Preference Aligned Diffusion Planner for Quadrupedal Locomotion Control

Xinyi Yuan*[1], Zhiwei Shang*[1], Zifan Wang[1], Chenkai Wang[3], Zhao Shan[2], Zhenchao Qi[1], Meixin Zhu†[1], Chenjia Bai†[2], Xuelong Li[2], *Fellow, IEEE*

***Abstract*— Diffusion models demonstrate superior performance in capturing complex distributions from large-scale datasets, providing a promising solution for quadrupedal locomotion control. However, offline policy is sensitive to Out-of-Distribution (OOD) states due to the limited state coverage in the datasets. In this work, we propose a two-stage learning framework combining offline learning and online preference alignment for legged locomotion control. Through the offline stage, the diffusion planner learns the joint distribution of state-action sequences from expert datasets without using reward labels. Subsequently, we perform the online interaction in the simulation environment based on the trained offline planer, which significantly addresses the OOD issues and improves the robustness. Specifically, we propose a novel *weak* preference labeling method without the ground-truth reward or human preferences. The proposed method exhibits superior stability and velocity tracking accuracy in pacing, trotting, and bounding gait under both slow- and high-speed scenarios and can perform zero-shot transfer to the real Unitree Go1 robots. The project website for this paper is at `https://shangjaven.github.io/preference-aligned-diffusion-legged/`**

I. INTRODUCTION

Learning-based approaches significantly enhance the agility and adaptability of quadrupedal robots to accomplish diverse locomotion tasks [1], [2]. While online learning demonstrates robustness in complex dynamic environments, extensive trial-and-error interactions in simulation are required to learn an effective policy. Thus, learning an online policy can be sample inefficient and requires a meticulously designed reward function. In contrast, offline learning can leverage the advantages of pre-collected offline datasets via model-based controller [3], animal imitation [4], or Reinforcement Learning (RL) policy [5], significantly improving data efficiency and reducing the cost of online interactions [6]. In offline policy learning, diffusion models [7] have shown superior performance in capturing complex action distributions from offline trajectories [8], which is promising to solve quadruped locomotion tasks with high-dimensional action space and complex action distribution in various terrains. As an example, DiffuseLoco [9] has recently proposed to train a diffusion planner for quadrupedal locomotion from offline trajectories.

*Equal Contribution.

†Corresponding Authors: Meixin Zhu (e-mail: meixin@ust.hk), Chenjia Bai (e-mail: baichenjia255@gmail.com)

[1] Intelligent Transportation Thrust, System Hub, Hong Kong University of Science and Technology (Guangzhou)

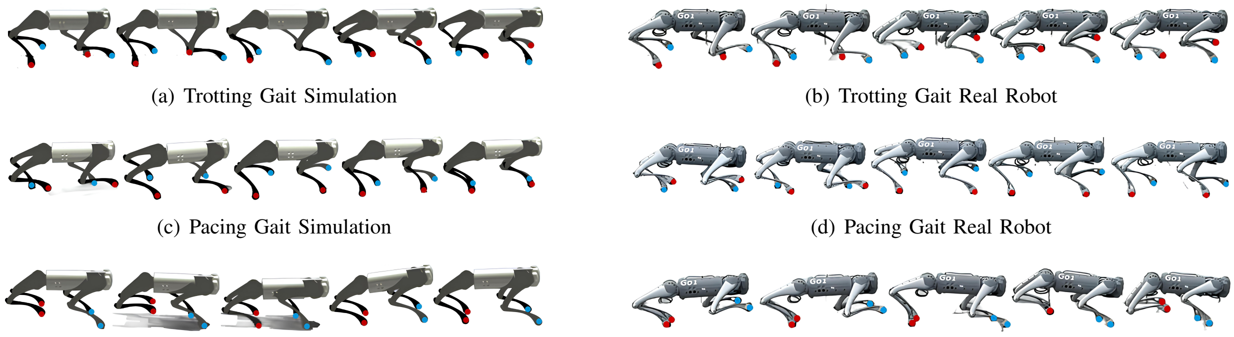
[2] Institute of Artificial Intelligence (TeleAI), China Telecom

[3] Department of Statistics and Data Science, Southern University of Science and Technology

However, learning robust locomotion planners in a purely offline manner requires collecting a large-scale dataset with wide state coverage [10], especially covering situations that the robot may encounter in the real world. This is especially challenging in quadrupedal locomotion since the robot has stochastic dynamics, and the external environment also has noises or disturbances. Therefore, it is challenging to learn a robust locomotion policy using the offline dataset with limited state coverage, as the learned policy may be sensitive to out-of-distribution (OOD) states in the real world. To address such problems, it is desirable to combine offline diffusion modeling with online interaction, where the collecting of online trajectories mitigates the effects of OOD samples. Dagger [11] is a traditional method that collects new samples in interaction but requires real-time action labels from the expert, which can be resource-intensive. Alternatively, Diffusion-QL [12] performs policy improvement for the diffusion policy while it requires reward labels for online transitions. As a result, our work aims to integrate online interactions after offline learning to address the OOD generalization problem for diffusion planners, providing effective policy refinement without the requirement of external expert labels or reward functions.

In this paper, we propose a two-stage learning framework combining offline and online learning for legged locomotion. In the offline learning stage, the diffusion planner learns the joint distribution of state-action sequence from an offline dataset collected by other policies and does not use reward labels in training. Then, we perform online interactions with the environment based on the diffusion planner, mitigating the effects of OOD samples and improving robustness via preference alignment. Specifically, we propose a novel online fine-tuning algorithm for the diffusion planner based on preferences, which resembles Direct Preference Optimization (DPO) [13] for diffusion models in text-to-image generation [14]. Importantly, the preference score in our method is measured by distances between the states and the nearest neighbors of expert trajectories, which signifies the optimality of a trajectory and is used as the metric to construct preference pairs. As a result, the preference data can be easily constructed from such a *weak* preference label without the ground-truth reward function or human preferences.

The contributions are summarized as follows: (i) We propose a novel two-stage learning framework that combines online interaction and offline diffusion learning to address the OOD issue; (ii) We give an efficient preference alignment algorithm for offline diffusion planner via DPO and weak



(a) Trotting Gait Simulation
(b) Trotting Gait Real Robot
(c) Pacing Gait Simulation
(d) Pacing Gait Real Robot
(e) Bounding Gait Simulation
(f) Bounding Gait Real Robot

Fig. 1. Video Frames in Simulation and Real World of the Proposed Architecture: (a-b) Trotting gait simulation and real-world test, (c-d) Pacing gait simulation and real-world test, (e-f) Bounding gait simulation and real-world test

preference labels; (iii) The resulting diffusion planner exhibits superior performance on stability and velocity tracking accuracy in simulation and can be zero-shot transferred to the real-world Unitree robot.

II. Related Work

A. Learning-Based Approaches for Legged Locomotion

For legged locomotion, learning-based methods automatically capture dynamic behaviors from interacting experiences, largely reducing the need for manual expertise in classical control [15]. Online RL has been widely applied in learning complex locomotion skills in simulation, and adaptive techniques like domain randomization are employed to facilitate the transfer from the simulation to real robot [16], [17], [18], [19], [20]. However, manually designing reward functions and tuning weights can be particularly challenging when dealing with complex tasks. Other methods [21] adopt imitation learning to extract agile locomotion strategies from real-world animal reference motions, while it requires the motion-capture dataset that is more expensive [22]. Recent studies utilized offline learning on locomotion control with the limited scope confined to gym simulation environments [23], [24]. While DiffuseLoco [9] steps further to introduce a diffusion-based locomotion planner for legged robots, the OOD generalization problem exists due to the limited coverage of datasets.

B. Diffusion Models in Robotics

Diffusion models have demonstrated superior generative capabilities in various robotics tasks such as robotic navigation [25], manipulation [26], [27], and decision-making [28]. For example, Diffuser [29] is a planner that analogizes the planning to the denoising process in diffusion models, demonstrating impressive adaptability in complex long-horizon manipulation tasks. Recent studies [30], [31] represent robot policies also as the diffusion process, where the policy generates joint actions based on multi-modal conditional inputs such as observations or visual information. However, most existing research has been limited to high-level tasks with low-dimensional action spaces, leaving room for exploration in more complex, high-dimensional scenarios such as legged locomotion.

III. Preliminaries

A. DDPM

The proposed framework uses *Denoising Diffusion Probabilistic Models* (DDPM) [32] with U-net and Transformer backbone. DDPM starts by adding Gaussian noise to the original data stepwise and uses the neural network to learn the inverse denoising process. During the forward process, Gaussian noise will be added gradually to the original data:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t I), \tag{1}$$

where β_t is the variance scheduler, $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ are samples from two adjacent diffusion steps.

During the reverse chain, the model starts from the pure Gaussian noise $\mathbf{x}_T$ and gradually extracts the noise to derive $\mathbf{x}_{T-1}$ ... $\mathbf{x}_0$. DDPM uses $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to approximate the conditional denoising distribution:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)). \tag{2}$$

For training, we use the simplified objective without the weighting term [32] as

$$\mathcal{L}_t^{\text{simple}}(\theta) = \mathbb{E}_{\mathbf{x}_0,\epsilon,t}\Big[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\Big]. \tag{3}$$

IV. Method

A. System Design

The architecture of the proposed framework is illustrated in Fig. 2. The entire system consists of four stages. Firstly, preference-free offline datasets (i.e., $\mathcal{D}$) containing various gait patterns are collected in the *Walk-These-Ways* environment. Secondly, behavior cloning is performed by extracting gait-specific diffusion policies from these datasets. Following this, the diffusion policy is fine-tuned using the direct preference optimization method on the constructed preference dataset (i.e., $\mathcal{D}_{\text{pref}}$). Finally, the latest model will be deployed on the Unitree Go1 robot.

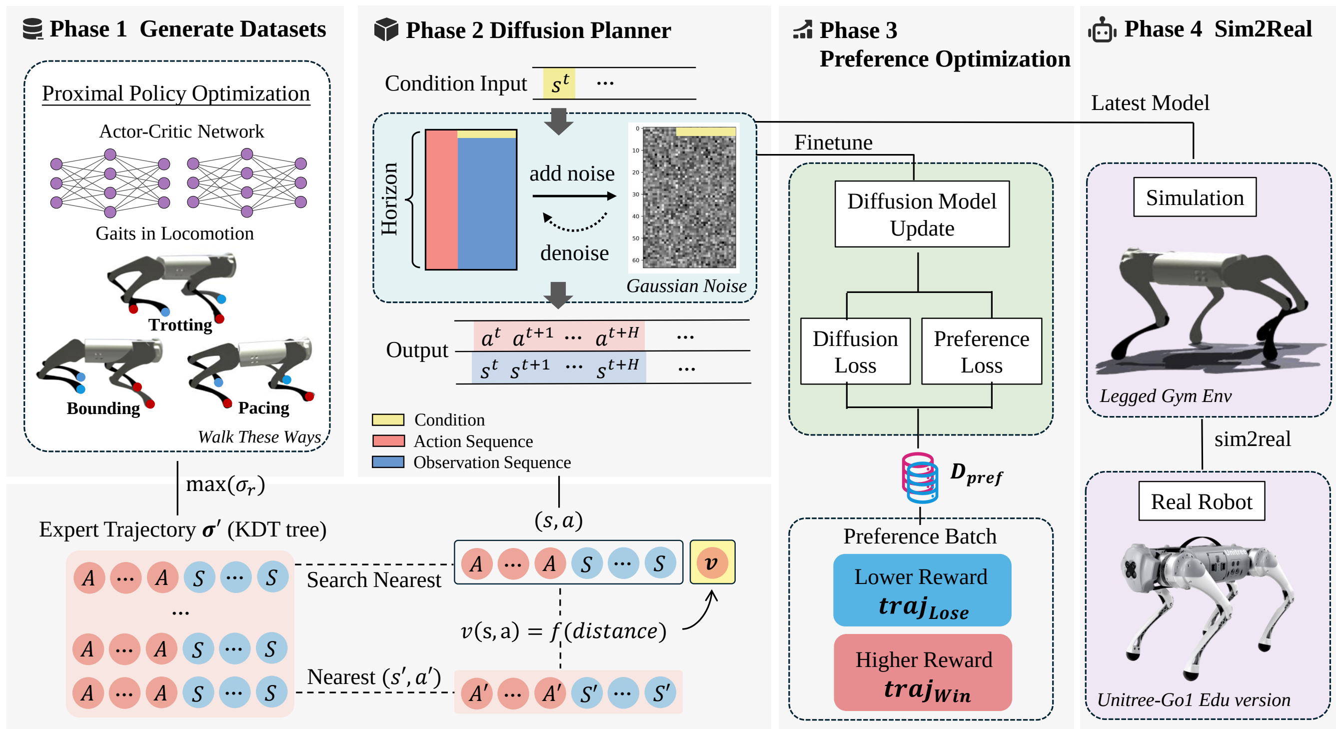


Fig. 2. Proposed Architecture Framework Overview: **(1) Generate Datasets**: the offline datasets among pacing, trotting, and bounding gait are collected through the expert PPO policy in the *walk-these-ways* task. **(2) Behavior Cloning**: given a condition input s^t, the diffusion policy can produce a sequence of states and actions. **(3) Preference Alignment**: Conduct the preference alignment on the offline diffusion planner based on proposed *weak* preference labels. **(4) Sim2Real**: The refined policy is deployed on the Unitree Go1 robot.

B. Offline Dataset Generation

This work utilizes *Walk These Ways* framework [16] as the source locomotion policy and collects offline datasets for the following gaits: pacing, trotting, and bounding. For each gait, we roll out 2048 episodes with 250 time steps. The data format as $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \mathbf{r}_t, \mathbf{d}_t)$, where the $\mathbf{d}_t$ turns True when the episode ends. The behavior policy is denoted as $\pi(\mathbf{a}_t|\mathbf{c}_t, \mathbf{b}_t)$, where $\mathbf{c}_t$ includes commands, and $\mathbf{b}_t$ indicates the behavior parameters. The reward functions of the locomotion controller include task reward (tracking the command speed), augmented auxiliary (task behavior related), and fixed auxiliary (promoting robot stability).

The Unitree Go1 robot possesses 12 degrees of freedom, with each leg comprising 3 degrees of freedom, corresponding to the hip, thigh, and calf joints. The observation space $\mathbf{o}_t$ of offline diffusion planner consists of the state vector $s_t = \{\mathbf{v}_t^{cmd}, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{t}_t\} \in \mathbb{R}^{31}$. Specifically, $\mathbf{v}_t^{cmd} \in \mathbb{R}^3$ incorporates the velocity command along x, y and z axis, $\mathbf{q}_t \in \mathbb{R}^{12}$ represent the joint positions, $\dot{\mathbf{q}}_t \in \mathbb{R}^{12}$ represent the joint velocities, $\mathbf{t}_t \in \mathbb{R}^4$ is the gait and timing related parameter. Each feature of the observations is normalized by Gaussian distribution before input into the diffusion models.

The action space $A_t \in \mathbb{R}^{12}$ represents each joint's position targets. We follow the actuator network [2] for actions-to-torques mapping. The legged robot will then execute the resultant torque.

C. Offline Diffusion Planner

We adopt a condition diffusion model to train an offline diffusion planner for locomotion. The conditional input contains the current observation state s_0 (ensuring the generated trajectory starts from the current state), and the output is the action sequence $a_{0:h}$ of length horizon. During the training process, the data loader will randomly sample the normalized trajectory segment from the offline dataset with batch size N, and then Gaussian noise will be added iteratively into the trajectories. Subsequently, the diffusion model will denoise the noisy trajectory to reconstruct the original data. The diffusion loss measures the Mean-Squared Error (MSE) between the true noise and predicted noise:

$$\mathcal{L} = MSE(\epsilon_k, \epsilon_\theta(a_k^{t:t+h}, s_0^t, k)) \tag{4}$$

Notably, ϵ_θ is the noise predictor, the superscript represents the time step in the trajectory (start from time t with the horizon h), while the subscript denotes the diffusion denoising iterations (k). Besides, a fixed mask will be applied on the loss to ignore the deviations in the observation condition (s_0).

During the inference, the Isaac Gym environment will be reset initially, and the current observation serves as the condition input for trajectory generation. Then, the trained diffusion model will denoise the pure Gaussian noise to derive the action sequence. Specifically, Classifier-Free Guidance (CFG) [33] is employed to blend the conditioned and unconditioned predictions with the weight w:

$$\bar{\epsilon}_\theta(\mathbf{x}_t, t, y) = (1 + w)\epsilon_\theta(\mathbf{x}_t, t, y) - w\epsilon_\theta(\mathbf{x}_t, t) \tag{5}$$

where $\bar{\epsilon}_\theta$ is the predictor under classifier-free guidance, y represent the conditional information.

After generating the output action sequence of length horizon (h), the robot will execute every action stepwise and record the state transitions. The inference and interaction will continue until the episode length reaches the pre-defined maximum length. To ensure real-time performance, we decrease the diffusion sampling step in inference to 10 steps without sacrificing the locomotion performance. Besides, since the whole generated action sequence will be executed, the required inference frequency within each episode will be reduced compared with only implementing the single action.

D. Preference Optimization

To mitigate the OOD issue in offline learning, we propose the preference alignment method to fine-tune the offline diffusion planner. Firstly, we roll out preference-free datasets $\mathcal{D}$ from the pre-trained offline diffusion planner. When constructing the preference dataset $\mathcal{D}_{\text{pref}}$, we randomly sample two segments from $\mathcal{D}$ without replacement and then assign preference labels based on the following method.

Considering that the ground-truth rewards are sometimes challenging to obtain, we propose a *weak* preference labeling method that allows for constructing preference labels without the reward labels, requiring only few expert trajectories. The optimal expert trajectory is selected from the given expert trajectories based on their cumulative reward. For each state-action pair (s, a) in the preference-free dataset $\mathcal{D}$, we search for the closest state-action pair (s', a') in the *optimal* expert trajectory and calculate their Euclidean distance. The value (v) of specific (s, a) is calculated as the equation:

$$v_t = \exp\left(-\frac{\beta \times d_t}{|\mathcal{A}|}\right) \tag{6}$$

where β is a hyperparameter with a value of 0.5, d_t represents the calculated Euclidean distance, $|\mathcal{A}|$ is the dimension of the action space. Notably, a similar technique was applied in [34]. However, we have modified this approach by querying only based on the current state and action.

Finally, given two trajectories σ_1 and σ_2, we compute the corresponding values for each state-action pair within them. The segment with a higher cumulative *value* will be determined as the winning segment (σ_{winning}), as inferred by the following equation:

$$\sigma^1_{\text{winning}} = \arg\max_{\sigma \in \{\sigma_1, \sigma_2\}} \left(\sum_{t=0}^{h} v_t^{(1)}, \sum_{t=0}^{h} v_t^{(2)}\right) \tag{7}$$

To demonstrate the effectiveness of our proposed *weak* preference, we provide the alternative reward-available Preference Label (*strong label*) for experiment comparison. Suppose the reward label can be obtained in the environment, and the segment with higher *cumulative reward* will be preferred, indicated in the following equation:

$$\sigma^2_{\text{winning}} = \arg\max_{\sigma \in \{\sigma_1, \sigma_2\}} \left(\sum_{t=0}^{h} r_t^{(1)}, \sum_{t=0}^{h} r_t^{(2)}\right) \tag{8}$$

According to the Bradley-Terry model, the preference model can be described as (here σ_{winning} is abbreviated as σ^+, the σ_{losing} is abbreviated as σ^-):

$$P(\sigma^+ > \sigma^-) = \text{Sigmoid}(\text{r}(\sigma^+) - \text{r}(\sigma^-)) \tag{9}$$

The proposed loss function during the preference alignment incorporates two components: the preference loss ($\mathcal{L}_{\text{DPO}}$) derived from DPO and the regularization term. The preference loss amplifies the difference between the "winning segment" and the "losing segment," thus improving the diffusion policy's performance to align with the preference labels and generate more preferred samples. More importantly, the regularization term helps to avoid significant deviations from the original policy and thus mitigate the OOD issue. The loss function for the preference alignment stage is as follows:

$$\mathcal{L}(\epsilon_\theta, \mathcal{D}_{\text{pref}}) = \mathcal{L}_{\text{DPO}}(\epsilon_\theta, \mathcal{D}_{\text{pref}}) - \mu \mathbb{E}_{\sigma \in \mathcal{D}} \log \epsilon_\theta(\sigma) \tag{10}$$

practical approximation and detailed proof can be found in [35].

V. Experimental Results

The proposed framework utilizes the CleanDiffuser [36] library to implement the diffusion model. The modularized design allows easy switching between different backbones and other customization. We conduct the classic velocity tracking task in the simulation environment with different gaits to quantify the performance of our proposed method. During the experiment, we choose $v = 0.5m/s$ to represent slow speed and $v = 1.0m/s$ to represent high speed.

For evaluation metrics, we consider the stability and average x-axis speed. We assess an episode as stable when the quadruped robot does not fall during all 250 steps. Additionally, the measured velocity along the x-axis is recorded. For every experimental parameter setting, we collect 1024 episodes for each seed with a total of three random seeds. Table I lists the hyperparameters used in the experiments.

TABLE I

Hyperparameters in Training, Inference, and Preference Alignment

Stage	Hyperparameter	Value
Offline Diffusion Planner	Batch size	64
	Horizon	64
	Solver	DDPM
	Diffusion steps	20
	Action loss weight	10.0
Inference	w_cg	0.0001
	Sampling steps	10
Preference Alignment	Regularization weight	1.0
	Bias	0
	Temperature	500

We chose three strong baselines for the experiments. The first one is **Conservative Q-learning (CQL)**. CQL mitigates the Q-value overestimation problem by introducing the regularization term in conservative Q-value estimation. CQL

TABLE II

RESULTS COMPARISON BETWEEN THE PROPOSED METHOD AND BASELINES

Gaits	Metrics	CQL	DDPM-Unet	DDPM-Transformer	*Weak-Preference* Aligned Planner	Strong-Preference Aligned Planner
Slow Pacing (0.5 m/s)	Stability	\	40.9	72.0	81.6	**87.3**
	x Velocity	fail	0.60	0.43	0.46	**0.49**
Slow Trotting (0.5 m/s)	Stability	\	39.5	58.7	78.5	**85.2**
	x Velocity	fail	0.59	0.36	0.42	**0.50**
Slow Bounding (0.5 m/s)	Stability	\	40.4	60.1	70.8	**72.4**
	x Velocity	fail	0.59	0.21	0.44	**0.48**
Quick Pacing (1.0 m/s)	Stability	\	77.4	81.4	91.6	**91.8**
	x Velocity	fail	0.49	0.62	0.64	**0.73**
Quick Trotting (1.0 m/s)	Stability	\	58.9	66.6	84.4	**89.5**
	x Velocity	fail	0.48	0.61	0.63	**0.67**
Quick Bounding (1.0 m/s)	Stability	\	32.4	46.6	84.3	**89.2**
	x Velocity	fail	0.42	0.78	0.72	**0.80**

is effective in high-dimensional state space and complex environments. The second and third baselines are offline diffusion planners with U-Net backbone and transformer backbone, respectively. They are noted as **DDPM-Unet** and **DDPM-Transformer**.

All experiments were performed on a high-performance server, equipped with an Intel Xeon Gold-6248R CPU, an NVIDIA GeForce RTX-A5000 GPU, and 256 GB of RAM, running Ubuntu 20.04 OS. Such a configuration ensured that the experiments could be executed efficiently and without computational bottlenecks.

A. *Performance of Preference-aligned Diffusion Planner*

Table II presents a comparison of the proposed method with baselines across different gaits (pacing, trotting, bounding) and speeds (0.5 m/s and 1.0 m/s) in terms of stability and average x-axis velocity.

CQL consistently failed all the locomotion tasks. We observed in the simulation environment that the quadruped robot either remained stationary on the ground or exhibited subtle irregular jitters. This indicates that CQL struggles with continuous control in complex locomotion tasks.

Stability Performance: The proposed preference-aligned diffusion planner outperformed all baselines in stability across all locomotion tasks. Specifically, for the quick bounding task, the proposed *weak* preference-aligned planer achieves 84.3% stability, much higher than 32.4% of the DDPM-Unet and 46.6% of the DDPM-Tranformer. Furthermore, given that the proposed policy utilizes the transformer backbone in the offline learning stage, the stability increased by 37.7% after the preference alignment. Additionally, in the slow trotting task, the stability increased by 19.8% relative to the offline DDPM-Transformer model. These results indicate that our preference-aligned planner exhibits superior stability performance across high-speed and slow-speed scenarios.

The improvement in stability from the offline diffusion planner can be attributed to the preference alignment stage, which reduces sensitivity to OOD states and better aligns the diffusion planner with real-world state distributions. Although the offline diffusion planner sometimes can generate reasonable action sequences (proper gait and no fall in an episode), it is inadequate for handling OOD states caused by slight deviations in action sequences in the simulation environment. This will further lead to severe deviated action predictions and significant cumulative error.

Velocity Tracking Performance: We evaluate the velocity tracking performance by the difference between the average measured x-axis velocity and the target velocity, and a more minor deviation indicates more accurate tracking and better velocity control. For example, in the quick bounding task, our proposed planner achieves an average velocity of 0.72 m/s, closely approaching the target speed of 1.0 m/s, significantly outperforming DDPM-Unet (0.42 m/s). In the slow-speed task, the proposed method demonstrates more precise velocity control with minor deviation from the target speed. For example, in the slow trotting task, our method achieved an average velocity of 0.42 m/s, closing matching the target and surpassing DDPM-Unet (0.59 m/s) and DDPM-Transformer (0.36 m/s).

To further present how our proposed method improves the velocity tracking performance, we conduct a case study based on the most challenging slow bounding task with the lowest stability in Fig. 3.

Fig. 3 depicts the velocity tracking process within an episode; the real-time velocities are smoothed with a moving average filter. The DDPM-Transformer exhibits a smaller standard deviation while it still deviates from the target velocity. Additionally, DDPM-Unet shows high fluctuation in measured velocity and fails to track the desired velocity. In contrast, the proposed preference-aligned diffusion planner (red line) reaches and maintains close to the target velocity.

In conclusion, the experimental result demonstrates that the proposed method outperforms existing baselines on stability and velocity tracking performance. The quantitative analysis supports the effectiveness and robustness of our proposed two-stage learning framework in challenging locomotion tasks.

TABLE III

ABLATION STUDIES ON PREFERENCE ALIGNMENT

		Speed = 1.0 m/s			**Speed = 0.5 m/s**		
		Pacing	Trotting	Bounding	Pacing	Trotting	Bounding
Preference Number	1024 episodes (*0.5)	89.6	78.8	75.3	69.5	73.0	63.5
	2048 episodes (*1.0)	90.1	77.7	75.9	76.4	79.1	66.4
	3072 episodes (*1.5)	91.8	89.5	89.2	87.3	85.2	72.4
Preference Quality	*Weak* Preference Label	91.6	84.4	84.3	81.6	78.5	70.8
	Strong Preference Label	91.8	89.5	89.2	87.3	85.2	72.4
Regularization	Without Regularization	fail	fail	fail	fail	fail	fail
	With Regularization	91.8	89.5	89.2	87.3	85.2	72.4

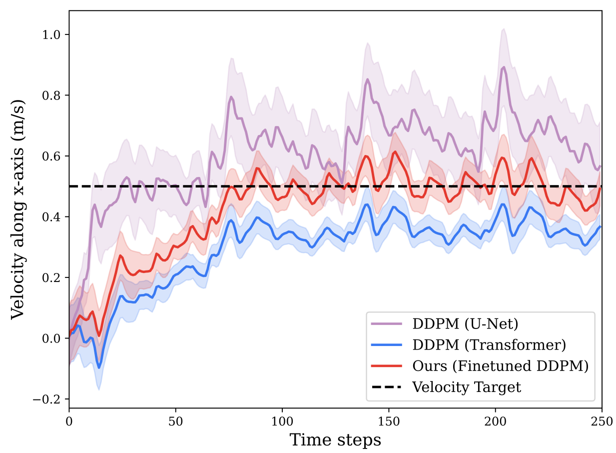


Fig. 3. Velocity tracking result in slow-bounding gait between models

B. Ablation Studies

The ablation studies in Table III investigate the impact of preference dataset size, preference label quality, and regularization methods on stability performance.

Preference Dataset Size: Increasing the preference dataset size from 1024 episodes to 3072 episodes improves the performance across all gaits and speeds. For example, in the slow-pacing task, the stability increased by 17.8%. Generally, the sensitivity to the size of the preference dataset varies on the difficulty of different locomotion tasks. In the simplest quick-pacing task, 1024 episodes of the preference dataset are sufficient to achieve relatively stable improvement. We observe in the simulation result that a smaller preference dataset may introduce the risk of insufficient coverage on state distribution, further influencing task performance. However, the overall results indicate our proposed preference alignment framework demonstrates stable improvement in the offline diffusion planner under limited preference data.

Preference Label Quality: We compare the performance between the reward-based preference label (strong preference) and our proposed reward-unavailable preference label (*weak* preference). The results in Table III indicate that weak preference labels can achieve strong performance across most locomotion tasks with slight differences compared with strong preference labels. For example, the performance of the weak preference label closely approaches the strong preference label in the quick-pacing and slow-bounding tasks. Notably, the preference alignment based on weak preference labels demonstrates significant improvement compared with the DDPM-Transformer on all locomotion tasks, indicating the effectiveness of the weak preference labeling method and proposed preference alignment framework.

Regularization Methods: The ablation results in Table III suggest the critical role of the proposed regularization term in the preference alignment stage. The CPOD-KL method eliminates the regularization term and solely emphasizes the difference between the winning segment (σ_{winning}) and the losing segment (σ_{losing}) under the current policy and reference policy. The results demonstrate that without the regularization term, the preference alignment will consistently cause *fail* in all locomotion tasks. In comparison, our proposed preference alignment method encourages the generation of winning segments while maintaining the overall likelihood of the winning and losing segments through the regularization term, thus effectively addressing the OOD issue.

VI. CONCLUSIONS

This paper presents a two-stage learning framework that integrates offline diffusion learning with online preference alignment to address the OOD issues. We leverage the offline diffusion planner to approximate the complex state-action sequences and further utilize the proposed *weak* preference label to conduct the preference alignment. Experiments indicate that our framework improved the stability and velocity tracking accuracy and can be deployed on Unitree Go1 robots. Future work can incorporate extra modalities, such as vision, into the framework to further enhance the generalization of the locomotion policy.

ACKNOWLEDGMENT

This study is supported by the National Natural Science Foundation of China under Grant 52302379 & 62306242; in part by Guangdong Provincial Natural Science Foundation-General Project with Grant 2024A1515011790; in part by Guangzhou Basic and Applied Basic Research Projects under Grants 2023A03J0106 and 2024A04J4290.

REFERENCES

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

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

[3] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini, "Mpc-based controller with terrain insight for dynamic legged locomotion," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2436–2442.

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

[5] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *Robotics: Science and Systems XVII*, 2021.

[6] A. Nie, Y. Flet-Berliac, D. Jordan, W. Steenbergen, and E. Brunskill, "Data-efficient pipeline for offline reinforcement learning with limited data," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 810–14 823, 2022.

[7] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[8] Z. Zhu, H. Zhao, H. He, Y. Zhong, S. Zhang, Y. Yu, and W. Zhang, "Diffusion models for reinforcement learning: A survey," *arXiv preprint arXiv:2311.01223*, 2023.

[9] X. Huang, Y. Chi, R. Wang, Z. Li, X. B. Peng, S. Shao, B. Nikolic, and K. Sreenath, "Diffuseloco: Real-time legged locomotion control with diffusion from offline datasets," *arXiv preprint arXiv:2404.19264*, 2024.

[10] J. D. Chang, M. Uehara, D. Sreenivas, R. Kidambi, and W. Sun, "Mitigating covariate shift in imitation learning via offline data with partial coverage," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=7PkfLkyLMRM

[11] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[12] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion policies as an expressive policy class for offline reinforcement learning," *arXiv preprint arXiv:2208.06193*, 2022.

[13] G. An, J. Lee, X. Zuo, N. Kosaka, K.-M. Kim, and H. O. Song, "Direct preference-based policy optimization without reward modeling," *Advances in Neural Information Processing Systems*, vol. 36, pp. 70 247–70 266, 2023.

[14] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik, "Diffusion model alignment using direct preference optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8228–8238.

[15] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Science Robotics*, vol. 9, no. 89, p. eadi9641, 2024.

[16] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning*. PMLR, 2023, pp. 22–31.

[17] X. B. Peng, G. Berseth, and M. Van de Panne, "Terrain-adaptive locomotion skills using deep reinforcement learning," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.

[18] L. Schneider, J. Frey, T. Miki, and M. Hutter, "Learning risk-aware quadrupedal locomotion using distributional reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 451–11 458.

[19] 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.

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

[21] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.

[22] L. Han, Q. Zhu, J. Sheng, C. Zhang, T. Li, Y. Zhang, H. Zhang, Y. Liu, C. Zhou, R. Zhao, *et al.*, "Lifelike agility and play in quadrupedal robots using reinforcement learning and generative pre-trained models," *Nature Machine Intelligence*, pp. 1–12, 2024.

[23] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review," *and Perspectives on Open Problems*, vol. 5, 2020.

[24] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," *arXiv preprint arXiv:1805.01954*, 2018.

[25] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 63–70.

[26] U. A. Mishra and Y. Chen, "Reorientdiff: Diffusion model based reorientation for object manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 867–10 873.

[27] P. M. Scheikl, N. Schreiber, C. Haas, N. Freymuth, G. Neumann, R. Lioutikov, and F. Mathis-Ullrich, "Movement primitive diffusion: Learning gentle robotic manipulation of deformable objects," *IEEE Robotics and Automation Letters*, 2024.

[28] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision-making?" *arXiv preprint arXiv:2211.15657*, 2022.

[29] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," *arXiv preprint arXiv:2205.09991*, 2022.

[30] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Z. Tan, I. Momennejad, K. Hofmann, *et al.*, "Imitating human behaviour with diffusion models," *arXiv preprint arXiv:2301.10677*, 2023.

[31] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[32] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[33] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[34] J. Lyu, X. Ma, L. Wan, R. Liu, X. Li, and Z. Lu, "Seabo: A simple search-based method for offline imitation learning," *arXiv preprint arXiv:2402.03807*, 2024.

[35] Z. Shan, C. Fan, S. Qiu, J. Shi, and C. Bai, "Forward kl regularized preference optimization for aligning diffusion policies," *arXiv preprint arXiv:2409.05622*, 2024.

[36] Z. Dong, Y. Yuan, J. Hao, F. Ni, Y. Ma, P. Li, and Y. Zheng, "Cleandiffuser: An easy-to-use modularized library for diffusion models in decision making," *arXiv preprint arXiv:2406.09509*, 2024.