# HAFT: HISTORY-AWARE FLOW POLICY FOR TORQUE-DEGRADED QUADRUPEDAL LOCOMOTION

**Paper ID: 9869**

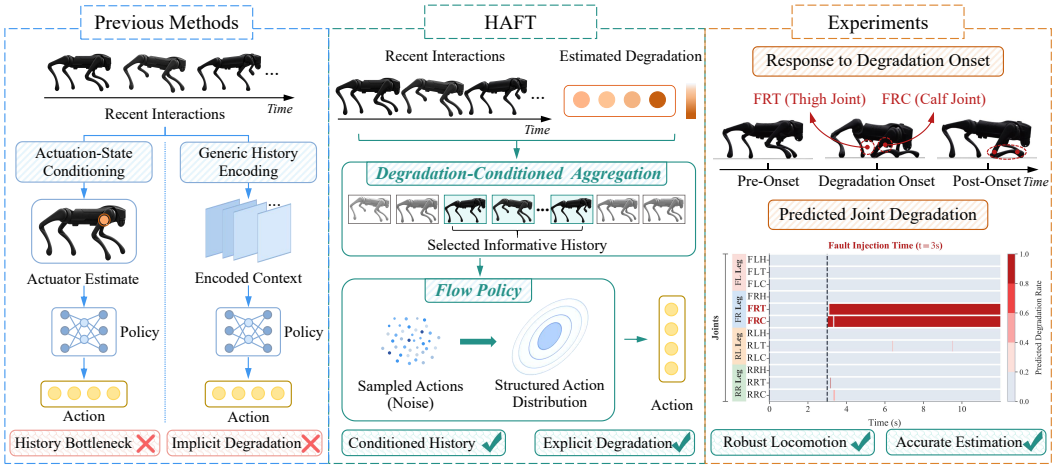


Figure 1: Conceptual comparison of prior paradigms and HAFT. *Left:* Prior methods typically use interaction history either to infer an actuator condition or to construct a generic history representation before action generation. *Middle:* HAFT explicitly estimates joint-wise degradation and uses it together with the current robot state to condition the aggregation of recent interaction history, which subsequently conditions a flow policy for whole-body action generation. *Right:* Locomotion responses before and after degradation onset, together with the corresponding online joint-wise degradation estimates.

## ABSTRACT

Legged locomotion policies are typically optimized under nominal actuation conditions, implicitly assuming a stable relationship between control commands and the resulting robot dynamics. Actuator degradation violates this assumption by reducing joint-level actuator effectiveness, thereby altering the closed-loop dynamics and compromising coordinated whole-body locomotion. A fundamental challenge is that actuator effectiveness is not directly observable from instantaneous proprioception and must instead be inferred from the robot's temporal responses to its recent actions. Viewing this challenge through the lens of online system identification, we exploit recent observation–action history as informative evidence of the underlying actuation condition. Based on this insight, we propose HAFT, a history-conditioned framework for degradation-adaptive legged locomotion. HAFT infers latent actuator effectiveness from recent interactions and selectively aggregates temporal context according to the current robot state and inferred degradation condition. The resulting context is used to condition a flow-matching policy for joint action generation, enabling the policy to adapt whole-body coordination to changes in available actuation. Extensive experiments on quadrupedal locomotion under diverse actuator degradation patterns and severities demonstrate that HAFT consistently improves locomotion robustness and adaptation. Our code is available at https://anonymous.4open.science/r/HAFT-1B35.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

# 1 INTRODUCTION

Quadrupedal robots have demonstrated agile locomotion in unstructured environments, motivating their use in inspection, exploration, and disaster response Hoeller et al. (2024); Kim et al. (2024). Sustained deployment, however, exposes actuators to impacts, wear, and abnormal operating conditions that can compromise their torque output Wu et al. (2023); Lee et al. (2025). We study *joint torque degradation*, in which the torque delivered by a joint is reduced relative to the nominal torque commanded by its low-level controller. Through coupled body dynamics and foot–ground interactions, local torque loss can affect whole-body motion rather than only the impaired joint. Maintaining locomotion therefore requires adapting coordinated joint commands to changes in actuation capability that are not directly observable during deployment.

Learning-based methods have improved robustness to actuator impairments by training policies over diverse actuator conditions and exploiting recent interaction history to infer changes in the underlying dynamics Kim et al. (2024); Lee et al. (2025). More recent studies further incorporate predictive objectives or longer temporal context to characterize the robot's response under changing dynamics Fu et al. (2025); Liu et al. (2025). As illustrated in Fig. 1 (left), representative approaches can be broadly abstracted into two paradigms: inferring an actuator condition from recent interactions and using it to condition the policy, or encoding the interaction history into a generic context representation. These paradigms emphasize complementary information. The former provides an explicit description of the location and severity of actuation loss, whereas the latter retains closed-loop action–response information under varying body configurations and contact conditions. However, the relevance of individual transitions depends jointly on the current robot state and the underlying degradation. This suggests that degradation estimation should not merely serve as an additional policy input, but should also influence how recent interaction history is aggregated.

The resulting context must then be mapped to coordinated whole-body actions. Torque degradation can require compensatory adjustments across both impaired and unaffected joints, with the appropriate coordination pattern varying with degradation severity, contact configuration, and motion state. Conventional diagonal-Gaussian policies provide an efficient parameterization, but impose conditionally independent stochastic variations across action dimensions. Recent flow-based policy optimization methods offer a more flexible alternative by enabling policy-gradient optimization over expressive action distributions McAllister et al. (2026); Yi et al. (2026). Such policies can model dependencies among joint commands and accommodate non-Gaussian action distributions, providing a suitable basis for whole-body compensation under varying actuation conditions.

Motivated by these considerations, we propose **HAFT** (**H**istory **A**ggregation with **F**low Policy for **T**orque-Degraded Locomotion), as illustrated in Fig. 1 (middle). HAFT explicitly estimates continuous joint-wise degradation from recent closed-loop interactions while retaining transition-level information for action generation. The current observation and estimated degradation jointly condition the aggregation of recent transitions, producing a context that reflects both the inferred actuation loss and its observed dynamical consequences. A conditional flow policy then maps this context to a distribution over whole-body joint commands. During training, degradation supervision and flow-policy optimization are jointly applied, while a conditioning curriculum gradually replaces ground-truth degradation with its estimate. At deployment, HAFT relies only on proprioceptive observations and recent interaction history. The main contributions of this work are summarized as follows:

- We develop a degradation-conditioned history aggregation method that uses explicit estimates of joint-wise torque loss to guide the aggregation of recent transitions, retaining action–response information beyond the degradation estimate for control.
- We propose HAFT, which integrates this representation with a context-conditioned flow policy for joint action generation. Joint training and a conditioning curriculum connect degradation estimation with policy adaptation while removing the need for privileged degradation information at deployment.
- We evaluate HAFT across diverse single- and multi-joint torque-degradation configurations. The results demonstrate improved locomotion robustness and velocity-tracking performance relative to the evaluated baselines.

# 2 Related Work

## 2.1 Fault-Tolerant Legged Locomotion under Actuator Impairments

Fault-tolerant legged locomotion has been studied under joint locking, actuator failure, and reduced torque capability. Learning-based methods typically improve robustness by randomizing actuator conditions during training, conditioning a unified policy on inferred actuator states, explicitly estimating fault-related variables, or reorganizing locomotion behaviors according to the remaining actuation capability Liu et al. (2022); Wu et al. (2023); Kim et al. (2024); Hou et al. (2024); Lee et al. (2025); Zhang et al. (2025); Xu et al. (2025); Gravina et al. (2026). Complementary approaches exploit discrepancies between predicted and observed motion to identify changes in the underlying dynamics and adjust locomotion accordingly Luo et al. (2023); Fu et al. (2025). These studies demonstrate that whole-body behavior can compensate for substantial actuator impairments. In this work, we focus on continuous joint torque degradation, where both the affected joints and the severity of torque loss are unavailable to the deployed policy and must be inferred from closed-loop interaction.

## 2.2 History-Based Adaptation to Latent Actuation Changes

Recent interaction history provides important evidence for adaptation when actuator or dynamics conditions are not directly observable. Existing approaches commonly encode proprioceptive histories into latent dynamics or actuator states, or estimate explicit fault-related variables that are subsequently used to condition or modulate locomotion policies Kumar et al. (2021); Wu et al. (2023); Kim et al. (2024); Chen et al. (2024); Lee et al. (2025); Ma et al. (2026). Predictive and long-context methods further show that historical interactions can reveal changes in robot dynamics and support adaptation to unseen morphology or actuator conditions Fu et al. (2025); Liu et al. (2025). However, an actuator estimate and the underlying interaction history are not interchangeable: the former characterizes the loss of actuation capability, whereas individual transitions additionally record how actions were realized under specific body configurations and contact conditions. For continuous torque degradation, this distinction motivates using the inferred degradation to guide the aggregation of transition-level history rather than compressing all historical information into a single latent or fault estimate.

## 2.3 Expressive Action Distributions for Robot Control

Most reinforcement-learning controllers for legged locomotion adopt Gaussian action policies because of their simplicity and efficient optimization Rudin et al. (2022); Kumar et al. (2021); Radosavovic et al. (2024). Recent generative-policy methods provide more expressive alternatives: diffusion-policy optimization enables reward-driven optimization of diffusion action distributions Ren et al. (2025), while flow-based policy optimization extends reinforcement learning to expressive flow-generated action distributions McAllister et al. (2026); Yi et al. (2026); Zhan et al. (2026). Recent studies further investigate efficient integration and inference for robotic flow policies Chen et al. (2026). Compared with commonly used diagonal-Gaussian policies, flow-based policies provide greater flexibility for modeling structured and non-Gaussian joint-action distributions. This property is particularly relevant to torque-degraded locomotion, where compensation may require coordinated adjustments across both impaired and unaffected joints according to the inferred actuation condition and the robot's recent motion responses.

# 3 Problem Formulation

We consider quadrupedal locomotion under joint torque degradation. At each time step $t$, the policy outputs a desired joint-position command $a^t$, which is converted by a low-level controller into nominal joint torques $\widehat{\boldsymbol{\tau}}^t$. Joint torque degradation is modeled as

$$\tau_i^t = (1 - d_i^t)\widehat{\tau}_i^t, \qquad d_i^t \in [0, 1], \tag{1}$$

where $d_i^t$ denotes the normalized degradation level of joint $i$, with $d_i^t = 0$ indicating nominal actuation and $d_i^t = 1$ complete torque loss. At the beginning of each episode, a degradation scenario $\xi = (\mathcal{F}, t_f, d_{\mathcal{F}}) \sim p_{\text{deg}}$ is sampled, where $\mathcal{F}$ specifies the degraded joints, $t_f$ is the degradation

onset time, and $d_{\mathcal{F}}$ contains their degradation levels. Once triggered, the sampled degradation remains fixed for the rest of the episode. We define $d^t$ as the degradation acting during the transition from time step $t$ to $t+1$; hence, degradation triggered at $t_f$ can first affect the observation at $t_f+1$. Through the coupled rigid-body and contact dynamics, local joint degradation can further influence whole-body locomotion.

The degradation state $d^t$ is not directly observable at deployment, and the current observation $o^t$ alone may be insufficient to identify the underlying actuation condition. We therefore consider a finite interaction history $\mathcal{H}^t = \left(o^{t-H:t}, a^{t-H:t-1}\right)$, which provides recent action–response information for inferring actuator degradation. Since $\mathcal{H}^t$ contains completed transitions only up to $(o^{t-1}, a^{t-1}, o^t)$, the degradation estimate available at control step $t$ is causally aligned with $d^{t-1}$.

Our objective is to learn a history-conditioned policy $a^t \sim \pi_\theta\left(\cdot \mid o^t, \mathcal{H}^t\right)$ that maximizes the expected discounted locomotion return over the distribution of degradation scenarios:

$$\max_\theta \ \mathbb{E}_{\xi \sim p_{\mathrm{deg}},\, \pi_\theta}\left[\sum_{t=0}^{T-1} \gamma^t r^t\right], \tag{2}$$

where $r^t$ denotes the locomotion reward and $\gamma \in [0,1)$ is the discount factor. Ground-truth degradation is available only during training as privileged information for degradation supervision and curriculum-based conditioning, but is never available to the policy at deployment.

# 4 METHOD

## 4.1 OVERVIEW

Figure 2 shows the overall framework of HAFT, which consists of degradation estimation, degradation-conditioned history aggregation, and context-conditioned flow policy. Given the current observation $o^t$ and recent history $\mathcal{H}^t$, a history encoder first extracts temporal features, from which the joint-wise degradation vector $\hat{d}^t$ is inferred. The current observation and inferred degradation jointly form the query for cross-attention over the historical features, producing a context representation for the current control step. Conditioned on this representation, a flow-matching policy generates the desired joint-position command $a^t$.

## 4.2 HISTORY-BASED DEGRADATION ESTIMATION

**Transition Encoding.** Given the interaction history $\mathcal{H}^t$, each completed transition is encoded together with its relative temporal offset:

$$z_j = f_{\mathrm{emb}}\left([o^j; a^j; o^{j+1}; \delta_j]\right), \qquad \delta_j = \frac{t-j-1}{H-1}, \quad j = t-H, \ldots, t-1, \tag{3}$$

where $\delta_j$ denotes the normalized temporal offset of transition $j$ relative to the current control step. Stacking the resulting embeddings as rows yields $Z^t \in \mathbb{R}^{H \times d_z}$, where $d_z$ denotes the embedding dimension. The resulting transition embeddings support both joint-wise degradation estimation and subsequent degradation-conditioned context aggregation.

**Joint-Wise Degradation Estimation.** The transition embeddings are mean-pooled and mapped to a joint-wise degradation estimate:

$$\bar{z}^t = \frac{1}{H}\sum_{j=t-H}^{t-1} z^j, \qquad \widehat{d}^t = \sigma\big(f_{\mathrm{deg}}(\bar{z}^t)\big), \tag{4}$$

where $f_{\mathrm{deg}} : \mathbb{R}^{d_z} \to \mathbb{R}^N$ is the degradation regressor and $\sigma(\cdot)$ denotes the element-wise sigmoid function. The output $\widehat{d}^t = [\widehat{d}^t_1, \ldots, \widehat{d}^t_N]^\top \in [0,1]^N$ denotes the degradation estimate available at control step $t$ and is supervised by the causally aligned target $d^{t-1}$, corresponding to the degradation during the most recent completed transition. Following degradation onset, $\mathcal{H}^t$ may contain both pre- and post-degradation transitions, and the estimate evolves as post-onset evidence accumulates.

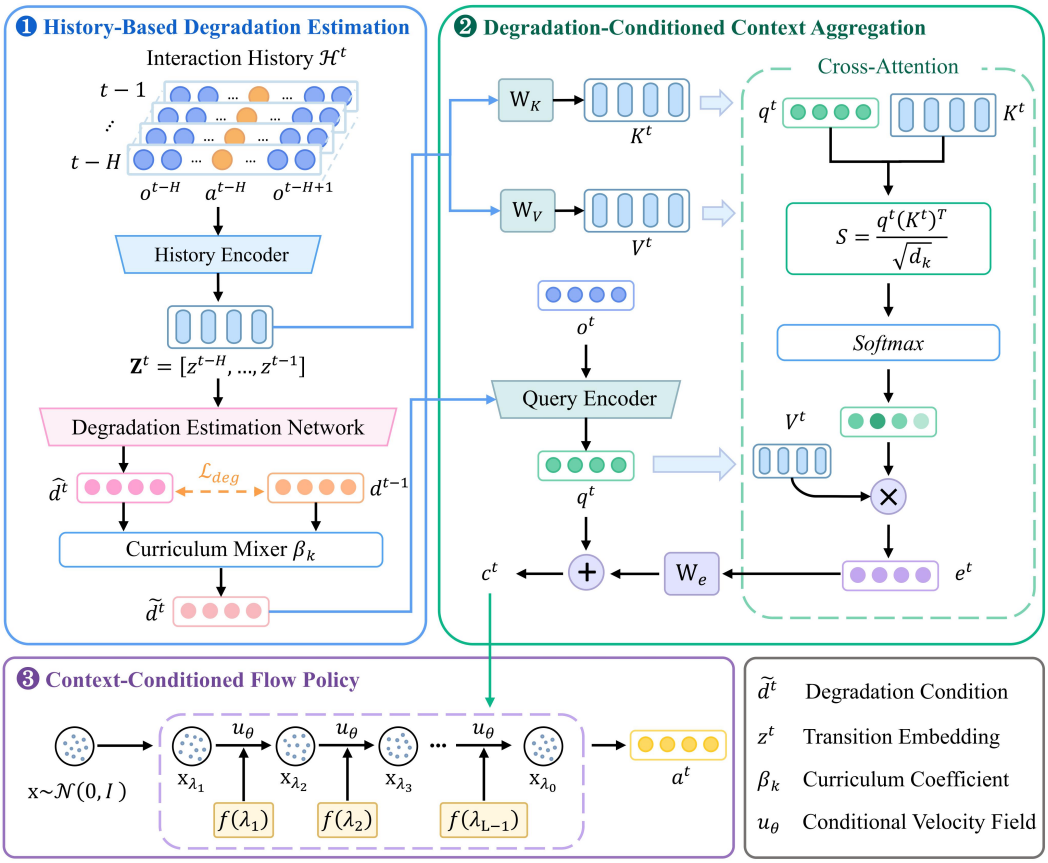


Figure 2: Overview of HAFT. The interaction history $\mathcal{H}^t$ is encoded into transition representations $Z^t$, whose pooled summary predicts the joint-wise degradation $\hat{d}^t$, while the full sequence is retained for context aggregation. During training, a curriculum combines $\hat{d}^t$ with the causally aligned target $d^{t-1}$ to form $\tilde{d}^t$, whereas $\hat{d}^t$ is used directly at deployment. Together with the current observation $o^t$, $\tilde{d}^t$ forms the query for cross-attention over $Z^t$, producing the context $c^t$. Conditioned on $c^t$, the flow policy maps $x_0^t \sim \mathcal{N}(0, I)$ to $x_1^t = a^t$.

To bridge the train–deployment gap in degradation conditioning, we employ a curriculum:

$$\tilde{d}^t = \beta_k d^{t-1} + (1 - \beta_k)\hat{d}^t, \tag{5}$$

where $\beta_k \in [0, 1]$ is annealed from 1 to 0 over training. Thus, the conditioning signal progressively shifts from the causally aligned ground-truth degradation $d^{t-1}$ to its estimate $\hat{d}^t$, and at test time $\tilde{d}^t = \hat{d}^t$.

### 4.3 Degradation-Conditioned Context Aggregation

The degradation estimate summarizes joint-wise actuation loss but need not capture all control-relevant information in recent interactions. We therefore retain the transition embeddings and aggregate them according to the current observation and the degradation condition. Specifically, the current observation $o^t$ and degradation condition $\tilde{d}^t$ are jointly encoded into a query representation:

$$q^t = f_q\left([o^t; \tilde{d}^t]\right), \tag{6}$$

where $\tilde{d}^t$ denotes the degradation condition supplied to the aggregation module.

The transition embeddings are projected into keys and values, and the query is used to perform multi-head cross-attention:

$$K^t = \mathbf{Z}^t W_K, \qquad V^t = \mathbf{Z}^t W_V, \qquad e^t = \mathrm{MHA}\left(q^t, K^t, V^t\right). \tag{7}$$

The attended history representation is then fused with the query:

$$c^t = \mathrm{LN}\left(q^t + W_e e^t\right), \tag{8}$$

where the projection matrices are learnable, $W_e$ maps the attended representation to the query dimension, and LN denotes layer normalization. The resulting context $c^t$ combines the current control condition with information selected from recent interactions.

### 4.4 Context-Conditioned Flow Policy

To parameterize the joint action distribution under varying degradation conditions, we employ a flow policy over the full action vector. Given the context $c^t$, the policy transforms a sample from a standard Gaussian distribution into a joint-position command through a context-conditioned velocity field. Let $x_\lambda^t \in \mathbb{R}^N$ denote the flow state, where $\lambda \in [0,1]$ is a continuous flow parameter distinct from the environment time index $t$. Its evolution is governed by:

$$\frac{dx_\lambda^t}{d\lambda} = u_\theta\left(x_\lambda^t, \lambda, c^t\right), \qquad x_0^t \sim \mathcal{N}(0, I), \tag{9}$$

where $u_\theta$ denotes the context-conditioned velocity field. For a fixed context $c^t$, integrating the flow from $\lambda = 0$ to $\lambda = 1$ transforms samples from the Gaussian base distribution into a conditional action distribution. The terminal state is taken as the desired joint-position command:

$$a^t = x_1^t. \tag{10}$$

The resulting distribution defines the conditional policy $\pi_\theta(\cdot \mid c^t)$. Since the velocity field operates on the complete action vector, the policy can represent dependencies among joint commands while adapting the action distribution to the degradation-conditioned context.

### 4.5 Training Objective

The degradation estimate available at control step $t$ is supervised by the degradation $d^{t-1}$ associated with the most recent completed transition:

$$\mathcal{L}_{\mathrm{deg}} = \mathbb{E}_t\left[\frac{1}{N}\left\|\widehat{d}^t - d^{t-1}\right\|_2^2\right], \tag{11}$$

where $N$ denotes the number of actuated joints.

For flow-policy optimization, we draw $M$ independent flow-matching samples for each rollout action $a^t$, where $M$ denotes the number of Monte Carlo interpolation samples used to estimate the policy objective. Specifically, we draw $\lambda_m \sim \mathcal{U}(0,1)$ and $\epsilon_m \sim \mathcal{N}(0, I)$, and construct the interpolated action as:

$$a_{\lambda_m}^t = (1-\lambda_m)\epsilon_m + \lambda_m a^t, \qquad m = 1, \ldots, M. \tag{12}$$

The corresponding target velocity is $a^t - \epsilon_m$, and the per-sample flow-matching error is defined as:

$$\ell_\theta^{(m,t)} = \left\|u_\theta\left(a_{\lambda_m}^t, \lambda_m, c^t\right) - \left(a^t - \epsilon_m\right)\right\|_2^2. \tag{13}$$

The sample-wise surrogate ratio is then defined from the change in flow-matching error:

$$\widehat{\rho}_\theta^{(m,t)} = \exp\left(\ell_{\theta_{\mathrm{old}}}^{(m,t)} - \ell_\theta^{(m,t)}\right), \tag{14}$$

where $\theta_{\mathrm{old}}$ denotes the frozen behavior-policy parameters used for rollout collection. The current and behavior-policy errors are evaluated using the same interpolation pair $(\lambda_m, \epsilon_m)$.

The advantage estimate $\widehat{A}^t$ is computed using generalized advantage estimation with the standard value function of the underlying actor–critic framework. Given $\widehat{A}^t$, the flow-policy objective is defined as:

$$\mathcal{J}_{\mathrm{flow}}(\theta) = \mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M}\psi_{\mathrm{ASPO}}\left(\widehat{\rho}_\theta^{(m,t)}, \widehat{A}^t\right)\right], \tag{15}$$

where $\psi_{\mathrm{ASPO}}$ denotes the asymmetric surrogate objective used for flow-policy optimization Yi et al. (2026).

The objective used to jointly optimize the policy components and degradation estimator is defined as:

$$\mathcal{L}_{\mathrm{HAFT}} = -\mathcal{J}_{\mathrm{flow}}(\theta) + w_{\mathrm{deg}}\mathcal{L}_{\mathrm{deg}}, \tag{16}$$

where $w_{\mathrm{deg}} > 0$ controls the relative contribution of degradation supervision.

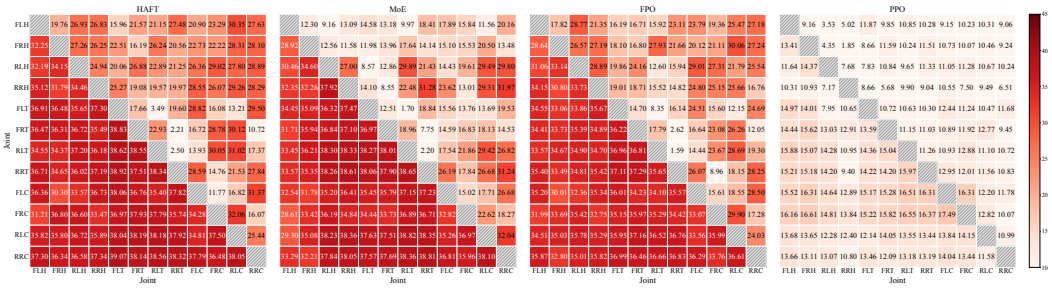


Figure 3: Performance comparison of HAFT and baseline methods under dual-joint torque degradation. Each off-diagonal cell reports the mean episodic reward for the joint pair specified by its row and column. The lower-left triangular region corresponds to partial torque degradation with $d = 0.5$ for both affected joints, whereas the upper-right triangular region corresponds to complete torque loss with $d = 1.0$. Hatched diagonal cells represent identical-joint pairs and are excluded from the dual-joint evaluation.

## 5 Experiments

In this section, we evaluate HAFT on a flat-terrain velocity-tracking task in NVIDIA Isaac Lab Mittal et al. (2025; 2023) using the Unitree Go2 quadruped under diverse joint torque degradation conditions. During training, the degraded-joint configuration, degradation onset time, and post-onset degradation levels are randomized. All methods share the same robot model, observation and action spaces, reward function, degradation distribution, and training budget, with 4096 parallel environments and 1500 training iterations. Our experiments address the following three questions: (1) How robust is HAFT across different degradation configurations and severity levels, particularly under dual-joint degradation? (2) How does HAFT maintain velocity tracking and reorganize support patterns under severe torque degradation? (3) What are the contributions of degradation-conditioned history aggregation and flow-based action generation?

**Baselines.** HAFT is compared with three baselines: **MoE** Turrisi et al. (2026), a mixture-of-experts reinforcement learning method that combines specialized policies through a gating mechanism; **FPO** McAllister et al. (2026), a flow-based on-policy reinforcement learning method using a conditional flow policy; and **PPO** Schulman et al. (2017), a conventional clipped-surrogate policy optimization method serving as the non-flow baseline.

### 5.1 Performance Evaluation and Analysis

Figure 3 compares HAFT with the baseline methods under dual-joint torque degradation in terms of mean reward. The results reveal substantial performance variations across degradation levels and affected-joint combinations, providing an overall measure of policy robustness under different actuator degradation configurations. Under $d = 0.5$, HAFT maintains relatively high episodic rewards across most degradation configurations. When both joints undergo complete torque loss, the performance of all methods decreases and the differences among degraded-joint combinations become more pronounced. Nevertheless, HAFT retains higher rewards over a broad range of dual-joint degradation configurations than the comparison methods, indicating improved robustness to severe multi-joint torque degradation. These results demonstrate that HAFT can adapt whole-body action coordination under substantial joint torque degradation. Additional single-joint evaluations covering different degradation locations and continuous degradation rates are provided in the appendix.

### 5.2 Case Study

Figure 4 compares the velocity-tracking responses of the evaluated methods after complete torque loss ($d = 1.0$) is introduced at $t = 3\,\mathrm{s}$. Under single-joint degradation of FLH, HAFT continues to track the commanded $x$-axis linear velocity, $y$-axis linear velocity, and yaw angular velocity with only limited transient disturbance. FPO preserves basic tracking but exhibits larger fluctuations, while MoE and PPO show more persistent deviations from the target commands. Under simultaneous degradation of RRT and FLC, HAFT recovers stable tracking of all three velocity components

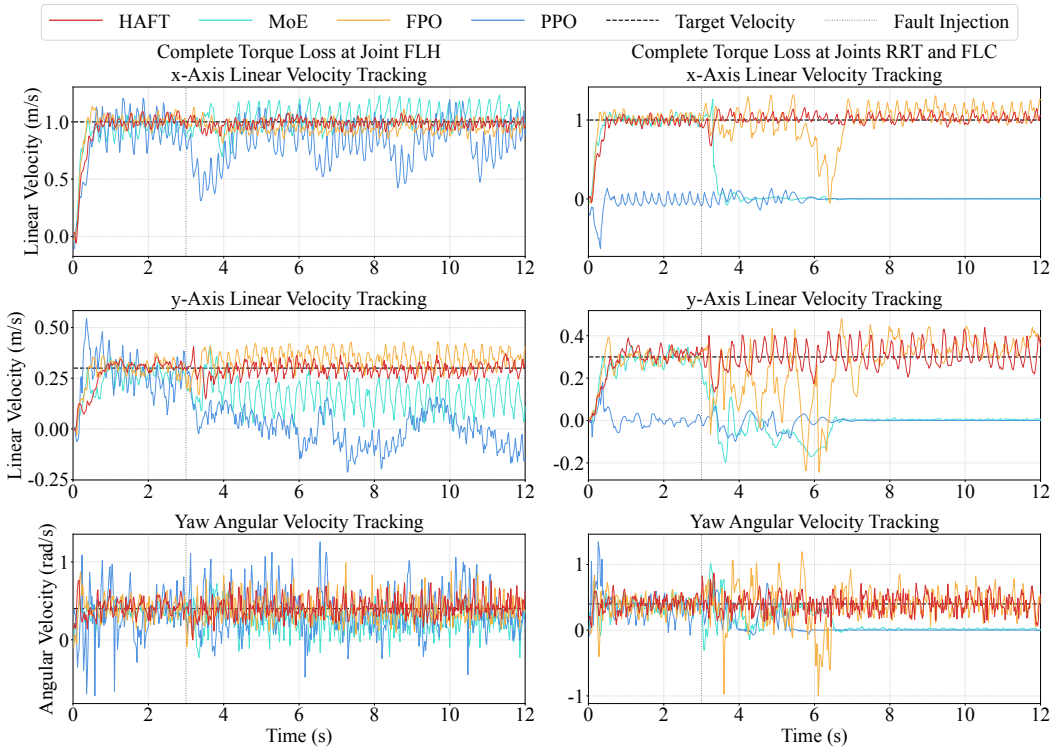


Figure 4: Velocity Tracking Performance of HAFT and Baseline Methods under Complete Torque Loss. The left column corresponds to single-joint degradation of FLH, while the right column corresponds to simultaneous degradation of RRT and FLC. Complete torque loss ($d = 1.0$) is introduced at $t = 3\,\mathrm{s}$. Horizontal dashed lines denote the commanded velocities, while vertical dotted lines indicate the degradation onset at $t = 3\,\mathrm{s}$.

after a short transient, whereas the comparison methods exhibit larger transient or sustained tracking errors. These results indicate that HAFT maintains command-tracking performance under severe joint torque degradation.

To further examine how this robustness is realized at the whole-body level, Fig. 5 analyzes the corresponding foot-contact patterns under different degradation severities. Under nominal actuation ($d = 0$), all four legs exhibit regular periodic contact. With partial degradation of FRT and FRC ($d = 0.5$), the front-right leg remains involved in support, while the contact timing of the remaining legs is redistributed. Under complete torque loss ($d = 1.0$), the front-right leg no longer contributes effective support after degradation onset, and the remaining three legs assume the dominant support roles while maintaining periodic contact. This progression shows that HAFT reorganizes whole-body support according to the available actuation capability, providing a behavioral explanation for the tracking robustness observed above.

### 5.3 Ablation Study and Sensitivity Analysis

As shown in Fig. 6, the full HAFT model achieves the highest mean reward in the later stage of training. HAFT_O converges to a lower reward after removing degradation-aware conditioning from history aggregation, indicating the benefit of incorporating the inferred degradation state into historical context aggregation. HAFT_F shows a larger performance drop when the flow policy is replaced by a diagonal-Gaussian policy, highlighting the contribution of flow-based action generation. These results suggest that both components contribute to the overall performance of HAFT. The right panel evaluates sensitivity to the curriculum starting point. Although the transition starts at iterations $0$, $200$, $400$, or $600$, all settings exhibit similar convergence behavior after fully switching to estimated degradation by iteration $1000$, indicating limited sensitivity within the evaluated range.

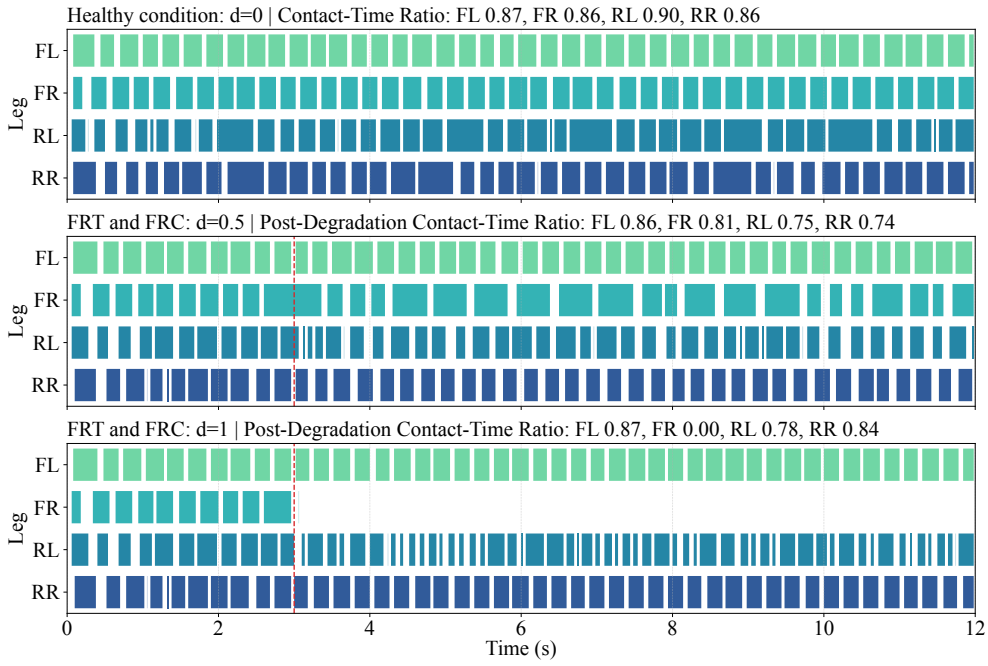


Figure 5: Foot Contact Timing and Contact-Time Ratios of HAFT under Different Joint Torque Degradation Rates. Colored bars indicate foot–ground contact intervals. The top panel shows healthy condition ($d = 0$), while the middle and bottom panels correspond to degradation of the FRT and FRC joints with $d = 0.5$ and $d = 1.0$, respectively. The red dashed line indicates the degradation onset at 3 s.

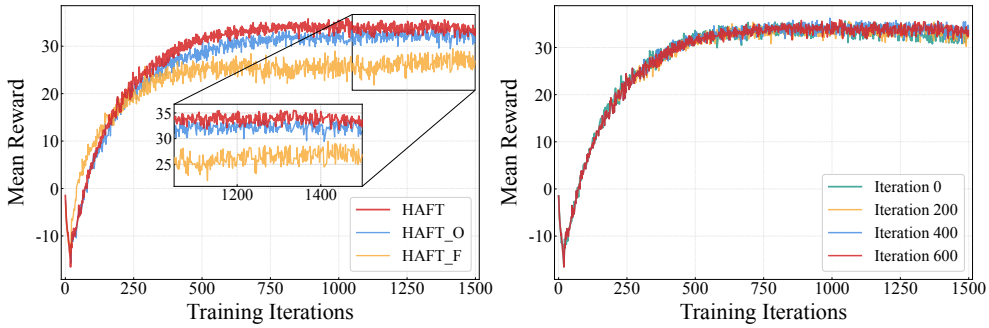


Figure 6: Ablation and curriculum-sensitivity studies of HAFT. *Left*: HAFT_O removes degradation-aware conditioning from history aggregation while retaining the interaction history, whereas HAFT_F replaces the flow policy with a diagonal-Gaussian policy while preserving the degradation-conditioned representation.*Right*: curriculum learning starts at iterations 0, 200, 400, or 600, with all settings fully transitioning to the estimated degradation state at iteration 1000.

## 6 Conclusion

In this paper, we proposed HAFT for quadrupedal locomotion under partially observable joint torque degradation, combining explicit degradation estimation, degradation-conditioned history aggregation, and flow-based whole-body action generation. Experiments under single- and multi-joint degradation settings demonstrate improved locomotion robustness and velocity-tracking performance over the evaluated baselines without access to ground-truth degradation at test time. Future work may extend this framework by incorporating predictive world models to anticipate degradation-induced dynamics and by coupling degradation-aware locomotion with vision-language-action (VLA) models, enabling actuator limitations to be considered jointly with visual perception, task semantics, and long-horizon decision making.

## AI USE STATEMENT

Large language models were used to assist with language polishing and minor adjustments to auxiliary code. All model-generated outputs were critically reviewed and, where appropriate, revised and validated by the authors before use. The authors retain full responsibility for the scientific content, implementation, experimental results, and conclusions of this work.

## REFERENCES

Ci Chen, Chao Li, Haojian Lu, Yue Wang, and Rong Xiong. Meta reinforcement learning of locomotion policy for quadruped robots with motor stuck. *IEEE Transactions on Automation Science and Engineering*, 22:5551–5565, 2024.

Zidong Chen, Zihao Guo, Peng Wang, ThankGod Itua Egbe, Yan Lyu, and Chenghao Qian. Densejump flow matching with non-uniform time scheduling for robotic policies: Mitigating multi-step inference degradation. In *IEEE International Conference on Robotics and Automation*, 2026.

Yangqing Fu, Yang Zhang, Qiyue Yang, Liyun Yan, Zhanxiang Cao, and Yue Gao. Contrastive forward prediction reinforcement learning for adaptive fault-tolerant legged robots. In *Conference on Robot Learning*, pp. 3285–3303, 2025.

Giovanbattista Gravina, Luca Rossini, Carlo Rizzardo, Arturo Laurenzi, and Nikos Tsagarakis. Learning fault-tolerant locomotion with adaptive gait timing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2026.

David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. ANYmal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024. doi: 10.1126/scirobotics.adi7566.

Taixian Hou, Jiaxin Tu, Xiaofei Gao, Zhiyan Dong, Peng Zhai, and Lihua Zhang. Multi-task learning of active fault-tolerant controller for leg failures in quadruped robots. In *IEEE International Conference on Robotics and Automation*, pp. 9758–9764, 2024.

Mincheol Kim, Ukcheol Shin, and Jung-Yup Kim. Learning quadrupedal locomotion with impaired joints using random joint masking. In *IEEE International Conference on Robotics and Automation*, pp. 9751–9757, 2024.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

Seunghyun Lee, I Made Aswin Nahrendra, Dongkyu Lee, Byeongho Yu, Minho Oh, Hyeonwoo Lee, and Hyun Myung. DreamFLEX: learning fault-aware quadrupedal locomotion controller for anomaly situation in rough terrains. In *IEEE International Conference on Robotics and Automation*, pp. 16001–16007, 2025.

Dikai Liu, Tianwei Zhang, Jianxiong Yin, and Simon See. Saving the limping: Fault-tolerant quadruped locomotion via reinforcement learning. *arXiv preprint arXiv:2210.00474*, 2022.

Min Liu, Deepak Pathak, and Ananye Agarwal. Locoformer: Generalist locomotion via long-context adaptation. In *Annual Conference on Robot Learning*, 2025.

Zeren Luo, Erdong Xiao, and Peng Lu. FT-Net: Learning failure recovery and fault-tolerant locomotion for quadruped robots. *IEEE Robotics and Automation Letters*, 8(12):8414–8421, 2023.

Yubiao Ma, Han Yu, Kai Guo, Boyang Xing, Xuemei Ren, and Dongdong Zheng. VPIES: Variational privileged information encoder as scaffold for legged locomotion learning. *IEEE Robotics and Automation Letters*, 11(4):5119–5126, 2026.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *International Conference on Learning Representations*, volume 2026, pp. 36352–36372, 2026.

Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, et al. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Munoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

Allen Ren, Justin Lidard, Lars Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, volume 2025, pp. 77288–77329, 2025.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pp. 91–100, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Giulio Turrisi, Ozan Pali, Luca Oneto, and Claudio Semini. Mixture-of-experts RL for fault-tolerant legged locomotion. *arXiv preprint arXiv:2606.25965*, 2026.

Xinyuan Wu, Wentao Dong, Hang Lai, Yong Yu, and Ying Wen. Adaptive control strategy for quadruped robots in actuator degradation scenarios. In *Proceedings of the Fifth International Conference on Distributed Artificial Intelligence*, pp. 1–13, 2023.

Tianyu Xu, Yaoyu Cheng, Pinxi Shen, and Lin Zhao. AcL: Action learner for fault-tolerant quadruped locomotion control. *arXiv preprint arXiv:2503.21401*, 2025.

Brent Yi, Hongsuk Choi, Himanshu Gaurav Singh, Xiaoyu Huang, Takara E. Truong, Carmelo Sferrazza, Yi Ma, Rocky Duan, Pieter Abbeel, Guanya Shi, Karen Liu, and Angjoo Kanazawa. Flow policy gradients for robot control. *arXiv preprint arXiv:2602.02481*, 2026.

Guojian Zhan, Letian Tao, Pengcheng Wang, Yixiao Wang, Yiheng Li, Yuxin Chen, Hongyang Li, Masayoshi Tomizuka, and Shengbo Eben Li. Mean flow policy with instantaneous velocity constraint for one-step action generation. In *International Conference on Learning Representations*, volume 2026, pp. 157132–157154, 2026.

Pei Zhang, Zhaobo Hua, Qiyu Qiu, and Jinliang Ding. FT-CPG: Learning central pattern generators for fault-tolerant quadruped locomotion under multi-joint failures. *IEEE Robotics and Automation Letters*, 10(7):6936–6943, 2025.

# A Appendix

This appendix supplements the main paper with the training procedure, implementation details, experimental settings, and additional evaluations of HAFT. These materials provide further methodological and empirical details beyond those included in the main text.

# B Complete Training Procedure of HAFT

**Algorithm 1** Training Procedure of HAFT

**Require:** Degradation distribution $p_{\mathrm{deg}}$, history length $H$, curriculum schedule $\{\beta_k\}$, flow samples $M$.

Initialize HAFT and the value function.
**for** training iteration $k = 1, 2, \ldots$ **do**
  Set $\beta_k$ and copy the current policy to the behavior policy $\theta_{\mathrm{old}}$.
  Collect rollout trajectories $\mathcal{D}_k$ under randomized degradation scenarios.
  **for** each environment step $t$ during rollout **do**
    Update the recent interaction history $\mathcal{H}^t$.
    Estimate the joint-wise degradation $\widehat{d}^t$ from $\mathcal{H}^t$.
    Form the degradation condition using $\beta_k$.
    Construct the degradation-conditioned context $c^t$ from $(o^t, \mathcal{H}^t, \widetilde{d}^t)$.
    Sample $a^t$ from the context-conditioned flow policy and execute it.
  **end for**
  Compute advantage estimates from $\mathcal{D}_k$.
  **for** each optimization epoch **do**
    Sample a minibatch from $\mathcal{D}_k$.
    Compute the degradation supervision loss $\mathcal{L}_{\mathrm{deg}}$.
    Estimate the flow-policy surrogate objective $\mathcal{J}_{\mathrm{flow}}$ using $M$ flow samples.
    Update HAFT using $\mathcal{L} = -\mathcal{J}_{\mathrm{flow}} + w_{\mathrm{deg}}\mathcal{L}_{\mathrm{deg}}$.
    Update the value function.
  **end for**
**end for**
**return** trained HAFT policy.

# C Additional Method Details

## C.1 History Initialization

The main text defines $\mathcal{H}^t$ using the $H$ most recent completed transitions. At the beginning of an episode, fewer than $H$ transitions are available. We therefore maintain a fixed-length history buffer and left-pad unavailable entries with $(o^0, 0, o^0)$. A binary mask

$$m_h^t \in \{0, 1\}, \qquad h = 1, \ldots, H, \tag{17}$$

indicates whether the $h$-th history entry corresponds to an observed transition.

The mask is used only to exclude padded entries from subsequent computations. In particular, the pooled transition representation is computed as:

$$\bar{z}^t = \frac{\sum_{h=1}^{H} m_h^t z_h^t}{\max\left(1, \sum_{h=1}^{H} m_h^t\right)}. \tag{18}$$

The same mask is applied to the cross-attention logits so that padded entries do not contribute to the attention normalization. Once $H$ completed transitions are available, all history entries are valid and the computation reduces to the formulation given in the main text.

### C.2 Curriculum-Based Degradation Conditioning

At the early stage of training, the estimated degradation $\widehat{d}^t$ may be insufficiently accurate to provide a reliable conditioning signal. Conversely, conditioning the policy exclusively on the ground-truth degradation $d^t$ would introduce a discrepancy between training and test-time inference. We therefore use the deterministic curriculum interpolation introduced in the main text, and specify its schedule below.

Let $n_k$ denote the cumulative number of environment interaction steps before training iteration $k$. The curriculum coefficient $\beta_k$ is defined as:

$$\beta_k = \begin{cases} 1, & n_k < N_{\text{start}}, \\ \dfrac{N_{\text{end}} - n_k}{N_{\text{end}} - N_{\text{start}}}, & N_{\text{start}} \le n_k < N_{\text{end}}, \\ 0, & n_k \ge N_{\text{end}}, \end{cases} \tag{19}$$

where $N_{\text{start}}$ and $N_{\text{end}}$ denote the cumulative environment steps at which the curriculum transition begins and ends, respectively.

Given $\beta_k$, the degradation condition used by the policy is constructed as:

$$\widetilde{d}^t = \beta_k d^{t-1} + (1 - \beta_k)\widehat{d}^t. \tag{20}$$

The coefficient $\beta_k$ is held fixed during rollout collection and the corresponding policy updates within training iteration $k$. Before $N_{\text{start}}$, the policy is conditioned on the causally aligned ground-truth degradation $d^{t-1}$. During the transition interval, the contribution of $\widehat{d}^t$ increases linearly as $\beta_k$ decreases. After $N_{\text{end}}$, the conditioning signal depends entirely on the estimated degradation.

At test time, ground-truth degradation is unavailable, and the degradation condition is therefore given by:

$$\widetilde{d}^t = \widehat{d}^t. \tag{21}$$

### C.3 Flow Policy Sampling and Optimization Details

The main text defines the conditional flow from a Gaussian base distribution at $\lambda = 0$ to the action distribution at $\lambda = 1$. This appendix specifies the numerical integration used for action sampling and the organization of flow samples during policy optimization.

For action generation, the interval $\lambda \in [0, 1]$ is discretized into $L$ uniform steps:

$$\Delta\lambda = \frac{1}{L}, \qquad \lambda_l = l\Delta\lambda, \qquad l = 0, \dots, L. \tag{22}$$

Starting from $x_0^t \sim \mathcal{N}(0, I)$, the conditional flow is integrated using a first-order explicit Euler update:

$$x^t_{\lambda_{l+1}} = x^t_{\lambda_l} + \Delta\lambda\, u_\theta\left(x^t_{\lambda_l}, \lambda_l, c^t\right), \qquad l = 0, \dots, L-1. \tag{23}$$

After $L$ integration steps, the terminal state defines the policy action:

$$a^t = x_1^t. \tag{24}$$

For policy optimization, the current policy is copied to a frozen behavior policy before each rollout phase:

$$\theta_{\text{old}} \leftarrow \theta. \tag{25}$$

Trajectories are subsequently collected using the behavior policy, which remains fixed throughout the corresponding parameter-update epochs. For each rollout timestep $t$, $M$ independent interpolation pairs are sampled:

$$\lambda_m \sim \mathcal{U}(0, 1), \qquad \epsilon_m \sim \mathcal{N}(0, I), \qquad m = 1, \dots, M. \tag{26}$$

For each $m$, the same pair $(\lambda_m, \epsilon_m)$ is used to evaluate the current- and behavior-policy flow-matching errors defined in Sec. 4.5. The resulting paired errors are used to construct the sample-wise

surrogate ratio in Eq. 14. Using common interpolation samples ensures that the comparison between the current and behavior policies is not confounded by independent Monte Carlo draws.

All $M$ flow samples associated with the same rollout timestep share the corresponding advantage estimate $\widehat{A}^t$, while their surrogate ratios are evaluated independently before aggregation in $\mathcal{J}_{\text{flow}}$. The behavior-policy parameters $\theta_{\text{old}}$ remain fixed throughout these updates and are refreshed only before the next rollout phase.

# D Implementation Details

## D.1 Network Architecture and Hyperparameters

The detailed network architecture and optimization hyperparameters used in the experiments are summarized in Tables 1 and 2. Unless otherwise specified, the same common training settings are used for all comparison methods to ensure a fair evaluation.

Table 1: Network architecture configuration of HAFT

| **Configuration Item** | **Value** | **Configuration Item** | **Value** |
|---|---|---|---|
| History window size | 10 | Action flow integration steps | 64 |
| Interaction token dimension | 128 | Context fusion dimension | 128 |
| Cross-attention heads | 4 | Optimizer | Adam |
| Actor hidden-layer width | 256 | Critic hidden-layer width | 768 |

Table 2: Experimental hyperparameter settings of HAFT

| **Hyperparameter** | **Value** | **Hyperparameter** | **Value** |
|---|---|---|---|
| Number of actuated joints | 12 | Training iterations | 1500 |
| Number of parallel environments | 4096 | Rollout length | 24 |
| Learning rate | 0.0001 | Discount factor $\gamma$ | 0.99 |

# E Additional Experimental Results

## E.1 Single-Joint Torque-Degradation Evaluation

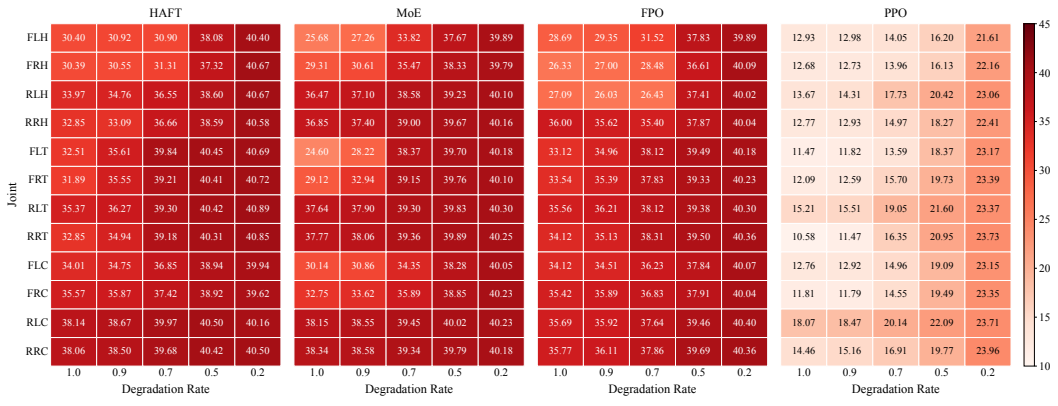


Figure 7: Performance Comparison of HAFT and Baseline Methods under Single-Joint Torque Degradation. Each cell reports the average episodic reward for a specific degraded joint and degradation rate. A larger degradation rate $d$ indicates more severe torque degradation, with $d = 1.0$ corresponding to complete torque loss.

Figure 7 reveal substantial differences in sensitivity across joints under the same degradation rate, indicating that the effect of actuator degradation strongly depends on the location of the affected joint.

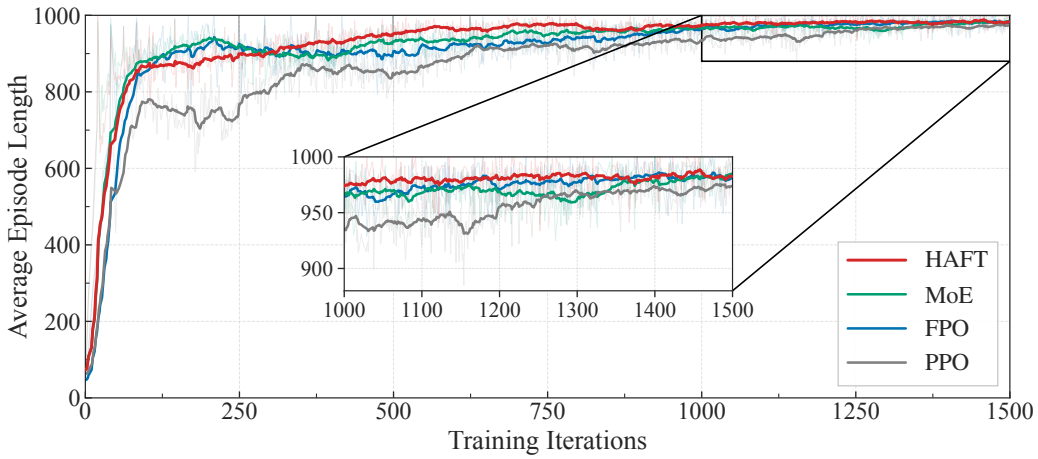


Figure 8: Average Episode Length of HAFT and Baseline Methods during Training. The inset provides an enlarged view of the late-stage training interval from iteration 1000 to 1500. The maximum episode length is 1000 simulation steps.

As the degradation rate increases, several baseline methods exhibit increasingly pronounced performance deterioration for specific joints. In contrast, HAFT maintains relatively high rewards over most joints and degradation rates, with smaller performance variations across degradation configurations. MoE and FPO++ remain competitive under several conditions but exhibit more pronounced low-performance regions, whereas PPO achieves lower overall rewards and is more sensitive to changes in degradation location and severity.

Together with the dual-joint results reported in the main text, these results show that HAFT remains robust across both isolated and concurrent actuator degradation.

### E.2 Training Stability

Figure 8 compares the average episode length of HAFT and the baseline methods throughout training. During the early stage of training, HAFT, MoE, and FPO++ exhibit relatively rapid improvements, whereas PPO improves more slowly and shows larger fluctuations during the middle stage. HAFT maintains a comparatively high average episode length with a smoother convergence trend over the middle and later stages. Although MoE and FPO++ eventually approach comparable episode durations, they exhibit more pronounced intermediate fluctuations. These results indicate that HAFT progressively reduces premature episode termination caused by unstable locomotion and maintains long-duration locomotion more consistently during training.