Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

Advancing Humanoid Locomotion: Mastering Challenging Terrains with Denoising World Model Learning

Xinyang Gu*
RobotEra TECHNOLOGY CO., LTD.
*Equal contribution.

Yen-Jen Wang*
Tsinghua University
Shanghai Qi Zhi Institute
Email: wangyenjen@berkeley.edu
*Equal contribution.

Xiang Zhu*
Tsinghua University
Shanghai Qi Zhi Institute
*Equal contribution.

Chengming Shi*
Tsinghua University
*Equal contribution.

Yanjiang Guo
Tsinghua University
Shanghai Qi Zhi Institute

Yichen Liu
Tsinghua University

Jianyu Chen
Tsinghua University
Shanghai Qi Zhi Institute
RobotEra TECHNOLOGY CO., LTD.
Email: jianyuchen@tsinghua.edu.cn

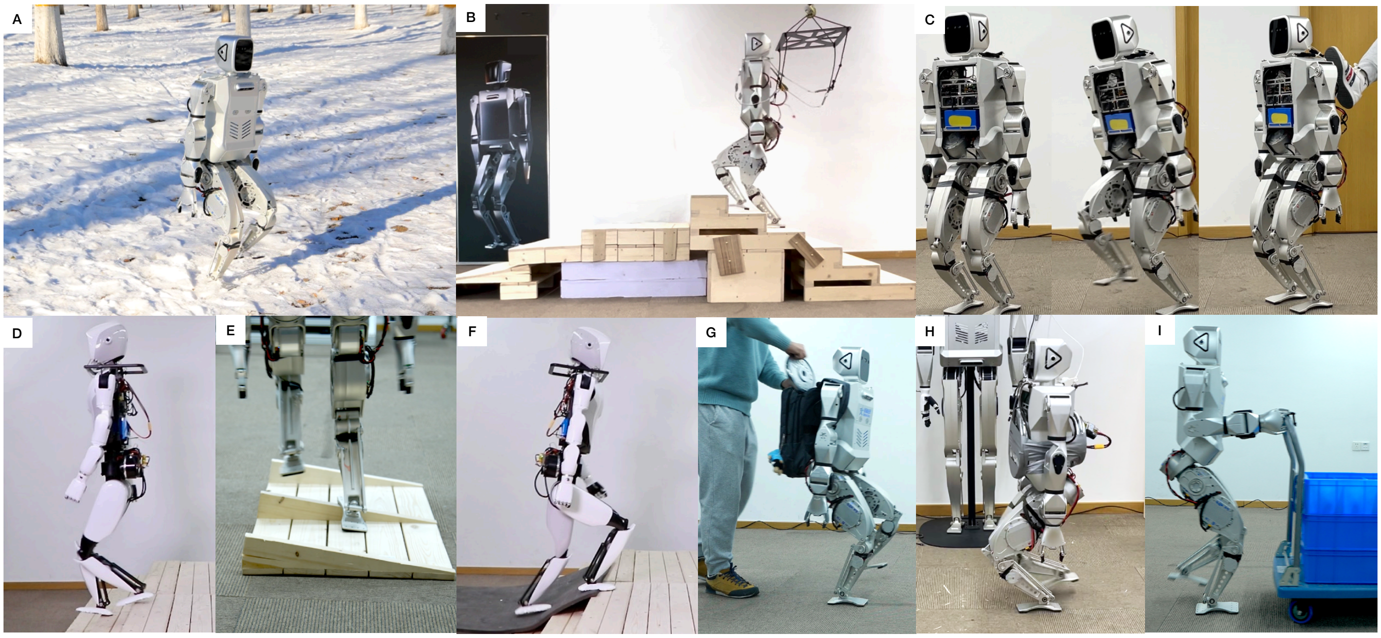

Fig. 1: Extensive showcase of locomotion skills using the proposed framework. Displayed is a sequence illustrating a humanoid robot skillfully executing various locomotion tasks in real world challenging environments.

Abstract—**Humanoid robots, with their human-like skeletal structure, are especially suited for tasks in human-centric environments. However, this structure is accompanied by additional challenges in locomotion controller design, especially in complex real-world environments. As a result, existing humanoid robots are limited to relatively simple terrains, either with model-based control or model-free reinforcement learning. In this work, we introduce Denoising World Model Learning (DWL), an end-to-end reinforcement learning framework for humanoid locomotion control, which demonstrates the world's first humanoid robot to master real-world challenging terrains such as snowy and inclined land in the wild, up and down stairs, and extremely uneven terrains. All scenarios run the same learned neural network with zero-shot sim-to-real transfer, indicating the superior robustness and generalization capability of the proposed method.**

I. Introduction

Modern environments are primarily designed for humans. Therefore, humanoid robots, with their human-like skeletal structure, are especially suited for tasks in human-centric environments and offer unique advantages over other types of robots. Their mobility is crucial for completing diverse tasks in the real world, underlining the necessity of their capacity to walk on complex terrains.

Previously, model-based control techniques such as Zero Moment Point (ZMP) and Model Predictive Control (MPC) combined with Whole-Body Control (WBC) have significantly advanced humanoid robots' locomotion abilities, enabling skills like walking, jumping, and even backflipping[2, 38, 5]. However, the success of these methods depends on accurately modeling the environment's dynamics, which can make it difficult to handle complex interactions with the environment, such as navigating challenging terrains.

Reinforcement learning (RL), on the other hand, relies less on exact environmental modeling. Recent progress in model-free RL has shown great potential, particularly in developing adaptive legged locomotion controllers [28]. This allows robots to learn and adapt to a wide range of situations, often surpassing the capabilities of traditional model-based control methods[22].

However, ensuring robustness in humanoid robots, as opposed to quadrupedal[28] and bipedal[27] counterparts, involves addressing several additional challenges. These include but are not limited to a higher center of gravity, instability during leg swinging, greater leg inertia, extra weight from the torso and arms, and their larger size overall. Therefore, to date, real-world applications of reinforcement learning (RL) for controlling humanoid robots, as demonstrated in the recent study[26], have been limited to relatively simple terrains.

In this work, we introduce **Denoising World Model Learning (DWL)** for controlling humanoid robots across varied and complex terrains. To the best of our knowledge, DWL enables the world's first humanoid robot to master real-world challenging terrains with end-to-end RL and zero-shot sim-to-real transfer. As shown in Fig.1, our humanoid robot is able to navigate stably through snowy inclined land in the wild, stairs, irregular surfaces, etc., and can resist large external disturbances. All scenarios run the same learned neural network policy, indicating its robustness and generalization. The key ingredient of DWL lies in establishing an effective representation learning framework to denoise the factors enlarging the sim-to-real gap. Furthermore, we are the first to enable active 2-DoF ankle control with a Closed Kinematic Chain Ankle Mechanism (shown in Fig.2) for humanoid robot locomotion learning. Unlike previous studies [32] with only one DoF ankle control or passive ankle control [26], our approach enables the robot to become extremely robust. The contributions of our work are summarized as follows:

1) Demonstrate the world's first humanoid robot mastering real-world challenging terrains with end-to-end RL through zero-shot sim-to-real transfer.
2) Propose DWL, a novel RL framework to bridge the sim-to-real gap and achieve robust generalizable performance.
3) Demonstrate the first humanoid robot using active 2-Dof ankle control with a Closed Kinematic Chain Ankle Mechanism with RL, which substantially enhances the stability and flexibility of the robots.

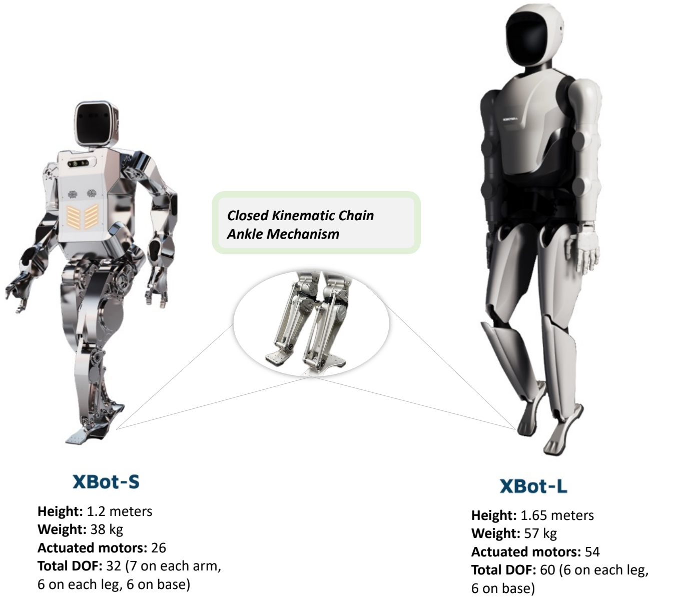
Closed Kinematic Chain Ankle Mechanism
XBot-S
Height: 1.2 meters
Weight: 38 kg
Actuated motors: 26
Total DOF: 32 (7 on each arm, 6 on each leg, 6 on base)
XBot-L
Height: 1.65 meters
Weight: 57 kg
Actuated motors: 54
Total DOF: 60 (6 on each leg, 6 on base)

Fig. 2: Illustration of the humanoid robot's hardware structure and the Closed Kinematic Chain Ankle Mechanism. This mechanism is notable for offering two degrees of freedom in each ankle while reducing leg inertia. Our works are tested on two distinct sizes of humanoid robots, XBot-S and XBot-L, provided by Robot Era.

II. Related Works

a) Learning Robot Locomotion: Reinforcement learning has become more promising to enable robots to perform stable locomotion[35, 12, 18]. Compared to previous RL efforts with quadrupedal robots [28] and bipedal robots like Cassie[21, 17], our focus on humanoid robots presents a significantly more challenging setup. Our proposed method excels in automating state representation learning [19], mastering end-to-end learning for both prediction and adaptation and facilitating a seamless zero-shot transfer to real-world scenarios by effectively bridging the sim-to-real gap.

Furthermore, conventional approaches, often encompassing multi-stage training processes [1], detailed reward designs [39], or behavior cloning [24], typically falter amidst the dynamic and unpredictable real-world scenarios. On the other hand, DWL instead integrates a world model within an encoder-decoder framework, employing a masking loss to predict the state from observations.

b) Humanoid Robot Locomotion Control: The evolution of humanoid locomotion began with early concepts and basic models, exemplified by WABOT-1 in the 1970s[13]. Progress in sensors and control algorithms enhanced humanoid robots' stability and adaptability. model-based control techniques[16] like ZMP[34], MPC[33, 20], and WBC[30] have significantly improved locomotive capabilities. Learning-based approaches, which are less reliant on precise dynamic models, offer better adaptability and robustness. Despite this, real-world applications of reinforcement learning for humanoid control, such as [26, 8], have been successful but limited to simpler terrains.

III. Problem Setting

A. Reinforcement Learning Background

Our approach utilizes a reinforcement learning problem setting, encapsulated in the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, \mathcal{O}, R, \gamma \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, with the transition dynamics $T(\mathbf{s}'|\mathbf{s}, \mathbf{a})$, the reward function $R(\mathbf{s}, \mathbf{a})$, and the discount factor $\gamma \in [0, 1]$. $\mathcal{O}$ represents the observation space.

Our framework distinctly adapts to both simulated and real-world environments. In the simulation, the agent is afforded complete visibility of state $\mathbf{s} \in \mathcal{S}$. On the other hand, the real world is plagued by partial observability. The agent only has access to partially observations $\mathbf{o} \in \mathcal{O}$, which provide incomplete information about the state due to sensory limitations and environmental noise. The policy $\pi(\mathbf{a}|\mathbf{o}_{\leq t})$ maps the historical observations to a distribution over actions. As a result, the agent operates within a discrete-time Partially Observable Markov Decision Process (POMDP), necessitating decision-making based on sporadic and partial data. The primary goal is to optimize this policy π to maximize the expected total return $J = \mathbb{E}[R_t] = \mathbb{E}\left[\sum_t \gamma^t r_t\right]$.

B. Humanoid Robot Hardware

We use two different sizes of humanoid robots for our experiments, as illustrated in Fig. 2. XBot-S weighs 38 kg and stands 1.2 meters tall. The robot is equipped with 26 actuated motors: 7 in each arm and 6 in each leg. XBot-L weighs 57 kg and stands 1.65 meters tall. The robot is equipped with 54 actuated motors. For the purposes of this study, we focus on leg control, keeping the arm motors stationary. Each leg is powered by 6 motors: the yaw and roll joint motors with a peak torque of $100\mathrm{N} \cdot \mathrm{m}$, the pitch and knee joint motors with $250\mathrm{N} \cdot \mathrm{m}$, and 2 ankle motors each providing $36\mathrm{N} \cdot \mathrm{m}$ of torque ($50\mathrm{N} \cdot \mathrm{m}$ in XBot-L). The ankle motors, situated near the knee, are operated via a *Closed Kinematic Chain Ankle Mechanism* as Fig. 2. This design aims to reduce leg inertia while ensuring an adequate degree of freedom.

IV. Methods

A. Denoising World Model Learning

Utilizing RL, various skills can be learned in simulation, but the transition to real-world robots faces significant challenges due to the sim-to-real gap, which is mainly caused by inaccurate simulation of the robot hardware and the limited information provided by onboard sensors. To overcome this barrier, we introduce **Denoising World Model Learning (DWL)**, which enables online adaptation and state estimation through representation learning. DWL is characterized by two primary features:

- An encoder-decoder architecture for world model learning, effectively embedding partially observed historical raw sensory data into a latent space and reconstructing the robot's full state from it.
- A policy gradient method that facilitates iterative improvements of the controller, allowing optimizing complex objectives through environmental interaction.

1) Encoder-Decoder Architecture of DWL: A perfect simulator with a fully observable state and accurate sensors would eliminate the sim-to-real gap. However, real-world scenarios only provide us with noisy, partially observed sensor data. The sim-to-real gap can be regarded as adding the following types of **noises** to the true state:

- Environmental noise: Real-world environments are complex and unpredictable, presenting challenges such as navigating on challenging terrains or unexpected external forces applied to the robot.
- Dynamics noise: Accurately simulating the true dynamics of the physical world is unfeasible, leading to oversimplifications in simulations, like approximations of ground friction or object deformability.
- Sensory noise: Physical sensors inherently contain measurement noise, for example, IMU drift and inaccuracies in joint position readings.
- Masking noise: Some information may be unobtainable in reality due to the absence of specific sensors on the robot, such as linear velocity and contact force measurements. This partial observability can be regarded as adding masking noise[6, 10, 11].

To mitigate these noises, we have developed a framework that firstly simulates noisy observations within the simulation and subsequently employs an encoder-decoder architecture to denoise these observations and accurately recover the true state and dynamics, as depicted in Fig. 3.

Additionally, to mimic the constraints of partial observability, we mask out some information that is not observable on real robots. We emulate the environment, dynamics, and sensor noises utilizing domain randomization (DR) methods[37]. In this approach, we introduce random perturbations to the actual state and dynamics, such as angular velocity and PD parameters. This procedure aligns with an observation model[9] expressed as $o_t \sim P_{\text{Noise}}(o_t|s_t)$. We elaborate on this in Section IV-D.

An encoder-decoder architecture is designed to denoise the observations. The recurrent encoder extracts latent state z_t from the robot's historical noisy sensor observations. This latent representation is the core of state estimation, providing a rich, condensed summary of the robot's situational awareness. Subsequently, the decoder endeavors to reconstruct the robot's true state from this latent state. The formal expression of this model is given by:

$$P(\tilde{\mathbf{s}}_t) = \mathrm{E}_{o_{\leq t}}\left[\int_z P_{\text{Decoder}}(\tilde{\mathbf{s}}_t|z_t) \cdot P_{\text{Encoder}}(z_t|o_{\leq t})\right] \qquad (1)$$

where $P(\tilde{\mathbf{s}}_t)$ represents the estimation of the real state distribution $P(\mathbf{s}_t)$ at time t. The encoder captures the conditional distribution P_{Encoder} of these latent variables given the noisy historical observations $o_{\leq t}$, and the decoder P_{Decoder} reconstructs the state from the latent representation z_t.

It is imperative to recognize that the dimension of $o_{\leq t}$ is much larger than that of z_t, implying z_t an effective information bottleneck [36]. This allows DWL to prioritize the salient aspects of sensory input. Furthermore, to enhance the efficiency and robustness of state estimation, sparsity within the latent representation is sought [4]. This is achieved by introducing an L1 regularization term in the latent domain. Moreover, since there is no need to generate new data from the latent space, and it is in a pure denoising process, a deterministic loss could be adopted instead of a variational loss [15]. The denoising loss is thus expressed as follows:

$$\mathcal{L}_{\text{denoise}} = \|\tilde{\mathbf{s}}_t - \mathbf{s}_t\|_2 + \lambda_r \|\boldsymbol{z}_t\|_1 \tag{2}$$

where λ_r represents the regularization coefficient. And s_t is the full state. By incorporating privileged information into the state, such as the ground's friction coefficients, the actuator's torque values, and terrain height scans, enables the agent to effectively conduct online adaptation and system identification.

2) Policy Learning in DWL: Within the DWL framework, an Asymmetric Actor-Critic architecture is employed, drawing upon the concept of privilege learning as elucidated in previous works[3] [25]. This architecture is instrumental in enhancing data utilization during the training phase, proving particularly beneficial in real-world scenarios where direct state information is inaccessible. The actor component of the model computes its loss via the Proximal Policy Optimization (PPO)[29], which is articulated as:

$$\begin{aligned}\mathcal{L}_\pi = \min \Bigg[& \frac{\pi(a_t \mid o_{\leq t})}{\pi_b(a_t \mid o_{\leq t})} A^{\pi_b}(o_{\leq t}, a_t), \\ & \text{clip}\left(\frac{\pi(a_t \mid o_{\leq t})}{\pi_b(a_t \mid o_{\leq t})}, c_1, c_2\right) A^{\pi_b}(o_{\leq t}, a_t)\Bigg]\end{aligned} \tag{3}$$

where π denotes the target policy to be optimized, π_{b} is the behavior policy employed for data sampling, and c_1, c_2 represents the PPO clipping range. In the context of DWL's encoder-decoder structure, the actor policy is defined as $\pi(a_t \mid P_{\text{Encoder}}(z_t \mid o_{\leq t}))$. On the other hand, the critic could use state information for calculations of the value function. Thus, the critic loss is given by the following formula:

$$\mathcal{L}_v = \|R_t - V(s_t)\|_2, \tag{4}$$

In this formulation, R_t denotes the cumulative return at time t, and $V(s_t)$ is the value function as determined by the critic at state s_t. The integration of privileged information within the state representation arms the learning agent with the capacity to make informed decisions. This approach aligns seamlessly with the DWL framework. It obviates the need for design privilege information by employing a unified state definition for both state estimation and value function assessment.

3) Formulating the DWL Loss Function: The DWL framework consolidates its learning objectives through a composite loss function that integrates the aspects of state reconstruction and policy optimization. The total loss function is a weighted sum of the denoising loss Equation(2), the policy loss Equation(3), and the value loss Equation(4), formally expressed as:

$$\mathcal{L}_{\text{DWL}} = \mathcal{L}_{\text{denoise}} + \lambda_\pi \mathcal{L}_\pi + \lambda_v \mathcal{L}_v, \tag{5}$$

where λ_π and λ_v are the weighting factors for the policy and value loss components, respectively. This approach enables DWL to fine-tune the learning process, ensuring precise state estimation and informed decision-making based on these estimates. Reconstructing the state from masking loss and domain randomization noise, DWL demonstrates robustness and adaptability in complex real-world scenarios, as elaborated in Section V.

In conclusion, the integration of masking noise and domain randomization noise cultivates a robust latent space for end-to-end state representation learning. When allied with policy gradient loss, this strategy propels a comprehensive approach to state estimation and policy optimization. This refined system is adept at narrowing the sim-to-real gap, effectively translating simulation-trained models to real-world applications.

B. Reward Formulation

Our reward function guides the robot to follow velocity commands, maintain a stable gait, and ensure gentle contact, thereby enabling robust locomotion across challenging terrains and above-ground obstacles.

1) Composition of Rewards: The reward function is structured into four key components: (1) velocity tracking, (2) periodic reward, (3) feet trajectory tracking, and (4) regularization terms. Our approach, drawing inspiration from previous works [31, 26, 32], employs a periodic reward to facilitate natural gait learning. Furthermore, we introduce a tracking loss defined as $\phi(e, w) := \exp\left(-w \cdot \|e\|^2\right)$, where e represents the tracking error and w the error tolerance strength. Detailed calculations can be found in the appendix, Section B.

A novel aspect of our reward design addresses the sparse nature of contact force feedback. Rather than relying exclusively on contact force, our system focuses on foot velocity tracking. This is achieved by designing foot trajectories that incorporate predetermined velocities upon ground contact, thus ensuring a consistent and robust reward signal at each step. Such a strategy promotes gentle ground contacts, leading to reduced impact forces and enhancing the effectiveness of the sim-to-real transfer.

2) Quintic Polynomial Foot Trajectory Interpolation: In our approach, we focus on refining the locomotion of humanoid robots through the strategic design of foot trajectories. Quintic polynomial interpolation is utilized to determine these trajectories, a method that is particularly effective in meeting the precise kinematic requirements of a humanoid robot's gait cycle. This technique not only facilitates smoother motion but also ensures accurate foot placement, a crucial factor for maintaining stability and efficiency in humanoid walking patterns.

Quintic polynomial interpolation offers an advantageous approach in robotic motion planning due to its ability to provide smooth trajectories and precise control over velocity and acceleration. The general form of a quintic polynomial is given by:

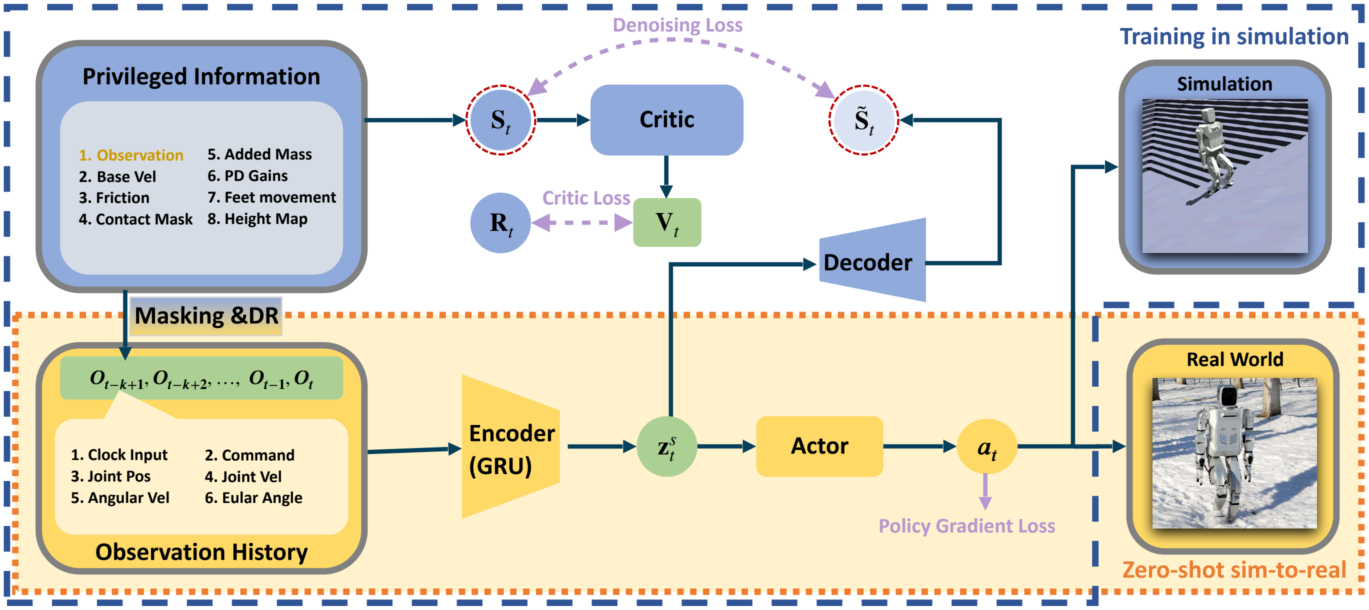

Fig. 3: Illustration of the Denoising World Model Learning Framework. This diagram details the information flow from sensory input to action output in both simulated and real-world settings. Raw observations are generated by adding masking and DR noise to privileged observations. This is then encoded into a latent state and decoded to reconstruct the true state via a denoising process.

$$f(t) = \sum_{k \leq 5} a_k t^k \tag{6}$$

Let t denote the time variable, and $a_0, a_1, \ldots, a_5$ be the coefficients that need to be determined. We denote the swing time by T. In our periodic reward design, one leg being in the swing phase implies the other is in the stance phase. One swing phase and one stance phase together complete a full gait cycle. The trajectory of the robot's foot during the swing phase is defined by $f(t)$, which is shaped through a series of kinematic constraints at critical moments in the robot's gait. These constraints are:

1) Initial foot height at $t = 0$, given by $f(0) = h_0$, where h_0 is the initial height.
2) Initial foot velocity at $t = 0$, determined by $f'(0) = v_0$, with v_0 being the initial velocity.
3) Initial foot acceleration, represented by $f''(0) = \text{acc}_0$, where acc_0 is the initial acceleration.
4) Reaching maximum foot height at the midpoint of the swing phase, $f(T/2) = h_{\text{max}}$, where h_{max} is the target feet height.
5) Final foot height at the end of the swing phase, $f(T) = h_{\text{swing}}$, with h_{swing} as the final height.
6) Final foot velocity at the end of the swing phase, $f'(T) = v_{\text{swing}}$, where v_{swing} is the final velocity.

To deduce the coefficients $a_{0\ldots5}$, a numerical optimization technique is employed. Once the coefficients are ascertained, they succinctly characterize the foot's vertical trajectory (i.e., the swing height). Quintic polynomial interpolation is instrumental in ensuring soft landings within humanoid robotic locomotion, offering granular control over the robot's swing height, foot acceleration, and velocity profiles. One optimization result and the corresponding trajectory plot are shown in Appendix Table IV and Fig. 7.

This method facilitates the manipulation of higher-order derivatives to attenuate impact forces at footfall. By adjusting the coefficients of the quintic polynomial, trajectories are crafted that not only elevate the foot to surmount obstacles but also maintain smooth movements and mitigate the impact upon contact. These fluent transitions enable a gentler touchdown, enhancing the robot's stability and advancing the creation of efficient, adaptable robots capable of safely traversing diverse terrains.

C. *Configuration of DWL training process*

In our DWL framework, as illustrated in Fig. 3, we utilize a Gated Recurrent Unit (GRU)[7] for the encoding process and a two-layer Multilayer Perceptron for both the decoding and actor networks. Details of the training configuration can be found in the appendix section C.

The robot's base pose is denoted by P^b, and the pose of the feet is denoted by P^f. The pose, which includes both position and orientation, is represented as a six-dimensional vector $[x, y, z, \alpha, \beta, \gamma]$. Here, x, y, z specifies the position, and α, β, γ represents the orientation in Euler angles. The policy network inputs include proprioceptive sensor data and a periodic clock signal, represented as $(\sin(t), \cos(t))$, in addition to command inputs defining the desired velocities $\dot{P}_{x,y,\gamma}$. These observations are detailed in Table I. The state includes privileged observations, which are typically unavailable to standard proprioceptive sensors on physical robots. This state also integrates the current step's reward, combining a reward model with the world model, which is expected to enhance the encoder's ability to capture the environmental context in the latent space.

Other important components of the state are the Periodic Stance Mask $I(t)$, a binary indicator of expected foot contact patterns for a periodic gait, and the Cycle Time, essential for computing foot trajectories as outlined in (6). The Feet Movement, indicating both the position P^f_{xyz} and velocity of the feet $\dot{P}^f_{xyz}$. Also, the height scan provides an approximate height map to further enhance the estimation of the state. Please note that the input to our policy includes only proprioceptive sensor data and does not incorporate any LiDAR or depth camera information. The height scans listed in Table I are privileged observations employed by the Critic during training.

TABLE I: Summary of Observation Space. The table categorizes the components of the observation space into observation and state. The table also details their dimensions.

Components	Dims	Observation	State
Clock Input	2	✓	✓
Commands	3	✓	✓
Joint Position	12	✓	✓
Joint Velocity	12	✓	✓
Angular Velocity	3	✓	✓
Orientation	3	✓	✓
Last Actions	12	✓	✓
Base Linear Velocity	3		✓
Frictions	1		✓
Push Force&Torques	6		✓
Cycle Time	1		✓
Periodic Stance Mask	2		✓
Feet movement	12		✓
Feet Contact	2		✓
Body Mass	1		✓
Current Reward	1		✓
Torques	12		✓
Height Scan	96		✓

TABLE II: Overview of Domain Randomization. Presented are the domain randomization terms and the associated parameter ranges. Additive randomization increments the parameter by a value within the specified range while scaling randomization adjusts it by a multiplicative factor from the same range.

Parameter	Unit	Range	Operator
Joint Position	rad	[-0.3, 0.3]	additive
Joint Velocity	rad/s	[-1, 1]	additive
Angular Velocity	rad/s	[-0.1, 0.1]	additive
Orientation	rad	[-0.1, 0.1]	additive
System Delay	ms	[0, 10]	-
Friction	-	[0.2, 2.0]	-
Motor Offset	rad	[-0.05, 0.05]	additive
Motor Strength	%	[90, 110]	scaling
Payload	kg	[-5, 20]	additive
PD Factors	%	[80, 120]	scaling

Each action $a_t \in \mathbb{R}^{12}$ determines the actuators' target positions, followed by a Proportional-Derivative controller to translate into joint torques. Our control policy functions at 100Hz, surpassing the usual rates in RL-based locomotion strategies (50Hz), thus providing finer granularity and enhanced precision in the robot's movements. The internal PD controller operates at a higher frequency of 500Hz.

For our simulations, we use the Isaac Gym environment [23]. However, its lack of support for the closed kinematic chain employed in our ankle control necessitates the addition of two virtual motors within the simulator. We then remap the joint targets to the actual motors for deployment. The policy optimization employs the DWL loss function (refer to Equation 5) with Adam optimizer[14]. This method capitalizes on the inherent advantages of the DWL framework to refine a robust locomotion policy, and Hyper-parameters can be found in Appendix TABLE VIII. Remarkably, the resulting policy is ready for direct application to the physical robot without the need for further adjustments, exemplifying a seamless zero-shot transfer from simulation to real-world deployment.

D. Domain Randomization

To bridge the gap between simulation and reality, our methodology emphasizes extensive domain randomization of crucial dynamics parameters. This addresses the main sources of real-world variability: environment noise, dynamics noise, and sensory noise.

Randomization covers environmental elements such as floor friction, orientation, and robot-specific aspects like mass and Center of Mass positioning. Variations in motor parameters, including PD controller settings, are introduced to acclimate the policy to a range of motor behaviors.

Furthermore, we incorporate system latencies and inject random deviations in the robot's Center of Mass, equipping the policy to handle unforeseen disturbances in real environments. This thorough randomization strategy is essential for ensuring the policy's resilience and flexibility in actual deployment scenarios. Further specifics are presented in Table II.

V. Experiments

In this section, we mainly focus on the performance of challenging settings in both indoor and outdoor environments. The benchmark comparisons discussed below were all conducted using the smaller humanoid robot, which stands 1.2 meters tall. Additionally, we deployed our algorithm on a larger humanoid robot, measuring 1.65 meters in height, as detailed in Fig. 6.

A. Benchmark Comparison

For the empirical assessment of our approach, we conduct a series of experiments using both the XBot-S and XBot-L, applying the learned policy in a zero-shot transfer to real-world settings. This deployment encompasses a range of intricate and challenging terrains, testing the limits of the locomotive capabilities of our robot. To the best of our knowledge, this represents the world first humanoid robot to robustly navigate such complex environments using end-to-end reinforcement learning.

Our evaluation framework includes comparisons with two baseline methodologies, providing a comprehensive perspective on the effectiveness and advancements offered by our approach. In summary, we run three kinds of algorithms:

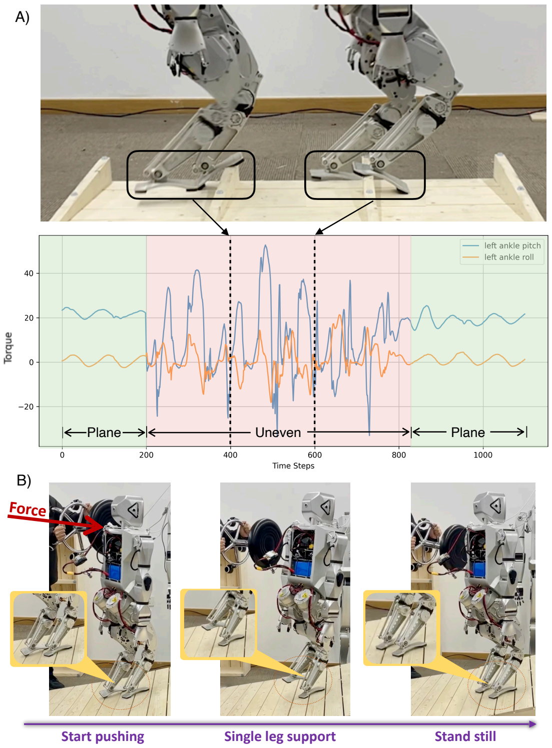

Fig. 4: Dynamic adaptation of the ankle control mechanism. A) The top image demonstrates the humanoid robot's ankle control system actively maintaining balance on uneven terrain. The associated torque plot reveals the control system's adjustments during steady locomotion. B) The bottom image shows the system's resilience to external perturbations during static standing, where the 2-DoF ankle control plays a key role in maintaining stability.

- **DWL Baseline(ours):** This baseline involves the application of the DWL policy with active ankle control. For this configuration, we set the PD gains of the ankle joints to $K_p = 20$ and $K_d = 5$. The network architecture details of DWL can be found in Appendix Table VI. The total trainable parameters of the DWL actor is about **320,192**.
- **PPO with Ankle Control:** Here, we eliminate the denoising loss component while retaining the other aspects of our methodology. This setup aims to underscore the enhanced adaptability of our approach in contrast to traditional methods. The network architecture details of PPO can be found in Appendix Table VII. The total trainable parameters of the PPO actor is about **333,312**.
- **DWL without Ankle Control:** Given the complexity of modeling closed kinematic chain ankle mechanisms in bipedal and humanoid robots, many previous RL-based locomotion controls have utilized passive ankle strategies. We conduct comparisons with a DWL variant employing **passive ankle control**[1] (with $K_p = 0$ and $K_d = 10$) to benchmark against this common approach.

Our experimental scenarios are diverse, including tasks such as snowy ground, up and down stairs, and disturbance rejection. During these tasks, the robot's arms are maintained in a stationary position to isolate the assessment of locomotive performance. This experimental setup provides a testbed to evaluate the versatility and robustness of our locomotion control strategy in real-world conditions. The subsequent benchmark comparisons are conducted exclusively using XBot-S to ensure consistency in our evaluation.

TABLE III: Real robot testing across various terrains. Bold values is our DWL with ankle control, DWL$_p$ is DWL with passive ankle, PPO control ankle as well.

Algorithm	**Slope**	**Stair-up**	**Stair-down**	**Irregular**
PPO	80%	20%	60%	20%
DWL$_p$	80%	20%	100%	40%
DWL	100%	100%	100%	100%

B. Indoor Experimental Validation

A comprehensive suite of real-world trials is executed to assess the robustness and adaptability of our algorithm in controlling the humanoid robot across a series of challenging terrains. Our indoor experiments employed four distinct terrain types with different difficulties, detailed as follows:

- **Slope Transit (Fig. 1F):** A sloped platform with a gradient of 0.25 to test the robot's capacity to adeptly shift from planar to inclined locomotion, encompassing both ascent and descent.
- **Stair Descent (Fig. 1D):** Tasked with a downward traversal, the robot encountered stairs, each spanning $20cm$ wide and $10cm$ high, commencing from the summit.
- **Stair Ascent (Fig. 1B):** Matching the descent staircase in dimension, the robot faces the challenge of ascending the stairs with limited sensor observations.
- **Irregular Terrain (Fig. 1E):** A custom-designed landscape with variable elevations up to $10cm$, simulating the unpredictability of challenging terrains.

The results of these experiments, quantified in Table III, reveal significant insights. When comparing PPO with ankle control to our DWL framework, our approach shows a significant improvement in walking performance, achieving **100%** success rates across various terrains. This highlights the superior sim-to-real capabilities of our method and its robust adaptability to diverse terrains. For relatively simple tasks like navigating slopes and descending stairs, both PPO and the passive variant of DWL (DWL$_p$) demonstrate competent success rates. However, in challenging situations such as walking on irregular terrain or climbing stairs, DWL distinctly

[1] We specify that "passive" ankle control in our context means the ankle's movement isn't directly controlled by the policy but responds based on predefined physical damping properties, which is different from active RL agent decision-making.

outperforms, showcasing the adaptability and robustness of our method.

C. Outdoor Experiments

In addition to extensive indoor testing on various complex terrains, we also conducted prolonged outdoor walking tests across diverse and challenging landscapes. We assessed walking performance on different surfaces and conditions, including cemented ground, brick roads, soil, and snowy terrain. Our algorithm allowed the robot to exhibit stable walking across the aforementioned varied road conditions. Particularly noteworthy is the walking evaluation on snow-covered terrain Fig. 1A , which presents a highly challenging task. Snow, with its deformable nature, poses difficulties as the robot's feet can sink into it, a situation challenging to simulate. Furthermore, snow surfaces tend to be slippery, making robots prone to sliding. Our DWL algorithm, however, demonstrates remarkable stability during prolonged walks on snowy terrain, affirming the robustness and adaptability of our algorithm to diverse terrains.

D. Robustness Testing

Domain randomization is a common approach to achieving a robust controller. However, its effectiveness is often limited by the sim-to-real gap and the impracticality of accounting for every possible real-world scenario.

In our framework, the agent is designed to predict the true state, thereby empowering the controller to rapidly adapt to a variety of situations. For example, if there is a tendency to fall, the controller can quickly recognize this and act to maintain balance. To assess the robustness of our controller, we conducted the following experiments:

1) Mass Displacement: As shown in Fig. 1G, the robot carried a bag into which we progressively added heavy objects while it was walking. Even with an additional weight of up to 15kg, over a third of the robot's weight, it managed to sustain stable locomotion.

2) Heavy load: We executed two experiments, depicted in Fig. 1H. It successfully walked with an additional load of up to 20kg, albeit with a slightly lowered height due to the added weight. Also, we attached the robot's hand effector to a loaded cart Fig. 1I, the robot was able to push a cart loaded with 60kg, demonstrating the controller's adaptability to handle significant loads.

3) Push Recovery: During our experimentation, we subjected the robot to external forces from multiple directions while it was executing continuous standing commands. These tests were carried out on both flat(Fig. 1C) and sloped terrain(Fig. 4), with the robot successfully maintaining its stance in both conditions.

VI. Result Analysis

In this section, we delve into the empirical results obtained from the deployment of the DWL framework on a humanoid robot, specifically focusing on its performance in terrain traversing. Our results are shown in Fig. 5.

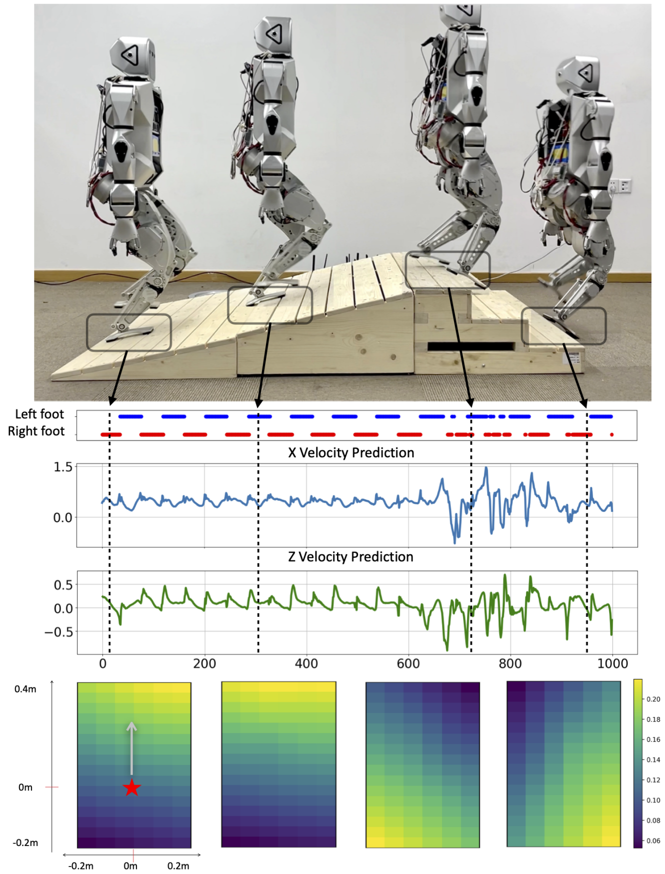

Fig. 5: State estimation results of DWL-facilitated complex terrain traversing and adaptation. This sequence of images visualizes the model's prediction of foot contact, base velocity, and heightmap when the humanoid robot navigates through a slope and stairs. The results demonstrate DWL's effectiveness in state estimation and online adaption.

1) Terrain Height Scan Prediction and Gait Adaptation: Our findings highlight the DWL's remarkable ability to predict terrain height, an essential factor for terrain adaptability. Utilizing only proprioceptive inputs, rather than relying on LiDAR or depth cameras, our system is capable of estimating a terrain's rough profile. At first glance, this might seem implausible, but our approach can discern the general trend of the terrain, as exemplified in Fig. 5. DWL's internal model adeptly encodes environmental features with distinct separability, thus facilitating accurate terrain recognition. The robot can identify whether it is walking up a slope or descending stairs. While precise shape prediction remains challenging, even a rough estimate proves immensely useful. This capability crucially affects the robot's gait, as observed when it transits from slope to stairs. Such gait modifications are imperative not just for navigating obstacles but also for ensuring balance and stability across diverse terrains.

2) Foot Contact Detection and Its Implications: As demonstrated in Fig. 5, foot contact patterns indicate a correlation with the type of terrain encountered. In humanoid locomotion, especially during single-leg support phases, accurate detection of foot contacts is essential for stability. The DWL framework aids in predicting these contact instances, thereby improving the planning of leg swing trajectories and facilitating effective obstacle avoidance. The frequency and pattern of foot contacts vary dynamically, guided by state estimation, leading to crucial gait adjustments and adaptations for complex terrain navigation.

3) Velocity Estimation: Velocity prediction, particularly linear velocity, is challenging to obtain directly from proprioceptive sensors but is critical for successful locomotion. DWL effectively estimates velocity states within a unified framework, addressing challenges like IMU angular yaw drift, as shown in Appendix Fig. 10. This estimation enhances command following and prevents yaw deviations. The congruence between actual and estimated velocities observed in our experiments significantly aids in the robot's locomotion, ensuring smoother and more predictable movements. Since real states cannot be directly obtained in real-world settings, we validate our state prediction using a sim2sim transfer. We tested our policy in MuJoCo and compared the state estimation with the ground truth. The predictions of linear velocity and Euler yaw are displayed in Appendix Fig.12 and Fig. 11.

The mean square error (MSE) of the forward velocity estimation in 60 seconds is 0.046 in the sim-to-sim scenario, while the IMU drift was reduced by around 87% in real-world experiments. These comparisons clearly demonstrate the accurate state estimation capabilities of the DWL algorithm.

4) Benefits and Importance of Ankle Control: By allowing activated control of both the two freedoms of the ankle, the robot is empowered to traverse complex terrains and resist extra forces. The role of ankle control was pivotal, as it allows the robot to preserve balance, even in single-leg support scenarios, generating human-like stability. Noticeably, ankle control becomes particularly pronounced when walking on irregular blocks and ascending stairs, as in Fig. 4, and also on deformable ground as shown in Fig. 6. Without ankle control, the robot's feet deformed significantly upon contacting the uneven ground, failing to recover. This was coupled with inadequate ankle joint contact forces, raising the risk of imbalance and falls during foot placements. In contrast, our ankle control method navigates these terrains with ease by adaptive contact forces at the ankle joints, as shown in the torque plot in Fig. 4, enabling the robot's adaptability to varied terrains.

VII. Conclusion

In this work, we proposed Denoising World Model Learning (DWL) for complex humanoid robot locomotion skill learning. The framework first masked out privileged information and injected appropriate noise into the true state observed in the simulation. It then designs an auto-encoder architecture to denoise the observations and reconstruct the true state. We achieved success in extensive real-world experiments in various complex environments such as snowy land, stairs, deformable ground, and irregular surfaces. Demonstrating the world's first humanoid robot to master challenging terrains with end-to-end RL and zero-shot sim-to-real transfer. In-depth result analysis indicated the effectiveness of the state estimation capability and the importance of the active 2-DoF ankle control. In the future, visual information will be added to enable more efficient navigation in challenging terrains while maintaining robustness.

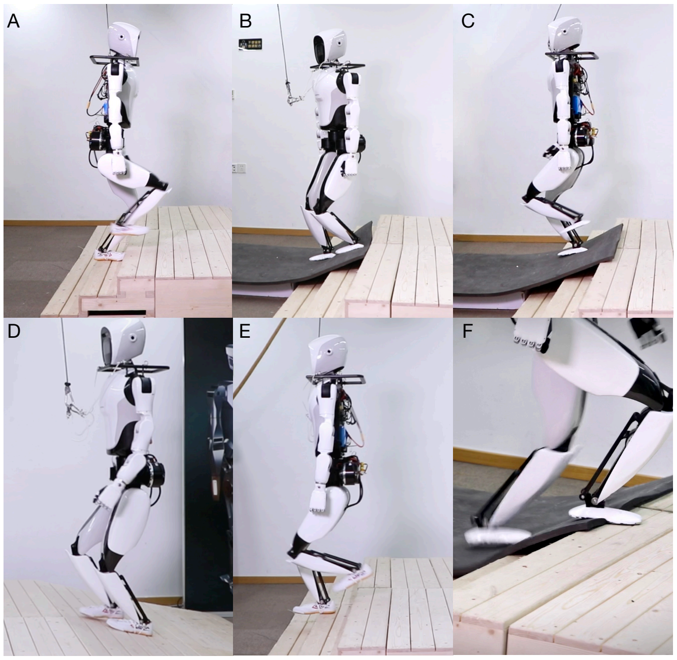

Fig. 6: We deployed the DWL agent on a large humanoid robot (1.65 meters tall, 57 kilograms) to evaluate its performance in various scenarios, including highly challenging deformable conditions. The active control of the 2-DOF ankle joints proved particularly beneficial for maintaining balance while standing, recovering from external disturbances, and traversing through intricate and deformable terrain.

References

[1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning*, pages 403–415. PMLR, 2023.

[2] Min Sung Ahn. *Development and Real-Time Optimization-based Control of a Full-sized Humanoid for Dynamic Walking and Running*. University of California, Los Angeles, 2023.

[3] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR, 2020.

[4] Tianlong Chen, Zhenyu Zhang, Pengjun Wang, Santosh Balachandra, Haoyu Ma, Zehao Wang, and Zhangyang Wang. Sparsity winning twice: Better robust generalization from more efficient training. *arXiv preprint arXiv:2202.09844*, 2022.

[5] Matthew Chignoli, Donghyun Kim, Elijah Stanger-Jones, and Sangbae Kim. The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2021.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE, 2017.

[8] Xinyang Gu, Yen-Jen Wang, and Jianyu Chen. Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer. *arXiv preprint arXiv:2404.05695*, 2024.

[9] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[10] Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. *Advances in neural information processing systems*, 31, 2018.

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[12] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[13] Ichiro Kato. Information-power machine with senses and limbs (wabot 1). In *First CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators*, volume 1, pages 11–24. Springer-Verlag, 1974.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[16] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40:429–455, 2016.

[17] Ashish Kumar, Zhongyu Li, Jun Zeng, Deepak Pathak, Koushil Sreenath, and Jitendra Malik. Adapting rapid motor adaptation for bipedal robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1161–1168. IEEE, 2022.

[18] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5 (47):eabc5986, 2020.

[19] Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018.

[20] Junheng Li and Quan Nguyen. Multi-contact mpc for dynamic loco-manipulation on humanoid robots. In *2023 American Control Conference (ACC)*, pages 1215–1220. IEEE, 2023.

[21] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[22] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Robust and versatile bipedal jumping control through multi-task reinforcement learning. *arXiv preprint arXiv:2302.09450*, 2023.

[23] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[24] Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control. *arXiv preprint arXiv:1811.11711*, 2018.

[25] Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017.

[26] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Dar-

rell, Jitendra Malik, and Koushil Sreenath. Learning humanoid locomotion with transformers. *arXiv preprint arXiv:2303.03381*, 2023.

[27] Jacob Reher, Wen-Loong Ma, and Aaron D Ames. Dynamic walking with compliance on a cassie bipedal robot. In *2019 18th European Control Conference (ECC)*, pages 2589–2595. IEEE, 2019.

[28] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

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

[30] Luis Sentis and Oussama Khatib. A whole-body control framework for humanoids operating in human environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2641–2648. IEEE, 2006.

[31] Jonah Siekmann, Yesh Godse, Alan Fern, and Jonathan Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7309–7315. IEEE, 2021.

[32] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[33] Jean-Pierre Sleiman, Farbod Farshidian, Maria Vittoria Minniti, and Marco Hutter. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6(3):4688–4695, 2021.

[34] Tomomichi Sugihara, Yoshihiko Nakamura, and Hirochika Inoue. Real-time humanoid motion generation through zmp manipulation based on inverted pendulum control. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1404–1409. IEEE, 2002.

[35] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

[36] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[37] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter 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)*, pages 23–30. IEEE, 2017.

[38] Patrick M Wensing and David E Orin. Development of high-span running long jumps for humanoids. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 222–227. IEEE, 2014.

[39] Chuanyu Yang, Kai Yuan, Shuai Heng, Taku Komura, and Zhibin Li. Learning natural locomotion behaviors for humanoid robots using human bias. *IEEE Robotics and Automation Letters*, 5(2):2610–2617, 2020.

Appendix

A. *Quintic Polynomial Foot Trajectory*

In Section.IV-B, we introduced the foot trajectory interpolation method we designed. Through this approach, a smooth foot trajectory can be generated as a reference, and a tracking reward is utilized to encourage the robot's feet to follow the generated trajectory. Table.IV presents the specific polynomial parameters and optimization conditions for generating foot trajectories. Figure.7 illustrates the curves of height, velocity, and acceleration for a generated foot trajectory. It can be observed that smoothness is maintained across the positional, velocity, and acceleration aspects.

TABLE IV: Quintic Polynomial Foot Trajectory Parameters $f(t) = \sum_{k \leq 5} a_k t^k$ and Optimization Conditions

Trajectory Parameters		**Optimization constraints**	
Coefficient	**Value**	**Objection**	**Value**
a_5	9.6	h_0	0.0
a_4	12.0	h_T	0.0
a_3	-18.8	v_0	0.1
a_2	5.0	v_T	0.0
a_1	0.1	h_{max}	0.1
a_0	0.0	T	0.5

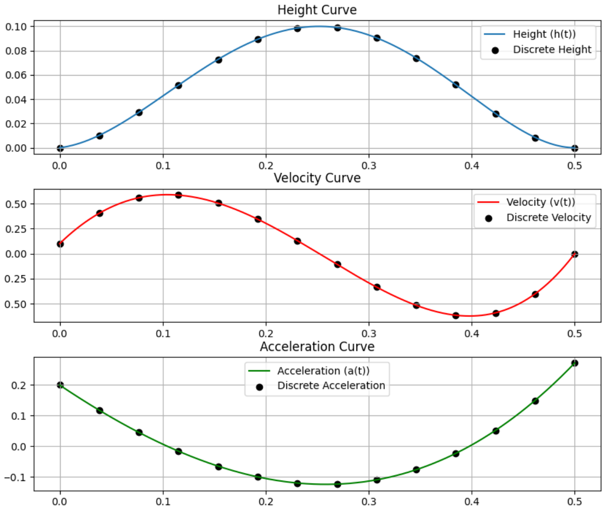

Fig. 7: The foot trajectory, modeled through quintic polynomial interpolation, is detailed in TABLE IV. It illustrates the critical velocity and acceleration constraints, ensuring a trajectory that facilitates seamless motion, harmonizing stability and gait efficiency in the robot's movement dynamics.

B. *Reward Function*

For periodic reward design, inspired by previous work [31, 32], our objective is to construct a reward function that leverages the distinct roles of foot forces and foot velocities. Specifically, the function aims to promote higher foot velocities during the swing phase and reasonable foot forces during the stance phase of locomotion. We introduce a binary feet contact indicator, $I(t)$, termed the periodic stance mask. As illustrated in Fig. 8, this indicator is set to 1 during the planned contact phase and to 0 during the planned swing phase, alternating for each leg throughout a locomotion cycle. The periodic rewards are formulated as follows:

$$r^{\text{periodic}}_{\text{Force}}(t) = I_L(t) \cdot F_L + I_R(t) \cdot F_R \tag{7}$$

$$r^{\text{periodic}}_{\text{velocity}}(t) = (1 - I_L(t)) \cdot \dot{P}^f_L + (1 - I_R(t)) \cdot \dot{P}^f_R \tag{8}$$

In this context, the symbol F denotes force, while L and R refer to the left and right foot, respectively. For clarity, we omit the scaling and clipping factors typically applied to the forces and velocities, which are necessary due to their differing magnitudes. For instance, forces, which can reach magnitudes of hundreds, are scaled down by a factor of 400 and subsequently clipped to the range [0, 1] to maintain consistency in the reward function's scale.

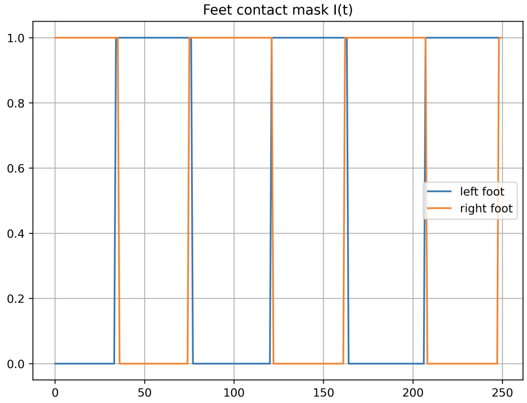

Fig. 8: The stance mask for the left (L) and right (R) feet, where 0 indicates the swing phase and 1 indicates the stance phase is expected.

The reward function is summarized in Table V. It is important to note that the commands CMD$_{z,\gamma,\beta}$ are intentionally set to zero. This is because we do not control them; rather, we aim to maintain their values at zero to ensure stable and smooth walking. Therefore, the total reward at any time step t is computed as the weighted sum of individual reward components, expressed as $r_t = \sum_i r_i \cdot \mu_i$, where μ_i represents the weighting factor for each reward component r_i.

C. *Training Details*

SectionIV-A provides a detailed explanation of our designed Denoising World Model Learning(DWL) method, and here in Table.VIII, we outline the hyperparameters for DWL. Table.VI presents the specific network architecture we utilized. It includes the encoder and decoder for DWL, as well as the actor and critic.

D. *Additional Experimental Results*

We present further results and demonstrations of our experiments, as illustrated in Fig. 9.

TABLE V: In defining the reward function, we use a tracking error metric denoted by $\phi(e, w)$. This metric is expressed as $\phi(e, w) := \exp(-w \cdot \|e\|^2)$, where e represents the tracking error, and w is the associated weight. The target base height is set to 0.7 m.

Reward	**Equation** (r_i)	**reward scale**(μ_i)
Lin. velocity tracking	$\phi(\dot{P}^b_{xyz} - \text{CMD}_{xyz}, 5)$	1.0
Ang. velocity tracking	$\phi(\dot{P}^b_{\alpha\beta\gamma} - \text{CMD}_{\alpha\beta\gamma}, 7)$	1.0
Orientation tracking	$\phi(P^b_{\alpha\beta}, 5)$	1.0
Base height tracking	$\phi(P^b_z - 0.7, 10)$	0.5
Periodic Force	$r^{periodic}_{Force}(t)$	1.0
Periodic Velocity	$r^{periodic}_{velocity}(t)$	1.0
Foot height tracking	$\phi(P^f_z - f_t, 5)$	1.0
Foot vel tracking	$\phi(\dot{P}^f_z - \dot{f}_t, 3)$	0.5
Default Joint	$\phi(\theta_t - \theta_0, 2)$	0.2
Energy Cost	$\|\tau\|\|\dot{\theta}\|$	-0.0001
Action Smoothness	$\|a_t - 2a_{t-1} + a_{t-2}\|_2$	-0.01
Feet movements	$\|\dot{P}^f_z\|_2 + \|\ddot{P}^f_z\|_2$	-0.01
Large contact	$\text{CLIP}(F_{L,R} - 400, 0, 100)$	-0.01

TABLE VI: DWL Network Architecture Details

Component	**Configuration**
Encoder	
RNN_memory(0)	GRU(47 → 256)
emb_model (1)	Linear(256 → 256)
emb_model (2)	ELU(alpha=1.0)
emb_model (3)	Linear(256 → 24)
Decoder	
denoise_net (0)	Linear(24 → 64)
denoise_net (1)	ELU(alpha=1.0)
denoise_net (2)	Linear(64 → 184)
Actor	
policy_net (0)	Linear(24 → 48)
policy_net (1)	ELU(alpha=1.0)
policy_net (2)	Linear(48 → 12)
Critic	
Critic_Net (0)	Linear(184 → 512)
Critic_Net (1)	ELU(alpha=1.0)
Critic_Net (2)	Linear(512 → 512)
Critic_Net (3)	ELU(alpha=1.0)
Critic_Net (4)	Linear(512 → 256)
Critic_Net (5)	ELU(alpha=1.0)
Critic_Net (6)	Linear(256 → 1)

TABLE VII: PPO Network Architecture Details

Component	**Configuration**
Actor	
RNN_memory(0)	GRU(47 → 256)
policy_net (1)	Linear(256 → 256)
policy_net (2)	ELU(alpha=1.0)
policy_net (3)	Linear(256 → 128)
policy_net (4)	ELU(alpha=1.0)
policy_net (5)	Linear(128 → 12)
Critic	
Critic_Net (0)	Linear(184 → 512)
Critic_Net (1)	ELU(alpha=1.0)
Critic_Net (2)	Linear(512 → 512)
Critic_Net (3)	ELU(alpha=1.0)
Critic_Net (4)	Linear(512 → 256)
Critic_Net (5)	ELU(alpha=1.0)
Critic_Net (6)	Linear(256 → 1)

TABLE VIII: Hyperparameters of DWL.

Parameter	**Value**
Number of Environments	12288
Number Training Epochs	2
Batch size	12288 × 24
Episode Length	2400 steps
Discount Factor	0.995
GAE discount factor	0.95
Entropy Regularization Coefficient	0.005
$c1$	0.8
$c2$	1.2
Learning rate	1e-5
regularization coefficient λ_r	0.002
policy coefficient λ_π	5
value coefficient λ_v	5

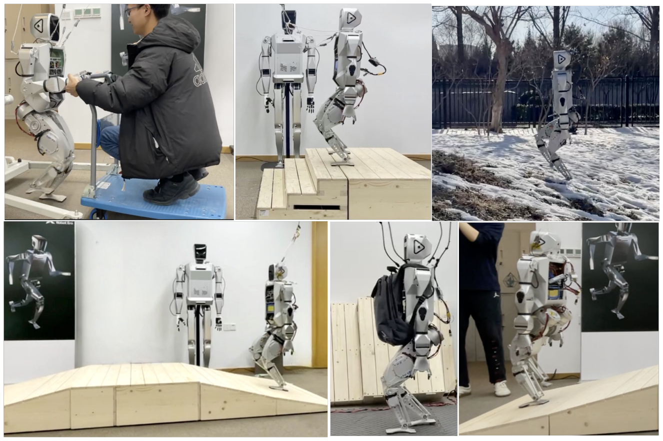

Fig. 9: Additional Experiment Setups and Results

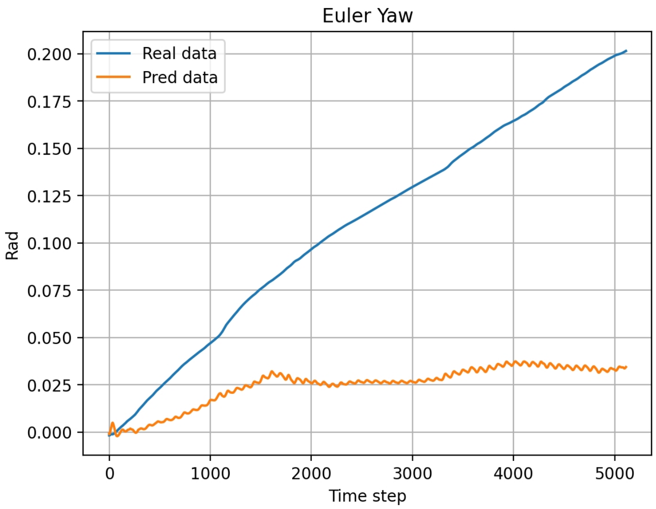

Fig. 10: Even when the robot is stationary, the actual IMU readings exhibit the phenomenon of IMU drift. Conversely, the DWL algorithm is capable of predicting real IMU data and mitigating the IMU drift by approximately 87%.

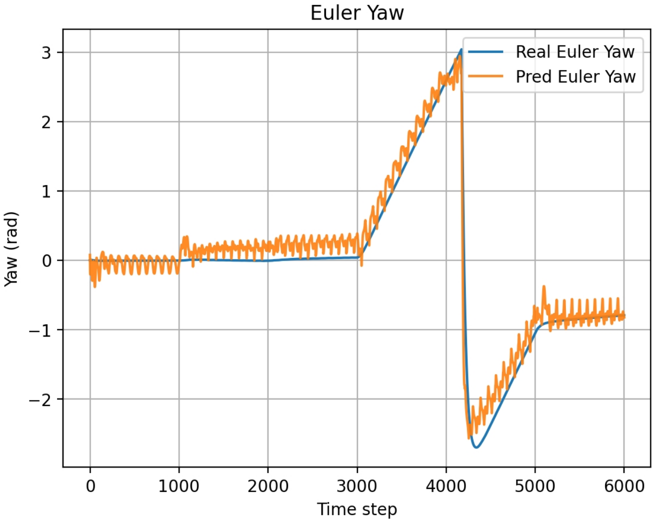

Fig. 11: To emulate human command inputs, we varied the command input every 1000 steps (10 seconds). In the MuJoCo simulation environment, the Euler yaw angle predicted by the DWL algorithm is close to the ground truth with MSE 0.074.

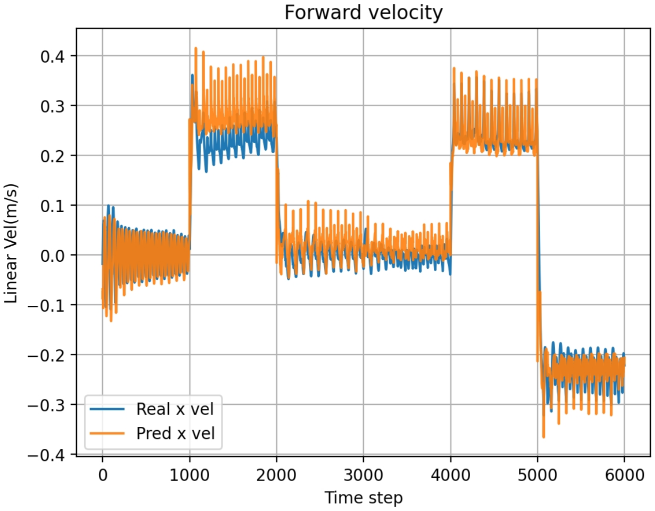

Fig. 12: Comparison of Estimated and Actual Velocity Values in MuJoCo: The forward velocity predicted by the DWL algorithm closely approximates the ground truth with MSE 0.046.