

ORIGINAL RESEARCH

Open access

Energy Efficiency Optimization in Industrial Systems Using Law of Conservation-Guided Neural Networks with Gradient Aggregation

Kwame Mensah^{1*}, Kojo Asante¹, Linda Owusu²

Abstract

Energy efficiency in industrial systems remains a major challenge, as conventional data-driven models are often constrained by limited precision and insufficient data. Integrating the physical laws that govern energy systems can enhance performance and ensure greater physical plausibility; however, such models often struggle with loss computation and often overlook dynamic interactions among subsystems, leading to excessive simplification and reduced real-world utility. Accordingly, this study explores a theoretical framework for constructing a conservation-law-guided neural network to improve energy-efficiency forecasting in industrial processes. The framework embeds physical principles directly into floating nodes built upon a long short-term memory (LSTM) architecture, enabling the model to capture relationships among process variables more effectively. At the same time, gradient aggregation enhances both fluidity and interpretability. Performance was assessed through two large-scale industrial case studies—vinyl chloride monomer production and detergent powder manufacturing—where the proposed model achieved notable gains in prediction accuracy and overall reliability, delivering test prediction improvements of 12.2% and 5.87% compared to previously published approaches. Compared with methods that rely on network architecture modifications, the proposed model demonstrated superior reliability and reproducibility in energy-efficiency forecasts. Furthermore, the model effectively identified energy inefficiencies, resulting in a 4.21% decrease in energy use and a corresponding reduction of 377.35 tons in carbon emissions.

Keywords Long short-term memory, Energy efficiency, Law of conservation, Physically guided neural network, Gradient aggregation

*Correspondence:

Kwame Mensah
kwame.mensah@gmail.com

¹ Department of Electrical Power Systems, Faculty of Engineering, University of Ghana, Accra, Ghana

² Department of Energy Optimization and Analytics, Faculty of Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

Introduction

Across all industrial sectors, energy efficiency is widely recognized as a significant concern that incurs substantial operational expenses and has considerable environmental consequences [1]. Although energy efficiency can be defined in various ways, the two most prevalent metrics are energy intensity and specific energy consumption [2]. These metrics standardize energy use against profit or production output, allowing quantification of energy expenditure throughout process operations or enabling energy benchmarking across entire industrial domains [3]. Traditionally, rigorous studies of energy efficiency analysis have emphasized the law of conservation and exergy-based approaches for system modeling [4]. Yet in numerous industries, the absence of complete measurement data hinders the development of system models using these techniques or even conventional process optimization strategies [5]. As a result, there has been a

growing shift toward artificial intelligence (AI)-driven techniques and data-driven modeling to overcome these constraints [6].

Background, trend, and timeline of energy efficiency research

Within the domain of energy research, a wide array of AI-based tools for improving energy efficiency has emerged. These AI models can be grouped into three distinct eras according to their complexity and historical development, as illustrated in Figure 1.

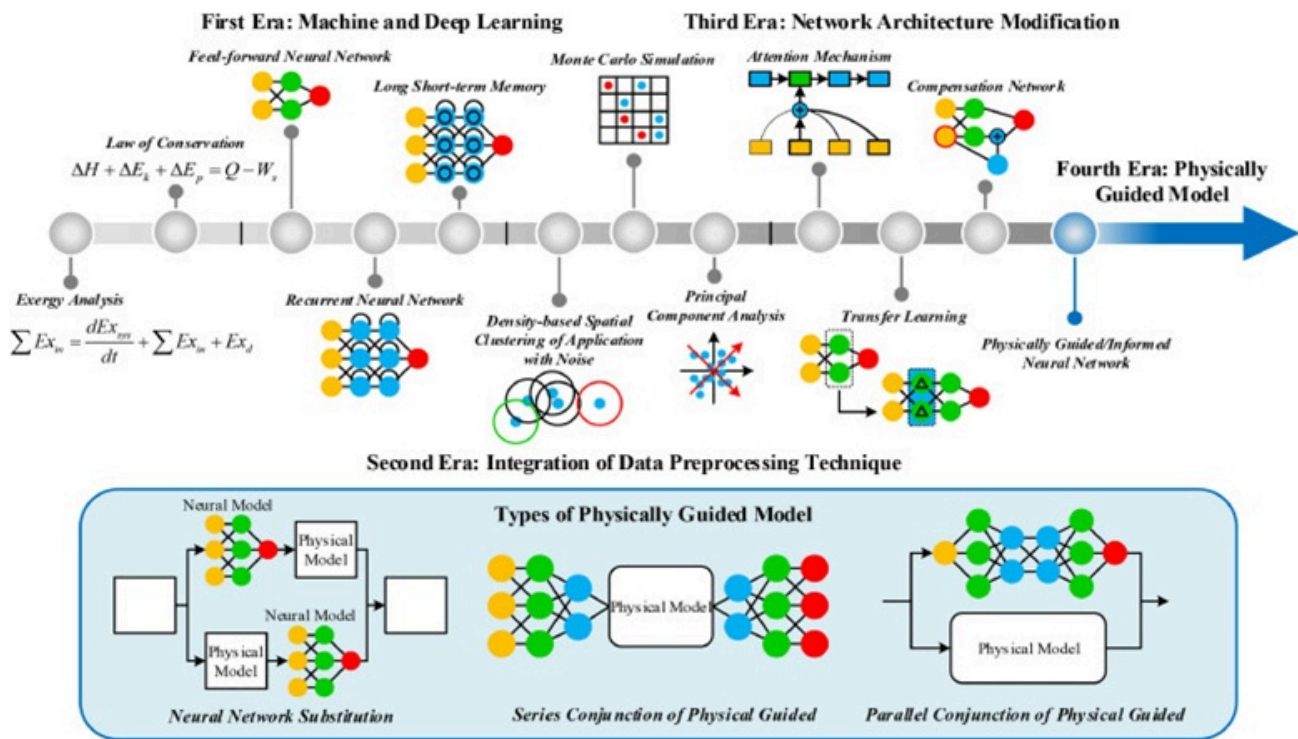


Figure 1. Timelines of energy efficiency modeling research and trends.

First era: machine learning and deep learning

The emergence of machine learning and deep learning techniques marked the first era. During its early phase, efforts primarily focused on identifying novel applications of AI in the energy sector, including energy structure optimization and energy efficiency prediction based solely on machine learning and deep learning [7]. For instance, Paterakis *et al.* [8] introduced a multi-layer perceptron approach for forecasting energy demand. Their model achieved greater prediction accuracy for energy consumption than conventional techniques such as linear regression and Gaussian process regression. Despite this progress, further improvements in prediction performance remained difficult, owing to challenges with hyperparameter tuning and the inherent limitations of the neural network architecture [9]. These constraints prompted exploration of alternative deep learning methods, including recurrent neural networks (RNN) [10], which offered the prospect of improved results in energy-related tasks. Although RNNs achieved higher accuracy, they were more prone to gradient explosion and vanishing gradients, which complicated training and compromised model stability [11]. Subsequent studies in the energy field addressed these drawbacks by adopting long short-term memory (LSTM) networks in various case studies. The memory cells and state variables in LSTMs proved capable of preserving long-range dependencies and alleviating gradient-related problems [12]. Nonetheless, ongoing limitations eventually directed energy research toward the following era, which centered on the incorporation of advanced data processing methods [13].

Second era: integration of data processing techniques

During this period, the majority of energy-related studies concentrated on incorporating data preprocessing steps into the modeling process to mitigate specific challenges. For instance, to counteract substantial noise present in process signals, Moghadasi et al. [14] employed a machine learning approach that combined density-based spatial clustering of applications with noise (DBSCAN) to remove noise and filter out outliers from time-series data before predicting steam usage in a gas sweetening unit. Three additional preprocessing strategies frequently adopted include Monte Carlo (MC) simulation [15], recursive feature elimination (RFE) [16], and principal component analysis (PCA) [17]. MC simulation is commonly used to generate synthetic samples that match the statistical properties of real observations, thereby addressing problems arising from small datasets or setpoint-dependent issues [18]. Similarly, RFE and PCA have been widely used for feature selection, dimensionality reduction, and the resolution of multicollinearity [19]. Despite these benefits, embedding data preprocessing steps increases both training time and computational demands, further complicating the overall model-building workflow. This added burden and intricacy do not invariably translate into commensurate improvements in model effectiveness, particularly when dealing with very large datasets or applications that demand real-time computation.

Third era: network architecture modification

The drawbacks of integrating data preprocessing prompted the emergence of the third era in model development: the modification of network architectures. In this phase, conventional preprocessing methods were largely supplanted by redesigned network topologies that incorporated cutting-edge structural elements such as compensation layers [20], attention mechanisms (AM) [21], transfer learning (TL) [22], and generative adversarial networks (GAN) [23]. Compensation-based networks provide more effective noise suppression than DBSCAN, primarily because they operate considerably faster and preserve the full size of the available dataset [24]. Attention mechanisms are frequently embedded within LSTM frameworks, giving rise to so-called AM-LSTM architectures [25]. This enhancement allows the model to dynamically assign importance weights to different time steps in the sequence, enabling selective emphasis on the most informative parts of the input without manual feature engineering [26].

Furthermore, techniques such as TL and GANs can serve as alternatives to MC simulation to address data scarcity. Transfer learning, often applied in experimental datasets or digital twin contexts, involves initially training a model on a large volume of data before fine-tuning it on a smaller set [27]. GANs, on the other hand, generate synthetic samples that replicate actual process dynamics, thereby expanding the training dataset and alleviating challenges related to insufficient or imbalanced data [28]. Nevertheless, despite these innovations, the third era has encountered certain constraints. The creation of intermediate variables within these architectures can increase complexity and introduce potential errors, since such variables may not consistently correspond to true physical phenomena [29]. Moreover, these models may fall short of fully incorporating the fundamental physical laws governing energy systems.

Gaps of physically guided neural network research

Although research efforts within and beyond the energy domain have progressed along parallel paths to date, the prevailing direction in non-energy fields is transitioning toward a new phase of modeling research centered on physically guided—or physically informed—neural networks [30]. This emerging era seeks to embed physical laws and domain-specific expertise directly into neural network structure. By incorporating the underlying physics into the AI framework, these approaches generate intermediate variables that remain consistent with physical principles, thereby reducing complexity while enhancing overall dependability [31]. Physically guided neural networks can be divided into three main categories: physically guided loss functions, physically guided initialization strategies, and physically guided architectures. Within the physically guided architecture category, existing literature outside the energy domain classifies the integration approaches into three types: substitution modules [32], series conjunctions [33], and parallel conjunctions [34], as outlined in Table 1, which provides a review of physically guided approaches in energy-related research.

Table 1. Summary of related energy research, methodologies, limitations, and physical rules.

Reference	Methodology	Physical rule	Limitation
Paterakis et al. [8]	Multi-layer perceptron model for prediction.	–	Performance limitations due to neural architecture constraints.
Moghadasi et al. [14]	Machine learning combined with DBSCAN for noise removal and outlier filtering.	–	Potential issues from limited data resulting from the cleaning process.
Panjapornpon et al. [24]	Transfer learning integrated with a multi-channel CNN framework to handle multi-rate sampling.	–	Parallel predictions merged using a common hidden node.
Gao et al. [34]	A hybrid mechanism combined with a least squares support vector machine model, optimized by a whale algorithm for error correction.	Hybrid mechanism with an enhanced compensation algorithm.	Heavy reliance on precise parameter tuning and data quality for accurate error correction.
Machalek et al. [35]	Substitution module replacing reaction rate equations with neural networks.	Substitution of reaction rates guided by conservation laws.	Simplified dynamics due to dependence on substitution modules.
Wang et al. [36]	Sequential integration of neural and physical models for improved predictions.	Sequential integration with physical constraints for robustness.	Increased error propagation between neural and physical model components.
Zhang et al. [37]	Physically informed neural network incorporating physical intermediate variables, optimized with the whale optimization algorithm.	Integration of the friction coefficient and heat transfer coefficient as physical variables.	Dependency on accurate historical data for physical parameters and challenges in hyperparameter tuning.
Gallup et al. [38]	Physics-guided loss functions, architectural modifications, and initialization techniques for better training and generalization.	Physics-based loss functions and initialization rules derived from empirical models.	Applied only to small systems, neglecting system-wide interactions.
Gao et al. [39]	Physics-embedded graph convolution kernel using Gaussian-Seidel iteration with correlative loss to capture physical relationships.	Gaussian-Seidel-based graph convolution kernel embedding physical topologies and constraints.	High computational complexity for large-scale systems and needs precise physical models for initialization.

Physically guided loss functions directly weave known physical constraints into the cost function of a neural network. This is achieved by appending an additional penalty term that discourages predictions that deviate from accepted physical laws, thereby stabilizing the optimization procedure and boosting the model's ability to generalize [40]. Physically guided initialization, by contrast, leverages domain expertise to set the network's starting weights and biases. Rather than using random values or statistics derived solely from training data, it draws on established physical equations or empirical relations, providing a meaningfully informed baseline that often accelerates training convergence and yields stronger final results [41].

Even so, both strategies come with important restrictions. Implementing physically guided losses or initializations requires highly accurate physical modeling and full observability of the relevant governing equations (e.g., mass/energy conservation principles). Whenever critical measurements or constitutive terms are absent, correctly evaluating the loss,

upholding physical plausibility, or selecting a valid initial condition becomes difficult or impossible, often leading to degraded accuracy and poorer overall performance [38].

In response to these shortcomings, researchers developed the physically guided architecture paradigm. Instead of modifying only the objective or starting point, this family of methods embeds physical knowledge at a structural level by replacing selected operations in classical physical simulators with learnable neural components [39]. A representative example is the substitution of a reaction kinetics sub-model with a neural surrogate in dynamic energy-system simulations, as proposed by Machalek *et al.* [35]. Although such replacements can substantially reduce the computational cost of certain expensive subroutines in traditional equation-based models, they risk overly simplifying nonlinear couplings, thereby sacrificing predictive fidelity [42].

Another widely explored strategy involves a series (sequential) conjunction, in which a neural network and a physics-based model are cascaded so that the output of one feeds directly into the next [37]. Wang *et al.* [36] demonstrated this idea by feeding neural predictions into mechanistic equations, producing hybrid forecasts with notably higher accuracy and greater robustness. Both substitution-based and sequential hybrid approaches are generally restricted to standalone energy units and tend to disregard time-varying exchanges among interconnected subsystems. An additional drawback is the strong sensitivity of end-to-end performance to the accuracy of each component: inaccuracies arising in either the data-driven or equation-based portion can accumulate and magnify throughout the pipeline [43].

Parallel conjunction physically informed models provide a promising way to sidestep many of these difficulties. In this configuration, neural networks and physics-based solvers operate concurrently, each tackling complementary aspects of the problem or furnishing independent yet synergistic information [44]. That said, the need for careful orchestration of parallel execution paths often discourages practitioners from deploying such architectures in physically constrained settings. Furthermore, most existing implementations link the various physically guided building blocks using only elementary summation or concatenation nodes [39]. These rudimentary junctions struggle to represent rich, nonlinear dependencies that emerge jointly from physical constraints and fluctuating input patterns. As a consequence, overly simplistic topologies may fail to reproduce transient behavior across subsystems faithfully. They can unintentionally erode model transparency—particularly when important hidden states cannot be directly observed.

Research contributions

To achieve more reliable energy-efficiency forecasting for intricate industrial processes, the present work proposes an advanced hybrid, physically guided framework, the Law of Conservation-Guided Neural Network (LOCGNN), that integrates parallel and sequential conjunctions. The architecture equips each floating node with dedicated LSTM units that model distinct subsystems. This design enables targeted learning of temporal dynamics and instantaneous input patterns within each subsystem while explicitly modeling cross-subsystem interactions. By strategically using gradient aggregation to infuse conservation principles throughout the network, LOGCNN overcomes several persistent weaknesses of earlier physically informed neural approaches. It enforces physically consistent coupling dynamically and markedly improves gradient flow and synchronization in hybrid (series/parallel) topologies. Crucially, the model dispenses with the need for unobservable latent states during either physical pre-training or loss-term evaluation, yet still generates dependable, human-interpretable energy-efficiency estimates even under partial observability of conservation-related quantities.

The work delivers the following four core contributions:

- (1) Presents the complete methodological blueprint of LOGCNN, which systematically embeds conservation-law constraints into recurrent neural architectures tailored for energy applications and characterized by enhanced information flow.

(2) Explains how gradient-aggregation mechanisms render the network's decisions more transparent and physically meaningful compared with standard data-driven energy models, while simultaneously strengthening coordination between sequential and concurrent hybrid branches.

(3) Investigates the interplay between architectural complexity and forecasting capability by closely examining transient inter-unit exchanges, recycle loops, and mass/energy connectivity flows, following the methodological guidelines introduced earlier.

(4) Clarifies the manner in which physically grounded interdependencies encoded within the network topology facilitate genuine energy-optimization tasks by guaranteeing that predicted trajectories remain compliant with fundamental balance equations.

(5) Provides rigorous empirical validation on two full-scale industrial processes—vinyl chloride monomer synthesis and detergent powder manufacturing.

Law of Conservation-Guided Neural Network

Interdependencies among the conservation laws

In engineering practice, the conservation laws—encompassing mass balance, energy balance, and component (or species) balance—stand as the most fundamental principles. These laws assert that mass, individual components, and energy are preserved within a defined system, undergoing only transformations between different forms rather than being created or destroyed. They form the essential basis for the governing equations that describe steady-state behavior in chemical processing systems. The inherent connectivity of industrial processes stems directly from the unified application of these conservation principles, as depicted in [Figure 2](#).

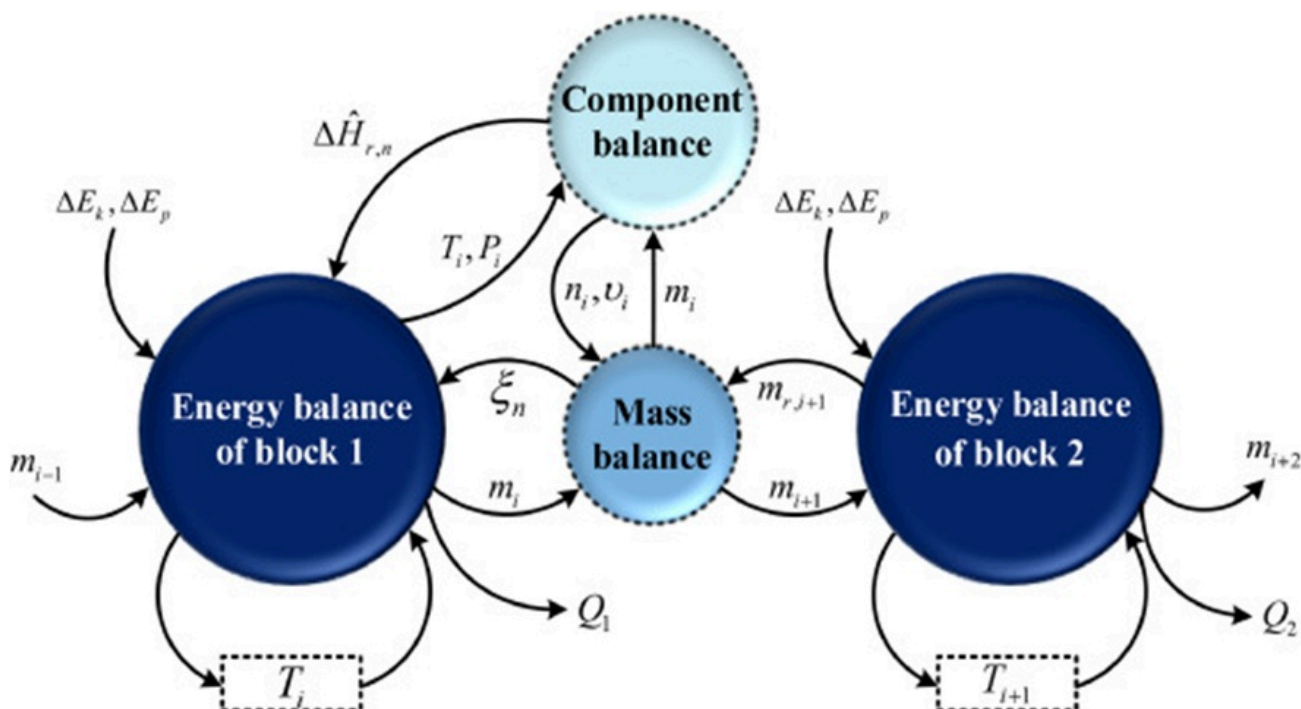


Figure 2. Interconnection between conservation laws in process modeling.

In the diagram shown in **Figure 2**, each block represents a set of equations that clarify the physical relationships linking mass, energy, and component balances, along with their mutual influences. Given that the primary focus of this investigation is energy efficiency evaluation, the analysis starts with the energy balance equation. **Eq.1** expresses the steady-state energy balance for an open system, relating variations in enthalpy (ΔH), kinetic energy (ΔE_k), and potential energy (ΔE_p) to the net heat input (Q) and shaft work (W_s).

$$\Delta H + \Delta E_k + \Delta E_p = Q - W_s \quad (1)$$

The contributions from kinetic and potential energy terms, ΔE_k as defined in **Eq. 2** and ΔE_p in **Eq. 3**, become relevant primarily in systems involving substantial fluid velocities or elevation differences. However, these effects are typically minor relative to enthalpy shifts.

$$\Delta E_k = \sum_{\text{output}} m_i v_i^2 / 2 - \sum_{\text{input}} m_i v_i^2 / 2 \quad (2)$$

$$\Delta E_p = \sum_{\text{output}} m_i g z_i - \sum_{\text{input}} m_i g z_i \quad (3)$$

In the open-system energy balance, the enthalpy variation at time step i , as outlined in **Eq. 4**, arises as the difference between the specific enthalpies (h_i) of the inlet and outlet streams. This expression further incorporates the heat of reaction ($\Delta \hat{H}_{r,n}$), explicitly tying energy variations to chemical transformations by accounting for the linkage between thermochemistry and reaction stoichiometry.

$$\Delta H = \xi_n \Delta \hat{H}_{r,n} (T_i, P_i) + \sum_{\text{output}} m_i h_i (T_i) - \sum_{\text{input}} m_i h_i (T_i) \quad (4)$$

The energy balance serves as the leading computation, drawing on kinetic and potential energy contributions, heat exchanges, and work terms to determine the enthalpy change associated with block 1. Simultaneously, the mass balance determines the inlet, outlet, and recycle mass flow rates through block 1 using **Eq. 5**, providing essential inputs to downstream blocks.

$$m_i = m_{i+1} + \Delta m \quad (5)$$

Here, Δm denotes the net accumulation of mass within the block, a quantity that is generally negligible under steady-state conditions. These identical mass flow rates also inform the component balance, which helps compute the molar flow rates (n_i) and stoichiometric coefficients (ν_i) for each reaction. The resulting molar flows loop back to influence the mass balance by enabling calculation of reaction extents (ξ_n) according to **Eq. 6**; these extents, in turn, adjust the energy balance to reflect the thermal impacts of the reactions.

$$\xi_n = \frac{m_{\text{inlet},j} - m_{\text{outlet},j}}{\nu_n M_j} = \frac{n_{\text{inlet},j} - n_{\text{outlet},j}}{\nu_n} \quad (6)$$

In this expression, M_j represents the molecular weight of species j participating in reaction n . Once the extent of the reaction is determined, the remaining quantity is the heat of reaction itself. The component balance employs standard heats of formation (or Hess's law) to compute the reaction enthalpy at the prevailing pressure and temperature conditions. Ultimately, the energy balance yields both the enthalpy change and the system's final temperature. The sequence of computations described here not only directs process regulation within individual blocks but also establishes

a logical data hierarchy for building a model that remains physically consistent. Mass balance information propagates to the component balance, which subsequently refines the energy balance, forming an iterative cycle of interdependent calculations that continuously sharpens the overall representation.

System decomposition and construction of the physically guided network

The initial industrial application examined in this research concerns the manufacturing of vinyl chloride monomer (VCM). **Figure 3** provides an overview of the full process flow for VCM production, highlighting the ethylene dichloride (EDC) cracking unit together with the subsequent VCM purification train. During the EDC cracking stage, elevated temperatures generated by fuel combustion drive the thermal decomposition of EDC—themselves generated upstream via chlorination and oxychlorination—into VCM, HCl, and various minor by-products, according to the primary reaction.

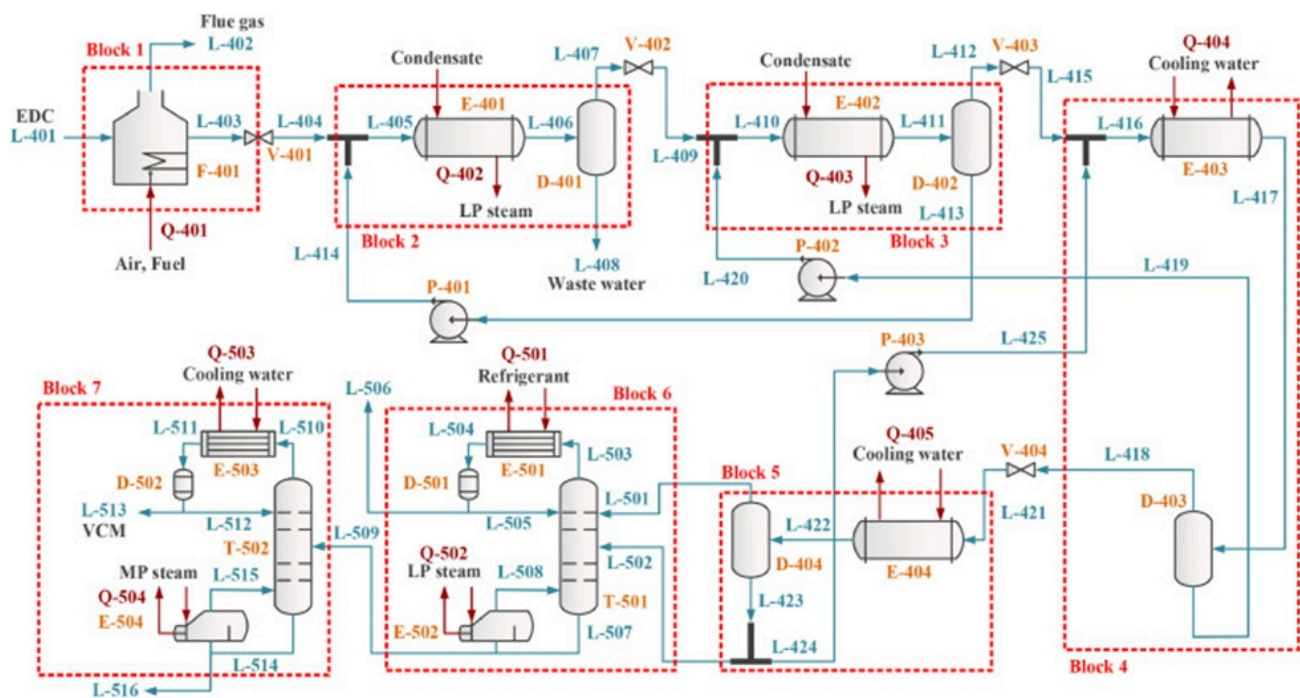
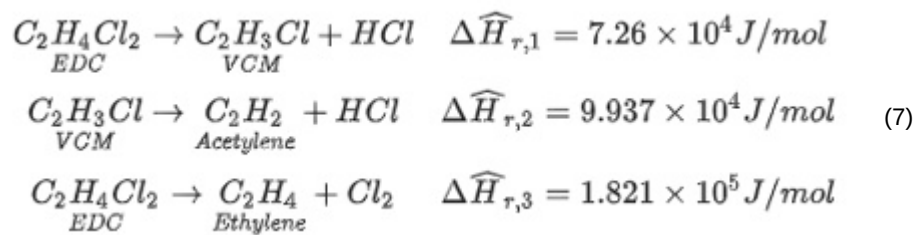


Figure 3. Process schematic of EDC cracking and VCM purification stage.

Following the cracking furnace, the hot effluent gas stream passes through a sequence of quench units. These quenchers rapidly cool and partially condense the mixture, utilizing condensate and cooling water as the primary heat sinks to facilitate separation of by-products. The cooled stream then advances to purification, where two distillation columns operate in series: the first column recovers unreacted EDC for recycle back to the cracking furnace, while the second isolates HCl, which is redirected to the oxychlorination section to produce additional EDC.

The construction of the physically guided network proceeds through the following six sequential stages:

Decomposition into subsystems

Segment the overall process into discrete sub-units, referred to as “blocks.” Each block must encompass at least one heat source or utility term (Q) to support a modular evaluation strategy. Applying this criterion to the VCM production system yields a division into seven distinct blocks, each incorporating at least one utility variable Q , where $Q \in \{Q_{401} - Q_{504}\}$. The precise delineation of each block’s boundaries is illustrated in [Figure 3](#).

Development of the Energy Balance Network

Construct a compact neural sub-network for each block based on its specific open-system energy balance equation, which captures the inflows and outflows relevant to that sub-unit. Typical input and output features for these blocks consist of readily observable process quantities such as temperature, pressure, and mass flow rates. Nevertheless, because the model’s structure is anchored in the plant’s inherent physical layout rather than in loss terms or initialization schemes that require exhaustive data for consistency checks, it exhibits considerable adaptability. This inherent flexibility stems from the system’s reliance on its underlying physical configuration, enabling robust operation even when certain sensor readings are unavailable. The conservation principles embedded in the framework confer high adaptability (“liquidity”), allowing the network to accommodate fluctuations or gaps in input measurements as long as the governing physical laws hold. The steady-state energy balance expressions applicable to blocks 1 through 7 appear in [Eqs. 8–14](#), respectively.

Furnace:

$$\begin{aligned} m_{L401}^{x_1} h_{L401}(T_{L401}) + Q_{401}^{y_1} &= m_{L402} h_{L402}(T_{L402}) + m_{L403} h_{L403}(T_{L403}^{x_2}) \\ &+ \xi_{F401} \Delta \widehat{H}_{r,F401} \left(T_{F401}^{x_3}, P_{F401} \right) \end{aligned} \quad (8)$$

Quenchers:

$$\begin{aligned} m_{L404} h_{L404}(T_{L404}) + m_{L414}^{x_{8,t-1}} h_{L414}(T_{L414}^{x_{9,t-1}}) + Q_{402}^{y_2} &= m_{L407} h_{L407}(T_{L407}^{x_5}) \\ &+ m_{L408} h_{L408}(T_{L408}) \end{aligned} \quad (9)$$

$$\begin{aligned} m_{L409} h_{L409}(T_{L409}) + m_{L420}^{x_{12,t-1}} h_{L420}(T_{L420}^{x_{13,t-1}}) + Q_{403}^{y_3} &= m_{L412} h_{L412}(T_{L412}^{x_7}) \\ &+ m_{L413}^{x_8} h_{L413}(T_{L413}^{x_9}) \end{aligned} \quad (10)$$

$$\begin{aligned} m_{L415} h_{L415}(T_{L415}) + m_{L425} h_{L425}(T_{L425}^{x_{14,t-1}}) + Q_{404}^{y_4} &= m_{L418}^{x_{10}} h_{L418}(T_{L418}^{x_{11}}) \\ &+ m_{L419}^{x_{12}} h_{L419}(T_{L419}^{x_{13}}) \end{aligned} \quad (11)$$

$$\begin{aligned} m_{L421} h_{L421}(T_{L421}) + Q_{405}^{y_5} &= m_{L424} h_{L424}(T_{L424}^{x_{14}}) + m_{L501}^{x_{15}} h_{L501}(T_{L501}) \\ &+ m_{L502} h_{L502}(T_{L502}) \end{aligned} \quad (12)$$

Distillation columns:

$$m_{L501}h_{L501}(T_{L501}) + m_{L502}h_{L502}(T_{L502}) + Q_{502}^{y_6} = m_{L506}^{x_{16}}h_{L506}^{x_{17}}(T_{L506})^{(HCl)} + m_{L509}h_{L509}(T_{L509})^{x_{18}} + Q_{501}^{y_7} \tag{13}$$

$$m_{L509}h_{L509}(T_{L509}) + Q_{504}^{y_8} = m_{L513}^{x_{19}}h_{L513}^{x_{20}}(T_{L513})^{(VCM)} + m_{L516}^{x_{21}}h_{L516}^{x_{22}}(T_{L516})^{(UnreactEDC)} + Q_{503}^{y_9} \tag{14}$$

In these expressions, superscripts x and y denote measurable quantities specific to the VCM facility. Variables marked x serve as inputs for energy efficiency forecasting, whereas those labeled y represent target outputs. In the context of VCM manufacturing, contributions from kinetic energy, potential energy, and pump work are comparatively negligible [45]; omitting them introduces only minor discrepancies in the computed overall energy performance. Table 2 summarizes the notation for all input and output variables used to predict energy efficiency in the VCM process.

Table 2. Notation of input and output variables for the VCM process.

No.	Variable	Unit	No.	Variable	Unit
x1	Feed EDC mass flow rate	kg/h	x17	Temperature at L-506	K
x2	Temperature at F-401 outlet	K	x18	Temperature at L-509	K
x3	Cracking temperature at F-401	K	x19	VCM flow rate at L-513	kg/h
x4	Mass flow rate at top D-401	kg/h	x20	Temperature at L-513	K
x5	Temperature at top D-401	K	x21	Recycled EDC flow rate at L-516	kg/h
x6	Mass flow rate at top D-402	kg/h	x22	Temperature at L-516	K
x7	Temperature at top D-402	K	y1	Energy efficiency of F-401	GJ/t
x8	Mass flow rate of recycle stream D-402	kg/h	y2	Energy efficiency of E-401	GJ/t
x9	Temperature at bottom D-402	K	y3	Energy efficiency of E-402	GJ/t
x10	Mass flow rate at top D-403	kg/h	y4	Energy efficiency of E-403	GJ/t
x11	Temperature at top D-403	K	y5	Energy efficiency of E-404	GJ/t
x12	Mass flow rate of recycle stream D-403	kg/h	y6	Energy efficiency of E-502	GJ/t
x13	Temperature at bottom D-403	K	y7	Energy efficiency of E-501	GJ/t
x14	Temperature at bottom D-404	K	y8	Energy efficiency of E-504	GJ/t
x15	HCl tower feed rate at L-501	kg/h	y9	Energy efficiency of E-503	GJ/t
x16	Mass flow rate of distilled HCl	kg/h	-	-	-

Handling recycle streams with temporal stamping (where applicable)

Recycle flows are tracked by associating each time step with its point of origin. Upon arrival at the destination, a placeholder variable is introduced to encode the stream's condition from the immediately preceding time step, thereby accounting for internal transport lags. The VCM flowsheet features three principal recycle loops: those associated with

quencher 2 (L-414), quencher 4 (L-420), and quencher 4 (L-425). The resulting topology, incorporating system partitioning, energy-balance subnetworks, and recycle linkages, is shown in **Figure 4a**.

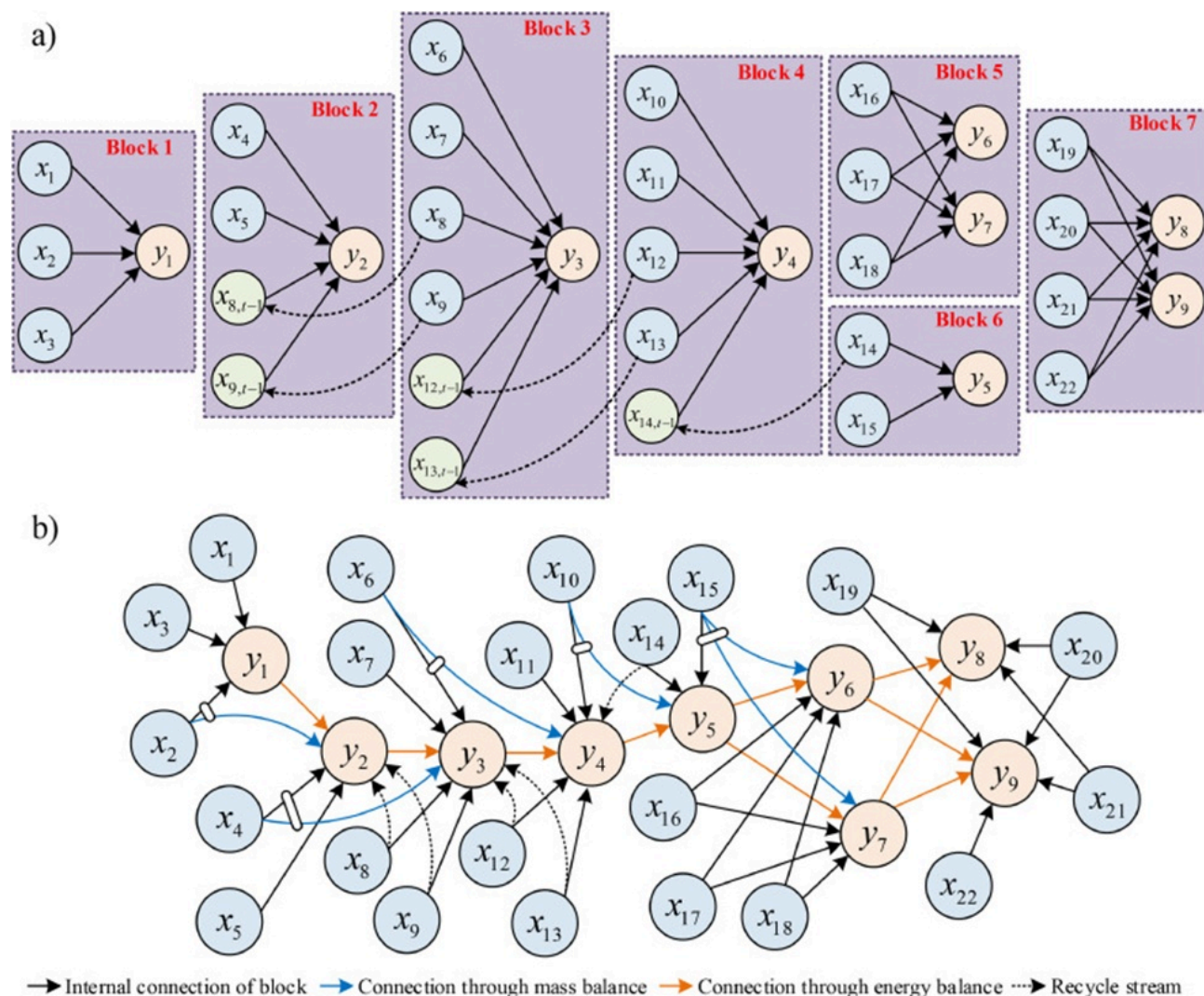


Figure 4. The final architecture of the LOGGNN model at the end of a) Step 3 and b) Step 6.

Propagation of shared mass-balance inputs across interconnected blocks (where applicable)

Whenever adjacent blocks are coupled via mass continuity, the connecting stream variable must be jointly utilized by both units. For instance, the temperature of stream L-403 (denoted x_2) exerts opposing influences on enthalpy in the linked blocks: an increase in this temperature lowers enthalpy in block 1 while elevating enthalpy transfer into block 2. Such bidirectional coupling gives rise to specialized nodes that affect multiple downstream outputs and leverage aggregated gradients during parameter updates within LOGGNN. Further elaboration on these specialized nodes is provided later.

Incorporation of direct energy-balance linkages (where applicable)

When the energy balance equations of successive blocks are directly interdependent, the computed variables from the upstream block are fed as inputs to the downstream block. This mechanism introduces supplementary weighting factors that explicitly encode cross-block dependencies, thereby enabling the model to simulate propagating effects of perturbations across different sections of the plant more realistically.

Holistic assembly of all blocks

Combine the individually formulated blocks into a unified, physically informed predictive model for energy-efficiency assessment. This final synthesis guarantees comprehensive coverage of the entire process; the completed LOCGNN topology after completion of all six stages is depicted in [Figure 4b](#).

Long short-term memory (LSTM) with gradient aggregation

The lines in [Figure 4](#) represent individual neural layers. Every interlayer connection in the LOCGNN architecture is implemented using LSTM units. Employing LSTM enables effective modeling of both short-range transients and extended temporal correlations inherent in the process variables, facilitated by its internal gating structure that regulates information retention and update [\[46\]](#).

Information propagation through an LSTM cell commences at the input gate, which determines the fraction of incoming data to be admitted into the cell's memory state according to [Eq. 15](#).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t, x_{t-1}] + b_i) \quad (15)$$

Where σ stands for the sigmoid function, h_{t-1} is the hidden state carried over from the immediately prior timestep, x_t denotes the fresh input arriving at the present timestep, x_{t-1} reflects the lagged input originating from the recycle stream logic defined earlier in step 3, and W together with b represent the trainable weight matrices and bias vectors.

The procedure then advances to revising the internal cell memory. The forget gate governs this revision, regulating how much of the previously stored information to keep versus discard, as formalized in [Eq. 16](#).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t, x_{t-1}] + b_f) \quad (16)$$

Thereafter, a prospective new cell-state candidate is proposed using [Eq. 17](#).

$$\hat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t, x_{t-1}] + b_C) \quad (17)$$

Where $\tanh$ signifies the hyperbolic tangent nonlinearity.

With the input gate vector, forget gate vector, and candidate cell value now available, the actual cell state is refreshed by blending these elements according to [Eq. 18](#).

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (18)$$

Next, the output gate determines which portion of the current cell state to expose to form the new hidden state. It does so by processing the current input, the delayed input from the previous cycle, and the earlier hidden state, as defined in [Eqs. 19–20](#).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t, x_{t-1}] + b_o) \quad (19)$$

$$h_t = o_t * \tanh(C_t) \quad (20)$$

Collectively, the three gates (input, forget, output) plus the candidate update mechanism allow the LSTM to selectively remember, forget, and transmit information across arbitrary time spans, thereby enabling it to discover and represent highly nonlinear and long-range patterns within sequential data.

Ordinarily, the LSTM's final hidden-state vector is forwarded to a dense (fully connected) projection layer that reshapes its dimensionality to match the number of target variables required for the downstream regression task, as given by [Eq. 21](#).

$$\hat{y}_t = W_{fc} \cdot h_t + b_{fc} \quad (21)$$

Where $\hat{y}_t$ is the network's forecast at timestep t ; subscripts i , f , $\tilde{C}$, o , and fc identify the input gate, forget gate, cell-state candidate, output gate, and final fully-connected projection, respectively.

In the LOCGNN topology, nodes fall into one of two broad categories depending on their connectivity pattern: (i) nodes whose activation affects exactly one downstream prediction target, or (ii) nodes whose activation simultaneously impacts several downstream targets (a category that also encompasses situations involving delayed/recurrent feedback), as visualized in [Figure 5](#).

Backpropagation begins by evaluating, for each target influenced by a given node, the regression error using the half-mean-squared-error criterion defined in [Eq. 22](#).

$$L = \frac{1}{N} \sum_{j=1}^n \left(\frac{1}{M} \sum_{t=1}^m (y_{t,j} - \hat{y}_{t,j}(\theta_{o,w,f,i}))^2 \right) + \lambda \Omega(W \in \theta_{o,w,f,i}) \quad (22)$$

Where M indicates the total count of training examples, N is the count of prediction targets affected by the node under consideration, $\hat{y}_{t,j}$ is the predicted value for target j at time t , θ collects all trainable weights/biases associated with the LSTM gates, λ governs the intensity of weight decay, and $\Omega(\cdot)$ is the chosen regularizer applied to the parameters.

Per-node gradients are subsequently derived with respect to θ by considering the scope of that node's influence, following the aggregation rule in [Eq. 23](#).

$$\frac{\partial L}{\partial y_t} = \begin{cases} j = 1, \frac{\partial L}{\partial y_1} \\ j = 2, \frac{\partial L}{\partial y_1} + \frac{\partial L}{\partial y_2} \\ j = n, \sum_{j=1}^n \frac{\partial L}{\partial y_j} \end{cases} \quad (23)$$

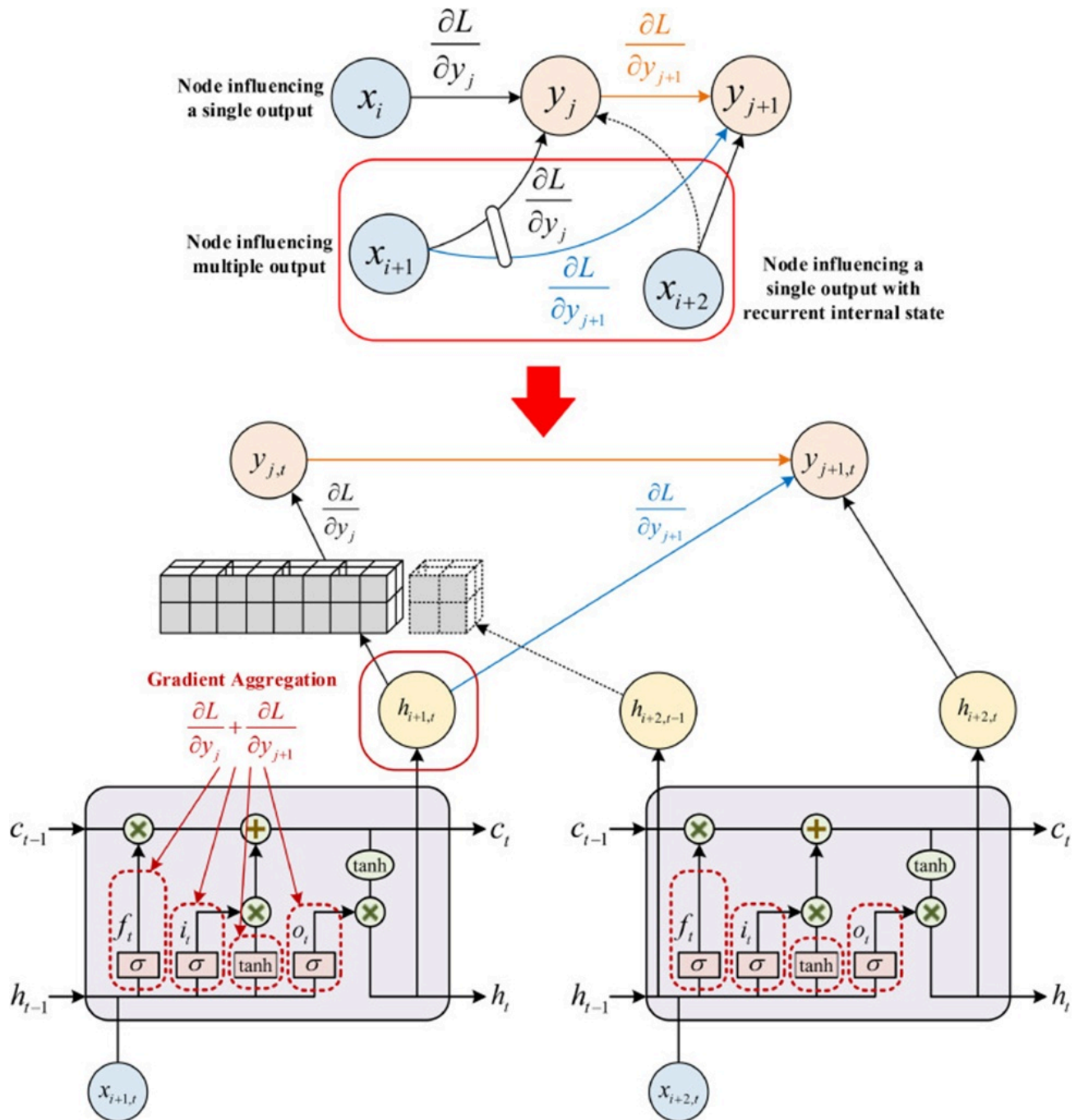


Figure 5. Categorization of parameter domains together with the gradient aggregation strategy inside the LSTM cells.

When a node connects to only one output target, its parameter gradient is obtained conventionally directly from that single loss term, in line with the transformation-gated LSTM formulation [47].

Conversely, when a node contributes to multiple output targets, the individual loss-derived gradients from each affected target are summed (aggregated) before the overall parameter update direction is determined [48]. For a standard LSTM cell, the learnable quantities typically comprise the weight matrices and bias vectors belonging to the input gate, forget gate, cell candidate computation, and output gate.

Gradient computation proceeds by successive application of the chain rule through each gate operation. The resulting derivative expressions—covering the output gate, cell state, input gate, and forget gate—are adapted from the reference treatment [49] and appear in Eqs. 24–27, respectively.

$$\begin{aligned}\frac{\partial L}{\partial \theta_o} &= \sum_t \left(\frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial o_t} \cdot \frac{\partial o_t}{\partial \theta_o} \right) \\ \frac{\partial y_t}{\partial o_t} &= \tanh(C_t) \\ \frac{\partial o_t}{\partial \theta_o} &= \sigma'(W_o \cdot [h_{t-1}, x_t, x_{t-1}] + b_o) \cdot [h_{t-1}, x_t, x_{t-1}]\end{aligned}\quad (24)$$

$$\begin{aligned}\frac{\partial L}{\partial \theta_c} &= \sum_t \left(\frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial C_t} \cdot \frac{\partial C_t}{\partial \theta_c} \right) \\ \frac{\partial y_t}{\partial C_t} &= o_t * \tanh h'(C_t) \\ \frac{\partial C_t}{\partial \theta_c} &= i_t * \tanh h'(W_c \cdot [h_{t-1}, x_t, x_{t-1}] + b_c) \cdot [h_{t-1}, x_t, x_{t-1}]\end{aligned}\quad (25)$$

$$\begin{aligned}\frac{\partial L}{\partial \theta_i} &= \sum_t \left(\frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial i_t} \cdot \frac{\partial i_t}{\partial \theta_i} \right) \\ \frac{\partial y_t}{\partial i_t} &= \frac{\partial y_t}{\partial C_t} \cdot \frac{\partial C_t}{\partial i_t} = o_t * \tanh h'(C_t) * \hat{C}_t \\ \frac{\partial i_t}{\partial \theta_i} &= \sigma'(W_i \cdot [h_{t-1}, x_t, x_{t-1}] + b_i) \cdot [h_{t-1}, x_t, x_{t-1}]\end{aligned}\quad (26)$$

$$\begin{aligned}\frac{\partial L}{\partial \theta_f} &= \sum_t \left(\frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial f_t} \cdot \frac{\partial f_t}{\partial \theta_f} \right) \\ \frac{\partial y_t}{\partial f_t} &= \frac{\partial y_t}{\partial C_t} \cdot \frac{\partial C_t}{\partial f_t} = o_t * \tanh h'(C_t) * C_{t-1} \\ \frac{\partial f_t}{\partial \theta_f} &= \sigma'(W_f \cdot [h_{t-1}, x_t, x_{t-1}] + b_f) \cdot [h_{t-1}, x_t, x_{t-1}]\end{aligned}\quad (27)$$

Once the aggregated gradients across all gate activations have been computed, the adaptive moment estimation (Adam) optimizer is used to update the network's trainable parameters. The process begins by calculating the first-moment estimate (momentum, m_t) and the second-moment estimate (uncentered variance scaling, v_t) according to Eqs. 28–29.

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1} \rightarrow m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \frac{\partial L}{\partial \theta} \quad (28)$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2} \rightarrow v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot \left(\frac{\partial L}{\partial \theta} \right)^2 \quad (29)$$

The parameters of LOCGNN are then adjusted using Eq. 30.

$$\theta(t+1) = \theta(t) - \alpha \cdot \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}} \quad (30)$$

Where $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected estimates of the first and second moments, respectively, β_1 is the exponential decay rate for the gradient moving average, β_2 is the exponential decay rate for the squared gradient moving average, α represents the global learning rate (tuned via hyperparameter search), and ϵ is a small positive constant introduced to avoid division by zero.

By incorporating gradient aggregation directly into the LSTM cells, the model is guided to reduce the half-mean-squared-error (HMSE) loss while respecting and reinforcing the conservation principles structurally embedded in the network topology.

Overall data processing framework and statistical indicators

In this investigation, the overall data-handling pipeline consists of three main phases, as illustrated in **Figure 6**: initial data preprocessing (standardization and dataset splitting), construction of the LOCGNN architecture, and Bayesian hyperparameter tuning. The preprocessing stage commences with z-score normalization, which eliminates scale differences among variables and promotes stable gradient-based learning by centering each feature around its mean and scaling it by its standard deviation. For every observation, this transformation subtracts the feature mean and divides by the feature standard deviation according to **Eq. 31**. The required statistics—mean and standard deviation for each input feature—are computed using **Eqs. 32–33**. As a result, the transformed dataset exhibits a distribution with zero mean and unit variance for every feature.

$$z_i^j = \frac{x_i^j - \mu^j}{\sigma^j} \quad (31)$$

$$\mu^j = \frac{\sum_{i=1}^n x_i^j}{n} \quad (32)$$

$$\sigma^j = \sqrt{\frac{\sum_{i=1}^n (x_i^j - \mu^j)^2}{n-1}} \quad (33)$$

Where x_i^j denotes the raw value of the i th sample for the j -th input feature, μ^j is the mean of the j -th feature across the entire dataset, and σ^j represents the corresponding standard deviation.

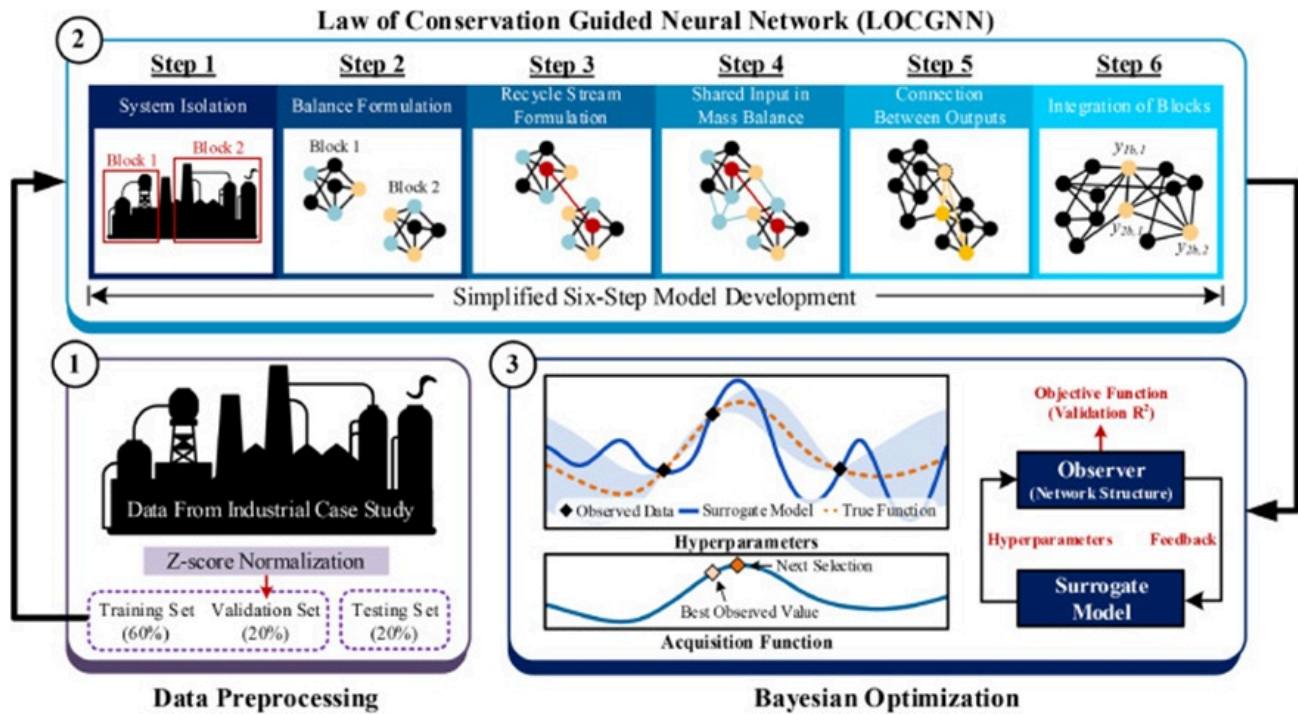


Figure 6. Overall data-processing framework for LOCGNN model development.

Subsequently, the complete dataset—comprising 2000 hourly measurements collected from the VCM production facility—is divided into training, validation, and test subsets using a 0.6:0.2:0.2 hold-out split.

Once preprocessing is complete, attention shifts to building the LOCGNN model itself, which follows a systematic six-step procedure:

- Step 1: Partition the overall process into modular blocks, ensuring that each block includes at least one utility or energy input term.
- Step 2: Derive the steady-state energy balance equation for every block to capture its specific input–output energy relationships.
- Step 3: Assign temporal stamps to recycle streams so that transport delays and cyclic material/energy flows are explicitly modeled.
- Step 4: Enable shared usage of critical stream variables across adjacent blocks to represent their mutual physical influences accurately.
- Step 5: Propagate computed outputs from upstream blocks as inputs to downstream blocks, thereby encoding sequential and plant-wide dependencies.
- Step 6: Assemble all individual blocks into a single, cohesive LOCGNN topology that enforces conservation-law consistency throughout.

During model training and hyperparameter validation, predictive performance is quantified using three complementary statistical measures: the coefficient of determination (R^2), root-mean-squared error (RMSE), and mean absolute percentage error (MAPE). These indicators are computed according to [Eqs. 34–36](#).

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (34)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (35)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (36)$$

Where $\bar{y}$ denotes the mean of the observed target values.

R^2 quantifies how well the model explains the variability in the target variable; values approaching 1 indicate an excellent fit and capture of the underlying patterns. However, R^2 provides no direct insight into the absolute or relative magnitude of prediction errors. RMSE provides an interpretable measure of average error in the target's original units. At the same time, MAPE expresses average error as a percentage of the actual values, offering an intuitive sense of relative accuracy. Together, these three indicators provide a well-rounded, multifaceted assessment of model quality, with each metric highlighting aspects that the others may overlook.

The final stage involves Bayesian optimization to tune LOCGNN's hyperparameters systematically. In this iterative framework, LOCGNN serves as the black-box objective function, reporting validation R^2 scores back to a surrogate (probabilistic) model. New hyperparameter configurations are proposed by maximizing the expected improvement (EI) acquisition function, which is evaluated using [Eq. 37](#).

$$EI(hy) = E_{hy} [\max(0, \mu_{hy}(hy_{best}) - f(hy))] \quad (37)$$

Where hy represents a candidate set of hyperparameter values for LOCGNN, $f(hy)$ is the best validation R^2 observed so far, hy_{best} is the predicted optimum according to the surrogate model, and μ_{hy} is the surrogate's predicted mean performance at the candidate point.

Expected Improvement (EI) serves as a key mechanism for directing the exploration toward promising hyperparameter configurations. It quantifies the anticipated gain relative to the best objective value encountered so far, thereby favoring combinations predicted to deliver the largest potential enhancement. In this work, Bayesian optimization was run consistently for 500 iterations. Through this methodical search, the algorithm efficiently traversed the intricate hyperparameter landscape of LOCGNN. Critical hyperparameters—namely the regularization coefficient, learning rate, and number of hidden units—were tuned within deliberately selected bounds to strike an effective compromise between predictive capability and model stability. The regularization coefficient was varied from 0 to 1; 0 implies no regularization, whereas 1 assigns equal importance to the regularization term and the primary accuracy objective. Higher values were deliberately excluded to avoid excessive emphasis on parameter shrinkage at the cost of fitting performance. The learning rate was sampled across the interval from $1e-7$ to 1, with the upper limit chosen to avert training instability and the lower limit set to guarantee reasonable convergence within the allotted epochs. For the number of hidden units, the allowable range ranged from 10 to an architecture-dependent maximum, as dictated by the guideline of Bardeen et al. [\[50\]](#). This heuristic recommends a minimum of 50 training examples per learnable parameter to promote sufficient coverage during learning and reduce the risk of overfitting. Because different architectures (e.g., LSTM versus feed-forward neural networks) possess distinct parameter counts owing to their differing structural designs, the upper bound on hidden units was adjusted accordingly for each model variant. By thoroughly probing these defined ranges, Bayesian

optimization secured strong, reliable performance for LOGCNN while respecting both the embedded physical principles and the constraints imposed by available data.

Results and Discussion

Hyperparameters optimization and training progress monitoring

The hyperparameter search trajectory depicted in **Figure 7** shows considerable variability in the estimated minimum RMSE, with occasional sharp increases reaching 3.0. Notably, these pronounced spikes frequently preceded substantial subsequent drops in the observed minimum RMSE, which fell to levels approaching 0.5 at multiple points (for instance, around sample indices 13 and 37). Such behavior underscores LOGCNN's ability to capitalize on favorable hyperparameter settings rapidly and highlights the structural benefits it offers for challenging energy-forecasting problems. Close correspondence between the estimated and observed minimum RMSE at these key points further confirms the model's architectural soundness, providing strong evidence of its ability to deliver dependable energy-efficiency predictions across diverse training regimes. Achieving approximately a 25% reduction in training duration while incurring only a modest RMSE penalty of 0.013 demonstrates the framework's computational efficiency without a meaningful loss of forecast quality. This favorable trade-off is especially valuable in practical settings that require rapid model deployment and periodic retraining in response to shifting process conditions or external factors. The small accompanying increase in RMSE represents an acceptable compromise to achieve markedly improved scalability and real-time adaptability, thereby affirming LOGCNN's suitability for energy-management applications in which prompt, actionable insights take precedence over absolute precision.

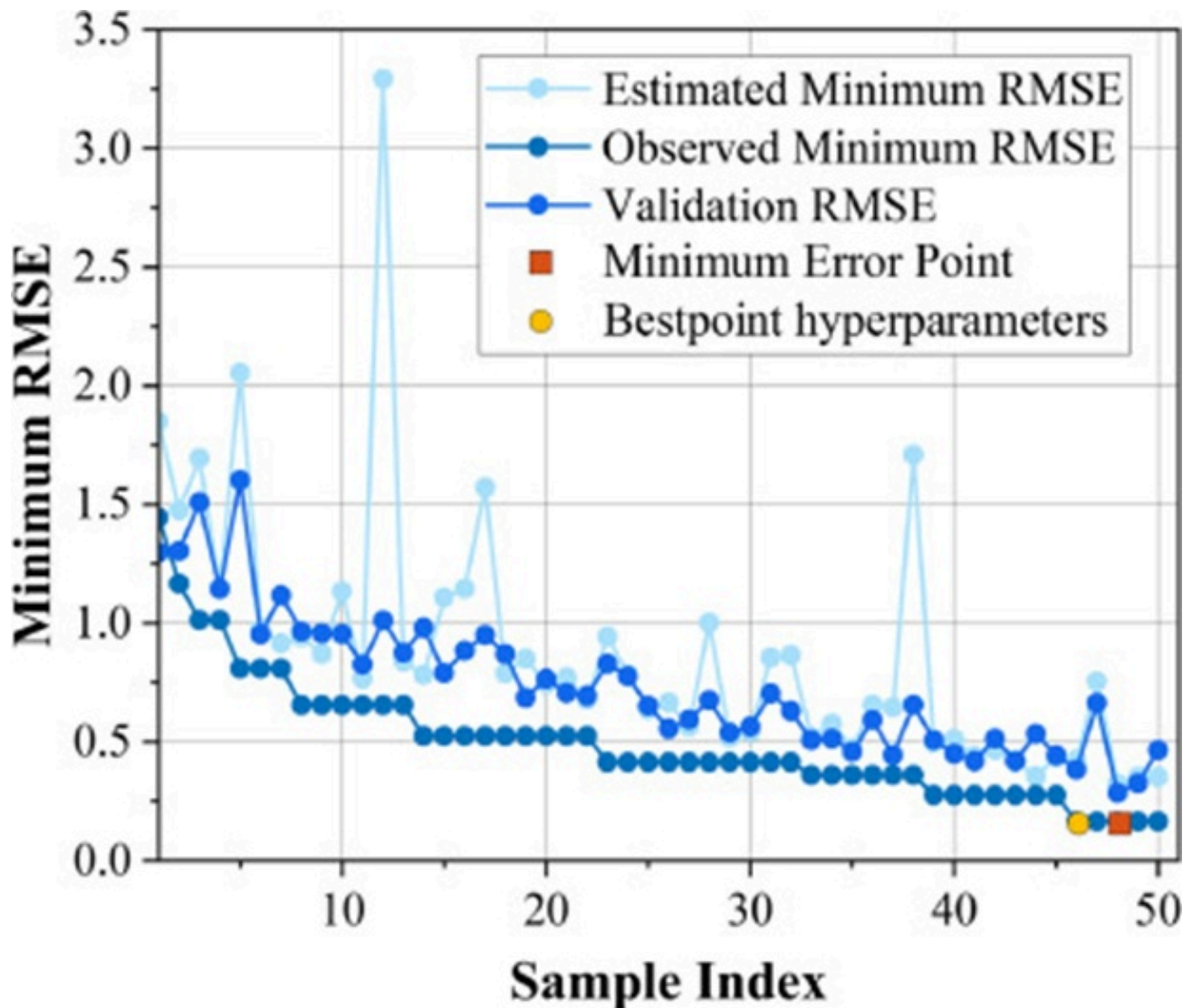


Figure 7. Hyperparameter optimization and efficiency trade-offs in LOCGNN for energy prediction.

Two primary aspects merit attention when assessing the training dynamics under the optimal hyperparameter set: (i) the separation between the training and validation RMSE trajectories, and (ii) the terminal value of the training RMSE. The vertical distance between the training and validation RMSE curves serves as an indicator of generalization quality—a narrow gap points to effective generalization, whereas a wide separation signals possible overfitting, where strong training-set performance fails to translate to unseen validation data. Meanwhile, the final training RMSE value reflects the extent to which the model has succeeded in minimizing errors on the training examples by the conclusion of the optimization process.

Training evolution, tracked via the aggregated loss across individual output channels, is shown in **Figure 8**. The curves demonstrate a steady, consistent decline in RMSE across all outputs, with pronounced gains visible by iteration 700. Throughout the process, the separation between training and validation RMSE remains modest, exhibiting no clear evidence of either overfitting or underfitting. The end-of-training RMSE values for each output, as displayed, confirm that the model has successfully driven down prediction errors across the full set of targets. By continuously enforcing conservation-law consistency during gradient-based updates, the network embeds physically meaningful inductive biases that foster stable, generalizable learning.

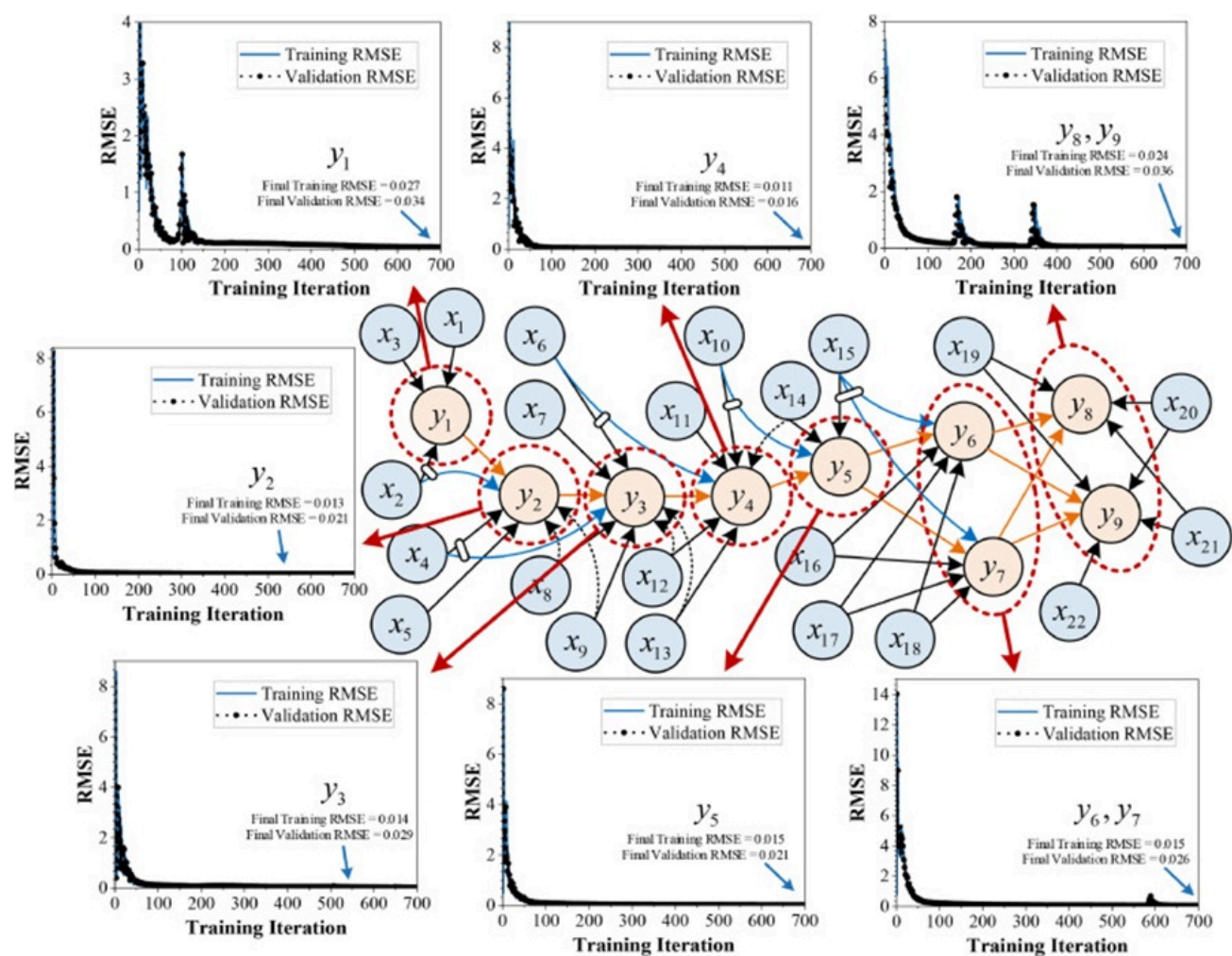


Figure 8. Training progress and loss convergence across outputs of the LOCGNN model.

Energy efficiency prediction result

This investigation evaluates predictive performance across ten distinct models: feed-forward neural network (FFNN), one-dimensional convolutional neural network (1DCNN), temporal convolutional network (TCN), nonlinear autoregressive exogenous model (NARX), long short-term memory (LSTM), bidirectional LSTM (BiLSTM), convolutional LSTM (CNN-LSTM), attention-augmented LSTM (AM-LSTM), time-series transformer (TST), and the proposed law of conservation-guided neural network (LOCGNN). **Table 3** summarizes the principal architectural characteristics, key strengths, and typical application domains of each approach.

Table 3. Architectural features, strengths, and use cases of the compared models.

Model	Strengths	Application
FFNN	Straightforward structure, fast and resource-efficient computation.	Regression tasks, analysis of tabular data
1DCNN	Efficiently captures local patterns, well-suited for short-range dependencies [51].	Signal processing, extraction of spatial characteristics

TCN	Effectively models long-range dependencies while maintaining convolutional speed. Avoids sequential computation limitations [52].	Time-series forecasting
NARX	Directly incorporates lagged inputs and exogenous variables [53].	Nonlinear time series prediction
LSTM	Strong at modeling long-term dependencies thanks to powerful gating mechanisms for sequential data.	Time-series forecasting
BiLSTM	Analyzes sequences in both forward and backward directions [54].	Sentiment analysis, Time-series forecasting
CNN-LSTM	Merges convolutional feature learning with recurrent temporal processing, ideal for spatio-temporal patterns [55].	Multi-dimensional sequence data
AM-LSTM	The attention mechanism emphasizes the most informative portions of the sequence [56].	Time-series forecasting
TST	Leverages self-attention to model global/long-distance relationships, highly effective for extended sequences [57].	Time-series forecasting, natural language processing

Simpler architectures such as FFNN and 1DCNN offer notable computational advantages. FFNN performs particularly well when processing static or non-sequential data, while 1DCNN efficiently extracts short-range patterns from time series. Nevertheless, neither model is equipped to model extended temporal dependencies, which restricts their suitability for intricate sequential prediction problems. TCN partially mitigates this shortcoming by using dilated convolutions, enabling efficient capture of long-range dependencies without the sequential processing bottleneck typical of recurrent networks. Its strictly causal formulation, however, prevents exploitation of future context. NARX explicitly accounts for lagged endogenous variables and exogenous inputs within a nonlinear autoregressive framework, making it well-suited to certain forecasting scenarios. However, it exhibits reduced flexibility when confronted with highly nonlinear or multifaceted dynamics.

More sophisticated architectures—CNN-LSTM and AM-LSTM—extend the foundational LSTM design to address additional challenges. CNN-LSTM integrates convolutional layers for spatial feature learning with recurrent units for temporal modeling. In contrast, AM-LSTM incorporates attention mechanisms that dynamically assign importance to different portions of the input sequence [58]. TST represents a further advancement by relying entirely on self-attention, which facilitates effective modeling of long-range and global dependencies in sequence transduction tasks [59]. These advanced models, however, demand substantially greater computational resources and typically require large volumes of training data to reach peak performance.

Comparative results for energy efficiency forecasting, assessed via three standard statistical metrics, are compiled in [Table 4](#).

Table 4. Comparative performance of a deep learning model on energy efficiency prediction.

Indicator	Model	Training	Validation	Testing
R ²	FFNN	0.861	0.857	0.861
	1DCNN	0.941	0.879	0.859
	TCN	0.952	0.951	0.954
	NARX	0.968	0.931	0.891
	LSTM	0.959	0.894	0.883

	BiLSTM	0.964	0.961	0.963
	CNN-LSTM	0.861	0.696	0.669
	AM-LSTM	0.976	0.978	0.975
	TST	0.987	0.979	0.980
	LOGCNN	0.993	0.989	0.991
RMSE	FFNN	47.962	51.159	46.972
	1DCNN	30.079	46.319	42.902
	TCN	28.317	34.165	33.362
	NARX	25.755	37.892	42.498
	LSTM	26.003	44.788	43.659
	BiLSTM	24.878	32.301	32.057
	CNN-LSTM	47.741	78.881	75.891
	AM-LSTM	23.878	29.133	30.317
	TST	22.503	28.946	29.592
	LOGCNN	21.753	27.083	27.997
MAPE	FFNN	0.395	0.566	0.539
	1DCNN	0.377	0.563	0.541
	TCN	0.308	0.091	0.091
	NARX	0.079	0.109	0.132
	LSTM	0.089	0.146	0.137
	BiLSTM	0.083	0.081	0.084
	CNN-LSTM	0.396	0.592	0.575
	AM-LSTM	0.071	0.064	0.123
	TST	0.058	0.062	0.073
	LOGCNN	0.051	0.053	0.066

Regarding the coefficient of determination (R^2), the top-performing models are AM-LSTM, TST, and LOGCNN. AM-LSTM records R^2 scores of 0.976 (training), 0.978 (validation), and 0.975 (testing). TST improves upon these figures, attaining 0.987 (training), 0.979 (validation), and 0.980 (testing). LOGCNN achieves the highest scores overall, with R^2 values of 0.993 (training), 0.989 (validation), and 0.991 (testing). These outcomes demonstrate that explicitly embedding physical conservation principles into the network architecture yields superior predictive fidelity compared with purely data-driven counterparts.

The same ranking emerges when evaluating root-mean-squared error (RMSE). LOGCNN delivers the lowest errors across all splits: 21.753 (training), 27.083 (validation), and 27.997 (testing). TST follows closely with 22.503, 28.946, and 29.592, while AM-LSTM reports 23.878, 29.133, and 30.317. Other architectures, including BiLSTM and TCN, exhibit

reasonably competitive RMSE values, whereas CNN-LSTM shows markedly larger errors—especially on the held-out test set.

For mean absolute percentage error (MAPE), LOCGNN again leads with the smallest discrepancies: 0.051 (training), 0.053 (validation), and 0.066 (testing). TST ranks second at 0.058, 0.062, and 0.073, followed by AM-LSTM with 0.071, 0.073, and 0.077 (Figure 9).

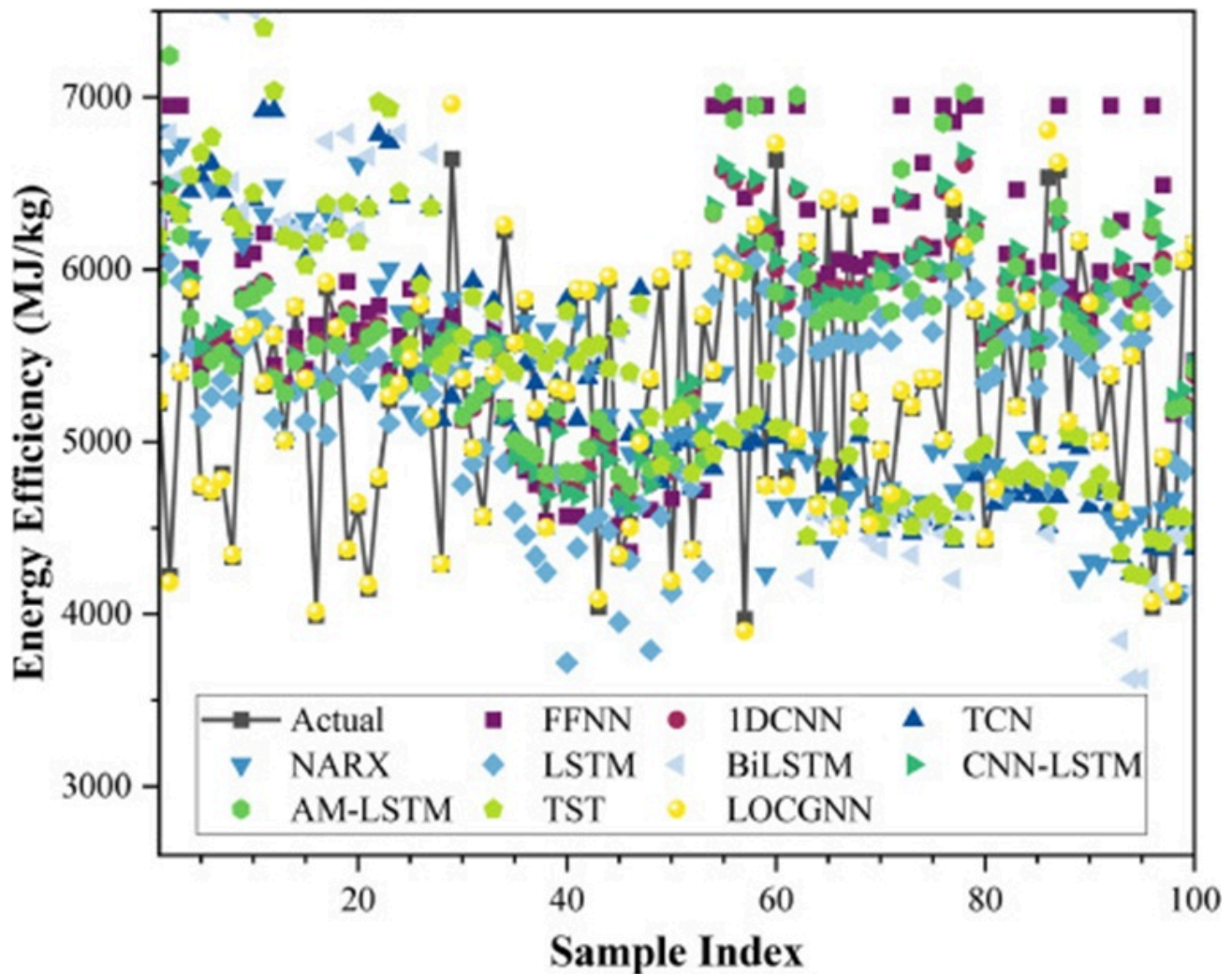


Figure 9. Actual-prediction comparison for energy efficiency prediction across different AI models.

By explicitly incorporating physical laws into the network's design, LOCGNN establishes a more robust modeling framework that leverages established scientific principles to improve both learning efficiency and forecast accuracy. This built-in physical guidance ensures the model remains compliant with fundamental constraints while more cohesively representing the system's overall behavior. Beyond confirming the physical plausibility of its outputs, LOCGNN reduces unrealistic deviations from observed system dynamics by aligning theoretical consistency with empirical observations, thereby lowering the risk of implausible predictions and enhancing generalization to unseen data compared with previously reported methods on the VCM dataset [60].

The fusion of domain-specific physics with neural network flexibility allows LOCGNN to represent the intricate couplings and feedback mechanisms typical of energy systems in a more intuitive manner than conventional purely statistical approaches. In contrast to models such as LSTM, CNN-LSTM, or AM-LSTM—which depend exclusively on statistical

pattern recognition—LOGCNN structurally embeds conservation principles (mass, component, and energy balances), guaranteeing that generated predictions remain consistent with the core physical rules dictating energy and material flows. These embedded constraints serve as an effective regularizer throughout optimization, substantially narrowing the parameter search space. A key practical advantage of LOGCNN lies in its enhanced interpretability: because the model is rooted in physical first principles, its outputs naturally convey meaningful insights into underlying process mechanisms, offering value that extends well beyond raw numerical performance. This property proves especially valuable in real-world engineering contexts where strict adherence to physical laws is non-negotiable.

A granular breakdown of LOGCNN's predictive performance across the nine individual target variables is illustrated in **Figure 10**. The model consistently attains excellent testing R^2 scores exceeding 0.985 for every output, peaking at 0.993 for y_7 and reaching a minimum of 0.986 for y_4 . RMSE values are notably higher for y_1 than for the remaining variables, registering 11.275 (training), 11.206 (validation), and 15.950 (testing).

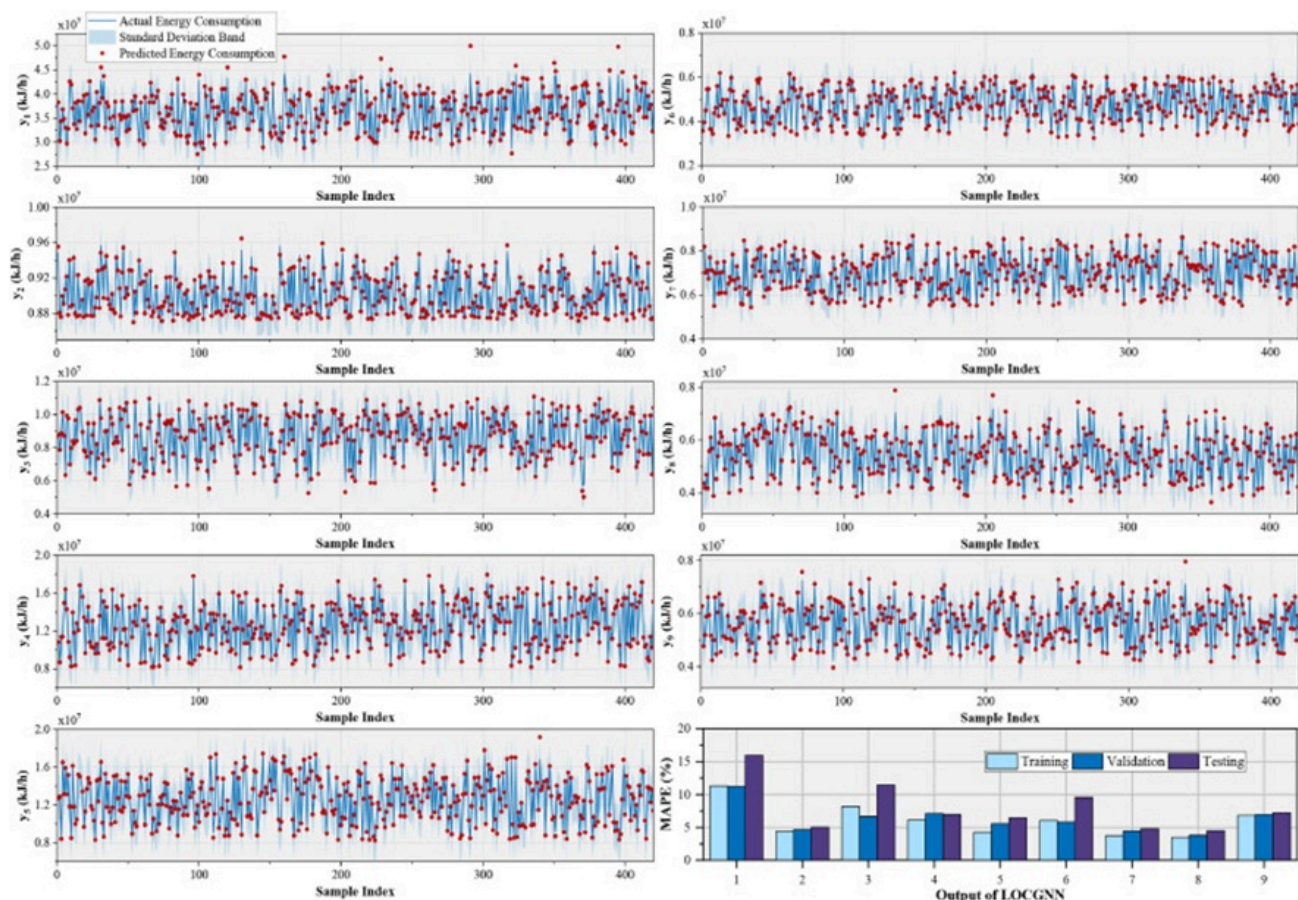


Figure 10. Detailed performance analysis of LOGCNN for energy efficiency prediction.

The comparatively lower accuracy observed for y_1 can be traced to its more limited connectivity with respect to the guiding conservation principles. Specifically, y_1 depends on only three input streams: the mass flow rate of L-401 (feed to the thermal cracker), the temperature of L-403 (cracker outlet temperature), and the temperature of L-401 (cracking coil temperature). This sparse linkage—limited to just three connections—stands in contrast to most other outputs, which typically draw on six to eight related streams. Furthermore, y_1 is directly affected by two independent exogenous streams: the feed stream, which acts as a primary disturbance propagating through the system, and the flue gas associated with the heating medium for the cracking reaction. These independent influences introduce extra sources of variability and complexity, making accurate prediction of y_1 more challenging. The reduced number of informative connections likely

limits the contextual information available to the model during learning, thereby constraining its ability to achieve the same level of precision for y_1 as for the other targets.

Table 5 presents a head-to-head comparison of LOCGNN against LSTM, AM-LSTM, CNN-LSTM, and TST, highlighting the trade-offs among error reduction, R^2 improvement, and computational expense. LOCGNN delivers substantial RMSE reductions of 35.9% relative to LSTM, 8.3% relative to AM-LSTM, and 5.4% relative to TST. Correspondingly, it improves R^2 by 12.2 % over LSTM (from 0.883 to 0.991) and by 1.1% over TST (from 0.980 to 0.991). These gains, however, entail higher training costs: LOCGNN requires 526.94 s, which is 14.5× longer than LSTM (36.28 s), 6.1× longer than AM-LSTM (87.01 s), and 5.5× longer than TST (96.09 s). Although CNN-LSTM trains faster (59.52 s), its markedly worse accuracy renders it less competitive overall. All timing measurements were conducted on a workstation equipped with an Intel® Core™ i7-3770 CPU, 32.0 GB RAM, Windows 10 Pro (64-bit), and an NVIDIA GeForce RTX 4060 GPU. These results underscore the inherent computational–performance trade-off in LOCGNN and emphasize the need to weigh increased resource demands against the accuracy gains achieved when selecting an appropriate model for specific use cases.

Table 5. Training time comparison for all energy efficiency models.

Model	Training duration (seconds)	Inference speed (milliseconds per sample)
FFNN	26.69	0.047
1DCNN	41.89	0.177
TCN	83.43	0.309
NARX	358.64	2.227
LSTM	36.28	0.266
BiLSTM	52.82	1.906
CNN-LSTM	59.52	0.315
AM-LSTM	87.01	0.351
TST	96.09	0.605
LOCGNN	526.94	0.532

Prediction trajectory and model simplification of LOCGNN

The predictive pathways and internal connection structure of LOCGNN are visualized in **Figure 11**. Color-coded links illustrate distinct physical relationships derived from the governing mass and energy conservation principles that dictate system-wide interactions. Blue links indicate direct influences from input variables to energy-related outputs via the energy balance equations. For example, inputs x_1 , x_2 , and x_3 exert substantial effects on y_1 , contributing approximately 33.64%, 30.57%, and 35.77%, respectively. Increasing the feed EDC mass flow rate could enhance the energy efficiency of furnace F-401 by up to 33.64% through more efficient utilization; however, exceeding the furnace's optimal capacity would introduce 33.64% inefficiency due to overuse.

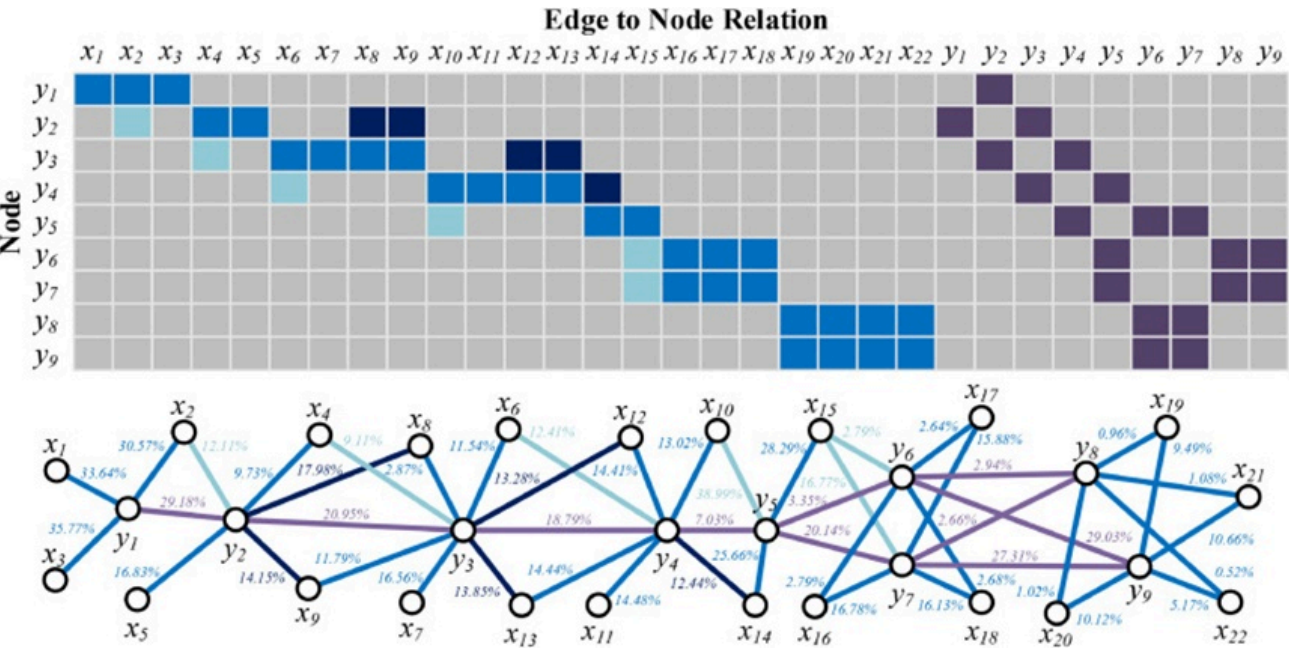


Figure 11. LOCGNN connection dynamics and prediction trajectories.

Dark blue connections highlight the influence of delayed (lagged) input variables on the energy efficiency of upstream blocks. A representative case involves the recycle stream L-413: elevated residual heat in L-413 necessitates greater cooling effort in the quench section to offset the additional enthalpy carried forward, thereby maintaining balance across the system. Collectively, the lagged contributions from x_8 and x_9 (enthalpy transfer from block 3 to block 2) account for roughly 32.04% of the utility demand variation. This mechanism helps stabilize enthalpy flows throughout the process, preventing adverse impacts on the energy efficiency of exchangers E-401 and D-401 caused by temperature swings in the recycled stream.

Light blue lines depict the linkages between selected input variables and energy efficiency arising from mass conservation principles, thereby coupling the behavior of distinct subsystems. For instance, blocks 3 and 4 are interconnected via streams L-412 and L-415. Variations in the energy efficiency of E-403 (labeled y_4) are strongly modulated by changes occurring in D-402, a unit within block 3. When y_3 rises, approximately 12.41% of that increment can be traced to mass flow adjustments in D-402 rather than to internal control actions within block 4. These light blue pathways play a vital role in sustaining and continuously improving energy efficiency in downstream equipment by ensuring proper mass continuity and effective resource allocation throughout the process. Finally, purple connections highlight how output variables from an upstream block propagate to influence a downstream block, representing internal feedback mechanisms that shape subsequent energy performance.

Further examination of these interactions prompts an important question: What would be the consequences of deliberately severing or modifying selected connections that embody key relationships dictated by conservation laws? **Figure 12** illustrates three simplified variants of the LOCGNN architecture in which specific linkage types have been intentionally removed (indicated by dark gray regions where connections between input and output nodes are absent): (a) inter-system dynamics, (b) recycle stream effects, and (c) shared mass-balance inputs across connected blocks. These ablation studies reveal the differential impact of each simplification on global model accuracy and energy-efficiency forecasting.

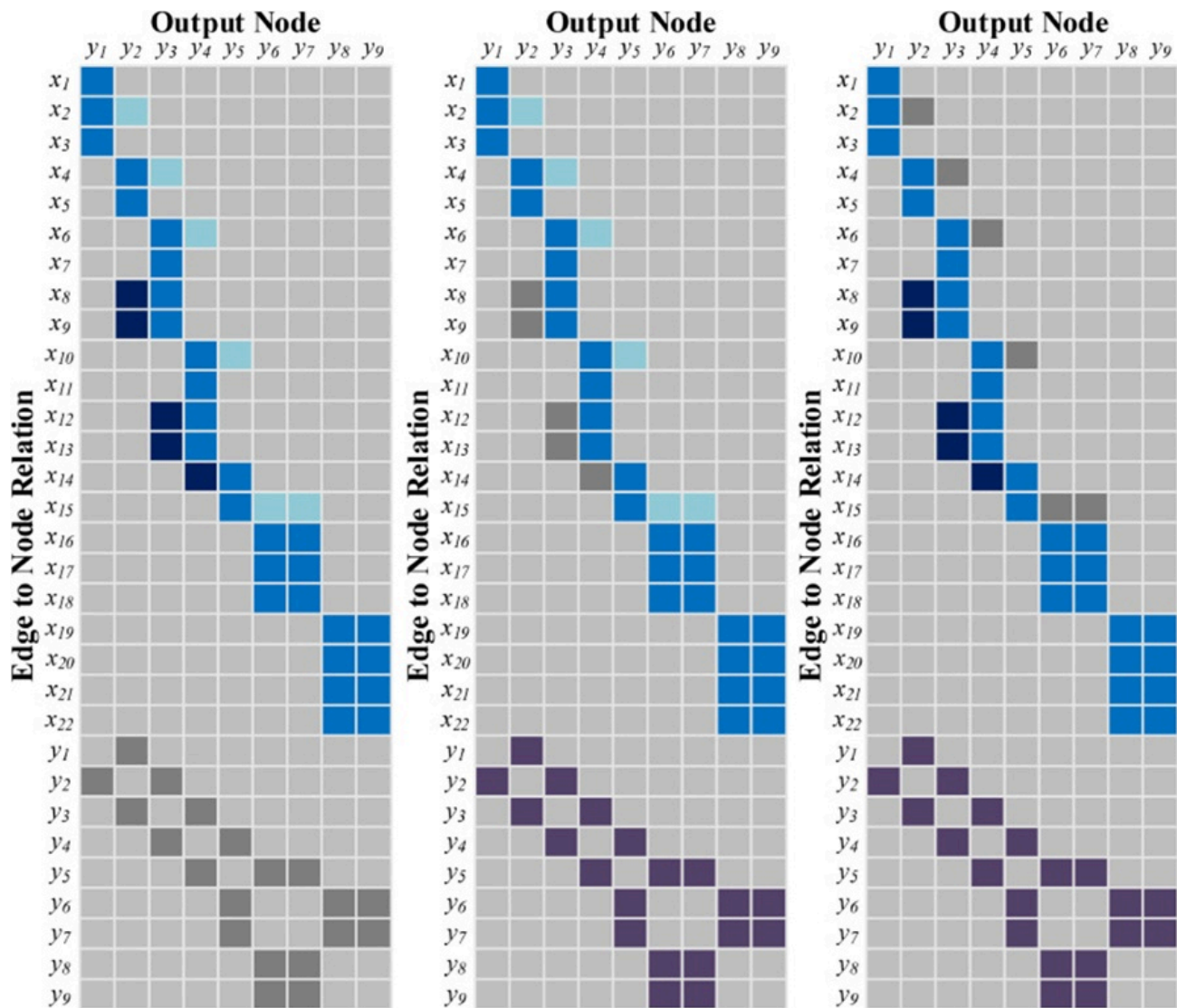


Figure 12. Simplified LOCGNN network by removing a) step 3 (dynamics between systems), b) step 4 (recycled effect), and c) step 5 (mass of connected stream) in model formulation.

In case (a), eliminating inter-system dynamic coupling markedly degrades predictive capability because critical cross-unit influences on energy efficiency are no longer accounted for. Overall testing R^2 falls from 0.991 to 0.979, with the largest deteriorations observed for y_6 (0.974) and y_8 (0.972). The corresponding box plot exhibits a broad interquartile range and short whiskers. While the removal does not uniformly impair individual block-level predictions, it substantially weakens the model's holistic understanding of system-wide energy behavior.

Case (b) demonstrates that disregarding recycle streams impairs the model's ability to capture energy reutilization, leading to reduced forecasting reliability. This configuration produces the widest whiskers in the box plot, indicating pronounced performance instability and underscoring the critical role of recycle flows in connected blocks. Outputs y_2 and y_3 suffer the most severe declines, reaching testing R^2 values of 0.967 and 0.951, respectively. Nevertheless, the overall drop in performance remains moderate—from 0.991 to 0.981—indicating that recycle effects, while important, exert a less pervasive influence than direct inter-system dynamics.

In case (c), omitting shared mass-balance inputs across linked blocks compromises mass continuity enforcement but results in comparatively milder degradation. Testing R^2 ranges from 0.991 to 0.976, indicating that although shared

variables support global energy accounting, they are less decisive than either inter-system couplings or recycling mechanisms. Consequently, the model retains relatively strong accuracy even without these connections.

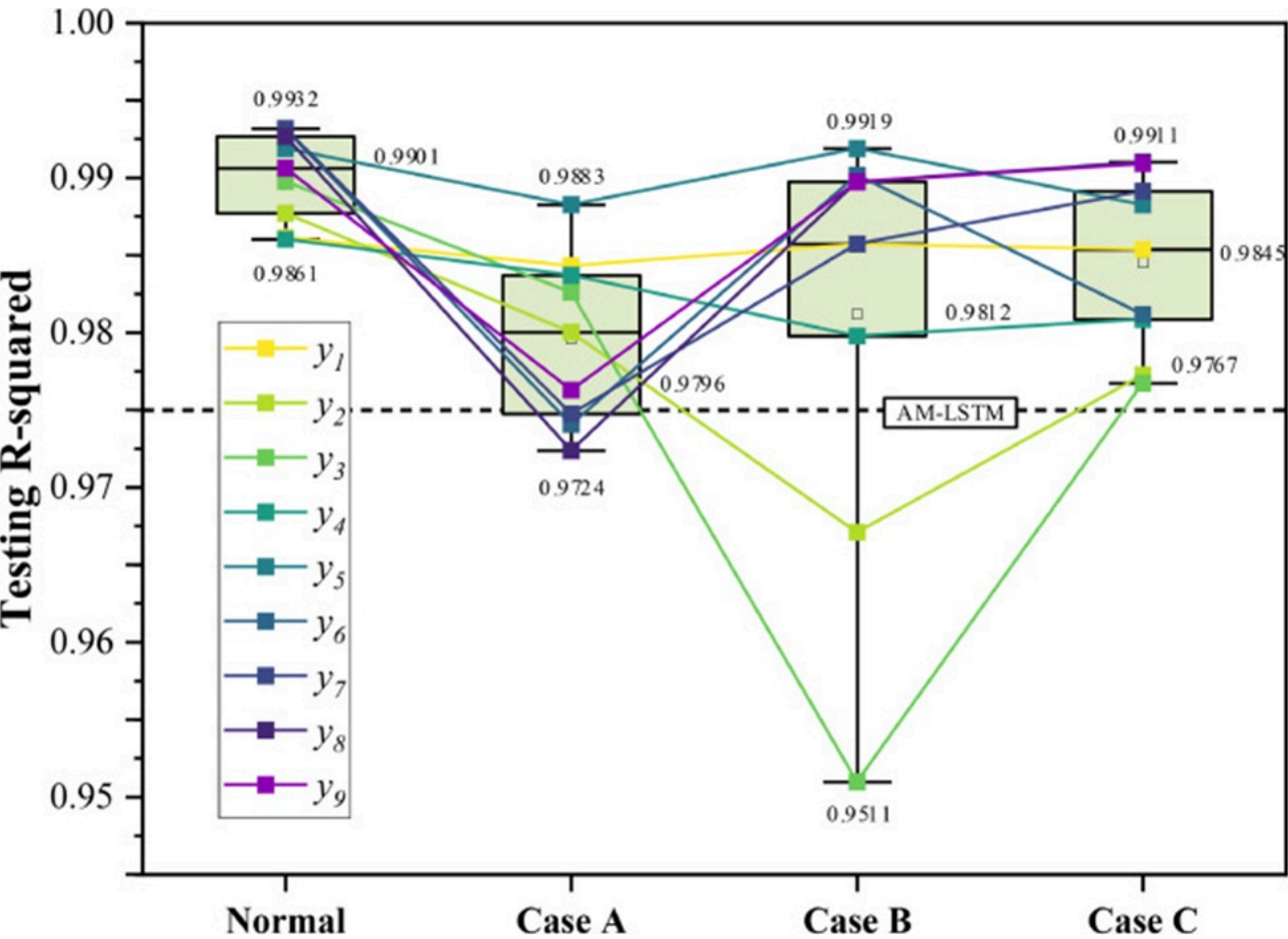


Figure 13. Impact of simplification cases on testing R² for energy efficiency prediction.

The testing R² performance across outputs y_1 through y_9 under the original configuration and the three simplification scenarios is summarized in Figure 13.

Energy efficiency optimization result

The inherent structure of the LOCGNN framework facilitates optimization of plant-wide energy efficiency by revealing how energy expenditure is distributed across different sections and blocks for a given production rate. Representative optimization outcomes obtained with LOCGNN are shown in Figure 14. At the baseline production rate of 12,683 kg/h, the process consumes a total of 124,112 GJ/h, allocated between the EDC cracking and VCM purification sections.

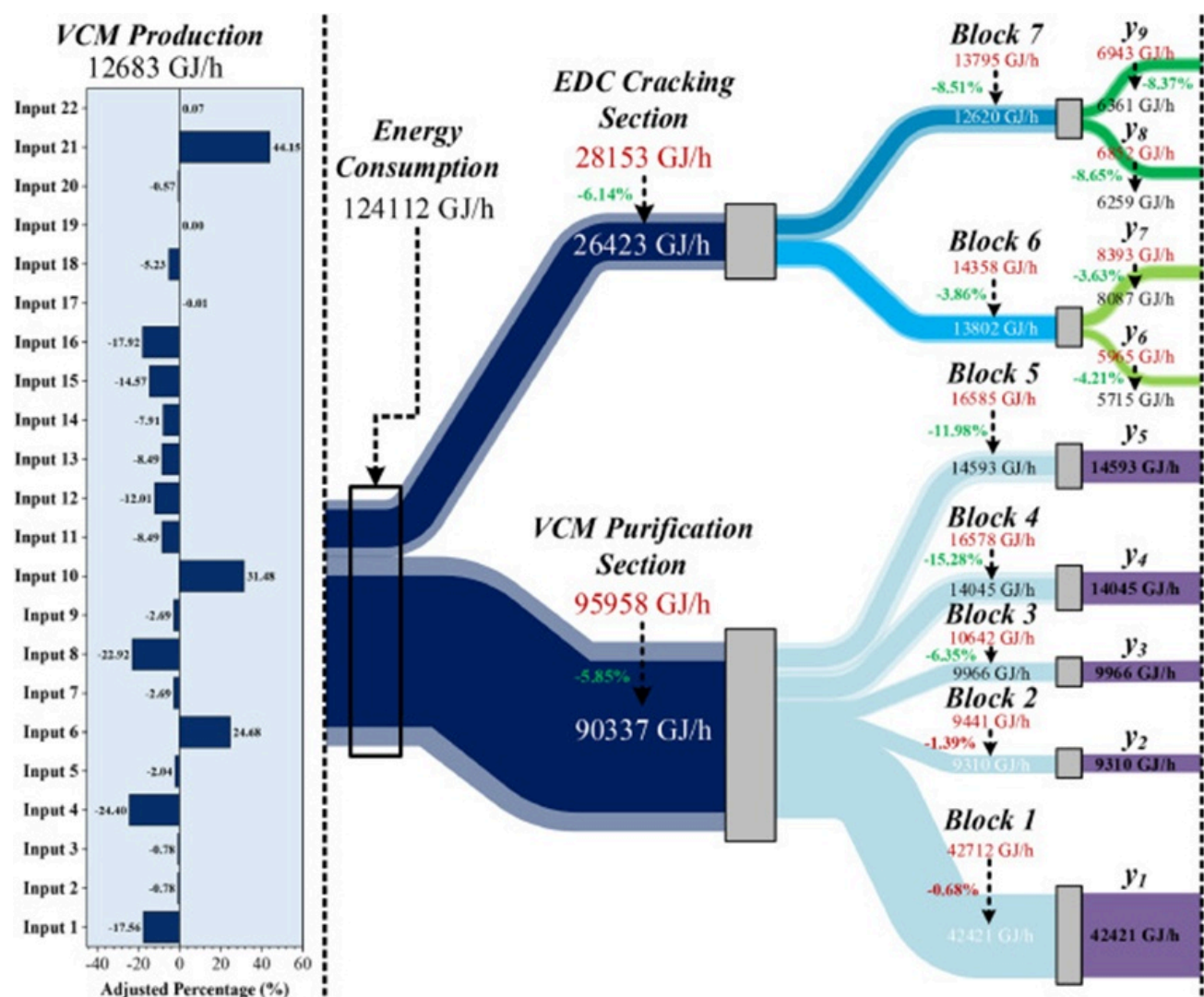


Figure 14. Energy consumption, distribution, and optimization adjustments in VCM production.

The EDC cracking section accounts for 28,153 GJ/h, of which 14,358 GJ/h is transferred to block 6 and 13,795 GJ/h to block 7. Concurrently, the VCM purification section consumes 95,958 GJ/h and distributes energy downstream as follows: 16,585 GJ/h to block 5, 16,578 GJ/h to block 4, 10,642 GJ/h to block 3, 9,441 GJ/h to block 2, and 42,712 GJ/h to block 1. These forward-propagated energy flows indicate that the existing operating point is suboptimal.

To identify improvement opportunities, sensitivity analysis was performed using LOGCNN to pinpoint the most influential input variables affecting specific energy consumption (SEC) and to determine optimal operating interactions that minimize total energy use. The optimization task was formulated as the minimization of aggregate energy consumption subject to a fixed production target of 12,683 kg/h, with LOGCNN serving as the predictive surrogate model, as expressed in [Eq. 38](#).

$$\begin{aligned}
 &\underset{x}{\text{Minimize}} : \sum_{m=1}^9 y_m \\
 &\text{Subject to} : y_m = \text{LOGGNN}(x_1, x_2, \dots, x_{22})_m, \forall m = 1, 2, \dots, 9 \\
 &x_{19} = x_{19}^{\text{target}} \\
 &x_n^{\min} \leq x_n \leq x_n^{\max}, \forall n = 1, 2, \dots, 22 \\
 &h_p(x_1, x_2, \dots, x_{22}) \leq 0, \forall p = 1, 2, \dots, q
 \end{aligned} \tag{38}$$

Where y_m denotes the energy consumption associated with each output variable, the complete LOGGNN predictor, written as $\text{LOGGNN}(x_1, x_2, \dots, x_{22})_m$, estimates energy consumption y_m from the input vector comprising x_1 through x_{22} . Decision variables x_n ($n = 1, 2, \dots, 22$) that affect energy use are constrained to lie within their historically observed operating ranges $[x_n^{\min}, x_n^{\max}]$. Additional operational restrictions, denoted h_p , encompass utility limits and equipment capacity bounds.

The sensitivity study highlighted the process variables that exert the greatest leverage over specific energy consumption (SEC). For the fixed throughput of 12,683 kg/h, the present operating regime departs noticeably from the energy-optimal point because of non-ideal values assigned to x_1 (EDC feed mass flow rate), x_4 (distillate flow from D-401 overhead), x_6 (distillate flow from D-402 overhead), x_8 (recycle mass flow returning from D-402), x_{10} (distillate flow from D-403 overhead), x_{12} (recycle mass flow returning from D-403), x_{15} (feed rate to the HCl column via L-501), x_{16} (distilled HCl mass flow rate), and x_{21} (recycled EDC flow via L-516). Coordinated modifications to these nine variables can profoundly reshape the facility's energy profile, delivering substantial gains in both overall energy productivity and process throughput for vinyl chloride monomer production.

Examination of furnace F-401 using LOGGNN indicates it is currently running overloaded. Furnaces designed for thermal cracking achieve the highest efficiency only within a well-defined feed-rate window; feeding beyond this limit increases fuel consumption to sustain reaction temperatures, lowers cracking selectivity, and increases pollutant emissions. A targeted cut of -17.56% in the EDC feed mass flow rate relieves furnace overload, curbs wasteful energy use, and smooths the operation of the downstream quench/condensation train. This upstream change reduces the overhead mass flow from D-401 by 24.40%, thereby lessening the thermal duty on condensers E-401 and E-402 by 1.39% and 6.35%, respectively.

To offset the reduced liquid traffic from D-402 and D-403, the recycle mass flows in L-413 and L-419 are reduced by 22.92% and 12.01%, respectively. These reductions improve net forward flow through the condensers toward the columns and diminish the parasitic energy penalty associated with repeated heating and cooling cycles in the recycle loops. As a result, energy efficiency rises by 15.28 % in E-403 and by 11.98 % in E-404. The leaner feed to the HCl rectification column also trims recycled HCl returning to oxychlorination by roughly 17.92%, yielding a 4.21% efficiency gain in that section. Finally, deliberately increasing feed to the stripping section of the second distillation column markedly enhances VCM separation performance, resulting in a 44.15% efficiency uplift in the tower and an 8.51% reduction in its energy requirement.

Emission factors and 100-year global warming potentials reported in the IPCC Sixth Assessment Report (AR6) [61] and by the U.S. Environmental Protection Agency [62] are as follows: CO_2 — 50.29 kg/TJ (GWP = 1), CH_4 — 0.95 kg/TJ (GWP = 29.8), N_2O — 0.094 kg/TJ (GWP = 273). The cumulative energy savings equate to an estimated monthly abatement of 377.35 tons of CO_2 -equivalent greenhouse gases. Valued at the Henry Hub natural gas benchmark price of USD 2.01/GJ published by the U.S. Energy Information Administration [63], these utility reductions generate monthly operating cost savings of USD 443,265. Adoption of the LOGGNN framework, therefore, simultaneously lowers resource consumption, unlocks potential revenue from carbon credits, and shrinks production expenses, advancing the facility toward genuinely sustainable performance.

Model Verification on the Practical Application

To substantiate the generalizability and dependability of LOCGNN beyond the VCM case, the framework was applied to an independent industrial dataset from the slurry-drying section of a commercial detergent powder plant [64]. Real-time measurements for this drying process were acquired from diverse instrumentation sources: process-stream thermocouples (slurry temperature), ambient-air sensors near air-preheat units, spray-dryer instrumentation (feed rate and internal pressure), exhaust-air temperature probes, laboratory moisture assays of the finished base powder, and shift logs recorded by operating personnel. The compiled dataset contained 1000 hourly records, which were subsequently allocated to training, validation, and test partitions in the proportions 0.6: 0.2: 0.2.

Detergent powder system description and network formulation

The drying section of the detergent powder facility is conceptually divided into three major functional zones, as outlined in **Figure 15**. Slurry is withdrawn from an upstream surge tank and pumped at high pressure into the spray dryer (block 3) at fluctuating rates between 0 and 150 t/h, depending on instantaneous production demand. The slurry temperature in the feed tank is regulated to 60–75 °C (typical median $\approx$ 70 °C) to maintain favorable atomization and drying characteristics. In parallel, ambient air enters the system through line L-301, receives preliminary heating in a process-to-air exchanger located in block 1, and is then conveyed to the direct-fired air heater (Q-302). Heat is supplied by combusting liquefied natural gas delivered via L-303, with fuel flow modulated between 400 and 750 m³/h to match the dryer's instantaneous moisture-removal requirement and target product moisture specification. The resulting high-temperature air stream is routed into block 3, where it contacts the finely atomized slurry droplets inside the spray dryer, rapidly evaporating water and producing free-flowing detergent powder with residual moisture held between 0 % and 5 %. The dried granules are discharged from the dryer base via line L-204 for subsequent screening, blending, and packaging. Moist exhaust air, laden with evaporated water vapor, exits the top of the drying chamber at 60–120 °C.

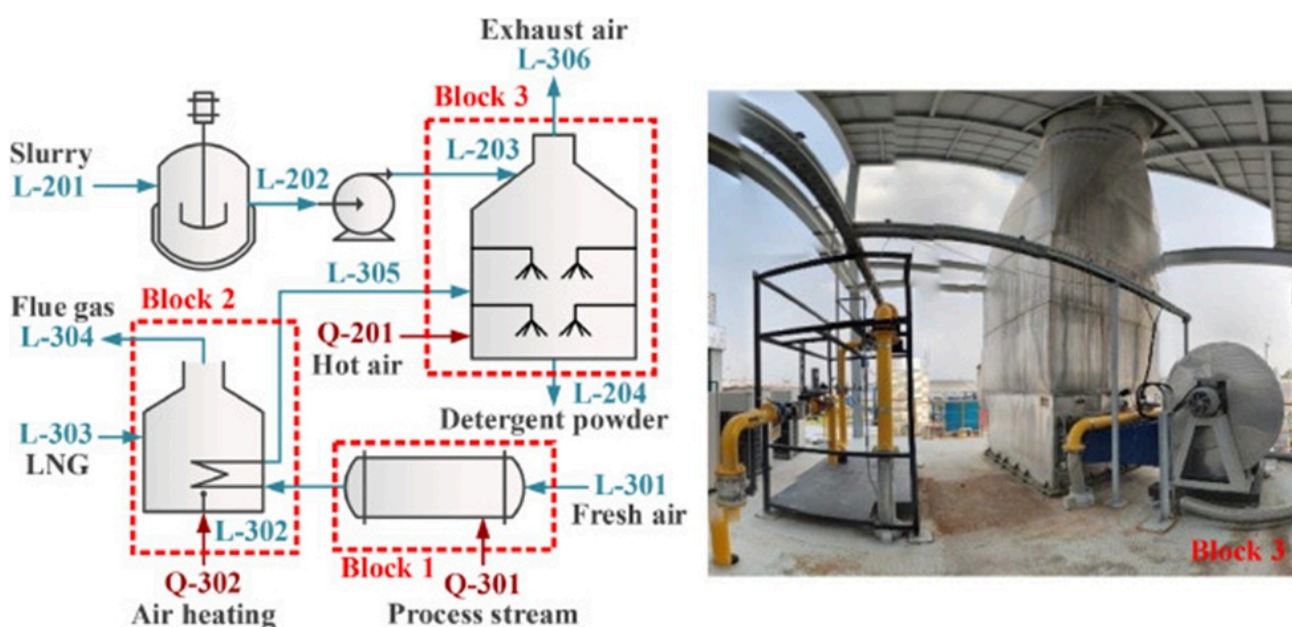


Figure 15. Process flow diagram of drying unit for detergent powder production.

Network training and energy efficiency prediction

The construction of the LOCGNN model for this process begins by partitioning the drying system into three distinct functional zones: the air preheater, the furnace, and the spray dryer. For each zone, open-system steady-state energy balance equations are derived to describe the respective energy inflows, outflows, and internal transformations. To enable modular mapping of inputs to outputs while preserving physical meaning, every subsystem incorporates at least one utility

(heat supply) term—specifically Q-201 for the air preheater, Q-301 for intermediate heating, and Q-302 for the main furnace—as expressed in Eqs. 39–42.

Air-preheater:

$$m_{L301} h_{L301} \left(T_{L301}^{x_1} \right) + Q_{301}^{y_1} = m_{L302} h_{L302} \left(T_{L302}^{x_2} \right) \quad (39)$$

Furnace:

$$m_{L302} h_{L302} \left(T_{L302}^{x_2} \right) + m_{L303} h_{L303} \left(T_{L303}^{x_3} \right) + Q_{302}^{y_2} = m_{L304} h_{L304} \left(T_{L304}^{x_4} \right) + m_{L305} h_{L305} \left(T_{L305}^{x_4} \right) \quad (40)$$

Spray dryer:

$$m_{L203} h_{L203} \left(T_{L203}^{x_5} \right) + m_{L305} h_{L305} \left(T_{L305}^{x_4} \right) + Q_{201}^{y_3} = m_{L204} h_{L204} \left(T_{L204}^{x_7} \right) + m_{L306} h_{L306} \left(T_{L306}^{x_8} \right) \quad (41)$$

$$m_{p,L306} = m_{L306} \left(1 - y_{H_2O}^{x_9} \right) \quad (42)$$

Once the individual energy balance expressions are established, inter-block couplings are introduced by identifying shared variables that connect adjacent zones. In this flowsheet, two key temperatures serve this role: the outlet temperature of L-302 (bridging blocks 1 and 2) and that of L-305 (bridging blocks 2 and 3). Explicitly embedding these linking variables into the LOCGNN topology produces the complete network architecture illustrated in Figure 16b, setting the stage for subsequent parameter training.

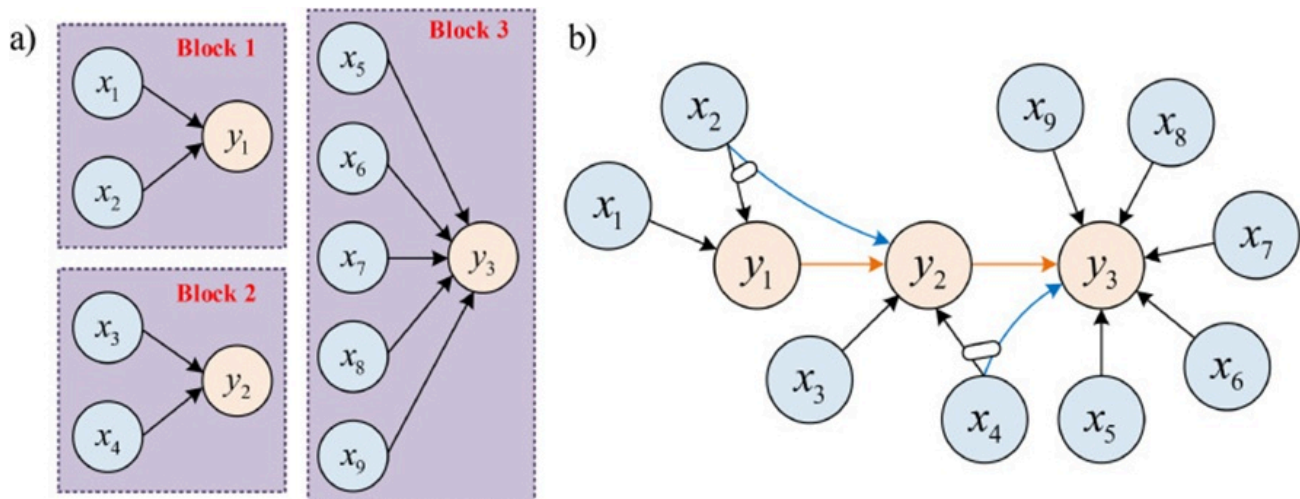


Figure 16. Final architecture of the LOCGNN for a case study of detergent powder production.

Table 6 presents a comparative evaluation of energy efficiency prediction performance in the detergent powder industry case study, comparing traditional methods with the digital twin-assisted transfer learning LSTM (TL-LSTM) [65]. These outcomes align closely with those observed in the VCM production case study, where the LOCGNN model consistently achieves the best results across all metrics and dataset splits. The LOCGNN model records the highest R^2 values of all approaches, reaching 0.9823 in training, 0.9743 in validation, and 0.9741 in testing. Regarding RMSE, which quantifies the average size of prediction errors, the LOCGNN model once more surpasses the alternatives, delivering the smallest errors: 0.0312 in training, 0.0364 in validation, and 0.0371 in testing. When evaluated in percentage-based error terms, the LOCGNN model exhibits exceptional accuracy, with MAPE scores of 0.6896 during training, 0.4631 during validation, and 0.4411 during testing, respectively.

Table 6. Model verification results on energy efficiency prediction.

Metric	Model	Training	Validation	Testing
R^2	FFNN	0.7438	0.7439	0.7521
	CNN	0.7502	0.7397	0.7214
	GRU	0.9005	0.8348	0.8439
	LSTM	0.9372	0.8234	0.8513
	TL-LSTM	0.9518	0.9413	0.9169
	LOGGNN	0.9823	0.9743	0.9741
RMSE	FFNN	0.1091	0.0943	0.0901
	CNN	0.1062	0.0968	0.0974
	GRU	0.0735	0.0881	0.0681
	LSTM	0.0596	0.0851	0.0664
	TL-LSTM	0.0508	0.0619	0.0507
	LOGGNN	0.0312	0.0364	0.0371
MAPE	FFNN	10.3833	8.1004	8.6902
	CNN	10.3079	8.4263	8.6254
	GRU	6.4661	7.2767	7.8551
	LSTM	5.2651	6.4973	7.2982
	TL-LSTM	4.3024	4.4204	6.7101
	LOGGNN	0.6896	0.4631	0.4411

Figure 17 compares the actual and predicted values for the three outputs of the LOCGNN model. The primary objective of this figure is to display the real versus forecasted energy efficiency, while highlighting three key observations: robustness to noise, absence of overfitting, and lack of heteroskedasticity. As demonstrated by the predictions for output 2, the model exhibits strong resilience against noise. It successfully captures sudden shifts that represent dynamic process transitions. Furthermore, the performance difference between training and testing datasets for LOCGNN is only 0.2485%, with the testing result surpassing the training one, indicating excellent generalization and no overfitting to the training data. Most notably, the residual plot reveals that errors remain consistently small and evenly distributed across

the time axis. This uniform and stable error variance confirms the effectiveness of the proposed model, with no evidence of heteroskedasticity in the predictions.

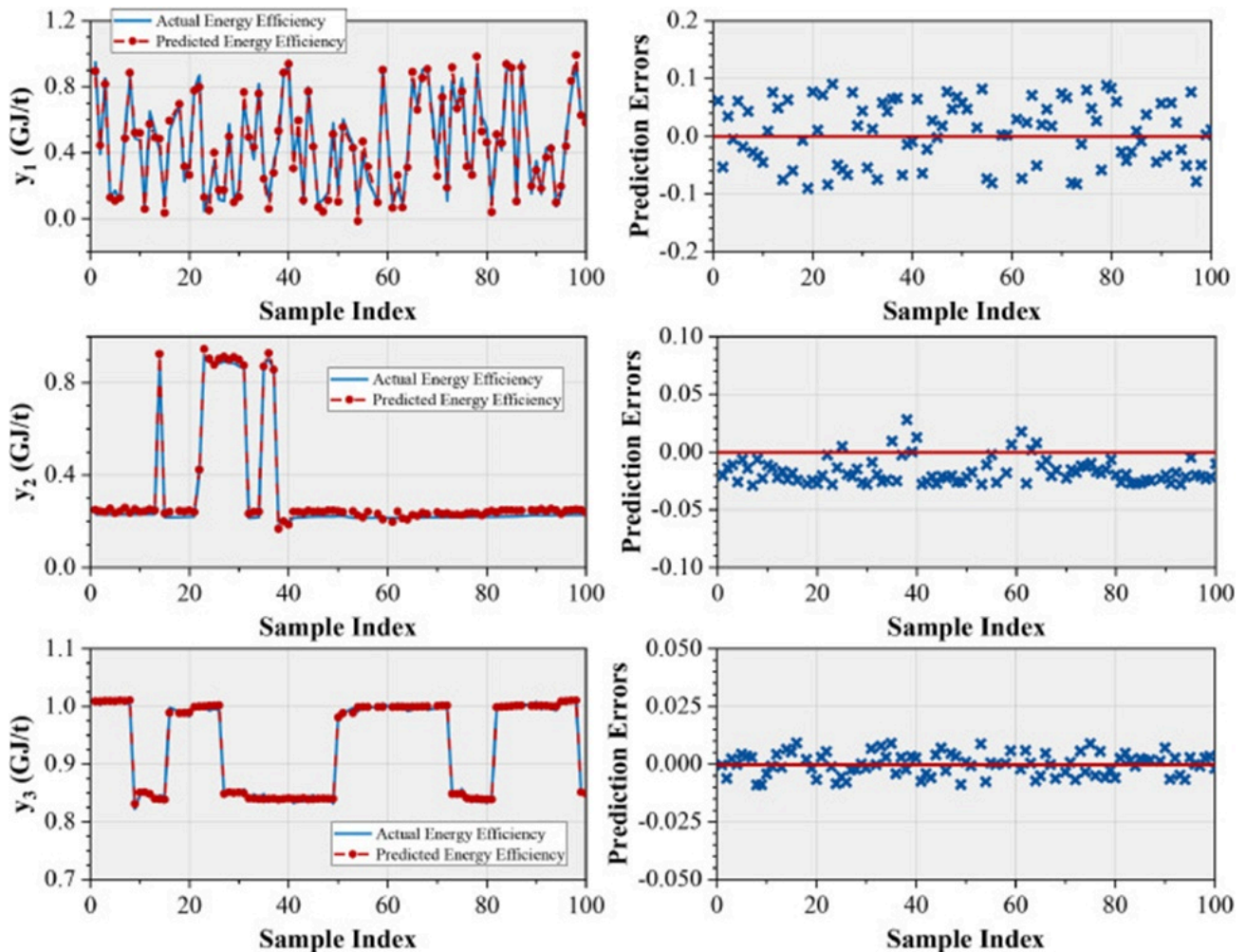


Figure 17. Actual-predicted energy efficiency comparisons for the detergent powder industry.

Conclusion

This study presents an original architecture for a conservation-law-constrained neural network (LOGCNN) that leverages gradient aggregation to improve the accuracy of energy-efficiency forecasting in industrial contexts, with long short-term memory units serving as the intermediate layer. By integrating the essential rules of physical conservation directly into the network's design, LOGCNN achieves noticeably better prediction precision and stability than competing modification strategies, including attention-based methods and transfer learning techniques. The approach also improves insight into the interactions among process variables across complex systems and supports greater explainability in AI-driven models. The method's validity was confirmed through two real-world case studies: vinyl chloride monomer (VCM) production and detergent powder manufacturing. The key findings are summarized below:

1. LOGCNN clearly outperformed standard deep learning architectures and other enhancement techniques (attention mechanisms and transfer learning), delivering superior forecasting performance. This is confirmed by R^2 scores of 0.993 (training), 0.989 (validation), and 0.991 (testing) in the VCM application.

2. The energy-flow representation generated by LOCGNN enables targeted operational corrections to prevent sudden energy surges, yielding a realistic energy-efficiency improvement of 4.21%. This gain translates into avoiding 377.35 tons of carbon emissions and saving USD 443,265 in operational expenses by reducing wasteful energy consumption.

3. Application to the detergent powder production process further validated the model's versatility and reliability in industrial settings, yielding a test-set R^2 of 0.9741 and a low average prediction error of 0.5% even with limited historical data.

4. Experiments on model simplification indicated that severing the edge-to-node dependency noticeably weakened performance, especially when the most influential relation types (in descending order) were dynamic energy flows, recycled stream linkages, and mass-conservation-based shared variables. The corresponding average test R^2 values dropped to 0.9796, 0.9812, and 0.9845, respectively.

The LOCGNN model demonstrated robust predictive performance and achieved substantial gains in energy efficiency optimization by embedding conservation laws directly into its network architecture. Nevertheless, several important limitations should be considered when deploying the model in real-world industrial settings.

A major constraint concerns data quality: although the model demonstrates some tolerance to noise, it requires consistent, complete sampling intervals across all input variables. Any discrepancies in sampling frequency or gaps in the data can compromise prediction performance, making thorough data acquisition and rigorous preprocessing essential.

Furthermore, while the six-step construction procedure was intentionally designed to be generic and support applications across a wide range of industrial systems, the network architecture must be tailored specifically to each process. This customization requires process-dependent adaptations and considerable effort to modify the framework so that it properly reflects the distinctive features of the target system, rendering the overall implementation effort and resource-intensive.

An additional drawback is the comparatively long training time due to the heavy computational burden of gradient computation. Although inference (prediction) remains fast and efficient, the prolonged training phase can pose difficulties in industrial contexts where process dynamics necessitate regular model updates and retraining.

Looking ahead, overcoming these constraints could greatly enhance the model's practicality and usability. Potential improvements include adopting more sophisticated methods for handling incomplete or irregular data to ensure reliable performance, as well as investigating streamlined or accelerated architectural variants that maintain high accuracy while significantly reducing training duration.

Acknowledgements

The author would like to acknowledge the support of the Faculty of Engineering, Kasetsart University (Grant No.66/04/CHEM/D.Eng), the Kasetsart University Research and Development Institute, the Center for Advanced Studies in Industrial Technology, and the Center of Excellence on Petrochemical and Materials Technology.

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 01 Oct 2023

Revised: 08 Nov 2023

Accepted: 17 Dec 2023

Published online: 20 March 2024

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

Ratanakuakangwan S, Morita H. Measuring the efficiency of energy planning under uncertainty. *Energy Rep.* 2022;8:544–51.
<https://doi.org/10.1016/j.egy.2021.11.164>.

Tantisattayakul T, Soontharothai J, Limphitakphong N, Pharino C, Chavalparit O, Kanchanapiya P. Assessment of energy efficiency measures in the petrochemical industry in Thailand. *J Clean Prod* 2016;137:931–41.

Lawrence A, Thollander P, Andrei M, Karlsson M. Specific Energy Consumption/Use (SEC) in energy management for improving energy efficiency in industry: meaning, usage and differences. *Energies.* 2019;12:247.
<https://doi.org/10.3390/en12020247>.

Guelpa E, Sciacovelli A, Verda V. Thermo-fluid dynamic model of large district heating networks for the analysis of primary energy savings. *Energy.* 2019;184:34–44.

Miriyala SS, Mitra K. Deep learning based system identification of industrial integrated grinding circuits. *Powder Technol.* 2020;360:921–36.
<https://doi.org/10.1016/j.powtec.2019.10.065>.

Qi C, Zeng X, Wang Y, Li H. Adaptive time window convolutional neural networks concerning multiple operation modes with applications in energy efficiency predictions. *Energy.* 2022;240:122506.
<https://doi.org/10.1016/j.energy.2021.122506>.

Han Y, Wu H, Geng Z, Zhu Q, Gu X, Yu B. Review: Energy efficiency evaluation of complex petrochemical industries. *Energy.* 2020;203:117893.
<https://doi.org/10.1016/j.energy.2020.117893>.

Paterakis NG, Mocanu E, Gibescu M, Stappers B, Van Alst W. Deep learning versus traditional machine learning methods for aggregated energy demand prediction. In: 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGTEurope). Torino: IEEE; 2017. p. 1–6.
<https://doi.org/10.1109/ISGTEurope.2017.8260289>.

Parmezan ARS, Souza VMA, Batista GEAPA. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Inf Sci.* 2019;484:302–7.
<https://doi.org/10.1016/j.ins.2019.01.076>.

Mawson VJ, Hughes BR. Deep learning techniques for energy forecasting and condition monitoring in the manufacturing sector. *Energy Build.* 2020;217:109966.
<https://doi.org/10.1016/j.enbuild.2020.109966>.

Lalapura VS, Amudha J, Satheesh HS. Recurrent neural networks for edge intelligence: A survey. *ACM Comput Surv.* 2021;54.
<https://doi.org/10.1145/3448974>.

Pujari KN, Miriyala SS, Mittal P, Mitra K. Better wind forecasting using evolutionary neural architecture search driven Green Deep Learning. *Expert Syst Appl.* 2023;214:119063.
<https://doi.org/10.1016/j.eswa.2022.119063>.

Pantula PD, Mitra K. A data-driven approach towards finding closer estimates of optimal solutions under uncertainty for an energy efficient steel casting process. *Energy.* 2019;189:116253.
<https://doi.org/10.1016/j.energy.2019.116253>.

Moghadasi M, Ozgoli HA, Farhani F. Steam consumption prediction of a gas sweetening process with methyldiethanolamine solvent using machine learning approaches. *Int J Energy Res.* 2021;45:879–93.
<https://doi.org/10.1002/er.5979>.

Han Y, Du Z, Geng Z, Fan J, Wang Y. Novel long short-term memory neural network considering virtual data generation for production prediction and energy structure optimization of ethylene production processes. *Chem Eng Sci.* 2023;267:118372.
<https://doi.org/10.1016/j.ces.2022.118372>.

Panjapornpon C, Rochpuang C, Bardeeniz S, Hussain MA. Machine learning approach with a posteriori-based feature to predict service life of a thermal cracking furnace with coking deposition. *Results Eng.* 2024;22:102349.
<https://doi.org/10.1016/j.rineng.2024.102349>.

Hasnen SH, Shahid M, Zabiri H, Taqvi SAA. Semi-supervised adaptive PLS softsensor with PCA-based drift correction method for online valuation of NOx emission in industrial water-tube boiler. *Process Saf Environ Protect.* 2023;172:787–801.
<https://doi.org/10.1016/j.psep.2023.02.080>.

Thebelt A, Wiebe J, Kronqvist J, Tsay C, Misener R. Maximizing information from chemical engineering data sets: Applications to machine learning. *Chem Eng Sci.* 2022;252:117469.

Yin Y, Jang-Jaccard J, Xu W, Singh A, Zhu J, Sabrina F, et al. IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSWNB15 dataset. *J Big Data.* 2023;10:15.

Qiu C, Zhu Y, Cheong KH, Yu D, Philip Chen CL. Continuous action iterated dilemma with data-driven compensation network and limited learning ability. *Inf Sci.* 2023;632:516–28.

Liu J, Chen L, Xu W, Feng M, Han Y, Xia T, et al. Novel production prediction model of gasoline production processes for energy saving and economic increasing based on AM-GRU integrating the UMAP algorithm. *Energy.* 2023;262:125536.

Xing Z, Pan Y, Yang Y, Yuan X, Liang Y, Huang Z. Transfer learning integrating similarity analysis for short-term and long-term building energy consumption prediction. *Appl Energy.* 2024;365:123276.

Liu X, Yu J, Gong L, Liu M, Xiang X. A GCN-based adaptive generative adversarial network model for short-term wind speed scenario prediction. *Energy.* 2024;294:130931.
<https://doi.org/10.1016/j.energy.2024.130931>.

Panjapornpon C, Bardeeniz S, Hussain MA. Improving energy efficiency prediction under aberrant measurement using deep compensation networks: A case study of petrochemical process. *Energy.* 2023;263:125837.
<https://doi.org/10.1016/j.energy.2022.125837>.

Han Y, Fan C, Xu M, Geng Z, Zhong Y. Production capacity analysis and energy saving of complex chemical processes using LSTM based on attention mechanism. *Appl Therm Eng.* 2019;160:114072.
<https://doi.org/10.1016/j.applthermaleng.2019.114072>.

Wan A, Chang Q, AL-Bukhaiti K, He J. Short-term power load forecasting for combined heat and power using CNN-LSTM enhanced by attention mechanism. *Energy.* 2023;282:128274.
<https://doi.org/10.1016/j.energy.2023.128274>.

Tian C, Liu Y, Zhang G, Yang Y, Yan Y, Li C. Transfer learning based hybrid model for power demand prediction of large-scale electric vehicles. *Energy.* 2024;300:131461.
<https://doi.org/10.1016/j.energy.2024.131461>.

Li G, Wu Y, Yoon S, Fang X. Comprehensive transferability assessment of short-term cross-building-energy prediction using deep adversarial network transfer learning. *Energy.* 2024;299:131395.
<https://doi.org/10.1016/j.energy.2024.131395>.

Lepori MA, Serre T, Pavlick E. Uncovering intermediate variables in transformers using circuit probing. 2023.
<https://doi.org/10.48550/arXiv.2311.04354>.

Zideh MJ, Chatterjee P, Srivastava AK. Physics-informed machine learning for data anomaly detection, classification, localization, and mitigation: A review, challenges, and path forward. *IEEE Access.* 2024;12:4597–617.
<https://doi.org/10.1109/ACCESS.2023.3347989>.

Ning L, Cai Z, Dong H, Liu Y, Wang W. Physics-informed neural network frameworks for crack simulation based on minimized peridynamic potential energy. *Comput Methods Appl Mech Eng.* 2023;417:116430.
<https://doi.org/10.1016/j.cma.2023.116430>.

Tompson J, Schlachter K, Sprechmann P, Perlin K. Accelerating Eulerian fluid simulation with convolutional networks. In: *Proceedings of the 34th International Conference on Machine Learning*. PMLR; 2017. p. 3424–33.

Du J, Zheng J, Liang Y, Xu N, Klemeš JJ, Wang B, et al. Deeppipe: A two-stage physics-informed neural network for predicting mixed oil concentration distribution. *Energy.* 2023;276:127452.
<https://doi.org/10.1016/j.energy.2023.127452>.

Gao K, Xu X, Jiao S. Prediction and visualization analysis of drilling energy consumption based on mechanism and data hybrid drive. *Energy.* 2022;261:125227.
<https://doi.org/10.1016/j.energy.2022.125227>.

Machalek D, Tuttle J, Andersson K, Powell KM. Dynamic energy system modeling using hybrid physics-based and machine learning encoder–decoder models. *Energy AI.* 2022;9:100172.
<https://doi.org/10.1016/j.egyai.2022.100172>.

Wang R, Kashinath K, Mustafa M, Albert A, Yu R. Towards physics-informed deep learning for turbulent flow prediction. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery; 2020. p. 1457–66.
<https://doi.org/10.1145/3394486.3403198>.

Zhang X, Hou L, Liu J, Yang K, Chai C, Li Y, et al. Energy consumption prediction for crude oil pipelines based on integrating mechanism analysis and data mining. *Energy.* 2022;254:124382.
<https://doi.org/10.1016/j.energy.2022.124382>.

Gallup E, Gallup T, Powell K. Physics-guided neural networks with engineering domain knowledge for hybrid process modeling. *Comput Chem Eng.* 2023;170:108111.
<https://doi.org/10.1016/j.compchemeng.2022.108111>.

Gao M, Yu J, Yang Z, Zhao J. A physics-guided graph convolution neural network for optimal power flow. *IEEE Trans Power Syst*. 2024;39:380–90.

<https://doi.org/10.1109/TPWRS.2023.3238377>.

Bolderman M, Lazar M, Butler H. On feed-forward control using physics-guided neural networks: Training cost regularization and optimized initialization. In: 2022 European Control Conference (ECC). London, United Kingdom: IEEE; 2022. p. 1403–8.

Bardeeniz S, Panjapornpon C, Fongsamut C, Ngaotranwiwat P, Azlan Hussain M. Digital twin-aided transfer learning for energy efficiency optimization of thermal spray dryers: Leveraging shared drying characteristics across chemicals with limited data. *Appl Therm Eng*. 2024;242:122431.

<https://doi.org/10.1016/j.applthermaleng.2024.122431>.

Cai J, Yang H. Attention mechanism-aided model ensemble method of chiller energy consumption prediction. *Int J Refrig*. 2024;165:111–21.

<https://doi.org/10.1016/j.ijrefrig.2024.05.017>.

Panjapornpon C, Bardeeniz S, Hussain MA, Chomchai P. Explainable deep transfer learning for energy efficiency prediction based on uncertainty detection and identification. *Energy AI*. 2023;12:100224.

<https://doi.org/10.1016/j.egyai.2022.100224>.

Ba Y, Zhao G, Kadambi A. Blending diverse physical priors with neural networks. 2019.

Chong S. Process simulation for VCM production. In: *Chemical Engineering Process Simulation*. Elsevier; 2017. p. 253–73.

<https://doi.org/10.1016/B978-0-12-803782-9.00012-1>.

Mansouri SA, Rezaee Jordehi A, Marzband M, Tostado-Véliz M, Jurado F, Aguado JA. An IoT-enabled hierarchical decentralized framework for multi-energy microgrids market management in the presence of smart prosumers using a deep learning-based forecaster. *Appl Energy*. 2023;333:120560.

Hu J, Zheng W. Transformation-gated LSTM: Efficient capture of short-term mutation dependencies for multivariate time series prediction tasks. In: 2019 International Joint Conference on Neural Networks (IJCNN). Budapest, Hungary: IEEE; 2019. p. 1–8.

<https://doi.org/10.1109/IJCNN.2019.8852073>.

Chen Z, Badrinarayanan V, Lee C-Y, Rabinovich A. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. 2018.

Li K. Quantization loss re-learning method. *ArXiv* 2019 ;abs/ 1905.13568.

Bardeeniz S, Panjapornpon C, Hounkim W, Dechakupt T, Tawai A. Artificial intelligence-driven control for enhancing carbon dioxide-based wastewater pH regulation in tubular reactor. *Comput Chem Eng*. 2025;192:108880.

<https://doi.org/10.1016/j.compchemeng.2024.108880>.

Walser T, Sauer A. Typical load profile-supported convolutional neural network for short-term load forecasting in the industrial sector. *Energy AI*. 2021;5:100104.

<https://doi.org/10.1016/j.egyai.2021.100104>.

Zhu J, Su L, Li Y. Wind power forecasting based on new hybrid model with TCN residual modification. *Energy AI*. 2022;10:100199.

<https://doi.org/10.1016/j.egyai.2022.100199>.

Yao Z, Chen T, Lin W, Feng Y, Wei Z. Comparative analysis of time series neural network methods for three-way catalyst modeling. *Energy AI*. 2024;17:100394.

<https://doi.org/10.1016/j.egyai.2024.100394>.

Kaur D, Islam SN, Mahmud MdA, MdE Haque, Anwar A. A VAE-Bayesian deep learning scheme for solar power generation forecasting based on dimensionality reduction. *Energy AI*. 2023;14:100279.

<https://doi.org/10.1016/j.egyai.2023.100279>.

Xu B, Ma W, Wu W, Wang Y, Yang Y, Li J, et al. Degradation prediction of PEM water electrolyzer under constant and start-stop loads based on CNN-LSTM. *Energy AI*. 2024;18:100420.

<https://doi.org/10.1016/j.egyai.2024.100420>.

Pydi DP, Advait S. Attention boosted autoencoder for building energy anomaly detection. *Energy AI*. 2023;14:100292.

<https://doi.org/10.1016/j.egyai.2023.100292>.

Ghimire S, Nguyen-Huy T, AL-Musaylh MS, Deo RC, Casillas-Pérez D, Salcedo-Sanz S. Integrated multi-head self-attention transformer model for electricity demand prediction incorporating local climate variables. *Energy AI*. 2023;14:100302.

<https://doi.org/10.1016/j.egyai.2023.100302>.

Cibaku E, Gama F, Park S. Boosting efficiency in state estimation of power systems by leveraging attention mechanism. *Energy AI*. 2024;16:100369.

<https://doi.org/10.1016/j.egyai.2024.100369>.

Piantadosi G, Dutto S, Galli A, De Vito S, Sansone C, Di Francia G. Photovoltaic power forecasting: a Transformer based framework. *Energy AI*. 2024;18:100444.

<https://doi.org/10.1016/j.egyai.2024.100444>.

Panjapornpon C, Bardeeniz S, Hussain MA, Vongvirat K, Chuay-ock C. Energy efficiency and savings analysis with multirate sampling for petrochemical process using convolutional neural network-based transfer learning. *Energy AI*. 2023;14:100258.

<https://doi.org/10.1016/j.egyai.2023.100258>.

Intergovernmental Panel on Climate Change. Sixth Assessment Report — IPCC 2021. <https://www.ipcc.ch/assessment-report/ar6/> (accessed June 25, 2024).

United States Environmental Protection Agency (U.S. EPA). GHG emission factors hub 2015.

<https://www.epa.gov/climateleadership/ghg-emission-factors-hub> (accessed July 8, 2022).

Energy Information Administration (EIA). Henry hub natural gas spot price (dollars per million Btu) 2024.

<https://www.eia.gov/dnav/ng/hist/rngwhhdm.htm> (accessed June 25, 2024).

Bardeeniz S, Panjapornpon C, Fongsamut C, Ngaotranwiwat P, Hussain MA. Energy efficiency characteristics analysis for process diagnosis under anomaly using self-adaptive-based SHAP guided optimization. *Energy*. 2024:133074.

<https://doi.org/10.1016/j.energy.2024.133074>.

Maschler B, Braun D, Jazdi N, Weyrich M. Transfer learning as an enabler of the intelligent digital twin. *Procedia CIRP*. 2021;100:127–32.

<https://doi.org/10.1016/j.procir.2021.05.020>.