Appendix

A Comparison Experiments

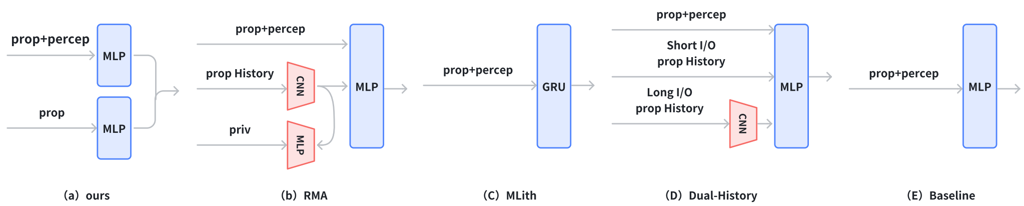

Figure 6: Comparative analysis of methods

To demonstrate our method's effectiveness, we compared it with RMA, MLith, Dual-History, and the baseline (Figure 6)[31, 20, 32]. Comprehensive experiments in complex environments show that while each method has strengths, ours excels in robustness, especially when perception fails. Our approach is superior in obstacle avoidance and climbing, where other methods often fail without external perception support, as shown in Table 3.

These results highlight our method's efficiency, applicability, and adaptability in real-world applications, enhancing robot autonomy and safety in dynamic environments.

Table 3: Performance Comparison

Method	Up Stair Success	Down Stair Success	Discrete Success	Stair XMD	Discrete XMD
Ours	97%	100%	90%	19.97	17.04
Ours w/o VAE	87%	100%	90%	16.42	14.99
MLith	0%	100%	84%	9.4	14.61
Dual-History	0%	100%	82%	10.9	13.77
Baseline	0%	100%	76%	7.8	11.53

B Ablation Studies

We conducted ablation experiments from multiple angles to examine the effectiveness of our policy in various aspects. The main ablation experiments we performed were:

- **Without VAE and cooperation regularization.**
- **Without pre-training the blind policy in the first stage.**
- **Our method with KL adaptive learning rate.**

The experimental results are shown in Figure 7. We found that both the VAE and our regularization term contribute to improving the final performance. Additionally, without the pre-trained model, training often fails, likely due to the difficulty in converging when training multi-agent systems. Moreover, this multi-agent training approach is very sensitive to the learning rate; an excessively high learning rate or adaptive adjustment of the learning rate can easily cause gradient explosion.

C Outdoors Experiments

We tested our controller across various outdoor terrains, which included actions such as climbing and dodging in complex terrains using perception, as well as navigating through grass, slopes, soft soil, and steps in cases where perception suddenly failed, as illustrated in Figure 8 and based on methodologies described by Li et al. [31].

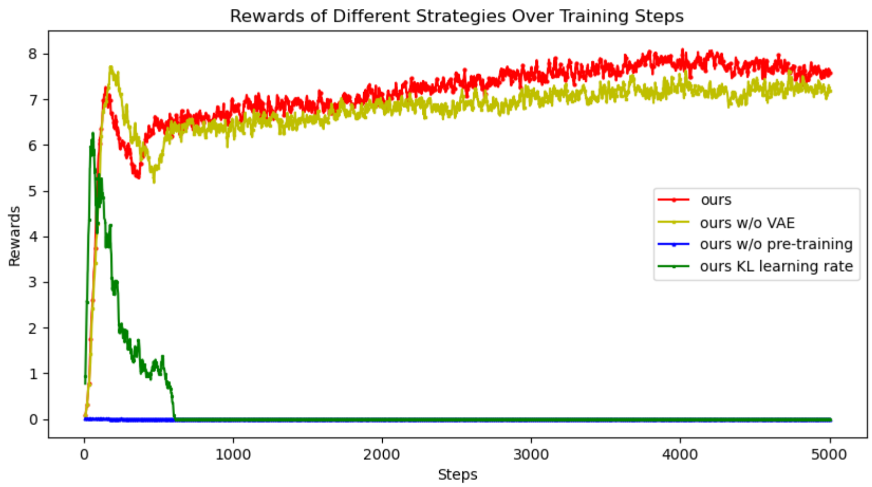

Figure 7: Rewards of Different Strategies Over Training Steps

Figure 8: Performance of Robot in Various Terrains.

D Reward Functions

We used the reward function as shown in Table 4, where the Task reward guides the robot to track the desired speed and complete motions on various terrains. Our setting for the regularization reward refers to Long et al. [33]; Kumar et al. [32];Agarwal et al. [20]; Cheng et al. [4]. Through extensive training trials, we optimized our reward weight settings to ensure that the robot moves in a relatively ideal manner.

Table 4: Reward Functions

Reward Type	Equation	Weight
Task Reward		
Linear Velocity Tracking	$\exp\left\{-\frac{\lVert v_{\text{xy}}^{\text{cmd}}-v_{\text{xy}}\rVert^2}{2\sigma}\right\}$	1.5
Angular Velocity Tracking	$\exp\left\{-\frac{(\omega_{\text{yaw}}^{\text{cmd}}-\omega_{yaw})^2}{\sigma}\right\}$	0.5
Linear Velocity Z	v_z^2	-1.0
Angular Velocity XY	$\omega_x^2+\omega_y^2$	-0.1
Regularization Reward		
Z Velocity	v_z^2	-1.0
X & Y Velocity	$\lVert\omega_{xy}\rVert_2^2$	-0.1
Orientation	$\lVert g\rVert_2^2$	-0.7
Dof Acceleration	$\sum_{i=1}^{12}\ddot{q}_i^2$	-1.5×10^{-7}
Collision	$r_{\text{Collision}}(7)$	-20.0
Action Rate	$\lVert a_t-a_{t-1}\rVert_2^2$	-0.11
Delta Torques	$\sum_{i=1}^{12}(\tau_t-\tau_{t-1})^2$	-1.0×10^{-7}
Torques	$\sum_{i=1}^{12}\tau_t^2$	-0.00001
Hip Position	$r_{\text{Pos}}(6)$	-0.8
Dof Error	$\sum_{i=1}^{12}(q-q_{\text{default}})^2$	-0.04
Feet Stumble	$\lvert F_{\text{feet}}^{\text{hor}}\rvert>4\times\lvert F_{\text{feet}}^{\text{ver}}\rvert$	-2
Termination	$-$	-5
Dof Position Limits	$\sum_{i=1}^{12}(q_i^{out},\ q_i>q_{\max}\vee q_i<q_{\min})$	-13.0

E Training Details

Robot Domain Randomizations: During the training process, we utilized the following domain randomization parameters to enhance the robustness of our policy. The range of randomization was referenced from Long et al. [33]; Wu et al. [34]. In actual robots, factors such as communication delays can lead to action execution delays of approximately 20ms. Therefore, domain randomization of action delays during robot training significantly improved the real-world performance of the robots.

Table 5: Robot Domain Randomizations

Parameter	Range [Min, Max]
Base Mass	[0,3] $\times$ default kg
CoM	[-0.2,0.2] $\times$ default m
Ground Friction	[0.6, 2.0]
Motor Strength	[0.8, 1.2] $\times$ default Nm
Joint Kp	[0.8, 1.2] $\times$ default
Joint Kd	[0.8, 1.2] $\times$ default
Initial Joint Positions	[0.5,1.5]×default
System Delay	[0,20] ms
Robot Pushing Interval	8s
Push Velocity XY	[0, 0.5]m/s

Heightmaps Domain Randomizations: We utilize the ‘Fast_lio’ odometer[35] and the method from P. Fankhauser and M. Hutter’s[24] to construct the elevation map. Due to inherent random errors typically associated with laser odometry in practical deployments, we have implemented domain randomization for both the elevation map and the z-axis height of the robot’s base.

Table 6: Heightmap Domain Randomizations

Parameter	Range [Min, Max]
Height map updates delay	100ms
Robot base Z Noisy	[-0.05,0.05] m
Height Gaussian Noisy	[-0.02, 0.02] m
Height Spike Noisy Proportion	5%
Height Spike Noisy	[0.1, 0.5]

Terrains Setting: We have designed a training environment containing six different types of terrains: slopes, stairs, discrete obstacles, pits, gaps, and pillars. The first three terrains are relatively easier for robot navigation, while the latter three require more reliance on external perception for anticipation.

- **Phase One: Blind Policy Training**

Table 7: Terrain Parameters and Proportion in Blind Policy Training

Terrain	Proportion	Parameters
Slope	30%	Inclination: [0, 40]
Stairs	60%	Step Height: [2cm, 15cm]
Discrete Obstacles	10%	Obstacle Height: [3cm, 18cm]

- **Phase Two: Advanced Perceptual Policy Training**

Table 8: Terrain Parameters and Proportion in Advanced Perceptual Policy Training

Terrain	Proportion	Parameters
Slope	10%	Inclination: [0, 40]
Stairs	60%	Step Height: [2cm, 15cm]
Complex Terrain	30%	Pit: [0.1m, 0.45m]; Gap: [0.15m, 0.45m]; Pillar: size [0.4m, 0.6m], center distance [1.6m, 1.4m]

Hyperparameters: Tables 9 and 10 list the hyperparameters used during our two-stage training process. It is important to note that multi-agent training, especially with MAPPO, is quite sensitive to hyperparameter settings, for which we referred to the settings recommended in Yu et al. [15]. We observed that the learning rate particularly impacts multi-agent training, where an excessively high learning rate can lead to issues such as gradient explosion.

- **Phase One: Blind Policy Training**

Table 9: PPO Parameters in Blind Policy Training

Parameter	Value
Discount factor	0.99
GAE discount factor	0.95
Timesteps per rollout	21
Epochs per Rollout	5
Minibatches per Epoch	4
Entropy Bonus	0.01
Value Loss Coefficient	1.0
Clip range	0.2
Learning rate	KL Adaptive Learning Rate
Desired KL Divergence	0.01
Environments	4096
Policy control frequency	50hz
PD controller frequency	200hz
Using history encoder frequency	20
Action Penalty Coefficient	0.1

- **Phase Two: Advanced Perceptual Policy Training**

Table 10: PPO Parameters in Advanced Perceptual Policy Training

Training Parameter	Blind Policy	Perceptive Policy
Discount factor	0.99	0.99
GAE discount factor	0.95	0.95
Timesteps per rollout	21	21
Epochs per Rollout	5	5
Minibatches per Epoch	4	4
Entropy Bonus	0.01	0.01
Value Loss Coefficient	1.0	1.0
Clip range	0.2	0.2
Learning rate	1×10^{-5}	1×10^{-4}
Environments	4096	4096
Using history encoder frequency	20	None
Action Penalty Coefficient	None	0.01

F Sim2Real Details

In sim2real deployment, our lidar and robot parameters, as shown in Table11, are based on configurations recommended by Agarwal et al. [20].

Table 11: Sim2real Parameters

Parameter	Value
Radar relative to base coordinates (xyz rpy)	[-0.33, 0, -0.35, -0.1, -0.55, 0]
Point cloud clipping height	[-0.5m, +0.5m]
Elevation map update frequency	50Hz
Other coefficients for elevation maps	size: 3m $\times$ 3m, resolution: 0.05m
Odometer update frequency	10Hz
Blind Policy frequency	50Hz (synchronized with Perceptive Policy)
Perceptive Policy frequency	50Hz (synchronized with Blind Policy)
PD controller frequency	1kHz
Joint Kp	40
Joint Kd	40