

Versatile Skill Control via Self-supervised Adversarial Imitation of Unlabeled Mixed Motions

Chenhao Li¹, Sebastian Blaes¹, Pavel Kolev¹, Marin Vlastelica¹, Jonas Frey², and Georg Martius¹

Abstract—Learning diverse skills is one of the main challenges in robotics. To this end, imitation learning approaches have achieved impressive results. These methods require explicitly labeled datasets or assume consistent skill execution to enable learning and active control of individual behaviors, which limits their applicability. In this work, we propose a cooperative adversarial method for obtaining single versatile policies with controllable skill sets from unlabeled datasets containing diverse state transition patterns by maximizing their discriminability. Moreover, we show that by utilizing unsupervised skill discovery in the generative adversarial imitation learning framework, novel and useful skills emerge with successful task fulfillment. Finally, the obtained versatile policies are tested on an agile quadruped robot called Solo 8 and present faithful replications of diverse skills encoded in the demonstrations.

I. INTRODUCTION

Reinforcement Learning (RL) has demonstrated its capability of learning diverse and complex skills for robotic platforms. In the field of legged systems, RL has achieved success in learning-based quadrupedal locomotion control in challenging environments [1]–[4]. Typically, deep RL techniques learn desired behaviors motivated by optimizing reward functions that are specifically tailored for the training task. As a result, it can sometimes become very demanding to develop complex skills with reward engineering, where various terms of motivation and regularization need to be carefully designed and balanced.

Given the availability of some expert references, Imitation Learning (IL) allows an agent to learn to reproduce expert behavior. In particular, Generative Adversarial Imitation Learning (GAIL, [5]) employs Generative Adversarial Networks (GANs, [6]), which train a policy to deceive an *imitation* discriminator that constantly tries to distinguish state transitions generated between the policy and the reference data distribution. The output of the discriminator is used as a training signal that encourages the learning agent to generate similar behaviors to the demonstration [7], [8]. However, given a large dataset of *unlabeled* motion clips with diverse behaviors, extracting and learning *individual* sensible skills can be challenging. Such scenarios are commonly encountered in motion captures from creatures or underactuated experts, whose motion execution produces intrinsic noise and inconsistency and thus consists of miscellaneous behaviors. Due to the unknown skill types in the dataset, supervised learning techniques fail to find correspondence between the policy and the individual skills it replicates.

¹Max Planck Institute for Intelligent Systems, Tübingen, Germany
{firstname.lastname}@tuebingen.mpg.de

²Robotic Systems Lab, ETH Zurich, Zurich, Switzerland
jonfrey@ethz.ch

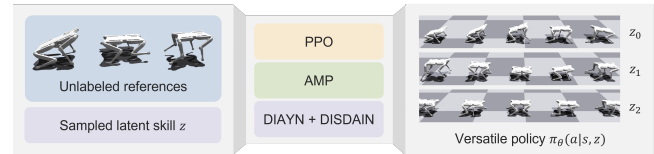


Fig. 1: Method overview. Cooperative Adversarial Self-supervised Skill Imitation (CASSI) enables skill extraction and learning from unlabeled diverse references by motivating diversity in a generative adversarial imitation learning setting using unsupervised learning techniques. The resulting versatile policy allows active control of individual skills. Supplementary videos and implementation details for this work are available at <https://sites.google.com/view/icra2023-cassi/home>.

In this case, unsupervised learning methods could help to address the problem. Unsupervised skill discovery has enabled learning agents to obtain intrinsic behaviors without explicit task rewards. One promising attempt for unsupervised skill discovery is based on maximizing the *discriminability* of skills represented by latent variables on which a policy is conditioned [9]–[13]. Policy training signals are often constructed from variational approximations of the mutual information between latent variables and traversed state histories using a learned *skill* discriminator. Such mutual information encodes both the diversity in state transitions and the predictability of them given known latent skills [12]. In this setting, strong discriminating performances are expected cooperatively by both the policy and the discriminator, where the latter is trained with supervised learning techniques. The resulting policy is capable of producing consistently distinguishable behavioral patterns, or *skills*.

In this work, we present Cooperative Adversarial Self-supervised Skill Imitation (CASSI), where we show that by applying unsupervised skill discovery in a generative adversarial imitation learning framework, a quadrupedal system manages to learn versatile policies that extract sensible high-level skills from a reference dataset containing diverse unlabeled motions (Fig. 1). The learned policy enables active control of the individual skills embedded in the reference dataset, which manifests itself by its execution of consistently distinguishable behaviors. The policy is also deployed and tested on the real robot without further adaptation. To the best of our knowledge, this is the first time that individual skills are extracted and learned simultaneously from an unlabeled dataset motivated by unsupervised RL algorithms. In summary, our contributions include: (i) A cooperative adversarial

self-supervised skill imitation method for extracting and learning diverse skills from unlabeled references. **(ii)** Fidelity and diversity evaluation of learned versatile policies using an oracle classifier. **(iii)** Experimental validation in simulation and on a quadruped robot.

II. RELATED WORK

Recent development in robotics has created many potential applications that require intelligent systems to not only make decisions but also perform expected physical movements. While learning a task is often formulated as an optimization problem, it has become widely recognized that having prior knowledge provided by an expert can result in more effective and efficient learning than attempting to solve the problem from scratch [14], [15].

Instead of relying on a handcrafted imitation objective, GAIL techniques train an adversarial discriminator to distinguish between behaviors generated by the agent and expert demonstrations [5], [16]. The effectiveness of imitation is thus measured by the performance of the discriminator, whose output encodes the similarity descriptions. To extend the applicability of GAIL to accommodate large datasets of unstructured motion clips, [7] employ an adversarial RL procedure that automatically selects the motion to perform in the reference, dynamically interpolating and generalizing from the dataset. While this method has achieved success in learning adaptive motions, *active* control over individual skill execution relies still on labeled motion datasets. To this end, one may either collect disparate reference motions and learn individual skills separately [8], or stack the training of multiple discriminators corresponding to each skill in parallel [17]. Both methods require *supervision* from prior knowledge of motion labels in the datasets. However, given a large dataset of diverse *unlabeled* motion clips, enabling autonomous acquisitions of individual skills requires unsupervised training techniques.

Unsupervised RL is motivated by intrinsic skill development of intelligent agents that are believed to learn in the absence of supervision in order to acquire repurposable task-agnostic knowledge. These skills can be quickly and efficiently utilized when specific tasks are later defined. Achieving this goal requires specifying a learning objective that ensures that each skill individually is distinct and that the skills collectively explore large parts of the state space [10]. In this setting, mutual information is commonly established as a notion of empowerment of an intrinsically motivated agent [18]. It is shown that a discriminability objective is equivalent to maximizing the mutual information between the skill and some aspect of the induced trajectory [19], [20]. [9], [21] try to maximize the mutual information between the skill and initial and final states. [10], [13] attempt to maximize the mutual information between the skill and states along the trajectory. [12], [22] propose to maximize the mutual information between the skill and the following state conditioned on the current state. With the skill diversity motivated by mutual information maximization in various settings, diverse skills are acquired with minimal

supervision. Finally, unsupervised RL algorithms have been shown effective in conjunction with task rewards, which serve as sparse guidance for generating regularized and robust behaviors [23], [24].

III. APPROACH

In this section, we describe our method, which enables learning from unlabeled demonstrations in a generative adversarial setting using unsupervised skill discovery techniques. In the following context, we denote the discriminator distinguishing state transitions from the policy and reference in the generative adversarial imitation learning as the *imitation discriminator* (d_ψ), and the discriminator distinguishing different skills yielded by the policy in the unsupervised skill discovery as the *skill discriminator* (q_ϕ).

A. Generative Adversarial Imitation

Our generative adversarial imitation framework is built upon the Adversarial Motion Prior (AMP, [7]). In this framework, the imitation discriminator reflects the imitation performance and thus specifies the training signal for the policy. It is crucial to select an appropriate set of features that provide effective learning feedback. We consider the imitation observation space \mathcal{O}^I . The reference demonstrations are formulated as sequences of $o_t^I \in \mathcal{O}^I$, where the full state space \mathcal{S} of the underlying Markov Decision Process can be mapped to the imitation observation space \mathcal{O}^I with a function $f^I : \mathcal{S} \rightarrow \mathcal{O}^I$. We denote trajectory segments of length H^I preceding time t by $\mathbf{o}_t^I = (o_{t-H^I+1}^I, \dots, o_t^I)$ for the reference observations and $\mathbf{s}_t = (s_{t-H^I+1}, \dots, s_t)$ for the states induced by the policy. For clarity, we omit the time index in the following. To simplify notation, we write $f^I(\mathbf{s})$ to express that each state in \mathbf{s} is mapped to \mathcal{O}^I . In our experiments, we select linear and angular velocities of the robot base in the robot frame, measurement of the gravity vector in the robot frame, the base height, and joint angular position and velocity as the observation space \mathcal{O}^I . As such, the discriminator’s goal in this setup is to distinguish samples of the policy transition distribution d^π from the reference motion distribution d^M .

1) *Imitation Discriminator Formulation:* Similar to [7], we use the least-squares GAN (LSGAN) loss [25] for discriminator optimization. Using H^I -step inputs and a gradient penalty, the discriminator objective is formulated as

$$\mathbb{E}_{d^M} \left[(d_\psi(\mathbf{o}^I) - 1)^2 \right] + \mathbb{E}_{d^\pi} \left[(d_\psi(f^I(\mathbf{s})) + 1)^2 \right] + w^{\text{GP}} \mathbb{E}_{d^M} \left[\|\nabla_{\mathbf{o}^I} d_\psi(\mathbf{o}^I)\|_2^2 \right], \quad (1)$$

where the last term denotes the penalty for non-zero gradients on samples from the dataset with weight w^{GP} to stabilize training [7].

Intuitively, the LSGAN loss forces the discriminator to output +1 for samples from the reference motion and -1 for those from the policy. In contrast, the policy is trained to deceive the discriminator by generating transition patterns

similar to those present in the reference dataset. The reward function for training the policy is then given by

$$r^I = \max \left[0, 1 - 0.25 \left(d_\psi(f^I(\mathbf{s})) - 1 \right)^2 \right]. \quad (2)$$

The imitation reward r^I provides a well-scaled output bounded between 0 and 1 which eases downstream policy learning.

B. Mutual Information Maximization

Our unsupervised skill discovery framework is built upon Diversity Is All You Need (DIAYN, [10]) with the Discriminator Disagreement Intrinsic reward (DISDAIN, [13]) for exploration motivation.

In the unsupervised RL setting, the agent seeks to develop a set of skills, indexed by a latent variable $z \in \mathcal{Z}$ and represented by a policy $\pi_\theta(\cdot | s, z)$ parameterized by θ . The latent skills $z \sim p_z$ are sampled at the beginning of each trajectory and then fixed over the episode. Therefore, each skill z represents a temporally extended behavior within a state sequence of length H^S . Similarly, we consider a function $f^S : \mathcal{S} \rightarrow \mathcal{O}^S$ that maps the full state space \mathcal{S} to the skill observation space \mathcal{O}^S . Thus, the transition in the extracted features can be represented by $\mathbf{o}_t^S = (o_{t-H^S+1}^S, \dots, o_t^S) \sim \tau^S(\pi_\theta(z))$, where τ^S denotes the state visitation distribution along the trajectory. Again, we omit the time index in the following for clarity. Clearly, trajectories sampled from the policy conditioned on a particular skill depend on the selection of z . We highlight this dependence by writing \mathbf{o}_z^S .

1) *Skill Discriminator Observation Space*: Since skills refer to consistently distinguishable behavioral motifs in the states of interest, how a skill distinguishes itself from other skills, or *discriminability* between skills, depends crucially on the definition of the skill observation space \mathcal{O}^S . In unsupervised RL literature, skills are often identified over only simple state transition patterns (e.g. different moving directions, speeds, or reached positions) [9], [10], [12], [13], as it is typically hard to extract high-level skill descriptions without any supervision. In our work, as the policy space is constrained by an imitation target, we are able to define a skill observation space \mathcal{O}^S that specifies high-level distinction (e.g. trot, leap).

Note that the selection of states in \mathcal{O}^S depends on users' interest. If we are interested in obtaining different base motion patterns, including only the base information in \mathcal{O}^S suffices. If we also aim to recover transitions with different joint configurations (e.g. different gaits) in the reference dataset, including joint information will help with discrimination. In addition, we assume that the skill space \mathcal{Z} is discrete and with cardinality N_z , although much of the discussion may extend to continuous skill space.

2) *Variational Approximation of Mutual Information*: A large and growing class of objectives for unsupervised skill discovery are derived from maximizing the mutual information between the latent skill z and the resulting trajectories

of extracted features \mathbf{o}_z^S . The formulation is expressed as

$$\begin{aligned} \mathcal{F}(\theta) &= \mathcal{I}(z; \mathbf{o}_z^S) = \mathcal{H}(z) - \mathcal{H}(z | \mathbf{o}_z^S) \\ &= \mathbb{E}_{z \sim p_z, \mathbf{o}_z^S \sim \tau^S(\pi_\theta(z))} [\log p(z | \mathbf{o}_z^S) - \log p_z(z)]. \end{aligned} \quad (3)$$

In practice, a variational approximation of the intractable conditional distribution $p(z | \mathbf{o}_z^S)$ with a learned parametric model $q_\phi(z | \mathbf{o}_z^S)$ is often applied to obtain a lower bound of $\mathcal{F}(\theta)$ [26]. The model $q_\phi(z | \mathbf{o}_z^S)$ is commonly referred to as a discriminator, as it is trained to discriminate between skills from state transitions. In our work, it is termed as the *skill discriminator*, to distinguish itself from the *imitation discriminator* (d_ψ) used in the generative adversarial imitation setting.

Optimizing the lower bound with respect to the policy parameters θ corresponds to training the policy with a skill reward

$$r^S = \log q_\phi(z | \mathbf{o}_z^S) - \log p_z(z). \quad (4)$$

To learn a full repertoire of skills, the skill prior is typically fixed to be uniform [10], [21], [27], in which case $-\log p_z(z) = \log N_z$. An arbitrary discriminator which simply ignores the trajectory will give a zero skill reward. In contrast, a perfect discriminator will yield a skill reward of $\log N_z$ [13].

To tighten the lower bound, the discriminator should approximate the true distribution $p(z | \mathbf{o}_z^S)$, which results in a supervised learning setting where the training data is collected from policy roll-outs. In our work, the skill discriminator is implemented as a classifier whose objective is to assign the correct skill to a given style observation by minimizing the cross-entropy classification loss.

As such, both the policy and the skill discriminator expect a strong discriminating performance by maximizing and tightening the lower bound of the mutual information cooperatively.

3) *Policy Exploration Bonus*: Ideally, the policy presents different state transitions conditioned on different skills. The discriminator receives these transitions and attempts to decode the original skill z . However, at the beginning of the training, before the policy relates its behavior to the skill, the agent may yield similar transitions under different latent variables z . As the skill is sampled randomly, this will generate mislabeled data which confuses the training of the discriminator. When the discriminator fails to distinguish the transition patterns, the skill reward will be trivial and thus does not generate any information for the policy to diversify its behaviors. This dead loop is termed as the *pessimistic exploration* problem in unsupervised skill discovery [13]. The reason behind this is that when the policy tries to maximize the skill reward r^S , it tends to reduce the *epistemic* uncertainty by visiting the same states it has visited before. To encourage exploration, [13] propose the use of a discriminator ensemble of size N and a resulting DISDAIN reward that compensates for the increase of the epistemic

uncertainty when the policy visits new states:

$$r^D = \mathcal{H} \left[\frac{1}{N} \sum_{i=1}^N q_{\phi_i}(z | \mathbf{o}_z^S) \right] - \frac{1}{N} \sum_{i=1}^N \mathcal{H} [q_{\phi_i}(z | \mathbf{o}_z^S)], \quad (5)$$

where ϕ_i corresponds to the parameters of the ensemble member i .

For trajectories with rich training data for the skill discriminator, the ensemble members should agree and result in a zero DISDAIN reward. In contrast, for state transitions of high discriminator disagreement, this reward will be positive and thus encourage exploration.

C. Overview

Note that the imitation reward r^I , the skill reward r^S , and the DISDAIN reward r^D are task-agnostic. This allows the training of primitive skills that facilitate downstream learning such as hierarchical RL when a task is later specified. Alternatively, we are able to learn high-level tasks alongside self-supervised skill extraction. In this setting, we define task reward r^T , in addition to some regularization terms r^R that enforce stable policy outputs on the real platform. Putting everything together, the total reward that the policy receives encompasses five parts

$$r = w^T r^T + w^I r^I + w^S r^S + w^D r^D + w^R r^R, \quad (6)$$

where w denotes the weight of each term and $w^D = w^R = 1.0$ is kept constant throughout our work.

The joint optimization of the policy and the imitation discriminator utilizes *adversarial* training. The imitation discriminator tries to distinguish state transitions between the agent and the reference motion, whereas the policy tries to make this difficult by generating similar behaviors. Meanwhile, the joint training of the policy and the skill discriminator forms a *cooperative* game. The agent samples a skill and tries to present a distinct behavior for the ease of discrimination by the skill discriminator, and the skill discriminator tries to decode the original skill and yield training signals to reward the policy for diversifying its skills.

Figure 2 provides a schematic overview of our method, and an algorithm overview is detailed in Algorithm 1.

IV. EXPERIMENTS

We evaluate our method on the Solo 8 robot, an open-source research quadruped robot that performs a wide range of physical actions [28], in simulation and on the real system.

The unlabeled motion dataset is constructed using mixed trajectories induced by individual expert policies learned in a previous work [8], including **crawl**, **walk**, **trot**, **leap**, **wave**, and **stilt**. These colors are used consistently throughout the paper. For each motion, 1000 trajectories are recorded in simulation from parallel robot instances with randomized mechanical properties. Each trajectory contains 120 consecutive time steps. To break down the consistency in skill execution, these consecutive state transitions are sliced into small motion clips of horizon length 8. In all of our experiments, we use Proximal Policy Optimization (PPO, [29]) in Isaac Gym [30].

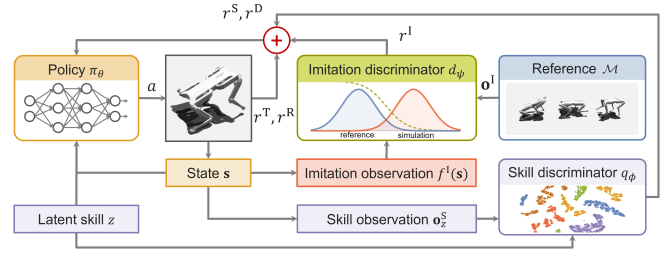


Fig. 2: System overview. Given an unlabeled dataset, an imitation discriminator learns sampled state transition patterns from demonstrations. At the beginning of each episode, a latent skill variable z is sampled as input to the policy. A skill discriminator is trained to decode the original skill from these motions. The policy is rewarded for making the skills distinguishable.

Algorithm 1 CASSI

- 1: **Input:** unlabeled reference dataset \mathcal{M} ; imitation and skill feature maps f^I, f^S ; transition horizons H^I, H^S ; latent skill cardinality N_z
- 2: initialize state transition buffer τ , replay buffer B
- 3: **for** learning iterations = 1, 2, ... **do**
- 4: sample latent skill variable $z \sim p_z$
- 5: collect \mathbf{s} with policy π_θ conditioned on skill z
- 6: extract features \mathbf{o}_z^S by applying f^S to \mathbf{s}
- 7: compute r^I, r^S, r^D according to Equations 2, 4 and 5
- 8: calculate total reward r according to Eq. 6
- 9: fill replay buffer B with $(\mathbf{s}, z, \mathbf{o}_z^S)$
- 10: **for** policy learning epoch = 1, 2, ... **do**
- 11: sample transition mini-batches $b^\pi \sim B$
- 12: update V and π_θ with PPO or another RL algorithm
- 13: **end for**
- 14: **for** imitation discriminator learning epoch = 1, 2, ... **do**
- 15: sample transition mini-batches $b^\pi \sim B$ and $b^M \sim \mathcal{M}$
- 16: update d_ψ using b^π and b^M according to Eq. 1
- 17: **end for**
- 18: **for** skill discriminator learning epoch = 1, 2, ... **do**
- 19: sample transition mini-batches $b^\pi \sim B$ with bagging
- 20: update q_ϕ using b^π with cross-entropy or loss
- 21: **end for**
- 22: **end for**

A. Skill Extraction and Imitation

In this section, we set task reward weight $r^T = 0$ and compare CASSI with spectral clustering and AMP in extracting and learning skills from unlabeled datasets.

1) *Spectral Clustering*: A potential strategy to obtain individual skills from unlabeled references is to perform pre-clustering on the dataset and learn each separately. To test its effectiveness, we perform spectral clustering with k -Nearest Neighbors on the reference dataset, compute the error rate with respect to the true labels, and compare it with the skill discriminator learned during the training of our method. We report the best result in Table I.

TABLE I: Learned skill discriminator vs. spectral clustering.

	CASSI		Spectral Clustering				
Horizon	8	8	20	30	40	60	120
Error %	0.016	70.2	29.1	27.3	21.1	17.9	0.08

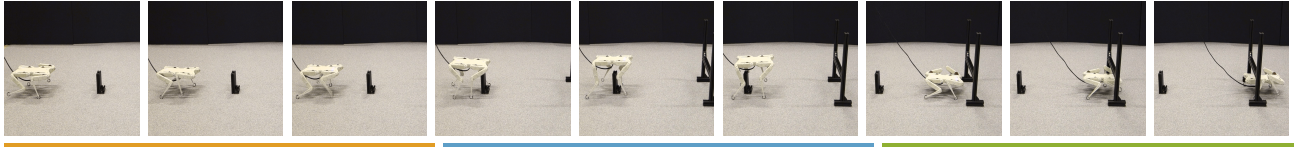


Fig. 3: Motion sequence composed of multiple skills, executed on the real Solo 8 in an obstacle avoidance environment.

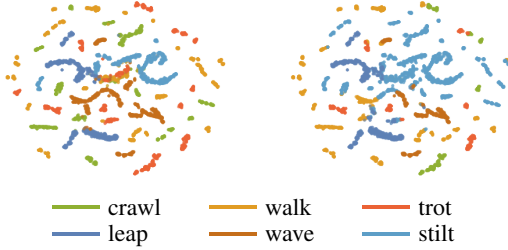


Fig. 4: Predicted motion clusters by spectral clustering (right) with a horizon of 8 on a random subset of the dataset (1/10 of tuples from each motion) compared with ground truth labels (left). The prediction error of spectral clustering reports 43.57% on this subset.

The result reveals that the skill discriminator learned using our method with a horizon of only 8 achieves better discriminating performance than spectral clustering on the whole trajectories. This indicates that for spectral clustering to work with decent accuracy, strong assumptions of consistent skill execution over trajectories with much longer horizons must be made. The spectral clustering label assignment is visualized using t-SNE in Fig. 4 next to the ground-truth labels.

2) *Oracle Classifier*: As we construct the dataset by mixing individual expert motion clips, we are able to evaluate the diversity and fidelity of the learned skills by training an oracle classifier C using ground-truth labels. We denote the true label by $c \in \mathcal{C}$ and the true number of skills present in the dataset by N_c , where \mathcal{C} denotes the true skill space. In our experiments, $\mathcal{C} = \{\text{crawl}, \text{walk}, \text{trot}, \text{leap}, \text{wave}, \text{stilt}\}$ and $N_c = 6$. Note that N_c is unknown to the design of the skill discriminator. Thus, the predefined number of skills N_z is a tuning parameter.

We write $p(c | \mathbf{o}_z^C)$ to represent the probability of the oracle classifier predicting the true skill c given a state trajectory of horizon length H^C , which is induced by the policy conditioned on a sampled latent skill z . When the policy conditioned on a skill z is able to generate a state trajectory that aligns well with the transition patterns exerted by a certain skill c in the true skill space \mathcal{C} , the probability $p(c | \mathbf{o}_z^C)$ is large for this specific skill $c \in \mathcal{C}$ and small for all other skills $c' \in \mathcal{C} \setminus \{c\}$. We thus can quantify the diversity and fidelity of the skills learned by the policy π_θ , using the following entropy expressions:

$$\text{div}(\pi_\theta) = \mathcal{H} \left[\frac{1}{N_z} \sum_{z=1}^{N_z} p(c | \mathbf{o}_z^C) \right], \quad (7)$$

$$\text{fid}(\pi_\theta) = -\frac{1}{N_z} \sum_{z=1}^{N_z} \mathcal{H} [p(c | \mathbf{o}_z^C)]. \quad (8)$$

Intuitively, fidelity is maximized if the policy is very certain that every generated trajectory corresponds to a true skill present in the reference dataset when conditioned on a latent skill. In contrast, diversity is maximized if the policy is able to generate trajectories corresponding to different true skills present in the reference dataset when conditioned on different latent skills.

We visualize the training performance in terms of $\text{div}(\pi_\theta)$, $\text{fid}(\pi_\theta)$, and $p(c | \mathbf{o}_z^C)$ in Figures 5 and 6 for $w^I = 1.0$, $w^S = 0.5$, and $N_z = 6$. Note that under the assumption $N_z = N_c$, the learned policy establishes a bijective correspondence $g : \mathcal{Z} \rightarrow \mathcal{C}$, where all reference motions are assigned to distinct latent variables. In addition, the policy encounters skill collapse with AMP only, due to the absence of the skill reward r^S which promotes discriminability among learned behaviors. We further analyze cases with $N_z \neq N_c$ in Sec. IV-B.

B. Task Execution

In this section, we evaluate the skills learned using CASSI with an additional velocity tracking task, which is achieved by maximizing the reward $r^T = \exp\{-\sigma^{-2}(u-v_x)^2\}$, where u denotes the velocity command sampled within the range $[0, 1]$. We denote by v_x the linear velocity of the robot base, expressed in the robot frame, and set the constant scaling factor $\sigma^2 = 0.25$. The task reward weight is set constant $w^T = 1.0$.

1) *Novel Skill Discovery*: In practice, motion types presented in the reference dataset are often unknown. Without prior knowledge of the actual number of component motions N_c , the choice of the latent skill cardinality N_z may require additional effort. We evaluate skill extraction performance for different N_z using the oracle classifier.

Figure 7b reveals that when $N_z = N_c$, all skills in the reference dataset are extracted and assigned to a distinct

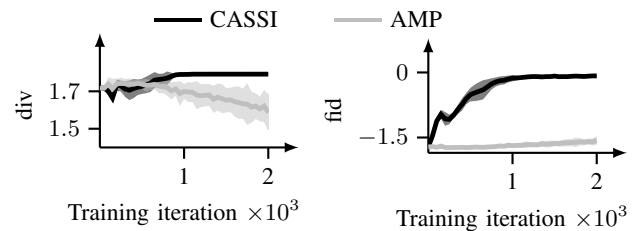


Fig. 5: Diversity (left, Eq. 7) and fidelity (right, Eq. 8) of the learned skills for CASSI and AMP ($w^S = 0$) over iterations.

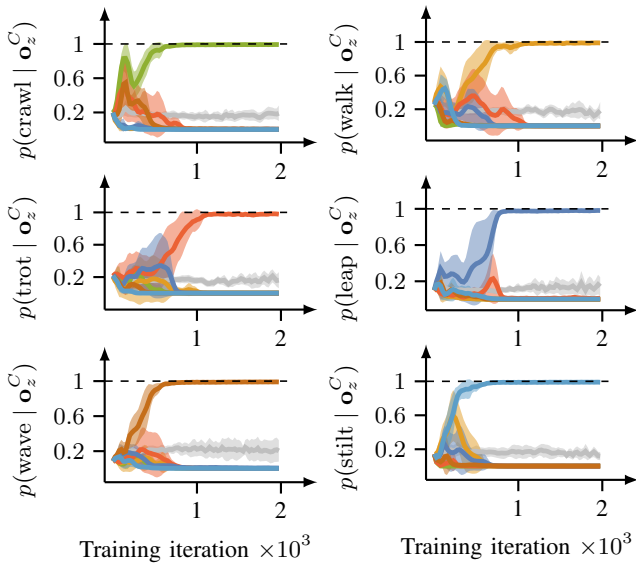


Fig. 6: Training performance for CASSI (colored) and AMP (grey, $w^S = 0$) over iterations. AMP witnesses a skill collapse where the policy generates single state transition patterns regardless of the latent skill z . In contrast, CASSI encourages each latent skill z to converge to a distinct reference motion c . As the mapping between \mathcal{Z} and \mathcal{C} may differ between runs, the plots show results after aligning correspondence across seeds.

latent variable z . In contrast, for $N_z \neq N_c$, it is observed that new skills emerge in a form of motion interpolation, whose existence is reflected by mixed predictions by the oracle classifier given a latent skill z (e.g. $z = 2, 3$ in Fig. 7a and $z = 6, 7$ in Fig. 7c). These skills present “mixed” features of the references. In addition, the skill-conditioned task rewards $r^T(z)$ report consistently high value for all latent skills with different choices of N_z . This indicates that the new skills also yield high task performance while presenting mixed motion patterns. Within the scope of the defined task, these discovered skills are considered to be both *novel* and *useful*.

The policy training using CASSI can be understood as a matching procedure between the latent skills and the reference skills. In this setting, the true skills $c \in \mathcal{C}$ can be viewed as vertices that define a skill space. The imitation reward r^I motivates the policy to learn behaviors that stay within this skill space, while the skill reward r^S motivates the behavior associated with each z to stay far away from the

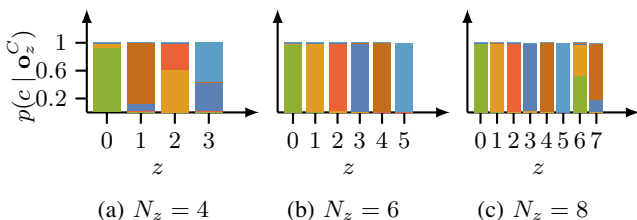


Fig. 7: Oracle skill prediction of learned latent skills z .

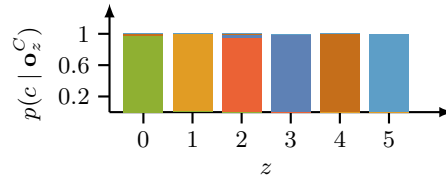


Fig. 8: Oracle predictions for the skills executed on the real Solo 8.

averaged behavior [31]. For $N_z = N_c$, each z takes a vertex of the skill space and is assigned to a distinct $c \in \mathcal{C}$. For $N_z > N_c$, more latent variables need to be fit into the skill space. Once the vertices are taken (pure skills), the remaining z have to stay in the interior of the space (mixed skills). For $N_z < N_c$, the latent variables have larger freedom to choose their relative position in the skill space to stay distant from each other, presenting both “pure” and “mixed” behaviors.

2) *Real-system Deployment*: We deploy the learned policy on the real system and record 10 state transition trajectories of 120 time steps when conditioned on each skill. The recorded motions are evaluated again with the oracle classifier as depicted in Fig. 8. We also illustrate the continuous change of behaviors within a trajectory by switching latent skill variables in Fig. 9. Figure 3 provides a motion sequence on Solo 8 with skill execution by the versatile policy.

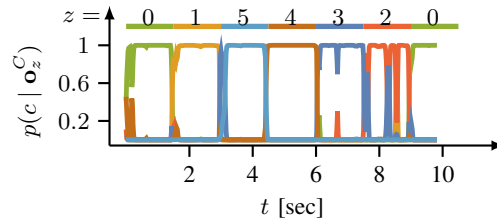


Fig. 9: Skill sequence executed on Solo 8 (top). Oracle predictions of the motions recorded from the robot (bottom).

V. CONCLUSION

In this work, we propose an adversarial imitation method named CASSI that is capable of extracting and learning individual skills from unlabeled datasets with diverse behaviors by utilizing unsupervised skill discovery techniques. Our results highlight the importance of the skill reward, whose absence results in skill collapses. The experiments also indicate that CASSI allows extracting robust versatile policies capable of active skill control with both high fidelity and diversity with respect to the original reference motions. Furthermore, policies can be learned with task specifications and are able to transfer to the real system without further adaptation. Finally, our experiments confirm the emergence of novel skills that yield high task performance and present common features shared by original reference motions.

ACKNOWLEDGMENT

This work is supported by the Volkswagen Stiftung (No 98 571) and the Tübingen AI Center (BMBF FKZ: 01IS18039B).

REFERENCES

- [1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.
- [2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.
- [3] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems XVII (RSS)*, 2021.
- [4] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
- [5] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [7] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–20, 2021.
- [8] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimmering, and G. Martius, "Learning agile skills via adversarial imitation of rough partial demonstrations," *Conference on Robot Learning*, 2022.
- [9] K. Gregor, D. J. Rezende, and D. Wierstra, "Variational intrinsic control," in *International Conference on Learning Representations, Workshop Track Proceedings*, 2017.
- [10] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," *International Conference on Learning Representations*, 2019.
- [11] S. Hansen, W. Dabney, A. Barreto, T. Van de Wiele, D. Warde-Farley, and V. Mnih, "Fast task inference with variational intrinsic successor features," *International Conference on Learning Representations*, 2020.
- [12] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman, "Dynamics-aware unsupervised discovery of skills," *International Conference on Learning Representations*, 2020.
- [13] D. Strouse, K. Baumli, D. Warde-Farley, V. Mnih, and S. Hansen, "Learning more skills through optimistic exploration," *International Conference on Learning Representations*, 2022.
- [14] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*. Springer, 2008, pp. 1371–1394.
- [15] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [16] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.
- [17] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter, "Advanced skills through multiple adversarial motion priors in reinforcement learning," *arXiv preprint arXiv:2203.14912*, 2022.
- [18] S. Mohamed and D. Jimenez Rezende, "Variational information maximisation for intrinsically motivated reinforcement learning," *Advances in neural information processing systems*, vol. 28, 2015.
- [19] C. Florensa, Y. Duan, and P. Abbeel, "Stochastic neural networks for hierarchical reinforcement learning," *International Conference on Learning Representations*, 2017.
- [20] K. Hausman, J. T. Springenberg, Z. Wang, N. Heess, and M. Riedmiller, "Learning an embedding space for transferable robot skills," in *International Conference on Learning Representations*, 2018.
- [21] K. Baumli, D. Warde-Farley, S. Hansen, and V. Mnih, "Relative variational intrinsic control," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6732–6740.
- [22] V. Campos, A. Trott, C. Xiong, R. Socher, X. Giró-i Nieto, and J. Torres, "Explore, discover and learn: Unsupervised discovery of state-covering skills," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1317–1327.
- [23] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson, "Maven: Multi-agent variational exploration," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [24] S. Kumar, A. Kumar, S. Levine, and C. Finn, "One solution is not all you need: Few-shot extrapolation via structured maxent RL," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8198–8210, 2020.
- [25] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.
- [26] D. Barber and F. Agakov, "Information maximization in noisy channels: A variational approach," *Advances in Neural Information Processing Systems*, vol. 16, 2003.
- [27] J. Achiam, H. Edwards, D. Amodei, and P. Abbeel, "Variational option discovery algorithms," *arXiv preprint arXiv:1807.10299*, 2018.
- [28] F. Grimmering, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols, J. Fiene, A. Badri-Spröwitz, and L. Righetti, "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3650–3657, 2020.
- [29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [30] V. Makovychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *Neural Information Processing Systems Track on Datasets and Benchmarks 1*, 2021.
- [31] T. Zahavy, B. O'Donoghue, G. Desjardins, and S. Singh, "Reward is enough for convex mdp," in *Advances in Neural Information Processing Systems*, 2021, pp. 25 746–25 759.