2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Diverse Skill Discovery in Fourier Latent Space via Unsupervised Learning

Ruopeng Cui[1], Yucong Sun[2], Xizhou Bu[1], Chao Wang[3,†], Wei Li[1,†]

***Abstract*—Unsupervised skill discovery acquires a diverse repertoire of skills through intrinsic motivation, offering the potential to alleviate the labor-intensive reward engineering in reinforcement learning and the reliance on costly task-specific data in imitation learning. However, such methods typically measure diversity based on single-step states, neglecting the trajectory phase coherence, whose absence disrupts the smoothness of state transitions. In this work, we explore skills in Fourier latent space via a simple mutual-information-based reward function, aiming to train a single versatile policy capable of executing diverse state transition patterns. Specifically, we utilize a spatio-temporal representation learned through a Periodic Autoencoder, which effectively captures the periodic or quasi-periodic nature of motion. These features, rather than raw states, are used to measure skill diversity. We validate our method on the 12-DOF quadruped robot Unitree A1, achieving varied gaits. Simulation results show that our method reduces high-frequency power by 73%, while improving state space coverage by 133% compared to the baseline. To accomplish specific tasks, we trained a high-level controller to orchestrate the learned skills, which improves training efficiency. Real-world experiments demonstrate that the learned skills can reliably execute tasks.**

I. Introduction

Deep Reinforcement Learning (DRL) has empowered quadruped robots to execute highly dynamic and agile maneuvers [1]–[4] and achieve robust locomotion control in challenging environments [5]–[8] by optimizing task-specific reward functions. However, the necessity of meticulously designing and tuning weights for multiple reward components, such as task performance, safety constraints, and energy consumption [1], [2], [9], significantly impedes the broader application of reinforcement learning. To circumvent the challenges of reward engineering, Imitation Learning (IL) learns policies by minimizing the pose error between the robot and reference motions derived from motion capture or animation data. Nevertheless, IL methods face significant challenges due to the considerable cost of acquiring high-quality reference data for specific tasks and the kinematic mismatch between the reference data and the physical robot [3], [10], [11]. Furthermore, some imitation learning methods based on generative adversarial networks often introduce additional hyperparameters, complicating the training process [12]–[14].

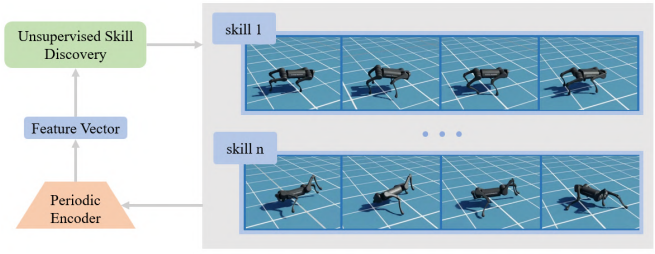

Fig. 1. Method overview. Fourier Latent Skill Discovery (FLSD) employs Fourier latent features to enable unsupervised skill discovery, producing policies that allow active control of diverse and distinct skills.

Unsupervised skill discovery methods typically maximize the state-skill mutual information, where skills are represented as latent variables that condition the policy [15], [16]. This intrinsic motivation mechanism drives agents to autonomously explore diverse behaviors, thereby eliminating the reliance on task-specific reward functions or additional imitation data [15], [17]. Within this framework, the reward signal is derived from a learned discriminator that performs a variational approximation of the mutual information. However, most existing approaches mainly rely on single-step states to measure and maximize diversity [16], [18]–[20]. To maximize inter-skill distinguishability, these methods enforce state exclusivity by prohibiting shared or similar states between skills. This prevents the algorithm from discovering transitional skills that connect existing behaviors, thereby limiting the diversity of skills. Furthermore, for robot locomotion, some certain states may serve as shared transitions essential for motion smoothness. Prohibiting inter-skill sharing of such states can induce unnatural movements, especially in quadruped robots where spatial-temporal synchronization is vital for coordinated gait.

To address these limitations, we present Fourier Latent Skill Discovery (FLSD), a framework that enables unsupervised skill discovery to generate policies by leveraging Fourier latent dynamics (Fig. 1). Specifically, during the

We thank Shanghai Institute for Mathematics and Interdisciplinary Sciences (SIMIS) for their financial support. This research was funded by SIMIS under grant number SIMIS-ID-2025-RB. The authors are grateful for the resources and facilities provided by SIMIS, which were essential for the completion of this work.

† Corresponding authors.

[1]Ruopeng Cui, Xizhou Bu and Wei Li are with College of Intelligent Robotics and Advanced Manufacturing, Fudan University, Shanghai, 200438, China. {23210860100, xzbu24}@m.fudan.edu.cn, fudan_liwei@fudan.edu.cn

[2]Yucong Sun is with the School of Aerospace Engineering, Tsinghua University, Beijing, 100084, China. sun-yucong@foxmail.com

[3]Chao Wang is with the Center for Automous Systems Research, Qiyuan Laboratory, Beijing, 100095, China. wangchao1@qiyuanlab.com

exploration process of the unsupervised skill discovery algorithm, a Periodic Autoencoder [21], [22] is employed to encode the robot's state sequences. The resulting latent vectors include the phase information of motion and effectively capture motion characteristics, which are used to evaluate skill diversity. The learned policy can consistently execute behaviors with clearly distinguishable characteristics.

In summary, our main contributions include:

- A novel Fourier Latent Skill Discovery framework is proposed to scale unsupervised skill discovery algorithms to quadruped robots.
- The experimental results demonstrate that the Fourier latent variables effectively capture the periodic and quasi-periodic characteristics of motion, leading to a 73% reduction in high-frequency power in the skills discovered by the unsupervised skill discovery algorithm. Furthermore, evaluating diversity from the perspective of state sequences enables the algorithm to achieve broader coverage of the state space by 133%.
- For specific downstream tasks, we trained a high-level controller to orchestrate the learned skills. Our real-world experiments demonstrate that these skills can serve as an effective low-level controller, enabling reliable task execution.

II. Related Work

A. Motion Representation

Generative models, such as Variational Autoencoders (VAEs, [23], [24]) and Generative Adversarial Networks (GANs, [25], [26]), are frequently utilized to learn motion representations in an unsupervised manner. By mapping high-dimensional motion data into a compact latent space, these approaches effectively capture the underlying features and temporal dynamics of long-horizon trajectories. However, the representations generated by these methods cannot be well aligned temporally. [27]–[29] address this by using phase variables that describe motion progression, but these methods are limited to contact-based movements and require careful computation of phase labels.

[21] developed the PAE framework, which integrates frequency-domain mapping within an encoding process to extract complex phase representations from unlabeled kinematic data. This method organizes movement patterns into a structured embedding, ensuring that the similarity measure within this latent space provides a physically more meaningful metric compared to distances calculated in the raw data domain. [22] further extends PAE by performing regression on multi-step forward prediction through propagation of latent dynamics.

B. Unsupervised Skill Discovery

Unsupervised skill discovery aims to acquire reusable skills without the need for extrinsic reward signals or task-specific data, akin to the autonomous exploration capabilities observed in biological agents. Intrinsic motivation in these frameworks is generally formulated as maximizing the mutual information between specific trajectory features and a sampled latent skill variable. Consequently, this mechanism facilitates the training of skill-conditioned policies that yield a diverse repertoire of distinct behaviors [15], [18], [19], [30].

Recent studies have explored how unsupervised skill discovery can equip quadruped robots with diverse locomotion skills. [31] employs off-policy methods to train quadruped robots in real-world environments. However, the resulting skills exhibit limited mobility, characterized by slow and simplistic movements resembling a shuffle. [32] employs a multi-constraint optimization framework to generate diverse navigation skills, but focuses on local navigation, which constrains its applicability to broader robotic tasks, and the generated skills are limited to forward movement patterns. In [13], unsupervised skill discovery is embedded as a skill extraction module within an adversarial imitation learning framework to learn skills from unannotated motion datasets. Despite its effectiveness, this approach relies on pre-trained policies to generate motion data, which limits its scalability. Additionally, previous methods are tested on lightweight robots with simplified dynamics and fail to learn dynamic behaviors such as high-speed locomotion or in-place rotation. Furthermore, [13] and [32] introduce numerous task-specific reward functions, significantly increasing the burden of reward design and tuning. In contrast, our work demonstrates that diverse skills can be effectively discovered even with the use of a simple mutual-information-based reward function, which are validated on a quadruped robot with complex dynamics.

III. Method

To address the limitations of existing unsupervised skill discovery algorithms, we propose a novel approach that maps state sequences into a Fourier latent space. By quantifying diversity through state representations in this space, the method generates smooth motions, enabling the discovered skills to be applicable to robotic systems.

A. Motion Representation in Fourier Latent Space

We leverage a Periodic Autoencoder (PAE) [21], [22] to capture the latent dynamics and temporal dependencies in the motion. The architecture is built upon a temporal convolutional autoencoder structure [33], augmented by a frequency domain mapping to introduce a necessary inductive bias.

The motion at time step t is formulated as a 33-dimensional observation vector $\boldsymbol{o}_t^m$:

$$\boldsymbol{o}_t^m = \begin{bmatrix} \boldsymbol{v}_t & \boldsymbol{\omega}_t & \boldsymbol{g}_t & \boldsymbol{\theta}_t & \dot{\boldsymbol{\theta}}_t \end{bmatrix}^T, \tag{1}$$

comprising the base linear velocity $\boldsymbol{v}_t$, the base angular velocity $\boldsymbol{\omega}_t$, the gravity direction vector in the body frame $\boldsymbol{g}_t$, the joint angle $\boldsymbol{\theta}_t$ and the joint angular velocity $\dot{\boldsymbol{\theta}}_t$. A motion trajectory of length H ending at time step t is defined as $\boldsymbol{o}_t^H = (\boldsymbol{o}_{t-H+1}^m, \ldots, \boldsymbol{o}_t^m) \in \mathbb{R}^{33\times H}$. This sequence corresponds to a centered time window $\mathcal{T} = \left[-\frac{\Delta t}{2}, -\frac{\Delta t}{2} + \frac{\Delta t}{H-1}, -\frac{\Delta t}{2} + \frac{2\Delta t}{H-1}, \ldots, \frac{\Delta t}{2}\right] \in \mathbb{R}^H$, where the current observation $\boldsymbol{o}_t^m$ aligns with the timestamp $\frac{\Delta t}{2}$.

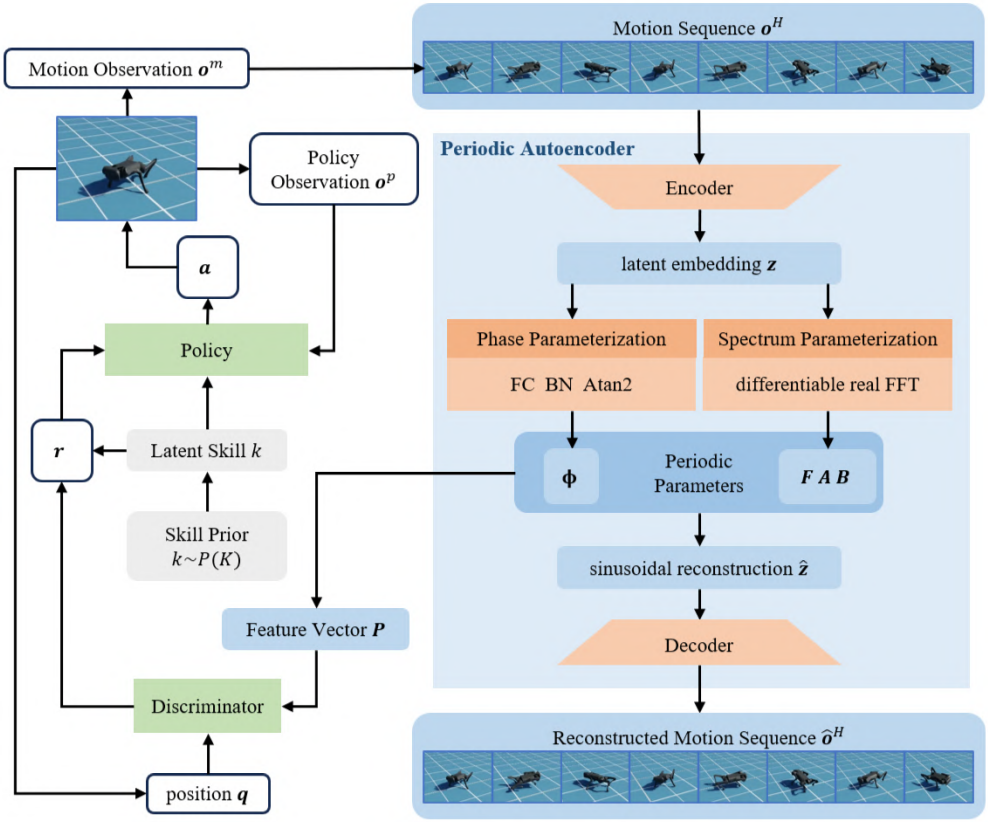

Fig. 2. Overview of the proposed FLSD framework. The policy, discriminator, and PAE are jointly optimized in a collaborative framework. The agent selects a skill and executes a specific state transition pattern. PAE extracts motion trajectory features to offer a more appropriate measure of diversity, aiding the discriminator in distinguishing skills. Meanwhile, the discriminator works to infer the sampled skill and provide training signals to encourage the policy to diversify its skills.

We utilize a temporal convolutional autoencoder to derive a compact representation of state sequences. The input motion sequence $\boldsymbol{o}^H$ is first encoded into a latent variable $\boldsymbol{z}$, and subsequently, the processed latent trajectory $\hat{\boldsymbol{z}}$ is decoded back to the original state space:

$$\boldsymbol{z} = \mathbf{enc}(\boldsymbol{o}^H), \quad \hat{\boldsymbol{o}}^H = \mathbf{dec}(\hat{\boldsymbol{z}}) \tag{2}$$

where $\boldsymbol{z}, \hat{\boldsymbol{z}} \in \mathbb{R}^{D \times H}$ and D represents the number of latent channels. The reconstruction loss is defined as the Mean Squared Error between the original trajectory $\boldsymbol{o}^H$ and the reconstructed trajectory $\hat{\boldsymbol{o}}^H$:

$$L_{\text{PAE}} = \frac{1}{H} \left\| \boldsymbol{o}^H - \hat{\boldsymbol{o}}^H \right\|_2^2 \tag{3}$$

To capture the underlying dynamics of the motion, we introduce a frequency-domain inductive bias within the latent space, enforcing each latent trajectory in $\hat{\boldsymbol{z}}$ to be parameterized as a sinusoidal function. Specifically, using the frequency $\boldsymbol{F}$, amplitude $\boldsymbol{A}$, offset $\boldsymbol{B}$, and the learned phase shift $\boldsymbol{\Phi}$, combined with the known time window $\mathcal{T}$, the reconstructed latent trajectory segments are computed as:

$$\hat{\boldsymbol{z}} = \boldsymbol{A} \sin(2\pi(\boldsymbol{F}\mathcal{T} + \boldsymbol{\Phi})) + \boldsymbol{B} \tag{4}$$

The aforementioned periodic parameters are computed via a differentiable signal processing layer. By performing a real Fast Fourier Transform (FFT) on each channel of $\boldsymbol{z}$, we generate a zero-indexed matrix of Fourier coefficients $\boldsymbol{c} \in \mathbb{C}^{D \times (M+1)}$, $M = \lfloor \frac{H}{2} \rfloor$. From this, the power spectrum is calculated as:

$$\boldsymbol{p}^{i,j} = \frac{2}{H} |c^{i,j}|^2, \tag{5}$$

where i denotes the specific channel and j represents the index for frequency bands. Given the sample frequency f_s, the corresponding parameters are calculated as follows:

$$\boldsymbol{F}^i = \frac{\sum_{j=1}^{M} \boldsymbol{f}_j \cdot \boldsymbol{p}^{i,j}}{\sum_{j=1}^{M} \boldsymbol{p}^{i,j}}, \quad \boldsymbol{A}^i = \sqrt{\frac{2}{H} \sum_{j=1}^{M} \boldsymbol{p}^{i,j}}, \quad \boldsymbol{B}^i = \frac{\boldsymbol{c}^{i,0}}{H}, \tag{6}$$

where $\boldsymbol{f} = (0, f_s/H, \ldots, M f_s/H)$ denotes the frequency vector. To facilitate temporal alignment across different motion clips, a separate fully-connected layer is assigned to each latent channel. This layer predicts a 2D vector (s_x, s_y) at the central frame of $\mathcal{T}$ to determine the signed phase shift:

$$(s_x, s_y)^i = FC^i(\boldsymbol{z}^i), \qquad \boldsymbol{\Phi}^i = \frac{1}{2\pi} \text{atan2}(s_y, s_x)^i \tag{7}$$

Through this design, the network learns the temporal alignment of poses by predicting rotating vectors and adjusts periodic embeddings by assigning varying phases. By capturing the time-varying nature of the amplitude and frequency

in each channel, PAE is capable of encoding both periodic and non-periodic motions.

The periodic parameters capture the local periodicity of motion curves, facilitating the construction of a phase manifold $\mathcal{P} \in \mathbb{R}^{2D}$. Since the manifold state at frame t corresponds to the timestamp $\tau = \frac{\Delta t}{2}$, the phase is derived by shifting the learned central phase $\mathbf{\Phi}^i$ by the accumulated frequency component. Consequently, the coordinates on the manifold are computed as:

$$\begin{aligned} \boldsymbol{P}_t^{2i} = \boldsymbol{A}^i \cos(\boldsymbol{\alpha}_t^i), \quad \boldsymbol{P}_t^{2i+1} = \boldsymbol{A}^i \sin(\boldsymbol{\alpha}_t^i), \\ \boldsymbol{\alpha}_t^i = 2\pi \left(\boldsymbol{F}^i \cdot \frac{\Delta t}{2} + \mathbf{\Phi}^i \right) \end{aligned} \tag{8}$$

To eliminate the state ambiguity caused by identical scalar values at different phases and ensure a continuous state representation, we employ manifold coordinates $\boldsymbol{P}$ to measure motion diversity. During training, each latent phase channel focuses on specific local dynamics, acting as a band-pass filter that isolates essential motion characteristics across various frequency and amplitude ranges. Moreover, these learned representations inherently achieve superior spatio-temporal alignment. By capturing the dependencies between consecutive frames, they provide a more physically meaningful similarity metric than the raw state space. Consequently, evaluating the diversity reward within this manifold space instead of the raw robot states encourages the algorithm to discover more diverse gaits and explore a broader region of the state space, as demonstrated in Section IV.

B. Skill Discovery

Within the unsupervised RL framework, the objective is to learn a diverse set of behaviors parameterized by a latent factor K. This formulation yields a policy $\pi_\theta(\boldsymbol{a}_t|\boldsymbol{o}_t^p, k)$ that executes distinct behaviors given the policy observation $\boldsymbol{o}_t^p$ and a latent skill k drawn from a discrete space $\mathbb{K}$ of cardinality N_k.

Existing methodologies for unsupervised skill discovery typically maximize the mutual information between the latent skill K and the observation O^d [15], [16], [18]:

$$\begin{aligned} F(\theta) &= I(K, O^d) = H(K) - H(K|O^d) \\ &= \mathbb{E}_{(k,\boldsymbol{o}^d)\sim p(k,\boldsymbol{o}^d)} \left[\log p(k|\boldsymbol{o}^d) - \log p(k) \right] \end{aligned} \tag{9}$$

To maximize the skill entropy $H(K)$, the prior distribution $p(k)$ is fixed to be uniform. Since calculating the conditional distribution $P(K|O^d)$ is intractable, a learned parametric discriminator $q_\psi(k|\boldsymbol{o}^d)$ is applied to approximate this posterior, which is trained to infer skills from states. Replacing p with q yields a lower variational bound $\tilde{F}(\theta)$ on $F(\theta)$ [34]:

$$\begin{aligned} F(\theta) &\geq \tilde{F}(\theta) \\ &= \mathbb{E}_{(k,\boldsymbol{o}^d)\sim p(k,\boldsymbol{o}^d)} \left[\log q_\psi(k|\boldsymbol{o}^d) - \log p(k) \right] \end{aligned} \tag{10}$$

We maximize the lower bound $\tilde{F}(\theta)$ by adopting the following pseudo-reward:

$$r_t = \log q_\psi(k|\boldsymbol{o}_{t+1}^d) - \log p(k) \tag{11}$$

Algorithm 1 FLSD

Input: latent skill cardinality N_k, sequence length H
Output: Policy π_θ
Initialize networks, motion sequence $\boldsymbol{o}^H$, replay buffer D
for learning iterations = 1,2, … **do**
  sample latent variable $k \sim p_k$
  for time step = 0,1,2, … **do**
    sample action $\boldsymbol{a}_t \sim \pi_\theta(\boldsymbol{a}_t|\boldsymbol{o}_t^p, k)$ from skill
    step environment $\boldsymbol{s}_{t+1} \sim p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t)$
    extract $\boldsymbol{o}_{t+1}^p$, $\boldsymbol{o}_{t+1}^m$, $\boldsymbol{q}_{t+1}$ from $\boldsymbol{s}_{t+1}$
    update motion sequence $\boldsymbol{o}^H$ with $\boldsymbol{o}_{t+1}^m$
    calculate motion features $\boldsymbol{P}_{t+1} = PAE(\boldsymbol{o}_{t+1}^H)$
    update $\boldsymbol{o}_{t+1}^d = \begin{bmatrix} \boldsymbol{q}_{t+1} & \boldsymbol{P}_{t+1} \end{bmatrix}^T$
    compute reward r_t according to Eq. 11
    fill replay buffer with ($\boldsymbol{o}_{t+1}^p$, $\boldsymbol{o}_{t+1}^H$, $\boldsymbol{o}_{t+1}^d$, k)
  end for
  for learning epoch = 1,2, … **do**
    sample mini-batches $d \sim D$
    update V and π_θ with PPO
    update PAE according to Eq. 3
    update q_ψ according to Eq. 12
  end for
end for

To ensure a tight lower bound, the discriminator $q_\psi(K|O^d)$ must approximate the true distribution $p(K|O^d)$ by minimizing the negative log likelihood loss through supervised training:

$$L_D = -\mathbb{E}_{(k,\boldsymbol{o}^d)\sim p(k,\boldsymbol{o}^d)} \left[\log q_\psi(k|\boldsymbol{o}^d) \right] \tag{12}$$

The policy π_θ is optimized through reinforcement learning, and the discriminator q_ψ is updated through supervised learning, with both cooperating to facilitate skill discovery. The agent samples a skill $k \sim P(K)$ and generates observations $\boldsymbol{o}^d$. The discriminator subsequently uses these observations to decode the original skill k. When the observations of different skills do not overlap, the discriminator will easily learn to infer the skill, and the policy receives higher rewards, ultimately promoting the discovery of diverse skills.

We set the skill observations $\boldsymbol{o}_t^d$ to include motion features $\boldsymbol{P}_t$ and robot position $\boldsymbol{q}_t$ in the world coordinate system:

$$\boldsymbol{o}_t^d = \begin{bmatrix} \boldsymbol{q}_t & \boldsymbol{P}_t \end{bmatrix}^T. \tag{13}$$

The feature $\boldsymbol{P}_t$ enables the discriminator to better distinguish skills, and diversifying robot position $\boldsymbol{q}_t$ encourages the generation of locomotion behaviors.

C. Overview

We denote the full state provided by the environment as s, from which the policy observation, the motion observation, and the robot position $\boldsymbol{q}$ are extracted. The policy observation includes motion observation, previous action $\boldsymbol{a}_{t-1}$ and the one-hot vector k representing the skill index:

$$\boldsymbol{o}_t^p = \begin{bmatrix} \boldsymbol{o}_t^m & \boldsymbol{a}_{t-1} & k \end{bmatrix}^T. \tag{14}$$

The policy action $\boldsymbol{a}_t$ is defined as the target joint positions, which a PD controller uses to compute the torque commands with predetermined parameters ($K_p = 20, K_d = 0.5$).

Fig. 2 illustrates our proposed framework, and the algorithm is detailed in Algorithm 1. The joint optimization of the policy, discriminator, and PAE forms a collaborative framework. The agent samples a skill and strives to manifest a distinct behavior; PAE extracts the robot's motion trajectory features to provide a more suitable similarity metric for easier discrimination by the discriminator; the discriminator then attempts to infer the sampled skill and provides reward signals for the policy to diversify its skills.

IV. Experiments

A. Experimental Setup

Simulation experiments are conducted on a 12-DOF quadruped platform, the Unitree A1. To facilitate successful sim-to-real transfer, we employ domain randomization [35] by varying training conditions to account for real-world uncertainties. Specifically, we randomize ground friction, restitution, base and link masses, action delays, and motor torques, and introduce random velocity disturbances to simulate external force perturbations, thereby enhancing the robustness of the model.

Our FLSD framework is compared with the following baselines in learning diverse locomotion skills in simulation:

- DIAYN [15]: This method discovers diverse skills by maximizing the reward as shown in Eq. 11, with the discriminator observation consisting of motion observation and robot position.
- CASSI [13]: This method extracts diverse skills from unlabeled robot motion data, with the total reward function including diversity reward similar to Eq. 11, imitation reward and regulation terms.
- DOMiNiC [32]: This method leverages a CMDP formulation to make a trade-off between diversity and task performance, aiming to maximize diversity while satisfying task-relevant constraints. It first trains an expert policy by maximizing extrinsic rewards and then maximizes the intrinsic diversity reward, provided the extrinsic rewards remain above a certain threshold. The intrinsic reward is the minimum squared distance between feature expectations of different skills and the extrinsic reward includes task reward, style reward and regulation terms.
- CASSI w/o ER: This method removes the imitation reward and the regulation terms in CASSI, only maximizing the diversity reward.
- DOMiNiC w/o ER: This method removes the task reward and the regulation terms in DOMiNiC, only maximizing the diversity reward.

We introduce two metrics to compare our proposed method with the baselines. The first one is the state coverage metric, where a broader state space covered by the learned skills indicates a higher level of skill distinctiveness and, consequently, enables the robot to execute a more diverse array of tasks. The second one is the motion jitter, which is based on the power spectral density (PSD) analysis of joint velocities. While total energy consumption is task-dependent and unsuitable for directly assessing skill quality, the PSD's frequency-domain distribution effectively characterizes policy performance: concentrated low-frequency power density reflects the efficacy of planned motions, whereas suppressed high-frequency power density demonstrates smoothness in the joint actuation commands generated by the control policy.

Fig. 3. Motion sequences of various skills. Each row illustrates the motion sequence corresponding to a distinct skill, with the robot performing movements at a specific velocity and posture.

B. Skill Discovery

The typical skills discovered by our algorithm are shown in Fig. 3, including movements with different postures, varying linear and angular velocities and three-legged locomotion.

We set $H = 51$, $N_k = 50$ and train the policy with Proximal Policy Optimization (PPO, [36]) for 50000 iterations consisting of 120 simulation steps with 2000 parallel

TABLE I
State Space Coverage

	lin. vel. x (m/s)		lin. vel. y (m/s)		lin. vel. z (m/s)	
	max	min	max	min	max	min
FLSD	0.94	**-1.47**	**1.26**	**-1.41**	1.27	-1.10
DIAYN	1.35	-1.07	0.79	-0.88	0.92	-0.98
CASSI	2.11	-0.09	0.41	-0.36	**2.92**	**-2.83**
DOMiNiC	**2.24**	-0.72	1.15	-1.13	0.96	-1.09
CASSI w/o ER	0	0	0	0	0	0
	ang. vel. x (rad/s)		ang. vel. y (rad/s)		ang. vel. z (rad/s)	
	max	min	max	min	max	min
FLSD	**19.97**	**-22.53**	**8.02**	**-7.97**	**8.63**	**-9.55**
DIAYN	15.61	-15.62	7.72	-7.53	5.99	-6.51
CASSI	0.25	-0.31	0.87	-1.30	0.15	-0.11
CASSI w/o ER	0	0	0	0	0	0
	base height (m)		coverage rate (‰)			
	max	min				
FLSD	**0.40**	**0.09**	**60.00**			
DIAYN	0.31	0.10	25.70			
CASSI	0.34	0.10	11.41			
CASSI w/o ER	0.20	0.10	0.12			

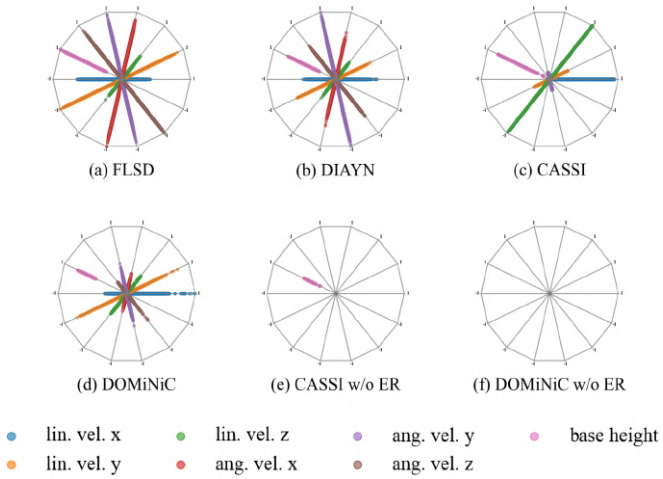

Fig. 4. State space coverage. We collected the linear velocity, angular velocity and base height of the trajectories executed by each skill. Then we normalized the data in each dimension and represented the data of each dimension with a specific color.

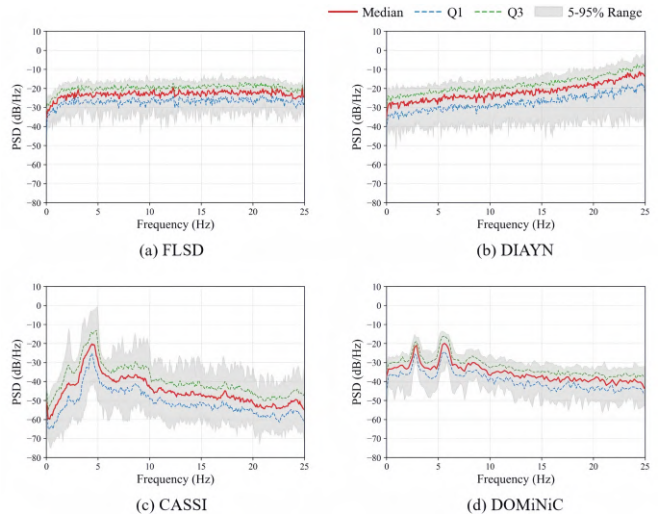

Fig. 5. Power Spectral Density Distribution of Angular Velocity. The red line indicates the median, the blue dashed line represents the lower quartile (Q1), the green dashed line represents the upper quartile (Q3), and the gray shaded area spans the 5th to 95th percentile range. The decibel (dB) scale is referenced to 1 $(\mathrm{rad/s})^2/\mathrm{Hz}$.

environments in Isaac Sim [37]. The control frequency of the policies is 50 Hz in simulation. For the baselines, we employ the same Unitree A1 simulation environment configuration and use the corresponding default parameter settings.

We evaluate the state coverage on the robot's linear velocity, angular velocity and base height. For each skill with a survival time exceeding 500 steps, a complete trajectory is executed to collect the visited states, which are then normalized in each dimension. The normalized state space is discretized with a step size of 0.5, and then we calculate the ratio of the discretized grid cells visited by the collected states. Fig. 4 and Table I show that our FLSD outperforms the baseline in state space coverage across almost all dimensions and improves the state space coverage by 133%, 426%, 680% respectively compared to DIAYN, CASSI and DOMiNiC. Both CASSI and DOMiNiC exhibit poor state coverage in the angular velocity dimension. CASSI w/o ER only generates skills that keep the robot static in different postures and DOMiNiC w/o ER does not generate skills with a survival time exceeding 500 steps. DIAYN demonstrates relatively broader state space coverage across most dimensions, yet it still underperforms compared to FLSD. Since DOMiNiC w/o ER failed to generate a single executable skill, its state space coverage result is not included in Table I.

We collect the joint angular velocity trajectories from various skills, with their power spectral density (PSD) computed through discrete Fourier transform (DFT). This yields frequency-specific PSD distributions across the spectrum. Fig. 5 displays a representative joint's PSD distribution, with other joints demonstrating comparable spectral characteristics. Table II presents the ratio of power in frequency components above 10 Hz to the total power across joint velocity trajectories generated by each algorithm. Compared to DIAYN, our FLSD reduces high-frequency power by 73%. However, there remains a significant gap in high-frequency suppression performance when compared to CASSI and DOMiNiC, which incorporate multiple regularization terms.

We employed t-SNE to project the robot states generated by different skills onto a two-dimensional plane. As shown in Fig. 6, some of the skills discovered by FLSD show similarities to those discovered by the baselines, while the styles of the skills identified by our method are more consistent and easier to distinguish, with the trajectories of each skill clustering together more tightly.

TABLE II
HIGH FREQUENCY POWER RATIO

	FLSD	DIAYN	CASSI	DOMiNiC
ratio (‰)	199.75±24.58	448.07±38.14	16.71±0.61	6.47±0.04

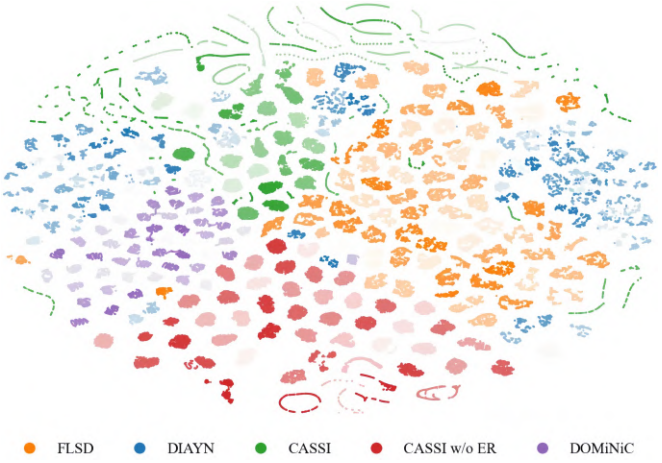

Fig. 6. T-SNE visualization for each skill. The same color indicates results generated by the same algorithm, while the shade of the color represents different skill trajectories.

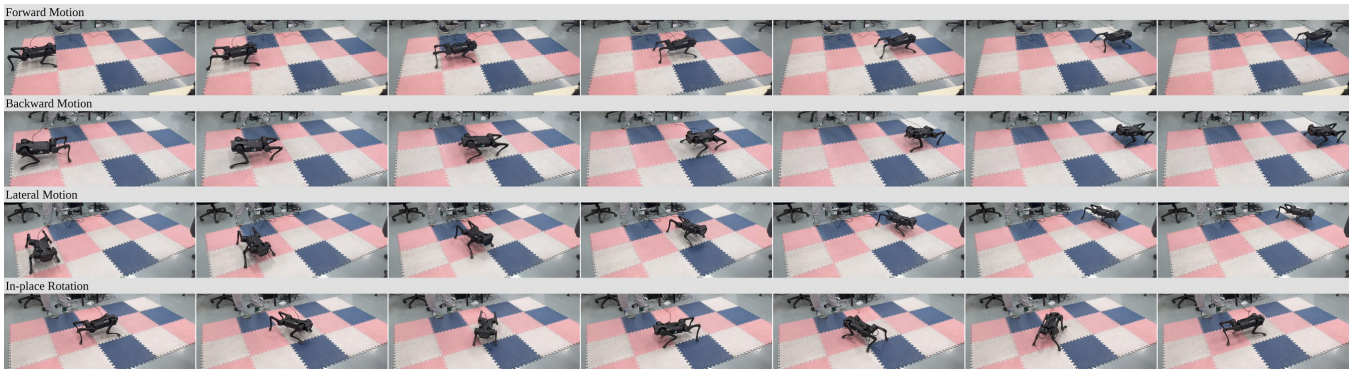

Fig. 7. Hardware demonstration of skill execution on the Unitree A1. The robot is instructed to track velocity commands.

C. Composition of Skills for Downstream Tasks

Our model does not incorporate task-specific reward functions during training. To accomplish downstream tasks, we adopt a hierarchical reinforcement learning framework: the trained model serves as the low-level controller, while a high-level controller is trained with task-specific reward functions. At each step, the high-level controller outputs a probability distribution over skills, from which the skill with the highest probability is selected and executed by the low-level controller. The low-level controller then generates the corresponding actions conditioned on the selected skill.

We adopt the locomotion task from [9] as the downstream task, where the quadruped robot is required to track velocity commands. The design of the reward function and the configuration of the training environment are consistent with [9]. We evaluate three training paradigms:

- FLSD-L: using the controller trained with FLSD as the low-level controller.
- DIAYN-L: using the controller trained with DIAYN as the low-level controller.
- Scratch: training the model from scratch.

Fig. 7 illustrates that, under the control of the high-level controller, our robot successfully tracks the given velocity commands. Fig. 8 shows that FLSD-L achieves a faster convergence rate, indicating that the policy trained with our method can effectively serve as a low-level controller for executing downstream tasks.

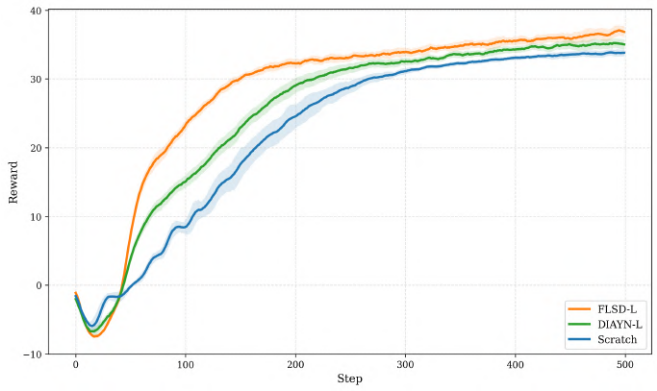

Fig. 8. Learning curves. The results show that FLSD-L achieves the fastest convergence.

V. Conclusion

In this work, we propose FLSD, a novel framework that leverages spatio-temporal state representations to measure skill diversity while implicitly modeling the temporal phase coherence of states within trajectories. The results show that our FLSD enhances the smoothness of state transitions and achieves broader state space coverage compared to the baselines. Our algorithm employs a simple mutual information-based reward function, enabling a unified policy to execute diverse state transition patterns. Through a hierarchical reinforcement learning framework, our model can serve as a low-level controller to execute downstream tasks.

In future work, we aim to advance skill discovery quality while limiting the reliance on expert knowledge, enabling algorithms to autonomously discover richer and more dynamic skill repertoires through self-guided exploration.

References

[1] D. Kim, H. Kwon, J. Kim, G. Lee, and S. Oh, "Stage-wise reward shaping for acrobatic robots: A constrained multi-objective reinforcement learning approach," *arXiv preprint arXiv:2409.15755*, 2024.

[2] R. Yang, Z. Chen, J. Ma, C. Zheng, Y. Chen, Q. Nguyen, and X. Wang, "Generalized animal imitator: Agile locomotion with versatile motion prior," *arXiv preprint arXiv:2310.01408*, 2023.

[3] A. Klipfel, N. Sontakke, R. Liu, and S. Ha, "Learning a single policy for diverse behaviors on a quadrupedal robot using scalable motion imitation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2768–2775.

[4] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," *arXiv preprint arXiv:2401.17583*, 2024.

[5] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024.

[6] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 443–11 450.

[7] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," *arXiv preprint arXiv:2309.05665*, 2023.

[8] S. Luo, S. Li, R. Yu, Z. Wang, J. Wu, and Q. Zhu, "Pie: Parkour with implicit-explicit learning framework for legged robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9986–9993, 2024.

[9] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[10] L. Han, Q. Zhu, J. Sheng, C. Zhang, T. Li, Y. Zhang, H. Zhang, Y. Liu, C. Zhou, R. Zhao *et al.*, "Lifelike agility and play in quadrupedal robots using reinforcement learning and generative pre-trained models," *Nature Machine Intelligence*, vol. 6, no. 7, pp. 787–798, 2024.

[11] C. Zhang, J. Sheng, T. Li, H. Zhang, C. Zhou, Q. Zhu, R. Zhao, Y. Zhang, and L. Han, "Learning highly dynamic behaviors for quadrupedal robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 9183–9189.

[12] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, and G. Martius, "Learning agile skills via adversarial imitation of rough partial demonstrations," in *Conference on Robot Learning*. PMLR, 2023, pp. 342–352.

[13] C. Li, S. Blaes, P. Kolev, M. Vlastelica, J. Frey, and G. Martius, "Versatile skill control via self-supervised adversarial imitation of unlabeled mixed motions," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2944–2950.

[14] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel, "Adversarial motion priors make good substitutes for complex reward functions," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 25–32.

[15] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," *arXiv preprint arXiv:1802.06070*, 2018.

[16] Z. Jiang, J. Gao, and J. Chen, "Unsupervised skill discovery via recurrent skill training," *Advances in Neural Information Processing Systems*, vol. 35, pp. 39 034–39 046, 2022.

[17] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion; state of the art and future perspectives," *arXiv preprint arXiv:2406.01152*, 2024.

[18] 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.

[19] D. Strouse, K. Baumli, D. Warde-Farley, V. Mnih, and S. Hansen, "Learning more skills through optimistic exploration," *arXiv preprint arXiv:2107.14226*, 2021.

[20] H. Kim, B. K. Lee, H. Lee, D. Hwang, S. Park, K. Min, and J. Choo, "Learning to discover skills through guidance," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[21] S. Starke, I. Mason, and T. Komura, "Deepphase: Periodic autoencoders for learning motion phase manifolds," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–13, 2022.

[22] C. Li, E. Stanger-Jones, S. Heim, and S. Kim, "Fld: Fourier latent dynamics for structured motion representation and learning," *arXiv preprint arXiv:2402.13820*, 2024.

[23] Y. Cai, Y. Wang, Y. Zhu, T.-J. Cham, J. Cai, J. Yuan, J. Liu, C. Zheng, S. Yan, H. Ding *et al.*, "A unified 3d human motion synthesis model via conditional variational auto-encoder," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 645–11 655.

[24] R. Briq, C. Zou, L. Pishchulin, C. Broaddus, and J. Gall, "Recurrent transformer variational autoencoders for multi-action motion synthesis," *arXiv preprint arXiv:2206.06741*, 2022.

[25] Q. Men, H. P. Shum, E. S. Ho, and H. Leung, "Gan-based reactive motion synthesis with class-aware discriminators for human–human interaction," *Computers & Graphics*, vol. 102, pp. 634–645, 2022.

[26] L. Mourot, F. Le Clerc, C. Thébault, and P. Hellier, "Jumps: Joints upsampling method for pose sequences," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 1096–1103.

[27] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–13, 2017.

[28] S. Starke, H. Zhang, T. Komura, and J. Saito, "Neural state machine for character-scene interactions," *ACM Transactions on Graphics*, vol. 38, no. 6, p. 178, 2019.

[29] S. Starke, Y. Zhao, T. Komura, and K. Zaman, "Local motion phases for learning multi-contact character movements," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 54–1, 2020.

[30] 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.

[31] A. Sharma, M. Ahn, S. Levine, V. Kumar, K. Hausman, and S. Gu, "Emergent real-world robotic skills via unsupervised off-policy reinforcement learning," *arXiv preprint arXiv:2004.12974*, 2020.

[32] J. Cheng, M. Vlastelica, P. Kolev, C. Li, and G. Martius, "Learning diverse skills for local navigation under multi-constraint optimality," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5083–5089.

[33] D. Holden, J. Saito, T. Komura, and T. Joyce, "Learning motion manifolds with convolutional autoencoders," in *SIGGRAPH Asia 2015 technical briefs*, 2015, pp. 1–4.

[34] D. Barber and F. Agakov, "The im algorithm: a variational approach to information maximization," *Advances in neural information processing systems*, vol. 16, no. 320, p. 201, 2004.

[35] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[37] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.