

FT-WBC: Learning Fault-Tolerant Whole-Body Control for Legged Loco-Manipulation

Yudong Zhong¹, Pengfei Mai¹, Sikai Guo¹, Jiahang Cao², Zhihai Bi¹
Qiuyue Liu¹, Ziyang Feng¹, Jinni Zhou¹, Jun Ma¹

¹The Hong Kong University of Science and Technology (Guangzhou)

²The University of Hong Kong

<https://ft-wbc.github.io/>

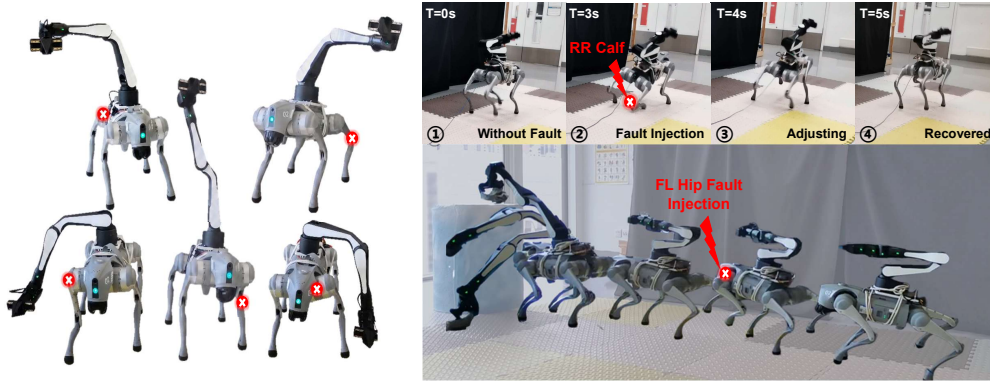


Figure 1: We present FT-WBC, a framework that enables robust, fault-tolerant whole-body control for legged loco-manipulation under actuator failure. FT-WBC governs an arm-mounted quadruped manipulator to dynamically adapt its whole-body posture and synthesized compensatory gaits upon detecting severe actuator failures. The left panel illustrates the whole-body control capability of our framework under failures in different joints. The upper-right panel shows the robot’s rapid gait recovery when the rear right calf joint loses actuation during walking, while the lower-right panel demonstrates a complete reaching and pick-and-place task under a front left hip joint failure.

Abstract: Legged manipulators combine the mobility of legged platforms with the manipulation capability of robotic arms. However, arm-induced Center-of-Mass shifts and dynamic disturbances make the system more prone to instability under actuator failures, potentially leading to falls, task failures, or safety risks. Existing fault-tolerant control methods mainly focus on locomotion alone, leaving the coupled problem of whole-body stability and arm reachability in fault-tolerant loco-manipulation largely unaddressed. To bridge this gap, we propose FT-WBC, a fault-tolerant loco-manipulation framework for robust whole-body control of legged manipulators under actuator failures. FT-WBC adopts a decoupled upper- and lower-body policy architecture and introduces two key modules: a Fault Estimator (FE) and a Posture Adaptation Module (PAM). The FE predicts faulty joints from lower-body proprioceptive histories, while the PAM uses this fault information to adapt the base posture plan generated by the arm policy, converting potentially unstable posture requests into safe and executable base posture commands. Through this fault-aware posture adaptation mechanism, FT-WBC synthesizes compensatory gaits under actuator failures and preserves as much arm workspace as possible while maintaining whole-body stability. Simulation and real-world experiments show that FT-WBC significantly improves survival rate and workspace under weakening or locked failures, and transfers zero-shot to a real legged manipulator in the real world.

Keywords: Fault-Tolerant Control, Whole-Body Control, Loco-Manipulation

1 Introduction

Integrating a robotic arm to a legged platform is a prevalent configuration of mobile manipulation systems. In such systems, the robot not only needs to move close to the target, but also uses body pitch or roll to assist the robotic arm in performing manipulation tasks. Such whole-body manipulation capabilities allow robots to perform more flexible reaching and interaction tasks [1, 2, 3], but they also introduce greater risks of actuator failures during real-world deployment. Unlike locomotion alone, loco-manipulation often requires the robot to maintain tilted base postures for extended periods to expand the arm workspace, which can place sustained loads and torques on specific lower-limb joints. These concentrated loads increase the risk of motor overheating or joint degradation. Such failures can lead to loss of stability, object drops, collisions, or even safety risks to nearby humans. Therefore, legged loco-manipulation systems intended for real-world deployment must be able to continue executing tasks safely under actuator failures.

Existing studies on fault-tolerant control mainly focus on quadrupedal locomotion. Recent methods based on deep reinforcement learning (DRL) have improved the ability of legged robots to maintain locomotion under diverse fault conditions [4, 5, 6]. However, these methods are primarily designed for robots without an active manipulator and mainly target fall prevention and velocity-command tracking. Fault-tolerant loco-manipulation introduces a more demanding stability problem: the mounted arm raises the Center-of-Mass (CoM) and introduces additional dynamic disturbances, making balance harder to maintain when lower-limb actuators fail. Meanwhile, preserving an effective arm workspace often requires base pitch or roll, which may become unsafe after a failure.

To address this gap, this paper proposes FT-WBC, a fault-tolerant whole-body-control framework that balances locomotion stability with workspace expansion under actuator failures. FT-WBC consists of two key modules: the Fault Estimator (FE) and the Posture Adaptation Module (PAM). The FE estimates faulty joint conditions from historical lower-body observations, providing fault information for robust locomotion. The PAM uses this information to adapt manipulation-driven base posture commands, preserving arm reachability while maintaining base stability under joint failures. Our main contributions are summarized as follows:

1. We propose FT-WBC, a fault-tolerant loco-manipulation framework for legged manipulators. The framework coordinates lower-body stability control with upper-body manipulation requirements under joint failures, allowing the robot to maintain robustness while executing whole-body loco-manipulation tasks;
2. We introduce two key modules for robust fault-tolerant control: the FE and the PAM. The FE predicts faulty joints and provides a fault vector for adaptive compensation of lower-limb failures. The PAM utilizes this fault information to revise unstable posture commands, thereby improving base stability and arm workspace under faults;
3. We achieve zero-shot sim-to-real transfer of FT-WBC. Extensive simulation and real-world experiments demonstrate that the framework enables robust control under diverse joint failures and successfully completes object grasping and placing tasks.

2 Related Work

2.1 Legged Loco-Manipulation

Legged loco-manipulation research first focuses on low-level whole-body control, where the main objective is to coordinate the legs and the arm so that the robot can adjust its base posture to expand the reachable workspace of the manipulator [1, 7, 8, 9]. Building on this foundation, subsequent studies further enhance the physical interaction capability of low-level controllers, including contact force regulation, heavy-payload transport, and more stable whole-body coordination [10, 11, 12], allowing legged manipulators to remain effective in more complex object-interaction tasks.

For more complex interaction tasks, relying on the arm alone is not always sufficient. This motivates studies that explore tighter coordination between the legs and the arm, such as coupling upper-limb

manipulation with lower-limb motions to accomplish complex tasks, or mounting small manipulators on the front legs of a quadruped so that the legs can also participate in manipulation [13, 14]. Other work uses the robotic arm as an auxiliary balance mechanism to improve stability during dynamic motion [15]. Meanwhile, stable low-level control alone is still insufficient for autonomous mobile manipulation in open environments.

Therefore, recent work further introduces high-level planning, visual or language instructions, and more complex leg-arm coordination strategies to improve task autonomy and generalization. For example, visual and linguistic information has been used to guide robots through diverse tasks [2, 16, 3, 17, 18, 19], and some studies investigate posture optimization under constrained viewpoints to improve perception and manipulation [20]. These works expand the capability boundary of legged loco-manipulation from low-level control, system morphology, and high-level task planning.

However, these capabilities mostly rely on the assumption of nominal actuator operation. Once actuator failures occur, the robot may no longer reliably generate the required support forces or execute base posture adjustments, making existing loco-manipulation capabilities unstable or even unsafe. Therefore, achieving fault-tolerant whole-body control under actuator failures remains an underexplored yet important problem in legged loco-manipulation.

2.2 Fault-Tolerant Locomotion with DRL

Existing fault-tolerant locomotion studies have substantially improved the ability of quadruped robots to keep moving under joint failures. Early work mainly focuses on recovery after sudden failures, enabling robots to regain stable locomotion after actuator damage [21, 22, 23, 24, 25, 26]. Yet, these methods often consider a limited set of fault conditions. To improve coverage, subsequent studies consider different faulty joints, fault severities, and failure modes, allowing robots to maintain locomotion under a broader range of predefined fault settings [27, 28, 6, 29, 30]. Since predefined robustness does not fully address failures outside the training distribution, more recent methods improve adaptation to unseen failures and reduce reliance on training-specific fault patterns [4, 31], thereby enhancing overall system operational resilience.

Nevertheless, these methods still focus on locomotion alone, such as maintaining balance, recovering gait, and tracking velocity commands. For a legged manipulator, the arm introduces additional payload and CoM shifts, and actively requires base posture adjustments to expand the end-effector workspace. As a result, actuator failures affect not only whether the robot can keep moving, but also whether the arm can perform manipulation stably and effectively. Existing locomotion-oriented fault-tolerant control methods are therefore difficult to directly apply to this coupled stability-and-reachability problem. To address this gap, we study fault-tolerant loco-manipulation and propose FT-WBC, which coordinates lower-body stability control and upper-body manipulation requirements in a fault-aware manner.

3 Methods

3.1 Framework Overview

As illustrated in Figure 2, FT-WBC adopts a fault-aware decoupled upper- and lower-body framework that connects upper-body manipulation intent, fault estimation, posture adaptation, and lower-body stability control into a closed-loop pipeline. Specifically, the arm policy π_{arm} takes the arm observation history $\mathbf{o}_{t-H:t}^{\text{arm}}$ as input and generates the arm action $\mathbf{a}_t^{\text{arm}}$ together with a desired base posture plan \mathbf{u}_t , where the target end-effector pose command is denoted by $\mathbf{c}_t^{\text{arm}}$. To support stable control under joint failures, FE E_θ estimates the current joint-fault condition from the lower-body proprioceptive history $\mathbf{o}_{t-H:t}^{\text{leg}}$ and outputs the predicted fault vector $\hat{\mathbf{f}}_t$, which provides explicit fault information to the leg policy π_{leg} . The PAM G_ϕ then uses the fault information from E_θ to safely remap the base posture plan \mathbf{u}_t into an adapted posture command $\tilde{\mathbf{u}}_t$, preventing posture commands that may shift the CoM toward a degraded support side under faults. Finally, $\tilde{\mathbf{u}}_t$, $\hat{\mathbf{f}}_t$, and $\mathbf{o}_{t-H:t}^{\text{leg}}$ are fed into π_{leg} , allowing it to generate lower-body joint actions $\mathbf{a}_t^{\text{loco}}$ for compensatory gait synthesis

3.3 Posture Adaptation Module

In loco-manipulation, arm reachability conflicts with base stability. For example, grasping a front-right object requires pitching the base forward-right. If the front-right leg fails and the base blindly executes this posture plan, the CoM shifts directly onto the incapacitated leg. Lacking ground reaction force, a catastrophic fall is inevitable.

To address this conflict, we introduce the PAM. In the system hierarchy, this module acts as a high-level intent arbitrator. It deeply fuses the upper-body’s original posture plan \mathbf{u}_t and the predicted fault vector $\hat{\mathbf{f}}_t$ in the feature dimension, feeding them into a non-linear mapping network for intent reconstruction. PAM maps the π_{arm} posture plan into a safe posture-command space:

$$\tilde{\mathbf{u}}_t = \tanh(\mathbf{W}_o \mathbf{h}_t + \mathbf{b}_o),$$

where \mathbf{h}_t represents the fused hidden feature representation, while \mathbf{W}_o and \mathbf{b}_o are the learnable weights and biases of the output layer.

The normalized output is then scaled and clipped within the robot’s physical boundaries:

$$\tilde{c}_t^p = \text{clip}(0.4\tilde{u}_t^p, -0.3, 0.3), \quad \tilde{c}_t^r = \text{clip}(0.4\tilde{u}_t^r, -0.2, 0.2),$$

where \tilde{u}_t^p and \tilde{u}_t^r are the pitch and roll components of $\tilde{\mathbf{u}}_t$. Thus, \tilde{c}_t^p and \tilde{c}_t^r are clipped to $[-0.3, 0.3]$ and $[-0.2, 0.2]$ radians, respectively, preventing infeasible base postures before command execution.

4 Experiment Results

To evaluate the effectiveness and robustness of our proposed fault-tolerant whole-body control framework, we conduct extensive experiments in both simulation and the real world.

Experimental setup: Our hardware platform consists of a Unitree Go2 quadruped equipped with an Airbot Play robotic arm. The control frequency for both the upper and lower body is strictly set to 50 Hz across all environments. For policy training, we utilize the Isaac Gym simulator powered by an NVIDIA RTX 4090 GPU. The training process lasts for a total of 80,000 PPO iterations. To stabilize the initial learning phase, we employ a curriculum strategy: during the first 10,000 iterations, the robotic arm is fixed to pre-train the base locomotion capability; subsequently, the full whole-body loco-manipulation training is activated. For real-world deployment, the learned policy is deployed zero-shot on a Jetson Orin Nano (8 GB) onboard computer for real-time inference.

Baselines and ablations: We use Robo-Duet [8] as our primary baseline, representing a state-of-the-art decoupled loco-manipulation framework without explicit fault-tolerance mechanisms. We further compare our method with two ablated variants: without the FE (**w/o FE**) and without the PAM (**w/o PAM**). In **w/o FE**, the PAM uses ground-truth fault labels instead of FE predictions.

4.1 Evaluation of the FE and PAM

Fault localization: The effectiveness of our adaptive whole-body control hinges on the system’s ability to precisely localize which actuator is impaired. To evaluate the performance of the FE (E_θ), we focus on its capability to identify the specific faulty joint index among the 12 leg actuators.

As illustrated in Figure 3, we present a timeline of fault identification. The x-axis represents time, and the y-axis represents the joint name, with different joint failures injected every 1 s. At $t = 2.0$ s, an initial failure is injected. The predicted fault index closely follows the ground-truth fault over time. Despite noisy proprioceptive data, the module filters transient oscillations and locks onto the correct joint index within approximately 0.05 s after injection.

Posture adaptation: We further validate the effectiveness of the PAM by testing two representative faulty joints on the FL leg and two end-effector targets in the FL lower region. Reaching such targets normally induces leftward roll and forward/downward pitch, which can shift the body load toward the impaired FL support. PAM suppresses these risky fault-directed posture adjustments while pre-

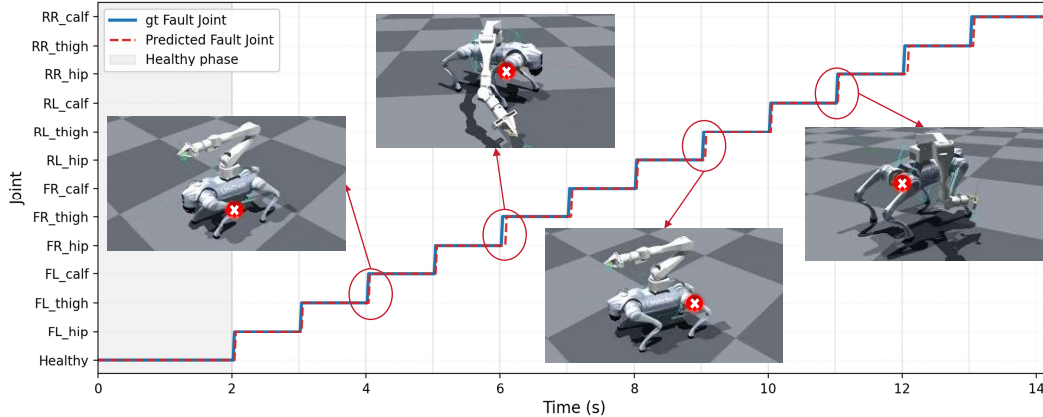


Figure 3: Evaluation of the FE. The x-axis denotes time, and the y-axis denotes the joint name. Different joint faults are injected every 1 s. The solid blue line indicates the ground-truth fault injection, while the dashed red line represents the predicted fault index. GT denotes ground truth.

servicing target reachability. We quantify this effect in Table 1 using two indicators: the fault-leg load ratio $R_j = F_{z,j} / \sum_i F_{z,i}$, where $F_{z,i}$ denotes the vertical contact force of foot i and j denotes the faulty leg, and the fault-side tilt score $S_j = \max(0, \mathbf{d}_j^\top \mathbf{g}_{xy})$, with $\mathbf{d}_j = \mathbf{r}_j^{xy} / \|\mathbf{r}_j^{xy}\|_2$. Here, $\mathbf{g}_{xy} = [g_x, g_y]^\top$ is the horizontal projected-gravity vector, and \mathbf{r}_j^{xy} is the horizontal position vector from the base to the faulty foot j . Lower values indicate reduced loading and tilt toward the faulty FL leg. Compared with Robo-Duet and **w/o PAM**, our method lowers both metrics and yields safer reaching postures under different joint failure conditions while maintaining the commanded target.

4.2 Simulation Experiments

To rigorously evaluate the system’s robustness, we use k_τ to denote the actuator weakening level, with its detailed definition provided in Appendix C. We then individually inject partial ($k_\tau = 0.1$) and complete ($k_\tau = 0.0$) weakening faults into the 12 lower-limb joints. Each evaluation episode sequentially performs locomotion tracking ($v_x = 0.4$ m/s) and continuous end-effector pose tracking. Evaluated over 1,000 randomized trials, the system is assessed by the **Survival Rate** (preventing mid-task falls) and **Workspace** (successfully reaching targets).

Table 2: Survival rate and workspace under partial ($k_\tau = 0.1$) and complete ($k_\tau = 0.0$) weakening across six representative joints. RD denotes Robo-Duet, and “-” indicates immediate task failure.

Faulty Joint	Survival Rate (%) \uparrow								Workspace (m^3) \uparrow							
	Partial Weakening ($k_\tau = 0.1$)				Complete Weakening ($k_\tau = 0.0$)				Partial Weakening ($k_\tau = 0.1$)				Complete Weakening ($k_\tau = 0.0$)			
	RD	w/o FE	w/o PAM	Ours	RD	w/o FE	w/o PAM	Ours	RD	w/o FE	w/o PAM	Ours	RD	w/o FE	w/o PAM	Ours
FL Calf	-	55.2	81.4	84.5	-	35.1	53.4	67.2	-	0.29	0.54	0.59	-	0.19	0.41	0.50
RL Calf	-	57.1	83.6	86.2	-	37.4	55.6	68.8	-	0.33	0.57	0.61	-	0.21	0.43	0.52
FR Thigh	52.3	68.4	91.2	93.4	39.4	55.2	73.1	84.3	0.33	0.45	0.75	0.77	0.24	0.35	0.63	0.70
RL Thigh	55.1	71.3	93.6	94.8	42.1	58.7	76.3	86.5	0.35	0.48	0.77	0.79	0.27	0.38	0.66	0.72
FL Hip	59.4	75.2	94.8	95.6	45.3	62.4	80.2	88.1	0.39	0.52	0.80	0.81	0.30	0.42	0.69	0.75
RR Hip	63.2	79.1	96.5	96.2	50.2	66.3	83.4	89.8	0.43	0.56	0.84	0.82	0.33	0.45	0.71	0.77
Average \uparrow	37.5	67.7	90.2	91.8	29.4	52.5	70.3	80.8	0.25	0.44	0.71	0.73	0.19	0.34	0.59	0.66

Survival rate and workspace: Table 2 shows that FT-WBC consistently improves both survival rate and workspace over Robo-Duet across different joint failures. The results reveal a clear difficulty hierarchy, with calf and thigh faults causing larger performance drops because these joints strongly

affect the base pitch required for reaching. Robo-Duet often fails immediately under critical calf faults due to the lack of fault awareness. The ablation results further show that both FE and PAM are necessary: without FE, the leg policy cannot perceive actuator damage and synthesize proper compensatory forces; without PAM, aggressive posture commands may increase workspace in low-risk cases but can destabilize the robot under severe faults. In contrast, full FT-WBC balances workspace and stability by adapting posture commands according to the estimated fault condition.

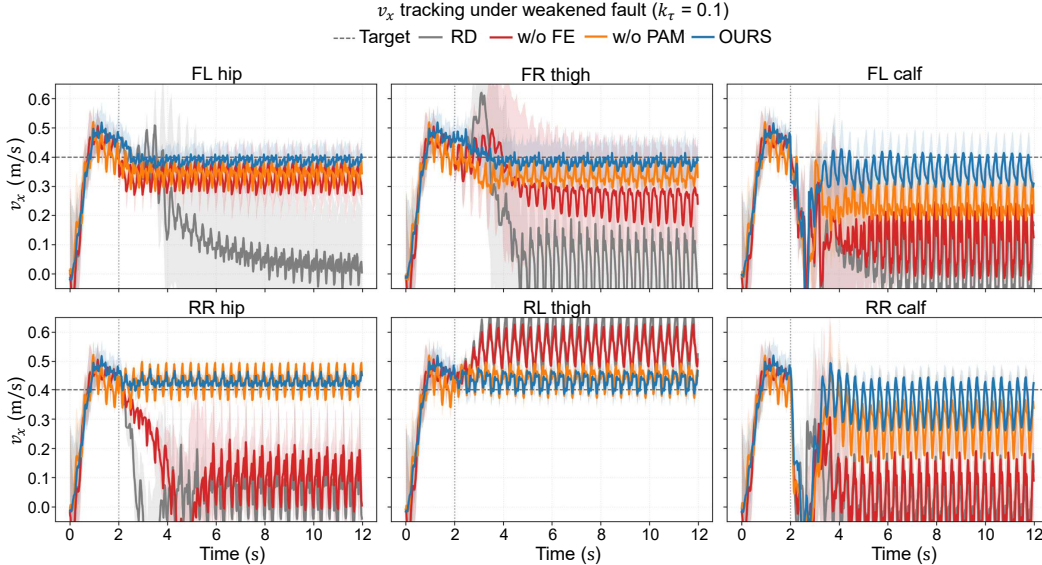


Figure 4: Forward linear velocity (v_x) tracking profiles under severe joint weakening ($k_\tau = 0.1$). The target command (dashed grey) is set to 0.4 m/s. The solid grey, red, orange, and blue lines represent Robo-Duet, w/o FE, w/o PAM, and our full framework, respectively. Our method significantly suppresses velocity oscillations and maintains tight tracking across diverse joint failures.

Velocity tracking: Figure 4 illustrates the real-time linear velocity (v_x) tracking profiles across six distinct failure scenarios. Compared with other methods, **RD** suffers from severe tracking collapse and a complete inability to maintain forward progress in multiple joint scenarios. Meanwhile, due to the inability to perceive actuator discrepancies, **w/o FE** exhibits divergent oscillations that quickly lead to falls. Incorporating the FE (as seen in **w/o PAM**) markedly improves survival compared to both **RD** and **w/o FE**, yet lacking posture adaptation still manifests as observable steady-state errors and speed fluctuations. Conversely, our full framework maintains precise tracking across all joint types. It rapidly synthesizes compensatory gaits to handle initial velocity drops caused by calf faults and effectively suppresses erratic swings induced by thigh faults. Overall, these results show the importance of combining rapid fault prediction with high-level posture adaptation.

4.3 Real-World Deployment

To validate the zero-shot sim-to-real transferability and the practical robustness of our framework, we design a ground-to-table pick-and-place task. The robot must pick up an object from the ground (requiring a deep downward pitch) and place it onto tables of varying heights: 50 cm, 75 cm, and 90 cm. The 90 cm target is exceptionally demanding, forcing the quadruped into an extreme upward body pitch—a precarious maneuver when a support leg is compromised. As visually demonstrated in Figure 5, our framework enables the robot to safely execute these dynamic reaching and placement sequences despite being subjected to diverse and severe impairments, including partial weakening (FL Hip), complete weakening (FL Thigh), and an unmodeled locked fault (RL Calf).

Physical trial results: We conduct 20 physical trials for each height under three conditions: Partial Weakening (PW), Complete Weakening (CW), and an unmodeled Locked Fault (LF). Notably, evaluating against locked (jammed) joints explicitly verifies the framework’s zero-shot generalization

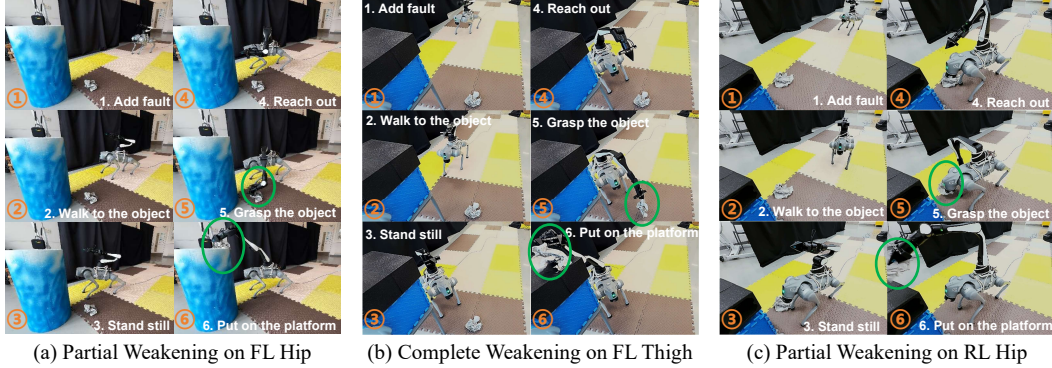


Figure 5: Real-world dynamic manipulation under different joint failures. Each subfigure shows fault-aware reaching on the left and the resulting pick-and-place motion on the right.

capability, as the policy is exclusively trained on torque-weakening faults. The quantitative results, measured by **Survival Rate** and **Success Rate** (completing the placement), are presented in Table 3. As shown, the framework demonstrates remarkable resilience. For standard heights (50 cm and 75 cm), the system maintains high survival and success rates.

Locked fault generalization: At the extreme 90 cm height, our framework still guarantees a 70% survival rate even under severe locked conditions, demonstrating strong adaptation to out-of-distribution actuator failures. While the task success rate drops to 45%, this occurs because the PAM conservatively restricts the extreme upward pitch to prevent the CoM from escaping the degraded support polygon. This “survival-first” behavior aligns with the design philosophy of real-world fault-tolerant systems: prioritizing catastrophic-damage prevention over aggressive task completion.

Table 3: Pick-and-Place Performance.

Height	Cond.	Survival	Success
50 cm	PW	20/20	19/20
	CW	20/20	18/20
	LF	19/20	16/20
75 cm	PW	20/20	18/20
	CW	19/20	16/20
	LF	17/20	14/20
90 cm	PW	19/20	16/20
	CW	16/20	13/20
	LF	14/20	9/20

5 Conclusion

This work presents FT-WBC, a fault-tolerant whole-body control framework that enables legged manipulators to maintain stable loco-manipulation under actuator failures. By incorporating fault estimation and posture adaptation into a decoupled upper- and lower-body policy architecture, FT-WBC dynamically adjusts base posture commands according to the current fault condition, allowing the robot to preserve whole-body stability while retaining as much arm workspace as possible.

Simulation and real-world experiments show that FT-WBC significantly improves survival rate and workspace under various joint weakening and complete failure scenarios, and further achieves effective zero-shot adaptation to locked faults that are not seen during training. These results demonstrate that the proposed framework improves not only locomotion stability under failures, but also the ability to sustain whole-body manipulation tasks. Overall, FT-WBC extends fault-tolerant control from conventional locomotion to loco-manipulation, providing a practical and scalable solution for reliable deployment of legged manipulators under actuator failures.

6 Limitations

Despite its significant progress, FT-WBC has two main limitations. First, it currently targets single-actuator failures and does not yet address concurrent multi-joint failures, where the possible fault combinations become high-dimensional. Second, our evaluations focus on stationary-base or sequential “move-then-manipulate” tasks, and extending fault-tolerant control to dynamic manipulation while walking remains future work for real-world legged manipulation.

Acknowledgments

If a paper is accepted, the final camera-ready version will (and probably should) include acknowledgments. All acknowledgments go at the end of the paper, including thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

References

- [1] A. E. Vijayan, A. Cramariuc, M. Risiglione, C. Gehring, and M. Hutter. Multi-critic learning for whole-body end-effector twist tracking. *arXiv preprint arXiv:2507.08656*, 2025.
- [2] X. Liu, B. Ma, C. Qi, Y. Ding, N. Xu, G. Zhang, P. Chen, K. Liu, Z. Jia, C. Guan, et al. Mlm: Learning multi-task loco-manipulation whole-body control for quadruped robot with arm. *IEEE Robotics and Automation Letters*, 11(1):81–88, 2025.
- [3] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. *arXiv preprint arXiv:2407.10353*, 2024.
- [4] Y. Fu, Y. Zhang, Q. Yang, L. Yan, Z. Cao, and Y. Gao. Contrastive forward prediction reinforcement learning for adaptive fault-tolerant legged robots. In *Conference on Robot Learning*, pages 3285–3303. PMLR, 2025.
- [5] M. Liu, D. Pathak, and A. Agarwal. Locoformer: Generalist locomotion via long-context adaptation. *arXiv preprint arXiv:2509.23745*, 2025.
- [6] S. Lee, I. M. A. Nahrendra, D. Lee, B. Yu, M. Oh, H. Lee, and H. Myung. Dreamflex: Learning fault-aware quadrupedal locomotion controller for anomaly situation in rough terrains. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16001–16007. IEEE, 2025.
- [7] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023.
- [8] G. Pan, Q. Ben, Z. Yuan, G. Jiang, Y. Ji, S. Li, J. Pang, H. Liu, and H. Xu. Roboduet: Learning a cooperative policy for whole-body legged loco-manipulation. *IEEE Robotics and Automation Letters*, 2025.
- [9] N. Fey, G. B. Margolis, M. Peticco, and P. Agrawal. Bridging the sim-to-real gap for athletic loco-manipulation. *arXiv preprint arXiv:2502.10894*, 2025.
- [10] T. Portela, G. B. Margolis, Y. Ji, and P. Agrawal. Learning force control for legged manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15366–15372. IEEE, 2024.
- [11] Q. Li, Q. Meng, Y. Qin, J. Chen, X. Ding, and K. Xu. Dynamic interaction control in legged mobile manipulators: A decoupled approach. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1406–1412. IEEE, 2024.
- [12] A. Rigo, M. Hu, S. K. Gupta, and Q. Nguyen. Hierarchical optimization-based control for whole-body loco-manipulation of heavy objects. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15322–15328. IEEE, 2024.
- [13] X. Zhu, Y. Chen, L. Sun, F. Niroui, S. L. Cleac’h, J. Wang, and K. Fang. Versatile loco-manipulation through flexible interlimb coordination. *arXiv preprint arXiv:2506.07876*, 2025.
- [14] Y. Niu, Y. Zhang, M. Yu, C. Lin, C. Li, Y. Wang, Y. Yang, W. Yu, T. Zhang, Z. Li, et al. Human2locoman: Learning versatile quadrupedal manipulation with human pretraining. *arXiv preprint arXiv:2506.16475*, 2025.

- [15] H. Huang, A. Loquercio, A. Kumar, N. Thakkar, K. Goldberg, and J. Malik. Manipulator as a tail: Promoting dynamic stability for legged locomotion. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9712–9719. IEEE, 2024.
- [16] M. Liu, Z. Chen, X. Cheng, Y. Ji, R.-Z. Qiu, R. Yang, and X. Wang. Visual whole-body control for legged loco-manipulation. *arXiv preprint arXiv:2403.16967*, 2024.
- [17] J. Zhang, N. Gireesh, J. Wang, X. Fang, C. Xu, W. Chen, L. Dai, and H. Wang. Gamma: Graspability-aware mobile manipulation policy learning based on online grasping pose fusion. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1399–1405. IEEE, 2024.
- [18] A. Jaafar, S. S. Raman, S. Harithas, Y. Wei, S. Juliani, A. Wernerfelt, B. Quartey, I. Idrees, J. X. Liu, and S. Tellex. λ : A benchmark for data-efficiency in long-horizon indoor mobile manipulation robotics. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8942–8949, 2025. doi:10.1109/IROS60139.2025.11246789.
- [19] R. Mendonca, E. Panov, B. Bucher, J. Wang, and D. Pathak. Continuously improving mobile manipulation with autonomous real-world rl. *arXiv preprint arXiv:2409.20568*, 2024.
- [20] S. Xie, C. Hu, D. Wang, J. Johnson, M. Bagavathiannan, and D. Song. Coupled active perception and manipulation planning for a mobile manipulator in precision agriculture applications. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12665–12671. IEEE, 2024.
- [21] K. Liu, Z. Wang, B. Li, L. Zhu, and H. Ding. Fault joint detection and adaptive fault-tolerant control of legged robots under joint partial failures. *IEEE Robotics and Automation Letters*, 2025.
- [22] Z. Luo, E. Xiao, and P. Lu. Ft-net: Learning failure recovery and fault-tolerant locomotion for quadruped robots. *IEEE Robotics and Automation Letters*, 8(12):8414–8421, 2023.
- [23] F. Yang, C. Yang, D. Guo, H. Liu, and F. Sun. Fault-aware robust control via adversarial reinforcement learning. In *2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 109–115. IEEE, 2021.
- [24] D. Liu, T. Zhang, J. Yin, and S. See. Saving the limping: Fault-tolerant quadruped locomotion via reinforcement learning. *arXiv preprint arXiv:2210.00474*, 2022.
- [25] W. Okamoto, H. Kera, and K. Kawamoto. Reinforcement learning with adaptive curriculum dynamics randomization for fault-tolerant robot control. *arXiv preprint arXiv:2111.10005*, 2021.
- [26] X. Wu, W. Dong, H. Lai, Y. Yu, and Y. Wen. Adaptive control strategy for quadruped robots in actuator degradation scenarios. In *Proceedings of the Fifth International Conference on Distributed Artificial Intelligence*, pages 1–13, 2023.
- [27] P. Zhang, Z. Hua, Q. Qiu, and J. Ding. Ft-cpg: Learning central pattern generators for fault-tolerant quadruped locomotion under multi-joint failures. *IEEE Robotics and Automation Letters*, 2025.
- [28] M. Kim, U. Shin, and J.-Y. Kim. Learning quadrupedal locomotion with impaired joints using random joint masking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9751–9757. IEEE, 2024.
- [29] Y. Qiu, X. Lin, J. Wang, X. Li, L. Qi, and M.-H. Yang. Umc: Unified resilient controller for legged robots with joint malfunctions. *arXiv preprint arXiv:2502.03035*, 2025.

- [30] C. Chen, C. Li, H. Lu, Y. Wang, and R. Xiong. Meta reinforcement learning of locomotion policy for quadruped robots with motor stuck. *IEEE Transactions on Automation Science and Engineering*, 22:5551–5565, 2024.
- [31] T. Hou, J. Tu, X. Gao, Z. Dong, P. Zhai, and L. Zhang. Multi-task learning of active fault-tolerant controller for leg failures in quadruped robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9758–9764. IEEE, 2024.

Appendix

A Leg Policy and Arm Policy

A.1 Policy Inputs and Actions

FT-WBC adopts a decoupled upper- and lower-body policy architecture. The lower-body leg policy π_{leg} generates quadruped joint actions for locomotion stability and compensatory gait synthesis, while the upper-body arm policy π_{arm} tracks the target end-effector pose and produces both manipulator actions and a desired base posture plan. The FE and PAM provide the fault-aware connection between these two policies: FE supplies the predicted fault vector, and PAM adapts the base posture plan before it is executed by the lower-body controller.

The policy π_{leg} takes a 30-step lower-body observation history $\mathbf{o}_{t-29:t}^{\text{leg}} \in \mathbb{R}^{1680}$ and target motion commands as input. The locomotion command $\mathbf{c}_t^{\text{leg}} \in \mathbb{R}^5$ contains the desired forward linear velocity v_x , lateral velocity v_y , yaw angular velocity ω_z , body pitch command, and body roll command, which specify the target motion and posture of the robot base. Each lower-body observation $\mathbf{o}_t^{\text{leg}} \in \mathbb{R}^{56}$ contains the projected gravity $\mathbf{g}_t \in \mathbb{R}^3$, leg states composed of relative leg joint positions $\mathbf{q}_t^{\text{leg}} - \mathbf{q}_0^{\text{leg}} \in \mathbb{R}^{12}$ and leg joint velocities $\dot{\mathbf{q}}_t^{\text{leg}} \in \mathbb{R}^{12}$, the previous leg action $\mathbf{a}_{t-1}^{\text{leg}} \in \mathbb{R}^{12}$, lower-body velocity and posture commands $\mathbf{c}_t^{\text{leg}} \in \mathbb{R}^5$, upper-body target commands $\mathbf{c}_t^{\text{arm}} \in \mathbb{R}^6$, base states composed of the roll angle $\phi_t \in \mathbb{R}$ and pitch angle $\theta_t \in \mathbb{R}$, and the predicted fault vector $\hat{\mathbf{f}}_t \in \mathbb{R}^{12}$ from FE. π_{leg} outputs leg joint actions $\mathbf{a}_t^{\text{leg}} \in \mathbb{R}^{12}$, which represent joint position offsets relative to the default joint positions. These actions are converted by the leg PD controller into joint torques $\boldsymbol{\tau}_t^{\text{leg}}$, which are finally sent to the robot to drive the twelve leg joints.

π_{arm} tracks the target end-effector pose based on a 30-step arm observation history $\mathbf{o}_{t-29:t}^{\text{arm}} \in \mathbb{R}^{600}$. The manipulation command $\mathbf{c}_t^{\text{arm}} \in \mathbb{R}^6$ is expressed in a polar-style task parameterization: l denotes the target distance from the base frame, while p and y define the target direction through pitch and yaw angles. The variables α , β , and γ further specify the desired roll, pitch, and yaw orientation of the end effector. Each arm observation $\mathbf{o}_t^{\text{arm}} \in \mathbb{R}^{20}$ contains arm states composed of the relative joint positions of the six arm joints $\mathbf{q}_t^{\text{arm}} - \mathbf{q}_0^{\text{arm}} \in \mathbb{R}^6$, the previous arm action $\mathbf{a}_{t-1}^{\text{arm}} \in \mathbb{R}^6$, the target end-effector command $\mathbf{c}_t^{\text{arm}} \in \mathbb{R}^6$, and base states composed of the roll angle $\phi_t \in \mathbb{R}$ and pitch angle $\theta_t \in \mathbb{R}$. The output of π_{arm} , $\mathbf{a}_t^{\text{arm}} \in \mathbb{R}^8$, contains six arm joint actions and a desired base posture plan $\mathbf{u}_t \in \mathbb{R}^2$ sent to PAM. The six joint-action components are converted by the arm PD controller into arm joint torques $\boldsymbol{\tau}_t^{\text{arm}}$, which are finally sent to the robot manipulator. This posture plan provides additional posture degrees of freedom for the arm and helps expand the reachable workspace of the end effector.

A.2 Adaptation Modules

Besides the FE and PAM, FT-WBC keeps the lightweight RMA-style adaptation modules used in the Robo-Duet policy decomposition. These modules should not be confused with the proposed fault-specific PAM: they do not output posture corrections and do not explicitly reason about the predicted fault vector $\hat{\mathbf{f}}_t$. Instead, they infer compact privileged context variables that are available in simulation during training but unavailable at deployment.

For each branch $b \in \{\text{leg}, \text{arm}\}$, where leg and arm denote the leg and arm branches, respectively, the adaptation module takes the corresponding observation history $\mathbf{o}_{t-H:t}^b$ as input and predicts a low-dimensional privileged vector $\hat{\mathbf{p}}_t^b = \phi_{\text{ada}}^b(\mathbf{o}_{t-H:t}^b)$. The notation $\mathbf{o}_{t-H:t}^b$ follows the Method section and denotes the history window from timestep $t - H$ to t . The network ϕ_{ada}^b uses hidden dimensions 256 and 128, followed by a linear output layer whose dimension matches the privileged-observation dimension of that branch.

For the leg branch, the privileged vector is $\mathbf{p}_t^{\text{leg}} = [\tilde{\mu}_t, \tilde{e}_t]^\top \in \mathbb{R}^2$, where $\tilde{\mu}_t$ and \tilde{e}_t denote the normalized ground-friction coefficient and restitution coefficient. For the arm branch, the privileged vector is $\mathbf{p}_t^{\text{arm}} = [\tilde{\mu}_t, \tilde{e}_t, \mathbf{c}_t^{\text{ee}}, \mathbf{q}_t^{\text{ee}}]^\top \in \mathbb{R}^9$, where $\mathbf{c}_t^{\text{ee}} = [l_t, p_t, y_t]^\top \in \mathbb{R}^3$ is the LPY target rep-

representation in the base frame and $\mathbf{q}_t^{\text{ee}} = [q_{w,t}^{\text{ee}}, q_{x,t}^{\text{ee}}, q_{y,t}^{\text{ee}}, q_{z,t}^{\text{ee}}]^\top \in \mathbb{R}^4$ is the end-effector orientation quaternion relative to the base frame.

The adaptation module is trained with the MSE loss $\mathcal{L}_{\text{ada}}^b = \|\hat{\mathbf{p}}_t^b - \mathbf{p}_t^b\|_2^2$ for $b \in \{\text{leg}, \text{arm}\}$, regressing the simulator privileged observations used by the teacher critic. During deployment, these privileged observations are removed, and the actor receives the predicted vector $\hat{\mathbf{p}}_t^b$ instead. The leg actor concatenates the leg observation history $\mathbf{o}_{t-29:t}^{\text{leg}} \in \mathbb{R}^{1680}$, the predicted privileged vector $\hat{\mathbf{p}}_t^{\text{leg}} \in \mathbb{R}^2$, and the fault vector $\hat{\mathbf{f}}_t \in \mathbb{R}^{12}$ before the policy MLP, while the arm actor encodes the non-current part of the arm history $\mathbf{o}_{t-29:t-1}^{\text{arm}} \in \mathbb{R}^{580}$ with a separate history encoder and combines it with $\mathbf{o}_t^{\text{arm}} \in \mathbb{R}^{20}$ and $\hat{\mathbf{p}}_t^{\text{arm}} \in \mathbb{R}^9$ before action generation. Thus, the adaptation modules provide compact environment and task context for robust policy execution, while explicit fault estimation and fault-aware base-posture modification are handled separately by FE and PAM.

The task command spaces during training and deployment are listed in Table 4. Here, v_x and ω_z denote the base forward-velocity and yaw-rate commands, respectively; l, p, y specify the end-effector target in the LPY representation; and α, β, γ specify the target end-effector orientation. The deployment ranges are slightly narrower for base velocity commands to improve real-world stability, while the manipulation command ranges are kept consistent with training.

Table 4: Command Ranges for Training and Deployment

Parameters	Training	Deployment
v_x (m/s)	[-1.0, 1.0]	[-0.8, 0.8]
ω_z (rad/s)	[-1, 1]	[-0.8, 0.8]
l (m)	[0.3, 0.77]	[0.3, 0.77]
p (rad)	[-1.41, 1.41]	[-1.41, 1.41]
y (rad)	[-1.57, 1.57]	[-1.57, 1.57]
α (rad)	[-1.41, 1.41]	[-1.41, 1.41]
β (rad)	[-1.05, 1.05]	[-1.05, 1.05]
γ (rad)	[-1.31, 1.31]	[-1.31, 1.31]

B Reward Function Design

In the reward function, we do not encode prior knowledge about specific actuator damage models. Instead, fault-tolerant behavior is learned from randomized fault injection, history-based fault estimation, and fault-aware posture adaptation. The reward is designed to jointly encourage velocity tracking, stable base motion, smooth and energy-efficient joint behaviors, accurate end-effector tracking, and conservative compensatory behaviors under actuator failures. Training is divided into two stages. In **Stage I**, only the locomotion policy is optimized. In **Stage II**, the arm policy, PAM, and whole-body loco-manipulation rewards are activated after PPO iteration T_{hyb} (8,000 from scratch, or 2,000 when resuming from a pretrained leg policy). After this switch, the per-timestep reward combines three components:

$$R_t = R_{\text{locomotion}} + R_{\text{manipulation}} + R_{\text{fault}}, \quad (1)$$

where each component consists of weighted reward or penalty terms. All reward weights are multiplied by the simulation step $\Delta t = 0.005$ s at initialization.

B.1 Locomotion Rewards

The locomotion reward encourages the base to track commanded velocities while maintaining stable, smooth, and energy-efficient leg motions. Stage-I locomotion training also includes swing-foot clearance, contact-force and contact-velocity shaping, collision penalties, and foot-slip penalties. We use Raibert foot placement and contact shaping to regularize gait behaviors instead of explicitly enforcing a fixed pairwise gait pattern. The locomotion reward terms are summarized in Table 5.

Table 5: Locomotion reward terms in Stage I.

Reward name	Reward formula	Function and symbols	Weight
$R_{\text{tracking.lin}}$	$\exp(-\ c_{xy} - \mathbf{v}_{xy}\ ^2 / \sigma_{\text{tracking}}^2)$	Tracks commanded planar velocity; c_{xy} and \mathbf{v}_{xy} are commanded and actual base velocities, with $\sigma_{\text{tracking}} = 0.25$.	1.0
$R_{\text{tracking.ang}}$	$\exp(-(\omega_z^{\text{cmd}} - \omega_z)^2 / \sigma_{\text{yaw}})$	Tracks commanded yaw rate and reduces heading drift; ω_z^{cmd} and ω_z are commanded and actual yaw rates.	0.5
$R_{\text{lin.vel.z}}$	v_z^2	Penalizes vertical base motion and suppresses height oscillation; v_z is the base velocity along the vertical axis.	-2.0×10^{-2}
$R_{\text{ang.vel.xy}}$	$\omega_x^2 + \omega_y^2$	Penalizes roll/pitch angular velocities to stabilize the base and reduce body oscillation during locomotion.	-1.0×10^{-3}
R_{torque}	$\sum_j \tau_j^2$	Reduces large joint torques; τ_j is the torque applied to joint j .	-1.0×10^{-4}
$R_{\text{dof.vel}}$	$\sum_j \dot{q}_{j,t}^2$	Penalizes fast joint motion; $\dot{q}_{j,t}$ denotes joint velocity at timestep t .	-1.0×10^{-4}
$R_{\text{dof.acc}}$	$\sum_j ((\dot{q}_{j,t} - \dot{q}_{j,t-1}) / \Delta t)^2$	Smooths joint trajectories by penalizing abrupt acceleration changes; Δt is the simulation step.	-2.5×10^{-7}
$R_{\text{action.rate}}$	$\sum_j (a_{j,t} - a_{j,t-1})^2$	Encourages smooth actions; $a_{j,t}$ is the current action of joint j .	-1.0×10^{-2}
$R_{\text{loco.energy}}$	$\sum_{j \in \text{legs}} (\tau_j \dot{q}_j)^2$	Penalizes leg-joint power to encourage efficient compensatory gaits.	-4.0×10^{-5}
R_{smooth}	$\sum_i m_{t,i} (q_{t,i}^{\text{target}} - 2q_{t-1,i}^{\text{target}} + q_{t-2,i}^{\text{target}})^2$	Penalizes target-position jerk; $m_{t,i}$ masks inactive actions.	-1.0×10^{-1}
R_{raibert}	$\sum_{i=1}^4 \ \mathbf{f}_i^{\text{des}} - \mathbf{f}_i^{\text{act}}\ ^2$	Regularizes foot placement for stable stepping; $\mathbf{f}_i^{\text{des}}$ and $\mathbf{f}_i^{\text{act}}$ are desired and actual foot positions.	-10.0
$R_{\text{clearance}}$	$\sum_i (h_i^{\text{des}} - h_i)^2$	Shapes swing-foot clearance to avoid low swings; h_i^{des} and h_i are desired and actual foot heights.	-30.0
$R_{\text{contact.force}}$	$\sum_i \max(\ \mathbf{F}_i\ - F_{\text{max}}, 0)^2$	Regularizes contact forces; \mathbf{F}_i is foot force with threshold F_{max} .	4.0
$R_{\text{contact.vel}}$	$\sum_i \mathcal{I}_i^c \ \mathbf{v}_i^{\text{foot}}\ ^2$	Penalizes stance-foot motion; \mathcal{I}_i^c indicates contact of foot i .	4.0
$R_{\text{collision}}$	$\sum_k \mathcal{I}_k^{\text{collision}}$	Penalizes undesired contacts or collisions; $\mathcal{I}_k^{\text{collision}}$ indicates collision at body part k .	-5.0
$R_{\text{ori.ctrl}}$	$\ \mathbf{g}_{xy}^{\text{proj}} - \mathbf{g}_{xy}^{\text{des}}\ ^2$	Tracks desired base posture for stable balance; $\mathbf{g}_{xy}^{\text{proj}}$ and $\mathbf{g}_{xy}^{\text{des}}$ are actual and desired gravity projections.	-5.0
R_{slip}	$\sum_i \mathcal{I}_i^c \ \mathbf{v}_{i,xy}^{\text{foot}}\ ^2$	Penalizes stance-foot slip; $\mathbf{v}_{i,xy}^{\text{foot}}$ is horizontal foot velocity.	-4.0×10^{-2}

Table 6: Manipulation and updated locomotion reward terms in Stage II.

Reward name	Reward formula	Function and symbols	Weight
$R_{\text{tracking.lin}}$	$\exp(-\ c_{xy} - \mathbf{v}_{xy}\ ^2 / \sigma_{\text{tracking}}^2)$	Maintains base translation tracking during reaching and posture adaptation; symbols follow Table 5.	0.7
$R_{\text{tracking.ang}}$	$\exp(-(\omega_z^{\text{cmd}} - \omega_z)^2 / \sigma_{\text{yaw}})$	Maintains yaw-rate tracking without over-constraining the adaptive base posture; symbols follow Table 5.	0.25
R_{manip}	$\exp(-w_{\text{ipy}} e_{\text{ipy}} - w_{\text{rpy}} e_{\text{rpy}})$	Tracks commanded end-effector pose; e_{ipy} and e_{rpy} denote position and orientation errors.	1.0
$R_{\text{ori.ctrl}}$	$\ \mathbf{g}_{xy}^{\text{proj}} - \mathbf{g}_{xy}^{\text{des}}\ ^2$	Strengthens base posture tracking during reaching and whole-body balancing; symbols follow Table 5.	-10.0
$R_{\text{ori.heur}}$	$\ \mathbf{g}_{xy}^{\text{proj}} - \mathbf{g}_{xy}^{\text{safe}}\ ^2$	Penalizes unsafe body tilting with safe posture reference $\mathbf{g}_{xy}^{\text{safe}}$.	-2.0
$R_{\text{hip.act}}$	$\sum_{j \in \text{hip}} a_{j,t}^2$	Regularizes hip actions to suppress aggressive lateral posture changes during reaching task.	-5.0×10^{-2}
$R_{\text{plan.smooth}}$	$\ \mathbf{u}_t - \mathbf{u}_{t-1}\ ^2$	Smooths base posture plans to prevent high-frequency oscillations, where \mathbf{u}_t represents the arm-policy pitch/roll plan.	-1.0×10^{-1}
$R_{\text{plan.limit}}$	$\ \max(\mathbf{u}_t - \mathbf{u}_{\text{max}}, 0)\ ^2$	Penalizes kinematically infeasible pitch/roll plans that strictly exceed the predefined posture-plan bound \mathbf{u}_{max} .	-5.0
$R_{\text{arm.energy}}$	$\sum_{j \in \text{arm}} (\tau_j \dot{q}_j)^2$	Penalizes excessive arm-joint power to reduce actuator wear and ensure stable, smooth manipulation.	-4.0×10^{-5}

B.2 Manipulation Rewards

In Stage II, the arm policy is activated and outputs both six arm-joint actions and a two-dimensional base pitch/roll posture plan. The manipulation rewards encourage the end effector to track the commanded pose while keeping the base posture plan smooth and physically feasible. During Stage II, the linear and angular velocity tracking weights are reduced to 0.7 and 0.25, respectively, because the robot must balance velocity tracking with end-effector reachability and posture adaptation. The orientation-control penalty is increased to strengthen base posture regulation during reaching. The Stage-II reward terms are shown in Table 6.

Table 7: Fault-oriented reward weights.

Category	Reward	Weight
Hold faulty-joint motion	$r_{\text{motion}}^{\text{fault}}$	-0.2
Contralateral support	r_{axis}	+0.6

B.3 Fault-Oriented Rewards

Fault-oriented rewards are used to encourage compensatory behaviors after actuator degradation. These terms are not designed from a detailed fault dynamics model; instead, they provide weak guidance that helps the policy reduce the reliance on failed joints and redistribute support to healthier legs. Let $F_j \in \{0, 1, 2\}$ denote the fault label of leg joint j , where 0 indicates a healthy joint, 1 indicates a locked joint, and 2 indicates a weakened joint.

- **Reward for hold faulty-joint motion:**

$$r_{\text{motion}}^{\text{fault}} = \sum_{j=1}^{12} m_j^{\text{fault}} \dot{q}_j^2, \quad (2)$$

where m_j^{fault} is a binary fault mask, with $m_j^{\text{fault}} = 1$ if joint j is faulty ($F_j > 0$) and $m_j^{\text{fault}} = 0$ otherwise. This penalty suppresses high velocities on impaired actuators, encouraging the policy to reduce failed-joint participation and avoid unstable commands that drive damaged motors.

- **Reward of contralateral compensatory support:**

$$r_{\text{axis}} = \exp\left(-\frac{y_{\text{healthy}}^2}{\sigma^2}\right), \quad \sigma = 0.2 \text{ m}, \quad (3)$$

where y_{healthy} is the lateral coordinate of the healthy contralateral foot in the yaw-aligned frame. When one leg in a front or rear pair is faulty, this reward encourages the healthy contralateral leg to move closer to the body sagittal plane, creating a more effective support configuration. Both the front pair and rear pair can contribute, so $r_{\text{axis}} \in [0, 2]$.

The final fault-oriented reward is

$$R_{\text{fault}} = w_m r_{\text{motion}}^{\text{fault}} + w_y r_{\text{axis}}, \quad (4)$$

where w_m and w_y are the corresponding weights. The values are listed in Table 7.

C Fault Definition and Fault Curriculum

Although policy training primarily uses torque weakening, zero-shot experiments also inject locked joints; both modes are defined below. Each leg joint j is labeled by $F_j \in \{0, 1, 2\}$, corresponding to normal, locked, and weakened states. Consistent with the main text, the weakening coefficient $k_\tau \in [0, 1]$ scales available joint torque: $k_\tau = 1$ is healthy, $k_\tau = 0.1$ denotes *partial weakening*, and $k_\tau = 0.0$ denotes *complete weakening* at evaluation time.

Locked joint ($F_j = 1$):

$$q_j^{\text{cmd}'} = \text{clip}(q_j^{\text{cmd}}, c_j - q_{\text{thr}}, c_j + q_{\text{thr}}), \quad q_{\text{thr}} = 0.05 \text{ rad}. \quad (5)$$

Weakened motor ($F_j = 2$):

$$\tau_j' = k_{\tau,j} \tau_j, \quad k_{\tau,j} \in [0, 1]. \quad (6)$$

During training, a faulty episode impairs all three joints of one leg (FL, FR, RL, or RR) with probability 0.95, enabling the policy to repeatedly experience leg-level actuator degradation and learn compensatory support behaviors across different impaired support configurations.

Weakening-severity curriculum. Uniform sampling of $k_{\tau,j}$ yields too few near-complete weakening samples. To emphasize severe actuator degradation, we linearly increase the fraction of samples whose $k_{\tau,j}$ approaches complete weakening ($k_{\tau} = 0$):

$$\rho(t) = \rho_0 + \text{clip}\left(\frac{t - t_0}{T}, 0, 1\right) (\rho_1 - \rho_0), \quad (7)$$

with $\rho_0 = 0$, $\rho_1 = 0.3$, $t_0 = 0$, and $T = 5000$. While $k_{\tau,j} \sim \mathcal{U}(0, 0.25)$, the fraction $\rho(t)$ is stratified toward the near-complete regime, producing a smooth transition from mild weakening to near-complete torque loss during training.

D Hyperparameters of Training

D.1 Adaptation Module and PPO

The adaptation modules are described in Appendix A.2. In Table 8, d_{hist}^b denotes the flattened observation-history dimension of branch $b \in \{l, a\}$, d_{obs}^a denotes the current arm-observation dimension, and d_{priv}^b denotes the privileged-vector dimension predicted by the corresponding adaptation module. In our implementation, the leg branch uses $d_{\text{obs}}^l = 56$, $d_{\text{hist}}^l = 56 \times 30 = 1680$, and $d_{\text{priv}}^l = 2$, while the arm branch uses $d_{\text{obs}}^a = 20$, $d_{\text{hist}}^a = 20 \times 30 = 600$, and $d_{\text{priv}}^a = 9$. Their implementation details and the PPO training hyperparameters are summarized below.

Table 8: Hyperparameters for the adaptation module and PPO.

Hyperparameter	Value
Leg adaptation MLP	$d_{\text{hist}}^l \rightarrow 256 \rightarrow 128 \rightarrow d_{\text{priv}}^l$
Arm adaptation MLP	$d_{\text{hist}}^a \rightarrow 256 \rightarrow 128 \rightarrow d_{\text{priv}}^a$
Adaptation target	branch privileged observations
Adaptation loss	MSE
Arm history encoder	$(d_{\text{hist}}^a - d_{\text{obs}}^a) \rightarrow 512 \rightarrow 256 \rightarrow 128$
Adaptation module learning rate	5×10^{-4}
Number of learning epochs	5
Number of mini-batches	4
Steps per environment per iteration	24
PPO learning rate	5×10^{-4}
Discount factor γ	0.99
GAE parameter λ	0.95
PPO clip ϵ	0.2
Entropy coefficient	0.01
Value loss coefficient	1.0
Desired KL divergence	0.01
Maximum gradient norm	1.0
Default total iterations	10^5
Parallel environments	4096
Simulation step Δt	0.005 s
Control decimation	4
Episode length	20 s
Policy observation history	30 steps
Hybrid switch iteration T_{hyb}	10,000 (2,000 if resuming legs)
Fault probability per episode	0.95
Actor-critic hidden layers	[512, 256, 128], ELU

D.2 Fault Estimator

Training hyperparameters of the FE (Sec. 3.2) are summarized in Table 9.

Table 9: Fault Estimator hyperparameters.

Hyperparameter	Value
Input dimension	$5 \times 56 = 280$
History length K	5
Learning rate	1×10^{-3}
Loss	MSE on fault probability
Network	$280 \rightarrow 512 \rightarrow 256 \rightarrow 128 \rightarrow 12$
GT-fault warmup iterations	3000

D.3 Posture Adaptation Module

Implementation and training hyperparameters of the PAM (Sec. 3.3) are summarized in Table 10.

Table 10: PAM structure and parameters.

Parameter	Value
Input dimension	$2 + 12 = 14$ (plan + fault)
Hidden layers	$128 \rightarrow 64$
Activation	ELU
Output dimension	2 (pitch / roll)
Output activation	tanh
Training	Joint PPO with arm policy
GT-fault warmup	3000 iterations
Pitch clip range (training)	$[-0.4, 0.3]$ rad
Roll clip range (training)	$[-0.4, 0.4]$ rad

E Additional Locked-Fault Experiments

We further evaluate zero-shot robustness under locked actuator faults, which are not included in the torque-weakening training distribution. Following the simulation protocol in Sec. 4.2, each episode evaluates locomotion tracking with $v_x = 0.4$ m/s and continuous end-effector pose tracking. Table 11 reports the survival rate, reachable workspace, and velocity-tracking error under locked failures. The velocity-tracking error is computed along the forward x direction.

Table 11: Survival rate, workspace, and velocity-tracking error under locked faults. RD denotes Robo-Duet, and “-” indicates immediate task failure. The velocity-tracking error is measured along the forward x direction.

Locked Joint	Survival Rate (%) \uparrow				Workspace (m ³) \uparrow				Vel. Tracking Err. (m/s) \downarrow			
	RD	w/o FE	w/o PAM	Ours	RD	w/o FE	w/o PAM	Ours	RD	w/o FE	w/o PAM	Ours
FL Calf	32.4%	64.3%	78.2%	84.1%	0.125	0.254	0.481	0.556	0.395	0.382	0.186	0.042
RR Calf	35.2%	66.5%	79.5%	85.3%	0.142	0.278	0.505	0.578	0.392	0.375	0.174	0.038
FR Thigh	60.1%	76.2%	88.4%	91.2%	0.324	0.426	0.641	0.691	0.321	0.298	0.082	0.015
RL Thigh	64.8%	80.4%	90.2%	93.1%	0.346	0.452	0.662	0.714	0.148	0.136	0.024	0.012
FL Hip	73.5%	84.1%	92.6%	96.2%	0.371	0.483	0.694	0.732	0.344	0.312	0.051	0.018
RR Hip	71.2%	85.3%	93.1%	96.5%	0.389	0.506	0.711	0.742	0.388	0.364	0.032	0.021
Average	56.2%	76.1%	87.0%	91.1%	0.283	0.400	0.616	0.669	0.331	0.311	0.092	0.024

Locked fault generalization: Table 11 shows that FT-WBC generalizes well from torque weakening to unseen locked joints. Across six locked-fault cases, our method obtains the best average survival rate (91.1%), workspace (0.669 m²), and velocity-tracking error (0.024 m/s). Locked calf faults remain the most challenging because they directly affect stance support and foot clearance, forcing the base to undergo significant vertical disturbances. The clear degradation in the w/o FE variant highlights that without an explicit fault belief, the leg policy exhibits severe trajectory tracking lag. Compared with Robo-Duet and the two ablations, full FT-WBC better balances survival, workspace, and tracking accuracy, confirming the benefit of combining explicit fault prediction with posture adaptation under jammed actuators. This dual-loop integration effectively prevents abrupt actuator locking from inducing catastrophic full-body over-turning.

F Snapshots of Simulation Experiments

This section supports the evaluation of FT-WBC using simulation snapshots of velocity tracking and workspace tasks under PW, CW, and LF conditions. These sequences highlight the coordination between full-body posture adaptation and dynamic gait compensation under joint failures.

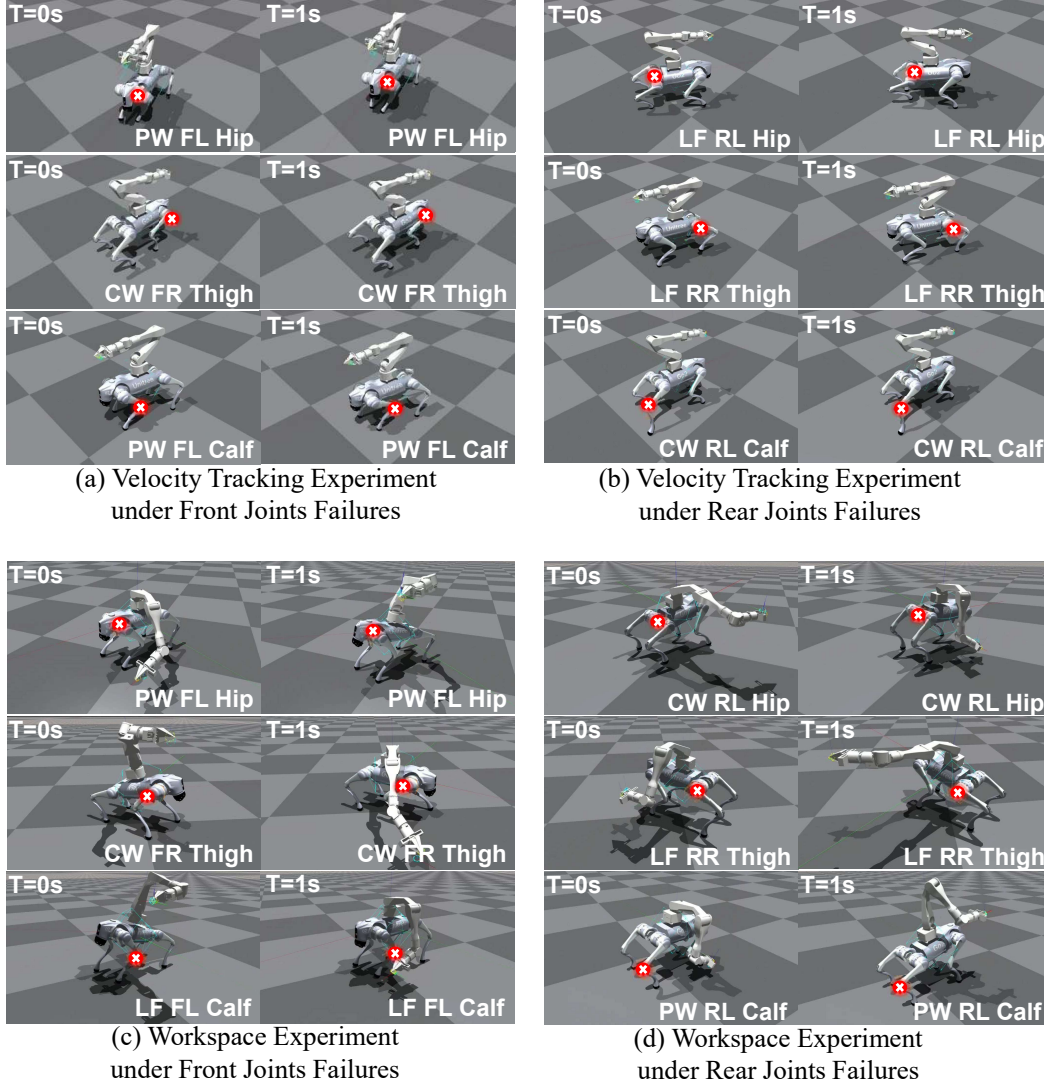


Figure 6: Sequential simulation snapshots at $T = 0$ s and $T = 1$ s under representative PW, CW, and LF joint failure configurations for velocity tracking (top) and workspace test (bottom). Red crosses indicate the joints with injected failures across the quadruped platform.

As shown in Figure 6 (top rows), during locomotion ($v_x = 0.4$ m/s), the FE continuously estimates actuator failures to help synthesize compensatory gaits. The snapshots at $T = 0$ s and $T = 1$ s exhibit adapted walking postures under hip, thigh, and calf failures, where healthy limbs dynamically adjust touchdown behaviors and swing heights to sustain target velocity.

Furthermore, Figure 6 (bottom rows) highlights the robust whole-body control capabilities of our strategy under unexpected joint failures. The adaptive modulation effectively prevents the CoM from escaping the safe support polygon, thereby simultaneously preserving the necessary end-effector workspace and enhancing the overall closed-loop tolerance against actuator failures.