

Self-Healing Action Verification against Instruction Tampering and Sensor Failures in Humanoid Robots

Sanford Bramlett¹, Enoch Hwang¹, Jueming Hu³, Jielun Zhang², and Fuhao Li¹

Abstract—As humanoid robots are increasingly used in complex real-world environments, reliable verification of robot actions becomes more important. Action verification should not only check whether the executed action matches the intended instruction, but also keep reliable performance when motor-sensor observations are affected by failures. Traditional instruction authentication mainly protects the instruction itself and cannot directly verify the final physical action of the robot. In addition, the sensing information used to observe robot execution may also become unreliable. In this paper, we propose a self-healing motor-based action verification framework that combines explainable sensor selection, temporal masked reconstruction, and multiscale action recognition. First, Local Interpretable Model-Agnostic Explanations (LIME) are used to identify motor encoders that provide important information for action discrimination. Then, the selected encoder signals are processed by a temporal masked reconstruction model to recover unreliable joint observations based on temporal continuity and cross-joint coordination. After that, a multiscale gallery recognizer with zero trainable classifier parameters is used to infer the physical action executed by the robot. The inferred action is finally compared with the trusted intended instruction to detect instruction-execution inconsistency. The proposed framework is evaluated on a Booster Robotics K1 Education humanoid robot under multiple sensor-failure modes and different corruption levels, and is compared with several existing methods. The proposed method achieves average accuracies of 98.4% and 94.8% for single-action and compound-action recognition, respectively. Under different sensor-failure levels, the proposed self-healing method achieves an average recognition accuracy of 92.6%.

Index Terms—Humanoid robots, instruction tampering, sensor failure, self-healing systems, action recognition, motor encoders, masked reconstruction, explainable artificial intelligence.

I. INTRODUCTION

HUMANOID robots are increasingly used in manufacturing, service, healthcare, and other complex environments, making the safety, reliability, and verifiability of robot actions more important [1]–[3]. For networked humanoid robots, protecting motion instructions only at the communication layer cannot guarantee that the final physical execution matches the intended behavior [4]–[6]. Abnormalities in the control path, actuator, or intermediate software layer may still cause execution errors [6], [7]. Therefore, independent physical evidence is needed to verify the final robot behavior. As illustrated in Fig. 1, motor-encoder observations provide

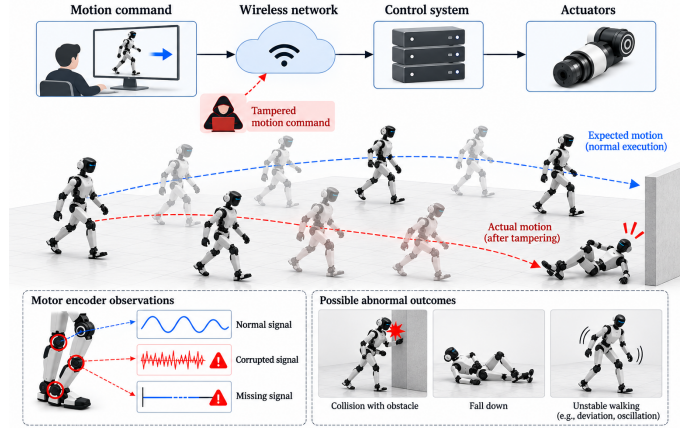


Fig. 1. Motivation of self-healing motor-based action verification. The executed action is verified against the trusted intended instruction through motor-encoder evidence that remains reliable under sensor failures.

such physical evidence for action verification. Motor encoders directly record joint motion and can provide direct evidence of the physical robot state [8], [9]. In addition, different robot actions usually produce different coordinated motion patterns across multiple joints, which provides useful information for action recognition [10], [11]. Therefore, motor encoder signals can be used as physical evidence of the actual robot execution. However, this execution-side verification should also remain reliable under sensor failures, because the physical evidence itself depends on sensor observations [6], [12], [13].

Existing robot security studies have explored attack detection using independent state information or physical consistency. Guerrero-Higueras et al. [14] analyzed cyberattacks against an indoor real-time localization system for mobile robots and showed that attack conditions can produce distinguishable changes in localization measurements. Rivera et al. [15] proposed an autoencoder-based method to detect sensor spoofing attacks using robot-state reconstruction errors, with a focus on LiDAR observations. Varshosaz et al. [16] used visual odometry as an independent motion reference and compared it with GPS trajectories to detect UAV GPS spoofing. Man et al. [17] used spatiotemporal consistency between object classes and motion tracks to detect perception misclassification attacks in autonomous systems. These studies show that independent physical or state information can provide useful evidence for detecting abnormal robot behavior. However, they mainly focus on localization, perception, or robot-state abnormalities, rather than directly verifying whether the final executed action matches the trusted intended

Sanford Bramlett, Enoch Hwang, and Fuhao Li are with the Computer Science Department, La Sierra University, Riverside, CA 92505 USA (e-mail: {sbra977, ehwang, fli}@lasierra.edu). Jielun Zhang is with the School of Electrical Engineering & Computer Science, University of North Dakota, Grand Forks, ND 58202 USA (e-mail: jielun.zhang@und.edu). Jueming Hu is with The University of Oklahoma, Norman, OK 73019 USA (e-mail: jueming@ou.edu). Corresponding author: Fuhao Li and Jielun Zhang.

instruction. In addition, the physical observations used for execution verification may themselves become unreliable because of sensor failures or malicious interference, reducing the reliability of action recognition and verification. Different methods have been studied to recover unreliable sensor data and corrupted time-series observations. For example, the Savitzky–Golay method [18] uses local polynomial fitting to smooth signals while preserving local variations. For more complex temporal dependencies, Temporal Convolutional Networks [19] use causal and dilated convolutions to model long-range sequential relationships. However, these methods mainly focus on signal smoothing, numerical recovery, or sequence modeling performance. A low reconstruction error does not guarantee that action boundaries, motion directions, and cross-joint temporal relationships are correctly preserved. It also does not guarantee that the recovered signals can support reliable semantic action recognition. Therefore, recovering useful physical motion information under unreliable sensor observations while maintaining stable instruction-execution consistency verification remains an important problem.

To tackle these issues, we propose a self-healing motor-based action verification framework that integrates action recognition, sensor recovery, and instruction-execution consistency verification. First, LIME is used to identify motor encoders that provide important information for action discrimination. Then, a multiscale gallery-based recognizer is built using the selected encoder signals, and motion information from different temporal scales is used to infer the action physically executed by the robot. This recognition process does not require trainable classifier parameters. For possible sensor failures, a temporal masked self-healing model is used to recover unreliable motor-encoder observations based on temporal continuity and cross-joint relationships. The recovered signals are then used for action recognition. Finally, the inferred action is compared with the trusted intended instruction to detect instruction-execution inconsistency. The proposed framework is evaluated on a Booster Robotics K1 Education humanoid robot under different sensor-failure conditions. Experimental results show that the proposed method achieves good performance for both single-action and compound-action recognition and maintains reliable recognition and verification under sensor failures. The main contributions are as follows: 1) We propose a self-healing motor-based action verification framework that directly verifies physical robot execution against a trusted intended instruction. 2) We develop a multiscale gallery-based action recognition method to infer the physically executed action without trainable classifier parameters. 3) We develop a temporal masked self-healing method to recover unreliable motor-encoder observations based on temporal continuity and cross-joint relationships for sensor failures. 4) We evaluate the proposed framework using real motor-encoder data collected from a humanoid robot under different sensor failures, corruption levels, and controlled instruction-substitution scenarios.

The remainder of this paper is shown as follows. Section II introduces the related work. Section III shows the system and threat model. Section IV describes the proposed method. Section V presents the experimental evaluation. Section VI shows the discussion, and Section VII concludes the paper.

II. RELATED WORK

A. Traditional Action Recognition Methods

Traditional action recognition relies on template matching or on hand-crafted features coupled with shallow classifiers. The simplest and most interpretable route is template matching, which provides low-latency decisions directly from reference recordings. For instance, the author in [20] introduced nearest-neighbor pattern classification, which matches a query against stored reference examples and remains the standard motion-matching baseline, where every decision can be traced to real recorded examples rather than to learned weights. Template matching alone is often insufficient, however, because real executions vary in phase, duration, and style. Shallow classifiers address this variability by separating engineered features with decision boundaries instead of raw distances. Following this idea, the author in [21] proposed random forests, which aggregate many decision trees into stable feature importances and tolerate nonlinear feature interactions without extensive tuning. Along similar lines, the author in [22] proposed support-vector networks, which separate classes with maximum-margin hyperplanes in kernel feature spaces and remain effective when the engineered descriptor is high-dimensional. In the activity-recognition literature these shallow pipelines are canonical. The author in [23] surveyed body-worn inertial activity recognition and organized the pipeline around segmentation, feature extraction, and temporal classification, noting that statistics over short windows already carry strong action evidence. While traditional recognition research is broad, very few studies treat the sensors as fallible evidence or ask whether recognition remains reliable when the sensing channel itself is corrupted. Many previous studies have nevertheless established that multijoint coordination carries strong action semantics and that engineered velocity, range, and timing features capture much of it.

B. Deep Learning-Based Action Recognition

Deep learning removes the need for hand-crafted features by learning representations directly from coordinated motion. The first widely adopted deep models for wearable activity recognition combined convolutional and recurrent processing. The author in [24] introduced DeepConvLSTM, which stacks convolutional feature extraction over recurrent sequence modeling so that local motion shape and long-range temporal structure are captured within one network. Extending this comparison of architectures, the author in [25] compared deep, convolutional, and recurrent models across wearable datasets and quantified their accuracy-complexity trade-offs, showing that carefully regularized simple recurrent models often match deeper alternatives. A comprehensive survey by the author in [11] documented deep learning for sensor-based activity recognition and highlighted persistent sensitivity to sensor placement and subject variation. Beyond wearable settings, articulated motion calls for explicit joint structure, and the author in [10] proposed spatial-temporal graph convolutional networks, which represent joints as a structured body rather than independent channels and propagate features over both spatial and temporal edges. Attention-based models take the

opposite route and let data decide the dependencies. The author in [26] introduced self-attention for sequence modeling, and the author in [27] adapted Transformer encoders to multivariate time series, capturing long-range interaction without recurrent state. These studies collectively show that deep models improve recognition when labeled data are sufficient. However, they typically require trained parametric classifiers and assume the input sensors are reliable, so a corrupted sensing channel silently degrades every layer of the model.

C. Data Recovery

Data recovery methods repair missing, noisy, or corrupted values so that downstream decisions can continue. Classical approaches repair each channel from its own local structure. The author in [18] proposed Savitzky–Golay filtering, which fits a local polynomial so that smoothing preserves local variation, and the author in [28] proposed Wiener smoothing under a statistical signal model. Outliers and low-rank corruption are handled by the robust suppression of the author in [29] and the principal-component repair reviewed in [30]; nearest-neighbor rules borrow values from similar reference observations [20], and moving averages, median filters, exponential smoothing, and linear interpolation remain widely used baselines. These methods are simple and inexpensive, however, they struggle when faults are nonlinear, multichannel, or severe, and none guarantees that the repaired signal preserves the decision that consumes it.

Deep models instead learn temporal and cross-variable structure from data. The author in [31] proposed GRU-D, which folds missingness and time decay into recurrent prediction, and the author in [32] proposed bidirectional recurrent imputation. The author in [33] proposed imputation with generative adversarial networks, and the author in [34] surveyed missing-data patterns and their effect on classification. Attention-based imputation follows this trend. The author in [35] proposed self-attention-based imputation, the author in [36] proposed conditional score-based diffusion imputation, and attention models extend imputation to irregular sampling [37]. Masked modeling generalizes reconstruction to representation learning [38], [39], while temporal convolutional networks [19] and convolutional autoencoders serve as learned denoisers for multichannel signals. These studies collectively show that learned reconstruction can restore degraded signals to low numerical error; how to make recovery serve a safety-relevant downstream decision rather than fidelity alone remains an open research opportunity.

In this work, we propose a self-healing motor-based action-verification pipeline that turns these recovery models toward a safety-relevant decision. A temporal masked reconstruction model restores unreliable joint observations from temporal continuity and cross-joint coordination under a curriculum of finite-valued sensor faults, and the reconstructed sequence is passed to an unchanged multiscale action recognizer, so recovery succeeds only if the executed action can again be inferred. The inferred action is then compared with a trusted intended instruction, promoting numerical reconstruction from an end in itself to the evidence channel of behavioral verification.

III. SYSTEM AND PROBLEM FORMULATION

A. Instruction-Execution Verification

Let $c_t \in \mathcal{C}$ denote the intended action at time t , where $\mathcal{C}$ is the set of the five locomotion actions. The intended sequence is retained by a trusted supervisory component before the command enters a potentially alterable control path. Let $a_t \in \mathcal{C}$ be the action physically executed by the robot and $\mathbf{x}_t \in \mathbb{R}^J$ the corresponding joint-angle observation from J selected encoders. The verification problem is to estimate a_t independently as $\hat{a}_t = f(\mathbf{x}_{1:t})$ and compare it with c_t .

$$z_t = \mathbb{I}[\hat{a}_t \neq c_t], \quad (1)$$

where $z_t = 1$ denotes an instruction–execution inconsistency.

A consistency decision is meaningful only if the physical-evidence path is sufficiently reliable. We therefore model the received observation as

$$\tilde{\mathbf{x}}_t = \mathcal{F}(\mathbf{x}_t; \mathbf{m}_t, q_t), \quad (2)$$

where $\mathbf{m}_t \in \{0, 1\}^J$ marks affected entries, q_t denotes a failure mode, and $\mathcal{F}$ applies the corresponding transformation. The self-healing function g estimates $\hat{\mathbf{x}}_{1:T} = g(\tilde{\mathbf{x}}_{1:T})$, after which f infers the executed action from the reconstructed sequence.

The complete information flow is illustrated in Fig. 2. The intended command reaches the verifier through a trusted reference path, while robot execution is assessed independently through joint encoders, self-healing, and multiscale recognition. The method can then trigger an alarm or a safe response, but the design of that response is outside the present scope.

B. Threat and Failure Model

The threat model assumes that an instruction may be modified between the trusted supervisor and the robot controller. The adversary does not control the trusted reference used in (1). If both the command channel and reference are replaced by the same value, no behavioral consistency method can recover the original intent. The model also assumes that enough joint motion remains to infer cross-joint structure; complete and persistent compromise of all physical evidence is outside scope.

Sensor unreliability may be accidental or adversarial. For clean value x , training standard deviation σ , and affected-entry mask m , failure severity ρ is the fraction of joint-time entries with $m = 1$. Seven failure modes are evaluated. Dropout produces missing values ($x' \leftarrow \text{NaN}$), stuck-at-zero produces $x' \leftarrow 0$, excessive noise produces $x' \leftarrow x + \mathcal{N}(0, (2\sigma)^2)$, offset drift produces $x' \leftarrow x \pm 2\sigma$, gain error produces $x' \leftarrow 2.5x$, polarity inversion produces $x' \leftarrow -x$, and transient spikes produce $x' \leftarrow x \pm 5\sigma$. Missing values disclose their locations through nonfinite encoding. Finite-valued faults do not disclose their affected locations to the recovery model.

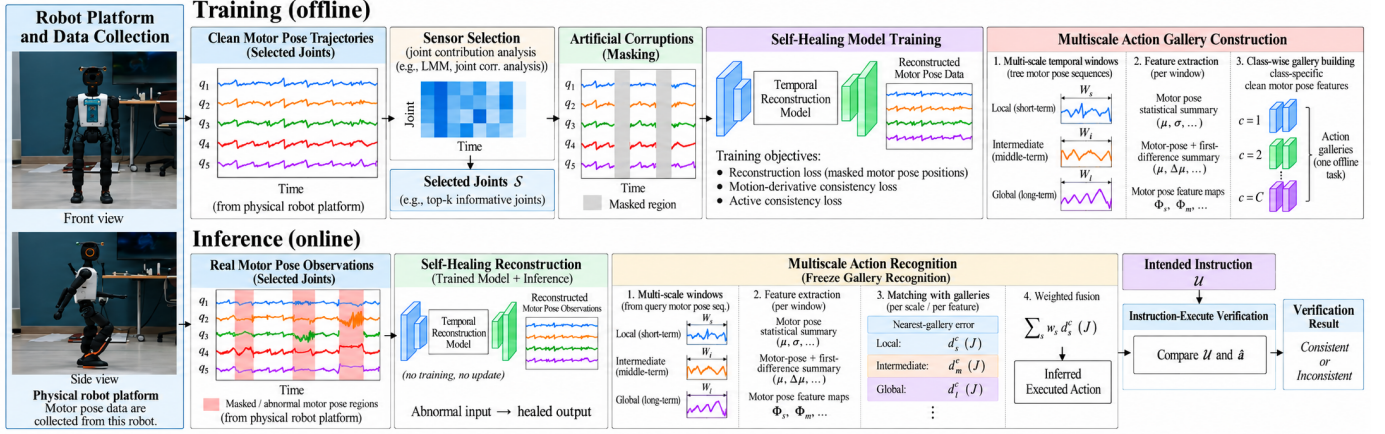


Fig. 2. Overview of the proposed self-healing instruction-verification framework. The trusted intended instruction and the robot’s physical evidence follow separate paths. Joint-encoder observations are processed by sensor self-healing and multiscale action recognition before the inferred action is compared with the trusted instruction.

C. Design Objectives

The method is designed around four objectives. First, motor selection should identify sensors that carry action evidence rather than assume that all joints are useful. Second, action recognition should handle different temporal scales and compound-session transitions. Third, self-healing should restore both numerical fidelity and motion semantics under unknown failure modes. Fourth, verification should remain separate from the potentially altered command path. These objectives lead to the four-stage methodology described next.

IV. METHODOLOGY

A. Overview of the Proposed Method

This section introduces the proposed self-healing action-verification method, as shown in Fig. 2. The method includes an offline configuration stage and an online verification stage, and it contains four main modules. In the offline stage, clean labeled motor-encoder data are first used to extract velocity, acceleration, and motion-range features. LIME is then applied to measure the contribution of each joint encoder to action discrimination, and a compact set of informative encoders is selected. Based on the selected encoders, multiscale motion features are extracted to build class-specific reference galleries, and k-center selection is used to keep representative samples. At the same time, different types and levels of sensor faults are added to clean motion windows to train the self-healing model, so that it can learn temporal continuity and cross-joint relationships. In the online stage, the selected encoder signals are first processed by the self-healing model to reconstruct unreliable observations, and overlapping windows are combined to obtain the healed motion sequence. The multiscale recognition module then compares the current motion features with the reference galleries and fuses the matching results from different temporal scales to infer the physically executed action. Finally, the inferred action is compared with the trusted intended instruction. If they are different, an instruction-execution inconsistency is reported. The details of each module are introduced in the following subsections.

B. Explainable Motor Encoder Selection

Let $\mathcal{J}_{\text{all}}$ denote the set of candidate encoders, $M = |\mathcal{J}_{\text{all}}|$, and $\mathbf{X} \in \mathbb{R}^{H_s \times M}$ denote a clean motion window. For each joint, we extract three complementary motion descriptors based on velocity, acceleration, and motion range:

$$v_j = \text{std}_{t \in \mathcal{T}_v} (X_{t,j} - X_{t-1,j}), \quad (3)$$

$$a_j = \text{std}_{t \in \mathcal{T}_a} (X_{t,j} - 2X_{t-1,j} + X_{t-2,j}), \quad (4)$$

$$r_j = \max_t X_{t,j} - \min_t X_{t,j}, \quad (5)$$

where $\mathcal{T}_v$ and $\mathcal{T}_a$ contain the valid temporal indices. The descriptors of all candidate joints are concatenated into a window-level feature vector and used to train an action-discriminative model h .

LIME is then applied to explain the local predictions of h for sampled training windows [40]. Let $\beta_{n,j}^v$, $\beta_{n,j}^a$, and $\beta_{n,j}^r$ denote the local LIME coefficients of the three descriptors of joint j for explanation sample $n \in \mathcal{S}_e$. We use the absolute coefficients because both positive and negative contributions indicate the importance of a joint. The importance of joint j is defined as

$$I_j = \frac{\frac{1}{|\mathcal{S}_e|} \sum_{n \in \mathcal{S}_e} (|\beta_{n,j}^v| + |\beta_{n,j}^a| + |\beta_{n,j}^r|)}{\max_{\ell \in \mathcal{J}_{\text{all}}} \frac{1}{|\mathcal{S}_e|} \sum_{n \in \mathcal{S}_e} (|\beta_{n,\ell}^v| + |\beta_{n,\ell}^a| + |\beta_{n,\ell}^r|)}. \quad (6)$$

The normalized score is used to select the informative subset with a fixed relative-importance threshold τ :

$$\mathcal{J}^* = \{j \in \mathcal{J}_{\text{all}} : I_j \geq \tau\}. \quad (7)$$

An encoder is retained when its aggregated importance reaches the predefined fraction τ of the strongest encoder.

C. Multiscale Gallery-Based Action Recognition

To recognize the physical action executed by the robot from the selected motor encoders, we develop a multiscale gallery-based recognition method. Let $J = |\mathcal{J}^*|$ and $\mathcal{L}$ denote a set of causal window lengths. Using only one temporal scale may be sensitive to motion speed and action phase. Therefore, motion

information from multiple temporal scales is used for action recognition.

For a motion window $\mathbf{W} \in \mathbb{R}^{\ell \times J}$, $\ell \in \mathcal{L}$, the statistical motion feature is defined as

$$\phi(\mathbf{W}) = [\mu, \sigma, x_{\min}, x_{\max}, x_{\max} - x_{\min}, \Delta x, \overline{|v|}], \quad (8)$$

where each term is computed for every selected joint and then concatenated. Here, Δx denotes the difference between the last and first positions in the window, and $\overline{|v|}$ denotes the mean absolute first difference. The features are normalized using statistics from the clean reference data.

For each action class c and temporal scale ℓ , a clean feature gallery $\mathcal{G}_{c,\ell}$ is constructed offline. Given the current causal window $\mathbf{X}_{t-\ell+1:t}$, its distance to class c is defined by the nearest reference feature:

$$d_\ell^c(t) = \min_{\mathbf{g} \in \mathcal{G}_{c,\ell}} \|\phi(\mathbf{X}_{t-\ell+1:t}) - \mathbf{g}\|_2^2. \quad (9)$$

Since the distance ranges may be different across temporal scales, the class distances are normalized at each frame, producing $\bar{d}_\ell^c(t)$. The normalized distances from multiple scales are then fused as

$$D^c(t) = \sum_{\ell \in \mathcal{L}} w_\ell \bar{d}_\ell^c(t), \quad \hat{a}_t = \arg \min_{c \in \mathcal{C}} D^c(t), \quad (10)$$

where $w_\ell \geq 0$ and $\sum_{\ell \in \mathcal{L}} w_\ell = 1$. The action with the smallest fused distance is selected as the physically executed action. Since each motion window ends at the current frame t , the recognition process is causal.

Consecutive motion windows are highly redundant. Therefore, each class-specific gallery is condensed offline using the classical k -center algorithm [41]. The procedure starts from the sample closest to the feature mean and repeatedly selects the sample farthest from the current retained set. This process keeps representative motion patterns while reducing redundant gallery samples.

D. Temporal Self-Healing of Motor Observations

To maintain reliable action recognition when motor observations become unreliable, we introduce a temporal self-healing module to reconstruct corrupted encoder signals. Let $\mathbf{X} \in \mathbb{R}^{H_r \times J}$ denote a clean motion window from the selected encoders. During training, a failure type $q \sim p_q$ and a corruption ratio $\rho \sim p_\rho$ are sampled, and a binary corruption mask $\mathbf{M} = (m_{t,j})$ is generated with $m_{t,j} \sim \text{Bernoulli}(\rho)$. The corrupted window is written as

$$\tilde{\mathbf{X}} = \mathcal{F}(\mathbf{X}; \mathbf{M}, q), \quad (11)$$

where $\mathcal{F}$ applies the corresponding failure transformation of Eq. (2) to the entries selected by $\mathbf{M}$.

After per-joint normalization, the recovery model receives both the motion values and an observation indicator. Let

$$\mathbf{O} = \mathbb{I}[\tilde{\mathbf{X}} \text{ is observed}], \quad (12)$$

where an entry is marked as unobserved when it is missing or identified as unreliable before reconstruction. Unobserved values are replaced by the normalized mean, producing $\hat{\mathbf{X}}$.

The value and observation streams are concatenated and projected into the latent space:

$$\mathbf{H}_0 = [\tilde{\mathbf{X}} \parallel \mathbf{O}] \mathbf{W}_{in} + \mathbf{P}, \quad (13)$$

where $\mathbf{P}$ is a learned positional embedding.

The latent sequence is processed by a stack of Transformer blocks to model temporal dependencies and relationships across the selected joints. Each pre-normalized block is written as [26]

$$\mathbf{H}' = \mathbf{H} + \text{MHA}(\text{LN}(\mathbf{H})), \quad (14)$$

$$\mathbf{H}'' = \mathbf{H}' + \text{FFN}(\text{LN}(\mathbf{H}')). \quad (15)$$

After B Transformer blocks, a residual temporal-convolution stage further refines local motion patterns:

$$\mathbf{H} = \text{LN}(\mathbf{H}_{tr} + \text{Conv}_{\kappa_2}(\text{GELU}(\text{Conv}_{\kappa_1}(\mathbf{H}_{tr}))))). \quad (16)$$

The recovered motion window is then obtained by

$$\hat{\mathbf{X}} = \mathbf{H} \mathbf{W}_{out}. \quad (17)$$

During training, the model is optimized to reconstruct corrupted entries while preserving reliable observations. Let $\Omega = \{(t, j) : m_{t,j} = 1\}$ denote the corrupted locations and $\bar{\Omega}$ denote the observed locations. The reconstruction loss on the corrupted entries is defined as

$$\mathcal{L}_{\text{masked}} = \frac{1}{|\Omega|} \sum_{i \in \Omega} (\hat{x}_i - x_i)^2. \quad (18)$$

To avoid unnecessary changes to reliable motor observations, an additional observed-entry consistency loss is defined as

$$\mathcal{L}_{\text{observed}} = \frac{1}{|\bar{\Omega}|} \sum_{i \in \bar{\Omega}} (\hat{x}_i - x_i)^2. \quad (19)$$

The final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{masked}} + \lambda_o \mathcal{L}_{\text{observed}}, \quad (20)$$

where λ_o controls the contribution of the observed-entry consistency term.

An exponential moving average of the model parameters is maintained during training as

$$\bar{\theta} \leftarrow \gamma \bar{\theta} + (1 - \gamma) \theta. \quad (21)$$

At inference, the trained recovery model is applied to overlapping motion windows. Predictions for the same frame from different windows are averaged:

$$\bar{x}_t = \frac{1}{|\mathcal{W}_t|} \sum_{w \in \mathcal{W}_t} [g_{\bar{\theta}}(\tilde{\mathbf{W}}_w)]_t, \quad (22)$$

where $\mathcal{W}_t$ denotes the set of windows covering frame t . The reconstructed value is used for entries marked as unreliable, while reliable observations are retained. The resulting healed motion sequence is then provided to the recognizer in Eq. (10) to infer the physically executed action.

E. Instruction–Execution Consistency Verification

The healed motion sequence is first passed to the action recognizer to obtain the inferred action sequence $\hat{\mathbf{a}}$. After temporal alignment with the trusted intended action sequence $\mathbf{a}$, the two sequences are compared frame by frame. A mismatch is reported when $\hat{a}_t \neq a_t$, indicating that the physically executed action is not consistent with the trusted intended instruction. To reduce isolated mismatches near action transitions, the binary mismatch sequence may be further processed using a short persistence rule before an alarm is generated. This post-processing does not require retraining and can be adjusted independently from the recovery and recognition modules. The verification module itself contains no learned parameters and only determines whether the physical execution is consistent with the trusted instruction; it does not identify which software or control component caused the inconsistency.

Algorithm 1 Self-Healing Action Verification Pipeline

Input: Clean training set $\mathcal{D}$, candidate encoders $\mathcal{J}_{\text{all}}$, explanation samples S_e , threshold τ , action classes $\mathcal{C}$, temporal scales $\mathcal{L}$, fusion weights $\mathbf{w}$, retained gallery size m , failure family $\mathcal{Q}$, distributions p_q and p_ρ , training index set $\mathcal{E}$, EMA coefficient γ , query observations $\hat{\mathbf{X}}$, and trusted intended actions $\mathbf{a}$.

Output: Selected encoders $\mathcal{J}^*$, recognizer $\mathcal{R}$, recovery parameters θ^* , inferred actions $\hat{\mathbf{a}}$, and verification flags $\mathbf{z}$

Phase 1: Explainable Motor Encoder Selection

for all $(\mathbf{X}_n, y_n) \in \mathcal{D}$ **do**

 Extract velocity, acceleration, and motion-range descriptors over $\mathcal{J}_{\text{all}}$
 Form the window-level feature vector $\mathbf{u}_n$

end for

Train an action-discriminative model h on $\{(\mathbf{u}_n, y_n)\}$

for all $n \in S_e$ **do**

 Apply LIME to h around $\mathbf{u}_n$ and obtain local coefficients β_n

end for

for all $j \in \mathcal{J}_{\text{all}}$ **do**

 Compute normalized joint importance I_j using Eq. (6)

end for

Select $\mathcal{J}^*$ using Eq. (7)

Phase 2: Multiscale Gallery-Based Action Recognition

for all $c \in \mathcal{C}$ and $\ell \in \mathcal{L}$ **do**

 Extract causal motion windows of class c at scale ℓ over $\mathcal{J}^*$

 Compute statistical motion features $\phi(\mathbf{W})$

 Normalize the features using clean-gallery statistics

 Apply k -center selection and retain m representative features as $\mathcal{G}_{c,\ell}$

end for

Construct the multiscale recognizer $\mathcal{R}$

Phase 3: Temporal Self-Healing Training

Initialize recovery parameters θ and EMA parameters $\bar{\theta}$

for all $e \in \mathcal{E}$ **do**

for all minibatches $\mathbf{B}$ from $\mathcal{D}|_{\mathcal{J}^*}$ **do**

 Sample failure type $q \sim p_q$ and corruption ratio $\rho \sim p_\rho$

 Generate corruption mask $\mathbf{M}$ and corrupted observations $\tilde{\mathbf{B}}$

 Construct the observation indicator $\mathbf{O}$

 Reconstruct $\hat{\mathbf{B}} \leftarrow g_\theta(\tilde{\mathbf{B}}, \mathbf{O})$

 Update θ using Eq. (20)

 Update $\bar{\theta} \leftarrow \gamma\bar{\theta} + (1 - \gamma)\theta$

end for

end for

Set $\theta^* \leftarrow \bar{\theta}$

Phase 4: Online Self-Healing and Instruction Verification

for all overlapping windows $\mathbf{W}$ of $\hat{\mathbf{X}}|_{\mathcal{J}^*}$ **do**

 Reconstruct each window using g_{θ^*}

 Accumulate reconstruction results for all covered frames

end for

Average overlapping predictions and form healed observations $\hat{\mathbf{X}}$

for all $t \in \mathcal{T}$ **do**

 Infer executed action $\hat{a}_t$ using Eq. (10)

 Set $z_t = 1$ if $\hat{a}_t \neq a_t$, otherwise $z_t = 0$

end for

return $\mathcal{J}^*, \mathcal{R}, \theta^*, \hat{\mathbf{a}}, \mathbf{z}$

V. EVALUATION

A. Dataset and Environment setup

The dataset contains motor-encoder joint-state recordings collected on a Booster Robotics K1 Education humanoid robot [42] at approximately 20 Hz. The robot provides coordinated multi-joint locomotion, making its motor-encoder signals suitable for studying action-level physical execution. Each frame records the joint angles of all 22 candidate encoders and the executed action label. Five locomotion action classes are included: forward, shift left, shift right, turn left, and turn right. The single-action part contains ten sessions per class, resulting in 50 sessions and 7,597 labeled motion frames. During configuration, the first eight sessions of each action class are used for model configuration and training, giving 40 sessions, while the remaining two sessions form a session-disjoint validation set of 10 sessions. After the model configuration is fixed, all 50 single-action sessions are used according to the evaluation protocol of each experiment. For clean single-action recognition, five session-held-out folds are used, each containing 40 training sessions and 10 test sessions. The compound-action part contains eight sessions with multiple action transitions and 8,003 labeled motion frames. These sessions are kept separate from the configuration and training data and are used for compound-action recognition, self-healing, and instruction-verification evaluation. The five action classes contribute 1,215, 1,880, 1,275, 1,813, and 1,820 frames, respectively. In total, the dataset contains 58 sessions and 15,600 labeled frames. All data partitions are performed at the session level rather than the frame level. All experiments are conducted on a workstation with an Intel Xeon w3-2423 CPU and an NVIDIA RTX 4000 Ada GPU. The experimental pipeline is implemented in Python 3.14.4 using PyTorch 2.13.0 with CUDA 13.0.

B. Model Initialization

This subsection introduces the model initialization, parameter settings, and training configurations. For explainable motor encoder selection, signals from all candidate encoders are used to construct motion features. A 100-tree random forest is used as the auxiliary model for LIME, and the LIME importance is aggregated over 300 sampled training windows. The relative-importance threshold is fixed at $\tau = 0.20$ according to Eq. (7). Encoder-selection results are presented in the following section. For the multiscale gallery-based action recognizer, the causal window lengths are set to $\mathcal{L} = \{3, 4, 6, 12\}$, and the four temporal scales are combined using equal fusion weights of 0.25. The maximum recognition context is 12 frames. For each action class and temporal scale, k -center selection retains 385 reference features, resulting in 7,700 gallery references. For the sensor self-healing module, the recovery model uses a 64-frame input window, a latent dimension of 128, four Transformer blocks with four attention heads, a feed-forward dimension of 256, and temporal-convolution kernel sizes of five and three. The observed-entry consistency weight λ_o is set to 0.05. During mixed-failure training, the corruption ratio is sampled from $\rho \sim \mathcal{U}(0.1, 0.7)$. The model is optimized using AdamW [43], [44] with an initial learning rate of

10^{-3} , weight decay of 10^{-4} , batch size of 128, dropout of 0.05, gradient clipping at 1.0, and cosine learning-rate decay to 5×10^{-5} . The recovery configuration is selected using 40 single-action training sessions and 10 session-disjoint validation sessions. Candidate configurations are compared on the validation recordings under the seven sensor-failure modes at 10%, 30%, 50%, and 70% corruption, and recovery RMSE is used for model selection. The selected configuration uses an exponential moving average with a decay of 0.9995 and a common training budget of 300 epochs. After the configuration is fixed, the recovery model is retrained using all 50 single-action sessions, and the EMA parameters are used for the final evaluation.

C. Evaluation Metrics

1) *Classification Performance*: To evaluate the classification performance, we use accuracy and F1 score, defined as

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (23)$$

$$F_1 = \frac{2TP}{2TP + FP + FN}, \quad (24)$$

where TP , TN , FP , and FN denote correctly classified positive, correctly classified negative, incorrectly classified positive, and incorrectly classified negative actions, respectively.

2) *Reconstruction Distance*: To evaluate the quality of sensor recovery, we use reconstruction RMSE and Wasserstein distance. For the affected entries Ω , RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{|\Omega|} \sum_{i \in \Omega} (\hat{x}_i - x_i)^2}. \quad (25)$$

The Wasserstein distance between the clean and reconstructed distributions of joint j is defined as

$$W_1(F_j, G_j) = \int_0^1 |F_j^{-1}(u) - G_j^{-1}(u)| du, \quad (26)$$

where F_j and G_j denote the clean and reconstructed distributions of joint j , respectively.

D. Evaluation Results of Motor Selection

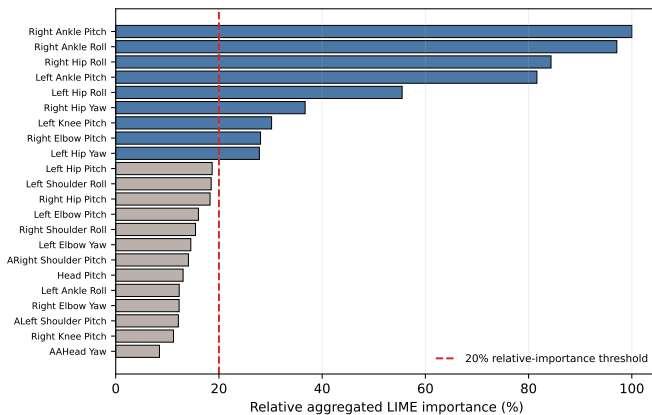


Fig. 3. Normalized LIME importance from clean single-action training sessions. Blue bars satisfy the fixed 20% relative-importance threshold; gray bars are excluded.

Figure 3 shows the normalized LIME importance of the 22 candidate encoders with the fixed 20% relative-importance threshold. The importance values are aggregated over 300 sampled training windows. Encoders above the threshold are shown in blue, while the remaining encoders are shown in gray. The encoders are ordered from the highest importance to the lowest importance. The highest scores are obtained by the right ankle pitch and right ankle roll, followed by the right hip roll and left ankle pitch. Nine encoders have importance scores above the 20% threshold. The ninth encoder has an importance score of 27.8%, while the tenth encoder has a score of 18.7%, giving a difference of 9.1 percentage points. According to the fixed threshold rule, these nine encoders are selected for the following action-recognition and self-healing experiments. Most of the selected encoders are located in the lower-body joints, while the right elbow pitch is also included. The encoder ranking is obtained only from the clean single-action training data. The nine selected encoders are further evaluated on ten session-disjoint single-action validation recordings and achieve a mean accuracy of 99.46%. This result shows that the selected encoder subset can provide high recognition accuracy on unseen single-action sessions. These validation recordings are not used for the LIME importance ranking or threshold determination.

E. Evaluation Results of Action Recognition

TABLE I
AVERAGE CLEAN ACTION RECOGNITION ACCURACY AND MACRO-F1 (%). SINGLE-ACTION VALUES AVERAGE FIVE SESSION-HELD-OUT FOLDS AND COMPOUND VALUES AVERAGE EIGHT SESSIONS.

Method	Single action		Compound	
	Acc.	F1	Acc.	F1
Gaussian NB	81.7	83.1	73.5	74.2
AdaBoost	81.4	81.2	77.2	77.1
Logistic regression	93.7	93.7	86.0	86.7
Decision tree	94.8	94.6	87.6	87.8
Random forest	98.0	98.0	91.6	91.9
RBF-SVM	98.4	98.3	91.0	91.4
1D-CNN	98.3	98.2	91.6	91.5
BiLSTM	98.4	98.3	92.3	92.3
Transformer	98.1	98.0	92.9	92.9
KNN-1	98.7	98.6	92.1	92.4
Proposed multiscale	98.4	98.3	94.8	95.1

Table I compares the proposed multiscale recognizer with ten baseline methods on both single-action and compound-action data. The baseline methods include traditional machine-learning classifiers, such as Gaussian naive Bayes, AdaBoost, logistic regression, decision tree, random forest, RBF-SVM, and KNN-1, as well as deep-learning models including 1D-CNN, BiLSTM [45], and Transformer. For single-action recognition, the proposed multiscale recognizer achieves a high accuracy of 98.4% and a macro-F1 of 98.3%, showing competitive performance among all comparison methods. Among the traditional methods, Gaussian naive Bayes and AdaBoost obtain relatively lower accuracies of 81.7% and 81.4%, while logistic regression and decision tree improve the results to 93.7% and 94.8%, respectively. Random forest and RBF-SVM

TABLE II
SENSOR-FAILURE RECOVERY AVERAGED OVER SEVEN MODES. RMSE AND STANDARDIZED WD ARE LOWER-IS-BETTER; ACTION ACCURACY AND MACRO-F1 ARE PERCENTAGES AND HIGHER-IS-BETTER.

Method	10% failure				30% failure				50% failure				70% failure			
	RMSE	WD	Acc.	F1	RMSE	WD	Acc.	F1	RMSE	WD	Acc.	F1	RMSE	WD	Acc.	F1
Unmitigated	—	—	55.3	55.8	—	—	31.8	28.2	—	—	26.9	21.2	—	—	28.0	22.7
Moving average	0.110	0.45	80.5	81.6	0.154	0.67	57.8	57.4	0.192	0.87	44.8	41.5	0.227	1.08	38.0	32.9
Median filter	0.099	0.26	91.2	91.6	0.176	0.60	65.7	66.7	0.239	1.01	42.2	41.0	0.284	1.40	34.6	31.1
Savitzky–Golay	0.147	0.69	77.1	77.9	0.189	0.87	53.4	53.6	0.223	1.05	40.3	39.2	0.255	1.23	33.8	29.8
Gaussian filter	0.127	0.56	80.6	81.6	0.164	0.75	57.8	57.2	0.198	0.92	45.9	42.4	0.230	1.11	39.3	34.3
Wiener filter	0.227	0.97	68.9	69.8	0.210	0.88	58.3	58.2	0.222	0.99	51.2	50.5	0.243	1.14	42.9	41.5
Hampel	0.225	0.73	88.0	88.3	0.280	1.16	57.3	57.9	0.323	1.59	34.5	32.2	0.344	1.85	29.1	24.5
Linear interpolation	0.325	1.81	58.4	58.8	0.330	1.81	38.7	35.0	0.334	1.81	34.7	29.1	0.336	1.81	34.2	29.0
Exp. smoothing	0.150	0.62	71.2	72.7	0.180	0.78	50.4	48.9	0.209	0.94	40.6	36.0	0.236	1.11	36.2	30.1
Butterworth	0.121	0.39	36.2	36.4	0.150	0.56	33.3	30.4	0.183	0.76	32.7	28.4	0.215	0.97	31.2	25.3
PCA	0.151	0.80	84.3	84.6	0.179	0.91	70.1	69.3	0.205	1.04	54.3	52.6	0.229	1.17	42.5	38.8
KNN	0.060	0.19	82.4	80.7	0.072	0.26	72.3	68.2	0.083	0.30	65.9	61.0	0.093	0.36	57.9	50.9
TCN	0.042	0.11	90.9	91.3	0.046	0.11	88.6	89.1	0.049	0.11	87.0	87.7	0.052	0.13	84.8	85.5
Conv. autoencoder	0.044	0.11	89.1	89.9	0.048	0.11	83.5	84.6	0.052	0.12	80.7	81.7	0.056	0.14	78.3	79.2
TMAE (ours)	0.028	0.08	93.0	93.1	0.028	0.06	92.7	93.0	0.030	0.06	92.7	93.0	0.033	0.07	92.1	92.4

further achieve 98.0% and 98.4% accuracy. The deep-learning models also obtain strong single-action performance, with 1D-CNN, BiLSTM, and Transformer reaching 98.3%, 98.4%, and 98.1% accuracy, respectively. KNN-1 achieves the highest single-action accuracy of 98.7% and macro-F1 of 98.6%. For compound-action recognition, the proposed multiscale recognizer achieves the highest accuracy of 94.8% and the highest macro-F1 of 95.1% among all evaluated methods. In comparison, Gaussian naive Bayes and AdaBoost obtain 73.5% and 77.2% accuracy, while logistic regression and decision tree reach 86.0% and 87.6%, respectively. Random forest, RBF-SVM, and KNN-1 further improve the compound-action accuracy to 91.6%, 91.0%, and 92.1%. The deep-learning methods also show better performance than most traditional classifiers, where 1D-CNN achieves 91.6%, BiLSTM reaches 92.3%, and Transformer obtains 92.9% accuracy. Overall, the results show that the proposed method maintains high recognition performance for isolated single actions and provides the best performance for the more complex compound-action sequences.

F. Evaluation Results of Sensor Recovery

Table II compares the proposed TMAE with thirteen recovery baselines under 10%, 30%, 50%, and 70% sensor-failure ratios. The comparison methods cover several commonly used recovery strategies. Moving average, median, Savitzky–Golay, Gaussian, Wiener, Hampel, exponential smoothing, and Butterworth are included as classical filtering methods, while linear interpolation estimates corrupted values from neighboring observations. PCA is used as a low-dimensional reconstruction baseline, and KNN performs recovery based on nearby clean motion patterns. TCN [19] and the convolutional autoencoder are included as learned temporal-recovery baselines. For a fair comparison, the learned recovery models use the same 50 single-action training sessions, the same seven-failure mixed corruption setting, the same AdamW optimizer, batch size

of 128, EMA procedure, and the same 300-epoch training budget. The same training data and random seed are also used for the learned baselines. PCA and KNN are fitted only on clean single-action training windows, while the compound-action recordings are not used for training and are used only for recovery and recognition evaluation. All methods receive the same corrupted compound-action sequences at each failure ratio. The reported RMSE and WD measure the reconstruction quality, while accuracy and macro-F1 measure whether the recovered signals can still support correct action recognition. Each value in Table II is averaged over the seven sensor-failure modes. The unmitigated setting directly uses the corrupted observations without any recovery and is included as a reference to show the performance loss when sensor failures are not handled.

At 10% failure, the proposed TMAE achieves the best overall performance, with an RMSE of 0.028, WD of 0.08, accuracy of 93.0%, and macro-F1 of 93.1%. Among the classical methods, median filtering performs relatively well and reaches 91.2% accuracy with an RMSE of 0.099, while moving average and Gaussian filtering obtain 80.5% and 80.6% accuracy, respectively. PCA reaches 84.3% accuracy, and KNN obtains a relatively low RMSE of 0.060 but only 82.4% action-recognition accuracy. The learned recovery methods give stronger results, where TCN achieves 90.9% accuracy and 91.3% macro-F1, while the convolutional autoencoder reaches 89.1% accuracy and 89.9% macro-F1. When the failure ratio increases to 30%, the proposed TMAE remains almost unchanged, with 92.7% accuracy, 93.0% macro-F1, an RMSE of 0.028, and WD of 0.06. In contrast, the performance of most comparison methods starts to decrease more clearly. TCN drops to 88.6% accuracy and the convolutional autoencoder to 83.5%, while KNN and PCA decrease to 72.3% and 70.1%, respectively. Most classical filtering methods are reduced to approximately 50%–66% accuracy. At 50% failure, TMAE still maintains 92.7% accuracy and 93.0% macro-F1, and its

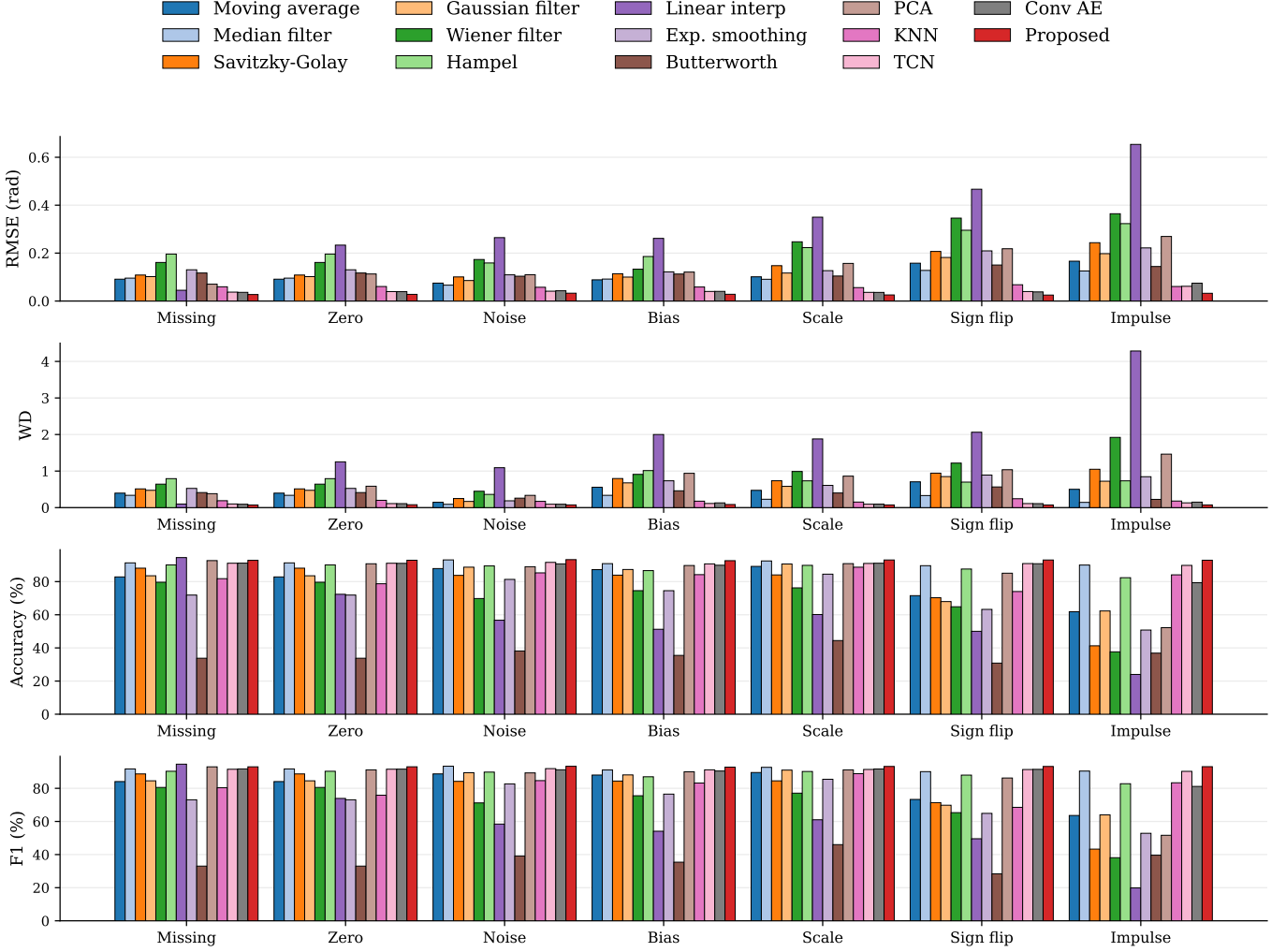


Fig. 4. Per-mode reconstruction and action-recognition results at 10% sensor failure. Each panel repeats the seven failure labels to avoid cross-panel ambiguity. The proposed method is shown in red.

RMSE increases only slightly to 0.030. This result shows that the performance of the proposed method changes very slowly as more observations become corrupted. By comparison, TCN decreases further to 87.0% accuracy, the convolutional autoencoder to 80.7%, KNN to 65.9%, and PCA to 54.3%, while most classical methods fall below 55% accuracy. At the highest 70% failure ratio, TMAE still achieves 92.1% accuracy and 92.4% macro-F1, with an RMSE of 0.033 and WD of 0.07. From 10% to 70% failure, its accuracy decreases by only 0.9 percentage points, from 93.0% to 92.1%. In contrast, TCN decreases from 90.9% to 84.8%, the convolutional autoencoder from 89.1% to 78.3%, KNN from 82.4% to 57.9%, and PCA from 84.3% to 42.5%. The unmitigated setting decreases from 55.3% to 28.0%. Overall, the proposed TMAE shows the slowest performance degradation as the failure ratio increases and maintains both low reconstruction error and high action-recognition performance across all four failure levels.

Figure 4 presents the recovery and recognition results for the seven individual sensor-failure modes at the 10% failure ratio. The evaluated failures include missing values, zero replacement, additive noise, bias offset, scale change,

sign flip, and impulse corruption. Missing and zero failures remove or replace the original observations, noise adds random perturbations, bias and scale change the signal level or amplitude, sign flip reverses the signal direction, and impulse corruption introduces large abnormal spikes. The four panels report RMSE, WD, accuracy, and macro-F1, respectively, and the proposed method is shown in red. At this relatively low failure ratio, most learned methods still maintain high recognition performance, but clear differences can already be observed among different failure modes. For missing failure, linear interpolation achieves the highest recognition accuracy of 94.50%, while the proposed method reaches 92.85% with a much lower RMSE of 0.027 rad and WD of 0.073. For zero failure, the proposed method achieves the highest accuracy of 92.90%, compared with 91.31% for median filtering and 91.08% for TCN. Under noise, the proposed method reaches 93.27% accuracy with an RMSE of 0.033 rad, while median filtering gives a similar accuracy of 93.10% but a higher RMSE of 0.066 rad. For bias and scale failures, the proposed method achieves 92.63% and 93.06% accuracy, respectively, and keeps the lowest reconstruction errors among the com-

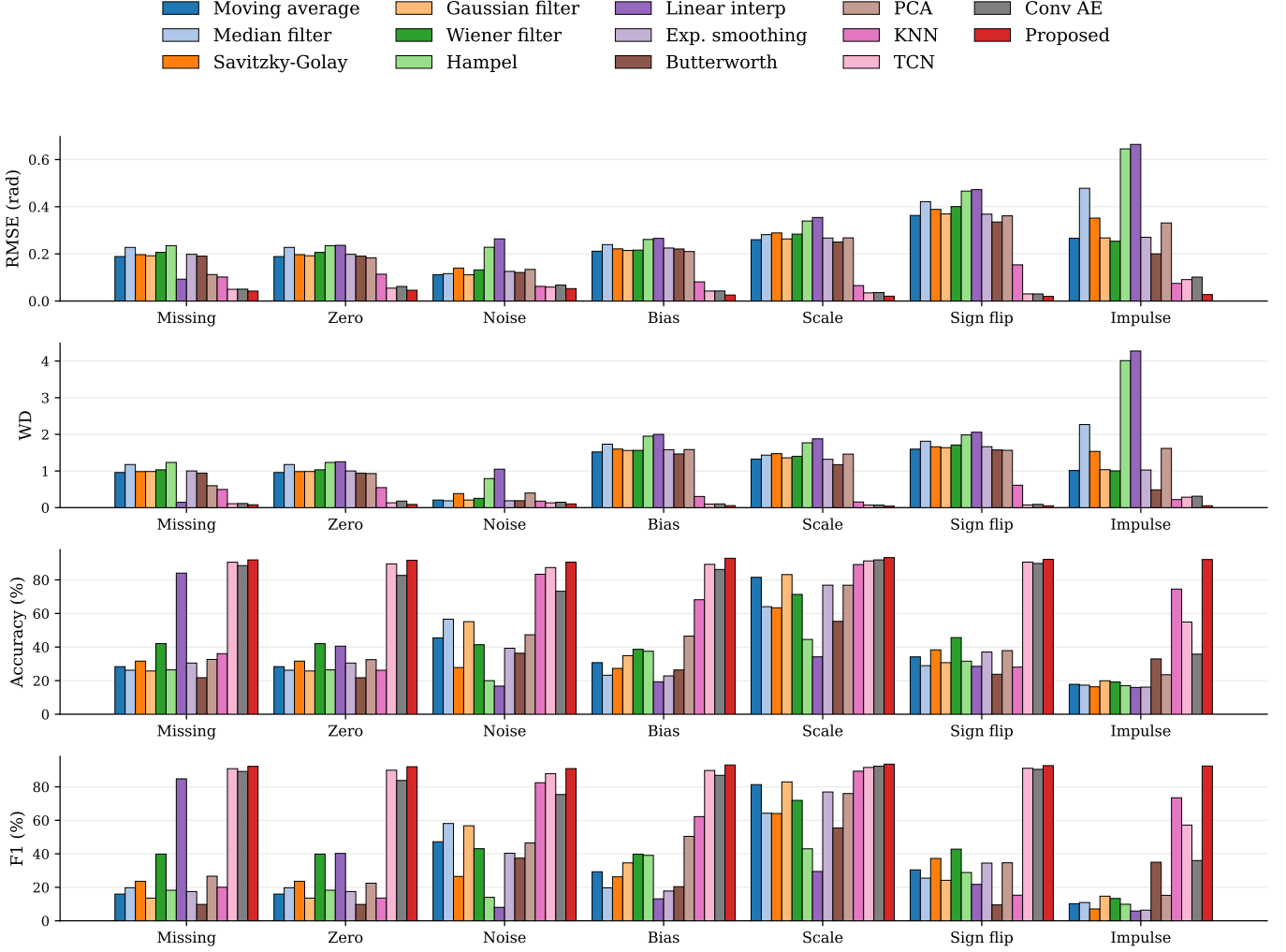


Fig. 5. Per-mode reconstruction and action-recognition results at 70% sensor failure. Severe finite-valued failures expose a widening gap over classical and convolutional baselines; TMAE leads every comparison method in the seven-mode averages.

pared methods. The difference becomes more obvious for sign-flip and impulse failures. Under sign flip, the proposed method achieves 93.05% accuracy, while TCN and the convolutional autoencoder reach 90.89% and 90.80%, respectively. Under impulse failure, the proposed method reaches 92.91% accuracy, compared with 89.99% for median filtering and 89.79% for TCN. These results show that the proposed method already provides stable reconstruction and recognition across all seven failure modes even at the 10% corruption level.

Figure 5 further presents the same seven failure modes when the sensor-failure ratio increases to 70%. Compared with Fig. 4, the performance gap between the proposed method and most comparison methods becomes much larger. For missing and zero failures, the proposed method still achieves 91.90% and 91.65% accuracy, while TCN reaches 90.50% and 89.53%, respectively. Under noise failure, the proposed method obtains 90.55% accuracy with an RMSE of 0.053 rad, compared with 87.34% accuracy and 0.059 rad RMSE for TCN. For bias and scale failures, the proposed method reaches 92.86% and 93.28% accuracy, with RMSE values of only 0.026 and 0.021 rad, respectively. The difference becomes especially clear for

sign-flip and impulse failures. Under sign flip, the proposed method achieves 92.22% accuracy and an RMSE of 0.020 rad, while TCN reaches 90.56% accuracy with an RMSE of 0.030 rad. Under impulse failure, the proposed method still achieves 92.13% accuracy with an RMSE of 0.027 rad, whereas the strongest baseline in recognition, KNN, reaches only 74.52% accuracy with an RMSE of 0.075 rad. TCN and the convolutional autoencoder decrease further to 54.92% and 35.77% accuracy under impulse failure. Overall, the proposed method remains above 90% accuracy for all seven failure modes at the 70% corruption level, while many comparison methods show a much larger performance decrease. This further shows that the proposed self-healing method degrades slowly as the failure ratio increases and can maintain stable action recognition under severe sensor corruption.

G. Case Study of End-to-End Self-Healing

Figure 6 presents a qualitative case study of the proposed end-to-end self-healing process under a 70% sensor-failure ratio. The four rows correspond to four selected motor encoders

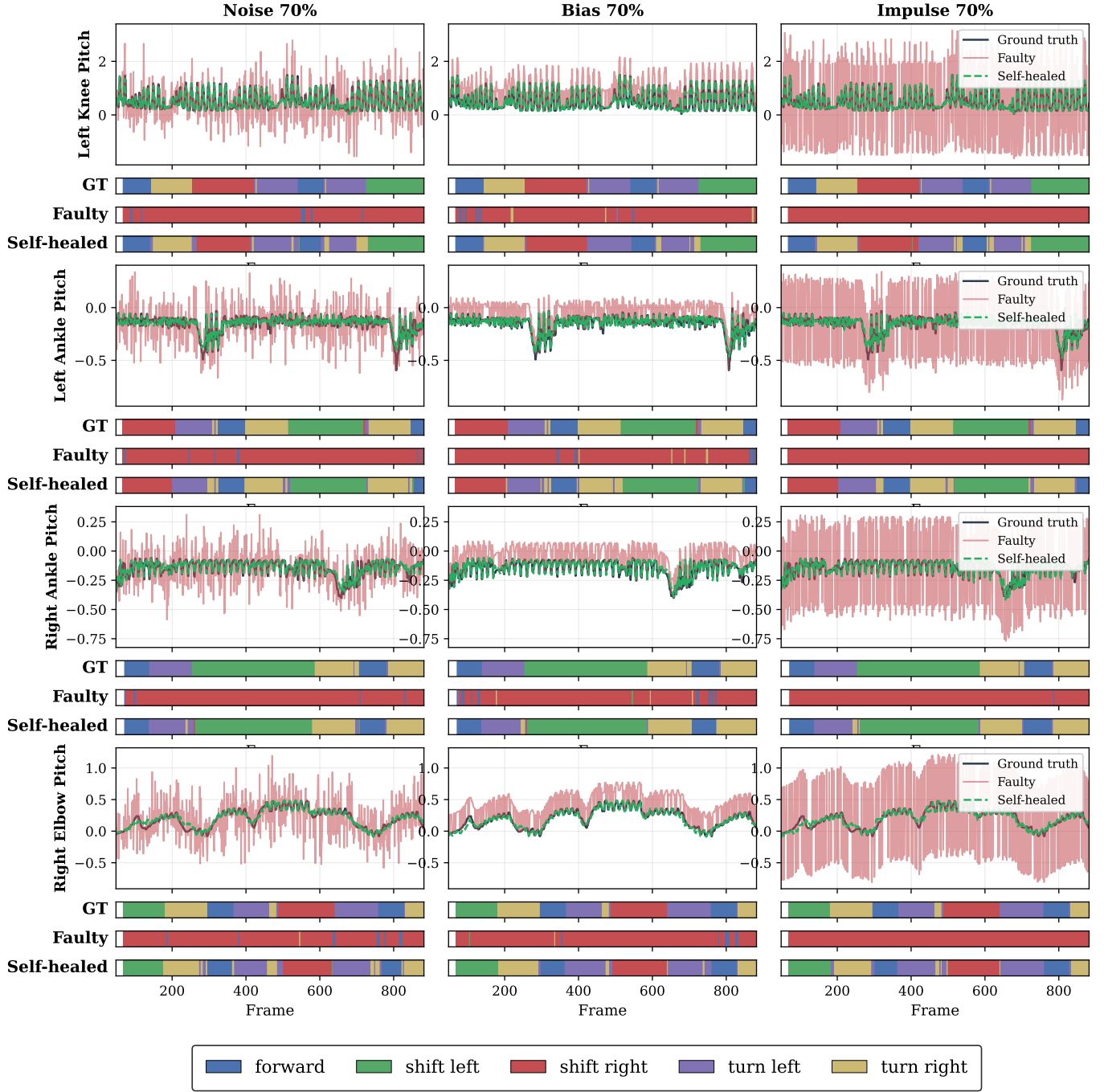


Fig. 6. End-to-end self-healing at 70% failure. Rows show four motors from four different compound sessions; columns show noise, bias, and impulse failure. Signal panels overlay ground truth (dark), faulty observation (red), and self-healed output (green dashed). The three bars below each panel show predicted actions for ground-truth, faulty, and self-healed sequences. Blue, green, red, purple, and yellow denote forward, shift left, shift right, turn left, and turn right, respectively.

from four different compound-action sessions, while the three columns show noise, bias, and impulse failures, respectively. In each signal panel, the ground-truth trajectory is shown in dark color, the faulty observation is shown in red, and the self-healed output is shown by the green dashed line. The three action bars below each signal panel show the action sequences obtained from the ground-truth, faulty, and self-healed motor signals. The five colors represent forward, shift left, shift right, turn left, and turn right.

The three failure modes produce different types of signal distortion. Under noise failure, the faulty observations fluctuate strongly around the original trajectory, while the self-healed signal follows the main motion pattern and removes most of the random variation. Under bias failure, the faulty signal is shifted away from the ground-truth trajectory for a long period, while the self-healed result restores the signal toward the original motion level. Under impulse failure, large abnormal spikes are introduced into the motor observations,

but the reconstructed signal removes most of these spikes and preserves the major motion changes.

The action bars show the same effect at the recognition level. The faulty observations frequently produce long incorrect action segments or collapse the prediction into only a small number of action classes. After self-healing, the predicted action sequences become much closer to those obtained from the ground-truth signals, and most major action transitions are recovered. This behavior is consistently observed across the four motors and different compound-action sessions, showing that the proposed method can recover useful motion information and support stable action recognition even under a severe 70% sensor-failure condition.

H. Evaluation Results of Instruction Verification

To evaluate the final instruction–execution verification stage, we construct controlled command-substitution cases from the compound-action recordings. For each executed-action frame, the matching action label is used as the normal intended command, while each of the other four action classes is separately used as a substituted command. This produces one normal case and four inconsistent cases for every frame and covers all 20 ordered action substitutions. Because the final verification rule directly compares the inferred physical action with the trusted intended command, its performance is mainly determined by the underlying action-recognition result.

With clean motor observations, the proposed framework achieves 96.8% balanced verification accuracy. Without self-healing, the balanced accuracy decreases from 72.0% at 10% sensor failure to 55.0% at 70% failure. After self-healing, it remains nearly unchanged, reaching 95.6% at 10% failure and 95.1% at 70% failure. At the highest 70% failure ratio, the normal-command acceptance still reaches 92.1%, while the tampered-command recall reaches 98.0%. These results show that the self-healing stage allows the final instruction–execution consistency verification to remain stable even under severe sensor corruption.

TABLE III
COUNTERFACTUAL INSTRUCTION-VERIFICATION RESULTS (%)

Evidence	Failure ratio	Normal accept.	Tamper recall	Balanced accuracy
Clean	0%	94.8	98.7	96.8
Unmitigated	10%	55.3	88.8	72.0
Unmitigated	30%	31.8	83.0	57.4
Unmitigated	50%	26.9	81.7	54.3
Unmitigated	70%	28.0	82.0	55.0
Self-healed	10%	93.0	98.2	95.6
Self-healed	30%	92.7	98.2	95.4
Self-healed	50%	92.7	98.2	95.4
Self-healed	70%	92.1	98.0	95.1

VI. DISCUSSION

The experimental results show that the proposed framework can maintain reliable action recognition and instruction–execution verification under both clean and faulty motor

observations. The proposed multiscale gallery-based recognizer achieves 98.4% accuracy for single-action recognition and 94.8% for compound-action recognition. This shows that motion information from different temporal scales is useful for describing both individual actions and action transitions. Different from methods that require classifier training, the proposed recognizer does not use trainable classifier parameters and directly matches the current motion features with class-specific reference galleries. Under sensor failures, the self-healing model further improves the reliability of the motor evidence. At 10%, 30%, 50%, and 70% failure ratios, the proposed method achieves average recognition accuracies of 93.0%, 92.7%, 92.7%, and 92.1%, respectively. The accuracy decreases by only 0.9 percentage points when the failure ratio increases from 10% to 70%, showing that the proposed method degrades slowly under more severe sensor corruption. This stability comes from using temporal and cross-joint information to reconstruct unreliable observations while using the observed-entry consistency term to preserve reliable observations. Therefore, the recovered signals can still keep the motion patterns needed by the downstream action recognizer. The final recognized action can then be compared with the trusted intended instruction for instruction–execution consistency verification.

Nevertheless, several limitations still need further study. First, although the multiscale gallery-based recognizer does not require trainable classifier parameters, it needs to store a separate reference gallery for each action class and temporal scale, which introduces a relatively large storage cost. This is because representative motion patterns from different action phases and temporal scales need to be retained for matching. The current k-center selection can reduce redundant gallery samples, but the storage requirement may still increase when the number of actions, temporal scales, or training sessions becomes larger. More compact prototype selection, feature quantization, or gallery compression can be studied in future work. Second, the current sensor failures are injected into clean recordings. This provides controlled failure types and corruption ratios, but cannot fully represent all physical encoder failures in real robots. Instruction tampering is also evaluated by controlled command substitution rather than a live compromised communication or control channel. In addition, the current dataset is collected from one humanoid robot and contains five locomotion actions, so the selected encoders and reference galleries may need to be updated for different robot platforms, actions, speeds, and operating environments. Finally, the current self-healing model processes buffered motion windows, which introduces additional delay. Future work will further evaluate real sensor failures and online instruction manipulation, extend the framework to more robot platforms and action types, and study causal or streaming recovery together with more compact gallery storage.

VII. CONCLUSION

Reliable verification of physical robot execution is important for humanoid robots operating under possible instruction tampering and sensor failures. Protecting the command path alone

cannot guarantee that the final robot behavior is correct, while unreliable motor observations can further reduce the reliability of execution verification. In this paper, we proposed a self-healing motor-based action verification framework that combines explainable motor-encoder selection, multiscale gallery-based action recognition, and temporal sensor recovery. The proposed recognizer infers the physically executed action without trainable classifier parameters, while the self-healing model reconstructs unreliable motor observations using temporal and cross-joint information. The inferred action is finally compared with the trusted intended instruction for consistency verification. Experimental results on a Booster Robotics K1 Education humanoid robot show that the proposed method achieves 98.4% accuracy for single-action recognition and 94.8% for compound-action recognition. Under seven sensor-failure modes, the self-healing method achieves average recognition accuracies of 93.0%, 92.7%, 92.7%, and 92.1% at 10%, 30%, 50%, and 70% failure ratios, respectively, showing only a small performance decrease as sensor corruption increases. At the 70% failure ratio, the final instruction-execution verification still achieves 95.1% balanced accuracy. Future work will evaluate real sensor failures and online instruction manipulation, extend the framework to more robot platforms and action types, and investigate causal recovery and more compact gallery storage for practical deployment.

REFERENCES

- [1] A. W. Yu and A. Nayak, "The Internet of humanoids: A survey of technologies, applications, and challenges," *IEEE Internet of Things Journal*, vol. 13, no. 6, pp. 10498–10521, 2026.
- [2] Q. Sheng, Z. Zhou, J. Li, X. Mi, P. Xiang, Z. Chen, H. Xu, S. Jia, X. Wu, Y. Cui, S. Ye, J. Yu, Y. Du, S. Zhai, K. Xu, Y. Yang, Z. Lou, Z. Song, Z. Yin, Y. Sun, R. Xiong, J. Zou, and H. Yang, "A comprehensive review of humanoid robots," *SmartBot*, vol. 1, no. 1, p. e12008, 2025.
- [3] Y. Tong, H. Liu, and Z. Zhang, "Advancements in humanoid robots: A comprehensive review and future prospects," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 301–328, 2024.
- [4] A. Haskard and D. Herath, "Secure robotics: Navigating challenges at the nexus of safety, trust, and cybersecurity in cyber-physical systems," *ACM Computing Surveys*, vol. 57, no. 9, pp. 1–48, 2025.
- [5] A. Bhardwaj, S. Bharany, A. U. Rehman, G. G. Tejani, and S. Hussien, "Securing cyber-physical robotic systems for enhanced data security and real-time threat mitigation," *EURASIP Journal on Information Security*, vol. 2025, no. 1, p. 1, 2025.
- [6] A. K. Shaik, A. Mohammadi, and H. Malik, "A systematic review of sensor vulnerabilities and cyber-physical threats in industrial robotic systems," *IET Cyber-Physical Systems: Theory & Applications*, vol. 10, no. 1, p. e70023, 2025.
- [7] A. Lomakin and J. Deutscher, "Reliable algebraic fault detection and identification of robots," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3821–3837, 2022.
- [8] H. Ma, A. Song, J. Li, L. Ge, C. Fu, and G. Zhang, "Legged odometry based on fusion of leg kinematics and IMU information in a humanoid robot," *Biomimetic Intelligence and Robotics*, vol. 5, no. 1, p. 100196, 2025.
- [9] C. W. Herron, A. J. Fuge, B. C. Beiter, Z. J. Fuge, N. J. Tremaroli, S. Welch, M. Stelmack, M. Kogelis, P. Hancock, I. F. E. Simões, C. Runyon, I. Pressgrove, and A. Leonessa, "PANDORA: The open-source, structurally elastic humanoid robot," in *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, 2024, pp. 24–31.
- [10] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [11] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.
- [12] R. Natarajan, S. Reddy, S. C. Bose, H. Gururaj, F. Flammini, and S. Velmurugan, "Fault detection and state estimation in robotic automatic control using machine learning," *Array*, vol. 19, p. 100298, 2023.
- [13] G. Yang, H. Rezaee, A. Serrani, and T. Parisini, "Sensor fault-tolerant state estimation by networks of distributed observers," *IEEE Transactions on Automatic Control*, vol. 67, no. 10, pp. 5348–5360, 2022.
- [14] Á. M. Guerrero-Higuera, N. DeCastro-García, and V. Matellán, "Detection of cyber-attacks to indoor real time localization systems for autonomous robots," *Robotics and Autonomous Systems*, vol. 99, pp. 75–83, 2018.
- [15] S. Rivera, S. Lagraa, A. K. Iannillo, and R. State, "Auto-encoding robot state against sensor spoofing attacks," in *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2019, pp. 252–257.
- [16] M. Varshosaz, A. Afary, B. Mojaradi, M. Saadatseresht, and E. Ghanbari Parmehr, "Spoofing detection of civilian UAVs using visual odometry," *ISPRS International Journal of Geo-Information*, vol. 9, no. 1, p. 6, 2019.
- [17] Y. Man, R. Muller, M. Li, Z. B. Celik, and R. Gerdes, "That person moves like a car: Misclassification attack detection for autonomous systems using spatiotemporal consistency," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 6929–6946.
- [18] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.
- [19] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.
- [20] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [21] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [23] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.
- [24] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.
- [25] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 1533–1540.
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [27] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2114–2124.
- [28] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. MIT Press, 1949.
- [29] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. Stahel, "The approach based on influence functions," *Robust Statistics*. Wiley, 1986.
- [30] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, no. 1, p. 100, 2022.
- [31] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific Reports*, vol. 8, no. 1, p. 6085, 2018.
- [32] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "BRITS: Bidirectional recurrent imputation for time series," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [33] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *International Conference on Machine Learning*, 2018, pp. 5689–5698.
- [34] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural Computing and Applications*, vol. 19, no. 2, pp. 263–282, 2010.
- [35] W. Du, D. Côté, and Y. Liu, "SAITS: Self-attention-based imputation for time series," *Expert Systems with Applications*, vol. 219, p. 119619, 2023.
- [36] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "CSDI: Conditional score-based diffusion models for probabilistic time series imputation," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 24 804–24 816.

- [37] S. N. Shukla and B. M. Marlin, “Multi-time attention networks for irregularly sampled time series,” in *International Conference on Learning Representations*, 2021.
- [38] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 979–15 988.
- [39] T. Du, L. Melis, and T. Wang, “ReMasker: Imputing tabular data with masked autoencoding,” in *International Conference on Learning Representations*, 2024, pp. 9002–9024.
- [40] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.
- [41] T. F. Gonzalez, “Clustering to minimize the maximum intercluster distance,” *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.
- [42] Booster Robotics, “Booster Robotics GitHub Organization,” <https://github.com/BoosterRobotics>, 2026, accessed: 2026-09-27.
- [43] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations (ICLR)*, 2015.
- [44] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, 2019.
- [45] Z. Chen, M. Bernabei, V. Mainardi, X. Ren, G. Ciuti, and C. Stefanini, “A novel and accurate bilstm configuration controller for modular soft robots with module number adaptability,” *Soft robotics*, pp. soro–2024, 2024.