WENFlow: Scalable Attention for Unsupervised Spatiotemporal Anomaly Detection in High-Dimensional Cyber-Physical Systems

Jacob Buckelew*, Jose Paolo Talusan*, Vasavi Sivaramakrishnan†,
Ayan Mukhopadhyay‡, Anurag Srivastava†, Abhishek Dubey*
*Vanderbilt University, Nashville, TN, USA
{jacob.buckelew, jose.paolo.talusan, abhishek.dubey}@vanderbilt.edu
†West Virginia University, Morgantown, WV, USA
{vs00016, anurag.srivastava}@mix.wvu.edu
‡William and Mary, Williamsburg, VA, USA
amukhopadhyay@wm.edu

***Abstract*—Real-time anomaly detection in high-dimensional data is crucial for ensuring the security of cyber-physical systems (CPS) such as power grids and water distribution networks. Such data commonly take the form of multivariate time series, often unlabeled and necessitating the need for unsupervised detection methods. However, many unsupervised deep learning methods make assumptions about the normality of training data, which is unrealistic in real-world CPS where training data often contain anomalies or rare patterns. Furthermore, these methods rely on inefficient mechanisms to learn spatiotemporal dependencies in the data and scale quadratically with the number of system features. To address these problems, we propose Wavelet-Enhanced Normalizing Flows (WENFlow), an unsupervised deep learning model that identifies anomalies in low-density regions of the data distribution and does not assume access to anomaly-free training data. Notably, WENFlow leverages a scalable Gated Selective Self-Attention mechanism for capturing the most critical spatial dependencies between features. Compared to existing models, WENFlow scales linearly with respect to the number of system features and meets real-time inference requirements for anomaly detection. In our experiments, WENFlow achieves superior AUC scores against baseline methods across datasets with varying anomaly ratios, showcasing its robustness against contaminated training data. We evaluate WENFlow on 2 real-world benchmark datasets and a simulated phasor measurement unit dataset collected from a power grid testbed.**

***Index Terms*—multivariate time series, unsupervised learning, wavelet transform, phasor measurement unit**

I. Introduction

Modern large-scale infrastructure such as electric power grids and water-treatment facilities operate as tightly coupled cyber-physical systems (CPS), where sensing, communication, computation, and actuation interact continuously in real time. In such systems, timely detection of abnormal events is essential, as faults in one component can propagate through the intertwined physical and cyber layers, potentially triggering cascading failures. Prior analyses in power grids show that even minor disturbances in the physical network or control layers can escalate into large-scale collapses when left undetected [1, 2]. Similar vulnerabilities have been documented in water-distribution systems, where sensor faults, valve malfunctions, or cyber intrusions have propagated through hydraulic and control interconnections, disrupting service and safety [3].

Maintaining situational awareness in such CPS requires continuous monitoring through dense sensor networks, such as phasor measurement units (PMUs) in power systems, which provide time-synchronized measurements of voltage, current, phase angle, and frequency at sampling rates up to 30–60 Hz [4]. Each sensor measures multiple interdependent physical quantities—changes in one bus's voltage angle can affect neighboring currents and reactive power flows—creating a coupled, dynamic environment. Effective monitoring demands a multivariate time-series (MTS) approach that jointly models temporal evolution and spatial dependencies across sensors. In contrast, univariate methods that analyze each signal independently fail to capture coordinated, benign adjustments that maintain system stability and often misclassify them as anomalies, while missing early signs of genuine faults.

Given the problem scale and the need to capture spatiotemporal dependencies, recent work has turned to machine learning methods to solve the MTS anomaly detection problem. Most techniques fall into two categories: reconstruction-based models and density-based models, with a smaller class focusing on out-of-distribution (OOD) detection. Reconstruction-based and density-based approaches typically begin by forming a time-windowed representation wherein each sample is a matrix whose rows correspond to features aggregated across sensors and columns correspond to temporal steps. Reconstruction-based approaches rely on attention [5] to capture long-range dependencies. Methods such as Autoformer [6], Informer [7], and AnomalyTransformer [8] use attention to model temporal structure, while TranAD [9] and DCDetector [10] incorporate short-term context and contrastive objectives. However, these approaches assume fault-free training data, an unrealistic condition in CPS where logs often contain sensor errors, unlogged maintenance events, and contamination from malicious attacks [11, 12, 13].

Density-based methods learn the underlying data distribution, with normalizing flows [14, 15] providing exact likelihoods through bijective mappings to simple priors. GANF [16] models features as nodes in a Bayesian network, and MTGFlow [17] uses self-attention to capture fully-connected dependencies. These methods exhibit robustness to contaminated training data since anomalies naturally map to low-density regions, but they still incur quadratic complexity when modeling spatiotemporal dependencies, limiting their scalability. The third class of methods relies on conformal anomaly detection [18] to identify OOD samples that deviate from the training distribution. Recently, CODiT [19] has been tested on vision and physiological datasets, but its scalability and robustness to anomaly contamination remain largely unproven.

Our Contributions. These limitations highlight the need for an anomaly detection framework that is robust to contaminated, unlabeled training data and scalable to high-dimensional CPS. To address these challenges, we introduce *WENFlow*, a wavelet-enhanced conditional normalizing flow framework designed specifically for real-time CPS monitoring. Its key features are as follows:

- **Wavelet-guided feature extraction.** We use discrete wavelet transform to decompose each sensor stream into low-frequency (trend) and high-frequency (transient) components, enabling the model to focus on anomalous trends and localized disruptions.
- **A scalable Gated Selective Self-Attention mechanism.** We introduce a feature-wise attention module that ranks sensor importance using high-frequency coefficients, computes attention only over the top-k critical features, and achieves *linear* complexity in the number of sensors D.
- **A conditional normalizing flow for likelihood estimation.** By conditioning RealNVP-based [14] flows on spatiotemporal embeddings, WENFlow directly models the data distribution and maps anomalous samples to low-density regions, improving robustness under contaminated training logs.
- **Interpretability.** WENFlow produces log-densities and feature-level importance scores per time window, enabling interpretable detection of localized anomalies and revealing temporal patterns observed during cascading failures.

To the best of our knowledge, WENFlow is the first framework to combine wavelet-based time–frequency decomposition, scalable feature-wise attention, and normalizing flows for MTS anomaly detection. This enables WENFlow to achieve strong detection performance while maintaining robustness to contaminated training data and scalability to high-dimensional CPS. We evaluate WENFlow against state-of-the-art baselines, including transformers (AnomalyTransformer, TranAD, DCDetector), flow-based models (GANF, MTGFlow), an OOD method CODiT, and a wavelet-based method MEGA [20]. Experiments span CPS benchmarks – SWaT [21] and WADI [3] for water-treatment systems. In addition, we utilize a Real-Time Digital Simulator (RTDS), used in power system studies for hardware-in-the-loop simulation, to generate a PMU dataset. As shown later in the paper, WENFlow achieves strong performance across various anomaly ratios, scales linearly with sensor dimensionality, and enables interpretable analysis of anomalies in high-dimensional CPS.

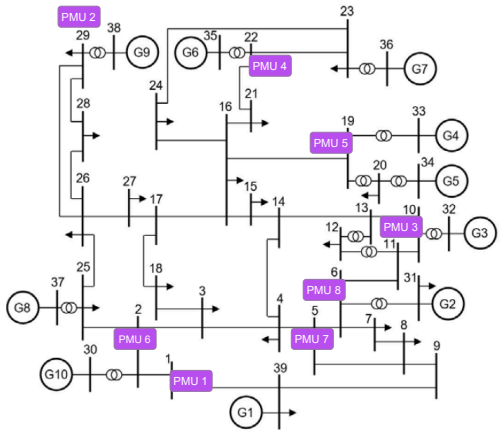


Fig. 1: IEEE 39-bus system with phasor measurement unit (PMU) locations shown.

II. Problem Formulation

We consider a CPS monitored by D heterogeneous sensors, each providing continuous measurements of physical or cyber state variables. At every time step t, the system generates a vector

$$\mathbf{x}_t = [\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^D]^\top \in \mathbb{R}^D,$$

where each entry corresponds to one sensor or feature. To capture short-term temporal evolution and cross-sensor interactions, we operate on fixed-length sliding windows of size W, forming

$$\mathbf{X}_t = [\mathbf{x}_{t-W+1}, \mathbf{x}_{t-W+2}, \ldots, \mathbf{x}_t] \in \mathbb{R}^{W \times D}$$

In realistic CPS, operational logs are unlabeled and often contain sensor noise, calibration drift, early-stage faults, and unlogged maintenance events. Thus, we do not assume access to clean or anomaly-free training data. Given an unlabeled collection of windowed samples $\{\mathbf{X}_t\}$, the objective of multivariate time-series anomaly detection is to learn a function

$$s_t = f_\theta(\mathbf{X}_t)$$

that assigns an anomaly score to each window. An example system we use later for discussion and evaluation is shown in Figure 1. This is a standard IEEE 39-bus system wherein we consider the placement of eight PMUs at buses $2, 5, 6, 10, 19, 22, 29$ and 39. Each PMU monitors 12 features including voltage magnitude and phase angle.

III. Related Work

Research on MTS anomaly detection spans classical techniques, deep learning-based reconstruction and density models, and more recent efforts in OOD detection and wavelet-based detection. Classical methods such as SVMs [22], k-means clustering, LOF [23], and isolation trees [24] can be effective for low-dimensional or univariate settings, but

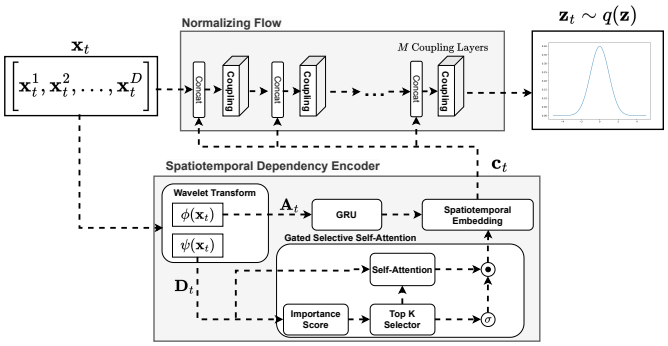


Fig. 2: WENFlow Architecture. Multivariate time series is sent through a spatiotemporal dependency encoder, which first computes the low-frequency and high-frequency wavelet coefficients. A GRU layer captures temporal dependencies and Gated Selective Self-Attention extracts important high-frequency interactions between features. A spatiotemporal embedding computes a context vector, which is then concatenated to the original time series and passed through a normalizing flow.

they fail to model temporal dependencies and cross-sensor interactions characteristic of CPS data [25]. Their inability to incorporate spatial structure or scale to hundreds of channels limits their applicability to modern sensor networks.

Reconstruction-based methods are the dominant class of deep learning methods in MTS anomaly detection. These methods require clean training data to learn the semantics of normal MTS. Early works employed recurrent neural networks such as LSTMs [26], using encoder–decoder models to reconstruct normal MTS and flag anomalies via reconstruction error. Following the introduction of attention [5], transformers have become the primary reconstruction-based method. AnomalyTransformer [8] introduces an attention mechanism to quantify association discrepancies; TranAD [9] incorporates sub-sequence attention to capture short-range temporal pattern shifts; and DCDetector [10] employs contrastive learning to improve representation quality. Hybrids such as AnomalyBERT [27] further extend the transformer family.

Density-based models offer an alternative by directly estimating the probability distribution of MTS. Normalizing flows [14, 15] compute exact likelihoods and thus naturally separate anomalous, low-density data from nominal behavior. GANF [16] models sensor relationships through a Bayesian-network formulation, while MTGFlow [17] incorporates self-attention to capture fully-connected feature-wise dependencies. Although flows mitigate the contamination issue, they still incur quadratic scaling in the feature dimension when modeling spatial relationships, limiting their scalability.

For CPS, **Out of Distribution** (OOD) detection methods such as CODiT [19], which leverage conformal anomaly detection [18], have shown promise for identifying departures from nominal distributions with calibrated confidence. Finally, wavelet-based methods provide a complementary perspective by analyzing signals across multiple temporal scales. Wavelet transforms can isolate localized transients—often the clearest indicators of faults—from slower operational trends, making them well-suited for CPS where disturbances frequently manifest as sharp, high-frequency anomalies. Prior work such as MEGA [20] demonstrate the value of discrete wavelet decomposition for extracting multi-scale representations before applying neural modeling, and wavelet-based diagnostics have a long history in power systems [28].

Overall, current research reveals two persistent gaps: difficulty handling contaminated operational logs, and poor scalability when modeling full cross-sensor dependencies. These limitations motivate the development of approaches that jointly capture multi-scale temporal structure, scalable spatial dependencies, and robust probabilistic behavior—particularly in the tightly coupled, high-dimensional settings characteristic of modern CPS.

TABLE I: Symbols Table.

Symbol	Description
$\mathbf{X}$	A window of time series
D	Number of data features
$\mathbf{z}$	Standard Normal Gaussian vector
W	Window size
M	Number of coupling layers
$\mathbf{a}$	Importance score vector
$\mathbf{A}$	Approximation coefficients
$\mathbf{D}$	Detail coefficients
$\mathbf{C}$	Spatiotemporal embedding
$\mathcal{A}$	Attention score matrix
$\mathbf{S}$	Spatial dependency matrix
$\mathbf{T}$	Temporal dependency matrix
k	Number of critical sensors
d	Hidden dimension size
B	Batch of samples

IV. WENFlow

WENFlow integrates multi-resolution temporal analysis, scalable spatial dependency modeling, and conditional density estimation into a unified architecture for MTS anomaly detection. As shown in Figure 2, each input window $\mathbf{X}$ is first decomposed into low- and high-frequency components using the discrete wavelet transform. The low-frequency components are then processed with a Gated Recurrent Unit (GRU) to capture long-term temporal structure without being dominated by noise-driven fluctuations. In parallel, the high-frequency components are used by a Gated Selective Self-Attention mechanism to identify a subset of critical sensors and model their dependencies with all other sensors, enabling scalable spatial reasoning while emphasizing meaningful interactions.

A combined spatiotemporal embedding is concatenated with the MTS and conditioned into a normalizing flow, which produces timestamp-level log-densities under the hypothesis that anomalies correspond to low-density regions of the data distribution. Together, these components allow WENFlow to handle contaminated training data and scale to high-dimensional CPS. We describe each major component in subsequent subsections. Important notation used throughout this section is provided in Table I.

A. Wavelet Transform

Many CPS anomalies manifest as sharp, localized transients—such as abrupt pressure spikes and voltage sags—or slowly evolving deviations from normal behavior. Directly modeling raw time-series signals forces the model to learn both slow trends and fast disturbances simultaneously, which increases complexity and makes the system more sensitive to contaminated training data. To address this, WENFlow applies a wavelet-based decomposition that separates each signal into multi-scale components, enabling the model to reason about slow operational behavior and high-frequency disruptions in a principled and computationally efficient manner.

Discrete wavelet transform (DWT) provides a multi-scale representation of signals [29], offering joint time–frequency localization. We apply DWT to $\mathbf{X}^\top \in \mathbb{R}^{D\times W}$ using a low-pass filter $\phi(\mathbf{X})$ and high-pass filter $\psi(\mathbf{X})$, applied feature-wise across the D sensor streams. We adopt the undecimated wavelet transform, which preserves temporal alignment and outputs approximation coefficients $\mathbf{A} \in \mathbb{R}^{D\times W}$ and detail coefficients $\mathbf{D} \in \mathbb{R}^{D\times W}$. The approximation coefficients capture low-frequency structure often associated with anomalous long-term trends [30], whereas the detail coefficients emphasize high-frequency behavior typically associated with abrupt changes [31].

In practice, the choice of wavelet basis influences the trade-off between smoothness, transient sensitivity, and computational efficiency. WENFlow uses commonly adopted orthogonal wavelet families—including Haar, Daubechies (e.g., db1 and db2), and Coiflets (e.g., Coif1 and Coif2)—which differ in the number of vanishing moments and support length [32]. The db2 wavelet provides compact support and strong localization of sharp disturbances, making it well-suited for detecting abrupt events such as electrical faults. Coiflets, by contrast, have vanishing moments in both the wavelet and scaling functions, enabling improved capture of both smooth operational trends and high-frequency anomalies; Coif1 offers a balanced representation, while Coif2 provides greater smoothness and richer approximation capability. These properties allow WENFlow to adapt effectively across diverse CPS domains.

B. Gated Selective Self-Attention

In CPS such as power networks, faults often originate in a small set of components before propagating through the network [33]. These anomalies can appear as high-frequency disturbances in only a few sensor streams, making it unnecessary to model fully-connected dependencies among all sensors. To exploit this structure, we introduce a Gated Selective Self-Attention mechanism that uses wavelet detail coefficients to identify critical sensors and compute feature-wise dependencies in a scalable manner.

Self-attention [5] projects each element into query, key, and value vectors and computes relevance scores through scaled dot-product attention, which can capture cross-sensor interactions. Transformer-based reconstruction models [17, 8, 10, 9] apply self-attention across *timesteps*—yielding a complexity of $O(W^2D)$ that is effectively linear in D for small window sizes W. However, flow-based models such as MTGFlow [17] apply fully-connected feature-wise attention in each time step, requiring $O(D^2W)$ operations and becoming expensive for high-dimensional CPS. Our design addresses this limitation by restricting feature-wise attention to a small subset of sensors, preserving expressivity while reducing the complexity of feature-wise attention from quadratic to linear in D.

Given the detail coefficients $\mathbf{D}$, we compute a learnable importance score for each feature using a projection vector $\mathbf{p} \in \mathbb{R}^{W\times 1}$ applied to its detail coefficients. The top-k features with highest scores form the matrix $\mathbf{D}_k$, which serves as the key set for self-attention. This reduces the complexity of feature-wise attention from $O(D^2W)$ to $O(DWk)$, where $k \ll D$. Given hidden dimension d, we form the query, key, and value matrices using learnable weights $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{W\times d}$:

$$\mathbf{Q} = \mathbf{D}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{D}_k\mathbf{W}^K, \quad \mathbf{V} = \mathbf{D}_k\mathbf{W}^V,$$

and compute the attention score matrix $\mathcal{A} \in \mathbb{R}^{D\times k}$ using the standard scaled dot-product attention formulation:

$$\mathcal{A} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right).$$

The resulting feature-wise attention output is then given by $\mathbf{H}_{att} = \mathcal{A}\mathbf{V}$. Because the importance scores identify sensors exhibiting strong transient behavior, we modulate the attention output using a gating mechanism. Letting $\mathbf{a} \in \mathbb{R}^{D\times 1}$ denote the learned importance score vector, we define the spatial dependency matrix as

$$\mathbf{S} = \mathbf{H}_{att} \odot \left(\sigma(\mathbf{a})\mathbf{1}^\top\right),$$

where $\sigma(\cdot)$ is the sigmoid function and $\mathbf{1} \in \mathbb{R}^{W\times 1}$ denotes a vector of ones. The gating operation amplifies dependencies associated with influential sensors and attenuates those arising from low-importance features, yielding an interpretable representation of cross-sensor interactions that is based on each sensor's significance.

C. Spatiotemporal Dependency Encoder

Complex CPS dynamics arise from the interaction of slowly varying operational trends and rapidly evolving localized disturbances. Effective anomaly detection therefore requires jointly modeling temporal evolution and spatial dependencies while preserving scalability. WENFlow achieves this by using the wavelet-based decomposition to separate these two aspects: the approximation coefficients $\mathbf{A}$ capture smooth, low-frequency structure well suited for temporal modeling, while the detail coefficients $\mathbf{D}$ highlight transient activity that informs selective spatial reasoning.

To capture temporal dependencies in $\mathbf{X}$, we process the approximation coefficients with a Gated Recurrent Unit (GRU), defining

$$\mathbf{T} = RNN(\mathbf{A}^\top) \in \mathbb{R}^{W\times d}.$$

Low-frequency trends provide a stable view of system evolution across the time window. To capture spatial dependencies, we apply the Gated Selective Self-Attention mechanism,

which identifies a small set of critical sensors and computes their influence on all remaining features, producing the spatial dependency matrix $\mathbf{S}$.

A learnable spatiotemporal embedding layer then combines the temporal and spatial representations to form a contextual embedding $\mathbf{C} \in \mathbb{R}^{W\times d}$ that can be conditioned on in a normalizing flow. Given weight matrices $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_3 \in \mathbb{R}^{D\times d}$, the embedding is computed as

$$\mathbf{C} = \alpha\big(\mathbf{T}\mathbf{W}_1 + \mathbf{S}^\top \mathbf{W}_3\big)\,\mathbf{W}_2,$$

where α denotes the ReLU activation function. This encoder provides a unified and scalable representation of the long-term temporal structure and critical spatial interactions, improving density estimation to detect anomalous events.

D. Density Estimation

To estimate the density of $\mathbf{X}$ given spatiotemporal information $\mathbf{C}$, we employ a conditional normalizing flow that maps the data distribution $p(\mathbf{x})$ to a standard Gaussian density $q(\mathbf{z})$. Normalizing flows provide exact likelihoods through a sequence of invertible transformations with tractable Jacobians, making them well-suited for anomaly detection under contaminated training data. In WENFlow, we adopt a RealNVP-style affine coupling architecture [14], where the inputs of each coupling transformation are conditioned on the spatiotemporal embedding $\mathbf{C}$ (Section IV-C). This ensures that density estimation adapts to spatiotemporal context.

Letting $p(\mathbf{X}) = p(\mathbf{x}_1)\,p(\mathbf{x}_2|\mathbf{x}_1)\cdots p(\mathbf{x}_W|\mathbf{x}_{<W})$, we reparameterize each conditional probability by setting $\mathbf{x}_{<w} = \mathbf{c}_w$, where $\mathbf{c}_w \in \mathbb{R}^d$ denotes the spatiotemporal dependencies extracted from timesteps before w. Applying $\log$ to the factorization yields

$$\log p(\mathbf{X}) = \sum_{w=1}^{W} \log p(\mathbf{x}_w; \mathbf{c}_w),$$

which computes W timestamp-level log-densities.

For each timestep w, the flow computes $\mathbf{z}_w = f(\mathbf{x}_w; \mathbf{c}_w)$ using a sequence of conditional affine coupling layers. Each coupling layer partitions $\mathbf{x}_w$ into two subsets $\mathbf{x}_w^{(1)}$ and $\mathbf{x}_w^{(2)}$ and applies a scale-and-shift transformation defined by

$$\mathbf{z}_w^{(1)} = \mathbf{x}_w^{(1)}, \qquad \mathbf{z}_w^{(2)} = \mathbf{x}_w^{(2)} \odot \exp\Big(s(\mathbf{x}_w^{(1)}, \mathbf{c}_w)\Big) + t(\mathbf{x}_w^{(1)}, \mathbf{c}_w),$$

where $s(\cdot)$ and $t(\cdot)$ are Multi-layer Perceptrons (MLPs) conditioned on $\mathbf{c}_w$. Because the Jacobian of an affine coupling layer is triangular, its determinant is efficient to compute. Using the change-of-variables formula gives the timestamp-level conditional log-density:

$$\log p(\mathbf{X}) = \sum_{w=1}^{W} \log q(f(\mathbf{x}_w; \mathbf{c}_w)) + \log \Big|\det \nabla_{\mathbf{x}_w} f(\mathbf{x}_w; \mathbf{c}_w)\Big|$$

Lower log-density values indicate that a window lies in a low-density region of the learned nominal distribution, making it more likely to be anomalous. During training, we minimize the negative log-density over a batch B of windows:

$$\mathcal{L} = \frac{-1}{|B|\,W} \sum_{\mathbf{X}\in B} \sum_{w=1}^{W} \Big[\log q(\mathbf{z}_w) + \log \Big|\det \nabla_{\mathbf{x}_w} f(\mathbf{x}_w; \mathbf{c}_w)\Big|\Big]$$

where $\mathbf{z}_w = f(\mathbf{x}_w; \mathbf{c}_w) \sim q(\mathbf{z})$ and q is chosen as a standard normal multivariate Gaussian. This objective enables WENFlow to learn context-aware densities that reflect both temporal evolution and localized spatial structure.

E. Complexity Analysis

Given WENFlow's design, each major component scales linearly in the number of features D. The discrete wavelet transform (DWT) operates independently on each feature and has complexity $O(DW)$ when an orthogonal wavelet basis is used. The GRU-based temporal embedding scales as $O(DWd)$, where d is the hidden dimension.

The spatiotemporal embedding layer has complexity $O(W(d^2 + Dd))$. For high-dimensional CPS where $D \gg d$, this simplifies to $O(DWd)$. The Gated Selective Self-Attention mechanism computes a $D\times k$ attention score matrix, reducing the cost of feature-wise attention from $O(D^2Wd)$ to $O(DWkd)$, which remains linear in D when k is treated as a constant. This mechanism is also linear in the window length W, offering flexibility when tuning temporal resolution and improved efficiency compared to temporal-wise attention used in transformers, which is typically quadratic in W.

For density estimation, a normalizing flow with M affine coupling layers contributes a complexity of $O(MDWd)$. Combining these components, and considering a batch of size $|B|$, the overall computational complexity of WENFlow is

$$O(|B|DWd(k + M)),$$

after removing constant factors. Linear scaling in D makes WENFlow more efficient than prior flow-based approaches whose spatial modeling incurs quadratic complexity.

F. Training and Inference

WENFlow is trained in a fully unsupervised manner without assuming that the operational logs used for training are free of anomalies. Instead, we rely on the normalizing flow objective to learn the underlying data distribution, allowing low-density regions to naturally correspond to anomalous behavior.

During training, we optimize the negative log-density loss using mini-batch gradient descent. For each window $\mathbf{X}$ in a batch, WENFlow computes the multi-resolution wavelet decomposition, extracts temporal and spatial dependencies through the GRU and Gated Selective Self-Attention modules, forms the spatiotemporal context $\mathbf{C}$, and conditions the normalizing flow to obtain timestamp-level latent variables $\mathbf{z}_w = f(\mathbf{x}_w; \mathbf{c}_w)$. The model selected for evaluation is the one that minimizes the loss on validation data.

At inference time, incoming data are processed in sliding windows, and we compute the anomaly score

$$s(\mathbf{X}) = -\log \text{Density}(\mathbf{X}),$$

TABLE II: Dataset statistics for all three CPS datasets. Train and Test denote the lengths of the training and test splits, respectively. Anomaly ratios, AR (%), are shown for the train and test splits.

Data	Domain	Dim.	Train	Train AR(%)	Test	Test AR(%)
SWaT	Water Dist.	51	269,951	17.7	89,984	5.2
WADI	Water Dist.	123	103,680	6.4	34,561	4.6
PMU	Power Grid	96	644,713	0.1	9,091	25.4

where lower log-densities indicate that a window is less consistent with the learned nominal distribution. Normal samples therefore exhibit low negative log-densities, while anomalous samples are assigned higher negative log-densities.

V. Experiments

We evaluate WENFlow across several dimensions: robustness to contaminated training data, overall class separability, computational efficiency, scalability with respect to feature dimension, and interpretability in scenarios involving multi-stage faults such as cascading failures. Our experiments are structured around five hypotheses. First, conditional density estimation should allow WENFlow to learn a stable representation of nominal CPS behavior even when the training data are partially contaminated. Second, combining wavelet decomposition with selective attention should improve class separability relative to existing baselines. Third, WENFlow should scale linearly in D, enabling efficient deployment in high-dimensional CPS. Fourth, each major component—wavelets, gated selective self-attention, and conditional flows—should contribute meaningfully to overall performance. Finally, the top-k feature selection mechanism should provide interpretable insights into how anomalies originate and propagate.

Datasets. We evaluate on three MTS datasets representative of operational CPS: SWaT and WADI from water-treatment and water-distribution testbeds, and a high-dimensional PMU dataset from an electric power grid simulation. The SWaT dataset [21] contains 51 sensors sampled at 1 Hz across four days, including 36 attacks that range from short transients to multi-stage faults. WADI [3] extends this architecture to 123 sensors and includes 15 attacks over two days. Both datasets contain strong cross-sensor dependencies and heterogeneous anomaly patterns. The PMU dataset is generated using a real-time digital simulator (RTDS) of the IEEE 39-bus system. Eight PMUs record 96 electrical features at high temporal resolution, capturing operational noise, isolated point anomalies, and five cascading-failure events. These characteristics make the PMU dataset particularly valuable for testing robustness and interpretability. Table II provides dataset statistics.

Baselines. We compare WENFlow against classical anomaly detection methods (LOF, Isolation Forest, OCSVM), reconstruction-based models (AnomalyTransformer, TranAD, DCDetector), density-based approaches (GANF, MTGFlow), the conformal OOD method CODiT, and the wavelet-based method MEGA (see Section III).

TABLE III: Hyperparameters selected for each dataset.

Data	Learning Rate	d	M	W	k	Wavelet
SWaT	0.001	16	1	64	2	coif1
WADI	0.001	16	1	32	12	coif2
PMU	0.006	32	1	16	9	db2

Training and Validation. For SWaT and WADI, we adopt a 60/20/20 train/validation/test split using the attack data files [16, 17]. For PMU, normal-operation logs are split 80/20 for training and validation, while all cascading-failure scenarios are held out exclusively for testing. Models are trained using Python 3.11 on a Linux machine with an AMD processor (32 cores), 96 GB RAM, and a single NVIDIA Titan X GPU (12 GB). Across all datasets, the batch size is 512 and training is capped at 50 epochs. We also perform a grid search on the following hyperparameters: number of coupling layers M (1, 2), hidden dimension size d (8, 16, 32, 64), window size W (16, 32, 48, 64, 96), wavelet basis (haar, db1, db2, coif1, coif2), and learning rate (0.001, 0.002, .003, .004, .005, .006, 0.0005). For tuning k, we test different ratios of D (0.05, 0.10, 0.15, 0.20, 0.25). The final model for each dataset is selected based on the lowest validation loss, and their hyperparameters are given in Table III.

Evaluation Metrics. Our primary evaluation metric is the area under the ROC curve (AUC), which is threshold-invariant and directly measures class separability. In CPS, anomaly prevalence and severity vary widely across datasets and operational conditions; threshold-dependent metrics can obscure true separability in such settings. For completeness, we report F1 scores, which represent the harmonic mean of precision and recall. This avoids tuning thresholds on training or unseen test data, ensuring consistent and unbiased comparisons across deep learning baselines. For each model, the anomaly score (negative log-density, reconstruction error, etc.) that maximizes F1 on the validation set defines the threshold ϵ. During testing, a window $\mathbf{X}$ is labeled anomalous if $s(\mathbf{X}) > \epsilon$, and its ground-truth label is 1 if any timestamp within the window is anomalous. Timestamp-level log-densities are also examined to support qualitative and interpretability analyses.

A. Anomaly Detection Performance

We begin by evaluating WENFlow's ability to separate anomalous and nominal behavior across three CPS datasets. This analysis addresses our first two hypotheses: that conditional density estimation confers robustness to contaminated training data, and that combining wavelet decomposition with selective attention improves class separability relative to reconstruction- and flow-based models.

Table IV reports AUC and F1 scores for all baselines and for WENFlow. WENFlow achieves the highest AUC scores, demonstrating superior threshold-invariant class separability. This is very evident for SWaT, where the training set contains a high anomaly ratio (17.7%). Reconstruction-based

TABLE IV: Anomaly detection results. For each metric, the best model's value is in bold and the second-best model's value is underlined. Standard deviations are also shown. We focus on AUC performance as it is threshold-invariant.

Dataset	SWaT		WADI		PMU	
Metric	F1	AUC	F1	AUC	F1	AUC
OCSVM	9.9 ± 0.0	32.6 ± 0.0	17.5 ± 0.0	18.3 ± 0.0	47.9 ± 0.0	29.1 ± 0.0
IF	24.1 ± 4.6	37.8 ± 1.9	30.2 ± 3.1	15.5 ± 1.0	40.5 ± 0.0	31.6 ± 2.8
LOF	10.1 ± 0.0	30.8 ± 0.0	17.1 ± 0.0	66.1 ± 0.0	53.3 ± 0.0	6.2 ± 0.0
CODiT	11.8 ± 0.8	48.4 ± 1.9	19.7 ± 0.7	45.9 ± 22.3	78.8 ± 0.8	47.1 ± 2.3
MEGA	11.8 ± 0.5	53.6 ± 2.6	**74.6 ± 1.7**	60.1 ± 4.8	78.8 ± 0.4	66.5 ± 3.2
DCDetector	32.7± 26.9	50.1 ± 0.9	54.1 ± 28.7	51.2 ± 0.6	17.9 ± 34.6	50.5 ± 0.5
AnomalyTransformer	35.5 ± 20.1	61.3 ±12.5	72.7 ± 6.1	79.9 ± 1.4	**96.5 ± 0.3**	83.0 ± 4.7
TranAD	23.1 ± 0.3	61.6 ± 1.5	67.9 ± 0.6	81.5 ± 0.4	86.3 ± 1.0	61.6 ± 0.5
GANF	54.6 ± 2.1	77.8 ± 0.6	29.3 ± 16.5	45.9 ± 22.4	73.1 ± 10.7	64.0 ± 3.3
MTGFlow	41.5 ± 15.0	77.7 ± 2.7	52.4 ± 13.8	78.4 ± 8.2	63.5 ± 4.3	59.2 ± 5.0
Ours	**65.3 ± 7.4**	**80.9 ± 1.5**	63.6 ± 0.6	**81.8 ± 0.5**	77.6 ± 3.9	**90.5 ± 1.0**

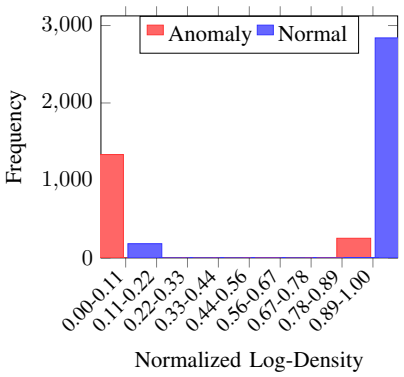


Fig. 3: Distribution of normalized log-densities on the PMU test dataset.

models (AnomalyTransformer, TranAD, DCDetector) exhibit substantial degradation in this setting, often reconstructing contaminated samples as if they were normal. Density-based models (GANF, MTGFlow) perform more reliably under contamination but still fall short of WENFlow, highlighting the benefit of multi-scale decomposition and selective attention in capturing discriminative structure.

On WADI, WENFlow attains the best AUC and competitive F1 performance, indicating that it can detect both isolated events and multi-sensor faults. On the PMU dataset, which includes high-frequency transients and multi-stage cascading failures, WENFlow achieves a notably high AUC of 90.5, outperforming MTGFlow and significantly surpassing reconstruction-based approaches. This reinforces the hypothesis that modeling timestamp-level densities conditioned on spatiotemporal context is effective for CPS domains where operational patterns vary sharply across timescales.

Figure 3 further illustrates WENFlow's class separation through the distribution of normalized log-densities on the PMU test set. Most anomalous windows cluster sharply in low-density regions, while normal windows occupy a higher-density regime. WENFlow occasionally assigns higher densities to brief point anomalies, which is expected given the relative extremity of cascading-failure signatures; however, these constitute a small fraction of the anomaly set and do not impede overall separability.

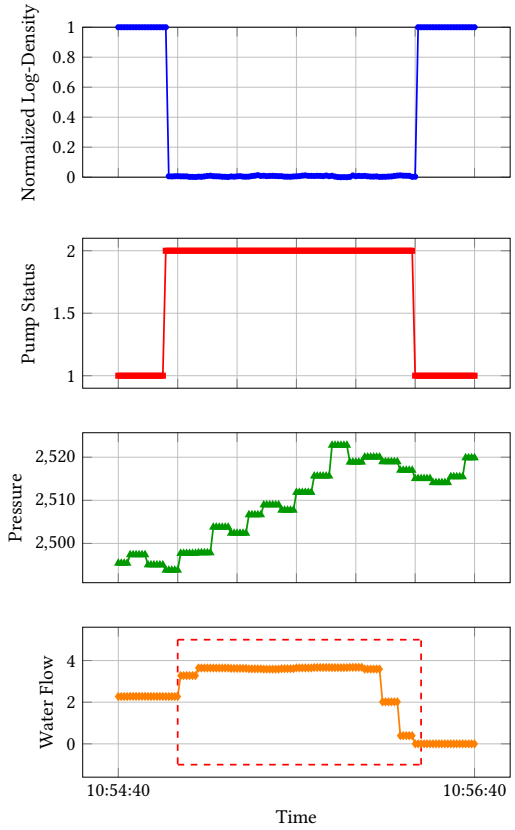


Fig. 4: Timestamp-level Normalized Log-Density scores (top) generated during an attack targeting a water pump in the WADI testbed. The red dashed box in the bottom plot indicates the period of time in which the attack took place.

To provide a fine-grained view, Figure 4 shows timestamp-level log-densities during a representative WADI attack in which a water pump is maliciously activated to induce a pipe burst. As a result of this action, nearby sensors measure abnormal changes in water pressure and water flow. The anomalous interval is clearly isolated in the low-density region, demonstrating that WENFlow produces interpretable, timestamp-resolved indicators of abnormal behavior.

TABLE V: Computational Efficiency results. Inference time (in milliseconds/sample), number of model parameters and GPU memory usage (in megabytes) are shown. Standard deviation is shown for inference time. GPU memory usage accounts for model size, input data, forward activations of the model, CUDA contexts, and reserved memory. Results for the best-performing models are shown in bold and results for the second-best models are underlined.

Data	Metric	CODiT	MEGA	DCDetector	AnomalyTransformer	TranAD	GANF	MTGFlow	**Ours**
SWaT	Parameters	1847191	67210	867372	4840516	225519	**9826**	20806	13517
	GPU Memory	794	4068	7191	3047	5077	2048	2361	**278**
	Infer Time	798.5 ± 53.0	0.07 ± 0.0	**0.02 ± 0.0**	**0.02 ± 0.0**	0.30 ± 0.01	0.07 ± 0.0	0.09 ± 0.0	0.07 ± 0.0
WADI	Parameters	4568791	95002	948347	5002387	1270023	**12036**	23016	20909
	GPU Memory	794	4404	7516	7757	10898	5341	5375	**262**
	Infer Time	783.6 ± 18.3	0.2 ± 0.0	0.06 ± 0.0	**0.02 ± 0.0**	0.90 ± 0.01	0.30 ± 0.05	0.30 ± 0.0	0.04 ± 0.0
PMU	Parameters	3661591	84580	907616	4947064	778704	**3204**	23016	36769
	GPU Memory	794	4068	6780	4343	9376	4238	4244	**243**
	Infer Time	134.8 ± 21.5	0.10 ± 0.0	0.16 ± 0.01	0.20 ± 0.01	0.60 ± 0.0	0.13 ± 0.04	0.20 ± 0.01	**0.03 ± 0.0**

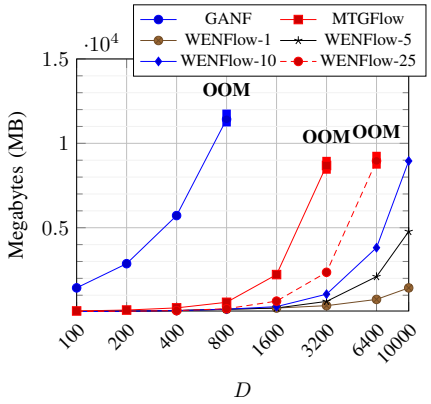


Fig. 5: WENFlow's memory usage for increasing values of D compared to flow-based methods with quadratic scaling. Each WENFlow-p model uses a different percentage ($p\%$) of D for defining top k selection. Vertical bold lines denote the point at which a model encounters an Out-of-Memory (OOM) error for higher values of D.

TABLE VI: Ablation results including AUC, parameter count, and inference time (ms/sample).

Model	**AUC**	**Params**	**Infer.**
PMU			
WENFlow	90.5	36,769	0.030
WF\A	73.9	31,488	0.020
WF\W	85.5	36,769	0.020
RNVP	76.3	14,848	0.002
WADI			
WENFlow	81.8	20,909	0.040
WF\A	72.8	16,764	0.040
WF\W	74.4	20,909	0.040
RNVP	70.9	8,940	0.020
SWaT			
WENFlow	80.9	13,517	0.070
WF\A	76.5	8,412	0.070
WF\W	77.6	13,517	0.070
RNVP	75.6	4,044	0.010

B. Model Efficiency and Scalability

We next evaluate efficiency and scalability, addressing our third hypothesis that WENFlow scales linearly in D and is practical for high-dimensional CPS. Table V summarizes inference latency (milliseconds/sample), GPU memory usage (MB), and parameter counts for all deep-learning methods. We use a common window size $W = 64$ across all models. Transformer-based models achieve fast inference, but they require substantially more GPU memory—often several gigabytes—because full temporal self-attention scales quadratically with sequence length. In contrast, WENFlow achieves real-time inference speeds (e.g., 0.03 ms/sample on the PMU dataset) and is the fastest flow-based model, owing to its attention mechanism that avoids dense feature-wise attention.

MTGFlow uses fully-connected feature-wise self-attention, exhibiting quadratic complexity in D, and resulting in slower inference and significantly higher memory consumption on high-dimensional datasets such as WADI. CODiT incurs the highest inference cost among all methods because it requires a separate model evaluation per sample, followed by multiple p-value computations for conformal prediction. Nevertheless, CODiT's memory footprint is relatively small because most conformal computations occur on the CPU. WENFlow, however, combines both low memory usage and fast GPU execution; on WADI, for example, it uses only about 2.4% of the memory required by TranAD.

To further assess scalability, we analyze GPU memory usage for large D (Figure 5). We compare GANF and MTGFlow with WENFlow variants in which k is set as a fraction of D (WENFlow-p with $k = \frac{p}{100}D$). GANF's graph-conditioned flow and MTGFlow's fully-connected attention both scale quadratically in D, leading to prohibitive memory demands. WENFlow-25 encounters out-of-memory errors beyond $D \geq 10000$ because k grows proportionally with D, but this setting is unrealistic for CPS, where only a small number of sensors typically initiate an anomaly. For small fixed k, WENFlow exhibits near-perfect linear scaling; for example, WENFlow-1 processes batches with $D = 10000$ using under 2000 MB of GPU memory. In practice, k is chosen such that $k \ll D$ and is independent of system dimension, ensuring that WENFlow maintains linear complexity and remains scalable.

C. Ablation and Sensitivity Analyses

To evaluate our fourth hypothesis—that each major component of WENFlow contributes meaningfully to overall perfor-

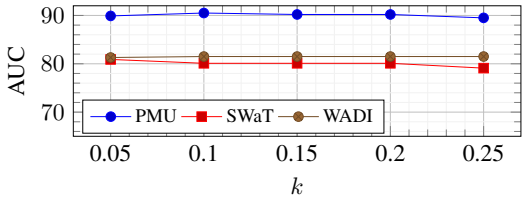


Fig. 6: Sensitivity analysis of k.

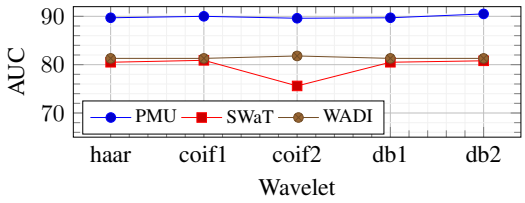


Fig. 7: Sensitivity analysis of wavelet type.

mance—we conduct a series of ablation and sensitivity studies across all three datasets. Table VI reports AUC, parameter counts, and inference latency for WENFlow and three ablated variants: WF\A (removing gated selective self-attention), WF\W (removing wavelet decomposition), and RNVP (an unconditional flow). WENFlow achieves the highest AUC on every dataset, indicating that its full architecture is needed for strong class separability. On the PMU dataset, the largest degradation occurs when removing attention, confirming that selective spatial modeling is critical for distinguishing cascading failures. Removing DWT also reduces performance across all datasets, demonstrating the value of multi-scale decomposition for separating slow trends from high-frequency disturbances. Importantly, incorporating DWT and attention does not significantly increase model size or inference time, and RNVP consistently underperforms due to its inability to capture spatiotemporal dependencies.

We further analyze the effect of key hyperparameters, including the top-k ratio, the choice of wavelet basis, and the temporal window size. Figure 6 shows that AUC remains stable across a wide range of k values, with many of the best-performing configurations corresponding to smaller k. This insensitivity supports the scalability of WENFlow, as strong performance can be maintained even when only a small subset of sensors is selected for attention—consistent with fault patterns in CPS, where anomalies typically originate from a limited set of components.

In Figure 7, we examine the effect of different wavelet bases, including Haar, Coiflet, and Daubechies bases. These wavelets differ in the number of vanishing moments and smoothness, with Haar providing highly efficient piecewise-constant approximations and Daubechies and Coiflets offering rich multi-scale structure. Across all datasets, WENFlow's detection performance varies only marginally with wavelet choice, with a small decrease on SWaT using the coif2 basis. This robustness suggests that the model is flexible with respect to the specific wavelet family used for DWT.

Finally, Figure 8 shows that WENFlow maintains high AUC across a broad range of window sizes, with dataset-specific

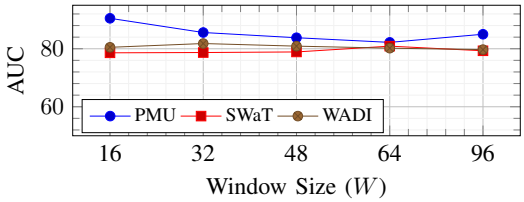


Fig. 8: Sensitivity analysis of the window size W.

optima reflecting underlying process timescales. Together, these ablation and sensitivity studies demonstrate that WENFlow's components—wavelet decomposition, gated selective attention, and conditional flows—are complementary, and that the model is stable across a wide range of hyperparameters relevant to high-dimensional CPS.

D. Interpretability Analysis

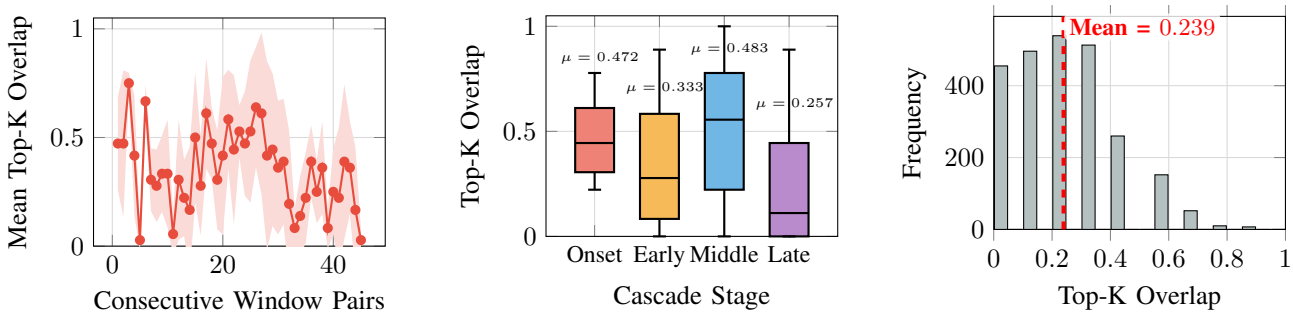
To evaluate our fifth hypothesis—that WENFlow's top-k feature selection mechanism provides interpretable insight into how anomalies originate and propagate—we analyze its behavior on the PMU dataset during cascading failures. For each window, WENFlow selects a set of k important features; the degree to which these sets persist over time reflects how stable or localized an anomalous event is. To quantify this, we compute the Top-k Overlap (TKO) between two consecutive windows $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(i+1)}$:

$$TKO = \frac{|Top\text{-}k(\mathbf{X}^{(i)}) \cap Top\text{-}k(\mathbf{X}^{(i+1)})|}{k}.$$

A high TKO indicates that the model consistently identifies the same sensors as important across adjacent windows, suggesting a persistent or localized anomaly. Conversely, low TKO values indicate nonstationary behavior or unrelated events.

For each cascading-failure simulation, we separate the failure sequence from the preceding normal operation and inject isolated point anomalies into random sensors before the onset of failure. The injected anomalies affect a single sensor for only one window and do not persist across windows, ensuring that they do not artificially introduce temporal correlation. Isolated anomalies that naturally persist across multiple windows are excluded from the TKO analysis. We compute TKO only between consecutive windows.

Figure 9(a) presents the average TKO over 45 pairs of consecutive windows, aggregated across five simulations. At the onset of failure (windows 0–2), the model focuses on a localized set of sensors associated with the initial fault, yielding high TKO values. During the early stage (windows 3–15), power-flow changes propagate to nearby regions and the model redirects attention, leading to decreased TKO. Once these new regions stabilize, TKO increases again. In the middle stage (windows 16–30), a second fault occurs and the model shifts to a new persistent set of features. In the late stage (windows 31–45), system-wide instability develops, causing TKO to decline steadily until a blackout occurs. Figure 9(b) highlights these stages, illustrating WENFlow's ability to track the evolving spatial footprint of cascading failures.



(a) (b) (c)

Fig. 9: TKO trends for cascading failures in the PMU dataset. Results are averaged across five simulations. (a) Mean TKO across 45 consecutive time windows, including standard deviation. An initial fault begins at time window 0 and second fault occurs in the middle stage, leading to a system-wide blackout. (b) TKO distribution across cascade stages from onset (windows 0-2) through early (windows 3-15), middle (windows 16-30), and late phases (windows 30-45). (c) Histogram of TKO ratios for samples containing independent point anomalies in the same dataset. The vertical dashed line denotes the mean.

To verify that WENFlow does not falsely infer relationships between unrelated events, we compute TKO for window pairs containing only the injected anomalies. Figure 9(c) shows that these TKO values concentrate between 0.0 and 0.3, reflecting minimal overlap and confirming that WENFlow does not impose spurious temporal structure on independent disturbances. Taken together, these analyses demonstrate that WENFlow's top-k mechanism yields temporally coherent patterns during cascading failures, while correctly recognizing the independence of isolated point anomalies.

VI. Conclusion

This paper introduced WENFlow, a scalable and robust framework for MTS anomaly detection in CPS. Through conditional density estimation, WENFlow learns a stable representation of nominal CPS behavior even when the training data are partially contaminated, enabling strong class separability across various anomaly ratios. Empirically, WENFlow achieves the highest AUC on three CPS-related datasets, outperforming reconstruction-based and flow-based baselines that degrade under contaminated conditions or fail to capture critical dependencies in sensor networks.

A core strength of WENFlow is its scalability. Unlike existing flow-based models that use fully-connected or graph-based spatial modeling—both of which scale quadratically in the number of features—WENFlow incorporates a selective attention mechanism that restricts spatial modeling to a small subset of adaptively chosen critical features. This design yields linear complexity in the feature dimension and results in substantial reductions in GPU memory usage and latency. In our experiments, the transformer-based methods require about 6.86 GB of GPU memory on average, while WENFlow only requires about 261 MB. WENFlow also maintains fast inference speeds, rivaling transformers in real-time efficiency.

Our interpretability analyses further demonstrate that WENFlow produces meaningful explanations. The top-k feature selection mechanism highlights sensors responsible for initiating and propagating anomalies, and the resulting Top-k Overlap trajectories capture the temporal evolution of cascading failures. Moreover, WENFlow maintains low overlap on isolated point anomalies, correctly identifying them as unrelated events. These findings illustrate WENFlow's ability not only to detect anomalies, but also to offer insight into the underlying structure of CPS faults.

Overall, WENFlow provides robustness to contaminated data, linear scalability, efficient inference, and interpretable reasoning—properties required for monitoring real-world CPS. WENFlow outperforms reconstruction-based methods that assume that training data is anomaly-free in the unsupervised setting, and flow-based models that do not scale to high-dimensional CPS. For future work, adaptive thresholding and resolving problems with partial observability (missing sensor readings) could further improve model robustness in the unsupervised setting. Finally, learnable wavelet transforms could improve adaptability across CPS domains.

Acknowledgment

This work is based upon the work sponsored in part by the National Science Foundation under grants CNS-2238815, CNS-1840052, CNS-1952011 and CMMI-2527359. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. Z. Islam, Y. Lin, V. M. Vokkarane, and V. Venkataramanan, "Cyber-physical cascading failure and resilience of power grid: A comprehensive review," *Frontiers in Energy Research*, vol. 11, 02 2023. [Online]. Available: https://www.osti.gov/biblio/1973774

[2] S. Gharebaghi, N. R. Chaudhuri, T. He, and T. F. L. Porta, "Dynamic modeling and mitigation of cascading failures in power grids with interdependent cyber and physical layers," *IEEE Transactions on Smart Grid*, vol. 15, no. 3, pp. 3235–3247, 2024.

[3] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "Wadi: a water distribution testbed for research in the design of secure cyber physical systems," in *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, 2017, p. 25–28.

[4] V. S. Kumar, T. Wang, K. S. Aggour, P. Wang, P. J. Hart, and W. Yan, "Big data analysis of massive pmu datasets: A data platform perspective," in *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2021, pp. 1–5.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, vol. 30, 2017.

[6] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems*, 2021.

[7] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, vol. 35, no. 12. AAAI Press, 2021, pp. 11 106–11 115.

[8] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *International Conference on Learning Representations*, 2022.

[9] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data," in *Proceedings of VLDB*, vol. 15, no. 6, 2022, pp. 1201–1214.

[10] Y. Yang, C. Zhang, T. Zhou, Q. Wen, and L. Sun, "Dcdetector: Dual attention contrastive representation learning for time series anomaly detection," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, p. 3033–3045.

[11] A. A. Alwan, M. A. Ciupala, A. J. Brimicombe, S. A. Ghorashi, A. Baravalle, and P. Falcarin, "Data quality challenges in large-scale cyber-physical systems: A systematic review," *Information Systems*, vol. 105, p. 101951, 2022.

[12] G. Bovenzi, A. Foggia, S. Santella, A. Testa, V. Persico, and A. Pescapé, "Data poisoning attacks against autoencoder-based anomaly detection models: a robustness analysis," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 5427–5432.

[13] G. Bovenzi, G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "Network anomaly detection methods in iot environments via deep learning: A fair comparison of performance and robustness," *Computers & Security*, vol. 128, p. 103167, 2023.

[14] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *International Conference on Learning Representations*, 2017.

[15] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] E. Dai and J. Chen, "Graph-augmented normalizing flows for anomaly detection of multiple time series," in *International Conference on Learning Representations*, 2022.

[17] Q. Zhou, J. Chen, H. Liu, S. He, and W. Meng, "Detecting multivariate time series anomalies with zero known label," in *Proceedings of the AAAI Conference on Artificial Intelligence*, no. 4, 2023, pp. 4963–4971.

[18] R. Laxhammar and G. Falkman, "Sequential conformal anomaly detection in trajectories based on hausdorff distance," in *14th International Conference on Information Fusion*, 2011, pp. 1–8.

[19] R. Kaur, K. Sridhar, S. Park, Y. Yang, S. Jha, A. Roy, O. Sokolsky, and I. Lee, "Codit: Conformal out-of-distribution detection in time-series data for cyber-physical systems," in *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems*, 2023, p. 120–131.

[20] J. Wang, S. Shao, Y. Bai, J. Deng, and Y. Lin, "Multiscale wavelet graph autoencoder for multivariate time-series anomaly detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023.

[21] A. P. Mathur and N. O. Tippenhauer, "Swat: a water treatment testbed for research and training on ics security," in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, 2016, pp. 31–36.

[22] K. Yang, S. Kpotufe, and N. Feamster, "An efficient one-class SVM for novelty detection in iot," in *Transactions on Machine Learning Research*, 2022.

[23] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, p. 93–104, may 2000.

[24] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*, 2008, pp. 413–422.

[25] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," *ACM Comput. Surv.*, vol. 57, no. 1, Oct. 2024.

[26] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," 2016.

[Online]. Available: https://arxiv.org/abs/1607.00148

[27] Y. Jeong, E. Yang, J. H. Ryu, I. Park, and M. Kang, "Anomalybert: Self-supervised transformer for time series anomaly detection using data degradation scheme," 2023. [Online]. Available: https://arxiv.org/abs/2305.04468

[28] S. Avdaković and N. Čišija, "Wavelets as a tool for power system dynamic events analysis – state-of-the-art and future applications," *Journal of Electrical Systems and Information Technology*, vol. 2, no. 1, pp. 47–57, 2015.

[29] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[30] B. K. Pandey, H. Tiwari, and D. Khare, "Trend analysis using discrete wavelet transform (dwt) for long-term precipitation (1851–2006) over india," *Hydrological Sciences Journal*, vol. 62, no. 13, pp. 2187–2208, 2017.

[31] R. H. G. Tan and V. K. Ramachandaramurthy, "Power system transient analysis using scale selection wavelet transform," in *TENCON 2009*, 2009, pp. 1–6.

[32] I. Daubechies, *Ten Lectures on Wavelets*. SIAM, 1992, vol. 61.

[33] Y. Yang, T. Nishikawa, and A. E. Motter, "Small vulnerable sets determine large network cascades in power grids," *Science*, vol. 358, no. 6365, p. eaan3184, 2017. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aan3184

[34] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *J. Mach. Learn. Res.*, vol. 22, no. 1, jan 2021.

VII. Appendix

A. Normalizing Flows

Our approach is inspired by advances in modeling probability densities using normalizing flows [15]. Normalizing flows map a complex data distribution $p_x(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^D$, to a known probability distribution $p_z(\mathbf{z})$ where $\mathbf{z} \in \mathbb{R}^D$ (e.g., multi-variate Gaussian). This transformation is done using a sequence of invertible and differentiable functions, $f = (f_1.f_2 \cdots f_N)$, where $\mathbf{z} = f(\mathbf{x})$ and $\mathbf{x} = f^{-1}(\mathbf{z})$. Here, f is the "normalizing direction" that enables density estimation, providing the likelihood of a real-world observation. Using the Jacobian matrix $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$, the change of variables technique can be applied for evaluating $p_x(\mathbf{x})$. The log-density of $\mathbf{x}$ is then given as

$$\log(p_x(\mathbf{x})) = \log(p_z(f(\mathbf{x}))) + \log\left|\det \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right| \quad (1)$$

In practice, f can be any transformation that is invertible and has a tractable Jacobian determinant. While many flows exist in the literature [34], we focus on RealNVP [14], which implements real-valued non-volume preserving affine transformations in the form of affine coupling layers. Each coupling layer takes the form

$$\mathbf{y}_{1:d} = \mathbf{x}_{1:d} \quad (2)$$

$$\mathbf{y}_{d+1:D} = \mathbf{x}_{d+1:D} \odot \exp(s(\mathbf{x}_{1:d})) + t(\mathbf{x}_{1:d}) \quad (3)$$

where d is used to partition the features and $s(\cdot)$ and $t(\cdot)$ denote scale and translation functions approximated by multi-layer perceptrons (MLPs). RealNVP enables tractable computation of the log-determinant of the Jacobian, making it easy to stack multiple coupling layers together to define an invertible flow.

Recently, many works have explored conditional normalizing flows for improving MTS anomaly detection [16, 17]. These works learn an embedding vector $\mathbf{c} \in \mathbb{R}^d$ that can capture spatiotemporal dependencies in the MTS and be conditioned on within the normalizing flow. Given a conditional normalizing flow $f : \mathbb{R}^D \times \mathbb{R}^d \to \mathbb{R}^D$, we can simply augment Equation 1 to include the conditional variable:

$$\log(p_x(\mathbf{x};\mathbf{c})) = \log(p_z(f(\mathbf{x};\mathbf{c}))) + \log\left|\det \frac{\partial f(\mathbf{x};\mathbf{c})}{\partial \mathbf{x}}\right| \quad (4)$$

Using conditional normalizing flows, we can better estimate the density of a time series by capturing historical context and dependencies. Given a time series $\mathbf{X} = [\mathbf{x}_1, \mathbf{x_2}, ..., \mathbf{x}_N]$ where $\mathbf{x} \in \mathbb{R}^D$ and N is the length of the time series, one can use the chain rule of probability to obtain $p(\mathbf{X}) = p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1)...(\mathbf{x}_N|\mathbf{x}_{<N})$, where $\mathbf{x}_{<N}$ denotes dependencies on all variables before N. The conditional probability $p(\mathbf{x}_n|\mathbf{x}_{<n})$ can then be reparameterized by setting $\mathbf{x}_{<n} = \mathbf{c}_n$ where $\mathbf{c}_n$ denotes historical information from previous timesteps and can be computed using deep neural networks.

B. Self-attention

Self-attention [5] enables neural networks to dynamically capture important dependencies between elements in a sequence. Computing self-attention requires learning matrices $\mathbf{W}^Q \in \mathbb{R}^{model_d \times key_d}$, $\mathbf{W}^K \in \mathbb{R}^{model_d \times key_d}$, $\mathbf{W}^V \in \mathbb{R}^{model_d \times value_d}$ which represent query, key, and value weights, respectively. Here, we let $model_d$ refer to the model dimension, key_d refer to the dimension of keys and queries, and $value_d$ refer to the dimension of values. Given a multivariate time series $\mathbf{X} \in \mathbb{R}^{T \times model_d}$ where T denotes the length of the time series, it is straightforward to calculate the self-attention matrix, $\mathbf{H} \in \mathbb{R}^{T \times value_d}$. First, we calculate $\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \mathbf{K} = \mathbf{X}\mathbf{W}^K, \mathbf{V} = \mathbf{X}\mathbf{W}^V$, which are key, query, and value projections, respectively. Then, a pair-wise attention score matrix $\mathbf{A} \in \mathbb{R}^{T \times T}$ is calculated by

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}(\mathbf{K})^\top}{\sqrt{key_d}}\right) \quad (5)$$

The self-attention matrix is then calculated by $\mathbf{H} = \mathbf{A}\mathbf{V}$. Temporal self-attention has shown success at capturing long-term temporal dependencies in time series and improving anomaly detection in MTS [8, 10, 9]. Alternatively, feature-wise self-attention (performing self-attention on $\mathbf{X}^\top$) has shown success at capturing spatial dependencies as it computes attention scores between features instead of timesteps [17].

C. Discrete Wavelet Transform Example

Discrete wavelet transform decomposes a signal into low-frequency and high-frequency components, allowing localized transients and global trends to be analyzed separately. The approximation coefficients capture the slow, low-frequency behavior of each sensor, while the detail coefficients highlight fast, high-frequency changes such as spikes, abrupt disturbances, or fault-induced transients.

Figure 10 shows an example. Panel (a) presents an MTS with 12 sensors and an anomalous event beginning at the red vertical line. The corresponding detail coefficients in panel (b) show sharp magnitude increases near the event, indicating localized, transient behavior. In contrast, the approximation coefficients in panel (c) track the smoother, long-term evolution of the signals. These complementary representations form the inputs to WENFlow's temporal and spatial components: $\mathbf{A}$ provides stable context, while $\mathbf{D}$ identifies sensors exhibiting fault-relevant activity.

D. PMU Dataset Generation

The PMU dataset is generated from an IEEE-39 bus power system modeled in a real-time digital simulator (RTDS) (see fig. 1). The system has 10 dynamic generator machine models and 18 loads, out of which 6 loads are varied every 10 seconds using a `load scheduler` component to mimic normal load changes. A total of 8 firmware `PMUs` are placed at buses $2, 5, 6, 10, 19, 22, 29$ and 39. The entire case is set to run on three Novacor cores on two racks using Inter Rack Communication (IRC) channel. The training dataset is mostly representative of normal operating conditions and includes two small fault events. The test dataset is split across five cascading failure simulations, and each simulation contains various point anomalies (gaussian noise injections) occurring before the start

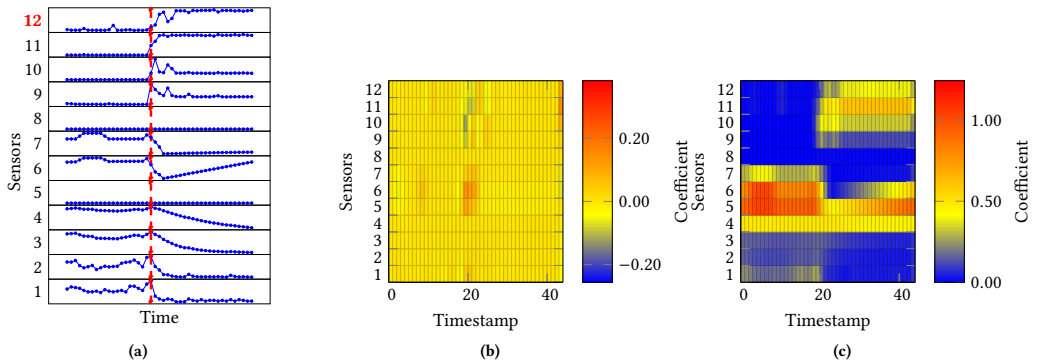


Fig. 10: (a) Example multivariate time series with 12 sensors; the red line marks the start of an anomalous event. (b) Detail coefficients and (c) approximation coefficients obtained from a one-level undecimated discrete wavelet transform.

of the cascading failure. For example, one of the cascading failures across multiple PMUs in a window is as follows: back-to-back bus faults at bus 16 at 300.01s and 300.34s, followed by tripping of line $16-17$ at 302.23s, line $16-15$ at 204.61s, lines $16-21$ and $16-23$ at 308.26s and lines $10-13, 4-5$ and $3-2$ at 311.22s. This results in multiple generators going out-of-synchronization, resulting in complete transient instability.

Frequency measurements and derivative of frequency (ROCOF) are generally less accurate than voltage measurements during dynamic events, as frequency changes more rapidly in response to power imbalances and disturbances. Unlike voltage, which is a directly measured value and estimated as part of the phasor, frequency is derived from phase angle differences, making it susceptible to errors from noise and transients. The filtering techniques used to reduce noise can also introduce lag. Thus, we rely on 3-phase voltage magnitudes, voltage phase angles, current magnitudes, and current phase angles as features for each PMU.

E. Additional Detection Results

In fig. 11, ROC curves are shown for each dataset. Each plot shows the false positive rate compared to the true positive rate when evaluated at different thresholds. The best models are those that have ROC curves with a higher AUC. For reference, a random classifier (shown by the diagonal line) has an AUC of 0.5. We present other classification metrics such as precision and recall in Table VII.

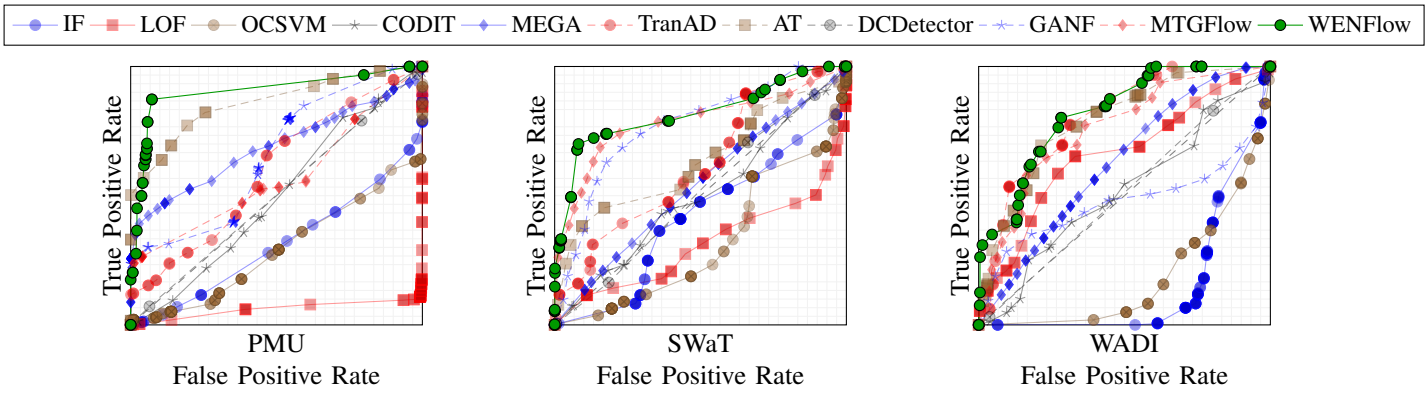


Fig. 11: ROC curves for 2 real-world CPS datasets and the PMU dataset.

TABLE VII: Precision and Recall (%) scores (using same thresholds as Table IV).

Data	SWAT		WADI		PMU	
Metric	Precision	Recall	Precision	Recall	Precision	Recall
OCSVM	30.1 ± 0.0	88.7 ± 0.0	5.2 ± 0.0	100.0 ± 0.0	31.9 ± 0.0	96.3 ± 0.0
IF	13.8 ± 3.1	98.3 ± 0.9	18.6 ± 2.2	100.0 ± 0.0	25.4 ± 0.0	100.0 ± 0.0
LOF	5.3 ± 0.0	100.0 ± 0.0	9.3 ± 0.0	100.0 ± 0.0	36.9 ± 0.0	96.0 ± 0.0
CODiT	6.3 ± 0.5	100.0 ± 0.0	12.6 ± 2.2	100.0 ± 0.0	65.0 ± 1.0	100.0 ± 7.6
MEGA	65.0 ± 0.5	99.0 ± 0.0	59.6 ± 2.2	100.0 ± 0.0	13.4 ± 6.0	99.9 ± 0.0
DCDetector	38.0 ± 39.4	81.0 ± 24.2	61.4 ± 31.0	50.0 ± 28.8	54.7 ± 45.2	16.3 ± 31.9
AnomalyTransformer	24.4 ± 18.8	98.2 ± 3.6	60.0 ± 4.4	92.3 ± 9.4	98.8 ± 0.7	94.2 ± 0.3
TranAD	13.0 ± 0.2	100.0 ± 0.0	58.7 ± 0.9	80.6 ± 0.0	81.9 ± 0.2	91.2 ± 1.9
GANF	40.2 ± 3.1	86.6 ± 8.2	27.6 ± 14.5	34.0 ± 21.9	82.4 ± 9.1	70.0 ± 18.4
MTGFlow	27.5 ± 11.8	98.2 ± 1.5	47.7 ± 14.5	60.5 ± 19.0	83.2 ± 4.3	51.7 ± 6.0
WENFlow	55.6 ± 3.1	80.1 ± 16.3	52.5 ± 0.8	80.5 ± 0.0	67.0 ± 1.0	92.1 ± 1.1