ENERGYCA-LLM: CONTEXT-GUIDED WINDOW-ADAPTIVE RESIDUAL LEARNING FOR UNIFIED RENEWABLE GENERATION AND LOAD FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Growing renewable penetration is broadening power-system forecasting from isolated load prediction to unified forecasting of electricity demand, wind power, photovoltaic generation, and net load. These tasks share temporal structure but differ in physical drivers, operating regimes, and forecast-origin information availability. Existing methods study them separately or use static, task-agnostic context, limiting dynamic specialization. To our knowledge, EnergyCA-LLM is the first framework for LLM-controlled residual adaptation in unified energy forecasting. Per window, an availability-aware schema organizes the forecast-origin information set, covering task semantics, asset and geographic attributes, calendar and weather information, recent operating states, and field availability. A frozen pretrained LLM encodes this schema into a shared semantic state that interacts with the numerical latent state to produce residual codes. These codes gate multilayer low-rank residual adapters for task- and regime-specific modulation of shared temporal representations. Across four public benchmarks, EnergyCA-LLM achieves the lowest reported error in 15 of the 16 task–metric comparisons under chronological evaluation with four complementary point-forecasting metrics, with PV RMSE close to the strongest reference. Unified-training, encoder-replacement, and adapter-control ablations assess the roles of parameter sharing, pretrained semantic encoding, and dynamic residual adaptation.

1 INTRODUCTION

Short-term energy forecasting underpins unit commitment, reserve allocation, market operation, and the integration of renewable generation. As renewable penetration grows, this forecasting problem extends beyond electricity demand to wind power, photovoltaic (PV) generation, and net load. These targets share recurring temporal structure, yet their dynamics arise from different physical processes: demand responds strongly to calendar and human-activity patterns, wind power to atmospheric conditions and turbine characteristics, PV generation to irradiance and plant configuration, and net load to the coupled variation of demand and renewable output Eren & Küçükdemiral (2024); Wang et al. (2021); Mayer (2022); Browell & Fasiolo (2021). Consequently, a similar numerical trajectory can imply different futures across tasks, assets, weather conditions, and operating regimes. Here, unified forecasting means joint training across four heterogeneous datasets, with one shared checkpoint per seed serving all four task families. This requires temporal knowledge sharing while preserving the context that gives each trajectory its operational meaning.

Task-specific energy models, general numerical forecasters, and LLM-enhanced time-series methods provide three complementary foundations for this setting. However, unified energy forecasting introduces two unresolved requirements. First, **forecast-origin availability** varies across tasks and windows: a field that is known for one forecast may be missing, delayed, or observed only after another forecast is issued. Second, **conditional specialization** is required because the same numerical trajectory can have different meanings under different task identities, assets, weather states, and operating regimes. Across these research lines, context is often supplied as ordinary covariates or task-agnostic auxiliary representations. Such conditioning makes it difficult to explicitly control how shared intermediate representations adapt to task identity and current operating state, and it can blur information boundaries when fields differ in forecast-time availability.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

This motivates a more specific question: can a frozen pretrained language model organize forecast-available energy context into a semantic control state for window-specific adaptation of a shared numerical forecaster? We answer this question with **EnergyCA-LLM**. An availability-aware schema organizes each window's task, asset, geographic, calendar, weather, and operating information. A frozen pretrained LLM encodes this schema into a semantic state, which interacts with the current numerical latent state to control gated low-rank residual adapters across the shared backbone. This division of roles connects forecast-time context to task- and regime-specific adaptation while keeping trajectory prediction in the numerical model.

We evaluate EnergyCA-LLM on four public benchmarks against twelve task-level baselines, and use controlled ablations to examine parameter sharing, pretrained semantic encoding, and dynamic adapter control. Our contributions are as follows:

- We formulate unified energy forecasting as multi-task conditional learning across load, wind power, PV generation, and net load, with an explicit forecast-origin information set.
- We introduce, to our knowledge, the first framework for LLM-controlled residual adaptation in unified energy forecasting. A semantic–numerical controller combines forecast-available context with numerical states to gate shared low-rank adapters.
- We evaluate one jointly trained EnergyCA-LLM checkpoint per seed across all four benchmarks against twelve task-level baselines, and conduct controlled unified-training, encoder-replacement, and adapter-control ablations under the same shared-model protocol.

2 Related Work

2.1 Covariate-Aware and Adaptive Numerical Forecasting

Numerical time-series forecasters span task-specific, covariate-aware, and universal pretrained models. Architectural designs include decomposition, patch, hierarchical interpolation, multi-period, variate-centric, frequency-domain, multiscale, non-stationarity correction, and adaptive dependency modeling Zeng et al. (2023); Nie et al. (2023); Challu et al. (2023); Wu et al. (2023); Zhou et al. (2022); Liu et al. (2022; 2024b); Wang et al. (2024a); Hu et al. (2025). For exogenous information, TFT selects relevant covariates and TimeXer separates endogenous and exogenous representations Lim et al. (2021); Wang et al. (2024b). UniTS, Moirai, TimesFM, MOMENT, Time-MoE, and Timer-XL demonstrate transfer across datasets, horizons, and forecasting configurations Gao et al. (2024); Woo et al. (2024); Das et al. (2024); Goswami et al. (2024); Shi et al. (2025); Liu et al. (2025b). Parameter-efficient methods use channel-aware low-rank adaptation or adapt frozen univariate foundation models to multivariate forecasting Nie et al. (2024); Benechehab et al. (2025). These developments motivate shared forecasting backbones that can specialize their intermediate representations to the context of each prediction window.

2.2 LLM-Enhanced Time-Series Forecasting

LLM-enhanced forecasters serialize time series for zero-shot extrapolation, reprogram numerical patches, prompt LLMs with time-series descriptions, align textual and numerical representations, or condition cross-domain and exogenous forecasting on language context Gruver et al. (2023); Zhou et al. (2023); Jin et al. (2024); Jia et al. (2024); Wang et al. (2025); Liu et al. (2025a; 2024a); Wu et al. (2026). Recent work combines textual priors with multiscale numerical features or energy-specific PV prompts Jia et al. (2026b;a). Controlled comparisons caution that some gains arise from architecture and training rather than language-specific capabilities Tan et al. (2024). This motivates a focused role for the frozen LLM: encoding forecast-available semantics to control adaptation of a numerical predictor. Our encoder-replacement and adapter-control ablations examine the contribution of this semantic pathway.

2.3 Task-Specific Energy Forecasting

Energy forecasting has largely progressed through target-specific models and protocols, while Real-E broadens evaluation toward robust and generalizable electricity forecasting across heterogeneous real-world series Shao et al. (2025). Load forecasting emphasizes weather, calendar effects, and

zonal heterogeneity Hong et al. (2016); Eren & Küçükdemiral (2024); wind forecasting combines meteorological inputs with turbine-level or spatial information Wang et al. (2021); Zhou et al. (2024); and PV forecasting incorporates physical structure, numerical weather prediction, weather foundation models, satellite signals, or spatial fusion Mayer (2022); He et al. (2025); Ma et al. (2024); Jia et al. (2026a). Net-load forecasting further couples demand and renewable uncertainty, motivating adaptive point and probabilistic models Browell & Fasiolo (2021); de Vilmarest et al. (2024); Telle et al. (2024). These studies provide domain-specific references for the four target families and highlight the importance of their distinct physical drivers and information sets. Our formulation brings these differences into an explicit forecast-origin schema for joint learning, with semantic and numerical states controlling window-specific adaptation.

3 Problem Formulation

We study joint multi-task learning across four heterogeneous datasets covering load, wind power, PV generation, and net load. At issuance time t_i, an instance $\omega_i = (q_i, a_i, t_i, L_i, H_i)$ specifies task q_i, asset or aggregation region a_i, history length L_i, and horizon H_i, with target $\mathbf{y}_i = \mathbf{y}_{i,t_i+1:t_i+H_i} \in \mathbb{R}^{H_i \times C_y(q_i)}$. Given the historical target $\mathbf{x}_i$, historical covariates $\mathbf{z}_i^{\text{hist}}$, forecast-time-known covariates $\mathbf{z}_i^{\text{known}}$, and static attributes $\mathbf{u}_i$, the model predicts $\hat{\mathbf{y}}_i$. Tasks retain their native cadences and channel sets; variable-type, cadence, task-identity, and missingness embeddings and masks preserve these distinctions within the shared backbone.

Each forecast is conditioned on information available at its issuance time. We write this information set as

$$\mathcal{I}_{i,t_i}^{\text{avail}} = \mathcal{I}_i^{\text{static}} \cup \mathcal{I}_{i,t_i}^{\text{hist}} \cup \mathcal{I}_{i,t_i}^{\text{known}} \cup \mathcal{I}_{i,t_i}^{\text{text}}.$$

The four terms comprise static attributes, observations up to t_i, forecast-time-known future covariates, and a deterministic text serialization of these fields. A field-level availability mask $\mathbf{m}_i \in \{0,1\}^F$ controls field inclusion across numerical, textual, and engineered inputs and represents the availability pattern in context. The deterministic constructor $\mathcal{C}$ maps task, asset, geographic, calendar, weather, operating, and availability fields to a semantic input s_i. We learn one shared conditional model

$$F_{\Theta;\Phi_{\text{LLM}}} : (\mathbf{x}_i, \mathbf{z}_i^{\text{hist}}, \mathbf{z}_i^{\text{known}}, \mathbf{u}_i, s_i, \mathbf{m}_i, q_i) \mapsto \hat{\mathbf{y}}_i,$$

where Φ_{LLM} is frozen and Θ is optimized jointly across all tasks. A single shared parameterization serves every task, with shared task/cadence embeddings and window-specific residual control inducing specialization. Training-split statistics define all numerical transformations and regime descriptors used by the model.

4 Methodology

Figure 1 summarizes the representation–adaptation–prediction pipeline. Forecast-available context and numerical observations form complementary states whose interaction controls gated low-rank adapters. A shared decoder and task-balanced objective complete the predictor.

4.1 Forecast-Origin-Aware Representation

The representation stage retains fine-grained continuous variables in the numerical stream and encodes task and operating semantics in a complementary text stream, within the forecast-time information boundary defined in Section 3. The deterministic constructor organizes the available context using a fixed field order:

$$\mathcal{C} : \left(\mathcal{I}_{i,t_i}^{\text{avail}}, \mathbf{m}_i, q_i, a_i\right) \mapsto s_i.$$

The schema covers task, asset, geography, issuance time, calendar, weather, recent operation, and field availability. Each included field is tagged as *static*, *observed*, *forecast*, or *known-future*. Unavailable fields contribute neither value-bearing tokens nor numerical values; only their absence is recorded in $\mathbf{m}_i$ and the availability block. For example, an hourly 24-step PV forecast may include a static station identifier and a known-future weekday indicator. It may also include a forecast-derived high-cloud regime and an observed rising-output regime. The availability record marks weather and recent operation as present, while capacity information remains unavailable.

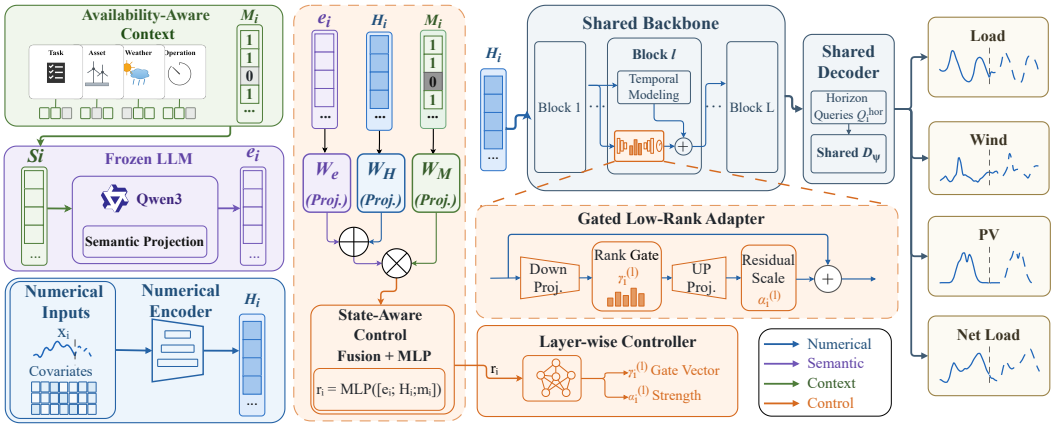


Figure 1: EnergyCA-LLM architecture. Semantic and numerical states generate window-specific control codes for gated low-rank adapters in a shared forecasting backbone; the inset details the adapter computation. Blue, purple, green, and orange arrows denote numerical, semantic, context, and control flows, respectively.

Continuous weather and operating variables remain in the numerical stream. Only training-split-defined, low-cardinality regime descriptors enter the text. All field rules, thresholds, and missing-value conventions are fixed before validation and testing. This construction also supports controlled field removal, shuffling, and counterfactual-style sensitivity edits.

Let $\mathbf{Z}_i^{\mathrm{LLM}} \in \mathbb{R}^{M_i \times d_{\mathrm{LLM}}}$ denote token states from the frozen language model and $\eta_{i,j} \in \{0,1\}$ its attention mask. Masked pooling followed by a trainable projection gives the semantic representation

$$\mathbf{e}_i = \mathrm{LN}\left(\mathbf{W}_e \frac{\sum_{j=1}^{M_i} \eta_{i,j} \mathbf{Z}_{i,j}^{\mathrm{LLM}}}{\sum_{j=1}^{M_i} \eta_{i,j}}\right) \in \mathbb{R}^{d_e}. \tag{1}$$

The forecasting model updates $\mathbf{W}_e$ while keeping the language model frozen.

The complementary numerical stream preserves the temporal resolution omitted from the semantic summary. Following the endogenous–exogenous separation in TimeXer Wang et al. (2024b), historical targets and covariates form temporal tokens, while forecast-known covariates, static attributes, and field availability form an auxiliary stream:

$$\begin{aligned} \mathbf{T}_{i,\mathrm{hist}}^{(0)} &= \mathrm{PatchEmbed}\big([\mathbf{x}_i; \mathbf{z}_i^{\mathrm{hist}}]\big) + \mathbf{E}_i^{\mathrm{var}} + \mathbf{E}_i^{\mathrm{cad}} + \mathbf{E}_i^{\mathrm{task}} + \mathbf{E}_i^{\mathrm{pos}}, \\ \mathbf{T}_{i,\mathrm{known}} &= \mathrm{AuxEmbed}\big(\mathbf{z}_i^{\mathrm{known}}, \mathbf{u}_i, \mathbf{m}_i\big), \\ \mathbf{H}_i^{(0)} &= \mathrm{Stem}\big(\mathbf{T}_{i,\mathrm{hist}}^{(0)}\big). \end{aligned} \tag{2}$$

Let $\mathbf{G}_i^{(0)} = \mathbf{H}_i^{(0)}$. Each shared block then models temporal dependence and injects forecast-known information:

$$\begin{aligned} \mathbf{U}_i^{(l)} &= \mathbf{G}_i^{(l-1)} + \mathrm{MSA}\big(\mathrm{LN}\big(\mathbf{G}_i^{(l-1)}\big)\big), \\ \mathbf{V}_i^{(l)} &= \mathbf{U}_i^{(l)} + \mathrm{CrossAttn}\big(\mathrm{LN}\big(\mathbf{U}_i^{(l)}\big), \mathbf{T}_{i,\mathrm{known}}, \mathbf{T}_{i,\mathrm{known}}\big), \\ \mathbf{H}_i^{(l)} &= \mathbf{V}_i^{(l)} + \mathrm{FFN}\big(\mathrm{LN}\big(\mathbf{V}_i^{(l)}\big)\big). \end{aligned} \tag{3}$$

Patch projections, attention blocks, and feed-forward layers are shared across tasks; variable-type embeddings, channel padding, and masks accommodate their different channels, cadences, and sequence lengths.

4.2 Context-Controlled Residual Adaptation

To condition specialization on both operating context and the current trajectory, EnergyCA-LLM combines semantic, numerical, and availability states in a window-specific control code that modulates the shared backbone.

Window-level residual control. The controller pools the unmodulated base state, embeds field availability, and combines additive and multiplicative semantic–numerical interactions:

$$\begin{aligned} \bar{\mathbf{H}}_i &= \mathrm{AttnPool}\big(\mathbf{H}_i^{(0)}\big), & \mathbf{v}_i &= \mathbf{W}_m \mathbf{m}_i, \\ \mathbf{c}_i &= [\mathbf{e}_i; \bar{\mathbf{H}}_i; \mathbf{v}_i], & \mathbf{h}_i^{\mathrm{ctrl}} &= \mathrm{SiLU}\big(\mathbf{W}_1 \mathbf{c}_i\big), \\ \mathbf{b}_i &= \big(\mathbf{W}_2 \mathbf{e}_i\big) \odot \big(\mathbf{W}_3 \bar{\mathbf{H}}_i\big), & \mathbf{r}_i &= \mathrm{LN}\big(\mathbf{W}_4 [\mathbf{h}_i^{\mathrm{ctrl}}; \mathbf{b}_i]\big). \end{aligned} \tag{4}$$

The semantic–numerical interaction lets identical context descriptions induce different adaptations under different recent trajectories, while forecast-time context and availability can distinguish windows with similar trajectories.

Layer-wise gated low-rank adaptation. For each selected layer $l \in \mathcal{L}_{\mathrm{mod}}$, EnergyCA-LLM learns shared low-rank bases $\mathbf{A}^{(l)} \in \mathbb{R}^{d \times r}$ and $\mathbf{B}^{(l)} \in \mathbb{R}^{r \times d}$, where $r \ll d$. The window code controls both rank components and overall residual strength:

$$\boldsymbol{\gamma}_i^{(l)} = \tanh(\mathbf{W}_g^{(l)} \mathbf{r}_i), \qquad \alpha_i^{(l)} = \sigma\Big((\mathbf{w}_\alpha^{(l)})^\top \mathbf{r}_i + b_\alpha^{(l)}\Big). \tag{5}$$

The resulting gated residual update is

$$\begin{aligned} \Delta \mathbf{H}_i^{(l)} &= \big(\big(\mathbf{H}_i^{(l)} \mathbf{A}^{(l)}\big) \odot \boldsymbol{\gamma}_i^{(l)}\big) \mathbf{B}^{(l)}, \\ \widetilde{\mathbf{H}}_i^{(l)} &= \mathbf{H}_i^{(l)} + \alpha_i^{(l)} \Delta \mathbf{H}_i^{(l)}. \end{aligned} \tag{6}$$

The low-rank bases are shared; their rank weighting and residual strength vary by window. For an unselected layer, $\widetilde{\mathbf{H}}_i^{(l)} = \mathbf{H}_i^{(l)}$; in both cases, $\mathbf{G}_i^{(l)} = \widetilde{\mathbf{H}}_i^{(l)}$ enters the next shared block. We initialize $\mathbf{B}^{(l)} = \mathbf{0}$ and $b_\alpha^{(l)} < 0$, so optimization begins from the shared numerical forecaster.

4.3 Unified Prediction Objective

A shared decoder maps the adapted representation to task-dependent horizons using task- and cadence-conditioned queries:

$$\mathbf{Q}_i^{\mathrm{hor}} = \mathbf{Q}_{1:H_i}^{\mathrm{pos}} + \mathbf{e}_{q_i}^{\mathrm{task}} + \mathbf{e}_{q_i}^{\mathrm{cad}}, \qquad \hat{\mathbf{y}}_i = D_\psi\big(\widetilde{\mathbf{H}}_i^{(L)}, \mathbf{Q}_i^{\mathrm{hor}}\big). \tag{7}$$

Horizon and channel masks select valid output steps and dimensions, with task information conditioning the shared query.

To balance tasks with different scales and sample counts, let $\kappa_{q,c} > 0$ be a scale computed only from the training split for task q and target channel c. We first define the scale-normalized Huber prediction objective

$$\begin{aligned} \ell_i &= \frac{1}{H_i C_y(q_i)} \sum_{h=1}^{H_i} \sum_{c=1}^{C_y(q_i)} \mathrm{Huber}_\delta\left(\frac{\hat{y}_{i,h,c} - y_{i,h,c}}{\kappa_{q_i,c}}\right), \\ \mathcal{L}_{\mathrm{pred}} &= \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{1}{|\mathcal{B}_q|} \sum_{i \in \mathcal{B}_q} \ell_i. \end{aligned} \tag{8}$$

Task-balanced averaging prevents larger datasets from dominating through window count. Regularizing the context-driven corrections gives the final objective

$$\begin{aligned} \mathcal{L}_{\mathrm{res}} &= \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{l \in \mathcal{L}_{\mathrm{mod}}} \frac{\big\|\Delta \mathbf{H}_i^{(l)}\big\|_F^2}{N_i d}, \\ \mathcal{L} &= \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{res}} \mathcal{L}_{\mathrm{res}}. \end{aligned} \tag{9}$$

We jointly optimize the numerical stem and backbone, semantic projection, residual-code generator, low-rank bases, controllers, and decoder while keeping the language model frozen. The learned parameter set serves all four task families at inference, without task-specific checkpoints or post hoc prediction blending.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets and splits. We evaluate four public energy benchmarks: GEFCom2014 Load Task 15 for hourly load forecasting Hong et al. (2016), SDWPF for 10-minute wind-power forecasting with turbine SCADA measurements Zhou et al. (2024), MMSP/FusionSF for hourly normalized PV forecasting with numerical weather predictions Ma et al. (2024), and the half-hourly Great Britain GSP-group benchmark for regional net-load forecasting with weather and capacity covariates Browell & Fasiolo (2021); de Vilmarest et al. (2022). All datasets use chronological train/validation/test partitions. Net-load forecasts are issued at 00:00 with lead times from 22 to 47.5 hours.

At their native cadences, Load and PV use $(L, H) = (168, 24)$, Wind uses $(1008, 144)$, and Net load uses $(336, 48)$ for day-ahead multi-horizon forecasting. The context schema combines task and energy type, asset or region metadata, calendar indicators, weather summaries, recent operating states, and field-availability indicators. Normalization and context preprocessing are fitted on the training partitions.

Baselines. The twelve baselines in Table 1 span three groups. General numerical methods include Persistence, DLinear Zeng et al. (2023), PatchTST Nie et al. (2023), TimeXer Wang et al. (2024b), and TFT Lim et al. (2021). LLM-enhanced methods include Time-LLM Jin et al. (2024) and CALF Liu et al. (2025a). Task-specific references include a calendar–temperature Load-GAM Hong & Fan (2016), the SDWPF Future-SCADA predictor Zhou et al. (2024), Clear-Sky and FusionSF for PV Ma et al. (2024), and GAM for net load de Vilmarest et al. (2024); Telle et al. (2024).

Metrics and shared protocol. We report NMAE, RMSE, sMAPE, and MASE, with lower values indicating better forecasting performance. NMAE normalizes absolute errors by the aggregate target magnitude, while MASE uses an in-sample daily seasonal naive forecast as its reference, with seasonal periods of 24, 144, 24, and 48 steps for Load, Wind, PV, and Net load, respectively. RMSE is reported in each dataset's native processed target units, whereas the other three metrics are scale-free.

The shared-model protocol uses task-balanced batches from the four training partitions and evaluates one jointly trained checkpoint per seed on all four tasks. Task and cadence embeddings together with horizon, channel, and availability masks provide the task interface. Protocol-matched reruns use the same chronological splits, horizons, historical inputs, and forecast-available exogenous variables.

Implementation. EnergyCA-LLM uses a frozen Qwen3-8B context encoder, a model dimension of 128, four shared temporal blocks, rank-8 residual adapters, and dropout of 0.1. All trainable comparison methods use the same training budget as EnergyCA-LLM. Detailed optimization settings and the common training budget are provided in Appendix A.3. Trainable methods are evaluated over three seeds, and results are reported as means with population standard deviations.

5.2 OVERALL FORECASTING PERFORMANCE

Using one jointly trained checkpoint per seed across all four tasks, EnergyCA-LLM achieves the lowest reported error in 15 of the 16 task–metric comparisons (Table 1). Load sMAPE decreases from 5.16 to 3.11, Wind sMAPE from 96.81 to 35.09 (approximately 63.7%), and Net-load RMSE from 136.87 to 118.08 (approximately 13.7%). For PV, sMAPE decreases from 36.84 to 31.70 (approximately 14.0%), while MASE decreases from the lowest baseline value of 0.71 to 0.65. Wind MASE decreases from 0.24 for SDWPF to 0.20. PV RMSE is the exception: EnergyCA-LLM gives 0.100042 versus 0.099877 for FusionSF, a relative increase of approximately 0.17%.

Figure 2 complements the aggregate metrics with trajectory-level behavior. EnergyCA-LLM follows the recurring load cycles, the timing of PV generation pulses, and the principal net-load peaks while reducing the short-scale deviations visible in the general numerical forecast. In the Wind window, it tracks the transition from the high-output regime to the low-output regime and follows the observed decline more closely than the general numerical and LLM-enhanced forecasts. The task expert remains competitive around several local extrema, while EnergyCA-LLM achieves lower aggregate Wind MASE in Table 1.

Panel A: Load and wind forecasting

Category	Method	Load NMAE	Load RMSE	Load sMAPE	Load MASE	Wind NMAE	Wind RMSE	Wind sMAPE	Wind MASE
General numerical	Persistence	0.14 ± 0.00	25.39 ± 0.00	14.26 ± 0.00	0.91 ± 0.00	0.94 ± 0.00	436.35 ± 0.00	100.48 ± 0.00	0.73 ± 0.00
	DLinear	0.13 ± 0.00	22.51 ± 1.02	13.27 ± 0.43	0.83 ± 0.03	1.38 ± 0.12	502.78 ± 41.67	128.97 ± 1.67	1.07 ± 0.10
	PatchTST	0.05 ± 0.00	9.19 ± 0.10	5.19 ± 0.11	0.33 ± 0.01	0.60 ± 0.07	255.75 ± 16.21	111.21 ± 0.98	0.46 ± 0.05
	TimeXer	0.05 ± 0.00	9.39 ± 0.29	5.26 ± 0.18	0.33 ± 0.01	0.56 ± 0.02	250.26 ± 10.22	113.18 ± 1.86	0.44 ± 0.02
	TFT	0.05 ± 0.00	9.17 ± 0.46	5.16 ± 0.25	0.33 ± 0.02	0.67 ± 0.03	278.52 ± 8.26	116.04 ± 0.99	0.52 ± 0.02
LLM-enhanced	Time-LLM	0.12 ± 0.00	19.90 ± 0.77	11.76 ± 0.45	0.76 ± 0.03	1.04 ± 0.02	412.40 ± 4.21	129.54 ± 0.72	0.81 ± 0.02
	CALF	0.11 ± 0.00	19.13 ± 0.41	11.08 ± 0.26	0.70 ± 0.02	1.02 ± 0.05	415.27 ± 16.28	137.95 ± 1.47	0.79 ± 0.04
Task-specific	Load-GAM	0.05 ± 0.00	9.88 ± 0.00	5.39 ± 0.00	0.35 ± 0.00	–	–	–	–
	SDWPF	–	–	–	–	0.31 ± 0.00	159.42 ± 0.58	96.81 ± 0.04	0.24 ± 0.00
EnergyCA-LLM		**0.03 ± 0.01**	**5.87 ± 0.69**	**3.11 ± 0.06**	**0.20 ± 0.01**	**0.25 ± 0.03**	**130.82 ± 1.82**	**35.09 ± 0.08**	**0.20 ± 0.03**

Panel B: PV and net-load forecasting

Category	Method	PV NMAE	PV RMSE	PV sMAPE	PV MASE	Net-load NMAE	Net-load RMSE	Net-load sMAPE	Net-load MASE
General numerical	Persistence	1.00 ± 0.00	0.29 ± 0.00	104.68 ± 0.00	2.47 ± 0.00	0.21 ± 0.00	360.33 ± 0.00	22.65 ± 0.00	1.66 ± 0.00
	DLinear	0.35 ± 0.00	0.11 ± 0.00	118.15 ± 0.80	0.87 ± 0.01	0.28 ± 0.00	513.98 ± 19.86	36.15 ± 0.41	2.18 ± 0.02
	PatchTST	0.30 ± 0.01	0.10 ± 0.00	112.49 ± 0.76	0.73 ± 0.02	0.16 ± 0.00	276.02 ± 2.15	17.96 ± 0.80	1.25 ± 0.01
	TimeXer	0.29 ± 0.01	0.10 ± 0.00	112.41 ± 0.36	0.72 ± 0.01	0.17 ± 0.04	285.12 ± 64.15	20.79 ± 7.06	1.35 ± 0.33
	TFT	0.29 ± 0.01	0.10 ± 0.00	112.46 ± 0.54	0.71 ± 0.02	0.17 ± 0.02	276.79 ± 33.50	20.60 ± 4.12	1.28 ± 0.17
LLM-enhanced	Time-LLM	0.45 ± 0.04	0.12 ± 0.00	120.10 ± 1.81	1.10 ± 0.09	0.09 ± 0.00	171.59 ± 7.24	9.73 ± 0.51	0.73 ± 0.04
	CALF	0.38 ± 0.01	0.11 ± 0.00	117.69 ± 0.74	0.95 ± 0.01	0.09 ± 0.00	160.17 ± 6.11	9.10 ± 0.55	0.68 ± 0.03
Task-specific	Clear-Sky	0.37 ± 0.00	0.13 ± 0.00	99.13 ± 0.00	0.91 ± 0.00	–	–	–	–
	FusionSF	0.29 ± 0.01	**0.10 ± 0.00**	36.84 ± 2.14	0.71 ± 0.03	–	–	–	–
	GAM	–	–	–	–	0.07 ± 0.00	136.87 ± 0.00	7.27 ± 0.00	0.51 ± 0.00
EnergyCA-LLM		**0.26 ± 0.01**	0.10 ± 0.00	**31.70 ± 1.66**	**0.65 ± 0.09**	**0.06 ± 0.00**	**118.08 ± 0.48**	**6.82 ± 0.39**	**0.50 ± 0.08**

Table 1: Point-forecasting results across four benchmarks and twelve baselines (lower is better). Values are shown with population standard deviations. Bold/underlining mark best/second-best values. PV RMSE before rounding is 0.099877 for FusionSF and 0.100042 for EnergyCA-LLM.

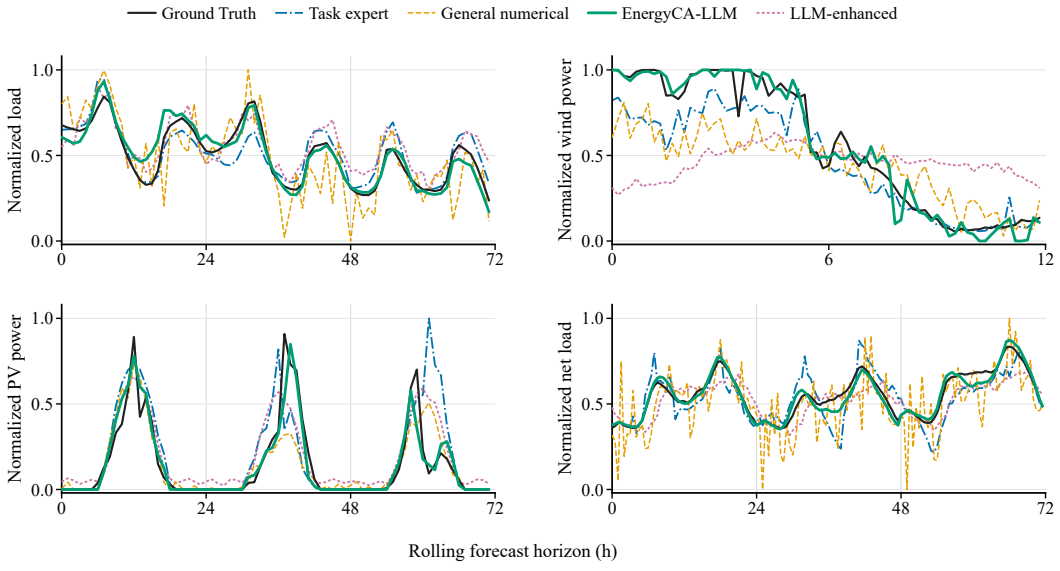


Figure 2: Rolling forecasts at native cadences over selected high-volatility windows: 12 hours for Wind and 72 hours for Load, PV, and Net load. Each panel compares observations, EnergyCA-LLM, and representative task-matched general numerical, LLM-enhanced, and task-expert baselines on the same window; Table 1 reports aggregate results.

5.3 Unified Training and Mechanism Analysis

Table 2 tests whether a shared model transfers across tasks, pretrained semantic encoding improves on simpler context encoders, and dynamic adapter control outperforms fixed residual capacity. All evaluated variants follow the same chronological protocol and use three seeds.

Panel A: Unified-training architecture

Configuration	Models/seed	Ckpts./seed	Load	Wind	PV	Net load
Four independent numerical models	4	4	0.132 ± 0.005	1.043 ± 0.018	0.371 ± 0.006	0.089 ± 0.001
Shared numerical backbone	1	1	0.095 ± 0.013	0.412 ± 0.029	0.350 ± 0.018	0.081 ± 0.003
Shared backbone + task interface	1	1	0.079 ± 0.022	0.359 ± 0.038	0.332 ± 0.034	0.078 ± 0.007
EnergyCA-LLM	1	1	$\mathbf{0.031 \pm 0.009}$	$\mathbf{0.253 \pm 0.026}$	$\mathbf{0.264 \pm 0.009}$	$\mathbf{0.063 \pm 0.003}$

Panel B: Semantic-context mechanism

Configuration	Pretr.	Frozen	Avail.	Load	Wind	PV	Net load
No context	–	–	–	0.141 ± 0.002	1.046 ± 0.021	0.382 ± 0.005	0.085 ± 0.002
Task/asset embeddings	No	No	–	0.057 ± 0.023	0.430 ± 0.027	0.344 ± 0.020	0.079 ± 0.013
MLP schema encoder	No	No	Yes	0.056 ± 0.022	0.418 ± 0.081	0.329 ± 0.032	0.078 ± 0.008
Fixed random text embeddings	No	Yes	Yes	0.094 ± 0.008	0.847 ± 0.035	0.374 ± 0.010	0.089 ± 0.008
EnergyCA-LLM	Yes	Yes	Yes	$\mathbf{0.031 \pm 0.009}$	$\mathbf{0.253 \pm 0.026}$	$\mathbf{0.264 \pm 0.009}$	$\mathbf{0.063 \pm 0.003}$

Panel C: Dynamic-adapter control

Configuration	Rank gate	Strength gate	Load	Wind	PV	Net load
Fixed shared residual adapter	No	No	0.053 ± 0.021	0.426 ± 0.059	0.347 ± 0.016	0.087 ± 0.002
EnergyCA-LLM	Yes	Yes	$\mathbf{0.031 \pm 0.009}$	$\mathbf{0.253 \pm 0.026}$	$\mathbf{0.264 \pm 0.009}$	$\mathbf{0.063 \pm 0.003}$

Table 2: Unified-training and mechanism analysis. NMAE values are shown with population standard deviations. Evaluated Panel B/C variants use one model and checkpoint. The EnergyCA-LLM rows use the shared checkpoints from Table 1. In Panel B, Avail. marks the availability-aware schema; No context removes the schema and semantic state while retaining target history and exogenous numerical channels.

Unified numerical training. Panel A compares four independent numerical predictors with progressively richer shared variants. The shared backbone lowers mean NMAE on all four tasks, indicating transferable temporal parameters, and the task interface provides further reductions. The full model gives the lowest NMAE on all tasks (0.031, 0.253, 0.264, and 0.063) using one model and checkpoint per seed.

Pretrained semantic encoding. Panel B varies the semantic encoder while retaining numerical inputs. The complete pathway lowers the equally weighted four-task mean NMAE from 0.414 to 0.153 (approximately 63.0%) relative to No context, with the largest absolute change on Wind (1.046 to 0.253). It also improves on the MLP schema encoder (0.220 to 0.153; approximately 30.6%) and every evaluated replacement on each task. Task/asset embeddings and MLP encoding perform similarly; fixed random text embeddings are substantially worse. These results suggest that pretrained semantics complement numerical features beyond providing an embedding channel.

Dynamic residual control. Panel C replaces the window-specific rank and strength gates with a fixed residual adapter. NMAE rises from 0.031 to 0.053 on Load, 0.253 to 0.426 on Wind, 0.264 to 0.347 on PV, and 0.063 to 0.087 on Net load, linking gains across all tasks to state-dependent control rather than residual capacity alone.

EnergyCA-LLM has the lowest NMAE in all twelve task–panel comparisons among the evaluated ablations. The largest relative NMAE reduction in each panel occurs on Load: approximately 76.5%, 78.0%, and 41.5% relative to the worst-performing evaluated variants in Panels A, B, and C, respectively. The corresponding Wind reductions are approximately 75.7%, 75.8%, and 40.5%. These results show gains across all four tasks, with the largest relative effects on Load and Wind.

6 Conclusion

EnergyCA-LLM connects forecast-origin-aware context to window-specific adaptation of a shared numerical forecaster for load, wind power, PV generation, and net load. A frozen pretrained LLM

supplies a semantic state that interacts with the numerical representation to control gated low-rank residual adapters. Across four public day-ahead benchmarks, the model achieves the lowest reported error in 15 of the 16 task–metric comparisons against twelve baselines, with PV RMSE close to FusionSF; controlled ablations support the contributions of unified training, pretrained semantic encoding, and dynamic residual control.

The evaluated setting covers four public benchmarks, point forecasts, and their context-availability regimes. Future work will extend the framework to additional renewable assets, probabilistic forecasting, richer weather products, longer horizons, and more diverse availability regimes.

REFERENCES

Abdelhakim Benechehab, Vasilii Feofanov, Giuseppe Paolo, Albert Thomas, Maurizio Filippone, and Balázs Kégl. AdaPTS: Adapting univariate foundation models to probabilistic multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 3731–3748, 2025. URL https://proceedings.mlr.press/v267/benechehab25a.html.

Jethro Browell and Matteo Fasiolo. Probabilistic forecasting of regional net-load with conditional extremes and gridded NWP. *IEEE Transactions on Smart Grid*, 12(6):5011–5019, 2021. doi: 10.1109/TSG.2021.3107159. URL https://doi.org/10.1109/TSG.2021.3107159.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. N-HiTS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023. doi: 10.1609/aaai.v37i6.25854. URL https://doi.org/10.1609/aaai.v37i6.25854.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 10148–10167, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Joseph de Vilmarest, Jethro Browell, Matteo Fasiolo, Yannig Goude, and Olivier Wintenberger. Supplementary materials for "adaptive probabilistic forecasting of electricity (net-)load", 2022. URL https://doi.org/10.5281/zenodo.7849665. Supplementary dataset including Great Britain regional net-load setting.

Joseph de Vilmarest, Jethro Browell, Matteo Fasiolo, Yannig Goude, and Olivier Wintenberger. Adaptive probabilistic forecasting of electricity (net-) load. *IEEE Transactions on Power Systems*, 39(2):4154–4163, 2024. doi: 10.1109/TPWRS.2023.3310280.

Yavuz Eren and Ibrahim Küçükdemiral. A comprehensive review on deep learning approaches for short-term load forecasting. *Renewable and Sustainable Energy Reviews*, 189:114031, 2024. doi: 10.1016/j.rser.2023.114031. URL https://doi.org/10.1016/j.rser.2023.114031.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. UniTS: A unified multi-task time series model. In *Advances in Neural Information Processing Systems*, volume 37, pp. 140589–140631, 2024. doi: 10.52202/079017-4463. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/fe248e22b241ae5a9adf11493c8c12bc-Abstract-Conference.html.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 16115–16152, 2024. URL https://proceedings.mlr.press/v235/goswami24a.html.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, volume 36, pp. 19622–19635, 2023. doi: 10.52202/075280-0861. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3eb7ca52e8207697361b2c0fb3926511-Abstract-Conference.html.

Fang He, Jiaqi Fan, Yang Deng, Xiaoyang Zhang, Ka Tai Lau, and Dan Wang. Weather foundation model enhanced decentralized photovoltaic power forecasting through spatio-temporal knowledge distillation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 9665–9673, 2025. doi: 10.24963/ijcai.2025/1074. URL https://doi.org/10.24963/ijcai.2025/1074.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016. doi: 10.1016/j.ijforecast.2015.11.011. URL https://doi.org/10.1016/j.ijforecast.2015.11.011.

Tao Hong, Pierre Pinson, Shu Fan, Hamidreza Zareipour, Alberto Troccoli, and Rob J. Hyndman. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *International Journal of Forecasting*, 32(3):896–913, 2016. doi: 10.1016/j.ijforecast.2016.02.001. URL https://doi.org/10.1016/j.ijforecast.2016.02.001.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. TimeFilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 24893–24911, 2025. URL https://proceedings.mlr.press/v267/hu25ac.html.

Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 23343–23351, 2024. doi: 10.1609/aaai.v38i21.30383. URL https://ojs.aaai.org/index.php/AAAI/article/view/30383.

Pengfei Jia, Jianghong Ma, Baoquan Zhang, Kenghong Lin, Xinyu Zhang, Chuyao Luo, Xutao Li, and Yunming Ye. Satellite-text-prompted large language model for photovoltaic power forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 38644–38652, 2026a. doi: 10.1609/aaai.v40i45.41207. URL https://doi.org/10.1609/aaai.v40i45.41207.

Shuning Jia, Baijun Song, Canming Ye, and Chun Yuan. M3time: Llm-enhanced multi-modal, multi-scale, and multi-frequency multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22265–22273, 2026b. doi: 10.1609/aaai.v40i27.39383. URL https://ojs.aaai.org/index.php/AAAI/article/view/39383.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Bryan Lim, Sercan Ö. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. doi: 10.1016/j.ijforecast.2021.03.012. URL https://doi.org/10.1016/j.ijforecast.2021.03.012.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025a. doi: 10.1609/aaai.v39i18.34082. URL https://ojs.aaai.org/index.php/AAAI/article/view/34082.

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. UniTime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, pp. 4095–4106, 2024a. doi: 10.1145/3589334.3645434. URL https://doi.org/10.1145/3589334.3645434.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022. doi: 10.52202/068431-0718. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4054556fcaa934b0bf76da52cf4f92cb-Abstract-Conference.html.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=JePfAI8fah.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=KMCJXjlDDr.

Ziqing Ma, Wenwei Wang, Tian Zhou, Chao Chen, Bingqing Peng, Liang Sun, and Rong Jin. Fusionsf: Fuse heterogeneous modalities in a vector quantized framework for robust solar power forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5532–5543, 2024. doi: 10.1145/3637528.3671509. URL https://doi.org/10.1145/3637528.3671509.

Martin János Mayer. Benefits of physical and machine learning hybridization for photovoltaic power forecasting. *Renewable and Sustainable Energy Reviews*, 168:112772, 2022. doi: 10.1016/j.rser.2022.112772. URL https://doi.org/10.1016/j.rser.2022.112772.

Tong Nie, Yuewen Mei, Guoyang Qin, Jian Sun, and Wei Ma. Channel-aware low-rank adaptation in time series forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3959–3963, 2024. doi: 10.1145/3627673.3679884. URL https://doi.org/10.1145/3627673.3679884.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Chen Shao, Michael Färber, Sebastian Pütz, Benjamin Schäfer, Yue Wang, Tobias Käfer, Zhanbo Huang, and Zhenyi Zhu. Real-e: A foundation benchmark for advancing robust and generalizable electricity forecasting. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 6523–6527, 2025. doi: 10.1145/3746252.3761637. URL https://dl.acm.org/doi/10.1145/3746252.3761637.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e1wDDFmlVu.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems*, volume 37, pp. 60162–60191, 2024. doi: 10.52202/079017-1922. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6ed5bf446f59e2c6646d23058c86424b-Abstract-Conference.html.

J. Telle, Ajay Upadhaya, Patrik Schönfeldt, T. Steens, Benedikt Hanke, and Karsten von Maydell. Probabilistic net load forecasting framework for application in distributed integrated renewable energy systems. *Energy Reports*, 11:2535–2553, 2024. doi: 10.1016/j.egyr.2024.02.015. URL https://www.sciencedirect.com/science/article/pii/S2352484724000969.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. ChatTime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12694–12702, 2025. doi: 10.1609/aaai.v39i12.33384. URL https://doi.org/10.1609/aaai.v39i12.33384.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=7oLshfEIC2.

Yun Wang, Runmin Zou, Fang Liu, Lingjun Zhang, and Qianyi Liu. A review of wind speed and wind power forecasting with deep neural networks. *Applied Energy*, 304:117766, 2021. doi: 10.1016/j.apenergy.2021.117766. URL https://doi.org/10.1016/j.apenergy.2021.117766.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems*, volume 37, pp. 469–498, 2024b. doi: 10.52202/079017-0015. URL https://papers.nips.cc/paper_files/paper/2024/hash/0113ef4642264adc2e6924a3cbbdf532-Abstract-Conference.html.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 53140–53164, 2024. URL https://proceedings.mlr.press/v235/woo24a.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Lan Wu, Xuebin Wang, Chenglong Ge, Ruijuan Chu, and Linyu Wang. ExoTimer: Leveraging large language models for time series forecasting with exogenous variables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26956–26964, 2026. doi: 10.1609/aaai.v40i32.39908. URL https://doi.org/10.1609/aaai.v40i32.39908.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317. URL https://doi.org/10.1609/aaai.v37i9.26317.

Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jian Tang, Jiantao Su, Yu Li, Ji Liu, Junfu Lyu, Yanjun Ma, and Dejing Dou. Sdwpf: A dataset for spatial dynamic wind power forecasting over a large turbine array. *Scientific Data*, 11(1):649, 2024. doi: 10.1038/s41597-024-03427-5. URL https://doi.org/10.1038/s41597-024-03427-5.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 27268–27286, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, volume 36, pp. 43322–43355, 2023. doi: 10.52202/075280-1877. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/86c17de05579cde52025f9984e6e2ebb-Abstract-Conference.html.

A SUPPLEMENTARY MATERIAL

This appendix consolidates the benchmark interfaces, forecast-origin context rules, and training configuration used in the main manuscript.

A.1 BENCHMARK INTERFACES AND SHARED-MODEL PROTOCOL

The four forecasting interfaces retain each dataset's native time steps. Load uses GEFCom2014 Load Task 15 at an hourly cadence with history length $L = 168$ and horizon $H = 24$. Wind uses SDWPF at a 10-minute cadence with $(L, H) = (1008, 144)$. PV uses MMSP/FusionSF at an hourly cadence with $(L, H) = (168, 24)$. Sequence lengths count time steps, and horizon lengths specify the number of outputs. The model retains these cadences, channel sets, and horizons through task/cadence embeddings and channel, horizon, and availability masks.

The benchmarks provide different auxiliary information: turbine SCADA measurements for Wind, numerical weather predictions for PV, and weather and capacity covariates for Net load. All four datasets use chronological training, validation, and test partitions. Numerical normalization and context preprocessing are fitted on training partitions. Validation selects checkpoints, and test partitions provide the reported evaluation.

For each seed, task-balanced training jointly updates one forecasting model using windows from all four training partitions. At inference, the resulting shared checkpoint serves Load, Wind, PV, and Net load through their native task interfaces. The frozen context encoder, shared numerical backbone, and window-specific controller together define this predictor. The four independent numerical models in Table 2, Panel A, instead use four models and four checkpoints per seed.

A.2 CONTEXT FIELDS AND FORECAST-ORIGIN AVAILABILITY

The deterministic schema uses the fixed field order described in the main text: task, asset, geography, issuance time, calendar, weather, recent operation, and field availability. The status of a field follows how its value is obtained and when it becomes available for the particular forecast. Available context fields vary across benchmarks.

The four source-status tags are *static*, *observed*, *forecast*, and *known-future*. Static fields contain attributes known at the forecast origin, such as asset identifiers or available geographic metadata. Observed fields contain measurements available no later than issuance, including recent operating states derived from historical observations. Forecast fields contain products available when the prediction is issued, such as weather-regime descriptors derived from available weather forecasts. Known-future fields contain future attributes already known at issuance, including calendar information for the prediction horizon. Unavailable fields are recorded by the field mask and availability block; their value-bearing text and numerical values are omitted.

Continuous weather and operating measurements remain in the numerical stream. Text contains low-cardinality regime descriptors defined from training data, together with the available task and contextual information. Field rules, descriptor thresholds, and missing-value conventions are fixed before validation and testing. The availability mask governs inclusion in both streams: unavailable fields supply neither value-bearing text tokens nor numerical values, while their absence remains visible to the controller.

The PV example in the main text illustrates this separation. An hourly 24-step forecast can include a static station identifier, a known-future weekday indicator, a forecast-derived high-cloud regime, and an observed rising-output regime. Weather and recent operation are marked present, while unavailable capacity is represented only through its absence. This illustrative example shows the roles of the context fields.

The frozen encoder produces token hidden states, which are pooled using the attention mask. A trainable projection and layer normalization form the semantic state. The numerical state comes from the historical stream, and the controller combines these two states with the availability representation. Shared low-rank bases therefore receive different rank and strength gates for different windows even when the forecasting checkpoint is unchanged.

A.3 Model Configuration and Common Training Budget

All trainable comparison methods use the same budget as EnergyCA-LLM: at most 20 epochs with 128 optimizer updates per epoch and early-stopping patience of five epochs. This gives an upper limit of 2,560 optimizer updates per seed; early stopping can reduce the actual number. Each trainable method is evaluated with seeds 42, 2027, and 3407.

EnergyCA-LLM uses a frozen Qwen3-8B context encoder, a numerical model dimension of 128, four shared temporal blocks, rank-8 residual adapters, and dropout of 0.1. AdamW uses a learning rate of 3×10^{-4}, weight decay of 10^{-4}, and gradient clipping at 1.0. Batch sizes for Load, Wind, PV, and Net load are 16, 2, 16, and 16, respectively. Training uses BF16 precision on one NVIDIA RTX 5090 GPU, with checkpoints selected by validation NMAE.

For EnergyCA-LLM, each update is task-balanced across the four training partitions. The language model remains frozen. The numerical stem and backbone, semantic projection, residual-code generator, adapter bases, gates, and decoder are optimized jointly.

The prediction objective averages scale-normalized Huber losses within each task and then equally across the four tasks. Training-split target scales define the normalization in this objective. The residual penalty in Equation 9 regularizes the low-rank corrections. The output basis of each adapter is initialized to zero and its strength-gate bias is negative, so training starts from the shared numerical forecaster.

Reported results summarize three independently trained seeds as the arithmetic mean plus or minus the population standard deviation, using the three seeds as the population for this summary. Each seed contributes one shared checkpoint for the complete model. The daily seasonal reference used by MASE has periods of 24, 144, 24, and 48 steps for Load, Wind, PV, and Net load, respectively.

Metric aggregation and consistency. NMAE and MASE aggregate all valid forecast points. For a task with a common evaluation mask and weights, let A be test MAE, U the mean absolute test target, and S the training-set daily seasonal naive MAE. When each task uses one global normalization scale for each metric, fixed across methods and seeds, NMAE equals A/U and MASE equals A/S. Thus $c = U/S$ gives $\mathrm{MASE} = c\,\mathrm{NMAE}$, and both the seed mean and population standard deviation scale by c. A per-asset seasonal scale followed by pooled averaging is a different aggregation rule and must be specified separately. Relative reductions in the main text use the available unrounded values where supplied and otherwise the displayed table values, before rounding the resulting percentages.