sensors

MDPI

Article

GRAG-Net: A Gated Residual Autoregressive–Exogenous GRU Soft Sensor for Dynamic SO_2 Estimation in Sulfur Recovery Units

Maha Mesfer Alghamdi

Department of Computer Science and Engineering, College of Applied Studies, King Saud University, Riyadh 11451, Saudi Arabia; mmesfer@ksu.edu.sa

Abstract

Accurate sulfur dioxide (SO_2) estimation is important for sulfur recovery unit (SRU) monitoring when reliable one-step information is required between analyzer observations. This study proposes the Gated Residual Autoregressive–Exogenous GRU Network (GRAG-Net), which separates multivariate process dynamics from strictly causal SO_2 history, predicts branch-specific residual corrections relative to the latest measured target, and combines them through a sample-dependent gate. The model was evaluated on 10,081 chronologically ordered industrial observations using three expanding future test periods and five final random seeds. Comparators included persistence, linear extrapolation, absolute and residual classical regressors, process-only and autoregressive GRUs, a joint GRU, LSTM–ARX, TCN–ARX, fixed fusion, and global gating. Across 15 descriptive fold–seed evaluations, GRAG-Net achieved MAE 0.00315 ± 0.00044 mole%, RMSE 0.00790 ± 0.00064 mole%, and $R^2 = 0.97753 \pm 0.00331$. Pooled seed-ensemble MAE and RMSE were 0.00291 and 0.00766 mole%, reductions of 32.5% and 18.0% relative to AR–GRU. Independently trained ablations identified residual reconstruction and sample-dependent fusion as the main architectural contributors, while dependence-aware bootstrap analysis, causal gate diagnostics, lag analysis, and cross-fold/cross-seed SHAP stability supported robust and interpretable one-step soft sensing.

Keywords: sulfur recovery unit; sulfur dioxide; dynamic soft sensor; gated recurrent unit; autoregressive–exogenous modeling; residual learning; adaptive fusion; time series regression

Check for updates

Academic Editor: Hyoungshick Kim

Received: 13 July 2026
Revised: 18 September 2026
Accepted: 24 September 2026
Published: 29 September 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Sulfur dioxide (SO_2) is an important atmospheric pollutant associated with industrial combustion, mineral processing, petroleum refining, and other sulfur-intensive operations. Following emission, SO_2 participates in atmospheric oxidation and aerosol-formation processes that contribute to acid deposition, visibility degradation, and fine-particle formation [1]. Exposure to polluted air is also associated with a substantial global burden of respiratory and cardiovascular disease [2]. In petroleum refineries, accurate monitoring of sulfur compounds is therefore required not only for environmental compliance, but also for maintaining stable process operation, detecting abnormal conditions, and supporting timely corrective action.

Sulfur recovery units (SRUs) convert hydrogen-sulfide-rich acid gases into elemental sulfur, commonly through a thermal stage followed by catalytic Claus conversion and

downstream tail-gas treatment. The residual H_2S and SO_2 concentrations depend on interacting feed conditions, airflow distribution, reaction conditions, catalyst state, and process inertia [3,4]. These relationships are nonlinear and time-dependent, and the effects of an operating adjustment may not be observed immediately because of transport delays and internal process dynamics. Hardware analyzers provide the reference gas-concentration measurements, but their readings can be delayed and their operation requires calibration and maintenance. Data-driven soft sensors can therefore provide complementary real-time estimates for monitoring and diagnostic support [3,5]. In the deployment setting considered here, the hardware analyzer remains the reference measurement source: GRAG-Net estimates the current SO_2 concentration from process measurements available through time t and measured analyzer history available only through $t-1$. The soft sensor is therefore intended to complement the analyzer by providing a model-based one-step estimate between successive validated measurements or during measurement delay; it is not intended to permanently replace the physical analyzer. Continued analyzer measurements remain necessary for reference supervision, performance monitoring, and provision of the causal autoregressive history. Sustained analyzer-outage forecasting, in which predictions must recursively replace missing analyzer values, is outside the scope of this study.

A soft sensor estimates a difficult, delayed, or costly quality variable from routinely measured process information. Data-driven soft sensors are particularly attractive when sufficiently accurate first-principles models are unavailable or too expensive for online use [6–8]. Their practical development, however, is complicated by correlated measurements, nonlinear behavior, temporal dependence, measurement imperfections, limited target observations, and changes in operating conditions. Model reliability can deteriorate under process drift, catalyst aging, equipment fouling, raw-material variation, and transitions between operating regimes [9,10]. These characteristics make both dynamic representation and leakage-aware temporal validation essential for industrial soft sensing.

Research on the benchmark SRU process reflects the broader evolution of the field. Early studies considered neural soft analyzers and linear or nonlinear autoregressive models with exogenous inputs [3,11]. Bidar et al. introduced a multi-state-dependent parameter formulation to represent changes in operating state [12], whereas Yuan et al. developed a stacked isomorphic autoencoder for nonlinear feature extraction [13]. Dynamic sequence modeling was later investigated using echo-state networks and recurrent neural networks [14,15]. More recent work has addressed transferability and adaptation through transfer–incremental learning [16], as well as limited labeled data and nonlinear target decomposition through an ISSA–VMD–ESN framework [17]. Collectively, these studies demonstrate the importance of temporal modeling, nonlinear representation, and adaptation in SRU soft sensing.

Dedicated SO_2 modeling has also progressed toward increasingly complex temporal architectures. Thameem et al. compared conventional machine learning methods, multilayer perceptrons, one-dimensional convolutional networks, LSTMs, and a CNN–LSTM encoder–decoder for multi-horizon SO_2 forecasting in an industrial SRU [4]. Sun et al. subsequently proposed a multistage framework combining variational mode decomposition, CEEMDAN, entropy-guided component reconstruction, an enhanced PatchTST model, and metaheuristic hyperparameter optimization for SRU sulfur gas forecasting [18]. Li and Hao developed a dual-path dynamic soft sensor in which a TCN–Transformer processes auxiliary process variables and a GRU processes historical target values; the two learned representations are combined by summation before final prediction [19]. These recent studies confirm that process-variable histories and target histories provide complementary information. They also expose a remaining question: how should their relative contributions be adjusted when the local operating condition changes?

Existing SRU models generally improve the overall representation through deeper encoders, decomposition, transfer learning, or fixed feature fusion. However, three limitations remain relevant to the present problem. First, process measurements and historical SO_2 values represent physically different predictive mechanisms. The latter primarily describe persistence and process inertia, whereas the former provide evidence of process-induced deviation. Embedding both sources in one representation, or combining them using a fixed operation, does not explicitly preserve this distinction. Second, the reliability of the two sources is sample dependent: target persistence can be highly effective under quasi-steady operation but less responsive during rapid transitions, while exogenous airflow information may become more valuable when the process state changes. Third, average test metrics alone do not reveal whether an adaptive fusion mechanism responds meaningfully to branch disagreement, changing operating conditions, or high-change target regimes. In addition, performance estimation for nonstationary time series should preserve temporal order and should separate model selection from evaluation on future observations [20].

To address these limitations, this study proposes the *Gated Residual Autoregressive–Exogenous GRU Network* (GRAG-Net) for one-step dynamic SO_2 soft sensing. GRAG-Net contains two independently parameterized recurrent branches. The process branch encodes a multivariate history of five routinely measured airflow variables, whereas the autoregressive branch encodes only the strictly causal SO_2 history. Rather than predicting the absolute concentration independently, each branch estimates a residual correction relative to the most recent available analyzer value. A context-dependent gate then combines the two corrections according to

$$\widehat{y}_t = y_{t-1} + g_t \Delta\widehat{y}_t^{x} + (1 - g_t)\Delta\widehat{y}_t^{y}, \qquad 0 \le g_t \le 1, \tag{1}$$

where $\Delta\widehat{y}_t^{x}$ and $\Delta\widehat{y}_t^{y}$ denote the process-driven and autoregressive residual corrections, respectively. The gate is conditioned on the projected features of both branches, the latest SO_2 value, the disagreement between the two proposed corrections, and the most recent change in the process variables. Thus, GRAG-Net performs sample-wise soft routing between process-driven and history-driven information instead of applying a global fusion weight. A robust Huber objective is used to reduce excessive sensitivity to isolated large errors [21].

The main contributions of this study are as follows:

1. A dual-branch residual architecture is developed in which exogenous process dynamics and autoregressive SO_2 persistence are encoded separately and fused through a sample-dependent reliability gate.
2. The estimation task is defined explicitly as causal rolling one-step soft sensing: process measurements are available through time t, measured SO_2 history is available only through $t - 1$, and neither the current target nor future targets are supplied to the model. Sustained analyzer-outage forecasting is therefore outside the scope of the present study.
3. The comparative evaluation is expanded to include persistence and linear extrapolation, absolute and residual classical ARX regressors, strict process-only and autoregressive GRUs, a joint GRU, LSTM–ARX, TCN–ARX, equal-weight fusion, and a global-gate control, in addition to the proposed dynamic gate.
4. Architectural claims are tested using independently trained ablations of the residual formulation, fusion mechanism, branch supervision, gate-balance regularization, disagreement input, and process-change input. Chronological folds are treated as the primary evaluation periods, and paired moving-block bootstrap intervals are used to quantify uncertainty in sample-level error differences.

5. The diagnostic analysis is made inference-aware: gate associations are evaluated against causal inputs, process–target delays are estimated using training-only lagged correlations, and SHAP explanations are aggregated across folds and seeds with explicit rank-stability analysis.

The experiments use a chronologically ordered industrial benchmark containing 10,081 observations, five airflow-related process measurements, and SO_2 concentration as the target [3,22]. The task is formulated as causal one-step soft sensing: process measurements available at time t are combined with target observations available only through $t-1$. The current target and all future target values are excluded from the model input.

The remainder of this paper is organized as follows. Section 2 describes the dataset, causal sample construction, nested model development protocol, GRAG-Net architecture, benchmark models, evaluation criteria, and interpretability procedures. Section 3 presents the comparative and diagnostic results. Section 4 summarizes the principal findings, industrial implications, limitations, and future research directions.

2. Materials and Methods

This section describes the industrial sulfur recovery unit (SRU) dataset, the causal formulation of the dynamic soft-sensing task, the proposed Gated Residual Autoregressive–Exogenous GRU Network (GRAG-Net), the reference models, and the nested chronological model development protocol. Particular attention is given to preserving the temporal ordering of the plant records and to maintaining a strict separation between hyperparameter selection and final performance estimation. All preprocessing parameters, model configurations, and epoch-selection decisions were derived exclusively from the training information available within the corresponding chronological fold. The evaluation framework further considers chronological robustness, variability across random initializations, target-change regimes, residual behavior, and model interpretability.

2.1. Industrial Sulfur Recovery Process and Dataset

2.1.1. Process Background and Data Source

The experimental data were obtained from a benchmark industrial sulfur recovery process associated with the fourth tail-gas line of an Italian refinery [3,22]. An SRU converts hydrogen-sulfide-rich acid gas streams into elemental sulfur, typically through a high-temperature thermal stage followed by catalytic Claus conversion and downstream tail-gas treatment. In the studied process, acid gas streams originating from the monoethanolamine (MEA) and sour-water-stripping (SWS) sections are combined with controlled air streams before entering the reaction system. The distribution of combustion and secondary air affects the stoichiometric balance of the sulfur-recovery reactions and, consequently, the residual sulfur-compound concentrations measured in the treated tail gas.

The mapping from routinely measured airflow variables to the resulting SO_2 concentration is nonlinear and time dependent. Variations in acid gas loading, air allocation, reaction conditions, catalyst state, and downstream treatment do not necessarily affect the measured tail-gas composition instantaneously. Instead, transport delay, process inertia, nonlinear reaction behavior, and interactions among operating variables can produce delayed and regime-dependent responses. These characteristics make the dataset suitable for evaluating dynamic soft sensors that explicitly retain both recent process history and target persistence.

The observations were recorded at one-minute intervals. The original benchmark contains five process measurements and two sulfur gas outputs, H_2S and SO_2. The present study focuses exclusively on SO_2 soft sensing; therefore, the analyzed data matrix contains

the five process inputs and the SO_2 target field, denoted by y2. Following column selection and numeric conversion, the retained dataset comprised

$$N = 10{,}081 \tag{2}$$

complete observations in their original chronological order. No missing values were present in the retained variables, and no interpolation or imputation was therefore required. The five deposited process channels are supplied on a scaled $[0, 1]$ benchmark representation, whereas the SO_2 target retains the concentration scale used in the originating SRU data and is reported in mole%. This interpretation is consistent with both the original refinery study and the recent public benchmark release [3,18]. The chronological order was preserved throughout the study, and no random shuffling was applied before constructing the training, validation, and test periods.

2.1.2. Process Variables and Prediction Target

Let

$$\mathbf{x}_t = [x_{1,t}, x_{2,t}, x_{3,t}, x_{4,t}, x_{5,t}]^\top \in \mathbb{R}^5 \tag{3}$$

denote the process measurements available at sampling instant t. The corresponding target is the SO_2 concentration y_t, reported in mole%.

The five airflow-related channels were treated as exogenous process measurements because they are routinely available from the plant instrumentation, whereas SO_2 represents the difficult-to-measure quality variable estimated by the soft sensor. The process channels retain their benchmark scaling, while SO_2 is reported in mole%. Physical variable names are retained in all tables, figures, lagged-feature labels, and SHAP explanations to preserve process-level interpretability.

2.1.3. Descriptive and Distributional Characteristics

Table 1 reports descriptive statistics for the complete benchmark dataset. These statistics are provided only to characterize the benchmark; they were not used to preprocess validation or test observations. Fold-specific preprocessing parameters were estimated from the relevant training block, as described in Section 2.2.

The SO_2 concentration is concentrated within a comparatively narrow operating range, with a mean of 0.1706, a median of 0.1665, and a standard deviation of 0.0569. Its third quartile is 0.1923, despite a maximum of 1.0 mole%, indicating that large concentration values are relatively infrequent. In contrast, the process measurements span broader and heterogeneous operating ranges. The median values of airflow in the SWS zone (x_4) and secondary airflow (x_5) are 0.7225 and 0.6953, respectively, showing that these channels operate near their upper benchmark-scaled regions during a substantial part of the observation period. Airflow in the MEA zone (x_3) is more centrally concentrated and exhibits the smallest process-variable standard deviation.

Table 1. Descriptive statistics of the industrial SRU benchmark ($N = 10{,}081$).

Variable	Mean	Std.	Minimum	Q_1	Median	Q_3	Maximum
MEA airflow (x_1)	0.5623	0.2054	0.0000	0.4130	0.5502	0.6977	1.0000
SWS airflow (x_2)	0.5623	0.2277	0.0000	0.3567	0.5857	0.7678	1.0000
Airflow in the MEA zone (x_3)	0.4461	0.1519	0.0000	0.3420	0.4494	0.5360	1.0000
Airflow in the SWS zone (x_4)	0.5940	0.2420	0.0000	0.4100	0.7225	0.7395	1.0000
Secondary airflow (x_5)	0.6011	0.2029	0.0000	0.5091	0.6953	0.7039	1.0000
SO_2 concentration (mole%)	0.1706	0.0569	0.0000	0.1435	0.1665	0.1923	1.0000

The histogram and density plots in Figure 1 show that several process variables have heterogeneous and visibly multimodal distributions. These modes indicate recurring

operating regions rather than one stationary Gaussian regime. The SO_2 target has a narrow dominant distribution with a sparse upper tail. This imbalance is relevant because a model may achieve a low average error in the dominant quasi-steady region while remaining less accurate during rare concentration excursions.

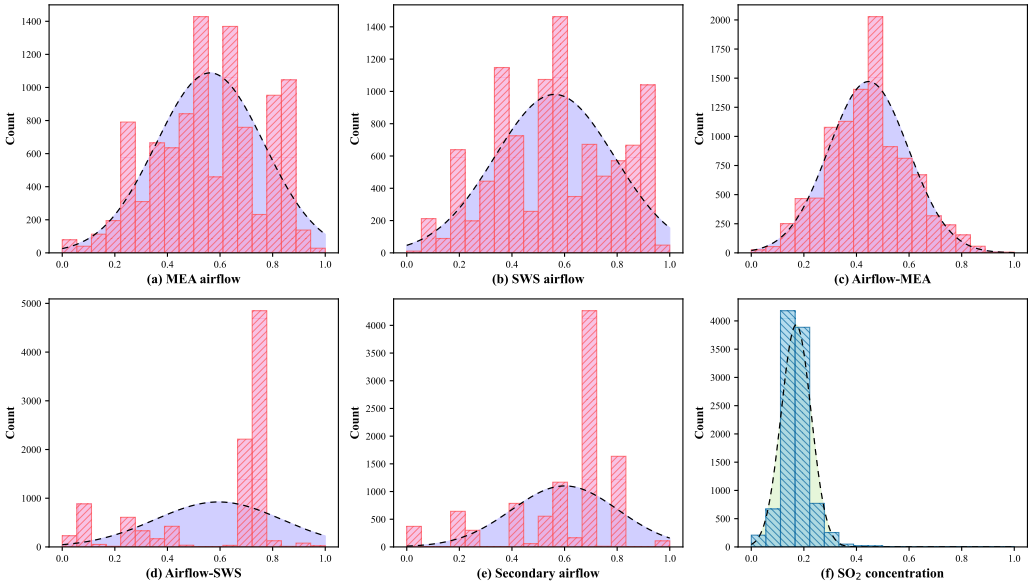


Figure 1. Histogram and density distributions of the five benchmark-scaled SRU process variables and SO_2 concentration. The process measurements exhibit heterogeneous and partly multimodal operating distributions, whereas the target is concentrated within a narrow dominant range with a sparse upper tail.

2.1.4. Correlation Structure and Target Persistence

The Pearson correlation structure is shown in Figure 2. Strong dependence is present among selected process variables. The largest positive process-input correlations are observed between MEA airflow and SWS airflow,

$$r(x_1, x_2) = 0.913, \tag{4}$$

and between airflow in the SWS zone and secondary airflow,

$$r(x_4, x_5) = 0.876. \tag{5}$$

These values indicate substantial coupling and partial redundancy among the airflow measurements. In contrast, the zero-lag correlations between the five process variables and SO_2 range approximately from -0.117 to 0.133.

The weak instantaneous input–target correlations do not imply that the process measurements are uninformative. Instead, they suggest that their effects on SO_2 are delayed, nonlinear, dependent on interactions among variables, or conditional on the operating regime. This observation motivates the use of historical multivariate process windows rather than a static mapping based only on measurements recorded at the current instant.

In contrast to the weak zero-lag process–target associations, the SO_2 sequence exhibits pronounced short-term persistence. The autocorrelation coefficients at lags one through five are

$$[\rho_y(1), \rho_y(2), \rho_y(3), \rho_y(4), \rho_y(5)] = [0.9531, 0.8487, 0.7330, 0.6273, 0.5431]. \tag{6}$$

The lag-one autocorrelation of 0.9531 confirms that the most recent SO_2 observation is a strong local predictor and motivates its use as the persistence reference in GRAG-Net. However, the gradual decay across subsequent lags shows that the useful temporal information is not restricted to a single previous observation. This supports a sequence-based autoregressive encoder with a history length selected through nested hyperparameter optimization.

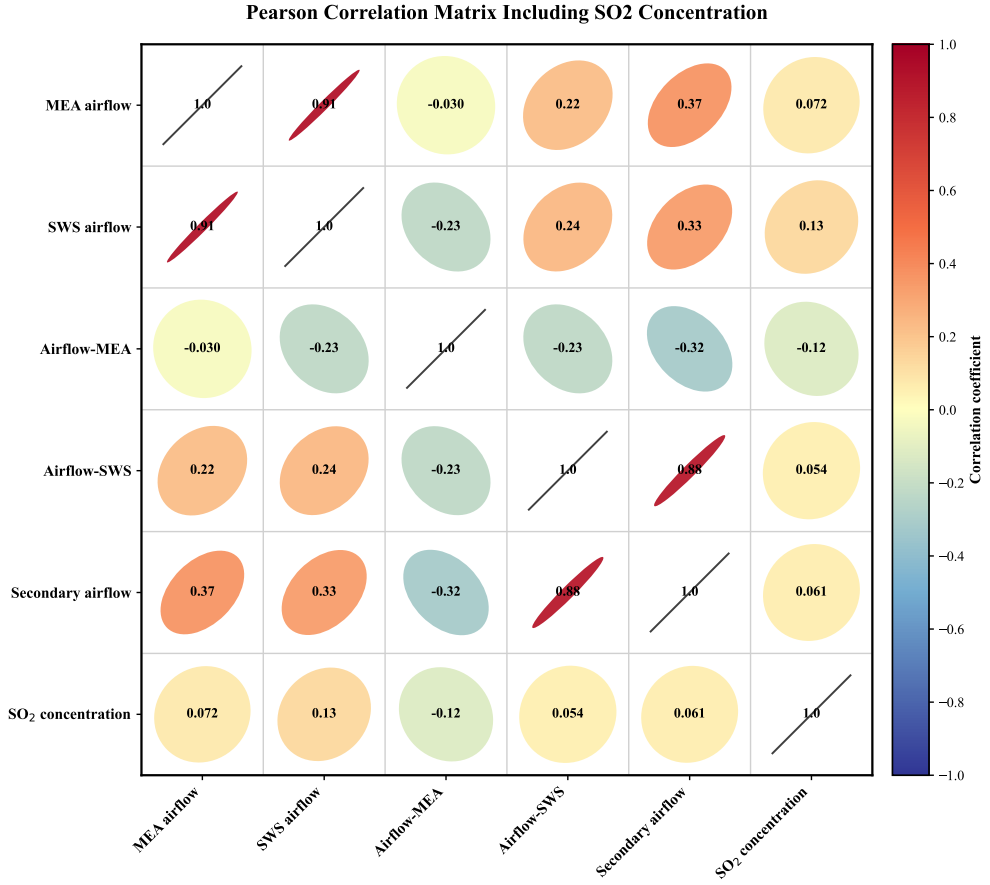


Figure 2. Pearson correlation ellipse matrix for the five SRU process variables and SO_2 concentration. Ellipse orientation indicates the sign of the relationship, whereas increasing eccentricity denotes stronger linear dependence.

Taken together, the exploratory analysis identifies three properties that directly motivate the proposed methodology: (i) heterogeneous and partly multimodal process-variable distributions; (ii) weak instantaneous but potentially delayed and nonlinear process–target relationships; and (iii) strong short-term target persistence. These characteristics support the separation of process history and target history into two distinct temporal pathways before their contributions are fused dynamically.

2.2. Problem Formulation and Causal Sample Construction

2.2.1. One-Step Dynamic Soft-Sensing Formulation

The objective is to estimate the current SO_2 concentration y_t from two information sources that are available at sampling instant t: (i) the recent history of the five routinely measured process variables and (ii) a strictly causal history of previously observed SO_2 values. The task is therefore formulated as *one-step dynamic soft sensing*, rather than unconstrained recursive forecasting. Current process measurements are assumed to be available from the distributed control system at time t, whereas the target history channel terminates at $t-1$. Consequently, neither the current target y_t nor any future target value is exposed to the model.

Let the instantaneous process vector be

$$\mathbf{x}_t = [x_{1,t}, x_{2,t}, x_{3,t}, x_{4,t}, x_{5,t}]^\top \in \mathbb{R}^5. \tag{7}$$

A multivariate process-history window of length L_x is defined as

$$\mathbf{X}_t = [\mathbf{x}_{t-L_x+1}, \mathbf{x}_{t-L_x+2}, \ldots, \mathbf{x}_t]^\top \in \mathbb{R}^{L_x \times 5}, \tag{8}$$

while the strictly causal SO_2 history of length L_y is

$$\mathbf{Y}_t = \left[y_{t-L_y}, y_{t-L_y+1}, \ldots, y_{t-1}\right]^\top \in \mathbb{R}^{L_y \times 1}. \tag{9}$$

The complete information set available for estimating y_t is therefore

$$\mathcal{I}_t = \{\mathbf{x}_\tau\}_{\tau=t-L_x+1}^{t} \cup \{y_\tau\}_{\tau=t-L_y}^{t-1}. \tag{10}$$

The general soft-sensing mapping is expressed as

$$\widehat{y}_t = \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{X}_t, \mathbf{Y}_t), \tag{11}$$

where $\mathcal{F}_{\boldsymbol{\theta}}(\cdot)$ denotes a nonlinear estimator with parameters $\boldsymbol{\theta}$. For GRAG-Net, y_{t-1} is extracted from the final element of $\mathbf{Y}_t$ and is used explicitly as both a gating-context variable and the residual reference in the final reconstruction. The proposed mapping can therefore be written more specifically as

$$\widehat{y}_t = \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{X}_t, \mathbf{Y}_t, y_{t-1}). \tag{12}$$

The two histories represent different predictive mechanisms. The process window $\mathbf{X}_t$ describes the recent operating trajectory and provides exogenous evidence of process-driven changes, whereas $\mathbf{Y}_t$ captures target persistence and process inertia. Their history lengths are allowed to differ because the useful temporal extent of the process variables need not match that of the target. Both L_x and L_y are therefore selected independently within the nested hyperparameter-optimization procedure.

2.2.2. Fold-Specific Standardization

Although the process channels are supplied on a scaled benchmark representation and SO_2 is retained in mole%, an additional z-score transformation was applied to improve numerical conditioning and to ensure that each model was developed using only the distributional information available in its training period. For process variable x_j,

$$\widetilde{x}_{j,t} = \frac{x_{j,t} - \mu_{x_j}^{\mathrm{tr}}}{\sigma_{x_j}^{\mathrm{tr}}}, \qquad j = 1, \ldots, 5, \tag{13}$$

and the target was transformed as

$$\widetilde{y}_t = \frac{y_t - \mu_y^{\mathrm{tr}}}{\sigma_y^{\mathrm{tr}}}. \tag{14}$$

Here, $\mu_{x_j}^{\mathrm{tr}}$, $\sigma_{x_j}^{\mathrm{tr}}$, μ_y^{tr}, and σ_y^{tr} were estimated exclusively from the training block available at the corresponding stage of the nested protocol. During inner hyperparameter optimization, the transformations were fitted only to the inner-training subset and were then applied unchanged to the inner-validation subset. After hyperparameter selection, new transforma-

tions were fitted to the complete outer-training block and applied unchanged to its held-out outer-test block.

The standardized history tensors are denoted by $\widetilde{\mathbf{X}}_t$ and $\widetilde{\mathbf{Y}}_t$. Neural-model training and prediction were performed in the standardized target domain. Each estimate was subsequently returned to the original benchmark scale using

$$\widehat{y}_t = \sigma_y^{\mathrm{tr}} \widehat{\widetilde{y}}_t + \mu_y^{\mathrm{tr}}, \tag{15}$$

before computing the reported evaluation metrics.

2.2.3. Sliding-Window Sample Generation

Dynamic samples were constructed with a stride of one observation. Under the zero-based indexing used in the implementation, the first admissible target index for a given pair (L_x, L_y) is

$$t_{\min} = \max\left(L_x - 1, L_y\right), \tag{16}$$

which guarantees that both the complete process window in Equation (8) and the complete target history sequence in Equation (9) are available. For each admissible t, one supervised sample was formed as

$$\left(\widetilde{\mathbf{X}}_t, \widetilde{\mathbf{Y}}_t, \widetilde{y}_t\right). \tag{17}$$

No downsampling, interpolation, or random reordering was applied during window construction.

The history lengths were treated as model hyperparameters rather than fixed a priori. The searched values were

$$L_x \in \{16, 32, 64\}, \qquad L_y \in \{4, 8, 16, 32\}. \tag{18}$$

Because the records were sampled at one-minute intervals, these values correspond to process histories of 16, 32, or 64 min and target histories of 4, 8, 16, or 32 min.

2.2.4. Causality and Chronological Boundary Handling

The chronological split boundaries were defined in terms of prediction indices, not isolated window matrices. At the beginning of a validation or test block, observations immediately preceding the boundary were retained as historical context when they belonged to the information set $\mathcal{I}_t$. This treatment reflects online deployment: when the first sample of a future period is estimated, the recent process measurements and the valid analyzer observations recorded before that instant are already available.

For every prediction index t, the process sequence terminates at $\mathbf{x}_t$ and the target sequence terminates at y_{t-1}. Thus, historical context may cross a training–validation or training–test boundary, but the prediction target itself and all future values remain excluded. The procedure does not create a gap between adjacent chronological blocks and does not discard the first L_x or L_y observations of each held-out period unnecessarily.

During outer-test evaluation, each prediction uses the measured SO_2 history available through $t-1$. Predictions are not recursively fed back as substitutes for subsequent analyzer measurements. The reported task therefore represents a rolling one-step inferential setting in which a recent valid target history is available for every sample. This assumption should be distinguished from prolonged analyzer-outage forecasting, which would require recursive prediction and is outside the scope of the present study.

2.2.5. Input Representation for the Baseline Models

The classical autoregressive–exogenous baselines cannot directly process two-dimensional sequences. Their standardized process and target histories were therefore flattened and concatenated:

$$\mathbf{v}_t = \left[\mathrm{vec}\left(\widetilde{\mathbf{X}}_t\right); \widetilde{\mathbf{Y}}_t\right] \in \mathbb{R}^{5L_x+L_y}. \tag{19}$$

The same causal information was consequently supplied to all learnable models, while only its representation differed. The support vector regressor and tree ensembles received $\mathbf{v}_t$, whereas GRAG-Net retained the temporal organization of $\widetilde{\mathbf{X}}_t \in \mathbb{R}^{L_x\times 5}$ and $\widetilde{\mathbf{Y}}_t \in \mathbb{R}^{L_y\times 1}$ and processed the two histories through independent recurrent branches. This design ensures that comparative performance differences are not caused by unequal access to process or target history information.

2.3. Proposed GRAG-Net Architecture

2.3.1. Design Rationale and Architectural Overview

The proposed Gated Residual Autoregressive–Exogenous GRU Network (GRAG-Net) is designed to model two complementary mechanisms governing one-step SO_2 soft sensing. The first mechanism is the strong short-term persistence of the target, which makes the most recent valid SO_2 observation an informative reference under quasi-steady operation. The second mechanism is the process-driven deviation from this recent trajectory, which is reflected in the evolving histories of the five airflow-related variables. Because the relative usefulness of these two information sources can change from one observation to another, GRAG-Net models them through independent temporal branches and combines their residual corrections using a sample-dependent reliability gate.

Let $\widetilde{\mathbf{X}}_t \in \mathbb{R}^{L_x\times 5}$ and $\widetilde{\mathbf{Y}}_t \in \mathbb{R}^{L_y\times 1}$ denote the standardized process and target history windows defined in Section 2.2. Rather than predicting the absolute target independently, each branch estimates a correction relative to the latest standardized SO_2 observation $\widetilde{y}_{t-1}$. The complete model is summarized by

$$\Delta\widehat{\widetilde{y}}_t^{\,x} = f_x\left(\widetilde{\mathbf{X}}_t; \boldsymbol{\theta}_x\right), \tag{20}$$

$$\Delta\widehat{\widetilde{y}}_t^{\,y} = f_y\left(\widetilde{\mathbf{Y}}_t; \boldsymbol{\theta}_y\right), \tag{21}$$

$$g_t = f_g\left(\mathbf{c}_t; \boldsymbol{\theta}_g\right), \qquad g_t \in [0,1], \tag{22}$$

$$\widehat{\widetilde{y}}_t = \widetilde{y}_{t-1} + g_t\Delta\widehat{\widetilde{y}}_t^{\,x} + (1-g_t)\Delta\widehat{\widetilde{y}}_t^{\,y}. \tag{23}$$

Here, $f_x(\cdot)$ and $f_y(\cdot)$ denote the process and autoregressive experts, respectively, while $f_g(\cdot)$ denotes the gating network. A value of g_t approaching one gives greater influence to the process-driven correction, whereas a value approaching zero favors the autoregressive correction.

This residual formulation is particularly suitable for the studied target. The skip connection through $\widetilde{y}_{t-1}$ explicitly preserves the strong persistence component, allowing the trainable branches to focus on the remaining change:

$$\Delta\widetilde{y}_t = \widetilde{y}_t - \widetilde{y}_{t-1}. \tag{24}$$

Thus, GRAG-Net does not force either branch to repeatedly learn the near- identity mapping between consecutive target observations. Instead, both branches solve the more focused problem of estimating how the current target departs from its latest measured value.

Figures 3 and 4 present the architecture in two complementary parts. Figure 3 illustrates the independent temporal encoders, projection modules, and branch-specific correction heads, while Figure 4 shows the dynamic context, reliability-gate estimation,

residual fusion, and final reconstruction. Sequence dimensions are denoted symbolically by L_x and L_y.

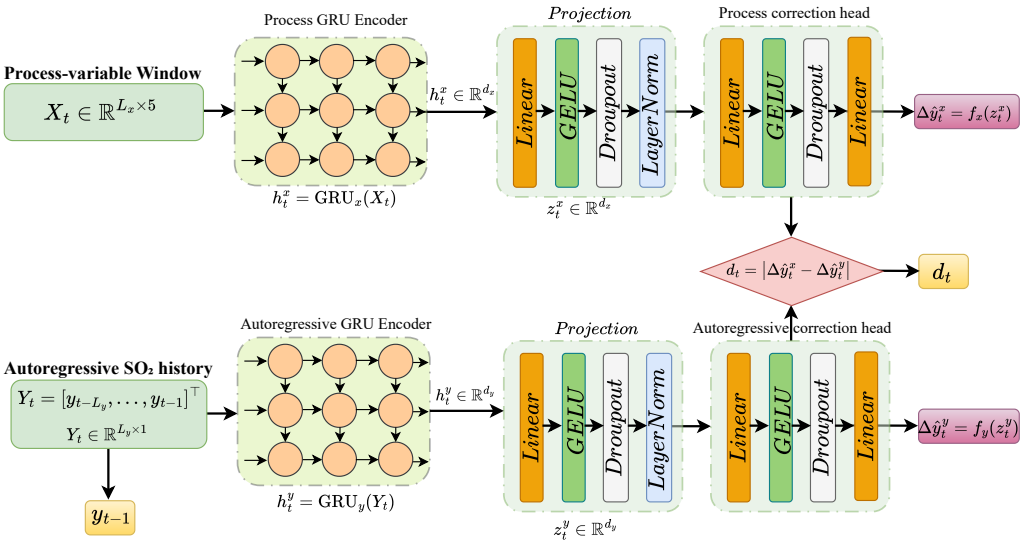


Figure 3. Dual-branch temporal encoding and residual-correction module of GRAG-Net. The process branch encodes the multivariate operating history, whereas the autoregressive branch encodes the strictly causal SO_2 history. Each branch produces a dedicated residual correction relative to $\widetilde{y}_{t-1}$, and the absolute difference between the two corrections defines the branch-disagreement statistic d_t.

2.3.2. GRU-Based Temporal Encoding

A gated recurrent unit (GRU) was selected for both temporal branches because it provides explicit memory-control mechanisms with fewer trainable gates than an LSTM. For an input vector $\mathbf{u}_\tau$ and previous hidden state $\mathbf{h}_{\tau-1}$, a GRU computes

$$\mathbf{r}_\tau = \sigma(\mathbf{W}_r \mathbf{u}_\tau + \mathbf{U}_r \mathbf{h}_{\tau-1} + \mathbf{b}_r), \tag{25}$$

$$\mathbf{q}_\tau = \sigma\big(\mathbf{W}_q \mathbf{u}_\tau + \mathbf{U}_q \mathbf{h}_{\tau-1} + \mathbf{b}_q\big), \tag{26}$$

$$\widetilde{\mathbf{h}}_\tau = \tanh[\mathbf{W}_h \mathbf{u}_\tau + \mathbf{U}_h(\mathbf{r}_\tau \odot \mathbf{h}_{\tau-1}) + \mathbf{b}_h], \tag{27}$$

$$\mathbf{h}_\tau = (1 - \mathbf{q}_\tau) \odot \widetilde{\mathbf{h}}_\tau + \mathbf{q}_\tau \odot \mathbf{h}_{\tau-1}, \tag{28}$$

where $\mathbf{r}_\tau$ and $\mathbf{q}_\tau$ are the reset and update gates, $\sigma(\cdot)$ denotes the logistic sigmoid, and $\odot$ represents element-wise multiplication. Separate parameter sets are learned for the process and autoregressive GRUs; no recurrent weights are shared between the two branches.

2.3.3. Process-Variable Expert

The process expert receives the standardized sequence $\widetilde{\mathbf{X}}_t$ and models nonlinear temporal interactions among the five process measurements. Its final GRU state is

$$\mathbf{h}_t^x = \mathrm{GRU}_x\left(\widetilde{\mathbf{X}}_t\right) \in \mathbb{R}^{d_x}, \tag{29}$$

where d_x is the process-branch hidden dimension. The final state is passed through a projection block consisting of a linear transformation, Gaussian error linear unit (GELU), dropout, and layer normalization:

$$\mathbf{z}_t^x = \mathrm{LN}\Big\{\mathrm{Drop}_p\Big[\mathrm{GELU}\Big(\mathbf{W}_p^x \mathbf{h}_t^x + \mathbf{b}_p^x\Big)\Big]\Big\} \in \mathbb{R}^{d_x}. \tag{30}$$

Layer normalization stabilizes the latent scale before fusion, whereas dropout regularizes the projected representation.

A two-layer correction head maps $\mathbf{z}_t^x$ to a scalar process-driven residual:

$$\mathbf{a}_t^x = \mathrm{Drop}_p[\mathrm{GELU}(\mathbf{W}_{x,1}\mathbf{z}_t^x + \mathbf{b}_{x,1})], \tag{31}$$

$$\Delta\widehat{\widetilde{y}}_t^x = \mathbf{W}_{x,2}\mathbf{a}_t^x + b_{x,2}. \tag{32}$$

The implemented head follows

$$\mathrm{Linear}(d_x, d_x/2) \to \mathrm{GELU} \to \mathrm{Dropout}(p) \to \mathrm{Linear}(d_x/2, 1). \tag{33}$$

The resulting branch-level estimate is

$$\widehat{\widetilde{y}}_t^x = \widetilde{y}_{t-1} + \Delta\widehat{\widetilde{y}}_t^x. \tag{34}$$

Accordingly, the process expert learns the correction implied by the recent evolution of the airflow-related measurements.

2.3.4. Autoregressive Expert

The autoregressive expert independently encodes the strictly causal target history. Its final hidden state is

$$\mathbf{h}_t^y = \mathrm{GRU}_y\left(\widetilde{\mathbf{Y}}_t\right) \in \mathbb{R}^{d_y}, \tag{35}$$

where d_y is selected independently from d_x. The projected autoregressive representation is

$$\mathbf{z}_t^y = \mathrm{LN}\left\{\mathrm{Drop}_p\left[\mathrm{GELU}\left(\mathbf{W}_p^y\mathbf{h}_t^y + \mathbf{b}_p^y\right)\right]\right\} \in \mathbb{R}^{d_y}. \tag{36}$$

Its correction head is defined as

$$\mathbf{a}_t^y = \mathrm{Drop}_p\left[\mathrm{GELU}\left(\mathbf{W}_{y,1}\mathbf{z}_t^y + \mathbf{b}_{y,1}\right)\right], \tag{37}$$

$$\Delta\widehat{\widetilde{y}}_t^y = \mathbf{W}_{y,2}\mathbf{a}_t^y + b_{y,2}, \tag{38}$$

with layer sequence

$$\mathrm{Linear}(d_y, d_y/2) \to \mathrm{GELU} \to \mathrm{Dropout}(p) \to \mathrm{Linear}(d_y/2, 1). \tag{39}$$

The corresponding branch-level estimate is

$$\widehat{\widetilde{y}}_t^y = \widetilde{y}_{t-1} + \Delta\widehat{\widetilde{y}}_t^y. \tag{40}$$

This branch provides a history-driven correction that exploits the shape, direction, and persistence of the recent SO_2 trajectory rather than using only the single value $\widetilde{y}_{t-1}$.

2.3.5. Dynamic Context Construction

The reliability gate receives the learned representations of both experts together with two explicit dynamic descriptors. The first descriptor is the absolute disagreement between the proposed branch corrections:

$$d_t = \left|\Delta\widehat{\widetilde{y}}_t^x - \Delta\widehat{\widetilde{y}}_t^y\right|. \tag{41}$$

A small value of d_t indicates that the two experts recommend a similar departure from the persistence reference, whereas a large value indicates competition between their process-driven and history-driven explanations.

The second descriptor summarizes the most recent change in the standardized process measurements:

$$s_t = \frac{1}{5}\sum_{j=1}^{5}\left|\widetilde{x}_{j,t} - \widetilde{x}_{j,t-1}\right|. \tag{42}$$

This statistic supplies the gate with a compact indicator of the local process dynamics. It is not used as a manually defined regime label and does not directly determine the gate value; instead, it is learned jointly with the latent branch representations.

The complete context vector is

$$\mathbf{c}_t = \left[(\mathbf{z}_t^x)^\top, \left(\mathbf{z}_t^y\right)^\top, \widetilde{y}_{t-1}, d_t, s_t\right]^\top \in \mathbb{R}^{d_x+d_y+3}. \tag{43}$$

Including $\widetilde{y}_{t-1}$ allows the gating decision to depend on the current target level, while d_t and s_t expose the instantaneous degree of branch disagreement and process variation.

2.3.6. Sample-Dependent Reliability Gate and Residual Fusion

The context vector is mapped to a scalar process-branch weight using a three-layer multilayer perceptron:

$$\mathbf{v}_{t,1} = \mathrm{Drop}_p\left[\mathrm{GELU}\left(\mathbf{W}_{g,1}\mathbf{c}_t + \mathbf{b}_{g,1}\right)\right] \in \mathbb{R}^{d_g}, \tag{44}$$

$$\mathbf{v}_{t,2} = \mathrm{Drop}_p\left[\mathrm{GELU}\left(\mathbf{W}_{g,2}\mathbf{v}_{t,1} + \mathbf{b}_{g,2}\right)\right] \in \mathbb{R}^{d_g/2}, \tag{45}$$

$$g_t = \sigma\left(\mathbf{W}_{g,3}\mathbf{v}_{t,2} + b_{g,3}\right). \tag{46}$$

The implemented sequence is therefore

$$\begin{aligned}\mathrm{Linear}(d_x + d_y + 3, d_g) &\to \mathrm{GELU} \to \mathrm{Dropout}(p)\\ &\to \mathrm{Linear}(d_g, d_g/2) \to \mathrm{GELU} \to \mathrm{Dropout}(p)\\ &\to \mathrm{Linear}(d_g/2, 1) \to \mathrm{Sigmoid}\,.\end{aligned} \tag{47}$$

The gate performs soft routing rather than hard branch selection. The fused residual is

$$\Delta\widehat{\widetilde{y}}_t = g_t\Delta\widehat{\widetilde{y}}_t^{\,x} + (1 - g_t)\Delta\widehat{\widetilde{y}}_t^{\,y}, \tag{48}$$

and the final standardized estimate is reconstructed as

$$\widehat{\widetilde{y}}_t = \widetilde{y}_{t-1} + g_t\Delta\widehat{\widetilde{y}}_t^{\,x} + (1 - g_t)\Delta\widehat{\widetilde{y}}_t^{\,y}. \tag{49}$$

Because g_t is computed independently for each sample, the relative branch contribution can vary throughout the operating sequence. The prediction on the original benchmark scale is obtained using the inverse transformation in Equation (15).

2.3.7. Joint Training Objective

A robust Huber loss is used because the target distribution contains a narrow central region together with infrequent large deviations. For a scalar residual e and transition parameter δ, the loss is

$$\ell_\delta(e) = \begin{cases}\dfrac{1}{2}e^2, & |e| \le \delta,\\ \delta\left(|e| - \dfrac{1}{2}\delta\right), & |e| > \delta.\end{cases} \tag{50}$$

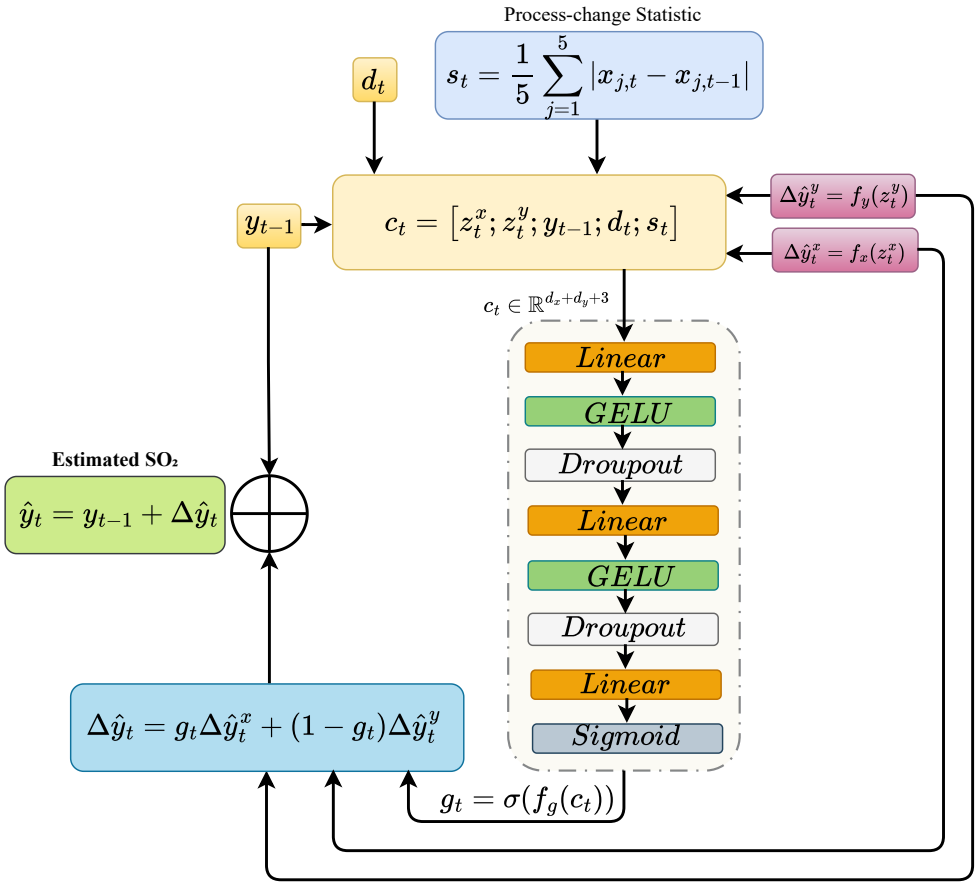


Figure 4. Context-dependent reliability gating and residual fusion in GRAG-Net. The projected process and autoregressive states, latest SO_2 observation, branch disagreement, and recent process-change statistic form the gating context. The resulting process weight g_t adaptively combines the two branch-specific residual corrections before addition to the persistence reference.

The transition value was fixed at $\delta = 0.05$ in the standardized target domain. For a mini-batch of size B, the principal prediction loss is

$$\mathcal{L}_{\text{pred}} = \frac{1}{B}\sum_{i=1}^{B}\ell_\delta\left(\widehat{\widetilde{y}}_i - \widetilde{y}_i\right). \tag{51}$$

Optimizing only the fused output can permit one expert to become weak while the gate collapses toward the other. To preserve two individually predictive experts, the branch-level estimates in Equations (34) and (40) are supervised through

$$\mathcal{L}_{\text{branch}} = \frac{1}{2B}\sum_{i=1}^{B}\left[\ell_\delta\left(\widehat{\widetilde{y}}_i^{\,x} - \widetilde{y}_i\right) + \ell_\delta\left(\widehat{\widetilde{y}}_i^{\,y} - \widetilde{y}_i\right)\right]. \tag{52}$$

This auxiliary objective encourages both branches to remain informative before fusion; it does not force their residual corrections to be identical.

A weak gate-balance term can additionally discourage premature batch-level collapse toward a single branch:

$$\mathcal{L}_{\text{gate}} = \left(\frac{1}{B}\sum_{i=1}^{B} g_i - 0.5\right)^2. \tag{53}$$

The regularizer constrains only the average gate value within a mini-batch and does not require every sample to receive equal process and autoregressive weights. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_b \mathcal{L}_{\text{branch}} + \lambda_g \mathcal{L}_{\text{gate}}, \tag{54}$$

where λ_b and λ_g are selected within the inner hyperparameter-optimization loop. When $\lambda_g = 0$, routing is governed entirely by the predictive and branch-supervision objectives.

2.3.8. Training-Time Autoregressive Perturbation

To reduce excessive dependence on a perfectly noise-free target history, zero-mean Gaussian perturbation is considered as a training-only regularizer:

$$\widetilde{\mathbf{Y}}'_t = \widetilde{\mathbf{Y}}_t + \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\text{AR}}^2 \mathbf{I}\right). \tag{55}$$

When perturbation is active, the residual reference provided to the model is the final perturbed history value,

$$\widetilde{y}'_{t-1} = \widetilde{\mathbf{Y}}'_t[-1], \tag{56}$$

so that the autoregressive sequence and skip connection remain internally consistent. The process history and regression target remain unperturbed. No noise is applied during validation or outer-test evaluation. The standard deviation σ_{AR} is selected through the inner optimization procedure.

2.3.9. Architecture Summary

Table 2 summarizes the implemented modules and tensor dimensions. GRAG-Net remains compact because each information source is processed by one recurrent layer and the fusion mechanism consists of a small multilayer perceptron.

Table 2. Implemented GRAG-Net modules, layer configurations, and output dimensions.

Module	Layer Sequence or Operation	Output Dimension
Process encoder	Single-layer GRU with input size 5 and hidden size d_x	$\mathbf{h}_t^x \in \mathbb{R}^{d_x}$
Process projection	Linear(d_x, d_x)–GELU–Dropout(p)–LayerNorm	$\mathbf{z}_t^x \in \mathbb{R}^{d_x}$
Process correction head	Linear$(d_x, d_x/2)$–GELU–Dropout(p)–Linear$(d_x/2, 1)$	$\Delta\widehat{\widetilde{y}}_t^x \in \mathbb{R}$
Autoregressive encoder	Single-layer GRU with input size 1 and hidden size d_y	$\mathbf{h}_t^y \in \mathbb{R}^{d_y}$
Autoregressive projection	Linear(d_y, d_y)–GELU–Dropout(p)–LayerNorm	$\mathbf{z}_t^y \in \mathbb{R}^{d_y}$
Autoregressive correction head	Linear$(d_y, d_y/2)$–GELU–Dropout(p)–Linear$(d_y/2, 1)$	$\Delta\widehat{\widetilde{y}}_t^y \in \mathbb{R}$
Gate context	$\mathbf{c}_t = \left[(\mathbf{z}_t^x)^\top, (\mathbf{z}_t^y)^\top, \widetilde{y}_{t-1}, d_t, s_t\right]^\top$	$\mathbf{c}_t \in \mathbb{R}^{d_x+d_y+3}$
Reliability gate	Linear$(d_x + d_y + 3, d_g)$–GELU–Dropout(p) –Linear$(d_g, d_g/2)$–GELU–Dropout(p) –Linear$(d_g/2, 1)$–Sigmoid	$g_t \in [0, 1]$
Residual reconstruction	$\widehat{\widetilde{y}}_t = \widetilde{y}_{t-1} + g_t \Delta\widehat{\widetilde{y}}_t^x + (1 - g_t)\Delta\widehat{\widetilde{y}}_t^y$	$\widehat{\widetilde{y}}_t \in \mathbb{R}$

2.4. Comparator Models and Formal Ablations

The revised evaluation was designed to separate three questions: whether target persistence alone is sufficient, whether conventional or sequence-based models can exploit the same causal information, and whether the proposed residual dual-branch gate provides value beyond simpler fusion rules. All learnable models were fitted using only information available within the corresponding outer-training period.

Two parameter-free references were included. Persistence uses $\widehat{y}_t = y_{t-1}$, while linear extrapolation extends the most recent target increment, $\widehat{y}_t = 2y_{t-1} - y_{t-2}$. Four classical regressors—SVR [23], Random Forest [24], Extra Trees [25], and XGBoost [26]—were evaluated in both absolute-target ARX form and residual form. The residual variants predict $\Delta y_t = y_t - y_{t-1}$ and reconstruct the target by adding the predicted increment to

y_{t-1}; this control is important because it separates the effect of residual learning from the recurrent architecture.

The sequence-model controls comprised a strict process-only GRU, an AR–GRU using only causal target history, a joint GRU receiving combined process and target history information, LSTM–ARX, and TCN–ARX. The process-only GRU receives neither target history samples nor the y_{t-1} residual anchor. Two fusion controls use the same dual-branch information as GRAG-Net: equal fusion fixes the branch weight at 0.5, whereas the global-gate model learns one scalar fusion weight shared by all samples. These controls distinguish the value of sample-dependent routing from the mere availability of two branches.

Formal ablations were independently trained for the full GRAG-Net, removal of the residual reconstruction, equal fusion, global gating, removal of branch supervision ($\lambda_b = 0$), removal of gate-balance regularization ($\lambda_g = 0$), removal of branch disagreement d_t, and removal of the process-change statistic s_t. AR-history perturbation is not interpreted as an active component of the selected final model because hyperparameter selection returned $\sigma_{\mathrm{AR}} = 0$ in all three outer folds.

2.5. Chronological Model Development and Reproducibility

2.5.1. Nested Chronological Design

Random partitioning was avoided because neighboring SRU samples are strongly autocorrelated and time-ordered evaluation is required to avoid optimistic performance estimates in autoregressive settings [27]. Three expanding outer-training periods were therefore followed by non-overlapping future evaluation periods, as summarized in Table 3. The outer-test targets were excluded from model fitting, preprocessing, hyperparameter selection, and epoch selection. During test-time rolling one-step evaluation, however, an SO_2 observation becomes admissible historical information after its time has passed; thus a prediction at t may use measured target history only through $t - 1$. Predicted targets are never recursively fed back.

Table 3. Chronological outer folds used for final evaluation. Intervals are half-open index ranges.

Fold	Training Interval	Training Size	Test Interval	Test Size
1	[0, 5544)	5544	[5544, 7056)	1512
2	[0, 7056)	7056	[7056, 8568)	1512
3	[0, 8568)	8568	[8568, 10,081)	1513

Figure 5 summarizes the nested workflow. Within each outer-training period, the earliest 80% of observations formed the inner-training subset and the latest 20% formed the inner-validation subset. Scaling parameters were estimated only from the active training subset. Optuna [28] with a seeded tree-structured Parzen estimator (2026) was used for configuration selection. Final neural models were evaluated with seeds $\{42, 52, 62, 72, 82\}$ after all model-selection decisions had been fixed.

2.5.2. Hyperparameter Search Spaces and Selected Configuration

To make the revised comparison reproducible, the search domains used for the reviewer-requested models are reported explicitly in Table 4. Search was performed independently inside each outer-training period using the chronological inner split described above; the outer-test block was never used to choose a configuration. Classical models used 18 Optuna trials per fold and neural models used 24 trials per fold. The final reviewer run reused the winning configurations saved from this model development stage, without test-set retuning.

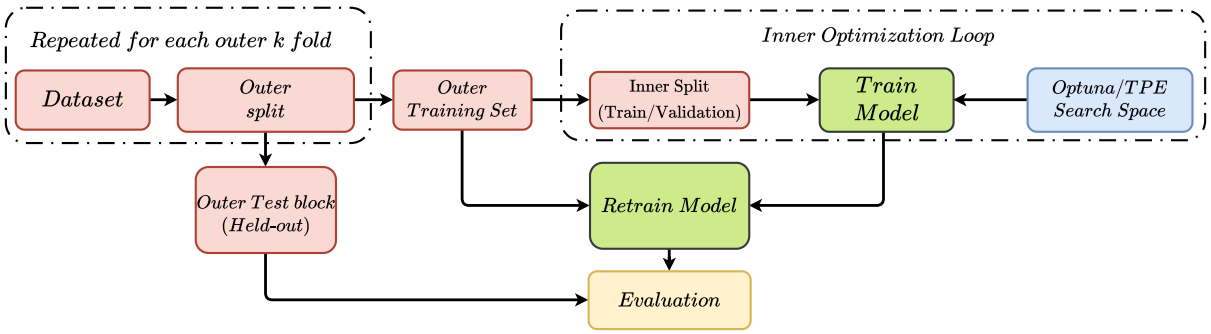


Figure 5. Leakage-aware nested chronological model development. Hyperparameters, preprocessing parameters, and epoch-selection decisions are determined using only the outer-training period. The subsequent future block is held out from model development and is evaluated under causal rolling one-step inference, where measured SO_2 history is available only through the preceding time instant.

Table 4. Hyperparameter search spaces used in the nested model development stage. Parameters not listed were fixed by the implementation. Structural ablations used the same numerical search domain while forcing the ablated component to its prescribed setting.

Model Family	Hyperparameter(s)	Search Domain
All learnable models	L_x, L_y	$\{16, 32, 64\}$; $\{4, 8, 16, 32\}$
SVR–ARX/residual SVR	C, ε, γ	$[10^{-1}, 10^{3}]$ (log); $[10^{-4}, 5 \times 10^{-2}]$ (log); $[10^{-4}, 10^{-1}]$ (log)
RF/Extra Trees (absolute and residual)	n_{trees}, depth, min. leaf, max. features	$\{300, 500, 700\}$; $\{\text{None}, 10, 20, 30\}$; $\{1, \ldots, 5\}$; $\{0.50, 0.75, 1.00\}$
XGBoost (absolute and residual)	n_{trees}, depth, learning rate, subsample, column subsample, min. child weight, λ	$\{200, 300, \ldots, 800\}$; $\{2, \ldots, 8\}$; $[0.01, 0.15]$ (log); $[0.70, 1]$; $[0.70, 1]$; $[1, 10]$; $[10^{-3}, 10]$ (log)
GRU/LSTM controls	h, p, learning rate, weight decay, batch size, σ_{AR}	$\{32, 64, 96\}$; $\{0, 0.1, 0.2\}$; $\{3, 5, 10\} \times 10^{-4}$; $\{10^{-5}, 10^{-4}, 10^{-3}\}$; $\{64, 128, 256\}$; $\{0, 0.01, 0.02\}$
TCN–ARX	channels h, p, learning rate, weight decay, batch size	$\{32, 64, 96\}$; $\{0, 0.1, 0.2\}$; $\{3, 5, 10\} \times 10^{-4}$; $\{10^{-5}, 10^{-4}, 10^{-3}\}$; $\{64, 128, 256\}$
Dual-branch fusion/GRAG-Net	d_x, d_y, d_g, p, learning rate, weight decay, batch size, λ_b, λ_g, σ_{AR}	$\{32, 64, 96\}$; $\{16, 32, 64\}$; $\{32, 64, 96\}$; $\{0, 0.1, 0.2\}$; $\{3, 5, 10\} \times 10^{-4}$; $\{10^{-5}, 10^{-4}, 10^{-3}\}$; $\{64, 128, 256\}$; $\{0, 0.025, 0.05, 0.10\}$; $\{0, 10^{-4}, 10^{-3}\}$; $\{0, 0.01, 0.02\}$

The full GRAG-Net configuration selected independently in each outer fold was identical across all three folds, as summarized in Table 5. This agreement emerged from independent inner-fold selection; the configuration was not imposed globally. In particular, $\sigma_{\text{AR}} = 0$ was selected in every fold, so autoregressive history perturbation is not treated as an active component of the final model. Exact fold-specific winning configurations for every comparator and formal ablation, together with the corresponding Optuna trial histories, were retained in the reproducibility outputs. These records are available with the revision materials and permit reconstruction of every reported model-selection decision.

After hyperparameter selection, the final training length for each neural model, outer fold, and seed was determined without using the outer-test block. Specifically, the earliest 85% of the corresponding outer-training period was used for fitting and the latest 15% for epoch selection, with a maximum of 150 epochs and early-stopping patience of 15. The selected epoch was then used to retrain the model from a new initialization on the complete outer-training period before evaluation. Thus, the temporary epoch-selection subset was returned to the final training set and no test observation was used to determine training duration.

The final revision was executed under Windows 11 using Python 3.13.9, NumPy 2.3.5, pandas 2.3.3, PyTorch 2.11.0 with CUDA 12.6, scikit-learn 1.7.2, XGBoost 3.2.0, and SHAP

0.51.0 on an NVIDIA T1000 8 GB GPU. The exact chronological splits, selected configurations, fitted preprocessing parameters, epoch-selection histories, model checkpoints, fold–seed predictions, metrics, and diagnostic outputs were retained as a reproducibility audit trail.

Table 5. GRAG-Net configuration selected independently in all three chronological outer folds.

Hyperparameter/Setting	Selected Value
Process-history length L_x	64
Target history length L_y	16
Process hidden dimension d_x	96
Autoregressive hidden dimension d_y	64
Gate hidden dimension d_g	64
Dropout probability p	0
Learning rate	5×10^{-4}
Weight decay	10^{-4}
Batch size	64
Branch-loss weight λ_b	0.10
Gate-balance weight λ_g	10^{-4}
AR-noise standard deviation σ_{AR}	0
Gate type	Dynamic
Residual reconstruction	Enabled
Branch-disagreement input d_t	Enabled
Process-change input s_t	Enabled

2.6. Evaluation, Statistical Analysis, and Diagnostics

All predictions were inverse transformed before evaluation, and SO_2 errors are reported in mole%. The primary predictive measures are MAE, RMSE, and R^2; sMAPE, median absolute error, explained variance, mean bias deviation, and Pearson correlation are retained as secondary diagnostics. Because the five random seeds within one chronological fold share the same test period, the three outer folds are treated as the primary independent evaluation periods; seed dispersion describes optimization variability rather than five additional independent datasets.

For fold k, the main error measures are

$$\mathrm{MAE}^{(k)} = \frac{1}{n_k} \sum_{i=1}^{n_k} |y_i - \widehat{y}_i|, \qquad \mathrm{RMSE}^{(k)} = \sqrt{\frac{1}{n_k} \sum_{i=1}^{n_k} (y_i - \widehat{y}_i)^2}. \tag{57}$$

The pooled seed-ensemble analysis averages the five predictions for each unique out-of-fold sample before computing metrics over the 4537 non-overlapping test observations. It is reported as a complementary aggregate, not as a replacement for fold-wise evidence.

Taylor diagrams are reported separately for the three outer folds. Each model is located by its Pearson correlation and predicted standard deviation, while centered RMSE contours summarize joint discrepancies in phase and variability. The ideal point has correlation one, predicted standard deviation equal to the observed standard deviation, and zero centered RMSE [29].

To quantify uncertainty while respecting serial dependence, paired moving-block bootstrap analysis was applied to sample-wise loss differences

$$\Delta_t = \ell(y_t, \widehat{y}_{\mathrm{GRAG},t}) - \ell(y_t, \widehat{y}_{B,t}), \tag{58}$$

using absolute and squared error losses. A block length of 30 samples is the primary analysis, with 15 and 60 samples retained as sensitivity checks. Negative Δ favors GRAG-

Net. Confidence intervals are computed within each chronological test period rather than pooling folds as if they were exchangeable independent replicates.

The diagnostic analysis was deliberately separated into inference-available and retrospective quantities. Gate behavior was summarized by Spearman association between the process-branch weight g_t and three quantities available at inference: branch disagreement d_t, process change s_t, and the latest measured target y_{t-1}. Any association with realized branch advantage is treated only as retrospective interpretation because it depends on y_t.

Temporal alignment was examined using training-only lagged Pearson cross-correlations between each process variable and SO_2. For each fold, the reported lag is the training-period lag that maximizes $|r|$; these correlations are descriptive and are not interpreted as causal delay estimates. Error reliability was assessed using the empirical CDF and the 50th, 75th, 90th, 95th, and 99th percentiles of absolute error. A retrospective target-change analysis partitions test samples by $|y_t - y_{t-1}|$: the lower 50% form the stable regime, the 50th–90th percentiles form the changing regime, and the upper 10% form the large-change regime. This analysis tests whether conclusions depend primarily on quasi-steady observations.

SHAP was computed for two predefined seeds (42 and 52) in each of the three folds. Global importance is based on mean absolute SHAP value aggregated across these six fitted artifacts. Explanation stability is quantified by pairwise Spearman correlation of feature ranks and top-20 Jaccard overlap. SHAP values describe the fitted model and should not be interpreted as causal process effects [30,31].

The public benchmark and the originating SRU literature report SO_2 concentration in mole%. The released benchmark does not document the make, calibration uncertainty, accuracy specification, or regulatory tolerance of the plant analyzer used to obtain the reference SO_2 measurements. Because these instrument-specific quantities cannot be recovered reliably from the available data, they are not inferred or approximated here. Accordingly, the study reports prediction errors in the available physical concentration unit but does not compare them with an unspecified analyzer-accuracy or compliance threshold.

3. Results and Discussion

The revised experiments assess GRAG-Net under the causal rolling one-step setting defined in Section 2.2. The three non-overlapping chronological test periods are treated as the primary evaluation periods. Five final seeds quantify optimization variability within each fold, while pooled seed-ensemble predictions are reported only as a complementary sample-level summary over the 4537 unique out-of-fold observations.

3.1. Comparative Predictive Performance

Table 6 reports descriptive mean ± standard deviation over the 15 fold–seed evaluations for each model. These values summarize both chronological and initialization variability; they are not interpreted as 15 independent datasets because the five seeds within a fold share the same test observations. GRAG-Net achieved the strongest primary performance, with MAE 0.00315 ± 0.00044 mole%, RMSE 0.00790 ± 0.00064 mole%, and $R^2 = 0.97753 \pm 0.00331$. It also obtained the highest Pearson correlation (0.98898 ± 0.00156) and the lowest sMAPE (2.258 ± 0.462%) among the learned full-model comparators.

The strongest sequence comparator was LSTM–ARX in RMSE and R^2, whereas linear extrapolation produced the second-lowest MAE. Nevertheless, neither matched GRAG-Net simultaneously in absolute error, squared-error sensitivity, variance explanation, and correlation. The dedicated AR–GRU was also substantially weaker, with MAE 0.00453 ± 0.00052 and RMSE 0.00979 ± 0.00086 mole%. Thus, the gain of GRAG-Net cannot be attributed merely to access to recent target history; the exogenous branch and adaptive residual fusion provide additional predictive information beyond a strong autoregressive recurrent model.

Table 6. Predictive performance across the three chronological outer folds and five final seeds. Values are descriptive mean ± standard deviation over 15 fold–seed evaluations per model. The three chronological folds, not the 15 repeated runs, are the primary independent evaluation periods; formal paired uncertainty is assessed separately by moving-block bootstrap. The best value is shown in bold and the second-best value is underlined. For MBD, ranking is based on absolute magnitude.

Model	MAE	RMSE	R^2	sMAPE (%)	MedAE	MBD ($\times 10^{-3}$)	Pearson R
Persistence	0.00650 ± 0.00120	0.01384 ± 0.00123	0.93164 ± 0.00171	4.219 ± 1.035	0.002925 ± 0.000721	0.002 ± 0.074	0.96581 ± 0.00088
Linear extrapolation	0.00351 ± 0.00073	0.01221 ± 0.00064	0.94636 ± 0.00608	2.582 ± 0.670	**0.00000001 ± 0.00000000**	**−0.00046 ± 0.00691**	0.97690 ± 0.00271
SVR–ARX	0.00426 ± 0.00082	0.01028 ± 0.00069	0.96209 ± 0.00306	3.234 ± 0.731	0.001481 ± 0.000442	0.175 ± 0.250	0.98136 ± 0.00178
Random Forest–ARX	0.03303 ± 0.00865	0.04466 ± 0.00835	0.24782 ± 0.33249	21.721 ± 6.979	0.024828 ± 0.009408	−9.354 ± 14.575	0.64258 ± 0.17946
Extra Trees–ARX	0.03862 ± 0.00547	0.05036 ± 0.00583	0.07906 ± 0.22073	24.658 ± 4.844	0.030795 ± 0.005488	−7.167 ± 13.422	0.52925 ± 0.19886
XGBoost–ARX	0.01673 ± 0.00684	0.02558 ± 0.00949	0.75924 ± 0.13539	11.233 ± 4.387	0.010690 ± 0.004234	0.144 ± 1.807	0.92815 ± 0.02049
Residual SVR	0.00396 ± 0.00086	0.01060 ± 0.00059	0.95945 ± 0.00638	2.863 ± 0.786	0.001126 ± 0.000403	−0.003 ± 0.324	0.98075 ± 0.00255
Residual RF	0.00771 ± 0.00167	0.01458 ± 0.00180	0.92173 ± 0.02439	5.889 ± 2.441	0.003865 ± 0.001026	−1.971 ± 3.542	0.96391 ± 0.00714
Residual ET	0.00718 ± 0.00097	0.01399 ± 0.00091	0.92972 ± 0.00666	4.850 ± 0.981	0.003897 ± 0.001175	−1.413 ± 2.479	0.96624 ± 0.00117
Residual XGBoost	0.00729 ± 0.00117	0.01473 ± 0.00172	0.92013 ± 0.02416	4.908 ± 1.110	0.003267 ± 0.000470	−1.244 ± 2.077	0.96041 ± 0.01096
Process-only GRU	0.03460 ± 0.01244	0.04377 ± 0.01536	0.20871 ± 0.56710	23.428 ± 10.797	0.030827 ± 0.013189	−6.764 ± 28.431	0.72142 ± 0.18620
AR–GRU	0.00453 ± 0.00052	0.00979 ± 0.00086	0.96555 ± 0.00484	3.133 ± 0.528	0.001856 ± 0.000343	0.089 ± 0.476	0.98290 ± 0.00229
Joint GRU	0.00609 ± 0.00043	0.01279 ± 0.00089	0.93960 ± 0.01466	4.003 ± 0.409	0.002707 ± 0.000479	−0.220 ± 1.280	0.97004 ± 0.00712
LSTM–ARX	0.00391 ± 0.00075	0.00889 ± 0.00097	0.97123 ± 0.00720	2.785 ± 0.676	0.001672 ± 0.000750	0.706 ± 1.027	0.98642 ± 0.00282
TCN–ARX	0.00563 ± 0.00131	0.01235 ± 0.00187	0.94489 ± 0.01330	3.742 ± 1.029	0.002436 ± 0.000695	0.104 ± 0.307	0.97235 ± 0.00701
Equal fusion	0.00408 ± 0.00084	0.00941 ± 0.00077	0.96815 ± 0.00445	2.958 ± 0.840	0.001669 ± 0.001331	−0.793 ± 1.676	0.98473 ± 0.00172
Global gate	0.00390 ± 0.00068	0.00908 ± 0.00116	0.97029 ± 0.00602	2.827 ± 0.724	0.001501 ± 0.000962	−0.858 ± 1.197	0.98557 ± 0.00290
GRAG-Net	**0.00315 ± 0.00044**	**0.00790 ± 0.00064**	**0.97753 ± 0.00331**	**2.258 ± 0.462**	0.000907 ± 0.000257	−0.034 ± 0.621	**0.98898 ± 0.00156**

MSE and explained variance were also computed and retained in the reproducibility outputs; they are omitted from the main table because they are largely redundant with RMSE and R^2. The residual classical controls additionally show that residualization alone is insufficient: residual SVR is the strongest classical residual model, but its MAE and RMSE remain above those of GRAG-Net. Likewise, equal fusion and global gating improve substantially over the strict process-only model but remain behind the dynamic gate, supporting the value of sample-dependent fusion rather than simply adding a second branch.

The pooled seed-ensemble analysis leads to the same conclusion. GRAG-Net attained MAE 0.00291 mole%, RMSE 0.00766 mole%, and $R^2 = 0.97937$. Relative to persistence, these correspond to MAE and RMSE reductions of 55.2% and 44.8%; relative to AR–GRU, the reductions are 32.5% and 18.0%. Because pooling averages the five predictions for each unique sample, these values are complementary ensemble statistics and are not substituted for the fold-level evidence below.

3.2. Chronological Robustness and Temporal Tracking

Table 7 reports fold-wise mean performance over the five final seeds for the principal sequence and fusion models. GRAG-Net has the lowest MAE and RMSE in all three chronological periods. Its MAE ranges from 0.00285 to 0.00372 mole%, while R^2 remains between 0.97344 and 0.97980. The third fold is the most difficult in absolute-error terms, yet the proposed model preserves high explained variance and maintains a clear margin over AR–GRU, LSTM–ARX, TCN–ARX, equal fusion, and global gating.

Figure 6 shows the seed-ensemble GRAG-Net trajectories. The ensemble MAE values are 0.002635, 0.002565, and 0.003534 mole% for Folds 1–3, respectively, with corresponding RMSE values of approximately 0.00695, 0.00769, and 0.00827 mole%. The predicted trajectories closely follow the observed concentration through both steady and rapidly varying intervals, with the largest deviations concentrated around abrupt excursions.

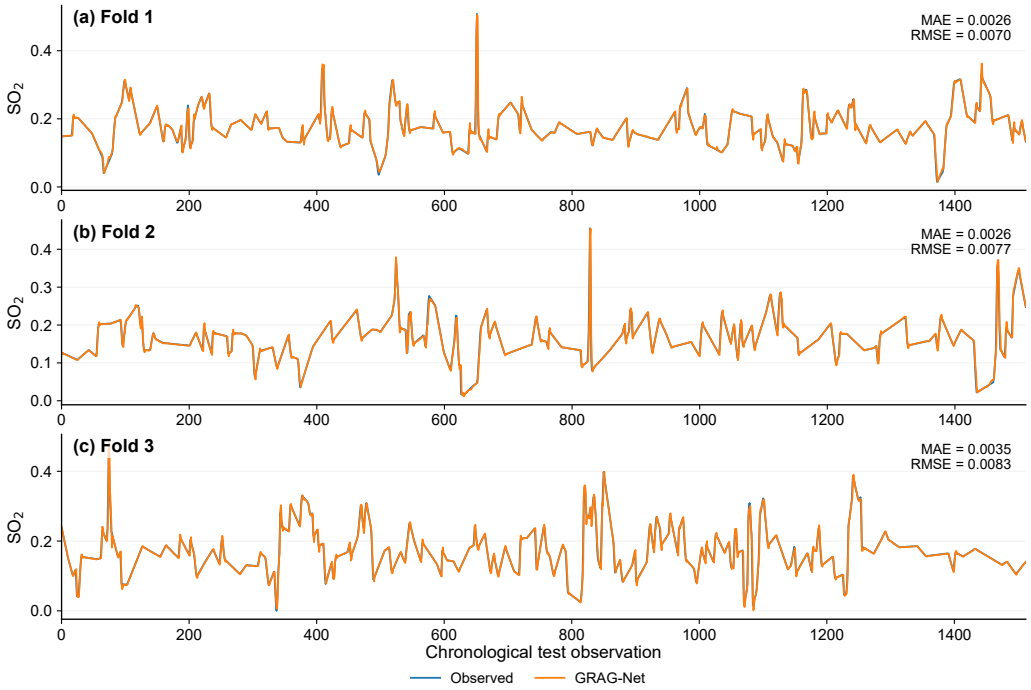


Figure 6. Observed and GRAG-Net-predicted SO_2 concentration over the three chronological outer-test periods. Predictions are seed-ensemble means; fold-specific MAE and RMSE are reported in each panel.

Table 7. Fold-level mean performance over five final seeds for the principal sequence and fusion models. Errors are in mole%.

Model	Fold 1 MAE/RMSE	Fold 2 MAE/RMSE	Fold 3 MAE/RMSE
Persistence	0.005999/0.013024	0.005622/0.013229	0.007867/0.015258
AR–GRU	0.004429/0.009054	0.004028/0.009738	0.005137/0.010575
LSTM–ARX	0.003099/0.007812	0.004444/0.009583	0.004193/0.009268
TCN–ARX	0.004208/0.010101	0.005681/0.013079	0.007007/0.013860
Equal fusion	0.003245/0.008557	0.004701/0.009559	0.004302/0.010111
Global gate	0.003175/0.008208	0.004230/0.008938	0.004297/0.010100
GRAG-Net	0.002853/0.007103	0.002867/0.008128	0.003724/0.008472

3.3. Taylor-Diagram Assessment

The fold-specific Taylor diagrams in Figure 7 provide a joint view of correlation, variance reproduction, and centered error. GRAG-Net attains correlations of 0.99035, 0.98829, and 0.99030 in Folds 1–3 and the lowest centered RMSE in every fold: 0.00695, 0.00769, and 0.00826 mole%, respectively. Its predicted standard deviations (0.05010, 0.05040, and 0.05948) closely match the corresponding observed values (0.04997, 0.04992, and 0.05898). These results indicate that the proposed model not only reduces pointwise error but also reproduces the amplitude and temporal variability of the observed SO_2 sequence more faithfully than the evaluated alternatives.

3.4. Formal Ablation of the Proposed Architecture

Figure 8 summarizes independently trained and re-optimized ablations. Relative to the full model's pooled MAE of 0.00291 mole%, fixed equal fusion increases MAE by 34.5%, a global gate by 27.7%, and removal of residual reconstruction by 28.5%. Removing branch supervision, disagreement, process-change context, or gate-balance regularization produces smaller pooled degradations of approximately 3–5.5%. The dominant performance losses therefore occur when the residual formulation or sample-dependent fusion mechanism is simplified, directly supporting the two central design choices of GRAG-Net.

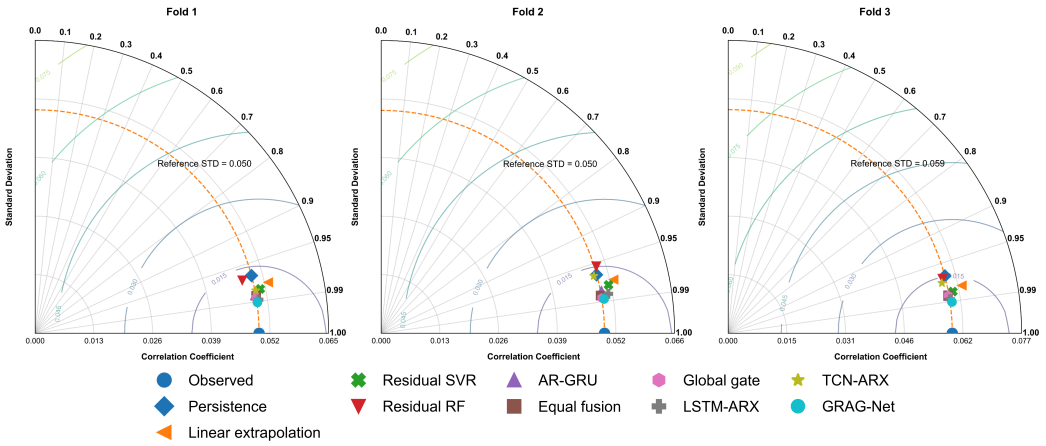


Figure 7. Taylor diagrams for the three chronological test folds. Radial distance denotes predicted standard deviation, angular position denotes Pearson correlation, and contours denote centered RMSE. The observation is the reference point.

The smaller components show more modest and fold-dependent effects and should be interpreted as regularizing or contextual refinements rather than as uniformly dominant mechanisms. In particular, the "no AR noise" row is a re-optimized control rather than evidence that perturbation itself is beneficial: the selected full-model configuration has $\sigma_{AR} = 0$, so autoregressive perturbation is inactive in the final GRAG-Net. This dis-

tinction prevents attributing a causal benefit to a component that was not used by the selected model.

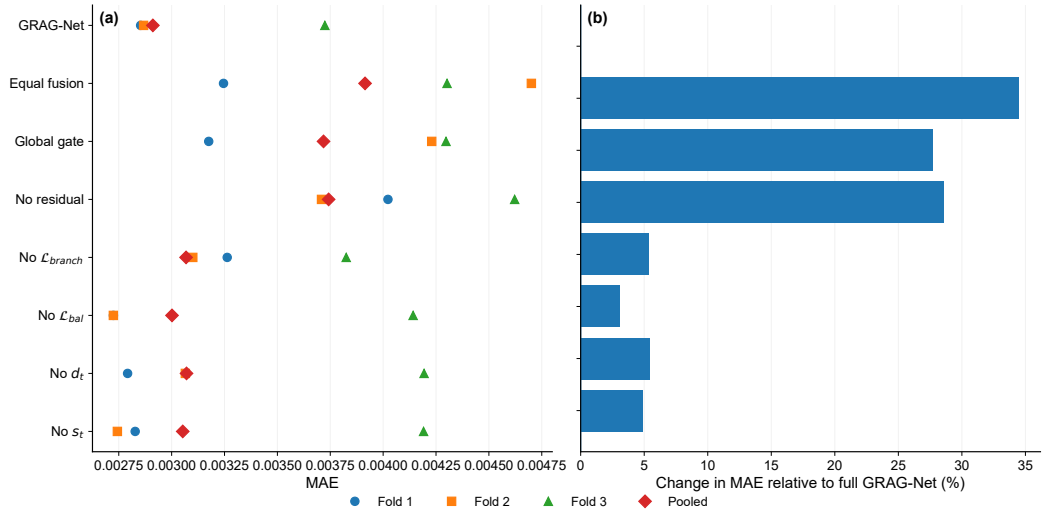


Figure 8. Formal independently trained ablations. (**a**) Fold-wise and pooled MAE. (**b**) Pooled percentage change in MAE relative to the full GRAG-Net. Positive values denote degradation relative to the full model.

3.5. Paired Moving-Block Bootstrap Evidence

The paired moving-block bootstrap provides dependence-aware uncertainty for sample-wise error differences within each chronological test period. Figure 9 reports the primary 30-sample block analysis for absolute error, where negative values of $|e_{\mathrm{GRAG}}| - |e_B|$ favor GRAG-Net. Against AR–GRU, the mean differences are -0.001595, -0.001329, and -0.001288 mole% in Folds 1–3, and all three 95% confidence intervals lie entirely below zero. GRAG-Net is likewise favored in all folds against LSTM–ARX, TCN–ARX, equal fusion, global gating, persistence, residual RF, and residual SVR. Linear extrapolation is inferior in Folds 1 and 3, while its Fold 2 interval slightly overlaps zero.

This analysis strengthens the descriptive metric comparison because it preserves temporal pairing and serial dependence rather than treating individual time samples as independent. The conclusion is also insensitive to the alternative block lengths of 15 and 60 samples retained in the sensitivity analysis: the principal comparisons remain directionally consistent. Thus, the advantage of GRAG-Net over the strongest recurrent controls is not explained solely by averaging across seeds or by a small number of isolated test points.

3.6. Gate Behavior and Training-Only Temporal Lags

The inference-aware gate analysis in Figure 10a shows that the process-branch weight is most strongly associated with branch disagreement ($\rho = 0.226$), whereas associations with the process-change statistic ($\rho = 0.053$) and the latest SO_2 level ($\rho = 0.064$) are weak. The gate should therefore be interpreted as a learned routing mechanism that reacts primarily to disagreement between the proposed process and autoregressive corrections, with weaker modulation by current process variation and target level. The retrospective association with realized process-branch advantage is only $\rho = 0.122$ and is not treated as causal evidence because it depends on the subsequently observed target.

Training-only lag analysis in Figure 10b identifies the clearest delayed linear relationships for x_1, x_2, and x_3, whose maximum absolute correlations generally occur about 6–24 min before the SO_2 response. The corresponding $|r|$ values are modest (approximately 0.13–0.21), indicating that no single process variable provides a dominant linear predictor.

The Fold 1 maximum for x_5 occurs at 114 min but has only $|r| \approx 0.049$; this weak boundary-like maximum is therefore not interpreted as a physically precise delay. The lag analysis supports the use of temporal process histories while reinforcing the need for a nonlinear multivariate representation.

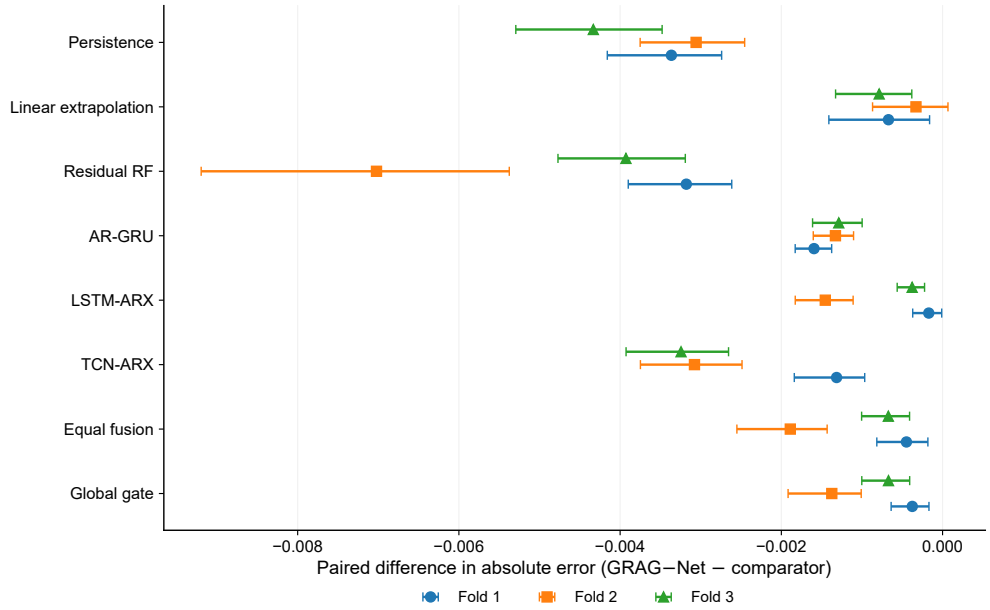


Figure 9. Paired moving-block bootstrap differences in absolute error for the primary block length of 30 samples. Each point is the mean of $|e_{\mathrm{GRAG}}| - |e_B|$ and the horizontal interval is the 95% bootstrap confidence interval. Negative values favor GRAG-Net.

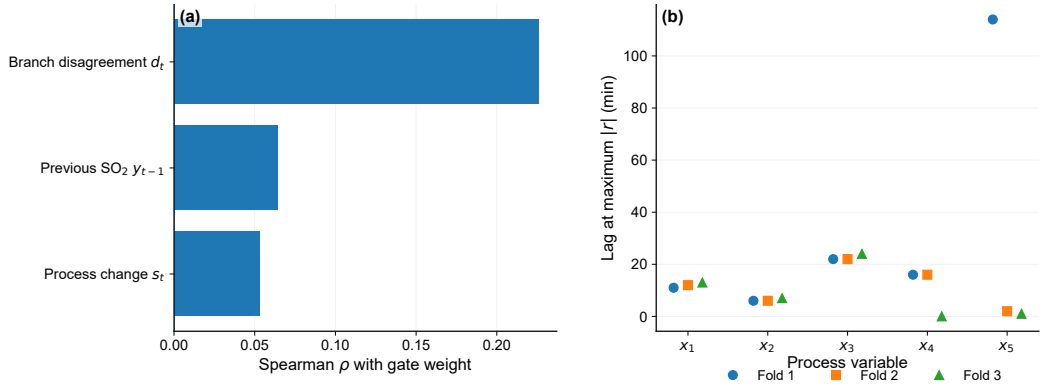


Figure 10. Inference-aware gate and temporal-lag diagnostics. (**a**) Pooled Spearman association between gate weight and causal gate-context variables. (**b**) Training-only lag at maximum absolute process–SO_2 correlation in each fold.

3.7. SHAP Interpretation and Stability

The aggregated SHAP analysis in Figure 11 confirms that recent SO_2 history provides the dominant persistence information. The four highest-ranked features are y_{t-1} through y_{t-4}, with mean absolute SHAP values of 1.379, 0.978, 0.367, and 0.149, respectively. The strongest exogenous contributions are current x_3, $x_1(t-1)$, and $x_3(t-1)$. This ordering is physically consistent with the architecture: the autoregressive branch supplies the dominant local baseline, while airflow histories provide complementary information for correcting the estimate.

The beeswarm representation further indicates nonlinear and sample-dependent effects rather than a single fixed coefficient for each process lag. These attributions describe the fitted predictor and should not be interpreted as causal plant effects, particularly because

neighboring lags and airflow channels are correlated. Importantly, the revision no longer relies on one fitted artifact. Across the six predefined fold–seed artifacts (three folds and seeds 42 and 52), the mean pairwise rank Spearman correlation is 0.933 (range 0.906–0.984), and the mean top-20 Jaccard overlap is 0.653 (range 0.538–0.818). The dominant feature ordering is therefore stable across chronological periods and random initialization, while lower-ranked features show the expected greater variability.

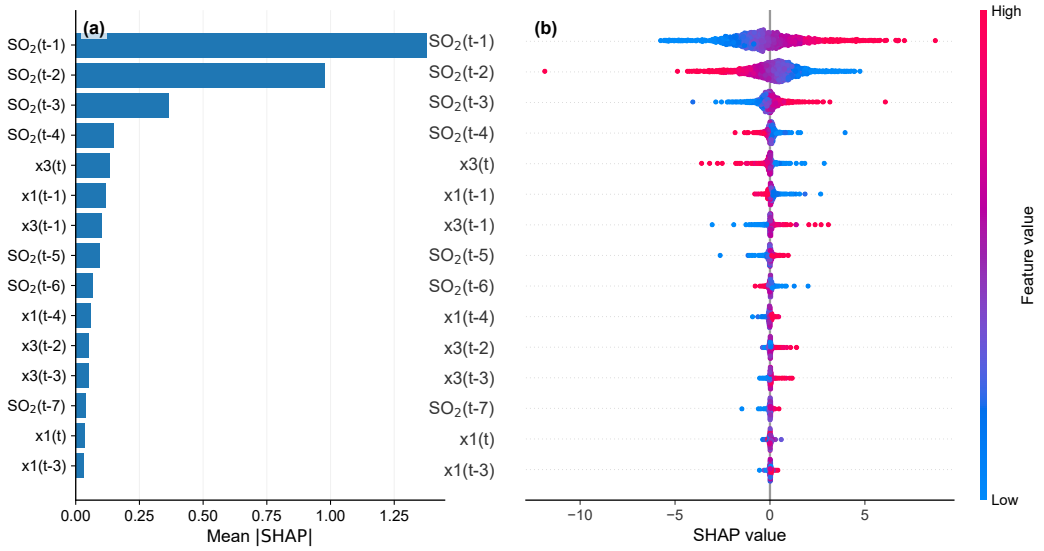


Figure 11. Aggregated SHAP interpretation of GRAG-Net. (**a**) Mean absolute SHAP importance for the 15 leading lagged features. (**b**) SHAP beeswarm showing the direction and magnitude of sample-level contributions.

Figure 12 summarizes the consistency of SHAP feature rankings across folds and seeds using Spearman correlation and top-20 Jaccard overlap.

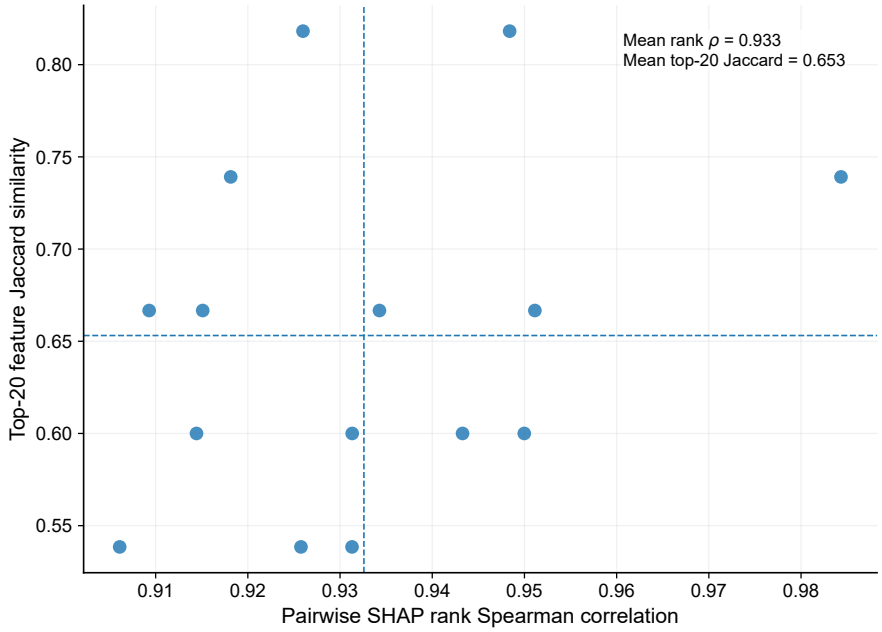


Figure 12. Cross-fold and cross-seed stability of SHAP rankings. Each point compares two fitted artifacts using full-rank Spearman correlation and top-20 Jaccard overlap; dashed lines denote the corresponding means.

3.8. Error Reliability and Target-Change Regimes

The error-reliability analysis in Figure 13 shows that GRAG-Net improves both the center and the upper tail of the error distribution relative to AR–GRU. Its median absolute error is 6.51×10^{-4} mole%, compared with 1.74×10^{-3} mole% for AR–GRU. At the 90th, 95th, and 99th percentiles, GRAG-Net errors are 0.00809, 0.01409, and 0.02935 mole%, whereas AR–GRU reaches 0.01080, 0.01866, and 0.03334 mole%, respectively. The performance gain is therefore not confined to a few extreme observations; it extends from typical errors through the upper tail.

The target-change analysis gives a complementary view. GRAG-Net attains MAE/RMSE of 0.00095/0.00210 mole% in the stable regime, 0.00243/0.00444 mole% in the changing regime, and 0.01465/0.02202 mole% during large changes. The corresponding AR–GRU values are 0.00131/0.00225, 0.00412/0.00553, and 0.02014/0.02690 mole%. GRAG-Net therefore outperforms the dedicated autoregressive recurrent model in all three regimes, with the largest absolute margin occurring during rapid target changes. This finding supports the intended role of the process branch: exogenous histories provide corrective information that is useful not only during transitions but also during routine one-step estimation.

Overall, the revised evidence is internally consistent across descriptive metrics, chronological folds, Taylor statistics, block-bootstrap intervals, formal ablations, and error-regime analysis. GRAG-Net does not obtain its advantage from target persistence alone: AR–GRU, linear extrapolation, and fixed/global fusion controls all remain weaker. Instead, the combination of residual reconstruction, separate process and autoregressive representations, and sample-dependent routing yields the most accurate and stable one-step predictions among the evaluated models. The gate, lag, and SHAP analyses further support a cautious mechanistic interpretation in which recent SO_2 history provides the dominant baseline and process history supplies complementary corrections when the two predictive pathways disagree.

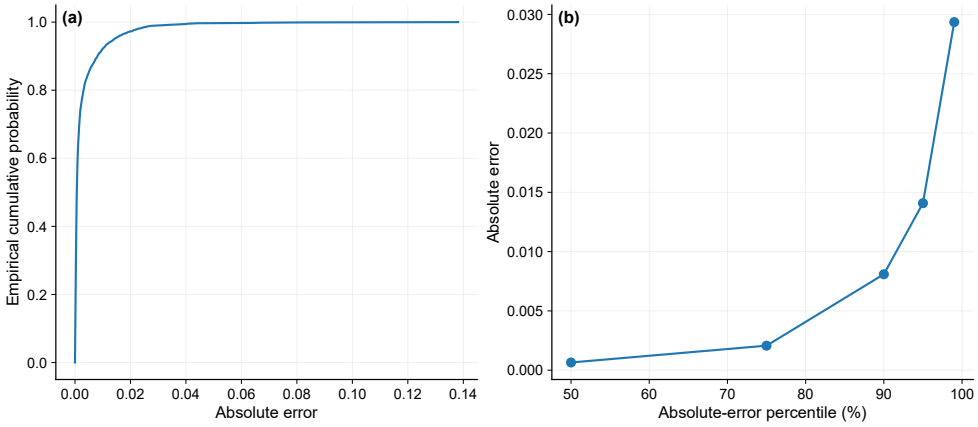


Figure 13. GRAG-Net error reliability. (**a**) Empirical CDF of absolute error. (**b**) Selected absolute-error percentiles, highlighting the concentration of typical errors and the upper tail.

4. Conclusions and Future Work

This study developed GRAG-Net for causal rolling one-step SO_2 soft sensing in an industrial sulfur recovery unit. The architecture separates exogenous process dynamics from autoregressive target persistence, predicts residual corrections relative to the latest measured SO_2 value, and combines the two branches through a sample-dependent gate. The major revision strengthens the evidence for this design through three chronological future periods, five final seeds, stronger sequence and residual baselines, independently trained ablations, and dependence-aware moving-block bootstrap analysis.

Across the 15 descriptive fold–seed evaluations, GRAG-Net achieved MAE 0.00315 ± 0.00044 mole%, RMSE 0.00790 ± 0.00064 mole%, and $R^2 = 0.97753 \pm 0.00331$, pro-

viding the strongest overall primary-metric performance among the evaluated full models. On pooled seed-ensemble predictions, MAE and RMSE were 0.00291 and 0.00766 mole%, representing reductions of 55.2% and 44.8% relative to persistence and 32.5% and 18.0% relative to AR–GRU. GRAG-Net also produced the lowest centered RMSE in all three Taylor folds, lower absolute-error quantiles from the median through the 99th percentile than AR–GRU, and lower MAE and RMSE in stable, changing, and large-change regimes. Paired moving-block bootstrap intervals favor GRAG-Net over AR–GRU in all three folds for absolute error, further supporting that the improvement is consistent within the chronological test periods rather than being driven by pooled averaging alone.

The formal ablations identify residual reconstruction and sample-dependent fusion as the principal architectural contributors: replacing dynamic routing with equal fusion, replacing it with a global gate, or removing residual reconstruction increases pooled MAE by approximately 34.5%, 27.7%, and 28.5%, respectively. Auxiliary losses and individual gate-context variables have smaller and more fold-dependent effects. Inference-aware gate analysis shows that routing is associated most strongly with branch disagreement, while training-only lag analysis identifies modest delayed relationships for the principal airflow channels. Aggregated SHAP analysis confirms that recent SO_2 lags dominate prediction and that process variables provide complementary exogenous information; the feature ranking remains stable across folds and seeds (mean rank $\rho = 0.933$).

The conclusions remain limited to one publicly available SRU benchmark and to a rolling one-step setting in which measured SO_2 history is available through $t - 1$. The study does not evaluate sustained analyzer outages, recursive multi-step prediction, cross-refinery transfer, or closed-loop deployment. Because the benchmark does not document analyzer accuracy or regulatory tolerance, the reported model errors are not interpreted as evidence of compliance with a particular hardware specification. Future work should prioritize external validation on additional SRUs, recursive outage-oriented evaluation, uncertainty calibration, drift-aware adaptation, and prospective deployment studies under documented analyzer specifications.

Funding: This research was funded by ongoing research funding program, (ORF-2026-2179), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The industrial SRU dataset analyzed in this study is publicly available in Mendeley Data, Version 1, at https://doi.org/10.17632/6p865gztg6.1. A reproducibility package containing the source code used for the final reviewer experiments, chronological split definitions, preprocessing protocol, hyperparameter search spaces and selected configurations, epoch-selection records, final random seeds, fold–seed predictions, evaluation metrics, bootstrap outputs, ablation results, and interpretability-analysis files has been retained and is available from the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Seinfeld, J.H.; Pandis, S.N. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*, 3rd ed.; Wiley: Hoboken, NJ, USA, 2016.
2. Lelieveld, J.; Evans, J.S.; Fnais, M.; Giannadaki, D.; Pozzer, A. The contribution of outdoor air pollution sources to premature mortality on a global scale. *Nature* **2015**, *525*, 367–371. [CrossRef] [PubMed]
3. Fortuna, L.; Rizzo, A.; Sinatra, M.; Xibilia, M.G. Soft analyzers for a sulfur recovery unit. *Control Eng. Pract.* **2003**, *11*, 1491–1500. [CrossRef]
4. Thameem, M.; Raj, A.; Berrouk, A.; Jaoude, M.A.; AlHammadi, A.A. Artificial intelligence-based forecasting model for incinerator in sulfur recovery units to predict SO_2 emissions. *Environ. Res.* **2024**, *249*, 118329. [CrossRef] [PubMed]

5. Quek, C.I.; Balasubramanian, R.; Rangaiah, G.P. Consider using 'soft analyzers' to improve SRU control. *Hydrocarb. Process.* **2000**, *79*, 101–106.
6. Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven soft sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795–814. [CrossRef]
7. Souza, F.A.A.; Araújo, R.; Mendes, J. Review of soft sensor methods for regression applications. *Chemom. Intell. Lab. Syst.* **2016**, *152*, 69–79. [CrossRef]
8. Jiang, Y.; Yin, S.; Dong, J.; Kaynak, O. A review on soft sensors for monitoring, control, and optimization of industrial processes. *IEEE Sens. J.* **2021**, *21*, 12868–12881. [CrossRef]
9. Kadlec, P.; Grbić, R.; Gabrys, B. Review of adaptation mechanisms for data-driven soft sensors. *Comput. Chem. Eng.* **2011**, *35*, 1–24. [CrossRef]
10. Brunner, V.; Siegl, M.; Geier, D.; Becker, T. Challenges in the development of soft sensors for bioprocesses: A critical review. *Front. Bioeng. Biotechnol.* **2021**, *9*, 722202. [CrossRef] [PubMed]
11. Bolf, N.; Mohler, I.; Golob, M.; Galinec, G.; Ivandić, M. Software sensor for sulphur recovery unit control. *Chem. Eng. Trans.* **2009**, *17*, 1191–1196.
12. Bidar, B.; Shahraki, F.; Sadeghi, J.; Khalilipour, M.M. Soft sensor modeling based on multi-state-dependent parameter models and application for quality monitoring in industrial sulfur recovery process. *IEEE Sens. J.* **2018**, *18*, 4583–4591. [CrossRef]
13. Yuan, X.; Wang, Y.; Yang, C.; Gui, W. Stacked isomorphic autoencoder based soft analyzer and its application to sulfur recovery unit. *Inf. Sci.* **2020**, *534*, 72–84. [CrossRef]
14. Patanè, L.; Xibilia, M.G. Echo-state networks for soft sensor design in an SRU process. *Inf. Sci.* **2021**, *566*, 195–214. [CrossRef]
15. Curreri, F.; Patanè, L.; Xibilia, M.G. RNN- and LSTM-based soft sensors transferability for an industrial process. *Sensors* **2021**, *21*, 823. [CrossRef] [PubMed]
16. Mou, T.; Liu, J.; Zou, Y.; Li, S.; Xibilia, M.G. Enhanced industrial process modeling with transfer-incremental-learning: A parallel SAE approach and its application to a sulfur recovery unit. *Control Eng. Pract.* **2024**, *148*, 105955. [CrossRef]
17. Wang, Q.; Li, L. Research on semi-supervised soft sensor modeling method for sulfur recovery unit based on ISSA-VMD-ESN. *Chem. Eng. Sci.* **2024**, *298*, 120397. [CrossRef]
18. Sun, W.; Li, L.; Lu, B.; Jiang, L.; Zhang, J. Hybrid multiscale forecasting of SRU sulfur gas concentrations using VMD, CEEMDAN and optimized PatchTST. *iScience* **2026**, *29*, 114986. [CrossRef] [PubMed]
19. Li, J.; Hao, Y. Dynamic soft sensor model based on combination of GRU and TCN–Transformer for chemical process application. *J. Meas. Sci. Instrum.* **2026**, *17*, 171–182. [CrossRef]
20. Cerqueira, V.; Torgo, L.; Mozetič, I. Evaluating time series forecasting models: An empirical study on performance estimation methods. *Mach. Learn.* **2020**, *109*, 1997–2028. [CrossRef]
21. Huber, P.J. Robust estimation of a location parameter. *Ann. Math. Stat.* **1964**, *35*, 73–101. [CrossRef]
22. Sun, W. Industrial Sulfur Gas Concentration Dataset for SRU Forecasting. In *Mendeley Data*; Version 1; Elsevier: Amsterdam, The Netherlands, 2026. [CrossRef]
23. Drucker, H.; Burges, C.J.C.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. *Adv. Neural Inf. Process. Syst.* **1997**, *9*, 155–161.
24. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
25. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
26. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794. [CrossRef]
27. Bergmeir, C.; Hyndman, R.J.; Koo, B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Comput. Stat. Data Anal.* **2018**, *120*, 70–83. [CrossRef]
28. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 2623–2631. [CrossRef]
29. Taylor, K.E. Summarizing multiple aspects of model performance in a single diagram. *J. Geophys. Res. Atmos.* **2001**, *106*, 7183–7192. [CrossRef]
30. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4765–4774.
31. Cao, L.; Ji, X.; Cao, Y.; Li, Y.; Siang, L.C.; Li, J.; Pediredla, V.K.; Gopaluni, R.B. Interpretable soft sensors using extremely randomized trees and SHAP. *IFAC-PapersOnLine* **2023**, *56*, 8000–8005. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.