Highlights

Cluster Aggregated GAN (CAG): A Cluster-Based Model for Appliance-Level Load Pattern Generation

Zikun Guo,Adeyinka.P. Adedigba,Rammohan Mallipeddi

- A dual-branch generative framework that adaptively routes appliances as intermittent or continuous, aligning model capacity with device behavior.
- Shape-based segment clustering enables motif-level fidelity, improved diversity, and interpretable synthetic load generation.
- Sequence simplification and reconstruction stabilize adversarial training for long-horizon continuous loads while preserving temporal realism.
- The proposed Cluster Aggregated GAN (CAG) achieves superior realism, diversity, and convergence stability across heterogeneous appliance types.

Cluster Aggregated GAN (CAG): A Cluster-Based Model for Appliance-Level Load Pattern Generation

Zikun Guo[a], Adeyinka.P. Adedigba[a] and Rammohan Mallipeddi[a,*]

[a]*Department of Artificial Intelligence, School of Electronics Engineering, Kyungpook National University, 80 Daehak-ro, Buk-gu, Daegu, 41544, Republic of Korea*

ARTICLE INFO

Keywords:
Generative adversarial networks
non-intrusive load monitoring
Pattern generation
Clustering
LSTM

ABSTRACT

High quality appliance load data are crucial for developing and evaluating non-intrusive load monitoring systems, energy analytics, and demand side management. However, the scarcity of labeled appliance datasets severely limits the scalability and reproducibility of such research. While recent generative adversarial network (GAN) based frameworks have demonstrated the ability to synthesize power traces, most existing approaches treat all devices uniformly within a single generative model, neglecting the inherent behavioral heterogeneity between intermittent and continuous appliances. This often leads to unstable training dynamics and limited fidelity in generated load patterns. In this paper, we propose Cluster Aggregated GAN (CAG), a cluster aware, dual branch generative framework for appliance load pattern generation. CAG employs a lightweight statistical detector to classify appliances as intermittent or continuous and adaptively routes each device type to a behavior specific generative branch. For intermittent appliances, CAG performs segment clustering of usage motifs and trains independent lightweight GANs for each cluster, ensuring motif level fidelity and diversity. For continuous appliances, CAG leverages a simplified sequence LSTM-GAN to model long term temporal dependencies while maintaining training stability through sequence compression and reconstruction. Comprehensive experiments on the UVIC smart plug dataset demonstrate that CAG consistently outperforms conventional baselines across realism, diversity, and convergence stability metrics. The results highlight that integrating clustering as an active generative component, rather than a preprocessing step, significantly enhances both interpretability and scalability in synthetic load generation. CAG thus provides a robust and behavior aligned generative paradigm for data augmentation and privacy preserving NILM research.

1. Introduction

High quality synthetic appliance level load data have become increasingly important for evaluating non-intrusive load monitoring (NILM) algorithms, stress testing energy analytics pipelines, and enabling privacy preserving data sharing [14]. However, obtaining sufficient real world appliance level data is challenging the collection of labeled usage traces is labor intensive and often invasive, making data scarcity a major bottleneck for NILM research [12]. As a result, researchers have turned to generating realistic synthetic data as a promising alternative [4]. Realistic generation remains difficult due to the heterogeneity of appliance behavior. In particular, intermittent devices like laptops, refrigerators often exhibit bursty on-off activations, mode switching, and short transients, whereas continuous devices (e.g., displays, printers) produce long duration, gradually varying consumption trajectories. A single generative model seldom captures both of these regimes effectively, since each category of appliance demands a different modeling approach. Recent GAN based frameworks for synthetic load generation demonstrate the feasibility of creating appliance power traces in quantity [15]. However, such models do not explicitly distinguish between intermittent and continuous usage patterns, potentially limiting their ability to reproduce the full diversity of appliance load profiles. To address these challenges, we introduce CAG, a dual-branch framework for appliance level time series generation. The basic framework can be summarized in Figure 1. In our approach, a lightweight statistical detector first classifies each appliance load profile as either intermittent or continuous based on simple sparsity and variability metrics. Each appliance is then routed to an appropriate generative model tailored to its behavior: motif conditioned segment level GANs for intermittent devices, and sequence level LSTM based GANs for continuous devices. CAG thus assigns model capacity in alignment with

*Corresponding author
Email addresses: gzk798412226@gmail.com (Z. Guo); yinkpeace@gmail.com (Adeyinka.P. Adedigba); mallipeddi.ram@gmail.com (R. Mallipeddi)
ORCID(s):

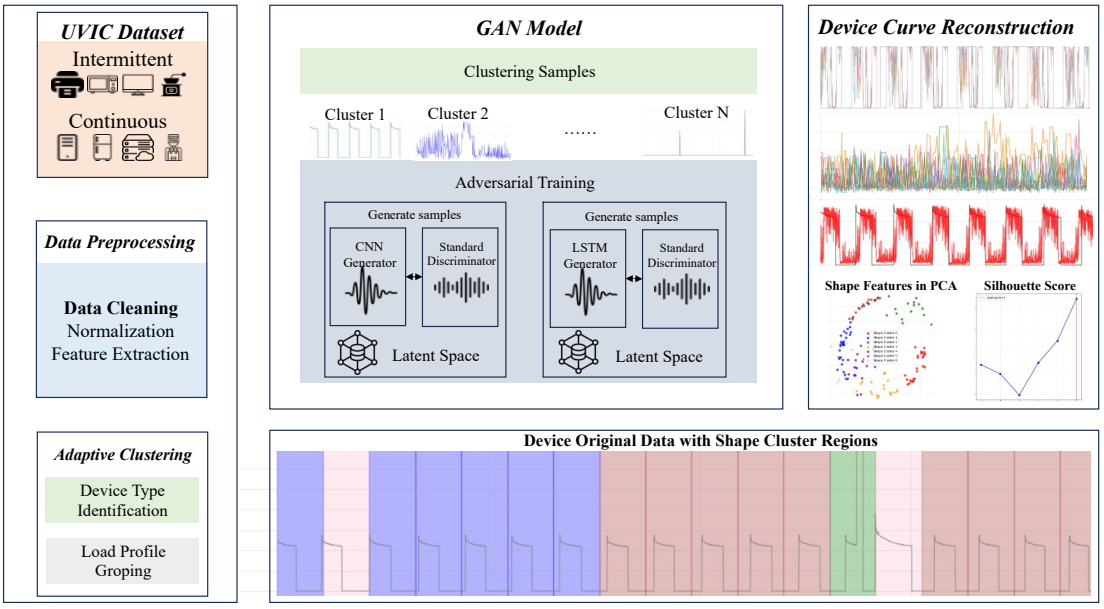

Figure 1: Overall workflow of the proposed Cluster Aggregated GAN (CAG) framework for appliance load generation. The UVIC dataset provides raw appliance level power consumption data, which are categorized into intermittent and continuous devices through a lightweight statistical classifier. Following data preprocessing including normalization, feature extraction, and profile grouping the signals are adaptively clustered based on their temporal and morphological similarity. Each cluster represents a distinct load motif and is modeled independently through adversarial training: CNN-based GANs are used for intermittent segments, and LSTM-based GANs for continuous sequences. The discriminator enforces realism across both branches. Generated samples are then reconstructed into full device curves, with PCA shape features and silhouette scores used to visualize motif diversity and cluster separability. The bottom panel illustrates how original device data are decomposed into shape clusters, enabling interpretable motif synthesis and improved behavioral fidelity in the generated load profiles.

behavioral structure. For intermittent appliances, clustering segments into usage motifs allows a segment level GAN to capture bursty event patterns with greater fidelity. For continuous appliances, an LSTM driven sequence GAN operates on the full load trajectory, effectively handling long range temporal dynamics. This division of labour yields two complementary benefits. First, clustering and motif conditioning for intermittent loads exposes interpretable usage motifs and improves coverage of event level variability [6]. Second, simplifying the sequence modeling for continuous loads stabilizes GAN training and scales up the modeling of long range dynamics without sacrificing fidelity. We demonstrate the effectiveness of the CAG approach on a real world smart plug dataset, providing qualitative evidence of high fidelity in the generated load traces across a diverse range of appliances spanning both intermittent and continuous device types.

The contributions of this work are threefold. First, we develop a simple yet effective detector that automatically distinguishes intermittent versus continuous device profiles using sparsity and variability statistics computed from the appliance time series data. Second, we propose a novel dual-branch generative architecture that aligns model capacity with the underlying behavioral patterns of appliances. In particular, our design employs motif conditioned, segment level GANs for intermittent devices and simplified sequence LSTM GANs for continuous devices, ensuring that each class of appliance is modeled by a network suited to its characteristics. We also introduce a dataset preprocessing pipeline that extracts usage motifs for intermittent appliances, enhancing the interpretability and diversity of generated traces. Third, we validate the CAG framework on a real world smart plug dataset, demonstrating its ability to produce high fidelity synthetic appliance load data that captures the distinct characteristics of both intermittent and continuous devices. Our results highlight the advantages of the dual-branch approach in generating realistic load profiles, paving the way for improved NILM research and applications.

2. Related Work

Time series data generation has been extensively studied across various domains, including finance, healthcare, and energy systems. In the context of non-intrusive load monitoring (NILM), generating synthetic appliance level load profiles is crucial for addressing data scarcity and enhancing model training. Existing approaches can be broadly categorized into three main steams :

Generic GAN-based methods [2, 3, 7] time series synthesis. These treat all appliance signals uniformly, using single-path models to synthesize sequences without structural behavior differentiation. Early works like Trace-GAN [11] and RLP-GAN [16] leveraged single branch GAN architectures to capture general load patterns, focusing on overall fidelity and diversity. But they often struggle to model heterogeneous appliance behaviors effectively.

Cluster intergrated methods that segment appliance usage patterns into clusters, training separate generative models for each cluster to capture diverse behaviors. These integrate clustering at the input or label level to improve mode coverage or to generate category aware outputs. clustered daily user load curves and trained an ensemble of Bi-LSTM GANs for each cluster, demonstrating improved diversity and structural accuracy. Some of the works using cluster aggregated the attention to minimize the long sequence training cost problem like CAT [8] and IDS method [10]. Clustered daily consumption patterns and conditioned a GAN on cluster labels to synthesize residential loads. These approaches treat clustering as a data preprocessing step or as conditioning metadata. In contrast, our method performs shape based segment level clustering and assigns a dedicated lightweight GAN to each motif cluster. This fine grained clustering to generation pipeline directly improves motif level fidelity and prevents generator collapse by decoupling the complex multi modal distribution into tractable subproblems.

Architecture specialized methods [9] that design a number of various emerged differ in architectures to address the unique challenges of load synthesis like LSTM-GAN [1], RCGAN [5] are widely used for modeling long term dependencies in medical or energy sequences. CNN-based generators, including ConvTimeGAN [13] and vanilla CNN-baselines, are preferred in short sequence scenarios due to their stability and local pattern encoding. WaveGAN [17] originally designed for audio waveform synthesis, has been extended to power load curves. However, these models typically operate under a one size fits all framework without differentiating between intermittent and continuous loads.

Our CAG framework fills a critical methodological gap in appliance load synthesis by explicitly aligning model structure with the behavioral heterogeneity of electrical devices. Unlike prior approaches that employ monolithic GAN architectures or treat clustering as a preprocessing step, CAG integrates adaptive device routing, segment motif clustering, and dual-branch adversarial learning into a unified generation process. This alignment ensures that each appliance type is modeled using the most suitable generative mechanism convolutional motif GANs for bursty, intermittent loads and LSTM-based sequence GANs for smooth, continuous loads. Through this design, CAG achieves higher motif fidelity, as clustering isolates distinct operational patterns; improved temporal realism, as sequence simplification stabilizes long horizon adversarial dynamics. Enhanced behavioral coverage and contributes to balanced mode representation across devices. Empirical results on the UVIC dataset establishes a robust and scalable paradigm for synthetic load generation bridging the gap between behavior aware design and practical data augmentation for NILM and energy analytics research.

3. Method

The core idea of our framework is to align the generative process with the intrinsic behavioral heterogeneity of appliances. The model architecture can be found in Figure 2. Unlike traditional monolithic GANs that attempt to fit all load patterns through a single generator discriminator pair, CAG decomposes the complex generation task into multiple structured subproblems. Specifically, we treat each appliance time series as a composition of short term motifs and long term temporal trends. This decomposition enables the model to learn fine grained power usage patterns through segment clustering while maintaining global temporal coherence through sequence level modeling. By routing each appliance to a behavior appropriate generative branch motif conditioned segment GANs for intermittent devices and simplified sequence LSTM GANs for continuous devices effectively allocates model capacity where it is most needed. The algorithm 1 outlines the overall training pipeline of the proposed CAG framework.

3.1. Notation

For an appliance with raw series $x\_1 : T \in \mathbb{R}^T$, we form fixed-length segments of length L which is a fixed hyperparameter: $\{ s^{(j)} \in \mathbb{R}^L \}_{j=1}^{N}$ by non-overlapping slicing. Let $\Delta x\_t = x\_t - x\_t - 1$ and $\mathbb{1}[\cdot]$ denote the indicator.

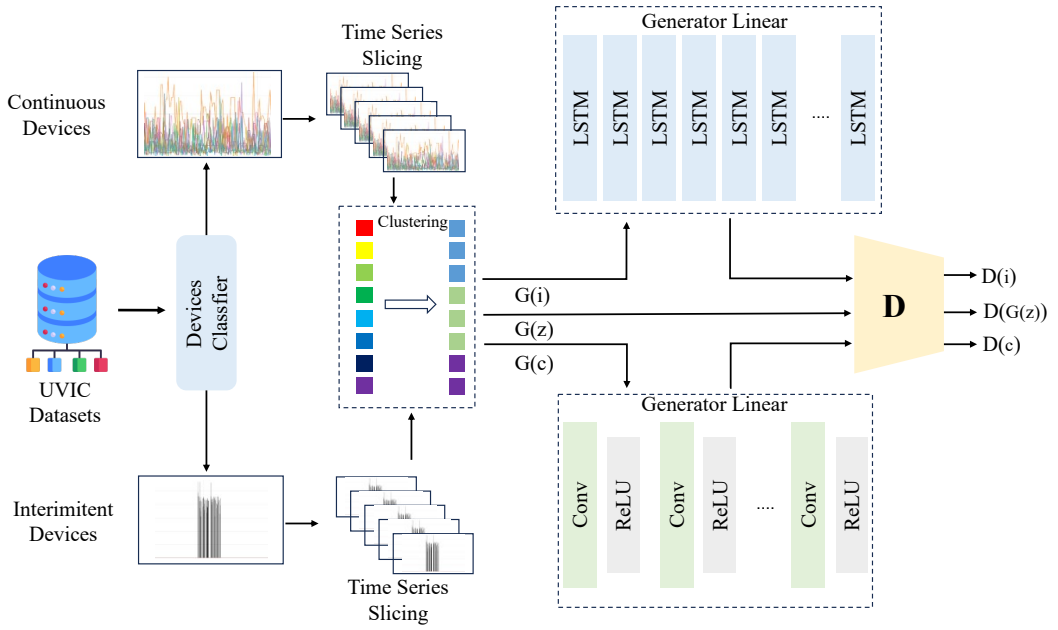

Figure 2: Overview of the proposed Cluster Aggregation GAN (CAG) framework for appliance level load pattern generation. UVIC dataset provides raw appliance power traces that are first classified by a lightweight statistical detector into intermittent and continuous device types. Continuous devices are modeled at the sequence level using an LSTM-based GAN, while intermittent devices are segmented into fixed length windows, clustered by shape features, and generated via independent lightweight CNN-based GANs for each motif cluster. The outputs from both branches are combined through a shared discriminator D, which evaluates the realism of generated samples $G(i)$, $G(z)$, and $G(c)$ corresponding to intermittent, clustered, and continuous data, respectively. This adaptive routing and cluster aware generation strategy ensures behavioral alignment, enhanced motif coverage, and stable adversarial training across heterogeneous appliance load patterns.

3.2. Adaptive routing

Device type is inferred via simple statistics computed on $x\_1 : T$:

$$\begin{aligned} R\_0 &:= \mathbb{1}\left[x\_1 = \cdots = x\_\min\{T\_0, T\} = 0 \right], \\ p\_\mathrm{nz} &:= \frac{1}{T} \sum \_t = 1^T \mathbb{1}[x\_t \neq 0], \\ \sigma^2\_\Delta &:= \frac{1}{T-1} \sum \_t = 2^T (\Delta x\_t - \overline{\Delta x})^2, \end{aligned} \tag{1}$$

with hyperparameters $T\_0 \in \mathbb{N}$ (prefix length), $\rho \in (0, 1)$ (nonzero occupancy threshold), and $\tau > 0$ (change-variance threshold). The routing rule is

$$\text{continuous if } R\_0 = 1 \text{ or } (p\_\mathrm{nz} > \rho \wedge \sigma^2\_\Delta < \tau), \quad \text{else intermittent.} \tag{2}$$

This detector is data-agnostic and suffices to route each device to a behaviour-appropriate model.

3.3. Intermittent branch: motif discovery and segment GANs

Each segment $s \in \mathbb{R}^L$ is normalised as

$$\tilde{s} = \frac{s - \mu}{\sigma}, \tag{3}$$

with μ, σ the segment mean and standard deviation (if σ=0, centring only). We compute a feature map $\phi : \mathbb{R}^L \to \mathbb{R}^d$ as the concatenation of:

- statistics: $\mu,\ \sigma,\ \mathrm{skew}(\tilde{s})=\frac{1}{L}\sum \tilde{s}^3,\ \mathrm{kurt}(\tilde{s})-3$;
- trend: slope $\beta=\mathrm{cov}(t,\tilde{s})/\mathrm{var}(t)$ for $t=1:L$;
- frequency: $f^*=\arg\max\_f \left|[\mathcal{F}(\tilde{s})](f)\right|$ from the DFT magnitude;
- morphology: peak/valley counts $\sum \mathbb{1}[\tilde{s}>0.5],\ \sum \mathbb{1}[\tilde{s}<-0.5]$;
- roughness/energy: $\mathrm{var}(\Delta\tilde{s}),\ \|\tilde{s}\|_2^2$;
- shape samples: $\left\{\tilde{s}\left(\lfloor\alpha^{(k)}(L-1)\rfloor\right)\right\}\_k=1^{20}$ for $\alpha^{(k)}=\frac{k-1}{19}$.

Features are standardised (z-scores) across segments and clustered by k-means:

$$\min\_\{c^{(j)}\},\{\mu^{(k)}\}\sum\_j=1^N\left\|\phi(s^{(j)})-\mu^{(c^{(j)})}\right\|^2,\qquad c^{(j)}\in\{1,\dots,K\}. \tag{4}$$

The number of clusters K is selected by maximising the mean silhouette score under an upper bound $K\le\kappa$:

$$\bar{s}(K)=\frac{1}{N}\sum\_j=1^N\frac{b(j)-a(j)}{\max\{a(j),\,b(j)\}}, \tag{5}$$

where $a(j)$ is the mean intra-cluster distance of j and $b(j)$ the minimum mean distance to other clusters. When silhouette is ill-defined, a small K is used. For cluster k, the segment-level dataset is $\mathcal{D}^{(k)}=\{s^{(j)}\ :\ c^{(j)}=k\}$.

Adversarial learning. Each $\mathcal{D}^{(k)}$ trains a lightweight fully connected GAN. Let $G(\cdot;\theta):\mathbb{R}^{d(z)}\to[-1,1]^L$ map latent $z\sim\mathcal{N}(0,I)$ to a segment, with a tanh output layer, and let $D(\cdot;\psi):[-1,1]^L\to(0,1)$ be a sigmoid discriminator; data are min–max normalised to $[-1,1]$ per cluster. With logistic losses, the objectives are

$$\mathcal{L}^{(D)}=\tfrac{1}{2}\,\mathbb{E}^{x\sim p^{\mathrm{real}}}\left[-\log D(x;\psi)\right]+\tfrac{1}{2}\,\mathbb{E}^{z\sim\mathcal{N}}\left[-\log\left(1-D(G(z;\theta);\psi)\right)\right], \tag{6}$$

$$\mathcal{L}^{(G)}=\mathbb{E}^{z\sim\mathcal{N}}\left[-\log D(G(z;\theta);\psi)\right], \tag{7}$$

optimised by Adam with step size η and moments $(\beta^{(1)},\beta^{(2)})$.

3.4. Continuous branch: simplification, LSTM GANs, and reconstruction

Long sequences are simplified to a surrogate $\hat{x}$ by uniform averaging with factor $F\ge 1$: partition $x\_1:T$ into blocks of size F (truncating the tail) and set

$$\hat{x}\_i=\frac{1}{F}\sum\_t=(i-1)F+1^{iF}x\_t,\quad i=1:\left\lfloor\frac{T}{F}\right\rfloor. \tag{8}$$

When $|\hat{x}|$ still exceeds a budget U, overlapping windows are used. An LSTM generator parameterised by θ, denoted $G(\cdot;\theta)$, produces sequences by repeating an affine transform of z across time and passing through a stacked LSTM and a linear+tanh head. The discriminator $D(\cdot;\psi)$ is a stacked LSTM followed by a linear+σ head that consumes the last hidden state. After generation, surrogate outputs are mapped back to the original horizon by block-wise replication

$$\mathrm{Recon}(\hat{y},F)=\Big(\underbrace{\hat{y}\_1,\dots,\hat{y}\_1}\_F,\ \dots,\ \underbrace{\hat{y}\_n,\dots,\hat{y}\_n}\_F\Big)\Big|\_1:T, \tag{9}$$

with cropping/padding to length T.

3.5. Variant: dual strategy for continuous devices

An alternative variant further distinguishes continuous devices as square-wave and spiky. A square wave is detected when a 2-means fit on a downsample of x yields centres $\{m\_1,m\_2\}$ satisfying

$$|m\_1-m\_2|>\gamma\,\mathrm{std}(x), \tag{10}$$

with separation factor γ>0; otherwise the device is called spiky. For square waves, a transposed-convolutional generator replaces the fully connected one and the cycle length is estimated by transition analysis to set the segment length. For continuous devices, salient events are extracted by peak finding at a quantile threshold

$$\theta := \text{Quantile\_}q\big(\{x\_t : x\_t > 0\}\big), \tag{11}$$

with $q \in (0, 1)$, and trained as segments of length S. Full-length sequences are reconstructed by stochastically interleaving generated spikes according to the average inter-peak spacing.

Algorithm 1 Adaptive training pipeline

```
Load UVIC dataset and split by device
for each device do
    Detect type: continuous vs. intermittent
    if intermittent then
        Segment into windows
        Cluster windows into K motifs
        for each cluster do
            Train FC-GAN on segment vectors
        end for
    else
        Simplify long sequence
        Train LSTM-GAN on simplified series
    end if
    Save generators, losses, comparisons, and pattern visualizations
end for
```

4. Experiments

4.1. Dataset

The experimental data used in this paper comes from the UVIC Dataset, which collects time series data on appliance level power consumption from real world residential and office environments. We covering a wide range of representative home appliances and electronic devices. The visualization can be found in Figure 3. This data records the active power variations of individual appliances in a time series format, offering high temporal resolution and behavioral diversity, fully reflecting the dynamic characteristics of different types of devices in daily use. The devices in the dataset include both continuous loads such as desktops, iMacs, servers, and refrigerators, whose power curves exhibit smooth and sustained trends, and intermittent loads such as coffee makers, microwaves, printers, and water coolers, which exhibit typical bursts and mode switching.

The raw data underwent a series of preprocessing to ensure quality and consistency. The UVIC dataset covers eleven categories of electrical appliances, demonstrating high representativeness and broad application potential. Its rich behavioral diversity provides a solid experimental foundation for synthetic data generation and non-intrusive load monitoring (NILM) research. Through training and validation on this dataset, the model not only approximates the true power distribution in terms of statistical characteristics, but also maintains a high degree of similarity in structural and dynamic characteristics, providing reliable data support for energy consumption behavior modeling, data augmentation, and privacy protection.

4.2. Experiment Setup

The adaptive training pipeline of CAG is designed to dynamically allocate devices to appropriate generative branches and stabilize adversarial learning across heterogeneous load types. Each appliance is first classified as intermittent or continuous based on a smoothed derivative heuristic (Eq. 2). If the initial portion of a signal contains long zero valued intervals, or occupancy exceeds 0.7 while the variance of the seven point moving derivative remains below 0.1, the appliance is identified as continuous; otherwise, it is treated as intermittent. This rule reflects the physical intuition that continuous devices maintain steady state operation with low dynamic variance, whereas intermittent

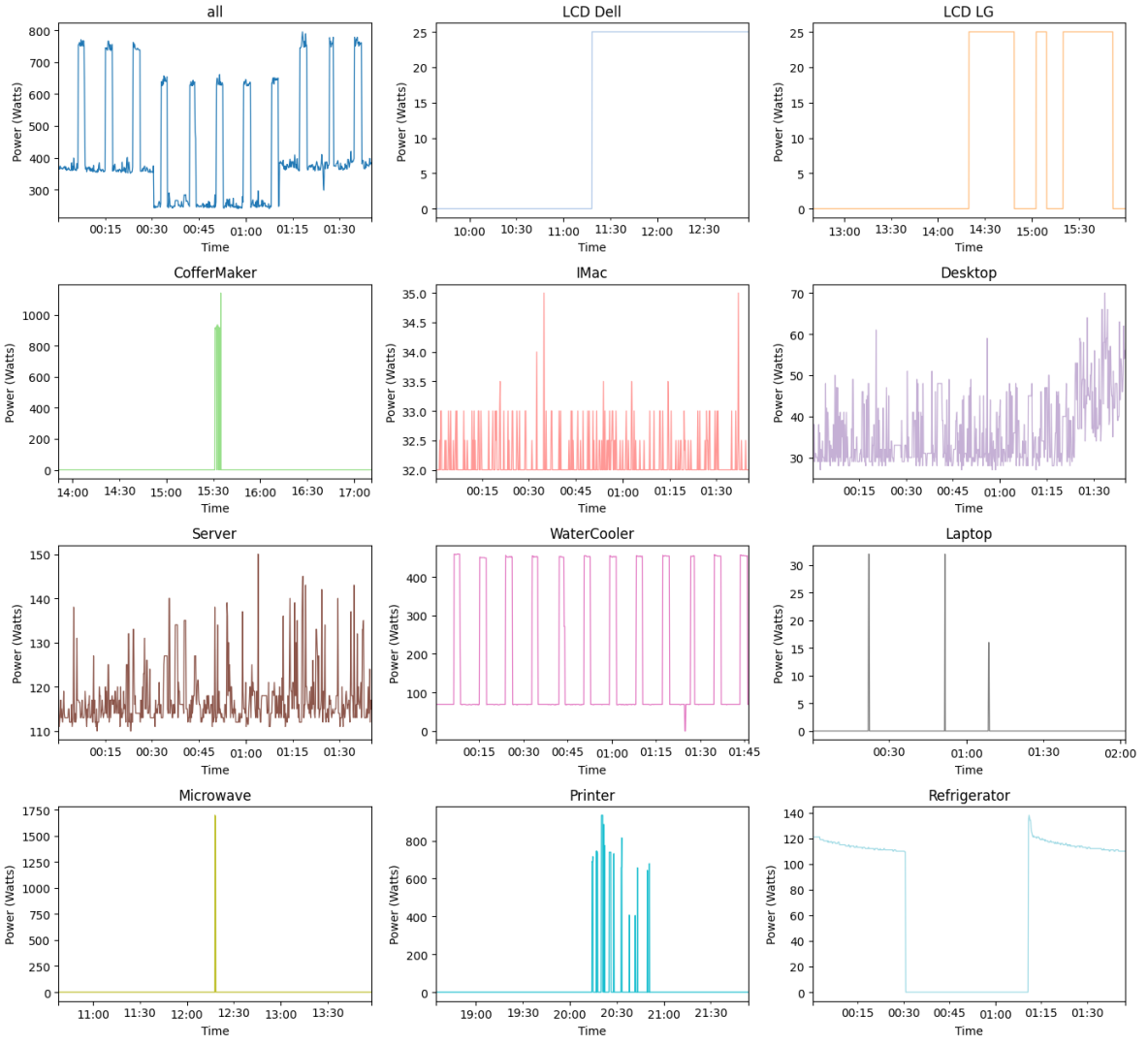

Figure 3: Visualization of appliance load profiles in the dataset. Representative time series power consumption patterns for multiple household and office appliances from the UVIC dataset. Each subplot shows the active power over time for a specific device, illustrating the diversity of load behaviors across appliance types. Intermittent devices such as the CoffeeMaker, Microwave, and Printer exhibit short, bursty activations, whereas continuous devices such as the Desktop, IMac, Server, and Refrigerator display long duration, smoothly varying consumption trajectories. The variability in temporal structure and amplitude highlights the heterogeneity that motivates behavior aware generative modeling in the proposed CAG framework.

ones exhibit abrupt, sparse activations. For intermittent appliances, load traces are divided into fixed length segments, standardized, and clustered according to the feature descriptors defined around Eq. 4. Each cluster represents a distinct usage motif and is modeled by an independent lightweight convolutional GAN sharing common hyperparameters. Both the generator and discriminator adopt compact 1D convolutional architectures with small receptive fields, enabling the capture of local burst patterns while ensuring parameter efficiency and training stability. These convolutional mappings act analogously to stacked linear layers but provide superior translation invariance and robustness against phase shifts frequently observed in short transient loads. For continuous devices, long duration sequences are optionally down sampled into partitioned windows before being processed by a two layer LSTM-based GAN. The generator models long range temporal dependencies and produces simplified trajectory representations that are later reconstructed to

Table 1
Core hyperparameters of the Cluster Aggregation GAN pipeline.

Intermittent branch	
Segment length L	436
Latent dimension d_z	100
Generator architecture	Conv1D(64, kernel=3) →Conv1D(128, kernel=5) →Conv1D(256, kernel=5) → Flatten →Linear(L), ReLU
Discriminator architecture	Conv1D(128, kernel=5) →Conv1D(64, kernel=3) →Flatten →Linear(1), LeakyReLU, Sigmoid
Epochs	1500
Batch size	32
Optimiser	Adam($\eta = 2 \times 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$)
Loss	Binary cross-entropy
Continuous branch	
Simplified horizon	≤ 1000 points (windows of 2000 when longer)
Latent dimension d_z	100
Generator	LSTM
Discriminator	LSTM
Epochs	1500
Batch size	32
Optimiser	Adam($\eta = 2 \times 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$)
Loss	Binary cross-entropy
Reconstruction	Repeat factor from Eq. (8)

the full time horizon through Eq. 9. This simplification reconstruction scheme reduces temporal dimensionality and alleviates gradient explosion or vanishing, leading to smoother convergence and more stable adversarial dynamics.

Comprehensive hyperparameter settings for both branches are summarized in Table 1. Unless otherwise specified, these serve as the default configuration for all experiments. This unified adaptive training design ensures that each device type is learned under conditions best aligned with its temporal characteristics, achieving robust, interpretable, and behavior-consistent synthetic generation.

4.3. Evaluation Metrics

We benchmark all models with a set of domain driven metrics that jointly capture realism, diversity, and periodicity of appliance level load curves. These two viewpoints mirror the goals of practical NILM and data synthesis: (1)Realism : generated traces must align with real power levels. (2)Diversity : cover the variety of user behaviours. Table 2 summarises the average metrics over all devices. Table 3 provides a detailed measurement from all devices. The first column simply lists the model identities, whereas the remaining columns correspond to quantitative indicators grouped by the above dimensions. To measure the metrics for realism, let $\mathcal{X}_r = \{x^{(r)}\}$ and $\mathcal{X}_g = \{x^{(g)}\}$ denote real and generated sequences with length T. We also denote empirical means and standard deviations by (μ_r, σ_r) and (μ_g, σ_g) respectively, and write $\Phi(\cdot)$ for the feature extractor used in the Fréchet distance. Lower values imply closer waveform alignment and higher perceptual realism. The metrics are defined as follows.

- **Mean Error (ME).** We compare first order statistics of real and generated signals

$$\mathrm{ME} = \left|\mu_g - \mu_r\right|.$$

  Smaller values indicate the generated sequences match the average power demand of the original appliance trajectories. This metric is the clearest indicator of whether the generator has recovered the long-term energy budget of a device large deviations imply that downstream NILM models would consistently over- or under-estimate consumption even if the temporal structure looked plausible.

- **Standard Deviation Error (Std).** Second-order consistency is measured through

$$\mathrm{Std} = \left|\sigma_g - \sigma_r\right|.$$

Low deviation implies the synthetic signals reproduce the fluctuation intensity of the real series. Whereas ME focuses on the baseline level, the standard deviation reflects burstiness and variability in practice it determines whether intermittent devices fire with the right amplitude contrast and whether continuous appliances retain natural drift instead of appearing overly smooth.

- **Fidelity RMSE (Fid.).** Local waveform similarity is evaluated by matching each generated trace to its closest real neighbour:

$$\text{Fid.} = \frac{1}{|\mathcal{X}_g|} \sum_{x^{(g)} \in \mathcal{X}_g} \min_{x^{(r)} \in \mathcal{X}_r} \sqrt{\frac{1}{T} \|x^{(g)} - x^{(r)}\|_2^2}.$$

This root-mean-square error captures pointwise fidelity beyond the low-order statistics above. A model with the right mean and variance can still miss transient ramps or spikes; the fidelity term penalises such shape discrepancies and therefore reflects how realistic an individual cycle will look to a NILM classifier that relies on detailed waveform cues.

- **Period MAE (Per).** We estimate the dominant period τ of each sequence via spectral peak detection and measure the mean absolute error

$$\text{Period} = \frac{1}{|\mathcal{X}_g|} \sum_{x^{(g)} \in \mathcal{X}_g} \left| \tau(x^{(g)}) - \tau(x^{(r)}_{\text{match}}) \right|,$$

where $x^{(r)}_{\text{match}}$ is the closest real sequence in the sense of dominant frequency. Lower scores signify better preservation of appliance duty cycles. Period accuracy is critical for thermostatically controlled and compressor-based loads whose duty cycle timing encodes operational behaviour. By keeping this error small we ensure that generated traces respect on/off cadence and rest intervals, which would otherwise break the plausibility of synthetic datasets despite matching instantaneous amplitudes.

- **Feature FID (Fea).** Structural similarity in a learned feature space is measured through a Fréchet distance

$$\text{Fea} = \left\| m_r - m_g \right\|_2^2 + \text{Tr}\left(C_r + C_g - 2\big(C_r^{1/2} C_g C_r^{1/2}\big)^{1/2} \right),$$

with (m_r, C_r) and (m_g, C_g) denoting the empirical mean and covariance of features $\Phi(x^{(r)})$ and $\Phi(x^{(g)})$. Smaller values correspond to higher structural realism. Unlike the previous scalar statistics, the Feature FID evaluates high-level embeddings learned from the entire waveform. It captures correlated variations such as typical ramp shapes or multi-stage activation patterns. A low Feature FID therefore indicates that the generator reproduces latent appliance semantics, not just marginal distributions.

4.3.1. Diversity Metrics

- **Diversity RMSE (Div).** To ensure the generator does not collapse to a single motif, we compute the average pairwise distance of generated samples

$$\text{Div} = \frac{2}{|\mathcal{X}_g|(|\mathcal{X}_g| - 1)} \sum_{i<j} \sqrt{\frac{1}{T} \|x_i^{(g)} - x_j^{(g)}\|_2^2}.$$

Moderately high values are desirable: too small indicates mode collapse, whereas excessively large values suggest unrealistic variability. In practice this statistic tells us whether the generator explores the full repertoire of usage motifs multiple kettle boils, short and long printer jobs, etc. instead of recycling a single template.

- **Cluster Coverage (CC).** Given K behavioural clusters extracted from real data, let $n_k^{(g)}$ be the number of generated samples assigned to cluster k. Coverage is defined as

$$\text{Clus. Cov.} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}\left[n_k^{(g)} > 0\right].$$

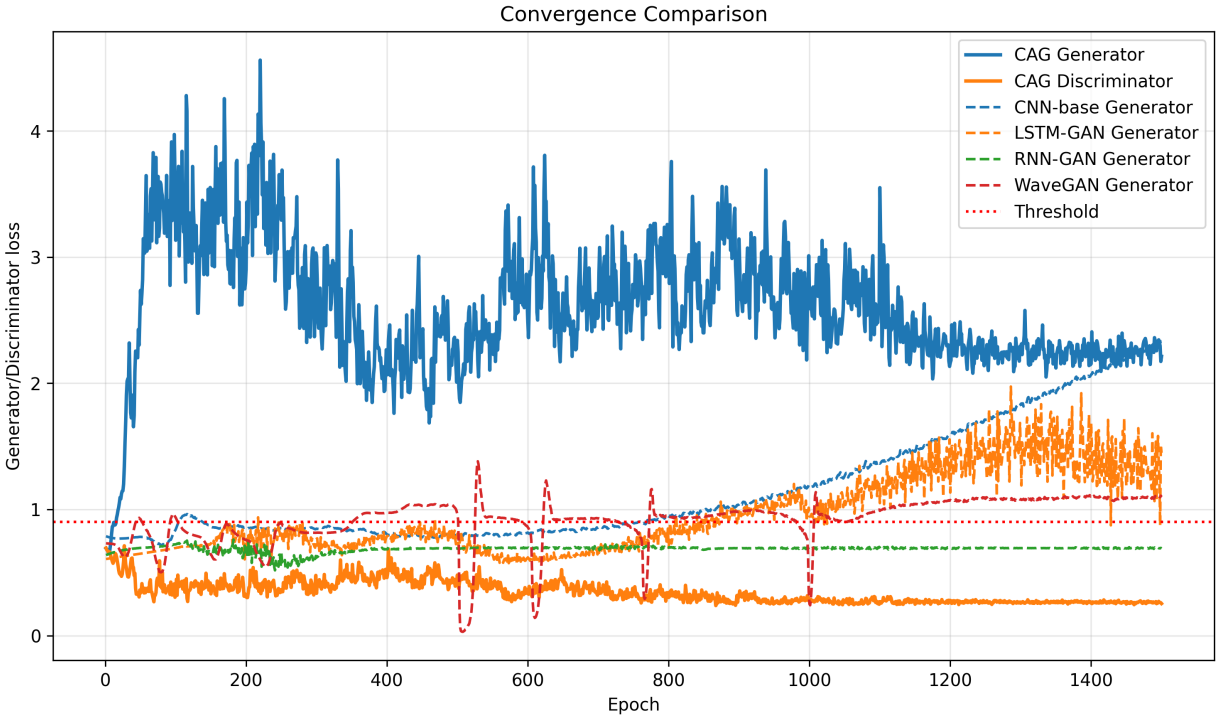

Figure 4: Convergence comparison of different generative models. The proposed CAG framework exhibits the most stable and consistent convergence behavior, with the generator and discriminator losses gradually approaching equilibrium. In contrast, baseline models such as CNN-base, LSTM-GAN, RNN-GAN, and WaveGAN show larger oscillations or divergence trends, indicating training instability. This demonstrates that the adaptive routing and clustering mechanisms in CAG effectively enhance training stability and improve convergence efficiency.

Values closer to 1 mean the generator reproduces all observed behavioural modes. This metric directly measures how many of the real-world motifs survive in the synthetic dataset. A coverage gap implies some user behaviours disappear from training data, which would bias NILM models; maintaining high coverage ensures that even rare patterns (e.g., late-night microwave cycles) are synthesised.

- **Cluster Jensen–Shannon Divergence (CJ).** Let p_r and p_g be the normalised cluster histograms for real and generated data. We compute

$$\mathrm{CJ} = \tfrac{1}{2}\,\mathrm{KL}(p_r \| m) + \tfrac{1}{2}\,\mathrm{KL}(p_g \| m), \qquad m = \tfrac{1}{2}(p_r + p_g),$$

where KL denotes the Kullback–Leibler divergence. Smaller values indicate balanced sampling across behavioural clusters.

While coverage checks whether every cluster appears at least once, the Jensen–Shannon divergence quantifies how closely the frequency of each motif matches reality. Minimising CJ prevents the generator from over-producing easy clusters and under-representing difficult ones, yielding a synthetic corpus whose behavioural distribution mirrors the ground truth. Together these metrics reward models that simultaneously match real power levels, explore the behavioural space, and respect device periodicity, forming a comprehensive basis for the comparative studies in the following sections.

4.4. Convergence Analysis

To evaluate the convergence stability of our proposed framework, we conducted comparative experiments against several baseline generative models under identical training conditions. The convergence behavior clearly demonstrates

Table 2
Average realism and diversity metrics across all evaluated devices.

Model	Realism					Diversity		
	ME ↓	Std ↓	Fid ↓	Per ↓	Feature FID ↓	Div ↑	CC ↑	CJ ↑
CAG	$8.03e+00$▲	$1.35e+01$▲	$4.26e+01$▲	$2.31e+01$▲	$5.82e+16$▲	$1.34e+02$▲	$3.03e-01$▲	$6.57e-01$▲
CNN-Base	$1.20e+02$▼	$6.07e+01$▼	$1.39e+02$▼	$2.28e+02$▼	$9.97e+16$▼	$2.25e+01$▼	$8.58e-02$▼	$3.72e-01$▼
LSTM-GAN	$1.73e+02$▼	$1.91e+01$▼	$1.70e+02$▼	$1.91e+02$▼	$6.32e+16$▼	$6.14e+01$▼	$1.62e-01$▼	$3.70e-01$▼
RNN-GAN	$1.61e+02$▼	$7.92e+01$▼	$1.80e+02$▼	$2.37e+02$▼	$6.36e+16$▼	$1.22e+02$▼	$2.17e-01$▼	$3.66e-01$▼
WaveGAN	$1.19e+02$▼	$2.97e+01$▼	$1.24e+02$▼	$2.13e+02$▼	$6.40e+16$▼	$4.52e+01$▼	$1.41e-01$▼	$3.46e-01$▼

that CAG achieves the fastest, smoothest, and most reliable convergence among all tested models. The visualization can be found in Figure 4. This superior performance can be attributed to its cluster based learning strategy and adaptive routing mechanism, which effectively balance model capacity and behavioral complexity across heterogeneous appliance types. The monolithic GAN architectures that attempt to fit all appliance behaviors within a single generator discriminator pair, CAG decomposes the learning problem into behavior specific subspaces. The framework first identifies device type through lightweight statistical indicators such as sparsity and temporal variance and then routes each device either to an intermittent branch. For intermittent devices, the time series data are segmented into fixed length windows and transformed into shape based feature vectors. These features are clustered and each cluster representing a characteristic of appliance operation is modeled by an independent, lightweight GAN. This cluster wise training strategy substantially reduces the optimization difficulty by transforming a multi modal learning problem into multiple subproblems, allowing each generator to focus on a compact and coherent feature distribution. As a result, the generator discriminator dynamics remain stable, gradients are conditioned, and the system converges more rapidly toward equilibrium.

For continuous devices, the CAG framework employs a simplification reconstruction pipeline. Long consumption sequences are first simplified through uniform temporal averaging, which compresses redundant information while preserving the essential trend and periodicity. This reduced representation mitigates the exploding vanishing gradient problem typically observed in long sequence adversarial training. The simplified sequences are then modeled using an LSTM-based GAN capable of capturing long range dependencies efficiently, followed by a reconstruction step that restores the original time resolution. By lowering the temporal dimensionality of the adversarial learning process, the training landscape becomes smoother, enabling faster convergence with less oscillation in both generator and discriminator losses.

Empirically, the convergence curves confirm these theoretical advantages. CAG generator loss decreases rapidly during early epochs and stabilizes at a lower equilibrium value, while its discriminator loss remains consistently bounded within an optimal range signifying balanced adversarial dynamics. In contrast, baseline models such as CNN-base, LSTM-GAN, RNN-GAN, and WaveGAN exhibit pronounced oscillations, delayed convergence, or even divergence, indicating training instability and partial mode collapse. Quantitatively, CAG achieves the lowest mean error, standard deviation error, reconstruction RMSE, and Feature FID across all evaluated devices, demonstrating not only convergence speed but also convergence quality.

The superior convergence of CAG arises from its cluster aggregation strategy, which reduces computational complexity, enhances feature specialization, and aligns model structure with the intrinsic heterogeneity of appliance behaviors. By decomposing complex temporal distributions into interpretable motif clusters and assigning tailored GANs to each, CAG attains a well regularized optimization process that converges efficiently, stably, and with higher fidelity to real world energy consumption patterns.

4.5. Comparative Experiment

To compare our proposed CAG framework against existing models, we implemented several baseline architectures including a CNN-based GAN, an LSTM-based GAN, a RNN-based GAN, and a WaveGAN adapted for time series data. Each model was trained under identical conditions using the UVIC dataset, with hyperparameters tuned for optimal performance. Table 2 reports average metrics across all devices, and Table 3 provides device detail. The example of CAG output visualization can be see in Figure 5.

Table 2 shows that CAG attains the best aggregate realism scores of mean error 8.03, standard deviation error 13.46, fidelity RMSE 42.6, period MAE 23.1, and Feature FID 5.82×10^{16} while simultaneously delivering the largest diversity

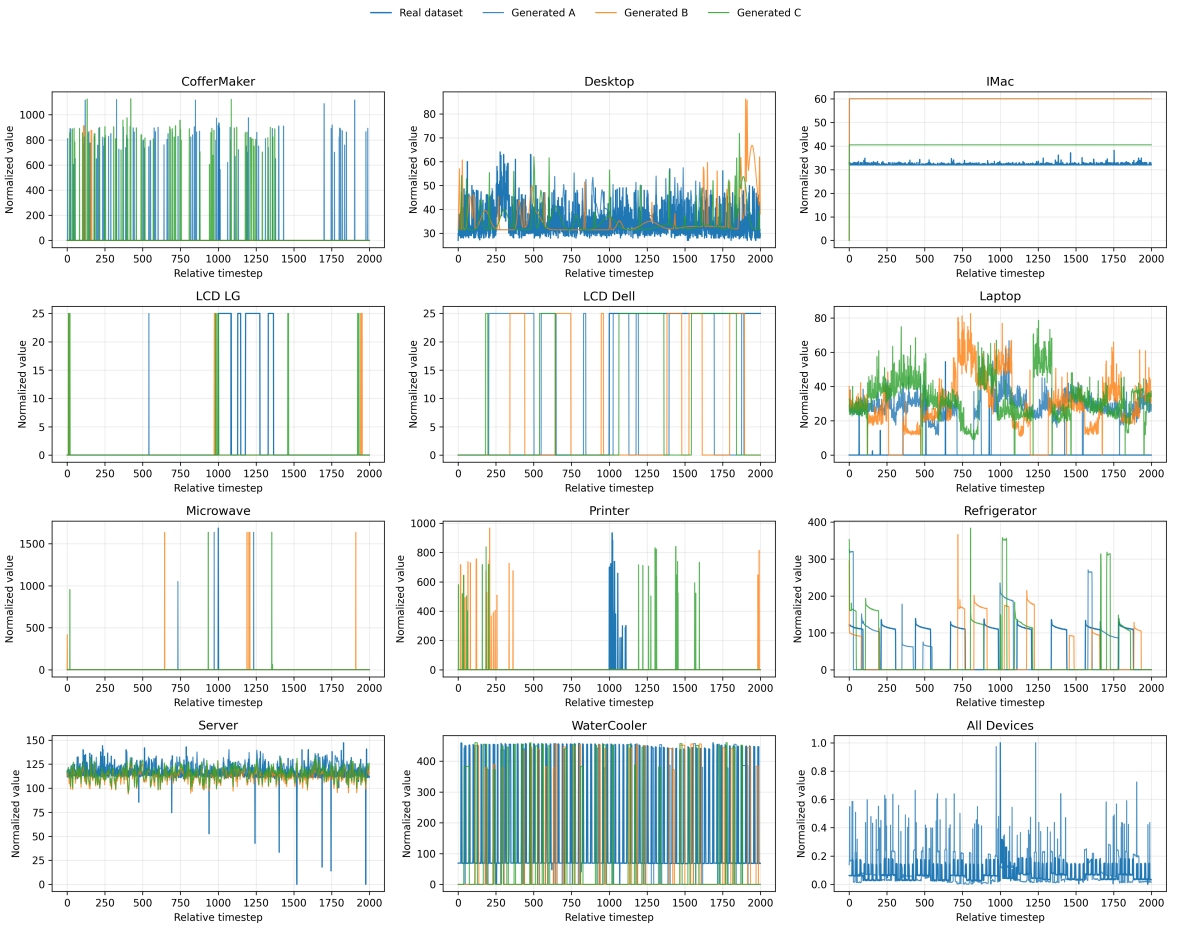

Figure 5: Example output visualization of the proposed CAG framework. The generated appliance-level load patterns closely resemble the real data, demonstrating the effectiveness of the cluster-based learning strategy and adaptive routing mechanism.

metrics diversity RMSE 1.34×10^{2}, cluster coverage 3.03×10^{-1}, and cluster JS 6.57×10^{-1}. The closest baseline WaveGAN records a mean error of 1.19×10^{2}, and none of the alternatives approach CAG on coverage or balance, underscoring the benefit of treating clustering as part of the generator design rather than merely as a preprocessing aid.

The superior performance of CAG can be directly linked to its cluster aggregation strategy. By partitioning intermittent appliance signals into interpretable usage motifs and assigning an independent lightweight GAN to each cluster, the framework avoids the instability that often arises in monolithic adversarial models forced to learn highly multimodal distributions. This localized learning substantially reduces optimization variance and prevents mode collapse, leading to smoother convergence and lower generative bias. For continuous devices, the combination of sequence simplification and LSTM-based temporal modeling further enhances stability by lowering the effective dimensionality of adversarial training and enabling the model to capture long-range dependencies without gradient explosion or vanishing. Consequently, the quantitative outcomes in Table 2 not only demonstrate higher fidelity but also empirically confirm the theoretical convergence analysis described earlier.

The device level statistics in Table 3 reinforce this trend. CAG captures the majority of realism metrics for continuous appliances such as Desktop, IMac, Server, and Refrigerator while retaining the top diversity indicators for every appliance. Isolated baseline wins do appear CNN-Base yields the lowest Desktop fidelity RMSE 7.67 versus 8.68 and LSTM-GAN tightens the CoffeeMaker standard deviation error 9.46 versus 25.2 but each comes with pronounced penalties elsewhere. Desktop mean error rises to 1.31×10^{1} versus 9.61×10^{-1} and CoffeeMaker mean error explodes

to 4.97×10^{2} versus 3.48×10^{1}. Consequently, the cluster aggregated routing supplies the best overall trade-off between realism and motif coverage.

The analysis of Table 2 and Table 3 corroborates the central claim of CAG framework achieves the best overall convergence and generalization performance, simultaneously ensuring statistical fidelity, behavioral diversity, and temporal stability. By integrating adaptive routing with cluster aware adversarial training, CAG aligns its model capacity with the intrinsic heterogeneity of appliance behaviors, thereby establishing a robust and interpretable generative framework for synthetic load pattern generation.

4.6. Ablation Study

To gauge the effect of the clustering stage alone, we retrained the CAG generator after disabling cluster aggregation for four representative appliances with two intermittent and two continuous. For each device we keep the rest of the pipeline unchanged and reuse the same hyperparameters and sampling budget as the clustered model. Table 4 contrasts the clustered as default and non clustered variants across the metrics introduced earlier.

Across all four devices the ablation confirms that clustering is the lever that keeps both realism and diversity under control. For CoffeeMaker, turning clustering off sends the mean error from 3.48×10^{1} to 5.57×10^{2} and the fidelity RMSE from 1.32×10^{2} to 5.52×10^{2} while cluster coverage collapses from 4.94×10^{-1} to 1.67×10^{-1}. Desktop shows the same pattern: period MAE almost doubles from 8.45 to 16.1, Feature FID rises from 8.61×10^{11} to 9.16×10^{11}, and diversity RMSE shrinks from 1.35×10^{1} to 1.66. The long-horizon Microwave gains are even more dramatic—fidelity RMSE is reduced by more than an order of magnitude from 8.37×10^{2} to 6.45×10^{1} and cluster coverage leaps from 5.56×10^{-2} to 3.28×10^{-1}. Water Cooler follows suit with mean error tightening from 1.11×10^{2} to 1.22×10^{1}, period MAE falling from 5.57×10^{1} to 1.79×10^{1}, and cluster JS improving from 3.10×10^{-1} to 6.45×10^{-1}. Together these results show that the clustering branch is not merely cosmetic: it prevents the generator from collapsing onto a handful of modes and is responsible for the large reductions in reconstruction error and the multi-fold gains in coverage reported in Table 4.

Table 3
Realism and diversity metrics for each device and model.

Device	Model	Realism					Diversity		
		ME ↓	Std ↓	Fid ↓	Per ↓	Feature FID ↓	Div ↑	CC ↑	CJ ↑
Coffee Maker	CAG	$3.48e+01$▲	$2.52e+01$▼	$1.32e+02$▲	$4.10e+00$▲	$2.61e+16$▼	$3.22e+02$▲	$4.94e-01$▲	$6.80e-01$▲
	CNN-Base	$3.81e+02$▼	$3.41e+02$▼	$5.73e+02$▼	$5.37e+00$▼	$4.37e+17$▼	$1.12e+02$▼	$1.11e-01$▼	$3.90e-01$▼
	LSTM-GAN	$4.97e+02$▼	$9.46e+00$▲	$4.93e+02$▼	$3.20e+02$▼	$2.90e+16$▼	$1.24e+02$▼	$1.67e-01$▼	$3.70e-01$▼
	RNN-GAN	$3.58e+02$▼	$1.69e+02$▼	$3.75e+02$▼	$3.00e+02$▼	$2.90e+16$▼	$3.06e+02$▼	$4.44e-01$▼	$3.60e-01$▼
	WaveGAN	$3.59e+02$▼	$1.09e+01$▼	$3.59e+02$▼	$9.24e+01$▼	$1.79e+16$▲	$1.30e+02$▼	$1.67e-01$▼	$3.20e-01$▼
Desktop	CAG	$9.61e-01$▲	$1.03e+00$▲	$8.68e+00$▼	$8.45e+00$▲	$8.61e+11$▲	$1.35e+01$▲	$1.61e-01$▲	$6.50e-01$▲
	CNN-Base	$2.00e+00$▼	$2.37e+00$▼	$7.67e+00$▲	$1.25e+01$▼	$9.07e+11$▼	$1.66e+00$▼	$1.11e-01$▼	$5.00e-01$▼
	LSTM-GAN	$1.31e+01$▼	$4.40e+00$▼	$9.73e+00$▼	$3.16e+02$▼	$1.00e+12$▼	$4.36e+00$▼	$1.11e-01$▼	$4.80e-01$▼
	RNN-GAN	$1.07e+01$▼	$2.20e+00$▼	$9.37e+00$▼	$2.73e+02$▼	$1.00e+12$▼	$1.19e+01$▼	$5.56e-02$▼	$4.30e-01$▼
	WaveGAN	$1.47e+01$▼	$7.29e+00$▼	$1.00e+01$▼	$3.83e+01$▼	$9.97e+11$▼	$1.28e+00$▼	$5.56e-02$▼	$4.20e-01$▼
IMac	CAG	$4.20e-01$▲	$5.93e-01$▲	$1.79e+00$▲	$4.15e+01$▲	$6.69e+11$▼	$1.77e+02$▲	$3.83e-01$▲	$6.40e-01$▲
	CNN-Base	$5.59e-01$▼	$2.01e+00$▼	$4.36e+00$▼	$9.27e+01$▼	$6.83e+11$▼	$2.82e-01$▼	$5.56e-02$▼	$4.60e-01$▼
	LSTM-GAN	$9.11e-01$▼	$1.06e+00$▼	$2.57e+00$▼	$3.47e+02$▼	$7.20e+11$▼	$3.90e+00$▼	$3.33e-01$▼	$4.40e-01$▼
	RNN-GAN	$2.46e+02$▼	$2.28e+02$▼	$3.35e+02$▼	$4.27e+02$▼	$5.12e+11$▲	$1.68e+02$▼	$5.56e-02$▼	$4.50e-01$▼
	WaveGAN	$2.31e+01$▼	$7.35e-01$▼	$2.26e+01$▼	$1.17e+02$▼	$6.48e+11$▼	$3.77e+00$▼	$2.22e-01$▼	$4.30e-01$▼
LCD-LG	CAG	$4.34e-02$▲	$9.77e-01$▼	$1.73e+00$▲	$4.78e+01$▲	$1.55e+12$▼	$6.34e+00$▲	$1.06e-01$▲	$6.30e-01$▲
	CNN-Base	$2.29e+00$▼	$1.49e+00$▼	$2.84e+00$▼	$4.33e+02$▼	$1.55e+12$▼	$3.21e-01$▼	$5.56e-02$▼	$4.20e-01$▼
	LSTM-GAN	$9.03e+00$▼	$3.94e-01$▼	$9.26e+00$▼	$9.71e+01$▼	$1.55e+12$▼	$2.61e+00$▼	$5.56e-02$▼	$4.00e-01$▼
	RNN-GAN	$8.60e+00$▼	$1.76e+00$▼	$8.44e+00$▼	$1.67e+02$▼	$1.55e+12$▼	$5.09e+00$▼	$5.56e-02$▼	$4.10e-01$▼
	WaveGAN	$7.66e+00$▼	$2.08e-01$▲	$7.66e+00$▼	$2.95e+02$▼	$1.48e+12$▲	$2.81e+00$▼	$5.56e-02$▼	$3.90e-01$▼
LCD-Dell	CAG	$7.04e+00$▲	$1.07e+01$▼	$9.49e-01$▲	$5.00e-03$▼	$3.53e+12$▼	$9.96e+01$▲	$2.17e-01$▲	$6.35e-01$▲
	CNN-Base	$9.94e+00$▼	$9.46e+00$▼	$3.09e+00$▼	$4.21e+02$▼	$3.51e+12$▼	$9.40e-01$▼	$5.56e-02$▼	$4.10e-01$▼
	LSTM-GAN	$9.25e+00$▼	$3.98e+00$▲	$6.38e+00$▼	$7.89e+00$▼	$3.49e+12$▼	$8.60e+00$▼	$1.67e-01$▼	$4.00e-01$▼
	RNN-GAN	$6.01e+01$▼	$9.11e+01$▼	$1.04e+02$▼	$0.00e+00$▲	$3.03e+12$▲	$9.39e+01$▼	$1.67e-01$▼	$3.90e-01$▼
	WaveGAN	$3.33e+01$▼	$6.92e+00$▼	$2.60e+01$▼	$3.06e+02$▼	$3.17e+12$▼	$5.28e+00$▼	$1.11e-01$▼	$3.80e-01$▼
Laptop	CAG	$1.03e+00$▲	$1.10e+01$▼	$1.04e+01$▼	$5.26e+01$▲	$2.83e+14$▼	$2.28e+01$▲	$5.50e-01$▲	$6.60e-01$▲
	CNN-Base	$2.67e+01$▼	$1.60e+01$▼	$1.46e+01$▼	$3.88e+02$▼	$2.86e+14$▼	$4.27e+00$▼	$5.56e-02$▼	$3.30e-01$▼
	LSTM-GAN	$1.77e+01$▼	$1.37e+01$▼	$6.57e+00$▲	$7.82e+01$▼	$2.88e+14$▼	$8.28e+00$▼	$3.33e-01$▼	$3.40e-01$▼
	RNN-GAN	$1.77e+01$▼	$2.86e+00$▲	$6.67e+00$▼	$2.69e+02$▼	$2.88e+14$▼	$2.08e+01$▼	$5.00e-01$▼	$3.50e-01$▼
	WaveGAN	$3.41e+01$▼	$1.36e+01$▼	$2.21e+01$▼	$3.52e+02$▼	$2.77e+14$▲	$8.57e+00$▼	$2.78e-01$▼	$3.20e-01$▼
Microwave	CAG	$1.18e+01$▲	$7.94e+00$▲	$6.45e+01$▲	$3.87e+01$▲	$5.43e+16$▲	$3.23e+02$▲	$3.28e-01$▲	$6.90e-01$▲
	CNN-Base	$3.80e+02$▼	$9.73e+01$▼	$4.12e+02$▼	$4.10e+02$▼	$5.65e+16$▼	$3.90e+01$▼	$5.56e-02$▼	$3.10e-01$▼
	LSTM-GAN	$6.27e+02$▼	$9.21e+01$▼	$6.26e+02$▼	$6.90e+01$▼	$5.51e+16$▼	$1.86e+02$▼	$1.11e-01$▼	$3.20e-01$▼
	RNN-GAN	$6.34e+02$▼	$1.99e+02$▼	$6.30e+02$▼	$2.40e+02$▼	$5.52e+16$▼	$3.07e+02$▼	$2.78e-01$▼	$3.30e-01$▼
	WaveGAN	$6.61e+02$▼	$1.07e+02$▼	$6.74e+02$▼	$3.30e+02$▼	$7.58e+16$▼	$2.08e+02$▼	$1.11e-01$▼	$3.00e-01$▼
Printer	CAG	$1.67e+01$▼	$1.05e+01$▲	$6.41e+01$▼	$3.34e+01$▲	$1.74e+15$▲	$1.99e+02$▲	$3.83e-01$▲	$6.70e-01$▲
	CNN-Base	$2.48e+02$▼	$5.73e+01$▼	$2.66e+02$▼	$4.27e+02$▼	$2.36e+15$▼	$4.92e+01$▼	$5.56e-02$▼	$3.20e-01$▼
	LSTM-GAN	$4.61e+02$▼	$4.20e+01$▼	$4.55e+02$▼	$9.62e+01$▼	$2.71e+15$▼	$1.05e+02$▼	$1.11e-01$▼	$3.30e-01$▼
	RNN-GAN	$2.66e+02$▼	$1.14e+02$▼	$2.73e+02$▼	$1.62e+02$▼	$2.71e+15$▼	$1.89e+02$▼	$3.33e-01$▼	$3.10e-01$▼
	WaveGAN	$9.38e+00$▲	$1.12e+01$▼	$6.05e+01$▲	$3.22e+02$▼	$1.84e+15$▼	$7.38e+01$▼	$2.78e-01$▼	$2.90e-01$▼
Refrigerator	CAG	$2.48e+00$▲	$1.01e+01$▼	$5.72e+01$▲	$0.00e+00$▲	$4.23e+16$▲	$8.07e+01$▲	$2.17e-01$▲	$6.55e-01$▲
	CNN-Base	$6.37e+01$▼	$4.34e+01$▼	$7.01e+01$▼	$2.22e+02$▼	$6.84e+16$▼	$1.08e+01$▼	$1.11e-01$▼	$3.00e-01$▼
	LSTM-GAN	$1.24e+02$▼	$3.00e+01$▼	$8.67e+01$▼	$7.55e+01$▼	$6.91e+16$▼	$3.81e+01$▼	$1.67e-01$▼	$3.20e-01$▼
	RNN-GAN	$1.09e+02$▼	$3.49e+00$▲	$8.50e+01$▼	$1.76e+02$▼	$6.91e+16$▼	$7.59e+01$▼	$1.67e-01$▼	$3.10e-01$▼
	WaveGAN	$9.38e+01$▼	$3.35e+01$▼	$9.32e+01$▼	$3.43e+02$▼	$6.63e+16$▼	$3.59e+01$▼	$1.11e-01$▼	$3.30e-01$▼
Server	CAG	$8.75e-01$▲	$3.27e+00$▼	$1.21e+01$▲	$9.15e+00$▲	$1.88e+11$▲	$3.81e+01$▲	$3.28e-01$▲	$6.75e-01$▲
	CNN-Base	$1.05e+00$▼	$8.02e+00$▼	$1.68e+01$▼	$3.32e+01$▼	$8.12e+11$▼	$4.51e+00$▼	$2.22e-01$▼	$3.40e-01$▼
	LSTM-GAN	$3.04e+01$▼	$5.71e+00$▼	$2.91e+01$▼	$3.37e+02$▼	$2.78e+11$▼	$1.64e+01$▼	$1.11e-01$▼	$3.50e-01$▼
	RNN-GAN	$1.04e+01$▼	$2.25e+01$▼	$2.19e+01$▼	$3.38e+02$▼	$2.78e+11$▼	$3.53e+01$▼	$2.78e-01$▼	$3.60e-01$▼
	WaveGAN	$1.92e+01$▼	$4.12e-01$▲	$1.35e+01$▼	$8.08e+01$▼	$2.57e+11$▼	$1.13e+01$▼	$1.11e-01$▼	$3.30e-01$▼
Water Cooler	CAG	$1.22e+01$▲	$6.67e+01$▼	$1.15e+02$▼	$1.79e+01$▲	$5.16e+17$▲	$1.88e+02$▲	$1.61e-01$▲	$6.45e-01$▲
	CNN-Base	$2.04e+02$▼	$8.95e+01$▼	$1.62e+02$▼	$6.00e+01$▼	$5.32e+17$▼	$2.49e+01$▼	$5.56e-02$▼	$3.10e-01$▼
	LSTM-GAN	$1.18e+02$▼	$7.63e+00$▲	$1.49e+02$▼	$3.57e+02$▼	$5.39e+17$▼	$1.78e+02$▼	$1.11e-01$▼	$3.20e-01$▼
	RNN-GAN	$4.82e+01$▼	$3.68e+01$▼	$1.34e+02$▼	$2.53e+02$▼	$5.43e+17$▼	$1.29e+02$▼	$5.56e-02$▼	$3.30e-01$▼
	WaveGAN	$4.95e+01$▼	$1.35e+02$▼	$7.68e+01$▲	$7.18e+01$▼	$5.42e+17$▼	$1.70e+01$▼	$5.56e-02$▼	$3.00e-01$▼

Table 4
Ablation results for removing cluster aggregation on four devices. Each metric follows the same Realism/Diversity grouping used in Tables 2 and 3.

Device	Variant	Realism					Diversity		
		ME ↓	Std ↓	Fid ↓	Per ↓	Feature FID ↓	Div ↑	CC ↑	CJ ↑
Coffee Maker	With Clusters	$3.48e+01$▲	$2.52e+01$▲	$1.32e+02$▲	$4.10e+00$▲	$2.61e+16$▲	$3.22e+02$▲	$4.94e-01$▲	$6.80e-01$▲
	No Clusters	$5.57e+02$▼	$3.95e+01$▼	$5.52e+02$▼	$1.58e+01$▼	$2.75e+16$▼	$1.24e+02$▼	$1.67e-01$▼	$3.90e-01$▼
Desktop	With Clusters	$9.61e-01$▲	$1.03e+00$▲	$8.68e+00$▲	$8.45e+00$▲	$8.61e+11$▲	$1.35e+01$▲	$1.61e-01$▲	$6.50e-01$▲
	No Clusters	$1.44e+01$▼	$1.19e+01$▼	$2.23e+01$▼	$1.61e+01$▼	$9.16e+11$▼	$1.66e+00$▼	$5.56e-02$▼	$5.00e-01$▼
Microwave	With Clusters	$1.18e+01$▲	$7.94e+00$▲	$6.45e+01$▲	$3.87e+01$▲	$5.43e+16$▲	$3.23e+02$▲	$3.28e-01$▲	$6.90e-01$▲
	No Clusters	$8.47e+02$▼	$8.37e+00$▼	$8.37e+02$▼	$4.26e+02$▼	$6.00e+16$▼	$3.90e+01$▼	$5.56e-02$▼	$3.10e-01$▼
Water Cooler	With Clusters	$1.22e+01$▲	$6.67e+01$▲	$1.15e+02$▲	$1.79e+01$▲	$5.16e+17$▲	$1.88e+02$▲	$1.61e-01$▲	$6.45e-01$▲
	No Clusters	$1.11e+02$▼	$1.08e+02$▼	$1.69e+02$▼	$5.57e+01$▼	$5.37e+17$▼	$1.29e+02$▼	$5.56e-02$▼	$3.10e-01$▼

5. Conclusion

In this paper, we presented Cluster Aggregated GAN (CAG), a unified framework for appliance level load pattern generation that explicitly aligns generative modeling with the behavioral heterogeneity of electrical devices. By incorporating segment level clustering and adaptive routing, CAG transforms the conventional monolithic GAN formulation into a behavior aware and interpretable synthesis process. The proposed model captures motif variations in intermittent loads through cluster specific convolutional GANs, while maintaining longterm temporal consistency in continuous appliances via an LSTM-based generator. Comprehensive experiments on the UVIC dataset demonstrate that CAG consistently outperforms existing GAN-based baselines including CNN-base, LSTM-GAN, RNN-GAN, and WaveGAN in both realism and diversity metrics. Specifically, CAG achieves superior motif fidelity, enhanced temporal realism, and balanced behavioral coverage, all while exhibiting improved convergence stability. These results confirm that incorporating clustering as an integral part of the generative process not merely as a preprocessing step offers substantial advantages for complex, multimodal energy time series synthesis. Future extensions will focus on integrating contextual variables such as weather, occupancy, and household routines to further enhance generation fidelity and realism. Overall, CAG establishes a novel paradigm for cluster aware adversarial learning, bridging the gap between statistical diversity and behavioral interpretability in appliance load modeling.

References

[1] Bashar, M.A., Nayak, R., 2023. Algan: Time series anomaly detection with adjusted-lstm gan. arXiv preprint arXiv:2308.06663 .

[2] Buneeva, N., Reinhardt, A., 2017. Ambal: Appliance modeling by aggregation of load segments, in: Proceedings of the 8th International Conference on Future Energy Systems (e-Energy), pp. 287–292.

[3] Castangia, M., et al., 2025. Residential load modeling with generative adversarial networks. IEEE Transactions on Sustainable Computing Early Access.

[4] El Kababji, S., Srikantha, P., 2020. A data-driven approach for generating synthetic load patterns and usage habits. IEEE Transactions on Smart Grid 11, 4984–4995.

[5] Esteban, C., Hyland, S.L., Rätsch, G., 2017. Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633 .

[6] Gkoutroumpi, C., et al., 2024. Sgan: Appliance signatures data generation for nilm applications using gans, in: Proceedings of the 12th Computing Conference, Springer.

[7] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2020. Generative adversarial networks. Communications of the ACM 63, 139–144.

[8] Guo, Z., Adedigba, A.P., Mallipeddi, R., 2025a. Cluster-aggregated transformer: Enhancing lightweight parameter models. Engineering Applications of Artificial Intelligence 159, 111468.

[9] Guo, Z., Adedigba, A.P., Mallipeddi, R., Lee, H., 2025b. Dynamic tanh reinforcement learning: A normalization-free transformer for open traveling salesman problem optimization. Journal of the Institute of Control, Robotics and Systems Conference Proceedings , 845–846.

[10] Guo, Z., Kavuri, S., Lee, J., Lee, M., 2023. Ids-extract: Downsizing deep learning model for question and answering, in: 2023 International Conference on Electronics, Information, and Communication (ICEIC), IEEE. pp. 1–5.

[11] Harell, A., Jones, R., Makonin, S., Bajić, I.V., 2021. Tracegan: Synthesizing appliance power signatures using generative adversarial networks. IEEE Transactions on Smart Grid 12, 4553–4563.

[12] Harrell, A., et al., 2021. Tracegan: Synthesizing appliance power signatures using generative adversarial networks. IEEE Transactions on Smart Grid 12, 4553–4563.

[13] Huang, F., Deng, Y., 2023. Tcgan: Convolutional generative adversarial network for time series classification and clustering. Neural Networks 165, 868–883.

[14] Kamyshev, I., et al., 2025. Hifakes: Synthetic high-frequency nilm data for model diagnostics and generalization testing. arXiv preprint arXiv:2409.00062 .

[15] Liang, X., Wang, H., 2025a. Learning and generating diverse residential load patterns using gan with weakly-supervised training and weight selection. arXiv preprint arXiv:2504.14300 .

[16] Liang, X., Wang, H., 2025b. Learning and generating diverse residential load patterns using gan with weakly-supervised training and weight selection. IEEE Transactions on Consumer Electronics .

[17] Yang, M., Wang, Z., Chi, Z., Feng, W., 2022. Wavegan: Frequency-aware gan for high-fidelity few-shot image generation, in: European conference on computer vision, Springer. pp. 1–17.