2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Toward Understanding Key Estimation in Learning Robust Humanoid Locomotion

Zhicheng Wang[1] Wandi Wei[1] Ruiqi Yu[1] Jun Wu[1,2] Qiuguo Zhu*[1,2]

***Abstract*— Accurate state estimation plays a critical role in ensuring the robust control of humanoid robots, particularly in the context of learning-based control policies for legged robots. However, there is a notable gap in analytical research concerning estimations. Therefore, we endeavor to further understand how various types of estimations influence the decision-making processes of policies. In this paper, we provide quantitative insight into the effectiveness of learned state estimations, employing saliency analysis to identify key estimation variables and optimize their combination for humanoid locomotion tasks. Evaluations assessing tracking precision and robustness are conducted on comparative groups of policies with varying estimation combinations in both simulated and real-world environments. Results validated that the proposed policy is capable of crossing the sim-to-real gap and demonstrating superior performance relative to alternative policy configurations.**

I. INTRODUCTION

Humanoid robots hold immense potential and practical value as general-purpose robots. However, their high degrees of freedom and system complexity pose significant challenges to achieving stable control.

Over the past few decades, researchers have proposed various methods to enhance the mobility of robots. Classical control methods can achieve stable static motion [1]–[3], and furthermore, optimization-based strategies can generate dynamic behaviors while adhering to constraints [4]–[7]. The most notable exemplar is the Boston Dynamics Atlas, capable of smoothly performing parkour, back-flipping, and object manipulation [8].

With the advancement of computational power, learning-based methods have become popular in legged robot control and have also achieved remarkable achievements. These learning methods were initially validated on quadruped robots, exhibiting the capability to traverse diverse terrains [9]–[12] and are beginning to incorporate visual information [13], [14]. On bipedal robots, learning policies are also capable of executing fundamental motions [15]–[18]. Additionally, there are endeavors that combine models with learning methods [19]–[21].

[1] The authors are with the Institute of Cyber-Systems and Control, Zhejiang University, 310027, China
[2] Qiuguo Zhu and Jun Wu are with State Key Laboratory of Industrial Control Technology, 310027, China
* Qiuguo Zhu (qgzhu@zju.edu.cn) is the corresponding author.
† This work was supported by the "Leading Goose" R&D Program of Zhejiang(Grant No. 2023C01177), the National Key R&D Program of China (Grant No. 2022YFB4701502), the Key R&D Project on Agriculture and Social Development in Hangzhou City (Asian Games)- (Grant No. 20230701A05), and the Key Research Project of Zhejiang Lab (Grant No. 2021NB0AL03)

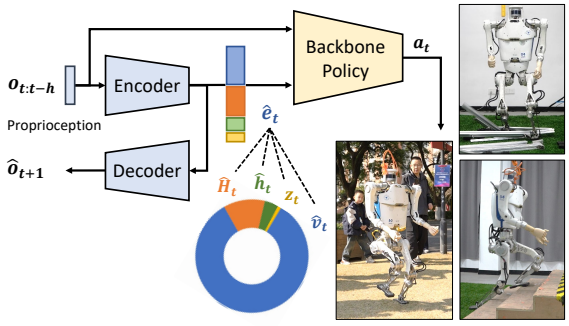


Fig. 1: Overview of key estimation policy. By quantifying the importance of the explicit estimation states and designing the key estimation architecture, the policy achieves real-world blind locomotion with a real Wukong-IV humanoid.

In theory, the more information a policy acquires, the better its performance. However, not all data is readily available in real-world robot deployments, such as contact forces, deformable surfaces, and precise velocities, which are also known as privileged states. Consequently, state estimation is crucial for legged robot control systems. Traditional control methods utilize contact dynamics and forward kinematics to estimate these privileged states [22]–[25]. Yet, these model-based approaches are computationally intensive and predicated on assumptions like non-slippery, stiff, or flat terrain, potentially impairing real-world performance.

Learning-based estimations provide a solution to the computation and assumption problems. Recently, numerous learning-based approaches have been developed to infer privileged states from available observational data, thereby implicitly estimating the robot's states and its surrounding environment. A prominent example is the RMA framework [26]. Some studies opt for explicit estimation techniques [27]. However, the states inferred by RMA are implicit and heavily reliant on the training process, which can complicate interpretation. Additionally, the traditional two-stage training process is often inefficient. In contrast, the novel framework integrates explicit estimates with implicit vectors, such as the robot's linear velocity [28] and height map [29], thereby enhancing the robot's locomotive performance and adaptability to various terrains.

For humanoid robots, diverse estimation methods are speculated to enhance policy-driven motion performance. In the interdisciplinary domain of robotics and AI, the meticulous design and selection of estimation modules are vital. These modules not only define the breadth and depth

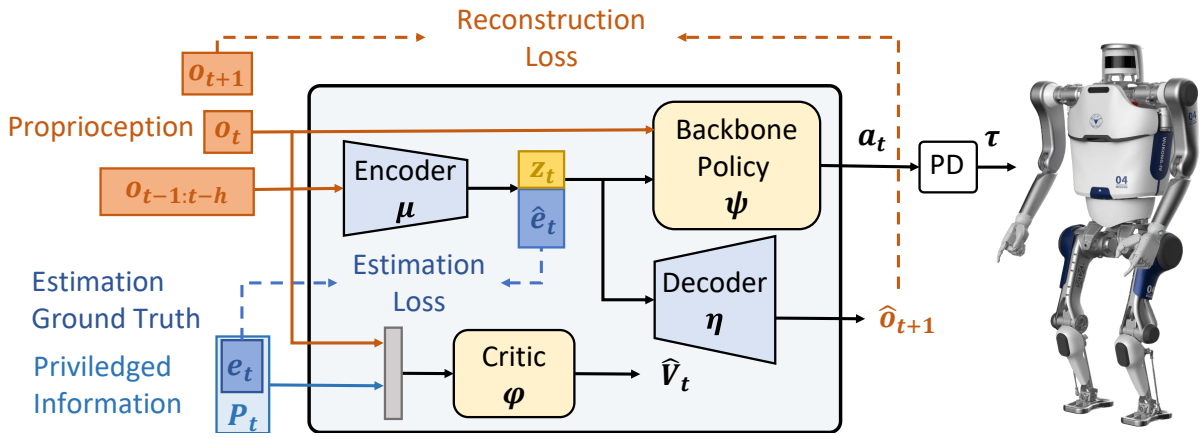


Fig. 2: Architecture of the proposed policy. The actor consists of an auto-encode (μ, η) and a backbone policy ψ. The encoder μ takes in history proprioception and generates estimations, implicit encoding z_t and explicit encoding $\hat{e}_t$. The decoder η reconstruct the current proprioception using z_t and $\hat{e}_t$. The $\hat{e}_t$ is fitted to the true values of corresponding physical variables e_t. Both encodings, along with current proprioception o_t, serve as input to the backbone policy, resulting in actions. The critic's input includes o_t and privileged information P_t that includes e_t and other useful data, then the critic computes value function $\hat{V}_t$.

of information assimilated by the learning algorithm but also leverage the extensive prior knowledge accumulated within the field of robotics. Consequently, several key questions arise: "Should estimations be explicit or implicit?" "Which variables should be estimated?" and "How do estimations assist the policy to finish the locomotion task?" While some research has investigated the significance of proprioception [30], quantitative analyses of the importance of estimation remain largely unexplored.

In this paper, we aim to explore the effect of estimation further. First, we trained a policy with the estimation of various privileged information. Through saliency analysis, we ranked the importance of each estimation and proposed an optimal combination. Then, we conducted various tests on policies with different estimations. The results indicate that the policy with the optimal combination of estimations achieves the best overall performance. During real robot deployment, our policy can walk through challenging environments like stairs, slopes, and obstacles when tracking velocity commands.

The main contributions of this work are:

- Quantitative analysis of the influence of the estimation variables on the performance of learned policies, and proposed the optimal combination.
- A controllable and adaptive framework for learning humanoid locomotion with the proposed effective estimation scheme based on asymmetric actor-critic.
- The proposed learning framework and estimation methodology are tested in the real world and prove to be capable of adapting to outdoor environments.

The structure of this paper is as follows: Section II describes the method to train our policy. Section III presents the training process and the experiments to evaluate our policy, including results and analysis. Section IV summarizes the paper and states the future work.

II. METHODOLOGY

A. Asymmetric Policy Architecture

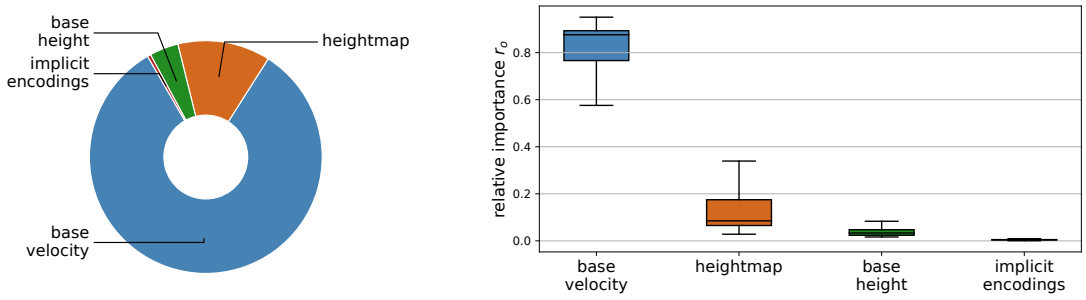
To avoid multiple training stages and imitation inaccuracies in teacher-student training architecture, this work follows an asymmetric actor-critic structure. The actor policy only has access to an h-step realistic observation history that contains delayed noisy proprioceptive information and commands. The critic policy has access to all kinds of states. The actor is composed of an auto-encoder (μ, η) that predictively reconstructs the proprioceptive information, and a backbone policy ψ that conditions joint-level position targets on the latest observation and estimation encodings. All components are fully connected neural networks. The size of hidden layers is described in Appendix III.

To train every part of the framework, the training loss function is formulated as a weighted sum of policy gradient loss, observation prediction loss, and estimation loss. In this work, the policy gradient loss is computed with Proximal Policy Optimization(PPO) [31].

The training process and the architecture of the policies are shown in Fig. 2.

B. State and Action

States are categorized into three types: observation, privilege, and command. The observation, denoted as $O_t \in \mathbb{R}^{42}$ includes the proprioceptive accessible from the actual robot. The observation is composed of the gravity vector, body angular velocity, joint position, joint velocity, and the previous action. The privileged information, represented as $P_t \in \mathbb{R}^{103}$, pertains to data that is intricate to acquire in the real world, including ground truth of body linear velocity, body height, small heightmap around the robot's feet and large heightmap around the robot's base. Except for a larger heightmap around the robot, all physical quantities



(A) (B)

Fig. 3: Saliency analysis of the estimation states. (A) Pie chart of the estimations' average relative importance. (B) Box plot of the ranges of the relative importance for all samples. The colored box refers to the range between 25% and 75% samples, the horizontal line is the median number, and the error bar shows the boundaries where $p < 0.05$.

in the privileged information can be explicitly estimated. User command $\in \mathbb{R}^7$ includes gait signal, desired linear velocity, and desired yaw angular velocity. All commands are expressed in the robot frame.

Action a_t is the desired joint position, which is executed by a subsequent PD controller. The policy runs and generates action at 100 Hz. The PD controller runs at 1kHz.

C. Reward

Bell-shape kernel functions are introduced into reward design to encourage the policy to survive. The adopted kernel functions can be formulated as:

$$G_{\alpha,\sigma}(x) = \alpha \exp\left(-\frac{x}{\sigma}\right)^2 \quad (1)$$

$$C_{\alpha,\beta,\sigma}(x) = \alpha\left(\left(\frac{x}{\sigma}\right)^{2\beta}+1\right)^{-1}, \beta = 1, 2, ... \quad (2)$$

where $G_{\alpha,\sigma}(x)$ represents the Gaussian kernel function, while $C_{\alpha,\beta,\sigma}(x)$ denotes the generalized Cauchy kernel function, with α, β, and σ serving as adjustable coefficients. The Gaussian function is frequently utilized in reward design due to its advantageous gradient behavior around zero, which enhances precision in tracking tasks. However, its gradients may become insignificantly small in distant regions. Conversely, the generalized Cauchy kernel function, characterized by its heavy-tailed feature, offers viable gradients for learning even when x is far from zero. Notably, this function possesses a stationary point at $(\pm\sigma, 0.5)$, simplifying the adjustment of the scaling coefficient σ. Moreover, the additional order coefficient β allows for modifications to the kernel's shape. A larger β value tends to shape the function more closely to a rectangular function, thereby imposing lesser penalties when x falls below σ.

The reward items in this work can be divided into the following categories: (1) *base command tracking*, (2) *gait*, (3) *smoothness and energy saving*.

1) Base command tracking: The commands include base height, base orientation, base linear, and angular velocity. The reward functions are defined as:

$$r_v = G_{0.1,0.02}(||v_t^* - v_t||) \quad (3)$$

$$r_\omega = G_{0.1,0.02}(||\omega_t^* - \omega_t||) \quad (4)$$

$$r_r = G_{0.1,0.0025}(1 - R_{2,2}^2) \quad (5)$$

$$r_h = G_{0.2,0.02}(|h^* - h|) \quad (6)$$

where h is the vertical distance from the base to the ground, $v \in \mathbb{R}^3$ represents the linear velocity, and $R_{2,2}$ is the bottom right corner value in the base rotation matrix, $1 - R_{2,2}$ measures the deviation between the Z-axis of robot base and the world frame.

2) Gait: We inherit the gait-related command and reward design from [32] to generate a controllable locomotion pattern. It penalizes contact force during the swing phase and foot movement during the contact phase. The reward functions are defined as:

$$r_{eVel} = C_{0.1,1,8}(Q_{v,l}V_l + Q_{v,r}V_r) \quad (7)$$

$$r_{eFrc} = C_{0.1,1,8}(Q_{f,l}F_l + Q_{f,r}F_r) \quad (8)$$

where $I_{v/f,l/r}$ represents the gait force/velocity penalization coefficient for each foot, $V_{l/r}$ is the foot velocity of the corresponding foot, and $F_{l/r}$ is the contact force of corresponding foot.

3) Smoothness and energy saving: We hope our policy can optimize the impact force on the foot, joint torque smoothness, joint velocity smoothness, and cost of transport(CoT). We adopt The reward functions are defined as:

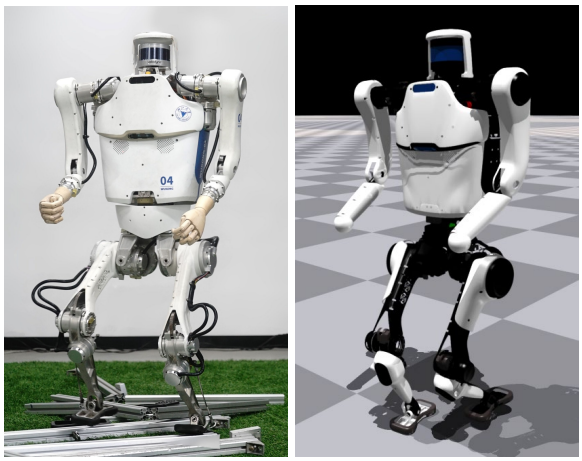
$$r_i = C_{0.1,3,0.2}\left(\frac{||F_t - F_{t-1}||}{mg}\right) \quad (9)$$

$$r_\tau = C_{0.1,2,160}(||\tau_t - \tau_{t-1}||) \quad (10)$$

$$r_{\dot{q}} = C_{0.1,1,8}\left(\frac{||\dot{q}_t - \dot{q}_{t-1}||}{||v_t||}\right) \quad (11)$$

$$r_{CoT} = C_{0.1,3,1.6}\left(\frac{\tau_t \cdot \dot{q}_t}{mg||v_t||}\right) \quad (12)$$

where F_t represents the foot force at step t, mg is the gravity of robot, τ_t is the joint torque at step t, $\dot{q}_t$ is the joint velocity at step t. CoT is a fairer cross-platform efficiency evaluation indicator [33], with better results than single penalty joint torque.



(A) (B)

Fig. 4: Wukong IV humanoid model. (A) Real-world Wukong-IV humanoid. (B) Simulated model in IsaacGym.

D. Training Environment Design

We adopted the game-inspired terrain curriculum from [34]. A sequence of complex terrain is generated from easy to hard. Once the robot moves far enough from the starting point in an episode, it will be spawned on the next-level terrain in the following episode. Conversely, the robot that failed to survive long enough will be spawned on an easier terrain in the next episode.

To achieve better real robot performance under a variety of commands, randomized commands, and domain randomization are applied to the training environment. Environment parameters and commands are sampled from associated uniform distributions at the beginning of every episode. Environment parameters include initial robot pose, mass payload, ground friction coefficient, motor strength, and observation latency. Commands include linear velocity, facing direction, and gait signal. Furthermore, a stochastic noise is added to the normalized proprioceptive observation to imitate sensor noise in real robots. The ranges of random parameters can be found in Appendix II.

III. EXPERIMENT AND RESULT

A. Platform description

The proposed training method is implemented on the Wukong-IV humanoid robot. It is 1.4 m tall, weighs 45 kg and is actuated by 21 electric motor joints. The robot has 6 degrees of freedom (DoF) on each leg and 4 DoFs on each arm. A picture of a real Wukong-IV robot and its dynamics model in a simulation environment is shown in Fig.4. The training environments are implemented in IsaacGym. The policies are built under PyTorch [35] framework. To prove that the proposed method is insensitive to random seeds, all mentioned policies are trained repeatedly for 6 times with CPU timestamp upon running as random seeds.

B. Saliency analysis

To understand the importance of estimation terms quantitatively, we adopted integrated gradients [36] from explainable artificial intelligence as saliency analysis metrics. It is a form of sensitivity analysis that computes the gradients of the output with respect to the input features and integrates these gradients along a path from a baseline input to the input of interest. Its goal is to identify the most influential features or input components that contribute to the model's output or decision. In practice, given N timesteps of input $x \in \mathbb{R}^n$ and a policy $F(x) \in \mathbb{R}^m$, the input x can be categorized into χ groups according to the physics meaning, and the dimension of the category is denoted as h, the saliency metrics can be formulated as:

$$G_{x_{i,t}} = \sum_{j=1}^{m} | \frac{x_{i,t} - \hat{x}_{i,t}}{p} \sum_{k=1}^{p} \frac{\partial F_j(\frac{p-k}{p}\hat{x}_t + \frac{k}{p}x_t)}{\partial x_{it}} | \quad (13)$$

$$S_d(x_{i,t}) = \max\left(G_{x_{i,t}} - \epsilon, 0\right) \quad (14)$$

$$\epsilon = \frac{1}{nN} \sum_{i=1}^{n} \sum_{t=1}^{N} G_{x_{i,t}} \quad (15)$$

$$S(x_{i,t}) = \frac{S_d(x_{i,t})}{\max(S_d(x_{i,t}))} \quad (16)$$

$${}^{E}I_i = \sum_{t=1}^{N} S(x_{i,t}) \quad (17)$$

$${}^{T}I_o = \frac{1}{h} \sum_{q=1}^{h} {}^{E}I_q \quad (18)$$

$$\iota_o = \frac{I_o}{\sum_{k=1}^{\chi} {}^{T}I_k} \quad (19)$$

where $G_{x_{i,t}}$ denotes the integrated gradient, a measure of how much the output changes with respect to a specific input feature. S_d signifies the absolute saliency, while S indicates the relative saliency, both of which quantify the significance of different input features. ${}^{E}I$ and ${}^{T}I$ represent element-wise and total importance, respectively, and they reflect the degree to which each input channel can sway the output. The Greek letter ι symbolizes the relative importance of the input category o. The term $\hat{x}_{i,t}$ refers to the nominal input, typically set to zero due to the normalization of all estimation terms. Lastly, $p = 25$ defines the integral horizon, a user-selected constant that determines the extent of the integration period.

Using saliency metrics, we systematically assess the impact of estimated values on the resultant actions. In our study, we begin by constructing a comprehensive estimation policy, a full estimation policy designed to encompass all potentially relevant states for humanoid locomotion tasks. Specifically, we incorporate explicit estimation terms pertaining to base linear velocity, base height, and the heightmap surrounding the feet, while incorporating fixed-width implicit information for subsequent saliency analysis. Considering the saliency identification about proprioceptive information has been studied in the literature [37], in this work, we only focus on the estimation terms computed by the encoder network

μ, rather than other proprioceptive information obtained from the simulation. The saliency results are shown in Fig.3.

According to the saliency experiments, we can see that the most crucial estimation is the linear velocity, which takes 0.845 relative importance on average. This can be expected because the training prioritizes linear velocity velocity tracking. The second important estimation is heightmap around the feet, which is closely connected to the next motion decision. The base height and implicit encoding showed a small influence on the final action.

C. Comparison group setup

After the saliency test, we have quantitatively understood the importance of different estimation terms to the action. In this section, we will examine the influence on the final performance of the policies in more detail. The saliency tests showed that the velocity estimation has the biggest impact on the behavior of the policy, so we assume that estimating states with higher importance would improve the overall locomotion performance. To compare the performance of policies that adopt different estimations, we set up six experimental groups as follows:

- **EstimatorNet**(EstNet) [27]: This policy's encoder only works as an explicit estimator. It estimates body linear velocity without any implicit encoding or decoder to reconstruct the proprioceptive observation.
- **Velocity Estimation**(Key1): This policy estimates the body velocity along with a 16-dimension implicit vector. The velocity proved to be the most important estimation in the saliency analysis, and we also left enough space for the policy itself to encode useful information from the proprioceptive history.
- **Key Estimation**(Key2): This policy estimates the two most important states, base linear velocity, and heightmap around the feet, which are the top 2 key estimation states. The encoder of this policy also encode the information into a 16-dimension implicit vector.
- **Full Estimation**(FullEst): This policy's structure is the same as the policy used in saliency analysis. It estimates all mentioned states, including base linear velocity, heightmap surrounding the feet, and body height. The encoder of this policy also encode the information into a 16-dimension implicit vector.
- **Irrelevant Estimation**(IrrEst): This policy estimates the least important explicit state, the base height, along with a 16-dimension implicit vector.
- **Implicit Encoding**(Implicit) This policy has no explicit estimation. The encoder gives only a 16-dimension implicit vector.

D. Performance metrics

In evaluating the performance of trained policies for humanoid locomotion, we employ a comprehensive set of metrics to assess various aspects of task execution.

We gauge the effectiveness of the learned policies by considering the final reward level at convergence, which is calculated as the mean episodic reward over the last 10 episodes of training. This metric provides insight into the overall success achieved by the policies in accomplishing the designated locomotion tasks. The the average reward level at convergence is denoted as $\bar{r}_{final}$.

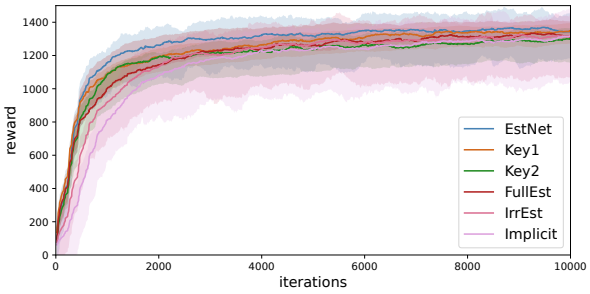


Fig. 5: Reward plots of different trials. The solid lines are the filtered mean episode reward, and the shaded area marks the ranges of the episode reward values.

Furthermore, we scrutinize the policies' velocity tracking accuracy by comparing the root mean square (RMS) velocity error. In simulation, the error is computed by testing the policies with 1024 10-second random constant command trajectories on flat terrain. In the real world, the error is computed by testing the policies with a predefined 60-sec template velocity command trajectory with ramps and slopes 16 times in indoor settings, and in this case, LiDAR odometry serves as the measurement of body velocity. In both scenarios, the maximal velocity command is 1.2m/s. The velocity tracking error is denoted as $RMS(\Delta v_{sim/real})$.

Additionally, we evaluate the policies' orientation stability by quantifying the RMS body orientation fluctuation when tracking the aforementioned template velocity commands on flat ground. We define orientation fluctuation by the distance between the gravity vector and its nominal value. The orientation fluctuation metrics are denoted as $RMS(\Delta g_{sim/real})$.

Moreover, to ascertain the traversability and robustness of the policies, we measure the successful traversal rate over a variety of challenging terrains in the real world, including 3-step 10cm stairs, 25-degree slopes, terrain featuring random 4cm metal profiles, and natural grass fields. The successful rates are computed based on 20 trials on each type of terrain with a human operator sending velocity commands. These metrics collectively provide a comprehensive evaluation framework, allowing us to assess the policies' performance across diverse locomotion scenarios and environments. The successful rates are denoted as $TSR_{stair}, TSR_{slope}, TSR_{metal}, TSR_{grass}$.

E. Experiments and Analysis

The training trials' reward curves are presented in Figure 5. The plots indicate a convergence towards a reward level averaging approximately 1300, suggesting that all tested policies successfully acquired and refined essential locomotion skills. Variances in average final rewards among different policy groups, with differences of up to 2.98%, are negligible considering the impact of domain randomization on mean episodic rewards. These minor discrepancies lead

TABLE I: Performance summary

	EstNet	**Key1**	**Key2**	**FullEst**	**IrrEst**	**Implicit**
$\bar{r}_{final}$	1318	**1347**	1308	1324	1332	1313
$RMS(\Delta v_{sim})$	0.2247	**0.2231**	0.2330	0.2256	0.3892	0.4343
$RMS(\Delta g_{sim})$	0.0228	0.0184	0.0219	0.0187	0.0275	**0.0162**
$RMS(\Delta v_{real})$	0.2396	0.2671	0.2553	**0.2270**	0.3945	0.3919
$RMS(\Delta g_{real})$	0.0283	0.0180	**0.0143**	0.0281	0.0339	0.0258
TSR_{stair}	0.75	0.75	**0.80**	0.70	0.10	0.25
TSR_{slope}	0.85	0.90	**0.95**	0.90	0.80	0.75
TSR_{metal}	0.70	**1.00**	**1.00**	**1.00**	0.50	0.50
TSR_{grass}	0.95	**1.00**	**1.00**	0.95	0.80	0.90

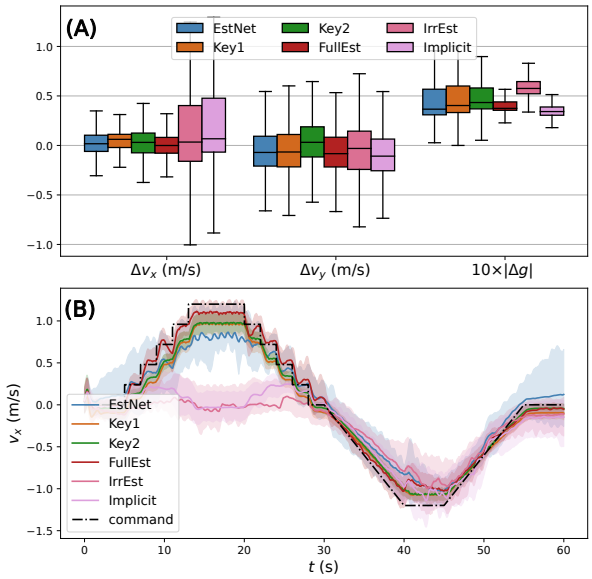


Fig. 6: Velocity tracking plots of various estimation policies. (A) The figure presents a box plot illustrating the distribution of velocity and orientation tracking errors in a simulated environment. Orientation error is magnified tenfold for improved visibility. Each colored box denotes the interquartile range, spanning from the 25th to the 75th percentiles, with the horizontal line representing the median value. Additionally, error bars indicate the boundaries where $p < 0.05$. (B) This figure includes velocity plots showcasing the policies' real-world performance in tracking a predefined command trajectory (indicated by the black dashed plot). The shaded area encompasses the range of measured velocities across all repeated trials, while the solid lines represent the mean velocity values derived from all recorded trajectories.

us to conclude that all policies have undergone equivalent training, affirming the effectiveness of our comparative group design in ensuring the policies' ability to master locomotion skills.

The base velocity tracking performance is displayed in Figure 6, with a quantitative summary provided in Table I. Figure 6(A) reveals significant variation in sagittal velocity tracking among policy groups, while coronal performances are more consistent. This variation is attributed to the humanoid robot's limited feasible task space, which restricts its capacity to counteract coronal velocity errors. Policies lacking explicit base velocity estimations exhibit poorer tracking performance in the sagittal direction. The asymmetrical velocity error distributions, with larger areas above zero, indicate a tendency to underperform on large forward velocity commands. This underscores the importance of explicit velocity estimation in learning humanoid locomotion tasks.

The right column of Figure 6(A) shows the orientation error distribution. Policies using implicit encoding mechanisms appear to offer superior pose stability compared to others. However, closer inspection reveals that this stability arises from a reduced propensity for movement. Optimal orientation stability is achieved by policies that incorporate full estimation strategies, explicitly accounting for a broader range of state variables.

Figure 6(B) illustrates the velocity profiles of the comparison groups, tracking a predefined trajectory of steps and ramps. Policies without explicit velocity estimation struggle to track forward velocities, though they comply with backward commands. In contrast, policies with explicit velocity estimation capabilities generally maintain a velocity of 1.2 m/s. This figure also reflects on sim-to-real transfer proficiency, with EstimatorNet policies showing a decline in real-world accuracy compared to simulation, indicating that implicit encoding can improve robustness even though it only has small importance.

Robustness findings are detailed in Table I. Policies with Key2 estimations, focusing on the two most critical variables, display the highest robustness across various terrains, adeptly navigating complex environments. FullEst policies, reacting strongly to unexpected obstacles, show lower robustness. Figure 7 presents snapshots of the robot traversing different terrains, with additional videos in supplementary materials. Overall, Key2 policies excel in tracking accuracy and real-world robustness.

IV. CONCLUSION

In this work, we quantitatively inspected the importance of learned explicit estimations and evaluated the locomotion performance of different estimation designs. Among all of the estimated states, velocity emerges as the paramount factor, with heightmap ranking second. Policies equipped with velocity estimation exhibit enhanced locomotion capabilities, particularly evident in the precision of velocity

Fig. 7: Snapshots of Wukong-IV humanoid traversing different terrains. (A) 3-step 10cm stairs. (B) 25 deg slope. (C) Indoor terrain with aluminum profiles. (D) Natural grass field. Check the supplementary video for more experiments.

tracking and the attainment of maximum speeds in both forward directions. Incorporating heightmap estimation bolsters adaptability to complex terrains, albeit with a lesser impact compared to sole velocity estimation. Regarding physical transferability, implicit encoding encompasses information not covered by explicit estimation, thereby enhancing policy adaptability during transitions from simulation to real-world environments.

There are some limits to be further studied. We have not considered the potentially complex effects of upper body movements on robots. If arm movements are substantial or involve carrying loads, inertial information might also influence performance. This inertial information may also require estimation. Besides, this policy is blind and doesn't include perception information. This work primarily focuses on the software and hardware configuration of the Wukong 4 robot, we acknowledge that additional information may indeed influence the distribution of the importance of state estimation.

ACKNOWLEDGMENT

We are grateful to have Shixin Luo, Songbo Li, Shuaichen Zhang, and Guanxun Lang from the Robotics and Machine Intelligence Lab, Zhejiang University to help with real robot experiments. We also thank Zhiyong Tang, Kai Xu, and Xueyin Zhang from DeepRobotics Inc. for maintaining and repairing the robot hardware. Last but not least, we would like to express our gratitude to Alex Zhibin Li from University College London for the inspiration and suggestions about real robot deployment.

APPENDIX

TABLE II: Randomization Parameters

Parameter	Range	Unit
Base CoM position	[-0.15,0.15]	m
Base load mass	[-2.0, 12.5]	kg
Friction rate	[0.25,1.25]	-
Motor strength	[0.8,1.2]	-
Kp factor	[0.9,1.1]	-
Kd factor	[0.9,1.1]	-
latency	[0,2]	network step

REFERENCES

[1] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.
[2] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE transactions on automatic control*, vol. 48, no. 1, pp. 42–56, 2003.
[3] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.
[4] J. Reher, W.-L. Ma, and A. D. Ames, "Dynamic walking with compliance on a cassie bipedal robot," in *2019 18th European Control Conference (ECC)*, 2019, pp. 2589–2595.

TABLE III: Hyperparameters for PPO and neural network

Parameter	Value
Number of environments	4096
Learning epochs	4
Initial learning rate	5e-4
Gamma	0.996
Lamda	0.95
Number of batches	4
Backbone hidden layers	[2048, 512, 128]
Encoder hidden layers	[1024, 256, 64]
Activation function	ELU
Velocity loss coefficient	1
Heightmap loss coefficient	0.5
Body height loss coefficient	2
VAE β	50
Prediction loss coefficient	2

[5] J.-K. Huang and J. W. Grizzle, "Efficient anytime clf reactive planning system for a bipedal robot on undulating terrain," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2093–2110, 2023.

[6] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *2019 American Control Conference (ACC)*, 2019, pp. 4559–4566.

[7] G. Gibson, O. Dosunmu-Ogunbi, Y. Gong, and J. Grizzle, "Terrain-adaptive, alip-based bipedal locomotion controller via model predictive control and virtual constraints," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6724–6731.

[8] B. Dynamics, "Picking up momentum," 2023. [Online]. Available: https://www.bostondynamics.com/resources/blog/picking-momentum

[9] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, Jan 2019.

[10] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[11] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[12] G. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," in *Robotics: Science and Systems (RSS)*, 2022.

[13] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *6th Annual Conference on Robot Learning (CoRL)*, 2022. [Online]. Available: https://openreview.net/forum?id=Re3NjSwf0WF

[14] A. Loquercio, A. Kumar, and J. Malik, "Learning visual locomotion with cross-modal supervision," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7295–7302.

[15] J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-real learning of all common bipedal gaits via periodic reward composition," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 7309–7315.

[16] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," in *Robotics: Science and Systems (RSS)*, ser. Robotics - Science and Systems, 2021.

[17] H. Duan, A. Malik, M. S. Gadde, J. Dao, A. Fern, and J. Hurst, "Learning dynamic bipedal walking across stepping stones," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ser. IEEE International Conference on Intelligent Robots and Systems, 2022, pp. 6746–6752.

[18] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Learning humanoid locomotion with transformers," *arXiv:2303.03381*, 2023.

[19] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel, "Model-based reinforcement learning via meta-policy optimization," in *Conference on Robot Learning*. PMLR, 2018, pp. 617–629.

[20] Z. Wang, W. Wei, A. Xie, Y. Zhang, J. Wu, and Q. Zhu, "Hybrid bipedal locomotion based on reinforcement learning and heuristics," *MICROMACHINES*, vol. 13, no. 10, OCT 2022.

[21] R. Batke, F. Yu, J. Dao, J. Hurst, R. L. Hatton, A. Fern, and K. Green, "Optimizing bipedal maneuvers of single rigid-body models for reinforcement learning," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 714–721.

[22] Z. Yoon, J.-H. Kim, and H.-W. Park, "Invariant smoother for legged robot state estimation with dynamic contact event information," *IEEE Transactions on Robotics*, vol. 40, pp. 193–212, 2024.

[23] S. Teng, M. W. Mueller, and K. Sreenath, "Legged robot state estimation in slippery environments using invariant extended kalman filter with velocity update," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3104–3110.

[24] Y. You, S. Cheong, T. P. Chen, Y. Chen, K. Zhang, C. Acar, F. L. Lai, A. H. Adiwahono, and K. P. Tee, "State estimation for hybrid wheeled-legged robots performing mobile manipulation tasks," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3019–3025.

[25] S. Fahmi, G. Fink, and C. Semini, "On state estimation for legged locomotion over soft terrain," *IEEE Sensors Letters*, vol. 5, no. 1, pp. 1–4, 2021.

[26] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: rapid motor adaptation for legged robots," in *Robotics: Science and Systems (RSS)*, 2021.

[27] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4630–4637, 2022.

[28] I. M. A. Nahrendra, B. Yu, and H. Myung, "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5078–5084.

[29] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," *arXiv preprint arXiv:2309.14341*, 2023.

[30] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li, "Multi-expert learning of adaptive legged locomotion," *Science Robotics*, vol. 5, no. 49, p. eabb2174, 2020.

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

[32] W. Wei, Z. Wang, A. Xie, J. Wu, R. Xiong, and Q. Zhu, "Learning gait-conditioned bipedal locomotion with motor adaptation*," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, 2023, pp. 1–7.

[33] J. Bastien and L. Birglen, "Power efficient design a compliant robotic leg based on klann's linkage," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 2, pp. 814–824, 2023.

[34] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *33rd International Conference on Neural Information Processing Systems (NIPS)*. Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 1–12.

[36] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3319–3328.

[37] W. Yu, C. Yang, C. McGreavy, E. Triantafyllidis, G. Bellegarda, M. Shafiee, A. J. Ijspeert, and Z. Li, "Identifying important sensory feedback for learning locomotion skills," *Nature Machine Intelligence*, vol. 5, no. 8, pp. 2522–5839, AUG 2023.