma} = 1$. This $\bar{R}(\mathbf{G}_T)$ is defined by the inner loop's policy optimization: $\bar{R}(\mathbf{G}_T) = \max_\pi R(\mathbf{G}_T, \pi)$, where $R(\cdot, \cdot)$ is the task-specific reward with a discount factor $\gamma < 1$. While effective in diverse morphology sampling, the high computational cost of re-training policies for each morphology presents a practical challenge.

The training mechanism of RoboNet can be divided into two phases:

Phase 1: Sampling Morphologies RoboNet's samples a population of diverse morphologies $\{m_i\}_{i=1}^{B}$ according to the current reward-proportional distribution i.e. $m \sim P(\mathcal{M}) \propto R(m, \pi(m))$, ensuring exploration of high-potential design regions while maintaining morphological diversity.

Phase 2: Policy Learning For each morphology m_i, RoboNet initializes a task-specific policy π_i^*, optimized using reinforcement learning (e.g., PPO) to maximize the reward $R(m_i, \pi_i^*)$. To achieve further sample efficiency, instead of only using the task reward $R(m_i)$, it computes an effective measure of goodness ($\bar{R}(m_i)$) of a morphology from immature policies learned from only small number of interactions. The resulting reward-morphology pairs $(m_i, \bar{R}_i)$ are added to a replay buffer and used to update $\bar{\pi}_\theta$ via GFlowNet objectives.

This process repeats across generations, allowing RoboNet to progressively shift its sampling distribution toward high-performing morphologies. The replay buffer further stabilizes training by leveraging informative samples from prior generations. While effective in diverse morphology sampling, the high computational cost of re-training policies for each morphology presents a practical challenge.

4 Transfer-Enhanced RoboNet Framework

4.1 Problem Statement

Building upon the RoboNet framework (discussed in Sec. 3.3), which addresses the robot co-design problem as a bi-level optimization over morphological configurations $\mathcal{M}$ and control policies Π, our objective remains to maximize the expected cumulative reward $\mathbb{E}[R(m, \pi)]$ for morphology-policy pairs $(m, \pi) \in \mathcal{M} \times \Pi$. While RoboNet effectively samples diverse morphologies $m \sim P(\mathcal{M}) \propto R(m, \pi(m))$, the challenge lies in the computational expense of training an optimal policy π^* from scratch for each newly sampled morphology. Thus, to sample efficienctly through better initialization of subsequent robots training, we introduce a novel policy transfer mechanism that allows for cross-morphology knowledge sharing. Formally, in our method, TE-RoboNet, we seek to solve:

$$\max_m \bar{R}(m, \pi_{\tilde{\theta}}^\star), \text{ s.t. } \pi_{\tilde{\theta}}^\star = \arg \max_{\pi_\theta \in \Pi, \text{start} \leftarrow \pi_{\tilde{\theta}}} R(m, \pi_\theta) \tag{5}$$

where $\pi_{\tilde{\theta}}$ is the warm start policy parameterized by $\tilde{\theta}$ which facilitates knowledge transfer across different morphologies. We derive $\tilde{\theta}$ aggregating the optimal policy parameters learnt across morphologies m. The following sections detail our approach to constructing and leveraging $\tilde{\theta}$, which

is specifically designed to enhance inner policy optimization efficiency via shared knowledge, thus significantly improving the overall sample efficiency of the co-design process.

4.2 Core-Adapter Policy Architecture

A key innovation in TE-RoboNet is the decomposition of the MLP into adapters and a core. Such MLP design enables both accommodation of diverse morphology sizes and effective knowledge transfer within a single, memory-efficient architecture, unlike Transformer-based alternatives (discussed in Sec. 2.2 and 2.4). Specifically, the adapters constitute the MLP's input and output layers, directly interfacing with observations and actions, respectively, while the core comprises its intermediate hidden layers. The decomposition into adapters and a core is predicated on the idea that adapters capture morphology-specific attributes, while the core learns foundational behavioral primitives. Hence, for any generated robot morphology m, we parameterize the control policy as:

$$\pi_m(a_t|.) = \text{Adapter}(\text{Core}(m))$$

where, Core(m) represents shared behaviors capturing morphology-invariant control primitives and Adapter$(\cdot)$ maps Core(m) representations to morphology-specific action spaces $\mathcal{A}_m$. Mathematically:

$$\text{Adapter} = [\theta_m^{\text{IA}}, \theta_m^{\text{OA}}] \text{ and Core} = [\theta_m^C]$$

where $\theta^{\text{IA}}, \theta^{\text{OA}}$ contain the input-output adapter parameters to handle different morphology-dependent input observations and output action dimensions, and θ_m^C is the parameterized core for a given morphology m.

4.3 Core Aggregation for Policy Transfer

Reward-Weighted Core Aggregation. To facilitate knowledge transfer across populations of morphologically diverse robots, we propose a reward-weighted averaging mechanism to initialize cores of next population robots based on cores learnt from previous robot populations. Given a population B of robot-policy pairs $\{(m_i, \pi_i)\}_{i=1}^B$ with corresponding performance scores $\{R_i\}_{i=1}^B$. We can, thus, compute the aggregated core parameters $\tilde{\theta}^C$ as:

$$\tilde{\theta}^C = \frac{\sum_{i=1}^B w_i \cdot \theta_{m_i}^C}{\sum_{i=1}^B w_i}, \qquad \text{where the weights are defined as:} \quad w_i = \frac{(R_i)^\alpha}{\sum_{j=1}^B (R_j)^\alpha} \tag{6}$$

The exponent parameter α modulates the concentration of the polynomial weighting distribution, with higher values accentuating the influence of superior-performing morphologies in the core parameter aggregation process. This polynomial formulation exhibits more stable gradient characteristics particularly when reward magnitudes vary significantly across morphologies.

Stabilization of Core Aggregation. The reward-weighted aggregation of core parameters, as defined in Eq. 6, can induce significant parameter shifts when used to update the previously aggregated core. Such abrupt changes may destabilize the training process for subsequent robot generations. To mitigate these disruptions and ensure a smoother learning trajectory, we introduce a blending parameter, $\lambda \in (0, 1)$, which governs the rate at which the newly aggregated information updates the shared core parameters. Thus, the core aggregation including this scalar blending parameter λ can be formalized as,

$$\tilde{\theta}_{final}^c = \tilde{\theta}_{k-1}^C + \lambda \cdot \tilde{\theta}_k^C \tag{7}$$

where $\tilde{\theta}_{k-1}^C$ is the parameterized core used to warm start each core of the B robot-policy pairs $\{(m_i, \pi_i)\}_{i=1}^B$. After behavior training, B cores are produced which are then aggregated using Eq. 6 to form $\tilde{\theta}_k^C$. Thus, λ acts on this $\tilde{\theta}_k^C$ to smoothly update $\tilde{\theta}_{k-1}^C$, thereby ensuring stabilized training across generations.

We maintain the same training workflow as RoboNet (presented in Alg. 1), except that in each new generation policies for a new batch B of robots are warm-started with the shared core, $\tilde{\theta}_{final}^c$.

Algorithm 1 TE-RoboNet Training

Require: Initial GFlowNet policy $\bar{\pi}_\theta$, initial core parameters θ_0^{C}, population size B, reward exponent α, regularization coefficient λ, number of generations N_{gen}

Initialize GFlowNet policy $\bar{\pi}_\theta$
Initialize shared core parameters θ_0^{C}
for $t \in (0, N_{gen} - 1)$ **do**

▷ Phase 1: Sampling Morphologies

  Sample a population of diverse morphologies $\{m_i\}_{i=1}^{B}$ using $\bar{\pi}_\theta$ (GFlowNet sampler)
  for each morphology m_i in the population **do**

▷ Phase 2: Policy Learning

    Initialize core parameters θ_i^{C} from $\tilde{\theta}^C$ (aggregated core based on Eq. 7)
    Randomly initialize morphology-specific adapter parameters for π_i
    Train π_i for m_i using RL (e.g., PPO) over T timesteps in a two-stage process:
      Stage A: Freeze θ_i^{C} and learn adapter parameters by minimizing:
        $\mathcal{L}_i = -\mathbb{E}_{\tau\sim\pi_i}[R(\tau)]$
      Stage B: Freeze adapter parameters and learn θ_i^{C} by minimizing:
        $\mathcal{L}_i = -\mathbb{E}_{\tau\sim\pi_i}[R(\tau)]$
    Evaluate final performance $R_i = \mathbb{E}_{\tau\sim\pi_i}[R(\tau)]$ for morphology m_i
  end for

▷ Phase 3: Policy Transfer

  Compute reward-weighted core for the next generation:
    $w_i = \frac{(R_i)^\alpha}{\sum_{j=1}^{B}(R_j)^\alpha}$ for all $i \in \{1, \ldots, B\}$
    $\tilde{\theta}^C = \frac{\sum_{i=1}^{B} w_i \cdot \theta_{m_i}^C}{\sum_{i=1}^{B} w_i}$
  Update GFlowNet policy $\bar{\pi}_\theta$ based on the new rewards $\{R_i\}_{i=1}^{B}$ (as in standard RoboNet)
  Update shared core parameters using Eq. 7
end for

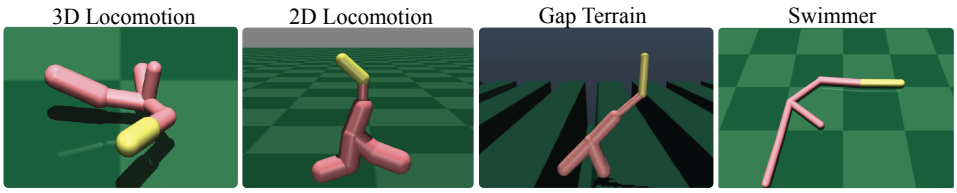

Figure 2: Different MuJoCo environments we use to evaluate our method TE-RoboNet with a randomly selected robot design in each.

4.4 Implementation Details

The core network employs a multi-layer perceptron (MLP) architecture with layer normalization and residual connections, designed to capture general behavioral patterns through a Proximal Policy Optimization technique [Schulman et al., 2017] independent of morphological specificities. Adapter layers exhibit morphology-awareness by explicitly encoding robot kinematic and dynamic properties through structured inductive biases.

The reward computation integrates task-specific performance rate of improvement [Nagiredla et al., 2024a] metric with morphological efficiency considerations and actuator effort. This multi-objective formulation encourages the discovery of morphologies that achieve high task performance while maintaining practical feasibility. This framework supports online learning scenarios where new morphologies can leverage previously discovered behavioral knowledge without requiring complete retraining, significantly reducing the computational overhead associated with traditional co-design approaches while exploring the search space more exhaustively.

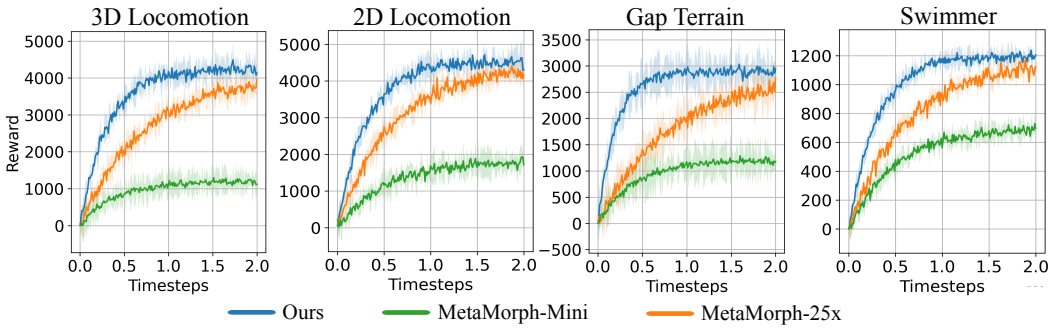

Figure 3: Training Curves showing best robot training performances for TE-RoboNet (blue), MetaMoprh-Mini (green) and MetaMorph-25x (orange) when trained for 2e6 timesteps over 5 independent seeds in 3D Locomotion, 2D Locomotion, Gaps Terrain and Swimmer environments (left to right).

5 Experiments and Results

In this section, we experimentally evaluate TE-RoboNet in 4 Gymnasium MuJoCo [Todorov et al., 2012] environments (presented in Fig. 2) where we retain all original physics settings. Specifically, we use *3D Locomotion* (based on Ant-v5 [Schulman et al., 2015]), *2D Locomotion* (based on Hopper-v5 [Erez et al., 2012]), *Gap Environment* (our modified terrain based on Hopper-v5) and *Swimmer* (based on Swimmer-v5 [Coulom, 2002]) environments. Each environment imposes a maximum limit on the number of links (or joints) per robot morphology as 10, 6, 8, and 6 respectively while also allowing for the sampling of morphologies with fewer links. We choose the MuJoCo simulator because it allows us to embed agents with different embodiments proposed by TE-RoboNet and baseline methods. While our method can handle robot morphologies with any number of joints, the maximum link limits are aimed at covering the search space exhaustively in reasonable runtime.

Experiment Setup Each TE-RoboNet robot morphology across these experiments is given a budget of $1e4$ timesteps per link and hence the maximum training budget allocated per robot is between $6e5$ to $10e5$ based on the environment (because of the enforced maximum link limits). We match this to the maximum training budget to the our baselines MetaMorph-Mini (same model parameters as TE-RoboNet i.e. $3.2e4$) and MetaMorph-25x (25x more model parameters compared to TE-RoboNet i.e., $3.2e5$), based on the same model architecture as Gupta et al. [2022]. Similar to Gupta et al. [2022], we use a dense morphology independent reward function across all environments instead of a tailored one for each morphology. In all experiments, our reward function promotes forward movement using small joint torques (the latter obtained via a small energy usage penalty).

5.1 Comparison of Best Performing Morphologies

We first evaluate how our core transfer mechanism based on the MLP architecture compares to a similar sized Transformer (TF) model in MetaMorph-Mini. We seek to answer if a smaller MLP model is more effective in learning behavior policy than a same sized TF-model. To present this result, we select top-5 co-designs in TE-RoboNet and compare to the output co-design from 5 different seeds of MetaMorph-Mini. As shown in Fig. 3, TE-RoboNet performs much better compared to a TF-based baseline model of same model size. We have observed that MetaMorph-Mini produced co-designs that are typically smaller and exhibited more conservative behavior to amass rewards by staying alive without moving rather than by moving forward. However, the larger model MetaMorph-25x was able to produce co-designs akin to TE-RoboNet and hence could per-

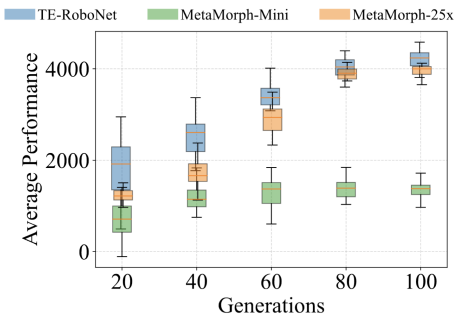

Figure 4: Average performance improvements across generations for TE-RoboNet and baseline methods in the 3D Locomotion environment.

form similarly. Fig. 5 illustrates the quality of the best designs produced across these methods in the 2D Locomotion environment through a comparison of their locomotion velocities. Additionally, Fig. 4 presents the average performance change across generations for TE-RoboNet and equivalent number of design updates for the baseline methods.

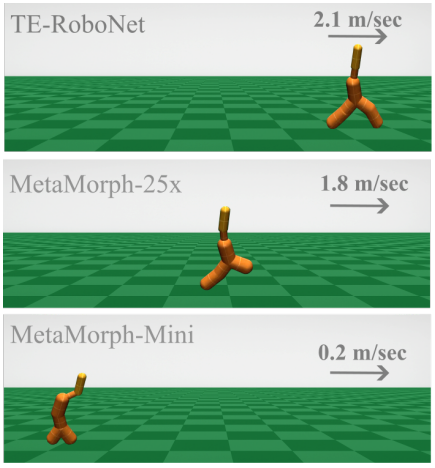

Figure 5: Visualization of co-designs produced by TE-RoboNet compared to baselines with forward velocity in m/sec in the 2D Locomotion environment.

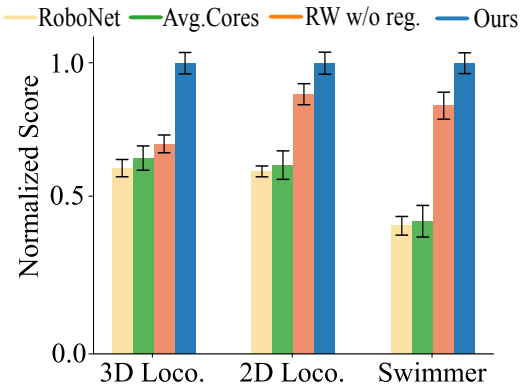

Figure 6: Normalized performance per million steps of top-performing designs by TE-RoboNet, its ablated versions, and the RoboNet [Nagiredla et al., 2024a] baseline.

5.2 Ablations

We performed ablations to study the impact of the individual components of the overall method, Fig. 6). We compare TE-RoboNet with and different variants of our method, Avg.Cores (which implements a simple averaging across cores to update the previous core) and RW w/o reg. (reward weighted knowledge transfer without regularization) along with RoboNet [Nagiredla et al., 2024a] in various environments. From Fig. 6 it is evident that RoboNet with no policy transfer performs poorly compared to other variants. While naive averaging of cores as in Avg. Cores, shows a slightly better performance, performs quite poorly when compared to reward weighted variants across all environments. Additionally, adding regularization to the aggregate core updates shows even higher performance compared to TE-RoboNet's variant without regularization of core parameter updates.

6 Discussion & Conclusion

We presented TE-RoboNet, a framework for generating diverse, top-performing robots with enhanced sample efficiency through cross-morphological policy transfer. Our key contribution is a reward-weighted core transfer mechanism with regularization. This framework, leveraging a novel adapter-core MLP architecture and split training, effectively handles cross-morphological policy transfer. Our experiments across different environments demonstrated up to a 40% increase in sample efficiency compared to baseline methods with similar model sizes.

Despite these promising results, several limitations and areas for future work exist. TE-RoboNet's current evaluation is confined to simulated environments. Transitioning to real-world robotic systems will require addressing challenges like the sim-to-real gap and unmodeled dynamics. Additionally, our framework relies on GFlowNets, whose theoretical convergence properties are still an active research area. While GFlowNets show empirical success, a complete understanding of their convergence guarantees remains elusive. However, recent theoretical advancements suggest GFlowNets exhibit more stable gradient updates with lower variance compared to traditional generative models [Malkin et al., 2022], contributing to observed training robustness.

Finally, the optimal division between the core and adapter components presents a practical challenge. The ideal balance depends on the diversity of sampled morphologies and their movement patterns.

Highly dissimilar movement patterns may necessitate a thinner core to prevent poor generalization, whereas significant behavioral commonality benefits from a thicker core to capture more transferable primitives. Determining this optimal balance requires empirical tuning and will be an important area for future investigation, potentially through adaptive core-adapter sizing mechanisms or through a hierarchical core representation.

References

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

Bizzi. Combining modules for movement. *Brain research reviews*, 57(1):125–133, 2008.

Josh Bongard. Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences*, 108(4):1234–1239, 2011.

Collin K Cappelle, Anton Bernatskiy, Kenneth Livingston, Nicholas Livingston, and Josh Bongard. Morphological modularity can enable the evolution of robot behavior to scale linearly with the number of environmental features. *Frontiers in Robotics and AI*, 3:59, 2016.

Flaviu Cipcigan, Jonathan Booth, Rodrigo Neumann Barros Ferreira, Carine Ribeiro dos Santos, and Mathias Steiner. Discovery of novel reticular materials for carbon dioxide capture using gflownets. *Digital Discovery*, 3(3):449–455, 2024.

Rémi Coulom. *Reinforcement learning using neural networks, with applications to motor control*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2002.

Adolfo Crespo Márquez. The curse of dimensionality. In *Digital Maintenance Management: Guiding Digital Transformation in Maintenance*, pages 67–86. Springer, 2022.

Stephane Doncieux, Nicolas Bredeche, Jean-Baptiste Mouret, and Agoston E Eiben. Evolutionary robotics: what, why, and where to. *Frontiers in Robotics and AI*, 2:4, 2015.

Tom Erez, Yuval Tassa, and Emanuel Todorov. Infinite-horizon model predictive control for periodic tasks with contacts. *Robotics: Science and systems VII*, 73, 2012.

Jiajun Fan, Hongyao Tang, Michael Przystupa, Mariano Phielipp, Santiago Miret, and Glen Berseth. Efficient design-and-control automation with reinforcement learning and adaptive exploration. In *AI for Accelerated Materials Design-NeurIPS*, 2024.

Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature communications*, 12(1):5721, 2021.

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

David Ha. Reinforcement learning for improving agent design. *Artificial life*, 25(4):352–365, 2019.

YiFan Hao, Yang Yang, Junru Song, Wei Peng, Weien Zhou, Tingsong Jiang, and Wen Yao. Heteromorpheus: Universal control based on morphological heterogeneity modeling. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure FP Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, et al. Biological sequence design with gflownets. In *International Conference on Machine Learning*, pages 9786–9801. PMLR, 2022.

Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5):421–436, 2018.

Hod Lipson and Jordan B Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978, 2000.

Haofei Lu, Zhe Wu, Junliang Xing, Jianshu Li, Ruoyu Li, Zhe Li, and Yuanchun Shi. Bodygen: Advancing towards efficient embodiment co-design. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kevin Sebastian Luck, Heni Ben Amor, and Roberto Calandra. Data-efficient co-adaptation of morphology and behaviour with deep reinforcement learning. In *Conference on Robot Learning*, pages 854–869. PMLR, 2020.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35: 5955–5967, 2022.

John C Montgomery. Roles for cerebellum and subsumption architecture in central pattern generation. *Journal of Comparative Physiology A*, 210(2):315–324, 2024.

Kishan Reddy Nagiredla, Arun Kumar AV, Thommen George Karimpanal, and Santu Rana. Robonet: A sample-efficient robot co-design generator. In *[CoRL 2024] Morphology-Aware Policy and Design Learning Workshop (MAPoDeL)*, 2024a.

Kishan Reddy Nagiredla, Buddhika Laknath Semage, Arun Kumar Anjanapura Venkatesh, Thommen George Karimpanal, and Santu Rana. Ecode: A sample-efficient method for co-design of robotic agents. In *Australasian Joint Conference on Artificial Intelligence*, pages 3–15. Springer, 2024b.

Michael Przystupa, Hongyao Tang, Glen Berseth, Mariano Phielipp, Santiago Miret, Martin Jägersand, and Matthew E Taylor. Efficient morphology-aware policy transfer to new embodiments. In *Reinforcement Learning Conference*, 2025.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Karl Sims. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, 1994.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

Tingwu Wang, Yuhao Zhou, Sanja Fidler, and Jimmy Ba. Neural graph evolution: Towards efficient automatic robot design. In *International Conference on Learning Representations*, 2019.

Ye Yuan, Yuda Song, Zhengyi Luo, Wen Sun, and Kris M Kitani. Transform2act: Learning a transform-and-control policy for efficient agent design. In *International Conference on Learning Representations*, 2021.