A CONDITIONAL KOOPMAN WORLD MODEL FOR DISTRIBUTION-AWARE ELECTRICITY FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Electric load and price time series exhibit typical damped oscillation patterns with periodic cycles and spike mean reversion, which are dominated by the dynamic evolution of power supply and demand contradictions. Existing time series forecasting models rarely incorporate such physical inductive biases, making it challenging to simultaneously achieve stable long-horizon rollouts, event-conditioned forecasting, and counterfactual simulation. To address this issue, this paper proposes the power world model (PWM), a conditional Koopman world model for distribution-aware electricity forecasting. Leveraging the dual core mechanisms of amplitude-conserving rotation and amplitude-dissipating contraction inherent to damped oscillation, PWM constructs a latent state space through delay embedding and designs a structurally decomposable condition-affine Koopman operator. The operator structurally limits the spectral radius to no more than 1, ensuring bounded and stable long-horizon forecasting results. Unlike conventional conditional forecasting methods that integrate conditional information at the input layer, our method embeds conditional signals into the state transition operator, enabling efficient event-conditioned forecasting and low-cost counterfactual rollouts. Under a unified experimental protocol across multiple public datasets and a private power dataset, PWM achieves state-of-the-art point prediction accuracy, outperforming existing mainstream time series forecasting models. Mechanism validations on synthetic load-shedding events demonstrate that PWM can effectively improve prediction performance in extreme scenarios.

1 INTRODUCTION

Electric load and price time series exhibit canonical damped oscillation characteristics, where regular periodic oscillations are frequently interrupted by short-term sharp spikes that rapidly decay back to the mean (Fig. 1(a)). Such inherent temporal patterns are not stochastic fluctuations but are governed by the dynamic game between power supply and demand. Extreme weather, unit outages and sudden load surges intensify supply–demand tension and trigger abrupt fluctuations in electricity prices and loads. With the regulation of unit redispatch, peak shaving and valley filling and demand response, system imbalance is gradually relieved and converges to temporary supply-demand equilibrium, corresponding to the amplitude-dissipating contraction process. Meanwhile, continuous external disturbances from weather and seasonal variations drive iterative cycles of imbalance and recovery, forming the amplitude-preserving phase rotation.

Mathematically, the linear generator of damped oscillators can be uniquely decomposed into a skew-symmetric rotation component and a symmetric contraction component, ensuring that all eigenvalues lie within or on the unit circle under the spectral constraint $\rho \leq 1$. For electricity time series, the rotation mechanism corresponds to periodic supply-demand cycles, while the dissipation mechanism represents system relaxation after external disturbances, which perfectly matches the bounded-fluctuation characteristics of power data. Targeting the coexistence of periodic steady-state behaviours and decaying transients in electricity data, this paper constructs a mechanism-aware inductive bias. We parameterize the transition operator with learnable and interpretable rotation and dissipation components, so that the model can better fit the intrinsic evolution laws of electricity time-series.

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.

Although electricity temporal dynamics are inherently governed by supply–demand-driven damped oscillations, existing deep forecasting models rarely incorporate such physical inductive biases. Modern patching transformers, inverted transformers, and 2D-folding convolutional neural networks (CNNs) (Zeng et al., 2023; Nie et al., 2023; Liu et al., 2024; Wu et al., 2023) possess strong representation capabilities, yet their structural priors focus solely on expressive feature learning without enforcing physically consistent dynamics. Moreover, these models operate unconditionally, making them unable to integrate external control events such as planned load shedding and unit outages (Palensky & Dietrich, 2011).

Koopman-based methods explicitly constrain latent linear dynamics and improve temporal stability (Liu et al., 2023; Bai & Ding, 2025; Mehouachi & Jabari, 2025; Zhang et al., 2025; Forootani et al., 2025), but existing parameterizations suffer from inherent limitations. Strictly unitary Koopman transitions eliminate transient divergence but discard dissipation entirely, conflicting with the mean-reverting nature of power load and price series (Wang et al., 2025). Conversely, unconstrained Koopman operators may still induce transient amplification and long-horizon instability due to non-normal dynamics, even when the spectral radius is bounded.

To address these limitations, this paper proposes PWM (power world model), a conditional Koopman world model for distribution-aware electricity forecasting. Following the Koopman line of work (Koopman, 1931; Korda & Mezić, 2018) and the universal-delay-embedding backbone (Wang et al., 2025), PWM constructs latent states via delay embedding and Transformer encoding, and evolves temporal dynamics using a condition-affine Koopman transition. The Koopman generator is structurally decomposed into interpretable dissipation and rotation components, strictly constraining the spectral radius within the unit circle. This design naturally encodes both periodic oscillation and mean-reverting dissipation, ensuring bounded and stable long-horizon prediction. Instead of concatenating conditional information at the input level, PWM embeds external control conditions directly into the transition operator, integrating event information as dynamic modulators. This mechanism enables robust event-conditioned forecasting and low-cost counterfactual rollouts. Equipped with a quantile prediction head optimized by pinball loss, PWM further achieves distribution-aware probabilistic forecasting.

The contributions of this work as follows: 1) We derive a physically grounded inductive bias for electricity forecasting by interpreting damped oscillatory power dynamics as coupled supply–demand evolution, where grid balancing induces amplitude dissipation and recurrent load shocks sustain periodic rotation. 2) We propose a dissipation–rotation decoupled Koopman parameterization that structurally guarantees bounded latent dynamics. 3) We integrate conditional control signals into transition dynamics, enabling effective event-conditioned forecasting and low-cost counterfactual simulation. 4) We validate PWM on diverse public benchmarks and a private electricity spot-market dataset, showing consistent superiority over mainstream time-series models.

2 Related Work

Modern long-term time series forecasting has witnessed rapid progress driven by linear baselines (Oreshkin et al., 2020; Challu et al., 2023; Zeng et al., 2023) and Transformer-based variants. Simple linear architectures such as DLinear achieve competitive performance by decomposing time series into trend and seasonal components (Zeng et al., 2023). PatchTST improves multivariate forecasting accuracy by segmenting individual variates into sequential patches as model tokens (Nie et al., 2023), while iTransformer reorganizes Transformer attention along feature dimensions instead of temporal dimensions to enhance cross-variate modeling (Liu et al., 2024). TimeMixer further captures multi-scale temporal dependencies via hierarchical resolution mixing (Wang et al., 2024), and TimesNet models complex temporal variations via frequency-aware 2D convolutions.

Recent Koopman-based forecasting methods introduce linear latent dynamics to enhance temporal modeling stability. Koopa integrates block segmentation with Koopman theory for stable long-term prediction (Liu et al., 2023). Subsequent studies further combine Koopman dynamics with neural ordinary differential equations (ODEs) (Bai & Ding, 2025), Kronecker-structured operators (Mehouachi & Jabari, 2025), structured linear recurrent neural networks (RNNs) (Zhang et al., 2025), and Transformer architectures (Forootani et al., 2025). Different from existing Koopman variants that rely on hard unitary constraints or post-hoc spectral regularization, our PWM embeds

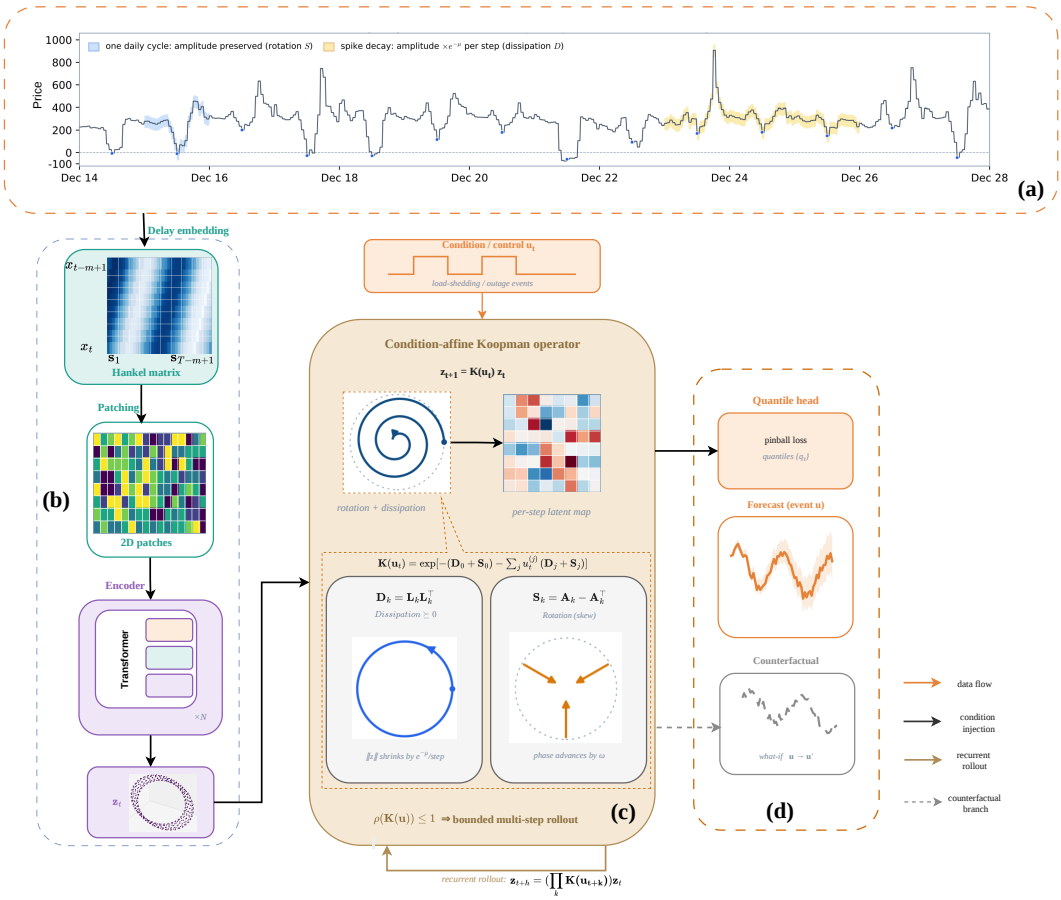

Figure 1: The PWM architecture. (a) Raw input; for clarity, only price series are shown and covariates are omitted. (b) The input is delay-embedded and mapped to a latent representation $\mathbf{z}_t$ via feature extraction. (c) Conditioning is incorporated into the transition operator, whose dissipation-plus-rotation generator guarantees $\rho(\mathbf{K}(\mathbf{u})) \leq 1$. (d) Distribution-aware forecasting.

conditional control into condition-affine transition dynamics and explicitly parameterizes the Koopman generator via decoupled dissipation and rotation components.

Beyond architectural differences, our delay-embedding Koopman paradigm fundamentally differs from TimesNet's explicit period-folding scheme. Instead of constructing period-aligned 2D state grids via fixed FFT (fast Fourier transform) priors, PWM builds latent states through universal delay embedding (Wang et al., 2025) and encodes periodicity implicitly via Koopman rotational eigenmodes. This recurrent, spectrally bounded dynamic formulation supports stable long-horizon rollouts and low-cost counterfactual simulation, and further allows external control signals to modulate temporal evolution.

3 Methodology

3.1 Problem Formulation

We consider the conditional forecasting of multivariate electricity time series. Let $\mathbf{x}_t = (x_t^{(1)}, \ldots, x_t^{(C)})^\top \in \mathbb{R}^C$ denote the observation vector at step t, whose C channels record customer loads, market prices, and exogenous covariates (e.g., weather, calendar); the series $\{\mathbf{x}_t\}_{t=1}^T$ of length T is sampled at a fixed resolution (5–60 minutes in our benchmarks). In addition to the observations, a *condition* sequence $\mathbf{u}_t \in \mathbb{R}^{C_u}$ carries C_u exogenous control or event channels—the

set-points commanded by automatic generation control (AGC), unit-status flags, published maintenance or load-shedding schedules—observed up to the forecast issue time t and known over the forecast horizon, in the market sense of a published next-day schedule. Given a lookback window $\mathbf{x}_{t-L+1:t}$ of the most recent L observations and the conditions $\mathbf{u}_{t-L+1:t+h}$, the task is to estimate the joint predictive distribution,

$$p\big(\mathbf{x}_{t+1:t+h} \,\big|\, \mathbf{x}_{t-L+1:t},\, \mathbf{u}_{t-L+1:t+h}\big) \tag{1}$$

over the next h steps ($h \in \{96, 192, 336, 720\}$ in our experiments); point forecasts and uncertainty bands are read off as quantiles of (1). The unconditional setting of standard long-term forecasting benchmarks is the special case that drops $\mathbf{u}$ (equivalently, $\mathbf{u} \equiv \mathbf{0}$). Three desirable properties of model (1) in the power-systems setting motivate our design: (i) *long-horizon stability*—the iterated h-step map must remain bounded as h grows, since unstable dynamics compound errors; (ii) *event conditioning*—predictions must respond correctly to conditions that change within the horizon; and (iii) *counterfactual simulation*—replacing $\mathbf{u}$ by an alternative schedule must be answerable without retraining.

3.2 Preliminaries: Koopman Operators and Delay Embedding

Koopman operators. Consider the discrete-time system $\mathbf{x}_{t+1} = F(\mathbf{x}_t)$ and an observable g; the Koopman operator acts by composition, $[\mathcal{K}g](\mathbf{x}) = g(F(\mathbf{x}))$ (Koopman, 1931). Though F may be strongly nonlinear, $\mathcal{K}$ is linear, so along any trajectory $g(\mathbf{x}_t) = [\mathcal{K}^t g](\mathbf{x}_0)$: the nonlinear dynamics become linear in the space of observables. The goal is to find a finite-dimensional subspace on which $\mathcal{K}$ acts (approximately) linearly and to advance the system there (Korda & Mezić, 2018). Where eigenfunctions exist, $\mathcal{K}\phi_k = \lambda_k \phi_k$ and $\phi_k(\mathbf{x}_t) = \lambda_k^t\, \phi_k(\mathbf{x}_0)$: $|\lambda_k|$ sets the mode's growth or decay rate and $\angle\lambda_k$ its oscillation frequency. This spectral structure maps directly onto power-system dynamics—lightly damped electromechanical modes have $|\lambda_k| \approx 1$ with nonzero phase, mean-reverting load processes $|\lambda_k| < 1$—and Section 3.3 exploits it by parameterizing the transition generator as dissipation plus rotation, which realizes exactly these eigenvalue locations.

Delay embedding. Koopman eigenfunctions are unknown and the observable space is infinite-dimensional, so we need a concrete, data-driven family of observables. By Takens' theorem, a generic observation function embeds the attractor in delay coordinates whenever $m \geq 2d + 1$ (d: attractor dimension), and the reconstructed dynamics is topologically conjugate to the original (Takens, 1981). For the multivariate load series with g the identity and unit lag, the delay vector is $\mathbf{s}_t = [\mathbf{x}_{t-m+1}^\top, \ldots, \mathbf{x}_t^\top]^\top \in \mathbb{R}^{mC}$. Stacking them for $t = m, \ldots, T$ yields the Hankel matrix,

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{T-m+1} \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{T-m+2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_m & \mathbf{x}_{m+1} & \cdots & \mathbf{x}_T \end{bmatrix} \in \mathbb{R}^{mC \times (T-m+1)}, \tag{2}$$

whose columns are the delay vectors (Fig. 1(b)): each column holds the complete m-step local history of all channels, and consecutive columns overlap in $m-1$ entries. Thus, the matrix encodes both the instantaneous state and its temporal neighborhood.

3.3 Model

PWM instantiates the conditional forecasting task of Section 3.1 as a world model: a learned latent state summarizes the observable history, and an explicit, condition-modulated transition evolves it. The architecture proceeds in four stages (Fig. 1(b)-(d)): (i) delay embedding (Hankel matrix) and patch tokenization of the channels (Fig. 1(b)); (ii) a Transformer encoder mapping the delay vectors to a finite-dimensional latent state $\mathbf{z}_t$ in an approximate Koopman-invariant subspace where the evolution is approximately linear; (iii) a condition-affine Koopman transition (Section 3.3) with dissipation-plus-rotation parameterization and event gating (Fig. 1(c)); (iv) a quantile head decoding predictive quantiles.

Input Encoding Continuous channels (time-series) are encoded through delay embedding with embedding length m and unit lag, forming a Hankel matrix that is split into 2D patches, projected, and position-encoded, following the UDE protocol (Wang et al., 2025). Condition channels are encoded separately: continuous controls (AGC set-points, reactive power) follow the same embedding

path but carry a control-type token. All tokens are aggregated per time step into the condition vector $\mathbf{u}_t$ used by the transition layer.

Conditional Koopman Transition The core transition is condition-affine in the latent space:

$$\mathbf{z}_{t+1} = \mathbf{K}(\mathbf{u}_t)\,\mathbf{z}_t, \qquad \mathbf{K}(\mathbf{u}_t) = \mathbf{K}_0 + \sum_j u_t^{(j)}\,\mathbf{K}_j, \tag{3}$$

where $\mathbf{K}_0$ is the zero-condition operator, and $\mathbf{K}_j$ is the contribution of condition channel j. Multi-step forecasting given a future condition sequence is the operator product $\mathbf{z}_{t+h} = \big(\prod_{k=0}^{h-1} \mathbf{K}(\mathbf{u}_{t+k})\big)\mathbf{z}_t$; under an approximately constant condition this reduces to the operator power $\mathbf{K}(\mathbf{u})^h$, giving a full-horizon prediction in one forward pass. Because conditioning lives inside the transition, a counterfactual (replacing $\mathbf{u}$ by an alternative $\mathbf{u}'$) costs a single operator product.

Dissipation-Rotation Parameterization A naive way to stabilize $\mathbf{K}_0$ is to constrain it to be unitary ($\mathbf{K}_0 = \exp(\mathbf{A})$ with $\mathbf{A}$ being the skew-Hermitian); this guarantees boundedness of every trajectory but removes dissipation entirely. That contradicts power-system physics: electromechanical modes are damped, and load processes mean-revert.

In continuous time, let $\dot{\mathbf{z}} = \mathbf{G}(\mathbf{u})\,\mathbf{z}$; any real matrix splits uniquely into a symmetric and a skew-symmetric part; setting $\mathbf{G}(\mathbf{u}) = -\mathbf{D}(\mathbf{u}) + \mathbf{S}(\mathbf{u})$ with $\mathbf{D}(\mathbf{u}) = -(\mathbf{G} + \mathbf{G}^\top)/2$ and $\mathbf{S}(\mathbf{u}) = (\mathbf{G} - \mathbf{G}^\top)/2$, we restrict to generators with $\mathbf{D}(\mathbf{u}) \succeq 0$, so that $\mathbf{D}$ acts as dissipation and $\mathbf{S}$ as rotation. Mapping to discrete time via the matrix exponential gives the **structurally stable conditional operator**,

$$\mathbf{K}(\mathbf{u}_t) = \exp\Big[-(\mathbf{D}_0 + \mathbf{S}_0) - \textstyle\sum_j u_t^{(j)}\,(\mathbf{D}_j + \mathbf{S}_j)\Big], \tag{4}$$

where the dissipation matrices $\mathbf{D}_k = \mathbf{L}_k\mathbf{L}_k^\top$ ($\mathbf{L}_k \in \mathbb{R}^{M\times r}$, low-rank) and the rotation matrices $\mathbf{S}_k = \mathbf{A}_k - \mathbf{A}_k^\top$ are positive semidefinite and skew-symmetric by construction. Both $\mathbf{L}_k$ and $\mathbf{A}_k$ learnable parameters.

The two components of the generator $\mathbf{S}(\mathbf{u})$ and $\mathbf{D}(\mathbf{u})$ play two physically distinct roles. The skew-symmetric part $\mathbf{S}(\mathbf{u})$ is a pure rotation: it reorients the latent state while leaving its norm unchanged, since $\mathbf{z}^\top\mathbf{S}(\mathbf{u})\mathbf{z} = 0$ for every $\mathbf{z}$; this is the oscillation mode (Fig. 1(a),(c)). The symmetric positive-semidefinite part $\mathbf{D}(\mathbf{u})$ is a pure contraction: for every state, $\frac{d}{dt}\|\mathbf{z}\|^2 = -2\mathbf{z}^\top\mathbf{D}(\mathbf{u})\mathbf{z} \le 0$, so $\mathbf{D}$ dissipates energy and pulls the state back toward equilibrium, a damping that corresponds to friction and to the mean reversion of the power market (Fig. 1(a),(c)). Equivalently, in discrete time each mode contributes an eigenvalue $\lambda = e^{-\mu+\mathrm{j}\omega}$ whose magnitude $|\lambda| = e^{-\mu}$ is set by $\mathbf{D}$ (the per-step amplitude shrinkage, $\mu \ge 0$) and whose phase $\angle\lambda = \omega$ is set by $\mathbf{S}$ (the per-step phase advance) (Fig. 1(b) and Fig. 2).

Mode Library and Event Gating The PWM can learn a small library of base operators rather than a single linear operator, each an independent instance of the structurally stable form of Eq. (4): operator r carries its own dissipation and rotation matrices $\mathbf{D}_{r,k} = \mathbf{L}_{r,k}\mathbf{L}_{r,k}^\top \succeq 0$ and $\mathbf{S}_{r,k} = \mathbf{A}_{r,k} - \mathbf{A}_{r,k}^\top$, with $k = 0$ the zero-condition part and $k = j$ the contribution of condition channel j. An event gating network $g_{\mathrm{gate}}(\mathbf{z}_t, \mathbf{u}_t)$ produces mode weights $\alpha_r(t) \ge 0$ with $\sum_r \alpha_r(t) = 1$, and the transition operator is their convex mixture:

$$\mathbf{K}(\mathbf{u}_t) = \sum_{r=1}^{R} \alpha_r(t)\,\mathbf{K}_r(\mathbf{u}_t), \tag{5}$$

where $\mathbf{K}_r(\mathbf{u}_t) = \exp[-\big(\mathbf{D}_{r,0} + \mathbf{S}_{r,0}\big) - \sum_j u_t^{(j)}\,\big(\mathbf{D}_{r,j} + \mathbf{S}_{r,j}\big)]$. Eq. (4) is thus the single-regime case R=1. In practice the gate is implemented as a single hidden-layer MLP with $\tanh$ activation followed by a softmax over the R modes, $\alpha_r(t) = \exp(l_r(t))/\sum_{r'}\exp(l_{r'}(t))$, with logits $\mathbf{l}(t) = \mathbf{W}_2\tanh\big(\mathbf{W}_1[\mathbf{z}_t;\mathbf{u}_t] + \mathbf{b}_1\big) + \mathbf{b}_2$, where $[\cdot]$ denotes concatenation.

Proposition 1 (Spectral bound)**.** *If $\sum_j u_t^{(j)}\mathbf{D}_j \succeq 0$ (nonnegative condition weights or nonnegative dissipation weights), then for every condition sequence the spectral radius satisfies $\rho(\mathbf{K}(\mathbf{u}_t)) = \max_i|\lambda_i| \le 1$, where $\lambda_i = \exp(-\mu_i + \mathrm{j}\omega_i)$ with $\mu_i \ge 0$ the logarithmic decay rate of mode i*

and ω_i its angular frequency. Hence the unforced latent norm is non-increasing and long-horizon rollouts are bounded; damped modes decay as $\exp(-\mu_i h)$, while neutral modes oscillate without growing.

The bound extends from the single operator of Eq. 4 to the mode library of Eq. 5 (Section 3.3). Every library member $\mathbf{K}_r(\mathbf{u}_t)$ carries the same dissipation-plus-rotation generator structure, so under the same condition on the condition weights—applied to every member—the symmetric part of each generator is negative semidefinite, the Euclidean norm is non-increasing along the flow $\dot{\mathbf{z}} = \mathbf{G}\mathbf{z}$, and $\|\mathbf{K}_r(\mathbf{u}_t)\|_2 \le 1$. Since the gate weights are nonnegative and sum to one, the triangle inequality gives $\|\mathbf{K}(\mathbf{u}_t)\|_2 \le \sum_r \alpha_r(t)\,\|\mathbf{K}_r(\mathbf{u}_t)\|_2 \le 1$, whence $\rho(\mathbf{K}(\mathbf{u}_t)) \le \|\mathbf{K}(\mathbf{u}_t)\|_2 \le 1$. The extension is necessarily posed at the norm level: the spectral radius is not convex in the matrix, so a convex combination of matrices with $\rho \le 1$ can itself have $\rho > 1$—which is why Eq. 5 mixes structured generators rather than free operators. The per-eigenvalue reading $\lambda_i = e^{-\mu_i + \mathrm{j}\omega_i}$ holds within each member; the mixture guarantees only the aggregate bound.

During training, each batch undergoes synthetic event injection: with probability p_{ev} we sample a duration $d_{\mathrm{ev}} \sim U(8, 24)$ steps and a channel subset $\mathcal{S} \subset \{1, \dots, C\}$ of size $C/4$, scale the tail of the input window and the head of the target by $1 - \eta =$ on those channels, and set the condition value to $u_b = \eta$ for the whole horizon ($u_b = 0$ otherwise). The model then removes the series level by subtracting the last observed value $\mathbf{x}_{\mathrm{last}}$, builds per-channel delay-embedding tokens $\mathbf{z}_c^{(0)} = \mathrm{LN}(\mathbf{W}_{\mathrm{in}}\bar{\mathbf{X}}_{:,c})$, and encodes them with the Transformer into a latent $\mathbf{Z}^{(1)}$. The latent is advanced by k_{ref} applications of the condition-affine operator $\mathbf{K}(u_b) = \exp[-(\mathbf{D}_0 + \mathbf{S}_0) - u_b(\mathbf{D}_1 + \mathbf{S}_1)]$ with $\mathbf{D}_k = \mathbf{L}_k\mathbf{L}_k^\top$ and $\mathbf{S}_k = \mathbf{A}_k - \mathbf{A}_k^\top$; for each horizon, a quantile head decodes the increments $\mathbf{\Delta}_{k,q}$ from the refined latent and the residual embedding $\mathbf{r} = \mathrm{GELU}(\mathbf{W}_r\mathbf{x}_{\mathrm{last}})$, and re-levels the forecasts as $\hat{\mathbf{x}}_{t+k}^{(q)} = \mathbf{x}_{\mathrm{last}} + \mathbf{\Delta}_{k,q}$. The training loss is the sum of pinball losses over all quantiles plus twice the MSE of the median forecast,

$$\mathcal{L} = \sum_{q \in \mathcal{Q}} \mathrm{pinball}_q(\hat{\mathbf{x}}^{(q)}, \mathbf{Y}_b) + \gamma\,\mathrm{MSE}(\hat{\mathbf{x}}^{(0.5)}, \mathbf{Y}_b). \tag{6}$$

4 Experiments

Datasets Our experiments use six time-series datasets in total: four electricity benchmarks — UCI Electricity (ECL) (Trindade, 2015), ETTh2, ETTm2 (Zhou et al., 2021), and a private electricity spot-market dataset, PrivEM — and two non-electricity benchmarks, Weather and Traffic. PrivEM defines two forecasting tasks: day-ahead price prediction (Task DA) and real-time price prediction (Task RT). All datasets follow the standard 70/10/20 split, with detailed descriptions provided in the Appendix. Since the datasets span different temporal granularities (from 5 minutes to 1 hour), we report prediction lengths in steps, $h \in \{96, 192, 336, 720\}$.

Baselines and Metrics DLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), TimesNet (Wu et al., 2023), iTransformer (Liu et al., 2024), TimeMixer (Wang et al., 2024), and Chronos-2 (Ansari et al., 2025), a 120M-parameter pretrained time-series language model, fine tuned with LoRA. MSE/MAE of the median prediction (0.5 quantile) in the original scale; CRPS (Continuous Ranked Probability Score) approximated from quantiles $\{0.1, 0.5, 0.9\}$ for distributional evaluation.

4.1 Headline Comparisons on Benchmarks

Public datasets Table 1 aggregates the five public benchmarks—ECL, ETTh2, ETTm2, Weather, and Traffic—at different horizons. All the results follow the TSLib (Wang et al., 2026) protocol. Table 1 reports MSE/MAE at $h \in \{96, 192, 336, 720\}$ on the five public benchmarks. PWM attains the best MSE at all four horizons on ECL and ETTm2, at $h \le 336$ on ETTh2 and Weather, and is second to iTransformer on Traffic at every horizon; its advantage is markedly larger on the electricity benchmarks than on others. Among the electricity datasets, PWM improves MSE over the strongest baseline by 10.7%, 4.5%, 8.2%, and 10.5% on ECL (vs. iTransformer, 0.141–0.2), by 11.4%–21.9% on ETTm2 with the margin widening from 11.4% at $h = 96$ to 21.9% at $h = 720$ (vs. PatchTST, 0.155–0.314), and by 12.5%–19.0% on ETTh2 at $h \le 336$ (over Chronos-2 at $h \le 192$ and iTransformer at $h = 336$), where it is second only at $h = 720$ (0.444 vs. 0.427). On the non-electricity

Table 1: MSE/MAE at horizons 96/192/336/720 on the five public benchmarks. Bold: best MSE or MAE in the column within each dataset block.

Dataset	Model	96	192	336	720
ECL (321 ch)	DLinear	0.218/0.293	0.208/0.292	0.232/0.314	0.288/0.361
	PatchTST	0.186/0.268	0.188/0.270	0.210/0.288	0.243/0.315
	TimesNet	0.198/0.290	0.264/0.330	0.281/0.342	0.337/0.372
	iTransformer	0.158/0.242	0.159/0.248	0.185/0.267	0.223/0.297
	Chronos-2	0.227/0.266	0.225/0.273	0.260/0.300	0.327/0.347
	PWM (ours)	**0.141/0.229**	**0.152/0.241**	**0.170/0.257**	**0.200/0.280**
ETTh2 (7 ch)	DLinear	0.333/0.387	0.477/0.476	0.594/0.541	0.831/0.657
	PatchTST	0.302/0.348	0.388/0.400	0.426/0.433	0.431/0.446
	TimesNet	0.340/0.374	0.402/0.414	0.452/0.452	0.462/0.468
	iTransformer	0.297/0.349	0.380/0.400	0.428/0.432	**0.427/0.445**
	Chronos-2	0.279/0.334	0.373/0.389	0.444/0.432	0.611/0.514
	PWM (ours)	**0.244/0.326**	**0.302/0.365**	**0.348/0.396**	0.444/0.455
ETTm2 (7 ch)	DLinear	0.193/0.292	0.284/0.362	0.369/0.427	0.554/0.522
	PatchTST	0.175/0.259	0.241/0.302	0.305/0.343	0.402/0.400
	TimesNet	0.187/0.267	0.249/0.309	0.321/0.351	0.408/0.403
	iTransformer	0.180/0.264	0.250/0.309	0.311/0.348	0.412/0.407
	Chronos-2	0.223/0.302	0.289/0.345	0.335/0.372	0.451/0.427
	PWM (ours)	**0.155/0.254**	**0.199/0.288**	**0.243/0.320**	**0.314/0.366**
Weather (21 ch)	DLinear	0.196/0.256	0.239/0.299	0.281/0.331	**0.345**/0.382
	PatchTST	0.175/0.218	0.224/0.258	0.279/0.299	0.352/0.345
	TimesNet	0.171/0.221	0.225/0.264	0.281/0.303	0.356/0.352
	iTransformer	0.175/0.216	0.224/0.257	0.282/0.299	0.361/0.352
	Chronos-2	0.290/0.240	0.314/0.282	0.364/0.322	0.462/0.380
	PWM (ours)	**0.167/0.209**	**0.215/0.253**	**0.271/0.293**	0.348/ **0.343**
Traffic (862 ch)	DLinear	0.697/0.429	0.647/0.407	0.653/0.410	0.694/0.429
	PatchTST	0.458/0.298	0.469/0.301	0.483/0.307	0.517/0.326
	TimesNet	0.604/0.317	0.626/0.324	0.638/0.339	0.674/0.345
	iTransformer	**0.393/0.268**	**0.416/0.278**	**0.429**/0.284	**0.462**/0.301
	Chronos-2	0.567/0.300	0.547/0.294	0.560/0.303	0.605/0.330
	PWM (ours)	0.404/0.272	0.427/0.282	0.433/**0.283**	0.463/**0.297**

benchmarks the gains are smaller: on Weather, PWM wins $h \leq 336$ by 2.3%–4.0% and trails DLinear by only 0.9% at $h = 720$ (0.348 vs. 0.345); on Traffic, iTransformer is best at all horizons and PWM is second, within 2.8% (0.404–0.463), yet PWM attains the best MAE at $h \in \{336, 720\}$. In terms of MAE, PWM is best at all horizons on ECL, ETTm2, and Weather, at $h \leq 336$ on ETTh2, and at $h \in \{336, 720\}$ on Traffic.

Price Forecasting on PrivEM The tasks on PrivEM are the deployment-relevant evaluation: forecasting actual clearing prices one to 7.5 days (15-minute resolution) ahead under market-publication information constraints. All models receive identical inputs respecting the constraints—for Task DA (day-ahead settlement price) the published day-ahead prices and all other market variables except the same-day actual real-time price (replaced by its previous-day value); for Task RT (real-time settlement price) all same-day variables except the real-time price itself (likewise lagged).

Table 2: MSE/MAE on the PrivEM (z-score space, lookback 96, calendar split, test = 2026-05-01 to 09-05; 11.6–12.2k test windows per horizon). Boldface: the best MSE/MAE in the column.

Task	Model	96	192	336	720
Task DA (day-ahead)	Chronos-2	1.101/0.723	1.319/0.778	1.634/0.822	2.558/0.920
	DLinear	0.555/0.504	0.754/0.605	0.940/0.686	1.197/0.796
	PatchTST	0.503/0.474	0.647/0.557	0.808/0.624	1.056/0.721
	TimesNet	0.532/0.514	0.742/0.613	0.993/0.711	1.463/0.866
	iTransformer	0.441/0.450	0.550/0.514	0.681/0.573	**0.868/0.649**
	PWM (ours)	**0.405/0.433**	**0.510/0.498**	**0.630/0.554**	0.872/0.654
Task RT (real-time)	Chronos-2	2.193/0.990	2.298/1.011	2.523/1.046	3.183/1.079
	DLinear	1.536/0.805	1.598/0.820	1.627/0.826	1.729/0.861
	PatchTST	1.597/0.858	1.658/0.867	1.732/0.886	1.668/0.867
	TimesNet	1.494/0.810	1.982/0.939	2.097/0.964	1.776/0.907
	iTransformer	1.572/0.847	1.667/0.880	1.682/0.879	1.635/0.869
	PWM (ours)	**1.234/0.731**	**1.378/0.779**	**1.473/0.810**	**1.545/0.837**

Table 2 reports MSE/MAE on PrivEM. PWM attains the best MSE on seven of the eight horizon–task combinations and the best MAE in the same seven settings; the only exception is Task DA at $h = 720$, where iTransformer is marginally ahead (0.868 vs. 0.872 MSE; 0.649 vs. 0.654 MAE). On Task DA, PWM improves MSE over the strongest baseline (iTransformer) by 8.2%, 7.3%, and 7.5% at $h = 96$, 192, and 336 (0.405 vs. 0.441, 0.510 vs. 0.550, 0.630 vs. 0.681). On Task RT, where all models obtain substantially higher errors than on Task DA, PWM leads by larger margins at every horizon: MSE is 17.4%, 13.8%, 9.5%, and 5.5% below the best baseline (TimesNet at $h = 96$, DLinear at $h \in \{192, 336\}$, iTransformer at $h = 720$), and MAE is best throughout (e.g., 0.731 vs. 0.805 at $h = 96$). TimesNet is the least stable on Task RT, with non-monotonic MSE across horizons. PWM degrades most gracefully, and its advantage is consistent on the harder real-time task and at short-to-mid horizons on the day-ahead task.

4.2 Conditional Forecasting: Synthetic Load-Shedding Events

To test the central claim—conditioning through the transition—we use a *semi-synthetic* protocol: the load series are the real ECL data; only the shedding events are synthetic, injected under a fixed, seed-controlled rule (Section 3.3). The synthesis is a necessity rather than a simplification: evaluating conditional forecasts requires events whose timing, depth, and channel coverage are known exactly, and real load traces carry no such ground truth. Each protocol parameter has an operational reading. *Frequency*: an event starts with probability 0.003 per hourly step—an expected inter-event gap of ≈333 hours, i.e. roughly one event every two weeks. *Depth*: while an event is active, the load of each affected channel is scaled by $(1 - \eta)$ with $\eta = 0.15$, a 15% suppression. *Coverage and duration*: each event hits a uniformly random quarter of the 321 channels (≈80 customers) and lasts 8–24 hours. *Information structure*: events are drawn from a random seed independent of the load process (exogenous), and the event indicator u is observed at forecast issue time and held constant over the horizon—semantically, a published next-day maintenance schedule rather than an unanticipated shock. The indicator u forms the condition channel. The conditional model is trained with the same injection protocol applied consistently to both inputs and forecast targets (we found this train–test protocol consistency to be critical: when events were injected into inputs but not into targets, conditioning produced negative gains). Two caveats bound the strength of the claim: (i) the protocol is identical in training and test, so the reported gain measures in-distribution conditioning ability, not generalization to unseen event magnitudes or patterns; and (ii) the comparison is against the model's own unconditional counterpart, not against input-concatenation conditioning.

Table 3: Conditional vs. unconditional per-channel z-score space of PWM under synthetic shedding events (on ECL with h=96).

Metric	Unconditional	Conditional	Gain
Event-window MSE (median)	2.196	2.089	+4.9%
Event-window CRPS (quantile)	1.094	0.767	+29.9%
Full-test MSE (median)	1.931	1.878	+2.8%

Table 3 reports the event-window comparison of the conditional PWM vs. its unconditional counterpart over 1,843 event windows on the test segment. Conditioning through the transition yields measurable gains concentrated in event windows, and the distributional gain (CRPS, -29.9%) is far larger than the point gain (MSE, -4.9%): the conditional model shifts its predictive distribution where the event changes the dynamics, while the median path moves less. This matches the design intent—the quantile head responds more sensitively to the condition signal than the point estimate.

4.3 K_j structure and a null conditional gain

The conditionally trained PWM on ETTh2 (h=96; events on 2 of 7 channels, η=0.15, 8–24 h, injected into both inputs and targets; same base schedule) learns a structured generator pair. Eigendecomposition of the 128-d $\mathbf{K}(0)$ (Fig. 2(a)–(c)) shows 49 of 128 modes (38%) with $|\lambda| > 0.9$ and none above 0.963; modes form complex-conjugate pairs, i.e., sustained yet damped oscillations—exactly the dissipation-plus-rotation signature ($\mathbf{D}_0 \succeq 0$ sets the decay, skew $\mathbf{S}_0$ the phase). The event operator $\mathbf{K}(1)$ shifts per-mode magnitudes down by 0.021 on average (largest drift 0.071; Fig. 2(d)–(f)): conditioning mildly increases dissipation and re-weights rotation while preserving the spectral layout—a mechanism-level response to the event signal. This structure, however, does

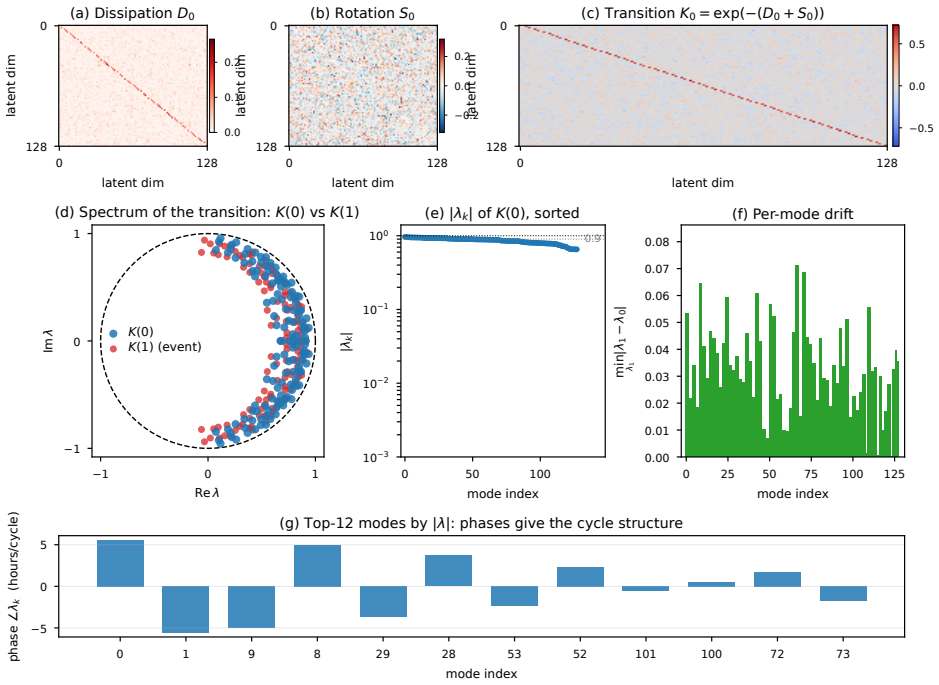

Figure 2: Learned generator structure on ETTh2 ($K(u) = \exp[-(D_0 + S_0) - u(D_1 + S_1)]$). (a) dissipation $D_0 = L_0 L_0^\top$; (b) rotation $S_0 = A_0 - A_0^\top$; (c) transition K_0 heatmap; (d) spectrum of $K(0)$ vs. $K(1)$ (unit circle); (e) eigenvalue magnitudes sorted (damped oscillatory modes); (f) per-mode drift under the event; (g) phases of the top-12 modes. 49/128 modes have $|\lambda| > 0.9$, all ≤ 0.963; $K(1)$ lowers mean $|\lambda|$ by 0.021.

not translate into end-to-end event-window accuracy on ETTh2: over 2,814 test event windows the gain is null (MSE 0.238 with $u = \eta$ vs. 0.237 with $u = 0$, -0.13%), echoing the ECL P0 finding that conditioning gains are implementation-sensitive and not robust at mechanism scale. We report the operator-level structure as evidence that the condition mechanism is *learnable*, and the null end-to-end gain as an honest bound on its current utility.

5 Conclusion

This paper began from a mechanism rather than an architecture. In electricity markets, the intensification of the supply–demand contradiction, a widening gap and violent price fluctuation, is formalized as a physical prior: amplitude-preserving rotation for intensification–relaxation cycles and amplitude-dissipating contraction for mean-reverting relaxation. PWM carries this prior into Koopman theory: the generator of the condition-affine operator is parameterized as dissipation plus rotation, yielding a fully physically-parameterized Koopman linear system. Experimentally, PWM attains the best point performance among supervised baselines on most public benchmarks and on both tasks of the private spot-market deployment (PrivEM); the conditional capability is validated by event injection, and the constructed $\mathbf{K}$ bounds and damps multi-step rollouts ($\sigma_{\max} = 1.0$). We are equally explicit about the limitations: the conditional mechanism is training-unstable at mechanism scale, and the injected events are in-distribution by construction. Production-scale conditional runs, and real market-event validation are the immediate next steps. The broader lesson stands regardless of these limitations: when a domain offers a physical account of its own dynamics, embedding that account as structure inside a linear-operator formalism is a viable path to forecasts that are accurate, bounded by construction, and auditable by construction.

ETHICS STATEMENT

This work involves no human subjects, and no institutional review board approval was required. The public benchmarks (ECL, ETTh2, ETTm2, Weather, Traffic) are standard, publicly released datasets used under their respective licenses. The private electricity spot-market dataset (PrivEM) was provided under a data-use agreement for academic research; it contains aggregate market-level records rather than personal or otherwise sensitive information and is not redistributed. The proposed method is a forecasting model for research purposes; price forecasts should not be construed as trading advice, and we foresee no risk of harmful misuse beyond that of standard time-series forecasting models. The authors have no conflicts of interest or competing financial interests to disclose. All experiments are reported with the data splits and evaluation protocols described in this paper to support reproducibility.

AI USE STATEMENT

Generative AI tools were used for the following tasks, for which disclosure is recommended: organizing the literature and references, drafting parts of the manuscript, and editing the paper for language and readability. We did not use generative AI for any task in the required-disclosure category (e.g., proposing hypotheses, designing methodology, implementing methods, formulating or proving mathematical claims), and these tasks are not applicable to this work. All AI-assisted text was reviewed and revised by the authors, and the reported results and claims were verified item by item against the experimental records. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we submit an anonymous code bundle as supplementary material, containing the PWM implementation, the evaluation scripts implementing the unified sampling protocol; every number reported in this paper is reproducible from this bundle. The complete model specification—forward computation (Section 3), the dissipation-plus-rotation parameterization and its spectral guarantee (Section 3.3), and the training procedure including the synthetic event-injection protocol (Algorithm 1)—together with all hyperparameters, hardware, and random seeds, is documented in the appendix and is not restated here. Supervised baselines use the official Time-Series-Library implementations and checkpoints, and are evaluated under the unified protocol via the scripts included in the bundle. Data-processing steps for the public benchmarks (hourly aggregation, normalization, and splits) are described in the Datasets and Preprocessing section; for the private spot-market dataset (PrivEM), a de-identified version is included in the supplementary material, and the full alignment, gap-removal, and calendar-split pipeline is provided in the appendix. The factor-isolation screen (Appendix B.1) toggles one factor at a time against a shared baseline, with per-configuration logs archived in the experiment bundle.

REFERENCES

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting, 2025. URL https://arxiv.org/abs/2510.15821.

Hanru Bai and Weiyang Ding. KoNODE: Koopman-driven neural ordinary differential equations with evolving parameters for time series analysis. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 2484–2519. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bai25b.html.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. NHITS: Neural hierarchical interpolation for time series forecast-

ing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Ali Forootani et al. DeepKoopFormer: Koopman spectral transfer for time series forecasting. *arXiv preprint*, 2025. URL https://arxiv.org/pdf/2508.02616. arXiv:2508.02616.

Bernard O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17(5):315–318, 1931.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Advances in Neural Information Processing Systems*, 2023.

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*, 2024.

Fares Mehouachi and Saif Eddin Jabari. Flowmixer: A depth-agnostic neural architecture for interpretable spatiotemporal forecasting. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 88811–88861. Curran Associates, Inc., 2025. doi: 10.52202/085713-2970.

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.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1ecqn4YwB.

Peter Palensky and Dietmar Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transactions on Industrial Informatics*, 7(3):381–388, 2011.

Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, volume 898 of *Lecture Notes in Mathematics*, pp. 366–381. Springer, Berlin, Germany, 1981.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

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*, 2024.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(10):11642–11662, 2026. doi: 10.1109/TPAMI.2026.3690845.

Zijian Wang, Peng Tao, Jifan Shi, Rui Bao, Rui Liu, and Luonan Chen. A time-series foundation model by universal delay embedding, 2025. URL https://arxiv.org/abs/2509.12080.

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.

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*, pp. 11121–11128, 2023.

Yitian Zhang, Liheng Ma, Antonios Valkanas, Boris N. Oreshkin, and Mark Coates. Skolr: Structured koopman operator linear rnn for time-series forecasting, 2025. URL https://arxiv.org/abs/2506.14113.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pp. 11106–11115. AAAI Press, 2021.

A APPENDIX

A.1 DATASETS AND PREPROCESSING

Data. We use the UCI Electricity dataset (321 load channels, hourly resolution, 26,304 steps over 2012–2014, each channel max-normalized to $[0, 1]$) under the standard 70/10/20 split. To test breadth beyond ECL and ETT, we add two further public multivariate benchmarks: Weather (21 meteorological variates, 10-min resolution, 52,696 steps) and Traffic (862 road-occupancy channels, hourly, 17,544 steps), both under the same split and z-score normalized with train-segment statistics. Finally, we evaluate on **PrivEM**, a private electricity spot-market dataset of 53 channels of 15-min market data (load, generation and its forecast, node and settlement prices, and two publication-lag features) covering 2025-01-01 to 2026-09-05: 52,512 steps after 15-min grid alignment and removal of incomplete segments, split by calendar time (train to 2026-02; validation 2026-03–04; test 2026-05 to 09-05) and retaining 360/59/128 complete days after gap removal.

Tasks. Two forecasting tasks on PrivEM mirror the market publication lags. *Task DA* predicts the actual day-ahead clearing price for the next $h \in \{96, 192, 336, 720\}$ time slots, using the day's published day-ahead forecast price and all other market variables but not the same-day actual real-time price (not yet published), whose lag-96 previous-day value is used instead. *Task RT* predicts the actual real-time clearing price over the same horizons, with all same-day variables available except the actual real-time price itself, whose previous-day value likewise enters.

Unified evaluation protocol. Long-horizon evaluation on multivariate data is sensitive to how test windows are sampled and how normalization is inverted: the reference library's default sliding-window evaluation produced MSE values 100× larger than a persistence ("predict last observation") baseline (0.385–0.62 vs. 0.004), because per-window re-normalization interacts with low-variance channels in z-score space. We therefore adopt a unified sampling protocol — starting at the official test boundary, one forecast origin every 96 steps (48–54 origins per horizon, depending on the model's native prediction length), with each model's native checkpoint and predictions de-normalized to the original $[0, 1]$ scale — so the reported numbers are directly comparable across models; the persistence baseline is reported for reference.

A.2 PWM IMPLEMENTATION

Forward computation. Given a lookback window $\mathbf{X} \in \mathbb{R}^{L\times C}$ ($L = 96$), the model computes forecasts in six steps:

1. *Level removal (RevIN-style residual).* $\mathbf{x}_{\text{last}} = \mathbf{X}[L-1, :] \in \mathbb{R}^C$; $\bar{\mathbf{X}} = \mathbf{X} - \mathbf{1}_L \mathbf{x}_{\text{last}}^\top$.
2. *Delay-embedding tokenization.* Each channel's L-step history is projected by a shared linear layer with layer norm: $\mathbf{z}_c^{(0)} = \text{LN}(\mathbf{W}_{\text{in}} \bar{\mathbf{X}}_{:,c})$, $\mathbf{W}_{\text{in}} \in \mathbb{R}^{d\times L}$, $d = 128$, giving $\mathbf{Z}^{(0)} \in \mathbb{R}^{C\times d}$.
3. *Cross-channel Transformer encoder.* $\mathbf{Z}^{(1)} = \text{TransformerEncoder}(\mathbf{Z}^{(0)})$ over the C channel tokens with $n_{\text{lyr}} = 2$ layers, $n_{\text{head}} = 4$ heads, feed-forward width $2d = 256$, dropout 0.1.
4. *Conditional operator.* $\mathbf{D}_0 = \mathbf{L}_0\mathbf{L}_0^\top$ ($\mathbf{L}_0 \in \mathbb{R}^{d\times d/2}$), $\mathbf{S}_0 = \mathbf{A}_0 - \mathbf{A}_0^\top$; the single condition channel uses $\mathbf{D}_1 = \mathbf{L}_1\mathbf{L}_1^\top$, $\mathbf{S}_1 = \mathbf{A}_1 - \mathbf{A}_1^\top$, giving $\mathbf{G}(u) = -(\mathbf{D}_0 + \mathbf{S}_0) - u(\mathbf{D}_1 + \mathbf{S}_1)$ and $\mathbf{K}(u) = \exp[\mathbf{G}(u)]$.
5. *Latent refinement by operator power.* $\tilde{\mathbf{Z}} = \mathbf{K}(u)^{k_{\text{ref}}}\, \mathbf{Z}^{(1)}$ with $k_{\text{ref}} = 8$; the same operator is applied identically to every channel.
6. *Residual decoding.* A level feature $\mathbf{r} = \text{GELU}(\mathbf{W}_r \mathbf{x}_{\text{last}})$ ($\mathbf{W}_r \in \mathbb{R}^{d_r\times C}$, $d_r = 64$) is concatenated with each channel state; a per-horizon head maps $[\tilde{\mathbf{Z}}^{(c)}; \mathbf{r}]$ through a hidden width of 256 to $h \cdot |\mathcal{Q}|$ quantile increments, re-leveled as $\hat{\mathbf{x}}_{t+k}^{(q)} = \mathbf{x}_{\text{last}} + \boldsymbol{\Delta}_{k,q}^{(c)}$, $q \in \mathcal{Q} = \{0.1, 0.5, 0.9\}$.

The training procedure—synthetic event injection, the forward pass above, and the quantile loss $\mathcal{L} = \sum_{q\in\mathcal{Q}} \text{pinball}_q(\hat{\mathbf{x}}^{(q)}, \mathbf{Y}) + \gamma\, \text{MSE}(\hat{\mathbf{x}}^{(0.5)}, \mathbf{Y})$—is summarized in Algorithm 1.

Algorithm 1 PWM training with the synthetic event-injection protocol.

Input: training windows $\{(\mathbf{X}_t, \mathbf{Y}_t)\}$ with $\mathbf{X}_t := \mathbf{x}_{t-L+1:t}$, $\mathbf{Y}_t := \mathbf{x}_{t+1:t+h}$; quantile set $\mathcal{Q}$, parameters $\theta = \{\mathbf{W}_{\text{in}}, \text{enc}, \mathbf{L}_0, \mathbf{A}_0, \mathbf{L}_1, \mathbf{A}_1, \mathbf{W}_r\}$; refinement powers k_{ref}, rate p_{ev}=0.02, depth η=0.15, duration $d_{\text{ev}} \sim U(8, 24)$, channel count $|\mathcal{S}|$=$C/4$

Output: trained parameters θ

for each batch $b \in \mathcal{B}$ **do** ▷ synthetic event injection
  $u_b \leftarrow 0$
  if $\text{rand}() < p_{\text{ev}}$ **then**
    draw duration $d_{\text{ev}} \sim U(8, 24)$ and channel subset $\mathcal{S} \subset \{1, \ldots, C\}$, $|\mathcal{S}| = C/4$
    $\mathbf{X}_b[L - d_{\text{ev}}{:}, \mathcal{S}] \leftarrow (1 - \eta)\, \mathbf{X}_b[L - d_{\text{ev}}{:}, \mathcal{S}]$ ▷ suppression in the input tail
    $\mathbf{Y}_b[{:}d_{\text{ev}}, \mathcal{S}] \leftarrow (1 - \eta)\, \mathbf{Y}_b[{:}d_{\text{ev}}, \mathcal{S}]$ ▷ suppression in the target head
    $u_b \leftarrow \eta$ ▷ condition value, held constant over the horizon
  end if
end for
$\mathbf{x}_{\text{last}} \leftarrow \mathbf{X}_b[L-1, :]$; $\bar{\mathbf{X}} \leftarrow \mathbf{X}_b - \mathbf{1}_L \mathbf{x}_{\text{last}}^\top$ ▷ level removal
$\mathbf{z}_c^{(0)} \leftarrow \text{LN}(\mathbf{W}_{\text{in}} \bar{\mathbf{X}}_{:,c})$ for each channel c ▷ delay-embedding tokens
$\mathbf{Z}^{(1)} \leftarrow \text{enc}(\mathbf{Z}^{(0)})$; $\mathbf{D}_k \leftarrow \mathbf{L}_k \mathbf{L}_k^\top$, $\mathbf{S}_k \leftarrow \mathbf{A}_k - \mathbf{A}_k^\top$, $k \in \{0, 1\}$
$\mathbf{K}(u_b) \leftarrow \exp[-(\mathbf{D}_0 + \mathbf{S}_0) - u_b(\mathbf{D}_1 + \mathbf{S}_1)]$
$\tilde{\mathbf{Z}} \leftarrow \mathbf{K}(u_b)^{k_{\text{ref}}}\, \mathbf{Z}^{(1)}$; $\mathbf{r} \leftarrow \text{GELU}(\mathbf{W}_r\, \mathbf{x}_{\text{last}})$
for $h \in \mathcal{H}$ **do**
  $\mathbf{\Delta}_{k,q} \leftarrow \text{head}_h([\tilde{\mathbf{Z}}; \mathbf{r}])$ ▷ direct multi-horizon decoding
  $\hat{\mathbf{x}}_{t+k}^{(q)} \leftarrow \mathbf{x}_{\text{last}} + \mathbf{\Delta}_{k,q}$, $k = 1, \ldots, h$, $q \in \mathcal{Q}$ ▷ re-leveling
end for
$\mathcal{L} \leftarrow \sum_{q \in \mathcal{Q}} \text{pinball}_q(\hat{\mathbf{x}}^{(q)}, \mathbf{Y}_b) + \gamma\, \text{MSE}(\hat{\mathbf{x}}^{(0.5)}, \mathbf{Y}_b)$ ▷ loss function
$\theta \leftarrow \text{AdamW}(\nabla_\theta \mathcal{L})$

Configuration. The encoder Transformer uses the architecture of step 3 (2 layers, 4 heads, $d = 128$), with dataset-specific variants: on ETT, cross-channel attention is ablated and the per-channel projection is a two-layer MLP, the configuration selected by the screen of Section B.3; on Weather and the PrivEM tasks, a learnable variate embedding is added to the channel tokens before attention; on Traffic, the latent is widened to $d = 512$ with 8 heads and 12,000 windows sampled per epoch at batch size 8. The latent transition is a single shared condition-affine operator (steps 4–5), i.e. the *mechanism-scale* instantiation of Section 3.3, with one condition channel and no mode library. Model size is 1.56M parameters at C=7 (1.61M for the ETT encoder variant). Training follows Algorithm 1 with AdamW (lr 5×10^{-4}, weight decay 10^{-4}, batch 32), cosine decay, and 3,000 windows sampled per horizon per epoch over 30 epochs (15 in the ECL mechanism study); the horizons $\{96, 192, 336, 720\}$ are trained jointly through shared encoder/transition weights and per-horizon heads, except the conditional ETTh2 variant, which trains only h=96. γ was selected as 2 for PWM in all datasets.

Note on k_{ref}. The operator exponent is a latent *refinement count*, not a physical time index—the quantile head performs direct multi-horizon decoding, so the power k_{ref}=8 is decoupled from the prediction length h (one model serves all four horizons). We chose k_{ref}=8 as a middle ground: too few powers ($\lesssim 2$) leave the transition nearly inert and conditioning ineffective, while too many over-contract the latent because the dissipative spectrum lies inside the unit circle; this also explains the latent-linearity negative result of Section A.3, where $\mathbf{K}$ acts as a refinement map rather than a one-step propagator of the encoder trajectory (the two roles would coincide only under explicit latent-trajectory supervision).

A.3 Delay-embedding State Space

The raw delay-embedding coordinates already exhibit the geometry that Section 3 parameterizes. Figure 3 plots, per dataset, a standardized 4-week window in the coordinates $(x_t, x_{t-\ell}, x_{t-\ell-6\text{h}})$ colored by hour of day, where the lag ℓ is chosen near the quarter phase of the daily cycle (correlation of x_t with $x_{t-\ell}$ near zero, opening the loop in projection) and the variate is the one with the cleanest sub-weekly cycle (raw LULL for ETTh2/ETTm2 at ℓ=15/31h). The raw states scatter in a thin tube, and the hour-of-day mean orbit (24 closed points, solid loop; 00:00 marked) winds once per 24 h in both the 3D view (top row, rotated near top-down) and the 2D projection (bottom row): the daily

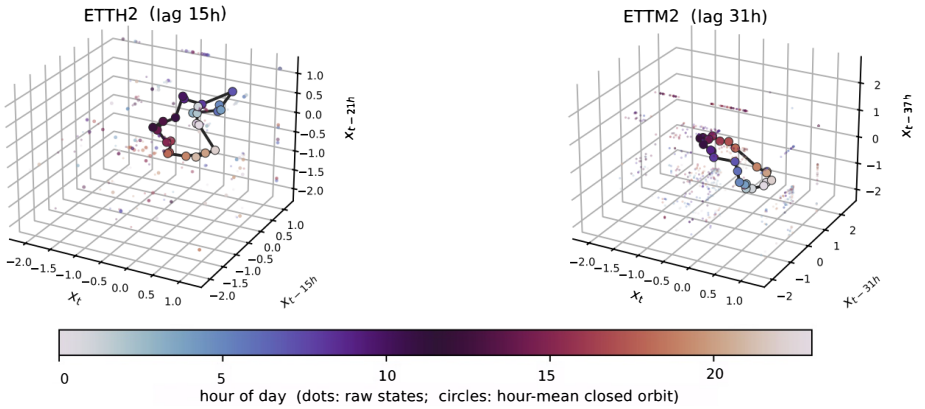

Figure 3: Delay-embedding state space of ETTh2/ETTm2 (standardized 4-week windows). 3D view of $(x_t, x_{t-\ell}, x_{t-\ell-6\mathrm{h}})$ with the hour-mean closed orbit overlaid (solid loop, 24 points, 00:00 starred). Variate/lag chosen per dataset at the quarter phase of the daily cycle (raw LULL at ℓ=15h/31h for ETTh2/ETTm2). The daily cycle appears as a closed loop that winds once per 24 h.

Table 4: Factor-isolation screen: average percentage change in test MSE vs. the shared cross-channel baseline (d=128, 4 heads, 3,000 sampled windows per epoch, batch 32), averaged over $h \in \{96, 192, 336, 720\}$; the first two rows give the horizon-averaged MSE for scale. Each factor is toggled alone; "+full" combines all three; "+d512+sub12k" is the vemb-free wide combination (run on Traffic, batch 8). Bold: most negative change per column. The selected per-dataset configurations and their average gap to iTransformer are shown in the last row. Single seed.

	Weather (21 ch)	Traffic (862 ch)	PrivEM DA (53 ch)	PrivEM RT (53 ch)
shared baseline, avg MSE	0.258	0.462	0.669	1.871
iTransformer, avg MSE	0.261	0.425	0.635	1.639
+vemb	$\mathbf{-3.4\%}$	$+3.5\%$	$\mathbf{-10.2\%}$	$\mathbf{-24.9\%}$
+d512	$+8.8\%$	$\mathbf{-6.3\%}$	$+7.9\%$	$+16.6\%$
+sub12k	$+1.8\%$	-4.9%	$+9.5\%$	$+29.5\%$
+full (all three)	$+5.9\%$	$+18.0\%$	$+6.0\%$	-16.3%
+d512+sub12k (no vemb)	—	-6.5%	—	—
selected (avg Δ vs. iTransformer)	+vemb: -4.2%	d512+sub12k: $+1.6\%$	+vemb: -5.6%	+vemb: -14.2%

rotation lives directly in the coordinates, while spike episodes depart from the loop and relax back onto it—the contraction that the $D \succeq 0$ part of the generator encodes.

B Additional Experiments

B.1 The Factor-Isolation Screen

This appendix documents the screen summarized in Table 4. It is a controlled one-factor-at-a-time grid against a shared cross-channel baseline: d=128, 4 attention heads, 3,000 sampled windows per horizon per epoch, batch 32, 30 epochs, single seed, with data protocol and evaluation identical to the corresponding main-table rows (TSLib borders and z-space for Weather/Traffic; the PrivEM calendar split of Table 2). The factors are: +**vemb**, a learnable variate embedding (one d-dimensional vector per channel, $0.02\mathcal{N}(0, 1)$-initialized, added to the channel tokens before cross-channel attention—iTransformer-style channel identity; without it the encoder is permutation-symmetric across channels); +**d512**, a widened latent (d=512, 8 heads; 8.9M vs. 1.7M parameters on Traffic); +**sub12k**, full-epoch sampling (12,000 windows per epoch); and +**full**, all three combined. On Traffic every d=512 run uses batch 8—at 862 channels the attention backward allocates a $\sim$27 GiB score tensor at batch 32, exceeding the 40 GB A100 together with the remaining activations—which is the only protocol deviation.

On the 53-channel PrivEM RT task the same +full combination behaves in the opposite way: it recovers most of the variate embedding's gain (-16.3% vs. -24.9%; 1.398 vs. 1.234 at h=96) even

Table 5: 200-step rollout behavior by parameterization (spectral diagnostics).

Parameterization	ρ	$\sigma_{\max}$	Rollout norm (200 steps)	Bounded?
Unconstrained (DMD)	0.999	1.25	9.0→65.9 (transient)	No
Hard-unitary	1.000	1.00	8.68→8.68 (undamped)	Yes; undamped
Dissipation+rotation (PWM)	0.999	**1.00**	9.2→7.0 (decays)	Yes

The unconstrained and unitary rows come from the controlled synthetic experiment (same latent dimension); the PWM row reports the transition learned on ECL with the identical diagnostics.

Table 6: Encoder-side ablations on ETTh2/ETTm2 (MSE, z-space, lookback 96, $h \in \{96, 720\}$; identical protocol, budget, and seed unless noted). "Scaled" = 8,000 sampled windows/epoch $\times$ 50 epochs; "per-horizon" = one model per horizon, the benchmark convention under which the ETT baseline columns of Table 1 are produced. Channel independence is the single largest factor; patching, pooling, schedule, and per-horizon specialization are neutral or harmful.

	ETTh2		ETTm2	
Variant	96	720	96	720
joint RevIN, cross-channel attention	0.268	0.489	0.166	0.342
− cross-channel attention (channel-indep.)	0.246	**0.438**	0.155	0.314
+ linear pathway (DLinear-style)	**0.244**	0.439	0.155	0.315
+ deeper token encoder (**ours**)	**0.244**	0.444	0.155	**0.314**
scaled schedule	0.255	0.453	0.156	0.318
linear pathway, scaled schedule	0.251	0.457	0.156	0.317
patch-temporal encoder (PatchTST-CI)	0.262	0.485	0.161	0.333
+ learned recency pooling	0.263	0.477	0.160	0.328
per-horizon models (deeper token enc.)	0.248	0.440	**0.153**	0.315

though capacity and exposure each hurt severely when toggled alone ($+16.6\%$ and $+29.5\%$). At this channel count the identity pathway dominates the wide configuration's overfitting—the mirror image of Traffic, where the identity pathway poisons the wide configuration.

B.2 Structural Stability of the Parameterization

Table 5 compares rollout stability of the learned transition under three generator parameterizations: unconstrained (raw DMD-style), hard-unitary, and dissipation-plus-rotation.

The three rows differ only in how the transition is constrained. The unconstrained baseline learns a raw transition matrix with no structural constraint (DMD-style). The hard-unitary variant restricts the transition to the orthogonal group,

$$K_u = UV^\top \;\text{ with }\; K = U\,\Sigma\,V^\top \text{ the SVD}, \qquad K_u^\top K_u = I, \tag{7}$$

realized in the experiment as the polar factor of the unconstrained matrix—the nearest orthogonal matrix to K, with all singular values set to one. Equivalently, its trainable form is $K_u = \exp(S)$ with $S^\top = -S$ skew-symmetric, the $D \equiv 0$ degenerate case of the generator of Eq. (4): every eigenvalue then lies exactly on the unit circle, $\lambda_i = e^{\mathrm{j}\omega_i}$ with $|\lambda_i| = 1$, so $\rho = \sigma_{\max} = 1$ hold by construction. The PWM row is the full dissipation-plus-rotation generator of Eq. (4), with $D \succeq 0$ learnable.

The unconstrained generator has spectral radius below one yet diverges under rollout—the classic non-normal transient-amplification phenomenon ($\sigma_{\max} > 1$)—and post-hoc spectral clipping cannot repair it. The hard-unitary constraint of Eq. (7) conserves the latent norm exactly, $\|K_u\,\mathbf{z}\| = \|\mathbf{z}\|$ for every state and hence $\|K_u^h\,\mathbf{z}\| = \|\mathbf{z}\|$ at every horizon: the rollout is bounded by construction, but with every decay rate forced to $\mu_i = 0$ the constraint erases damping, producing undamped equal-amplitude oscillations that contradict load physics. Only the dissipation-plus-rotation parameterization simultaneously bounds the rollout ($\sigma_{\max} = 1.0$) and preserves decay, confirming the proposition of Section 3.3 empirically.

B.3 Encoder-Side Ablations on ETT

Table 6 reports the structure-targeted screen behind the PWM rows of the ETT block of Table 1, run after the RevIN fix with one factor varied at a time under identical protocol, budget, and seed. Three findings stand out. **(i) Channel independence is the single largest accuracy factor.** Removing the encoder's cross-channel attention improves all 8 dataset–horizon cells, by 6–8% on both the hourly and the 15-min series: the seven ETT variates (two transformer loads plus engineered temperature/exogenous channels) are too heterogeneous for attention mixing to help, echoing the channel-independence analyses behind DLinear and PatchTST-CI. A deeper per-channel token MLP adds a further small gain on ETTh2 (-0.8% at $h{=}96$) and is neutral on ETTm2. **(ii) Capacity and budget are not the limiter.** Scaling the training budget (8,000 windows/epoch, 50 epochs) degrades nearly every cell—validation loss still improves while test loss rises, i.e. the longer schedule over-fits training-regime structure—and the DLinear-style linear pathway is neutral ($\pm 0.5\%$). **(iii) Protocol is not the limiter either.** Training one model per horizon (the benchmark convention) changes cells by at most 1.6% with mixed signs (its single advantage is ETTm2 $h{=}96$, -1.3%), so the joint multi-horizon quantile protocol of the ETT block of Table 1 is not a handicap.

The two patch-encoder rows deserve a separate note because they are informative negative results. A PatchTST-style per-channel patch encoder under-performs the plain dense token projection everywhere, and the gap widens with more training. To test whether the deficit comes from pooling granularity, we replaced mean pooling over patch tokens with a *learnable recency-weighted pooling* (softmax over patch positions, initialized as a linear ramp favoring the most recent patch). The learned weights are diagnostic: on ETTm2 the model strengthens the recency ramp (last/first weight ratio $3.6\times$) yet gains nothing over mean pooling; on ETTh2 it flattens the ramp away; and on neither dataset does the *final* patch receive the largest weight—recency information is already supplied by the last-observation anchor branch, so the encoder learns to look slightly further back. Pooling granularity, in short, is not the bottleneck; the information-carrying capacity of the single-token latent is, consistent with the reading of the ETT block of Table 1.