Co-Imitation: Learning Design and Behaviour by Imitation

Chang Rajani[1,2], **Karol Arndt**[2], **David Blanco-Mulero**[2], **Kevin Sebastian Luck**[2, 3], **Ville Kyrki**[2]
[1]University of Helsinki, [2]Aalto University, [3]Finnish Center for AI

Abstract

The co-adaptation of robots has been a long-standing research endeavour with the goal of adapting both *body and behaviour* of a system for a given task, inspired by the natural evolution of animals. Co-adaptation has the potential to eliminate costly manual hardware engineering as well as improve the performance of systems. The standard approach to co-adaptation is to use a reward function for optimizing behaviour and morphology. However, defining and constructing such reward functions is notoriously difficult and often a significant engineering effort. This paper introduces a new viewpoint on the co-adaptation problem, which we call *co-imitation*: finding a morphology and a policy that allow an imitator to closely match the behaviour of a demonstrator. To this end we propose a co-imitation methodology[1] for adapting behaviour and morphology by matching state distributions of the demonstrator. Specifically, we focus on the challenging scenario with mismatched state- and action-spaces between both agents. We find that co-imitation increases behaviour similarity across a variety of tasks and settings, and demonstrate co-imitation by transferring human walking, jogging and kicking skills onto a simulated humanoid.

1 Introduction

Animals undergo two primary adaptation processes: behavioural and morphological adaptation. An animal species adapts, over generations, its morphology to thrive in its environment. On the other hand, animals continuously adapt their behaviour during their lifetime due to environmental changes, predators or when learning a new behaviour is advantageous. While the processes operate on different time scales, they are closely interconnected and crucial elements leading to the development of well-performing and highly adapted organisms on earth.

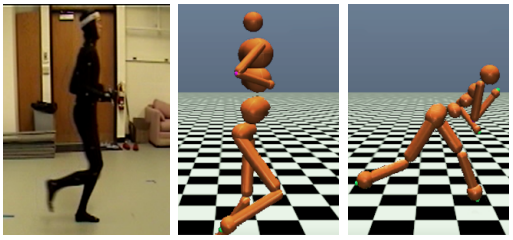


Figure 1: The proposed *co-imitation* algorithm (**centre**) is able to faithfully match the gait of human motion capture demonstrations (**left**) by optimizing both the morphology and behaviour of a simulated humanoid. This is opposed to a pure behavioural imitation learner (**right**) that fails to mimic the human motion accurately.

While research in robot learning has largely been focused on the aspects of behavioural learning processes, a growing number of works have sought to combine behaviour learning and morphology adaptation for robotics applications via *co-adaptation* [24, 22, 32, 15]. Earlier works focused primarily on the use of evolutionary optimization techniques [34, 30], but with the advent of deep learning, new opportunities arose for

[1]Additional material can be found at https://sites.google.com/view/co-imitation

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

the efficient combination of deep reinforcement learning and evolutionary adaptation [32, 24]. In contrast to fixed behaviour primitives or simple parameterized controllers used earlier, deep neural networks allow a much greater range of behaviours given a morphology.

Existing works in co-adaptation, however, focus on a setting where a reward function is assumed to be known, even though engineering a reward function is a notoriously difficult and error-prone task [35]. Reward functions tend to be task-specific, and even minor changes to the learner dynamics can cause the agent to perform undesired behaviour. For example, in the case of robotics, changing the mass of a robot may affect the value of an action penalty. This means that the reward needs to be re-engineered every time these properties change.

To overcome these challenges, we propose to reformulate co-adaptation by combining morphology adaptation and imitation learning into a common framework, which we name *co-imitation*. This approach eliminates the need for engineering reward functions by leveraging imitation learning for co-adaptation, hence, allowing the matching of *both the behaviour and the morphology* of a demonstrator.

Imitation learning uses demonstration data to learn a policy that behaves like the demonstrator. However, in the case where the two agents' morphologies are different, we face the following challenges: (1) state spaces of demonstrating and imitating agents may differ, even having mismatched dimensionalities; (2) actions of the demonstrator may be unobservable; (3) transition functions and dynamics are inherently disparate due to mismatching morphologies.

To address these issues we propose a co-imitation method which combines deep imitation learning through state distribution matching with morphology optimization. Summarized, the contributions of this paper are:

- Formalization of the problem of co-imitation: matching both the behaviour and morphology of a demonstrator.
- The introduction of *Co-Imitation Learning* (CoIL), a new co-imitation method, which adapts the behaviour and morphology of an agent by state distribution matching considering incompatible state spaces.
- A comparison of morphology optimization using learned non-stationary reward functions with our proposed approach of using a state distribution matching objective.
- A demonstration of CoIL by learning behaviour and morphology of a simulated humanoid given real-world demonstrations recorded from human subjects in tasks ranging from walking, jogging to kicking (see Fig. 1).

2 Related work

Deep Co-Adaptation of Behaviour and Design While co-adaptation as a field has seen interest since at least as early as the 90s [28, 34], in this section we look at previous work in the field especially in the context of deep reinforcement learning. Recent work by Gupta et al. [14] proposes a mixed evolutionary- and deep reinforcement learning-based approach (DERL) for co-optimizing agents' behaviour and morphology. Through mass parallelization, DERL maintains a population of 576 agents, which simultaneously optimize their behaviour using Proximal Policy Optimization (PPO) [33]. Based on their final task performance (i.e. episodic return), DERL optimizes the morphological structure of agents using an evolutionary tournament-style optimization process.

Schaff et al. [32] use deep reinforcement learning (RL) for the joint optimization of morphology and behaviour by learning a single policy with PPO. Again, the final episodic return of a design is used to optimize the parameters of a design distribution with gradient descent, from which the subsequent designs are sampled. Similarly, Ha [15] proposes to use REINFORCE to optimize policy parameters and design parameters of a population of agents in a joint manner. The co-adaptation method presented by Luck et al. [24] improves data-efficiency compared to return-based algorithms by utilizing the critic learned by Soft Actor Critic (SAC) [16] to query for the expected episodic return of unseen designs during design optimization. While the method we present is closest to the former approach, all discussed co-adaptation methods require access to a reward function, and are thus not capable of co-adapting the behaviour and design of an agent without requiring an engineer to formulate a reward function.

Imitation Learning with Morphology Mismatch Imitation learning approaches learn a policy for a given task from demonstrator data. In many cases this data can only be produced by an agent (or human) that has different dynamics from the imitator. We will give a brief overview on previous

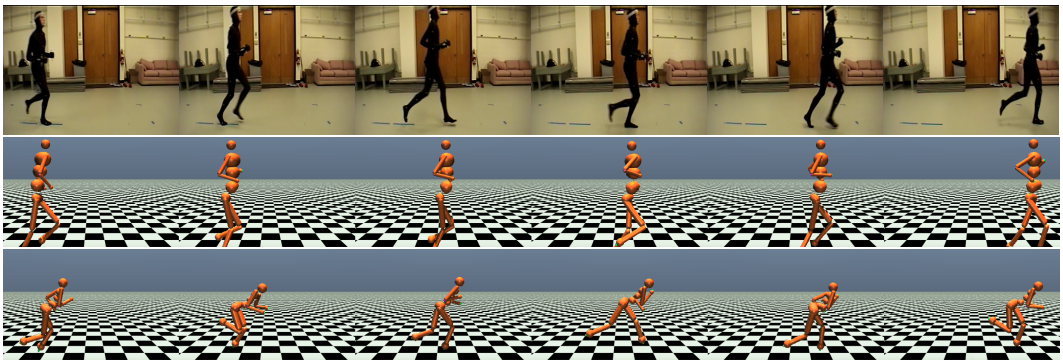


Figure 2: **Top**: A demonstrator of jogging from the CMU MoCap Dataset [4]. **Middle**: The co-imitation Humanoid produces a more natural looking jogging motion whereas the pure imitation learner (**bottom**) learns to run with a poor gait.

work where a policy is learned in presence of such transfer.

The work by Desai et al. [6] discusses the imitation transfer problem between different domains and presents an action transformation method for the state-only imitation setting. Hudson et al. [20] on the other hand learn an affine transform to compensate for differences in the skeletons of the demonstrator and the imitator. These methods are based on transforming either actions or states to a comparable representation.

To perform state-only imitation learning without learning a reward function, Dadashi et al. [5] introduced Primal Wasserstein Imitation Learning (PWIL), where a reward function is computed based directly on the primal Wasserstein formulation. While PWIL does not consider the case where the state space and the morphology are different between the demonstrator and the imitator, it was extended into the mismatched setting by Fickinger et al. [8]. They replace the Wasserstein distance with the Gromov-Wasserstein distance, which allows the state distribution distance to be computed in mismatched state spaces. In contrast, our method addresses the state space mismatch by transforming the state spaces to a common feature representation, allowing for more control over how the demonstrator's behaviour is imitated. Additionally, in contrast to these works, we optimize the morphology of the imitator to allow for more faithful behaviour replication.

Peng et al. [29] propose an imitation learning pipeline allowing a quadrupedal robot to imitate the movement behaviour of a dog. Similarly, Xu and Karamouzas [38] use an adversarial approach to learn movements from human motion capture. Similar to us, these papers match markers between motion capture representations and robots. However, in the first, a highly engineered pipeline relies on a) the ability to compute the inverse kinematics of the target platform, and b) a hand-engineered reward function. In the latter, imitation learning is used for learning behaviour, but neither method optimizes for morphology.

3 Preliminaries

Imitation Learning as distribution-matching For a given expert state-action trajectory $\tau^E = (\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{s}_n, \boldsymbol{a}_n)$, the imitation learning task is to learn a policy $\pi^I(\boldsymbol{a}|\boldsymbol{s})$ such that the resulting behaviour best matches the demonstrated behaviour. This problem setting can be understood as minimizing a divergence, or alternative measures, $D(q(\tau^E), p(\tau^I|\pi^I))$ between the demonstrator trajectory distribution $q(\tau^E)$ and the trajectory distribution of the imitator $p(\tau^I|\pi^I)$ induced by its policy π^I (see e.g. [27] for further discussion).

While there are multiple paradigms of imitation learning, a recently popular method is *adversarial* imitation learning, where a discriminator is trained to distinguish between policy states (or state-action pairs) and demonstrator states [19, 26]. The discriminator is then used for providing rewards to an RL algorithm which maximizes them via interaction. In the remainder of the paper we will be focusing on two adversarial methods with a divergence-minimization interpretation which we will now discuss in more detail.

Generative Adversarial Imitation Learning (GAIL)

GAIL trains a standard classifier using a logistic loss which outputs the probability that a given state comes from the demonstration trajectories [19]. The reward function is chosen to be a function of the classifier output. Many options are given in literature for the choice of reward, evaluated extensively by Orsini et al. [26]. Different choices of rewards correspond to different distance measures in terms of the optimization problem. Here, we consider the reward introduced by Fu et al. [11]:

$$r(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}) = \log(\psi(\boldsymbol{s}_t)) - \log(1 - \psi(\boldsymbol{s}_t)), \tag{1}$$

where ψ is a classifier trained to distinguish expert data from the imitator. Maximizing this reward corresponds to minimizing the Kullback-Leibler divergence between the demonstrator and policy state-action marginals [12].

State-Alignment Imitation Learning (SAIL)

In contrast to GAIL, SAIL [23] uses a Wasserstein-GAN-style [1] critic instead of the standard logistic regression-style discriminator. Maximizing the SAIL reward corresponds to minimizing the Wasserstein distance [37] between demonstrator and policy state-marginals (see Liu et al. [23] for details).

4 A General Framework for Co-Imitation

We formalize the problem of co-imitation as follows. Consider an expert MDP described by (S^E, A^E, p^E, p_0^E), with state space S^E, action space A^E, initial state distribution $p_0^E(\mathbf{s}_0^E)$, and the transition probability $p^E(\mathbf{s}_{t+1}^E|\mathbf{s}_t^E, \mathbf{a}_t^E)$. Furthermore, assume that the generally unknown expert policy is defined as $\pi^E(\mathbf{a}_t^E|\mathbf{s}_t^E)$. In addition, an imitator MDP is defined by $(S^I, A^I, p^I, p_0^I, \pi^I, \xi)$, where the initial state distribution $p^I(\mathbf{s}_0^I|\xi)$ and transition probability $p^I(\mathbf{s}_{t+1}^I|\mathbf{s}_t^I, \mathbf{a}_t^I, \xi)$ are parameterized by a morphology-parameter ξ. The trajectory distribution of the expert is given by

$$q(\tau^E) = p_E(\mathbf{s}_0^E) \prod_{t=0}^{T-1} p_E(\mathbf{s}_{t+1}^E|\mathbf{s}_t^E, \mathbf{a}_t^E)\pi^E(\mathbf{a}_t^E|\mathbf{s}_t^E), \tag{2}$$

while the imitator trajectory distribution is dependent on the imitator policy $\pi^I(\mathbf{a}|\mathbf{s}, \xi)$ and chosen morphology ξ

$$p(\tau^I|\pi^I, \xi) = p_I(\mathbf{s}_0^I|\xi) \prod_{t=0}^{T-1} p_I(\mathbf{s}_{t+1}^I|\mathbf{s}_t^I, \mathbf{a}_t^I, \xi)\pi^I(\mathbf{a}_t^I|\mathbf{s}_t^I, \xi). \tag{3}$$

It follows that the objective of the co-imitation problem is to find an imitator policy π^{I*} and the imitator morphology ξ^* such that a chosen probability-distance divergence measure or function $\mathcal{D}(\cdot, \cdot)$ is minimized, i.e.

$$\xi^*, \pi^{I*} = \arg\min_{\xi, \pi^I} \mathcal{D}(q(\tau^E), p(\tau^I|\pi^I, \xi)). \tag{4}$$

For an overview of potential candidate distance measures and divergences see e.g. Ghasemipour et al. [12]. For the special case that state-spaces of expert and imitator do not match, a simple extension of this framework is to assume two transformation functions $\phi(\cdot) : S^E \to S^S$, and $\phi_\xi(\cdot) : S^I \to S^S$ where S^S is a *shared feature space*. For simplicity we overload the notation and use $\phi(\cdot)$ for both the demonstrator and imitator state-space mapping.

5 Co-Imitation by State Distribution Matching

We consider in this paper the special case of co-imitation by state distribution matching and present two imitation learning methods adapted for the learning of behaviour and design. The co-imitation objective from Eq. (4) is then reformulated as

$$\mathcal{D}(q(\tau^E), p(\tau^I|\pi^I, \xi)) \stackrel{\text{def}}{=\!=} \mathcal{D}(q(\phi(\mathbf{s}^E)), p(\phi(\mathbf{s}^I)|\pi^I, \xi)). \tag{5}$$

Similar to Lee et al. [21] we define the marginal *feature-space* state distribution of the imitator as

$$p(\phi(\boldsymbol{s}^I)|\pi^I,\xi) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{\substack{\boldsymbol{s}_0^I\sim p^I(\boldsymbol{s}_0^I|\xi)\\ \boldsymbol{a}_t^I\sim\pi^I(\boldsymbol{a}_t^I|\boldsymbol{s}_t^I,\xi)\\ \boldsymbol{s}_{t+1}^I\sim p^I(\boldsymbol{s}_{t+1}^I|\boldsymbol{s}_t^I,\boldsymbol{a}_t^I,\xi)}} \left[\frac{1}{T}\sum_{t=0}^{T}\mathbb{1}(\phi(\boldsymbol{s}_t^I)=\phi(\boldsymbol{s}^I))\right],$$

while the feature-space state distribution of the demonstrator is defined by

$$q(\phi(\boldsymbol{s}^E)) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{\substack{\boldsymbol{s}_0^E\sim p^E(\boldsymbol{s}_0^E)\\ \boldsymbol{a}_t^E\sim\pi^E(\boldsymbol{a}_t^E|\boldsymbol{s}_t^E)\\ \boldsymbol{s}_{t+1}^E\sim p^E(\boldsymbol{s}_{t+1}^E|\boldsymbol{s}_t^E,\boldsymbol{a}_t^E)}} \left[\frac{1}{T}\sum_{t=0}^{T}\mathbb{1}(\phi(\boldsymbol{s}_t^E)=\phi(\boldsymbol{s}^E))\right].$$

Intuitively, this formulation corresponds to matching the visitation frequency of each state in the expert samples in the shared feature space. In principle any transformation that maps to a shared space can be used. For details of our specific choice see Section 6.1. Importantly, this formulation allows us to frame the problem using any state marginal matching imitation learning algorithms for policy learning. See Ni et al. [25] for a review of different algorithms.

An overview of CoIL is provided in Algorithm 1. We consider a set of given demonstrator trajectories T_E, and initialize the imitator policy as well as an initial morphology ξ_0. Each algorithm iteration begins with the robot training the imitator policy for the current morphology ξ for N_ξ iterations, as discussed in Section 5.1. The set of collected imitator trajectories T_ξ^I and morphology are added to the dataset Ξ. Then, the morphology is optimized by computing the distribution distance measure following Algorithm 2. The procedure is followed until convergence, finding the morphology and policy that best imitate the demonstrator. We follow an alternating approach between behaviour optimization and morphology optimization as proposed by prior work such as Luck et al. [24].

5.1 Behaviour Adaptation

Given the current morphology ξ of an imitating agent, the first task is to optimize the imitator policy π^I with $\pi^I_{\text{next}} = \arg\min_{\pi^I} \mathcal{D}(q(\phi(\mathrm{s}^E)), p(\phi(\mathrm{s}^I)|\pi^I,\xi))$.

The goal is to find an improved imitator policy π^I_{next} which exhibits behaviour similar to the given set of demonstration trajectories T^E. This policy improvement step is performed in lines 4–11 in Algorithm 1. We experiment with two algorithms: GAIL and SAIL, which learn discriminators as reward functions $r(\boldsymbol{s}_t, \boldsymbol{s}_{t+1})$. Following [26] we use SAC, a sample-efficient off-policy model-free algorithm as the reinforcement learning backbone for both imitation learning algorithms (line 10 in Alg. 1). To ensure that the policy transfers well to new morphologies, we train a single policy π^I conditioned both on s_t^I and on ξ. Data from previous morphologies is retained in the SAC replay buffer. Further details about the changes made to the these algorithms for behaviour adaptation in the co-imitation setting are stated in the Appendix.

Algorithm 1 Co-Imitation Learning (CoIL)

Input: Set of demonstration trajectories $\mathrm{T}^E = \{\tau_0^E, ...\}$
1: Initialize π^I, $\xi = \xi_0$, $\mathrm{T}^I = \varnothing$, $\Xi = \varnothing$, and RL replay R_{RL}
2: **while** not converged **do**
3: Initialize agent with morphology ξ
4: **for** $n = 1, \dots, N_\xi$ episodes **do**
5: With current policy π^I sample state-action trajectory $(\boldsymbol{s}_0^I, \boldsymbol{a}_0^I, \dots, \boldsymbol{s}_t^I, \boldsymbol{a}_t^I, \boldsymbol{s}_{t+1}^I, \dots)$
6: Add tuples $(\boldsymbol{s}_t^I, \boldsymbol{a}_t^I, \boldsymbol{s}_{t+1}^I, \xi)$ to replay R_{RL}
7: Add state-trajectory $\tau_{n,\xi}^I = (\boldsymbol{s}_0^I, \boldsymbol{s}_1^I, ...)$ to T^I
8: Compute rewards $r(\phi(\boldsymbol{s}_t^I), \phi(\boldsymbol{s}_{t+1}^I))$ using IL strategy
9: Add rewards $r(\phi(\boldsymbol{s}_t^I), \phi(\boldsymbol{s}_{t+1}^I))$ to R_{RL}
10: Update policy $\pi^I(a_t^I|s_t^I,\xi)$ using RL and R_{RL}
11: **end for**
12: Add (ξ, T_ξ^I) to Ξ with $\mathrm{T}_\xi^I = \{\tau_{0_\xi,\xi}^I, ..., \tau_{N_\xi,\xi}^I\}$
13: $\xi = \text{Morpho-Opt}(\mathrm{T}^E, \Xi)$ ▷ Adapt Morphology (Alg. 2)
14: **end while**

5.2 Morphology Adaptation

Adapting the morphology of an agent requires a certain exploration-exploitation trade-off: new morphologies need to be considered, but changing it too radically or too often will hinder learning. In general, co-imitation is challenging because a given morphology can perform poorly due to either it being inherently poor for the task, or because the policy has not converged to a good behaviour.

Previous approaches have focused on using either returns averaged over multiple episodes, (e.g [15]) or the Q-function of a learned policy [24] to evaluate the fitness of given morphology parameters. They then perform general-purpose black-box optimization along with exploration heuristics to find the next suitable candidate to evaluate. Since both approaches rely on rewards, in the imitation learning setting they correspond to maximizing the critic's *approximation* of the distribution distance. This is because the rewards are outputs of a neural network that is continuously trained and, hence, inherently non-stationary. Instead, we propose to minimize in the co-imitation setting the *true* quantity of interest, i.e. the distribution distance for the given trajectories.
Given the current imitator policy $\pi^I(\boldsymbol{a}_t^I|\boldsymbol{s}_t^I, \xi)$ our aim is to find a candidate morphology minimizing the objective

$$\xi_{\text{next}} = \arg\min_{\xi} \mathcal{D}(q(\phi(\mathbf{s}^E)), p(\phi(\mathbf{s}^I)|\pi^I, \xi)). \tag{6}$$

Bayesian Morphology Optimization

In order to find the optimal morphology parameters we perform Bayesian Optimization (BO), which is a sample-efficient optimization method that learns a probabilistic surrogate model [10]. Here, we use a Gaussian Process (GP) [31] as surrogate to learn the relationship between the parameters ξ and the distance $\mathcal{D}(q(\phi(\mathbf{s}^E)), p(\phi(\mathbf{s}^I)|\pi^I, \xi))$. This relationship is modeled by the GP prior

$$g(\xi) = \mathcal{GP}(\mu(\xi), k(\xi, \xi')), \tag{7}$$

Algorithm 2 Bayesian Morphology Optimization

Output: ξ_{next}, next candidate morphology
1: **procedure** MORPHO-OPT(T^E,Ξ)
2: Define observations $X = \{\xi_n\}, \forall \xi_n \in \Xi$
3: Compute $Y = \{y_n\}$, $\forall(\xi_n, \mathrm{T}_n^I) \in \Xi$ ▷ Using Eq. (9)
4: Fit GP $g(\xi)$ using X and Y
5: $\mu_g(\tilde{\xi}), \sigma_g(\tilde{\xi}) = p(g(\tilde{\xi})|X, Y)$ ▷ Compute posterior
6: $\alpha(\tilde{\xi}) = \mu_g(\tilde{\xi}) + \beta\, \sigma_g(\tilde{\xi})$ ▷ Compute UCB
7: $\xi_{\text{next}} = \arg\min_{\tilde{\xi}} \alpha(\tilde{\xi})$ ▷ Provide next candidate
8: **end procedure**

where $\mu(\cdot)$ defines the mean function, and $k(\cdot, \cdot)$ the kernel (or covariance) function. We show that adapting the morphology in CoIL via this approach increases performance over the co-adaptation and imitation baselines in Section 6.
Modelling the relationship between the parameters ξ and the distance $\mathcal{D}(\cdot, \cdot)$ is surprisingly challenging because the policy evolves over time. This means that morphologies evaluated early in training are by default worse than those evaluated later, and thus should be trusted less. The BO algorithm alleviates this problem by re-fitting the GP at each iteration using only the most recent episodes. By learning the surrogate GP model $g(\xi)$ we can explore the space of morphologies and estimate their performance without gathering new data. The optimization problem can be defined as

$$\xi_{\text{next}} = \arg\min_{\xi} g(\xi), \tag{8}$$

where ξ_{next} is the next proposed morphology to evaluate. The GP model is trained using as observations the set of morphologies used in behaviour adaptation $X = \{\xi_n\}, \forall \xi_n \in \Xi$, and as targets $Y = \{y_0, \cdots, y_N\}$ the mean distribution distance for each morphology, that is

$$y_n = \frac{1}{N_\xi} \sum_{k=0}^{N_\xi} \mathcal{D}\left(q(\tau^E), p(\tau_{k,\xi}^I|\pi^I, \xi)\right). \tag{9}$$

The predictive posterior distribution is given by $p(g(\tilde{\xi})|X, Y) = \mathcal{N}(\tilde{\xi}|\mu_g(\tilde{\xi}), \sigma_g(\tilde{\xi}))$, where $\tilde{\xi}$ is the set of test morphologies and $\mu_g(\tilde{\xi})$ and $\sigma_g(\tilde{\xi})$ are the predicted mean and variance. In order to trade-off between exploration and exploitation we use the Upper Confidence Bound (UCB) as acquisition function $\alpha(\tilde{\xi}) = \mu(\tilde{\xi}) + \beta\sigma(\tilde{\xi})$, where β is a parameter that controls the exploration. The morphology optimization procedure is depicted in Algorithm 2. The GP is optimized by minimizing the negative marginal log-likelihood (MLL). Then, the posterior distribution is computed for the set of test morphologies $\tilde{\xi}$. The values of $\tilde{\xi}$ for each task are described in the Appendix. Finally, the acquisition function is computed and used to obtain the next proposed morphology. We provide in the Appendix a study of the morphology optimization routine comparing the proposed BO approach to Random Search (RS) [2], and covariance matrix adaptation evolutionary strategy (CMA-ES) [18].

6 Experiments

Our experimental evaluation aims at answering the following research questions:
(Q1) Does imitation learning benefit from co-adapting the imitator's morphology?
(Q2) How does the choice of the imitation learning algorithm used with CoIL impact the imitator's morphology?
(Q3) Is morphology adaptation with CoIL able to compensate for major morphology differences, such as a missing joint or the transfer from a real to a simulated agent?
To answer these questions, we devised a set of experiments across a range of setups and imitation learning methods.

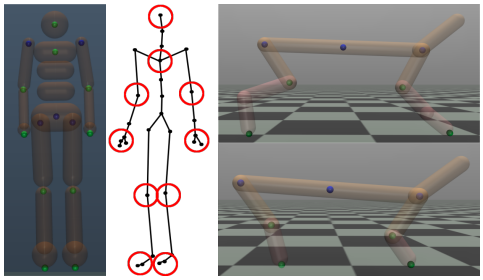


Figure 3: **Left:** Markers used for matching the MuJoCo Humanoid to motion capture data. **Right:** Markers used for the Cheetah tasks. Green markers are used as data, while blue markers serve as reference points for green markers (more details in Section 6.1).

6.1 Experimental Setup

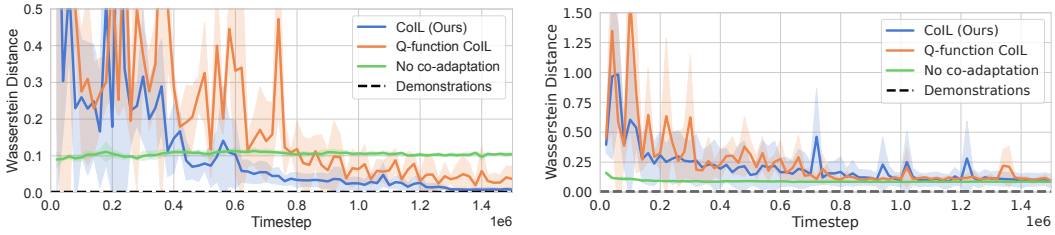
In all our experiments, we use the MuJoCo physics engine [36] for simulating the dynamics of agents. As discussed in Algorithm 1, the policies are trained using the same morphology for $N_\xi = 20$ episodes. The BO algorithm details as well as more detailed technical information can be found in the Appendix.

Joint feature space: As discussed in Section 4 our method assumes that demonstrator and imitator states are in different state-spaces. To address this mismatch, the proposed method maps the raw state observations from the demonstrator and the imitator to a common feature space. The selection of the feature space can be used to influence which parts of the behaviour are to be imitated. In our setups, we manually selected the relevant features by placing markers along each of the limbs in both experimental setups, as shown in Figure 3). The feature space is then composed of velocities and positions of these points relative to the base of their corresponding limb (marked in blue in the figure).

Evaluation Metric: Evaluating the accuracy of imitation in a quantitative manner is not straightforward, because—in general—there does not exist an explicit reward function that we can compare performance on. While most imitation learning works use task-specific rewards to evaluate imitation performance, it is not a great proxy for e.g. learning similar gaits. Recently, previous work in state-marginal matching has used forward and reverse KL divergence as a performance metric [25]. However, rather than evaluating the KL divergence, we opted for using the Wasserstein distance [37] as the evaluation metric. The main motivation behind this choice was that this metric corresponds to the objective optimized by SAIL and PWIL, two state-of-the-art imitation learning algorithms. Additionally, it constitutes a more intuitive quantity for comparing 3D positions of markers than KL divergence—the Wasserstein distance between the expert and imitator feature distributions corresponds to the average distance by which markers of the imitator need to be moved in order for the two distributions to be aligned. Therefore, for both morphology optimization and evaluation we use the exact Wasserstein distance between *marker position* samples from the demonstrator $q(\phi(\boldsymbol{s}^E))$ and imitator $p(\phi(\boldsymbol{s}^I)|\pi^I, \xi)$ state marginal distributions. This also allows us to avoid an additional scaling hyperparameter when optimizing for morphologies, since velocities and positions have different scales. The Wasserstein distances are computed using the *pot* package [9]. For all runs we show the mean and standard deviation of 3 seeds represented as the shaded area.

6.2 Co-Imitation from Simulated Agents

We adapt the HalfCheetah setup from OpenAI Gym [3] by creating a version with two leg-segments instead of three (see Fig. 3). We then collect the demonstration datasets by generating expert trajectories from a policy trained by SAC using the standard running reward for both variants of the environment. We refer to these tasks as *3to2* and *2to3* corresponding to imitating a 3-segment demonstrator using a 2-segment imitator and vice versa. For both experiments we used 10 episodes of 1000 steps as demonstration data. Further details can be found in the Appendix.



(a) Imitation of 2-joint Cheetah using 3-joint Cheetah. (b) Imitation of 3-joint Cheetah using 2-joint Cheetah.

Figure 4: Wasserstein distance for 3 seeds between demonstrator and imitator for both HalfCheetah tasks. While co-imitation via CoIL (blue) outperforms SAIL (green) in 2to3 (a), all methods show the same performance in 3to2 (b).

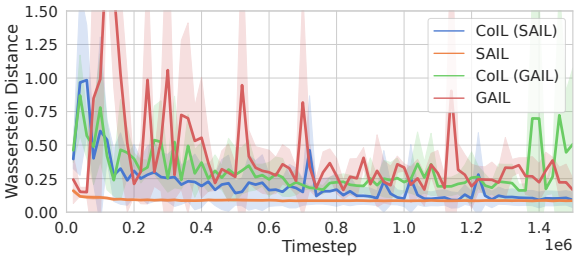
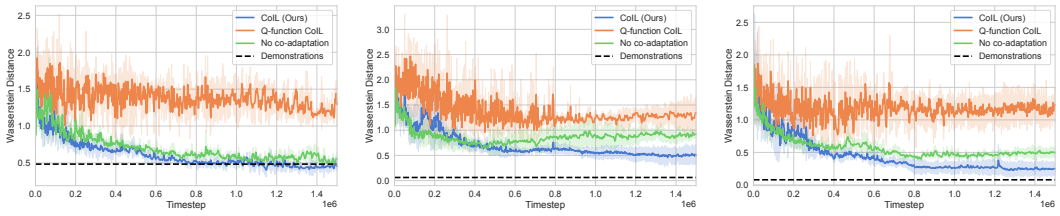


Figure 5: Wasserstein distance for three seeds between demonstrator and imitator trajectories on the *3to2* Cheetah task on co-imitation (CoIL) and pure imitation learning algorithms (SAIL, GAIL).

First, we answer RQ1 by investigating whether co-adapting the imitator's morphology is at all beneficial for their ability to replicate the demonstrator's behaviour, and—if so—how different state marginal matching imitation learning algorithms perform at this task (RQ2). To this end, we analyze the performance of two imitation learning algorithms, GAIL and SAIL on the HalfCheetah setup, with and without co-adaptation. We use BO as the morphology optimizer, as it consistently produced good results in preliminary experiments (see Appendix). The performance for both imitation algorithms on the *3to2* task is shown in Figure 5. We observe that SAIL outperforms GAIL both with and without morphology adaptation. Our results indicate that this task does not benefit from morphology optimization as SAIL and CoIL achieve similar performance. However, it is encouraging to note that CoIL does not decrease performance even when the task does not benefit from co-adaptation. Based on these results we select SAIL as the main imitation learning algorithm due to its higher performance over GAIL.

Figure 4 shows the results in the two HalfCheetah morphology transfer scenarios. To address RQ3, we compare CoIL to two other co-imitation approaches: using the cheetah without morphology adaptation, as well as to using the Q-function method adapted from Luck et al. [24]. Since this method is designed for the standard reinforcement learning setting, we adapt it to the imitation learning scenario by using SAIL to imitate the expert trajectories, and iteratively optimizing the morphology using the Q-function. See the Appendix for further details of this baseline. In the 3to2 domain transfer scenario (Figure 4b), where the gait of a more complex agent is to be reproduced on a simpler setup, the results are even across the board. All methods are able to imitate the demonstrator well, which indicates that this task is rather easy, and that co-adaptation does not provide much of a benefit. On the other hand, in the 2to3 scenario shown in Figure 4a, after co-adaptation with CoIL, the more complex Cheetah robot is able to reproduce the gait of the simpler, two-segment robot very closely. A closer look at the results reveals that the morphology adaptation algorithm achieves this by setting the length of the missing link in each leg to a very small, nearly zero value (see Appendix). Thus, at the end of training, CoIL can recover the true morphology of the demonstrator.

While the Q-function optimization procedure from [24] also optimizes for the Wasserstein distance metric via the reward signal, the final performance is somewhat worse. We hypothesize that with more interaction time the Q-function version would reach the performance of CoIL on this simple task.



(a) Soccer kick (b) Jogging (c) Walking

Figure 6: The average Wasserstein distances (of 10 test episodes, 3 seeds) for the three CMU motion-capture to MuJoCo Humanoid tasks. The baseline "Demonstrations" refers to the mean distance between the individual demonstration trajectories. We can see that CoIL (blue) consistently performs better than the compared methods, even reaching the mean distance between the demonstration trajectories (black) in the soccer task.

6.3 Co-Imitation from Human Behaviour

Next, we address RQ3 by evaluating CoIL in a more challenging, high-dimensional setup, where the goal is to co-imitate demonstration data collected from a real-world human using a simplified simulated agent. Here, we use a Humanoid robot adapted from OpenAI Gym [3] together with the CMU motion capture data [4] as our demonstrations. This setup uses a similar marker layout to HalfCheetah's, with markers placed at each joint of each limb, with additional marker in the head (see Figure 3 for a visualization). We follow the same relative position matching as in the Cheetah setup. We also include the absolute velocity of the torso in the feature space to allow modelling forward motion.

The performance of the Humanoid agent on imitating three tasks from CMU motion capture dataset: *walking*, *jogging*, and *soccer kick*, is shown in Figure 6. We observe that, in all three tasks, CoIL reproduces the demonstrator behaviour most faithfully. A comparison of the morphology and behaviour learned by CoIL vs standard imitation learning (here SAIL) in the jogging task is shown in Figure 2. In the *soccer kick* task, CoIL's performance matches the distance between individual demonstrations, while for the two locomotion tasks—jogging and walking—there is still a noticeable performance gap between CoIL and the individual expert demonstrations (with $p = 0.0076$, Wilcoxon signed rank test).

We also observe that, in all three setups, not performing co-adaptation at all (and using the default link length values for the OpenAI Gym Humanoid instead) outperforms co-adaptation with the Q-function objective. We hypothesize that this counter-intuitive result might stem from the increased complexity of the task—learning a sensible Q-function in the higher-dimensional morphology- and state feature-space of Humanoid is likely to require a much larger amount of data, and thus a longer interaction time. In contrast, optimizing the morphologies using the Wasserstein distance directly makes the optimization procedure easier, since it does not rely on the Q-function "catching up" with changes both to policy and to the adversarial reward models used in GAIL and SAIL.

7 Conclusion

In this paper we presented Co-Imitation Learning (CoIL): a methodology for co-adapting both the behaviour of a robot and its morphology to best reproduce the behaviour of a demonstrator. This is, to the best of our knowledge, the first deep learning method to co-imitate both morphology and behaviour using only demonstration data with no pre-defined reward function. We discussed and presented a version of CoIL using state distribution matching at its core for co-imitating a demonstrator in the special case of mismatching state and action spaces. The capability of CoIL to better co-imitate behaviour and morphology was demonstrated in a difficult task where a simulated humanoid agent has to imitate real-world motion capturing data of a human.

Although we were able to show that CoIL outperforms non-morphology-adapting imitation learning techniques in the presented experiment using real-world data, we did not consider or further investigate the inherent mismatch between physical parameters (such as friction, contact-forces, elasticity, etc.) of simulation and real world or the use of automatic feature-extraction mechanisms. We think that these challenges present interesting avenues for future research and that the presented co-imitation methodology opens up a new exciting research space in the area of co-adaptation of agents.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[2] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012. URL http://jmlr.org/papers/v13/bergstra12a.html.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[4] CMU. Cmu graphics lab motion capture database. http://http://mocap.cs.cmu.edu/, 2019.

[5] Robert Dadashi, Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal wasserstein imitation learning. In *ICLR 2021-Ninth International Conference on Learning Representations*, 2021.

[6] Siddharth Desai, Ishan Durugkar, Haresh Karnan, Garrett Warnell, Josiah Hanna, and Peter Stone. An imitation from observation approach to transfer learning with dynamics mismatch. *Advances in Neural Information Processing Systems*, 33:3917–3929, 2020.

[7] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43, 1995. doi: 10.1109/MHS.1995.494215.

[8] Arnaud Fickinger, Samuel Cohen, Stuart Russell, and Brandon Amos. Cross-domain imitation learning via optimal transport. In *International Conference on Learning Representations*, 2021.

[9] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[10] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[11] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkHywl-A-.

[12] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pages 1259–1277. PMLR, 2020.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

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

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

[16] 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*, pages 1861–1870. PMLR, 2018.

[17] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[18] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. doi: 10.1162/106365601750190398.

[19] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[20] Eddy Hudson, Garrett Warnell, Faraz Torabi, and Peter Stone. Skeletal feature compensation for imitation learning with embodiment mismatch. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2482–2488. IEEE, 2022.

[21] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

[22] Thomas Liao, Grant Wang, Brian Yang, Rene Lee, Kristofer Pister, Sergey Levine, and Roberto Calandra. Data-efficient learning of morphology and controller for a microrobot. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2488–2494. IEEE, 2019.

[23] Fangchen Liu, Zhan Ling, Tongzhou Mu, and Hao Su. State alignment-based imitation learning. In *International Conference on Learning Representations*, 2019.

[24] 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.

[25] Tianwei Ni, Harshit Sikchi, Yufei Wang, Tejus Gupta, Lisa Lee, and Ben Eysenbach. f-irl: Inverse reinforcement learning via state marginal matching. In *Conference on Robot Learning*, pages 529–551. PMLR, 2021.

[26] Manu Orsini, Anton Raichuk, Léonard Hussenot, Damien Vincent, Robert Dadashi, Sertan Girgin, Matthieu Geist, Olivier Bachem, Olivier Pietquin, and Marcin Andrychowicz. What matters for adversarial imitation learning? *Advances in Neural Information Processing Systems*, 34, 2021.

[27] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[28] Jahng-Hyon Park and Haruhiko Asada. Concurrent design optimization of mechanical structure and control for high speed robots. In *American Control Conference*, number 30, pages 2673–2679, 1993.

[29] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning Agile Robotic Locomotion Skills by Imitating Animals. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi: 10.15607/RSS.2020.XVI.064.

[30] Jordan B Pollack, Hod Lipson, Sevan Ficici, Pablo Funes, Greg Hornby, and Richard A Watson. Evolutionary techniques in physical robotics. In *International Conference on Evolvable Systems*, pages 175–186. Springer, 2000.

[31] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X.

[32] Charles Schaff, David Yunis, Ayan Chakrabarti, and Matthew R Walter. Jointly learning to construct and control agents using deep reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9798–9805. IEEE, 2019.

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

[34] Karl Sims. Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372, 1994.

[35] Avi Singh, Larry Yang, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. In *Robotics: Science and Systems*, 2019.

[36] 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.

[37] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[38] Pei Xu and Ioannis Karamouzas. A gan-like approach for physics-based imitation learning and interactive character control. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 4(3):1–22, 2021.