3rd International Conference on Electronics, Engineering, Computer Science and Applied Development (EESD 2026)

Article

Multi-Source Sensor Fusion and Dynamic Accuracy Retention Control for Collaborative Robots in Digital Manufacturing

Junyang Liu[1,*]

[1] Shandong University of Science and Technology, Qingdao, China
* Correspondence: Junyang Liu, Shandong University of Science and Technology, Qingdao, China

Abstract: Collaborative robots deployed in digital manufacturing environments must sustain high positioning accuracy despite payload variation, external interference, sensor noise, and communication delay. Existing approaches frequently concatenate asynchronous sensor streams, apply fixed fusion weights, or decouple abnormal-state prediction from bounded control updating, thereby limiting their practical reliability. This study proposes a confidence-aware multi-source sensor fusion and safety-bounded residual control framework designed to address these limitations. The proposed method integrates delay-aware synchronization, modality-specific temporal convolutional encoders, reliability-gated fusion, dual residual and anomaly prediction heads, and projected online controller adaptation. Supervised in-domain evaluation was conducted on the AURSAD dataset, while cross-domain transfer assessment was performed using the official voraus-AD test set through shared joint and electrical channels. Closed-loop tracking performance was further validated within a UR3e digital-twin environment. Across five independent runs, the proposed method achieved Macro-F1 scores of 88.4 ± 1.3% on AURSAD and 86.2 ± 1.5% in the transfer evaluation, surpassing Transformer-based model predictive control by 0.8 and 1.3 percentage points, respectively. The framework also attained a precision retention rate of 89.7 ± 1.6%, exceeding the strongest baseline by 3.6 percentage points. Cartesian root mean square error was 1.12 ± 0.08 mm, comparable to the 1.09 ± 0.08 mm of Transformer-based model predictive control with no statistically significant difference. These results demonstrate that the proposed framework primarily enhances interference tolerance, transfer stability, and bounded accuracy retention in collaborative robotic systems.

Keywords: collaborative robots; sensor fusion; accuracy retention; reliability-gated learning; safety-bounded control

Received: 06 May 2026
Revised: 29 June 2026
Accepted: 09 July 2026
Published: 12 July 2026

1. Introduction

Collaborative robots are increasingly deployed in digital manufacturing for precision tasks, automated screwdriving, workpiece handling, human–robot shared production, and small-batch high-mix manufacturing [1]. These applications require robots to maintain stable motion and positioning accuracy while interacting with changing tools, workpieces, and production conditions. However, the nominal repeatability specified under controlled conditions does not directly reflect the accuracy achieved during dynamic operation. Payload variation, joint friction, unintended contact, tool wear, sensor noise, and communication delay may all cause deviations between the commanded trajectory and the actual end-effector motion. Such deviations can reduce task quality, increase rework, and interrupt coordinated manufacturing processes.

A single joint encoder is insufficient for identifying the physical origin of these deviations [2]. Similar position errors may result from geometric calibration error, loadinduced compliance, transient contact force, motor or joint abnormalities, sensor drift, or temporary signal loss. These conditions require different corrective actions and cannot be reliably distinguished from joint position measurements alone. A more informative robot-state representation therefore requires the coordinated use of joint position, velocity, torque, motor current, end-effector force or torque, and controller command signals. Multi-source sensing can provide complementary mechanical, electrical, and process-level information, but its effectiveness depends on how heterogeneous signals are synchronized, fused, and incorporated into the control loop.

Existing studies still present three practical limitations. First, sensor streams often have different sampling rates and communication delays. Direct concatenation may therefore treat measurements from different physical moments as a single system state. Second, many fusion models use fixed or implicitly learned weights without estimating whether a sensor source is currently reliable. Consequently, noisy, drifting, or temporarily missing channels may continue to influence the fused representation. Third, prediction and control are frequently treated as separate tasks. Although some methods can detect abnormal operation or estimate trajectory error, they do not specify how predicted errors should update joint-level commands under velocity, torque, gain, and control-increment specifications [2].

To address these limitations, this study makes the following contributions: (1) A delay-aware synchronization mechanism is developed to align multirate joint, electrical, and process-sensing signals on a common control timeline through timestamp correction, interpolation, and validity masking. (2) A reliability-gated temporal fusion module is introduced to estimate time-varying confidence for each sensor source and adjust modality contributions under sensor noise, short-term dropout, and operating-condition changes. (3) A safety-bounded residual controller is designed to convert predicted Cartesian errors into constrained joint-level compensation commands while retaining the nominal controller and limiting online parameter updates within predefined operational boundaries [3].

The proposed framework is trained and externally evaluated using public collaborative-robot datasets, while a digital-twin environment is used to examine closed-loop compensation under controlled operational variations [1]. The model jointly predicts Cartesian residuals and abnormal operating states, and projection onto feasible regions is applied to controller gains and command updates. This design aims to improve dynamic accuracy retention, increase robustness to sensor degradation, and provide interpretable sensor weights and compensation increments. The imposed boundaries support transparent safety verification, although they do not constitute industrial safety certification or formal compliance approval.

2. Related Works

2.1. Multi-Source Sensing in Collaborative Robotics

Multi-source sensing is widely used to improve state observation in collaborative robotic systems. Joint encoders provide high-frequency position and velocity measurements, while motor current signals indirectly reflect actuator load and mechanical impedance [2]. These signals are readily available from robot controllers and require little additional hardware. Joint torque and end-effector force/torque sensors provide more direct information about contact interactions and payload variations, which is valuable in precision tasks such as component insertion, fastening operations, and human–robot interaction. Visual sensing further provides spatial information about tools, workpieces, and surrounding objects.

However, each modality has limitations. Encoder and current signals cannot always distinguish geometric error from contact-induced deformation. Force and torque measurements are affected by calibration drift and transient impacts. Vision offers stronger spatial interpretation but is more sensitive to illumination changes, occlusion, processing latency, and lower sampling frequency. These differences complicate multimodal integration.

Early fusion combines raw or low-level features before temporal modeling. It is simple and preserves cross-sensor information, but it is sensitive to scale differences and timestamp misalignment. Late fusion processes modalities separately and combines high-level representations, improving modularity but potentially weakening short-duration cross-modal relationships. Attention-based fusion can model dynamic dependencies, but attention weights represent feature relevance rather than sensor reliability. A highly weighted source may still be noisy, drifting, or temporarily unavailable [1]. Explicit reliability estimation is therefore needed when fusion outputs are used for online control.

2.2. Data-Driven Accuracy Compensation

Statistical filtering is commonly used for robot-state estimation due to its low computational cost, interpretable variables, and clear update procedures [4]. It performs well when system dynamics and noise can be reasonably modeled, but its effectiveness may decline under nonlinear friction, contact transitions, tool wear, and coupled electromechanical variations.

Recurrent neural networks and temporal convolutional models can represent nonlinear and history-dependent errors by learning relationships among joint motion, current, torque, and process signals. Prediction-based controllers can also use estimated future errors to compensate for short-term deviations. However, data-driven compensation remains dependent on the training distribution [5]. Under unseen payloads, tool degradation, sensor faults, or new trajectories, predicted corrections may become inaccurate or unstable.

Some studies report only offline regression error or anomaly-classification accuracy. These metrics indicate predictive performance but do not show whether the estimated residual improves closed-loop trajectory accuracy [6]. Control-oriented evaluation should therefore include tracking error, compensation stability, and response to external influences.

2.3. Dynamic and Safety-Constrained Control

PID controllers are widely used due to their simple structure and low implementation cost [7]. However, fixed gains exhibit limited adaptability to variations in payload, friction, contact stiffness, and tool condition.

Adaptive control adjusts parameters based on observed errors, but measurement noise or delays may lead to oscillatory or overly aggressive updates. Learning-based controllers can capture complex nonlinear dynamics; however, unregulated outputs risk generating abrupt commands or destabilizing a stable nominal controller. In collaborative manufacturing, learned compensation must remain auxiliary to the nominal control law and respect operational limits on velocity, torque, gain, and control-increment magnitude [8].

2.4. Research Gap and Positioning

Existing studies have separately examined sensor fusion, anomaly monitoring, and trajectory compensation. However, little attention has been devoted to a unified mechanism that aligns heterogeneous sensor streams, estimates source reliability, and transforms predicted errors into bounded online corrections. Specifically, temporal attention alone fails to yield an explicit measure of sensing confidence, and offline prediction accuracy does not guarantee stable closed-loop compensation. This study bridges the gap via a confidence-aware fusion and safety-projected residual control framework. The framework integrates delay-aware synchronization, source-specific reliability estimation, joint residual and anomaly prediction, and constrained controller updating [9]. Its core contribution lies in tightly coupling multisensor state representation with bounded-accuracy compensation, rather than treating sensing, prediction, and control as isolated stages.

3. Methodology

3.1. Overall Architecture

Figure 1 presents the proposed confidence-aware multi-source fusion and safety-bounded residual control framework. Its inputs include reference joint commands, measured joint positions and velocities, joint torques, motor currents, TCP force/torque or screwdriving-process signals, timestamps, and validity masks [4]. Delay correction, resampling, and missing-value masking align these heterogeneous signals onto a common control timeline.

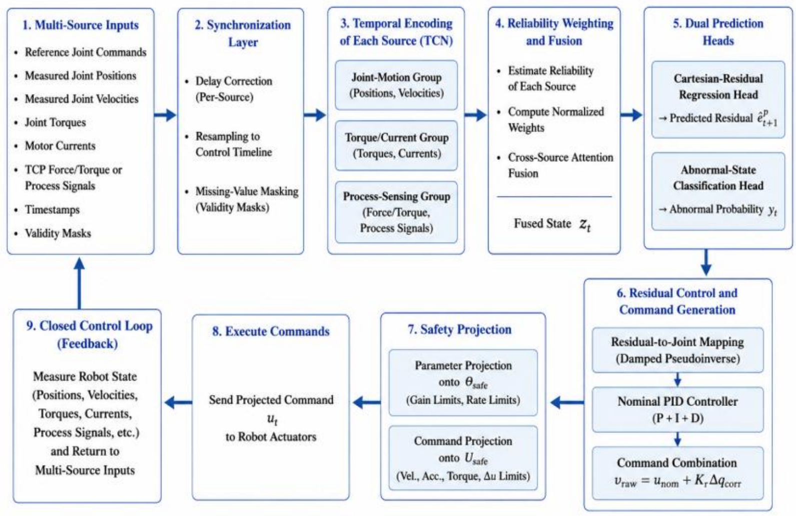

Figure 1. Overall Architecture of the Proposed Confidence-Aware Multi-Source Fusion and Safety-Bounded Residual Control Framework.

The synchronized inputs are grouped into joint-motion, torque/current, and process-sensing modalities. Each group is processed by a temporal convolutional network comprising three causal residual blocks with kernel size 3, dilation rates of 1, 2, and 4, and channel dimensions of 32, 64, and 64. Batch normalization, ReLU activation, and dropout with rate 0.2 are applied within each block. The resulting encoded representations are dynamically weighted by reliability scores and fused via four-head cross-source attention.

The fused state feeds two parallel heads: a Cartesian-residual regression head and an abnormal-state classification head, each implemented with fully connected layers of 64 and 32 units. The predicted residual is transformed into a joint-space correction and superimposed on a nominal PID controller output [10]. Controller parameters and commands are projected onto predefined feasible sets prior to execution. The measured robot state is fed back to the sensing layer, completing the closed-loop control structure.

3.2. Problem Formulation and Core Mechanism

Dynamic accuracy retention is formulated as a two-stage constrained perception-and-control problem. At control time t, the synchronized multi-source window is denoted by $\mathcal{X}_t$. The perception model with parameters ψ estimates the next-cycle Cartesian residual and abnormal-state probability [11]. During deployment, the controller with parameters θ_t converts these estimates into bounded compensation commands. The offline and online objectives are separated as

$$\psi^* = \arg\min_{\psi} \mathbb{E}[\mathcal{L}_{\mathrm{train}}], \mathrm{u}_t = \pi_{\theta_t}(\mathcal{X}_t, \hat{e}^{p}_{t+1}, \hat{y}_t), \theta_t \in \Theta_{\mathrm{safe}}, \mathrm{u}_t \in \mathcal{U}_{\mathrm{safe}}, \quad (1)$$

where $\mathcal{L}_{\mathrm{train}}$ is the offline perception loss, π_{θ_t} is the online control policy, and Θ_{safe} and $\mathcal{U}_{\mathrm{safe}}$ constrain controller parameters and executable commands. Thus, the perception network is trained offline, while only bounded controller parameters are adapted online [12].

3.3. Delay-Aware Data Synchronization

Let M denote the number of sensor sources. The k-th observation from source m is $x_k^{(m)}$, with timestamp $t_k^{(m)}$. Its aligned value at control time t is

$$\tilde{x}_t^{(m)} = \mathcal{J}_m\left(\left\{\left(t_k^{(m)}, x_k^{(m)}\right)\right\}_{k=1}^{N_m}, t + \hat{\delta}_m\right), \quad (2)$$

where $\mathcal{J}_m$ is the interpolation operator, N_m is the number of observations, and $\hat{\delta}_m$ is the source-specific delay estimated via cross-correlation on the training set. Continuous signals—including position, velocity, torque, current, and process variables—are linearly interpolated; discrete states employ nearest-neighbor holding. Intervals missing beyond a predefined threshold are indicated by validity masks rather than imputed values. Delay estimates and normalization statistics are derived exclusively from training data.

3.4. Temporal Encoding and Reliability Estimation

A causal window of length L is concatenated with its binary validity mask and processed by a modality-specific temporal convolutional network [13].

$$h_t^{(m)} = \mathrm{TCN}_m\left(\left[\tilde{x}_{t-L+1:t}^{(m)}, r_{t-L+1:t}^{(m)}\right]\right), \quad (3)$$

Here, $r^{(m)}$ denotes the binary validity mask and $h_t^{(m)} \in \mathbb{R}^{64}$ represents the encoded representation of source m.

Source uncertainty and the corresponding normalized fusion weight are computed as follows:

$$s_t^{(m)} = \mathrm{softplus}\left(w_m^T h_t^{(m)} + b_m\right), \alpha_t^{(m)} = \frac{\exp\left(-s_t^{(m)}\right)}{\sum_{j=1}^{M} \exp\left(-s_t^{(j)}\right)}, \quad (4)$$

In this formulation, w_m and b_m are trainable parameters; a larger value of $s_t^{(m)}$ reflects lower estimated reliability and yields a smaller fusion weight $\alpha_t^{(m)}$.

3.5. Reliability-Gated Fusion and Dual Prediction

The source representations are projected into a common feature space and fused using reliability-weighted aggregation and cross-source attention.

$$z_t = \mathrm{LayerNorm}\left[\sum_{m=1}^{M} \alpha_t^{(m)} P_m h_t^{(m)} + \mathrm{MHA}\left(P_1 h_t^{(1)}, \ldots, P_M h_t^{(M)}\right)\right], \quad (5)$$

where P_m is a trainable projection matrix, MHA denotes multi-head attention, and z_t is the fused state [14]. Reliability gating attenuates contributions from degraded sources, while attention models inter-source dependencies.

Cartesian residual labels and prediction outputs are defined as

$$e_t^p = f_{FK}\left(q_t^{ref}\right) - f_{FK}\left(q_t^{act}\right), \hat{e}_{t+1}^p = H_p(z_t), \hat{y}_t = \sigma(H_a(z_t)), m_t^p, m_t^a \in \{0,1\}, \quad (6)$$

where f_{FK} is the forward-kinematics mapping, q_t^{ref} and q_t^{act} are reference and measured joint vectors, and H_p and H_a are the residual-regression and abnormal-state classification heads. The binary label $y_t \in \{0,1\}$ indicates normal or abnormal operation. The task masks m_t^p and m_t^a indicate whether Cartesian residual and anomaly labels are available for a given sample.

Anomaly labels from the AURSAD dataset are used for supervised classification and in-domain evaluation. Residual labels are derived from samples with paired reference and measured joint states, supplemented by data from the UR3e digital twin [15]. The official voraus-AD test set serves exclusively for external transfer evaluation and is excluded from parameter optimization. During transfer testing, only jointly available joint and electrical channels are retained; unavailable process channels are marked invalid. The three AURSAD damaged-thread operations are excluded from supervised training and reserved for operation-level unknown-anomaly analysis.

3.6. Joint Training Objective

Given that the data sources do not consistently provide both label types, the perception model is trained using a task-masked objective:

$$\begin{aligned}\mathcal{L}_{train} = \; & m_t^p\left[\lambda_p \mathrm{Huber}\left(e_{t+1}^p, \hat{e}_{t+1}^p\right) + \lambda_s \parallel \hat{e}_{t+1}^p - \hat{e}_t^p \parallel_2^2\right] \\ & + m_t^a\left[\lambda_a \mathrm{BCE}(y_t, \hat{y}_t) + \lambda_b (\hat{y}_t - y_t)^2\right],\end{aligned} \quad (7)$$

where λ_p, λ_s, λ_a, and λ_b denote weighting coefficients for residual regression, temporal smoothness, anomaly classification, and probability calibration, respectively.

Each term is averaged only over samples where the corresponding task mask equals one. The Huber loss mitigates the impact of isolated residual outliers, while the smoothness term suppresses abrupt variations between consecutive predictions.

3.7. Dynamic Accuracy Retention Controller

The predicted Cartesian residual is transformed into a joint-space correction and integrated with the nominal PID control law:

$$u_t^{raw} = K_{p,t} e_t^q + K_{i,t} \sum_{\tau=1}^{t} e_\tau^q \Delta t + K_{d,t} \dot{e}_t^q - K_{r,t} J^\dagger(q_t) \hat{e}_{t+1}^p, (8)$$

where $e_t^q = q_t^{ref} - q_t^{act}$, $J^\dagger$ denotes the damped Jacobian pseudoinverse, and $K_{p,t}$, $K_{i,t}$, $K_{d,t}$, and $K_{r,t}$ represent the proportional, integral, derivative, and residual-compensation gains, respectively.

The online controller loss function is defined as:

$$\ell_t = \| e_t^q \|_2^2 + \lambda_{\Delta u} \| u_t^{raw} - u_{t-1} \|_2^2, \quad (9)$$

where the first term penalizes joint tracking error and $\lambda_{\Delta u}$ regulates temporal variation between consecutive control commands [5].

Confidence-scaled parameter updates and command projection are subsequently applied as:

$$c_t = 1 - \sum_{m=1}^{M} \alpha_t^{(m)} \bar{s}_t^{(m)}, \theta_{t+1} = \Pi_{\Theta_{safe}}[\theta_t - \eta c_t \nabla_\theta \ell_t], u_t = \Pi_{\mathcal{U}_{safe}}(u_t^{raw}), \quad (10)$$

where $\bar{s}_t^{(m)} \in [0,1]$ is the normalized source uncertainty, $c_t \in [0,1]$ denotes the overall confidence metric, η is the online update rate, and θ_t comprises the PID and residual-compensation gains [16]. Feasible sets enforce constraints on gain variation, joint velocity, acceleration, torque, and single-cycle command increments. When $\hat{y}_t$ exceeds the predefined abnormal-state threshold, parameter adaptation is temporarily suspended and conservative nominal gains are reinstated.

3.8. Training and Online Control Procedure

Offline training estimates source delays and learns the temporal encoders, reliability module, and prediction heads from operation-level sequences. During deployment, the trained model predicts the next-cycle residual and abnormal-state probability. Only bounded controller parameters are adapted online, and all commands undergo projection prior to execution.

Algorithm 1. Training and deployment of CMF-SRC

Input: Multi-source sequences $\mathcal{X}$, timestamps $\mathcal{T}$, reference commands $\mathcal{Q}^{ref}$, available labels $\mathcal{Y}$, task masks $\mathcal{M}$, window length L, and safety sets Θ_{safe} and $\mathcal{U}_{safe}$

Output: Trained perception model F_ψ, prediction heads H_p and H_a, and bounded controller parameters θ

1: Initialize perception parameters ψ and conservative controller parameters θ_0

2: Estimate source-specific delays and normalization statistics from training data

3: for epoch $= 1$ to E do

4: for each operation-level training sample do

5: Align all sensor streams on the common control timeline

6: Construct causal windows, validity masks, and task-availability masks

7: Encode each source using its modality-specific TCN

8: Estimate source uncertainties and normalized reliability weights

9: Fuse the source representations and compute both prediction outputs

10: Generate available Cartesian residual and abnormal-state labels

11: Compute Equation (7) and update only ψ

12: Freeze ψ after validation-based early stopping

13: for each online control step t do

14: Predict $\hat{e}_{t+1}^p$ and $\hat{y}_t$, compute Equation (8), and update θ_t using Equation (10)

15: Project and execute u_t ; if $\hat{y}_t$ exceeds the threshold, freeze adaptation and restore conservative gains

4. Results and Analysis

4.1. Experimental Setup

Experiments were conducted on a workstation equipped with an Intel Core i7-12700K processor, 32 GB RAM, and an NVIDIA RTX 3060 GPU running Ubuntu 22.04 LTS [5]. The implementation was based on Python 3.10 and PyTorch 2.3.1, using a URSim-based UR3e digital twin. Five independent experimental runs were performed using random seeds 13, 21, 42, 87, and 102. Full reproducibility was ensured by archiving Git commits, dependency specifications, dataset versions, configuration files, execution logs, random seeds, and plotting scripts.

Two publicly available datasets were employed. AURSAD comprises 2,045 UR3e screwdriving operations sampled at 100 Hz: 1,420 normal, 221 damaged-screw, 183 extra-component, 218 missing-screw, and three damaged-thread instances. It is licensed under CC BY 4.0. The damaged-thread cases were excluded from supervised training and reserved for unknown-anomaly evaluation [3]. The voraus-AD dataset captures six-axis Yu-Cobot pick-and-place tasks and provides 130 mechanical and electrical signals. Its official partition includes 948 normal operations for training and 1,174 test operations—comprising 419 normal and 755 anomalous cases across 12 distinct anomaly types. The dataset is released under CC BY-NC-SA 4.0, while its associated code is distributed under the MIT License.

The official 100-Hz version of voraus-AD was adopted. Each operation was segmented into non-overlapping 128-sample windows with a stride of 16, ensuring no window crossed operation boundaries. Z-score normalization parameters were computed exclusively from the training data. Missing sequences of up to five consecutive samples were forward-filled; longer gaps retained validity masks to preserve integrity. AURSAD was partitioned by operation into 70% training, 15% validation, and 15% testing subsets. The classification branch was trained on AURSAD and evaluated on both its test set and the official voraus-AD test set, using shared joint and electrical signal channels. The transfer threshold was determined solely on the AURSAD validation set. Digital-twin joint states served as inputs for residual learning and closed-loop performance assessment.

Baseline methods included an extended Kalman filter–PID controller, an unscented Kalman filter–adaptive PID controller, an LSTM-based residual controller, and a Transformer-based model predictive controller. For Kalman filter variants, normalized innovation magnitude served as the anomaly score; neural baselines relied on prediction residuals; and the proposed method used classification probability. All decision thresholds were optimized on validation data [2]. Macro-F1 scores were reported for in-domain AURSAD evaluation and cross-dataset AURSAD-to-voraus-AD transfer. Cartesian root-mean-square error and perturbation recovery rate—defined as the proportion of perturbed control steps achieving end-effector positional error ≤2 mm—were measured in the digital twin environment. Results are presented as mean ± standard deviation across five runs, with statistical significance assessed via paired two-sided t-tests (α = 0.05) and 95% confidence intervals.

4.2. Performance Comparison

Table 1 compares the proposed CMF-SRC with four baselines on AURSAD, voraus-AD, and the UR3e digital-twin control task. The proposed method achieved the highest Macro-F1 on both public datasets and the highest precision retention rate, although Transformer-based MPC obtained a slightly lower average Cartesian RMSE.

Table 1. Performance Comparison on Two Public Datasets and the UR3e Digital-Twin Environment, Reported as Mean ± Standard Deviation over Five Runs.

Method	AURSAD In-Domain Macro-F1 (%) ↑	voraus-AD Transfer Macro-F1 (%) ↑	Cartesian RMSE (mm) ↓	PRR (%) ↑
EKF–PID	78.2 ± 2.3	71.9 ± 2.6	1.84 ± 0.16	78.6 ± 2.9
UKF–Adaptive PID	80.7 ± 2.0	75.4 ± 2.3	1.62 ± 0.13	82.1 ± 2.5
LSTM Residual Controller	85.9 ± 1.7	81.8 ± 1.9	1.43 ± 0.11	84.8 ± 2.2
Transformer-based MPC	87.6 ± 1.5	84.9 ± 1.6	1.09 ± 0.08	86.1 ± 1.9
Proposed CMF-SRC	88.4 ± 1.3	86.2 ± 1.5	1.12 ± 0.08	89.7 ± 1.6

Compared with Transformer-based MPC, CMF-SRC improved AURSAD In-Domain Macro-F1 by 0.8 percentage points ($p=0.031$, 95% CI: 0.1--1.5) and voraus-AD Transfer Macro-F1 by 1.3 percentage points ($p=0.018$, 95% CI: 0.3--2.3). PRR increased by 3.6 percentage points ($p=0.009$, 95% CI: 1.5--5.7), indicating that more control steps remained within the 2-mm tolerance under operational perturbations. However, its RMSE was 0.03 mm higher than that of Transformer-based MPC, and the difference was not significant ($p=0.210$). This suggests that reliability gating and safety projection primarily enhanced consistency across varying operational conditions and cross-condition recognition rather than minimizing average error.

4.3. Ablation and Mechanism Verification

Table 2 evaluates the three core mechanisms. Removing delay-aware alignment increased tracking error during rapid motion transitions. Removing reliability gating caused the largest decline in both dataset-level Macro-F1 scores, particularly on voraus-AD, where heterogeneous abnormal conditions rendered fixed source contributions less effective. Removing safety projection slightly reduced average RMSE but lowered PRR by 2.9 percentage points. Stronger unrestricted corrections thus improved some moderate errors while occasionally introducing larger deviations. The full framework achieved the optimal balance between recognition accuracy and bounded control performance.

Table 2. Ablation Analysis of Synchronization, Reliability Gating, and Safety-Bounded Compensation.

Variant	AURSAD In-Domain Macro-F1 (%) ↑	voraus-AD Transfer Macro-F1 (%) ↑	Cartesian RMSE (mm) ↓	PRR (%) ↑
Full CMF-SRC	88.4 ± 1.3	86.2 ± 1.5	1.12 ± 0.08	89.7 ± 1.6
w/o delay-aware alignment	86.8 ± 1.5	84.4 ± 1.7	1.27 ± 0.10	87.1 ± 2.0
w/o reliability gating	85.5 ± 1.8	82.3 ± 1.9	1.33 ± 0.11	85.9 ± 2.1
w/o safety projection	88.1 ± 1.4	85.9 ± 1.6	1.08 ± 0.09	86.8 ± 1.9

4.4. Convergence Analysis

Figure 2 presents the mean training loss and validation Cartesian RMSE over five repeated runs, with shaded regions denoting ± one standard deviation. Moderate fluctuations occurred between epochs 20 and 35 due to modality dropout altering available sensor combinations and varying proportions of atypical samples across mini-batches. Validation RMSE decreased more consistently after reliability weights stabilized and settled within a narrow range by approximately epoch 65. Early stopping with a patience of 15 epochs was triggered when further reduction in training loss no longer improved validation RMSE. A small training–validation gap persisted but showed no sustained growth.

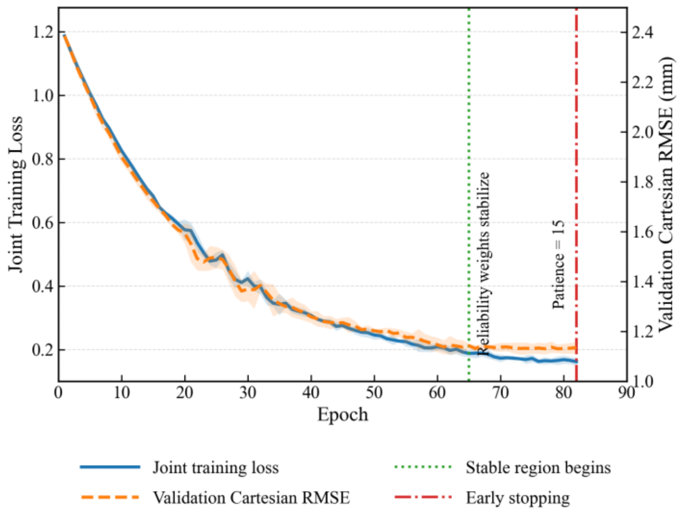

Figure 2. Training and Validation Convergence of the Proposed Model over Five Repeated Runs. Shaded Regions Denote ± One Standard Deviation.

4.5. Stability, Robustness, and Interpretability

Figure 3 evaluates performance under a 15% payload increase, 20% torque-channel signal loss, timestamp offsets of 10–30 ms, current noise equal to 5% of the training standard deviation, and the retained unknown anomaly category. PRR decreased from 91.1 ± 1.4% under nominal conditions to 88.3 ± 1.7% under payload variation, 87.4 ± 2.0% under torque signal loss, and 86.2 ± 2.3% at the largest timestamp offset. These reductions indicate improved robustness, though performance remains sensitive to significant sensing variations. Unknown-anomaly Macro-F1 reached 72.6 ± 3.0%, remaining below the results for known abnormal states.

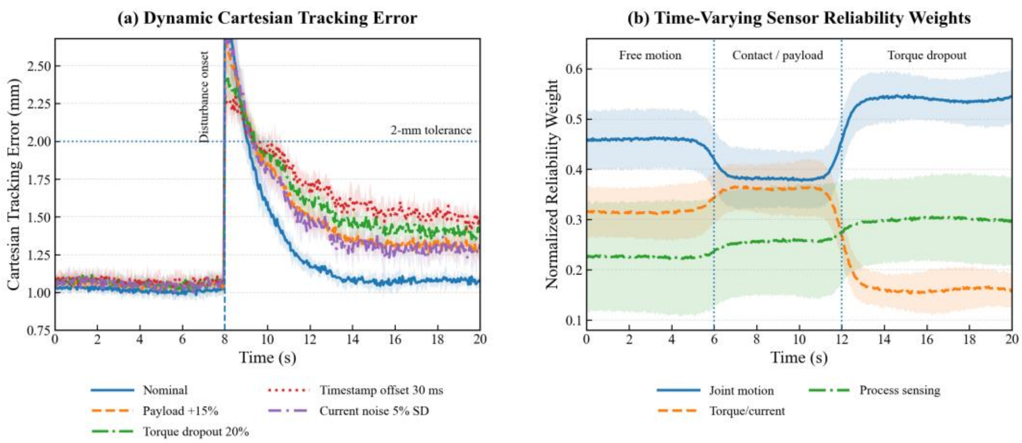

Figure 3. Stability, Robustness, and Interpretability Analysis: (A) Dynamic Cartesian Tracking Error under Nominal and Perturbed Conditions; (B) Time-Varying Reliability Weights of Joint-Motion, Torque/current, and Process-Sensing Sources. Shaded Regions Denote 95% Confidence Intervals over Five Runs.

During free motion, joint position and velocity received the largest weights. Under contact and payload changes, torque/current and process-sensor contributions increased. When torque signal loss was introduced, its mean weight decreased from 0.34 ± 0.04 to

0.16 ± 0.03, while the joint-motion weight increased from 0.42 ± 0.05 to 0.54 ± 0.04. These changes support the intended reliability mechanism but should not be interpreted as causal importance [8]. All commands remained subject to velocity, torque, gain, and single-cycle increment limits.

5. Conclusion

The results show that the three proposed mechanisms contributed to different aspects of collaborative-robot performance. Delay-aware synchronization reduced errors caused by asynchronous sensing, as removing this mechanism increased Cartesian RMSE from 1.12 ± 0.08 mm to 1.27 ± 0.10 mm. Reliability-gated fusion improved abnormal-state recognition under heterogeneous sensing conditions. CMF-SRC achieved Macro-F1 scores of 88.4 ± 1.3% on AURSAD and 86.2 ± 1.5% in the AURSAD-to-voraus-AD transfer evaluation, exceeding Transformer-based MPC by 0.8 and 1.3 percentage points, respectively. These differences were statistically significant on AURSAD (p=0.031) and voraus-AD (p=0.018). The safety-bounded residual controller produced a PRR of 89.7 ± 1.6%, which was 3.6 percentage points higher than the strongest baseline (p=0.009). However, its Cartesian RMSE of 1.12 ± 0.08 mm was slightly higher than the 1.09 ± 0.08 mm achieved by Transformer-based MPC, and the difference was not significant (p=0.210). The results therefore indicate that the main advantage of CMF-SRC lies in robustness to environmental variations and bounded accuracy retention rather than minimum average tracking error.

Several limitations remain. The two public datasets cover only screwdriving and pick-and-place operations, and some AURSAD anomaly categories contain very few samples. Digital-twin experiments cannot fully reproduce long-term production-line conditions, while forward-kinematics labels do not capture all calibration, compliance, and tool-deformation errors. The multi-source network also requires greater computation than conventional filtering controllers. In addition, safety projection limits command variation but does not constitute functional-safety certification.

Future work should conduct long-duration hardware-in-the-loop and physical-robot validation on UR3e and other collaborative platforms. Visual and temperature sensing may be incorporated while controlling model size and latency. Further research should examine few-shot transfer across robot platforms, continuous adaptation to tool wear and sensor drift, and formal verification of controller specifications and fallback behavior.

References

1. Z. Xu, "A Real-Time Precise Positioning Method for Indoor Robots Based on Multi-Sensor Fusion," *Journal of Social Science and Cultural Development*, vol. 2, no. 9, 2025.
2. Z. Cui, C. Sun, Z. Chen, J. A. Huang, and Y. Zhang, "Robot autonomous learning system based on multi-source sensor information fusion," *Robotica*, vol. 44, no. 2, pp. 540–564, 2026.
3. X. Ni, R. Su, F. Yu, Y. Sun, Z. Zhu, and D. Chen, "Multi-sensor fusion-based inter-row autonomous following navigation for collaborative harvesting robots," *Smart Agricultural Technology*, p. 102139, 2026.
4. Z. Ye and X. Li, "Research Status and Development Trends of Multi-Sensor Data Fusion Technology in Robotics," in *2026 IEEE 9th World Conference on Computing and Communication Technologies (WCCCT)*, Apr. 2026, pp. 398–403.
5. Z. Huang, G. Ye, P. Yang, and W. Yu, "Application of multi-sensor fusion localization algorithm based on recurrent neural networks," *Scientific Reports*, vol. 15, no. 1, p. 8195, 2025.
6. Y. T. Liu, R. Z. Sun, X. N. Zhang, L. Li, and G. Q. Shi, "An autonomous positioning method for fire robots with multi-source sensors," *Wireless Networks*, vol. 30, no. 5, pp. 3683–3695, 2024.
7. W. Huang, R. Zhang, and H. Lin, "Research on Intelligent Pipeline Robot Navigation and Defect Identification Based on Multi-Sensor Fusion," *Journal of the European Academy Open University*, vol. 1, no. 1, 2025.
8. Z. Sun, M. Kong, and G. Zhu, "Multi-source sensor data fusion and deep reinforcement learning anomaly detection method for power UAV inspection," *International Journal for Housing Science & Its Applications*, vol. 46, no. 3, p. 1997, 2025.
9. Y. Shi, L. Lao, X. Zheng, and P. Chen, "Real-time Spatiotemporal Synchronization for Human-robot Collaborative Systems: An Integrated Digital Twin Framework with Adaptive Data Processing," *Journal of Intelligent & Robotic Systems*, vol. 112, no. 1, p. 12, 2026.

10. M. Cui, H. Liu, W. Liu, and Y. Qin, "An adaptive unscented kalman filter-based controller for simultaneous obstacle avoidance and tracking of wheeled mobile robots with unknown slipping parameters," *Journal of Intelligent & Robotic Systems*, vol. 92, no. 3, pp. 489–504, 2018.
11. W. Huang, Y. Lin, M. Liu, and H. Min, "Velocity-aware spatial-temporal attention LSTM model for inverse dynamic model learning of manipulators," *Frontiers in Neurorobotics*, vol. 18, p. 1353879, 2024.
12. V. Lertpiriyasuwat, M. C. Berg, and K. W. Buffinton, "Extended Kalman Filtering Applied to a Two-Axis Robotic Arm with Flexible Links," *The International Journal of Robotics Research*, 2000.
13. S. Wu, J. Wu, X. Cao, F. Zhang, G. Yu, J. Zhao, and J. Duan, "Transformer-based explicit model predictive control with variable prediction horizon," *Engineering Applications of Artificial Intelligence*, vol. 172, p. 114345, 2026.
14. A. Mohammad, J. Piosik, D. Lehmann, T. Seel, and M. Schappler, "Fast Contact Detection via Fusion of Joint and Inertial Sensors for Parallel Robots in Human-Robot Collaboration," *IEEE Robotics and Automation Letters*, 2025.
15. Y. Cohen, A. Biton, and S. Shoval, "Fusion of computer vision and AI in collaborative robotics: A review and future prospects," *Applied Sciences*, vol. 15, no. 14, p. 7905, 2025.
16. Y. Zhou, L. Quang, C. Nieto-Granda, and G. Loianno, "Coped-advancing multi-robot collaborative perception: A comprehensive dataset in real-world environments," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6416–6423, 2024.