APPENDIX A
ADDITIONAL RESULTS AND ANALYSIS

A. Robustness Analysis against Perception Failures

A critical challenge in real-world deployment is the discrepancy between the perfect perception available in simulation and the imperfect sensing on the physical robot. To evaluate the policy's robustness against out-of-view and occlusion, we design a simulator-side evaluation protocol that explicitly mimics physical perception limits.

1) Simulation of Perception Limits: We model perception availability $m_t \in \{0, 1\}$ as the intersection of geometric constraints and temporal data losses. The simulator provides a ball measurement only when $m_t = 1$, defined by:

$$m_t = m_t^{\text{fov}} \cdot m_t^{\text{GE}}. \tag{17}$$

Geometric Occlusion: We align the simulated sensor with the fisheye camera used in deployment. Let $\mathbf{p}_t^{\text{ball}} = [x_t, y_t]^\top$ denotes the ball position in robot's base frame. A measurement is valid ($m_t^{\text{fov}} = 1$) only if the ball falls within the effective field-of-view (FOV) cone defined by a half-angle $\theta_{\max}$ and a maximum reliable depth $d_{\max}$:

$$m_t^{\text{fov}} = \mathbb{I}\left(\frac{x_t}{\|\mathbf{p}_t^{\text{ball}}\|} \geq \cos\theta_{\max}\right) \cdot \mathbb{I}\left(\|\mathbf{p}_t^{\text{ball}}\| \leq d_{\max}\right), \tag{18}$$

where $\mathbb{I}(\cdot)$ is the indicator function. In our setup, we set $\theta_{\max} = 120°$ and $d_{\max} = 3.0\,\text{m}$.

Temporal Burst Loss: To capture bursty failures caused by motion blur or communication delays (which are temporally correlated), we employ a Gilbert-Elliott (GE) model. This implies a two-state Markov chain $z_t \in \{\text{G}, \text{B}\}$ with transition probabilities:

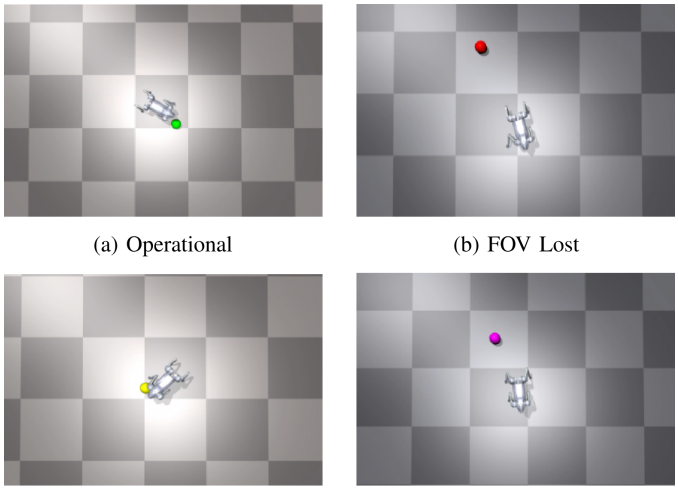
$$\begin{aligned} Pr(z_{t+1} = \text{B} \mid z_t = \text{G}) &= p_{\text{GB}}, \\ Pr(z_{t+1} = \text{G} \mid z_t = \text{B}) &= p_{\text{BG}}. \end{aligned} \tag{19}$$

The temporal mask is then defined as $m_t^{\text{GE}} = \mathbb{I}(z_t = \text{G})$. This formulation allows us to simulate realistic, consecutive frame drops rather than independent noise.

2) Predictive State Estimation: To bridge the gaps created by these simulated failures, we implement the same Kalman Filter (KF) used in the real robot following [6]. When $m_t = 0$, the measurement update is skipped, and the policy relies on the KF's constant-velocity prediction.

3) Evaluation Results and Visualization: We conduct stress tests where the robot commands are generated by a randomly operated joystick while subject to the aforementioned perception limits. As shown in Fig. 9 and the accompanying video, we visualize the perception status using color-coded ball to verify the system's behavior under different failure modes:

- **Green (Operational):** Both geometric and temporal checks pass ($m_t^{\text{GE}} = 1, m_t^{\text{fov}} = 1$). The policy receives the corrected observation.
- **Red (FOV Lost):** The ball moves out of the camera view ($m_t^{\text{GE}} = 1, m_t^{\text{fov}} = 0$). This tests the policy's ability to handle self-occlusion or rear-side dribbling.
- **Yellow (GE Lost):** The ball is within the FOV but data is lost due to temporal burst failure ($m_t^{\text{GE}} = 0, m_t^{\text{fov}} = 1$). This evaluates robustness against communication delays or motion blur.
- **Purple (Both Lost):** The system suffers from simultaneous geometric and temporal failures ($m_t^{\text{GE}} = 0, m_t^{\text{fov}} = 0$).



(a) Operational (b) FOV Lost (c) GE Lost (d) Both Lost

Fig. 9. Example snapshots of the perception availability model induced by the FOV gate and the Gilbert–Elliott (GE) loss process.

During intervals of Red, Yellow, or Purple indicators, the policy relies entirely on the predicted state. We observe that despite frequent and sometimes prolonged occlusion (e.g., the ball remaining in the Red/Purple state while behind the robot), the robot maintains stable dribbling behavior. The policy successfully anticipates the ball's trajectory based on the KF prediction and adjusts its turning rate to re-acquire the ball into the valid FOV. These results confirm that the combination of robustness via randomized training and continuity via state estimation is sufficient to handle intermittent ball loss in dynamic dribbling tasks.

B. Robustness Analysis on Ramp Traversal Limits

We conduct a quantitative stress test using the original pre-trained policy, which is trained on up to slope with 10%, to identify the performance boundaries of our method on ramp terrains. We evaluate success rates across a wide range of slope gradients—specifically ramp-up from 0% to 30% (0°-16.7°) and ramp-down up to 45% (24.2°)—with 10 000 trials performed in simulation for each setting. As illustrated in Fig. 10, our hierarchical framework significantly extends the robustness boundary compared to baselines; notably, at a 20% slope, our policy maintains a 60% success rate on ascent and over 75% on descent, whereas baseline performance degrades to near zero. Note that ramp-up performance is limited by the dribbling skill's capacity to generate sufficient torque. Specifically, at a 25% slope, the robot struggles to push the ball against gravity, leading to timeout failures (exceeding max episode length). Conversely, ramp-down performance is constrained by the locomotion skill. At extreme slopes (e.g., 35%), the high velocity required to catch the gravitationally

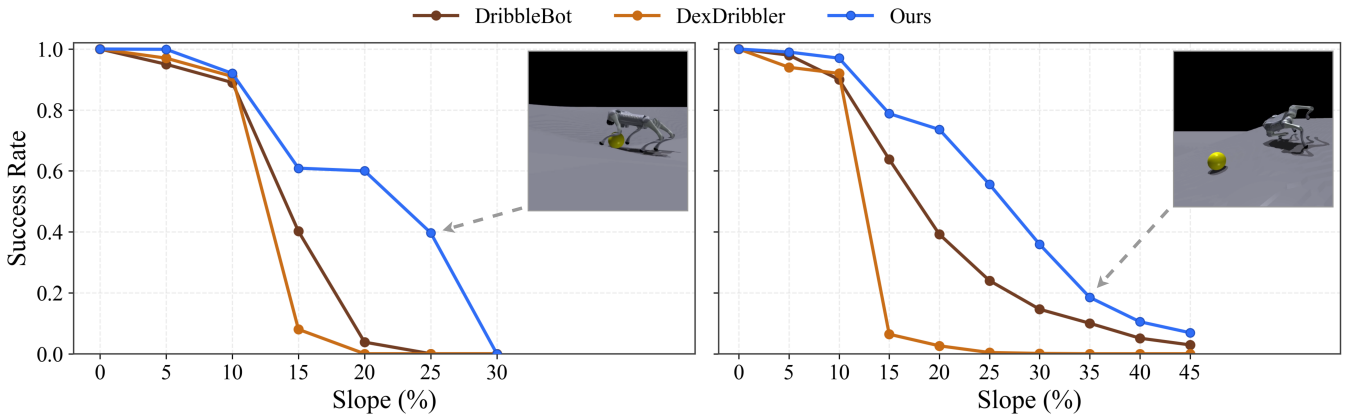


Fig. 10. **Quantitative robustness evaluation on ramp traversal limits.** Comparison of success rates on Ramp-Up (left) and Ramp-Down (right) scenarios. Insets visualize failure modes at physical limits: 1) insufficient torque to push the ball against gravity; 2) locomotion instability caused by excessive slope.

accelerated ball forces the policy into an out-of-distribution (OOD) state, resulting in loss of balance.

C. Training Process Analysis

To explain the performance disparity between DSF-PO and the *Focus-Only* baseline (masked-gradient PPO), we analyze the optimization dynamics through the lens of policy uncertainty and gradient magnitudes.

Premature Mode Collapse in Baselines As observed in Fig. 4, the *Focus-Only* baseline exhibits rapid early improvement but saturates at a suboptimal level. This transient advantage stems from aggressive parameter updates that ignore the selector's uncertainty. In the baseline setting, full-magnitude gradients are backpropagated to the active command head even when the high-level selector is effectively guessing (high entropy). This forces the policy to prematurely commit to specific skills to maximize immediate rewards, leading to premature mode collapse. Consequently, the policy loses the exploration capacity required to master complex transitions, getting trapped in local optima.

Confidence-Aware Weighting and Theoretical Justification In contrast, DSF-PO introduces a confidence-aware weighting mechanism via the skill probability term $w_{d_t}(s_t)$. This acts as a "soft hand-off": when the high-level policy is uncertain, the gradient update to the continuous command head is scaled down. This preserves skill diversity (higher entropy) throughout training, preventing the "thrashing" observed in the baseline.

This mechanism is grounded in the theoretical decomposition of the joint policy's KL divergence. For a hierarchical policy $\pi(a|s)$ composing a discrete selector π^d and continuous command heads π^c, the KL divergence decomposes as:

$$D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{old}}) = D_{\mathrm{KL}}\big(\pi^d_\theta \,\|\, \pi^d_{\mathrm{old}}\big) + \sum_{k=1}^{K} w_k(s)\, D_{\mathrm{KL}}\Big(\pi^c_\theta(\cdot \mid s,k) \,\|\, \pi^c_{\mathrm{old}}(\cdot \mid s,k)\Big). \tag{20}$$

Eq. (20) reveals that to respect the trust region of the joint policy, the optimization of a skill's command distribution must be weighted by its usage probability $w_k(s)$. The *Focus-Only* baseline violates this formulation by removing the weighting term, implicitly optimizing an objective inconsistent with the true policy distribution. DSF-PO restores this consistency, ensuring that skills specialize only when the high-level policy confidently commits to them.

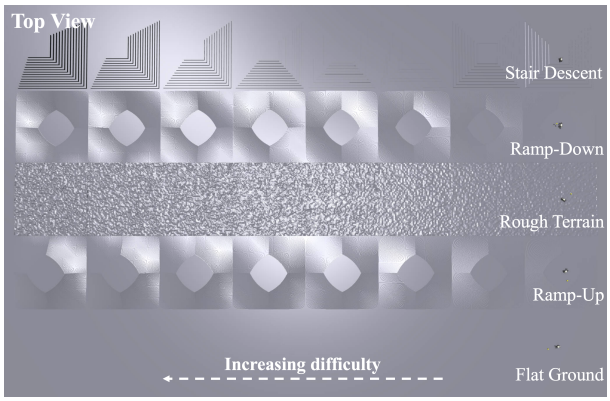


Fig. 11. **Top-view visualization of the progressive terrain curriculum.** From right to left, the curriculum difficulty increases across terrain segments including flat ground, ramp-up, ramp-down, rugged terrain, and stair descent. The same curriculum layout is used for both our method and the baseline to ensure a fair comparison.

D. Performance Analysis on the End-to-End Baseline

We implement an enhanced monolithic baseline based on DribbleBot, denoted as **DribbleBot-Terrain**, which is explicitly trained for rugged terrains. To ensure a rigorous comparison, this baseline incorporates two key enhancements over the original implementation:

- **Enhanced Terrain Rewards:** As detailed in Table V, we augment the reward function with three specialized terms to enhance robustness on rugged terrain: a *Swing Foot Clearance Penalty* to enforce high-stepping behaviors critical for stair climbing, a *Terrain-aware Speed Penalty* to restrict unsafe velocities on irregular surfaces, and a *Roll/Pitch Angular Velocity Penalty* to minimize body oscillation and prevent tipping on steep slopes.

TABLE V
REWARD TERMS OF DRIBBLEBOT-TERRAIN

Original Reward Terms		
Term	Expression	Weight
Projected Ball Velocity	$\exp\left(-\delta_v \|\mathbf{v}^b - \mathbf{v}^{\text{cmd}}\|^2\right)$	0.5
Robot Ball Distance	$\exp\left(-\delta_p \|\mathbf{b} - \mathbf{p}_{\text{FRHip}}\|^2\right)$	4.0
Yaw Alignment	$\exp\left(-\delta_\psi (e^2_{\text{rbcmd}} + e^2_{\text{rbbase}})\right)$	4.0
Ball Velocity Norm	$\exp\left(-\delta_n (\|\mathbf{v}^{\text{cmd}}\| - \|\mathbf{v}^b\|)^2\right)$	4.0
Ball Velocity Angle	$1 - (\psi_b - \psi_{\text{cmd}})^2 / \pi^2$	4.0
Swing Phase Schedule	$(1-\boldsymbol{\kappa}) \exp\left(-\delta_{\text{cf}} \|\mathbf{f}^{\text{foot}}\|^2\right)$	4.0
Stance Phase Schedule	$\boldsymbol{\kappa} \exp\left(-\delta_{\text{cv}} \|\mathbf{v}^{\text{foot}}_{\text{xy}}\|^2\right)$	4.0
Joint Limit Violation	$\mathbb{1}_{q_i > q_{\max} \vee q_i < q_{\min}}$	-10.0
Joint Position Deviation	$\|\mathbf{q} - \mathbf{q}^{\text{default}}\|^2$	-0.05
Joint Torque	$\|\boldsymbol{\tau}\|^2$	-0.0001
Joint Velocity	$\|\dot{\mathbf{q}}\|$	-0.0001
Joint Acceleration	$\|\ddot{\mathbf{q}}\|$	-2.5e-7
Hip/Thigh Collision	$\mathbb{1}_{\text{collision}}$	-5.0
Projected Gravity	$\|\mathbf{g}_{\text{xy}}\|^2$	-5.0
Action Smoothing	$\|\mathbf{a}_{t-1} - \mathbf{a}_t\|^2$	-0.1
Action Smoothing 2	$\|\mathbf{a}_{t-2} - 2\mathbf{a}_{t-1} + \mathbf{a}_t\|^2$	-0.1
Additional Reward Terms		
Term	Expression	Weight
Swing Foot Clearance Penalty	$\frac{1}{N_{\text{swing}}} \sum_{i \in \text{swing}} [h_{\text{tgt}} - h_i]^2_+$	-8.0
Speed Penalty	$\mathbb{I}_{\|\mathbf{v}_{\text{cmd}}\| > v_g} \cdot [v_{\text{obs}} - \bar{v}]^2_+$	-1.5
Roll/Pitch Angular Velocity Penalty	$\|\boldsymbol{\omega}_{xy}\|^2$	-0.2

- **Adaptive Terrain Curriculum:** To rule out optimization difficulties caused by sparse rewards, we utilize the exact same progressive terrain curriculum as our HRL method, ensuring the baseline is gradually exposed to increasing difficulty levels (visualized in Fig. 11).

E. Modality-Skill Consistency Analysis

To provide a more granular analysis of the high-level policy's decision-making logic, we assess whether the high-level skill selection aligns with the contact modality indicated by the impulse-based contact flag $i_t \in \{0, 1\}$: dribbling skills $\mathcal{D} = \{1, 2\}$ are expected under contact (i_t=1) and locomotion skills $\mathcal{L} = \{3, 4\}$ are expected under non-contact (i_t=0).

From the trajectory in Fig. 7, we construct a confusion matrix between the contact label i_t and the predicted skill category $d_t \in \{\mathcal{D}, \mathcal{L}\}$. The matrix is row-normalized and summarized by two scalar metrics reported in Table VI: balanced accuracy (BAcc) and Matthews correlation (MCC). BAcc captures average correctness under class imbalance (good if ≥ 0.80, very strong if ≥ 0.90), while MCC (-1 to 1) reflects overall agreement (strong if ≥ 0.70). In our data, $\text{BAcc} = 0.866$ and $\text{MCC} = 0.727$, indicating strong, well-balanced alignment: the selector reliably switches to $\mathcal{D}$ upon contact and maintains $\mathcal{L}$ otherwise.

APPENDIX B
DEPLOYMENT DETAILS

Hardware and Inference: Our system is deployed on a Unitree Go2 quadruped robot. A downward-facing fisheye camera with a 240° field of view is mounted on the head for ball tracking, which replaces the built-in head LiDAR. All neural network inference and vision processing run onboard on an NVIDIA Jetson Orin NX. Policies are transferred to the physical robot in a zero-shot manner.

Vision System: We utilize a YOLOv11 detector to identify the ball in the fisheye frames. Pixel coordinates are converted to metric distances using the equidistant fisheye model $r = f\theta$. To ensure robust 2D ball localization, we employ two geometric solvers (one based on apparent diameter and another on the angle–range relationship) and fuse their outputs using a Kalman filter.

Low-level Control: The joint-level control is implemented via a PD controller. For most tasks, we set k_p=20 and k_d=0.5. However, for the rough-terrain locomotion skill π^L_4, the gains are increased to k_p=40 and k_d=1.0 to enhance passability and torque response on irregular surfaces.

TABLE VI
ROW-NORMALIZED CONFUSION MATRIX AND METRICS OF MODALITY–SKILL CONSISTENCY

Actual i_t	**Predicted** d_t		BAcc	MCC
	$\mathcal{D}$ (dribbling)	$\mathcal{L}$ (locomotion)		
1 (contact)	0.897	0.103	0.866	0.727
0 (no contact)	0.166	0.834		

APPENDIX C

A. Network Implementation Details

The actor network architecture consists of the following components:

- **Feature Extractor:** A three-layer MLP with $[512, 256, 128]$ units and ELU activations.
- **Skill Head π^d:** A linear layer followed by a *softmax* function, producing a 4D categorical distribution for sampling the skill index d_t.
- **Command Head π^c:** A linear layer with *tanh* activation that outputs the mean $\mu_t \in (-1, 1)$ of a 5D normal distribution $\mathcal{N}(\mu_t, \Sigma)$, from which the low-level commands $\mathbf{c}^L_t$ are sampled. The first two dimensions correspond to dribbling commands, and the remaining three correspond to locomotion commands.

B. Training Hyperparameters

All runs use PPO with standard legged-robot hyperparameters (Table VII), identical across DSF-PO and other ablations for fair comparison in Sec.IV-A1.

TABLE VII
HYPER-PARAMETERS FOR HIGH-LEVEL POLICY TRAINING

Hyperparameter	**Value**
Rollout buffer size	24 steps
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
PPO clipping ratio (ϵ)	0.2
Number of epochs per update	5
Minibatch size	4
Optimizer	Adam
Learning rate	1×10^{-3}
Entropy coefficient	0.01
Value function coefficient	1.0