WHY LEARN WHAT YOU CAN COMPUTE? CLOSED-FORM OPERATORS FOR PERIODIC TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Periodic patterns are a defining characteristic of many real-world time series, and exploiting them has become central to modern long-horizon forecasting. Yet most existing learned forecasters that leverage such patterns still rely on iterative gradient-based optimization. We show that period folding transforms a period-p wide-sense cyclostationary process into phase-indexed sequences with cycle-invariant first- and second-order statistics, and that multi-horizon forecasting on the resulting finite delay-coordinate dictionary is a Tikhonov-regularized Galerkin projection with a unique closed-form solution. Under ergodic sampling, the empirical estimator converges to its population projected counterpart, while exact Koopman closure arises as a special case when the folded process has an affine cycle-conditional mean. This theoretical result directly yields **CAPE** (**C**losed-form **A**pproximation of **P**eriodic **E**mbeddings), a gradient-free algorithm that estimates a compact cycle-level predictor, supports direct transfer when source and target datasets have compatible projected cycle structure, and adaptively reduces to a direct multi-output ridge autoregressor when $p^\star = 1$. To evaluate CAPE, we conduct extensive experiments across zero-shot, few-shot, in-distribution, and cross-domain tasks. CAPE ranks first in 20 of 28 in-distribution MSE settings, wins seven of eight within-family and all three cross-domain zero-shot transfers, and achieves the best few-shot result, without gradient-based or GPU training and with under one second of CPU fitting. Code is available at `https://anonymous.4open.science/r/cape-iclr-7D22`

1 INTRODUCTION

On the limitations of learned forecasters. Modern time-series forecasting models, whether linear (Zeng et al., 2023; Lin et al., 2024b), spectral (Xu et al., 2024), or Transformer-based (Nie et al., 2023; Liu et al., 2024; Wang et al., 2025), share a common design paradigm: the forecasting function is *learned* from data via iterative gradient-based optimization. While highly effective in-distribution, this paradigm often induces a degree of *dataset specialization*, limiting transferability and requiring re-training or fine-tuning when applied to domains with different statistical properties. Moreover, the optimization process may capture dataset-specific noise and inductive biases, such as spectral bias, which can reduce robustness under distribution shift, even when underlying temporal structures are shared.

Our observation. The success of recent lightweight forecasters (Lin et al., 2024a;b; Ma et al., 2025) suggests that periodicity is not merely a convenient inductive bias, but a recurring structural property of real-world time series. However, existing approaches typically exploit this structure heuristically, through periodic reshaping, sparse mappings, or frequency-domain components. For cyclostationary systems, whose statistics repeat over a fixed dominant period, Koopman operator theory (Koopman, 1931) provides a principled alternative: nonlinear dynamics can be represented linearly after lifting the system into an appropriate observable space. This perspective shifts the central challenge from model design to constructing a suitable observable space, a task CAPE solves in closed form while retaining competitive performance even on aperiodic benchmarks.

Consequences. We show that period-folding, reshaping a length-L time series into an $S \times p$ matrix aligned with its dominant periodic structure (see Figure 1), constructs a finite delay-coordinate

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.

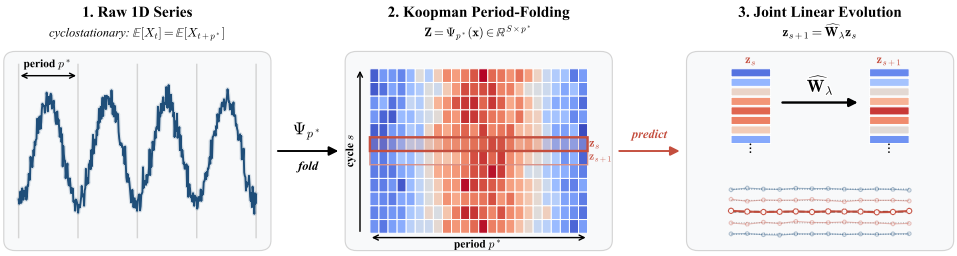


Figure 1: Period-folding maps a cyclostationary series into phase-aligned cycles, motivating a finite-dimensional cycle-transition operator (Section 3.2).

dictionary with phase-wise cycle-invariant first- and second-order statistics. On this dictionary, forecasting reduces to a closed-form projected operator estimation problem, where the regularized evolution operator is given by the unique solution of a normal equation, rather than learned through iterative optimization. Because this operator captures cycle-level dynamics rather than dataset-specific parameters, it naturally supports zero-shot forecasting across datasets that share similar periodic patterns. Importantly, when no dominant periodic structure is present, the formulation degrades gracefully to a ridge autoregressive model, ensuring robust performance beyond strictly cyclostationary settings.

Contributions.

1. **Theoretical.** We prove that period folding transforms a wide-sense cyclostationary process into phase-indexed sequences with cycle-invariant first- and second-order statistics. We then characterize CAPE as a closed-form Tikhonov-regularized Galerkin projection of multi-step cycle evolution onto a finite delay-coordinate dictionary, establish its consistency under ergodic sampling, and identify affine cycle-conditional dynamics as a sufficient condition for exact Koopman closure.
2. **Method.** We introduce **CAPE**, a closed-form Tikhonov-regularized EDMD estimator that requires no gradient-based training. CAPE computes the evolution operator directly, encodes cycle-level dynamics instead of dataset-specific weights, and *gracefully reduces to a ridge autoregressor in the absence of periodic structure*, enabling robust performance on both periodic and aperiodic time series.
3. **Analysis.** We establish (Theorem 2) an $\mathcal{O}(1/\sqrt{N_{\text{total}}})$ generalization bound for CAPE under stated moment and dependence assumptions, and show (Proposition 1) that gradient-based training and Tikhonov regularization impose distinct spectral filters, while CAPE recovers the Tikhonov solution in closed form.
4. **Empirical.** We demonstrate that CAPE outperforms strong baselines in both zero-shot and few-shot forecasting. It wins $7/8$ zero-shot transfer pairs against TIME-LLM, ranks first on all reported ETT transfer pairs against TimeMixer++, PatchTST, and iTransformer, surpassing the eight reported baselines in few-shot settings, while requiring no gradient-trained parameters, no GPU training, under one second of CPU fitting in the reported timing experiment, and achieving a $13.6\times$ speedup.

2 Related Work

Linear and spectral forecasters. Recent work has shown that simple architectures can rival complex models for time-series forecasting. Zeng et al. (2023) demonstrated that a single linear layer can match or surpass Transformer-based approaches, while SparseTSF (Lin et al., 2024b) introduces period-aware downsampling to sparsify linear mappings. FITS (Xu et al., 2024) instead operates in the frequency domain via learned low-pass filters. Despite their simplicity, these methods rely on gradient-based optimization and treat model parameters as learned weights, without explicitly encoding the underlying temporal structure.

Transformer-based forecasters. Transformer-based models such as PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), Crossformer (Zhang and Yan, 2023), FEDformer (Zhou et al., 2022), TimesNet (Wu et al., 2023), and TiDE (Das et al., 2023) model temporal and cross-variate dependencies using attention or structured architectures. TimeMixer++ (Wang et al., 2025) achieves SOTA performance across multiple forecasting tasks, including zero-shot transfer, and establishes the evaluation protocol we adopt. While effective, these models require substantial computing and are typically retrained or fine-tuned for each target dataset.

Classical cyclostationary signal processing. Cyclostationary and periodically correlated processes have long been studied in signal processing, including cyclic spectral analysis and the spectral theory of periodically correlated sequences (Gardner, 1994; Hurd and Miamee, 2007). This literature directly characterizes periodic second-order structures, Our work bridges this gap by connecting the cyclostationary structure to Koopman-aligned operator estimation.

Koopman operator theory in machine learning. Koopman operator theory (Koopman, 1931) provides a linear representation of nonlinear dynamical systems in an appropriate observable space, with EDMD (Williams et al., 2015) offering a data-driven approximation. Recent approaches, including KNF (Wang et al., 2023), Koopa (Liu et al., 2023), KoopAGRU (Ait Yahia and Berrada, 2025), and deep autoencoder-based methods (Lusch et al., 2018), construct this space using neural networks trained via gradient-based optimization. In contrast, period folding constructs a finite delay-coordinate dictionary on which CAPE estimates the Galerkin-projected cycle Koopman evolution in closed form, without learned embeddings.

Zero-shot and transfer learning in time series. Foundation models such as TimesFM (Das et al., 2024), Moirai (Liu et al., 2026), TimeLLM (Jin et al., 2024), GPT4TS (Zhou et al., 2023) and Chronos (Ansari et al., 2024) achieve zero-shot transfer through large-scale pretraining on diverse datasets. TTM (Ekambaram et al., 2024) reduces model size but still requires pretraining on hundreds of millions of samples. TimeMixer++ further demonstrates strong zero-shot performance under direct transfer protocols. CAPE offers a qualitatively different mechanism: it transfers *cycle-level operators*, requiring no pretraining, only a single source dataset and a closed-form computation.

3 CAPE (CLOSED-FORM APPROXIMATION OF PERIODIC EMBEDDINGS)

Let $\mathbf{z}_{t+1} = f(\mathbf{z}_t)$ be a discrete-time dynamical system on a smooth manifold $\mathcal{M}$. The **Koopman operator** $\mathcal{K}$ lifts the nonlinear state dynamics to a linear operator on observables, acting as $(\mathcal{K}g)(\mathbf{z}) = g(f(\mathbf{z}))$ for $g : \mathcal{M} \to \mathbb{R}$. Let $\mathcal{F} = L^2(\Omega)$ be the space of square-integrable observables on the state space $\Omega \subseteq \mathcal{M}$. A subspace $\mathcal{V} \subseteq \mathcal{F}$ is **Koopman-invariant** when $\mathcal{K}\mathcal{V} \subseteq \mathcal{V}$.

Given a finite dictionary of observables $\mathbf{\Psi} = (\psi_1, \ldots, \psi_S)$, where $\psi_i : \Omega \to \mathbb{R}$, and a regularization parameter $\lambda > 0$, **EDMD** (Extended Dynamic Mode Decomposition) estimates a finite-dimensional approximation $\widehat{\mathbf{K}}$ of $\mathcal{K}$ via the Tikhonov-regularized (Tikhonov, 1963) normal equation:

$$\widehat{\mathbf{K}} = \Big(\sum_t \mathbf{\Psi}(\mathbf{z}_t)\mathbf{\Psi}(\mathbf{z}_t)^\top + \lambda \mathbf{I}\Big)^{-1} \sum_t \mathbf{\Psi}(\mathbf{z}_t)\mathbf{\Psi}(\mathbf{z}_{t+1})^\top. \tag{1}$$

Thus, EDMD shifts the difficulty from learning the dynamics to selecting an observable dictionary. Our key observation is that, in cyclostationary systems, this dictionary is not external to the problem: it is revealed by the dominant periodic structure of the data.

3.1 PERIOD FOLDING AS A PROJECTED CYCLE-SHIFT REPRESENTATION

We now formalize the observable dictionary implicit in cyclostationary time series. Let $\{X_t\}$ be a time series of length L. $\{X_t\}$ is a **wide-sense cyclostationary** process with dominant period p if its mean $\mu(t) = \mathbb{E}[X_t]$ and lag-τ covariance $C(t, \tau) = \mathrm{Cov}(X_t, X_{t+\tau})$ satisfy:

$$\mu(t) = \mu(t + p), \qquad C(t, \tau) = C(t + p, \tau), \quad \forall\, t, \tau \in \mathbb{Z}, \tag{2}$$

where τ denotes the temporal lag. Stationarity is recovered when $p = 1$ (constant mean and covariance). Several standard forecasting benchmarks, including ETT, Electricity, and Traffic, exhibit strong daily or weekly periodic structure (Appendix B.1), although CAPE does not assume that every dataset contains a single exact physical period.

The **period-folding map** $\Psi_p : \mathbb{R}^L \to \mathbb{R}^{p\times S}$, with $S = \lfloor L/p \rfloor$, reshapes the series $\{X_t\}$ into S consecutive windows of length p, aligning observations by their phase within the cycle. We denote the observation at phase i within cycle s as $Z_s^{(i)} = X_{i+(s-1)p}$, and collect all phases into the lifted state vector $\mathbf{z}_s = (Z_s^{(1)}, \ldots, Z_s^{(p)})^\top \in \mathbb{R}^p$. Each $\mathbf{z}_s$ represents one complete cycle, and $\{\mathbf{z}_s\}_{s=1}^S$ is the resulting cycle-level trajectory.

Theorem 1 (Period folding and projected cycle evolution)**.** *Let $\{X_t\}_{t\in\mathbb{Z}}$ be a square-integrable wide-sense cyclostationary process with period p. For each phase $i \in \{1, \ldots, p\}$, define the cycle-indexed sequence $Z_s^{(i)} = X_{i+(s-1)p}$, the S-cycle history $h_s^{(i)} = \left(Z_{s-S+1}^{(i)}, \ldots, Z_s^{(i)}\right)^\top \in \mathbb{R}^S$, and the M-cycle target $y_s^{(i)} = \left(Z_{s+1}^{(i)}, \ldots, Z_{s+M}^{(i)}\right)^\top \in \mathbb{R}^M$. Then the following statements hold:*

1 - Phase-wise second-order stationarity. *For every pair of phases i, j and every cycle lag ℓ, the mean $\mathbb{E}[Z_s^{(i)}]$ and covariance $\mathrm{Cov}(Z_s^{(i)}, Z_{s+\ell}^{(j)})$ are independent of the cycle index s.*

2 - Projected cycle evolution. *Let $G = \sum_{i=1}^p \mathbb{E}[h_s^{(i)} h_s^{(i)\top}]$ and $A = \sum_{i=1}^p \mathbb{E}[h_s^{(i)} y_s^{(i)\top}]$. For $\lambda > 0$, the unique minimizer of the population Tikhonov objective $\sum_{i=1}^p \mathbb{E}\left[\|y_s^{(i)} - W^\top h_s^{(i)}\|_2^2\right] + \lambda\|W\|_F^2$ is $W_\lambda^\star = (G + \lambda I)^{-1} A$. Each column of $W_\lambda^\star$ is the regularized L^2 projection of a future cycle-shifted observable onto the finite-dimensional history dictionary.*

3 - Empirical consistency. *If the folded phase sequences are jointly stationary and ergodic across cycles and have finite second moments, define $\mathbf{G}_n = \sum_{r=1}^n \sum_{i=1}^p \mathbf{h}_r^{(i)} \mathbf{h}_r^{(i)\top}$ and $\mathbf{A}_n = \sum_{r=1}^n \sum_{i=1}^p \mathbf{h}_r^{(i)} \mathbf{y}_r^{(i)\top}$. If the empirical regularization satisfies $\lambda_n/n \to \lambda > 0$, then $\widehat{\mathbf{W}}_n = (\mathbf{G}_n + \lambda_n \mathbf{I})^{-1} \mathbf{A}_n \xrightarrow{\text{a.s.}} \mathbf{W}_\lambda^\star$ as $n \to \infty$.*

4 - Exact closure as a special case. *If the folded process has a common affine cycle-conditional mean, $\mathbb{E}[Z_{s+1}^{(i)} \mid h_s^{(i)}] = a^\top h_s^{(i)} + b$ for every $i \in \{1, \ldots, p\}$, then the affine delay-coordinate observable space is invariant under the associated stochastic cycle Koopman operator. Without this additional condition, CAPE estimates its finite-dimensional Galerkin projection and does not assume exact invariance.*

Proof: Appendix A.1.

3.2 The CAPE Pipeline

Theorem 1 establishes that period folding produces phase-indexed trajectories whose first- and second-order statistics are constant across cycles, and characterizes the optimal regularized map from S historical cycles to $M \geq 1$ future cycles as a finite-dimensional Galerkin projection. CAPE implements the empirical form of this projected cycle evolution by pooling phases and channels as trajectories and estimating $\widehat{\mathbf{W}}_\lambda \in \mathbb{R}^{S\times M}$ through a single closed-form solve. Each column of $\widehat{\mathbf{W}}_\lambda$ contains the projection coefficients for one future cycle, rather than representing an exact finite-dimensional one-step Koopman operator. We call this the **CAPE estimator** and use $\widehat{\mathbf{W}}_\lambda$ exclusively hereafter. As shown in Figure 2, the CAPE pipeline consists of the following four steps:

1-Period detection. The dominant period p^* is identified automatically via a scoring function that combines the FFT power spectrum (Cooley and Tukey, 1965) and autocorrelation (ACF) (Box et al., 2015), with mixing weight $\alpha \in [0, 1]$:

$$\mathcal{S}(p) = \alpha \cdot \frac{\sum_{h=1}^3 P[h \cdot L/p]}{3(\max_k P[k] + 10^{-10})} + (1 - \alpha) \cdot \max(0, R(p)), \qquad p^* = \arg\max_{p\in\mathcal{P}} \mathcal{S}(p), \tag{3}$$

where $P[\cdot]$ denotes the FFT power spectrum, with $h \in \{1, 2, 3\}$ capturing the fundamental and first two harmonics, sufficient to detect non-sinusoidal periodic signals while avoiding high-frequency noise ($h \geq 4$), following Wu et al. (2023); $R(p)$ is the empirical ACF at lag p, and $\mathcal{P} = \{p \in \mathbb{Z}^+ : p \mid L,\ p \geq 8\} \cup \{1\}$. The constraint $p \geq 8$ avoids degenerate high-frequency fitting. When no dominant period is detected, $p^* = 1$, and CAPE reduces to a ridge autoregressor, consistent with Theorem 1.

2-Fold. Each input series $X \in \mathbb{R}^{L\times C}$ is mapped by Ψ_{p^*} into $S = \lfloor L/p^* \rfloor$ historical cycles. We adopt Channel Independence (CI) (Nie et al., 2023), treating each of the C channels independently and pooling their lifted windows into the design matrices $\mathbb{X} \in \mathbb{R}^{N_{\text{total}}\times S}$ and $\mathbb{Y} \in \mathbb{R}^{N_{\text{total}}\times M}$, where $N_{\text{total}} = B \cdot C \cdot p^*$, B is the batch size, H the forecast horizon, and $M = \lceil H/p^* \rceil$ denotes the

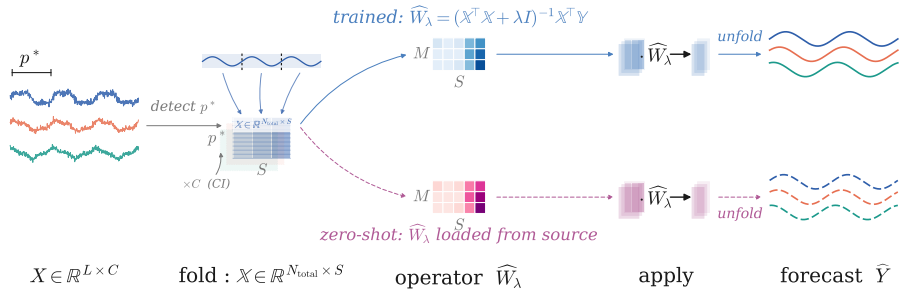


Figure 2: **CAPE pipeline.** Period-folding lifts the input into a period-aligned delay-coordinate dictionary, where a single matrix inversion yields the forecasting operator $\widehat{\mathbf{W}}_\lambda$. Solid path: in-distribution forecasting. Dashed path: zero-shot transfer without retraining.

number of forecast cycles. Importantly, CI is not merely a computational convenience: it preserves the period-aligned delay-coordinate structure at the channel level while increasing the number of pooled training trajectories by a factor of C (Appendix E.3). Prior to folding, each series is preprocessed via instance normalization and residual smoothing (Appendix C.1).

3-Solve. The forecasting operator is estimated in a single closed-form solve (CAPE estimator):

$$\boxed{\widehat{\mathbf{W}}_\lambda = \left(\mathbb{X}^\top\mathbb{X} + \lambda\mathbf{I}\right)^{-1}\mathbb{X}^\top\mathbb{Y}} \tag{4}$$

yielding $\widehat{\mathbf{W}}_\lambda \in \mathbb{R}^{S\times M}$. For a standard configuration ($L = 720$, $p^* = 24$, $H = 96$), $\widehat{\mathbf{W}}_\lambda \in \mathbb{R}^{30\times 4}$ contains 120 parameters, compared to 518,400 for DLinear (4,320$\times$ fewer).

4-Unfold. At inference, given a test input $\mathbf{X} \in \mathbb{R}^{L\times C}$, forecasts are obtained by folding $\mathbf{X}$ via Ψ_{p^*}, applying the estimated operator $\widehat{\mathbf{W}}_\lambda$, and unfolding the result is:

$$\widehat{\mathbf{Y}} = \Psi_{p^*}^{-1}\left(\Psi_{p^*}(\mathbf{X})\,\widehat{\mathbf{W}}_\lambda\right) \tag{5}$$

3.3 CAPE Analysis

CAPE is structured EDMD, not generic ridge. The key modeling choice is the observable space. CAPE folds the series using the dominant period p^*, turning phases and channels into repeated cycle-transition trajectories before applying the Tikhonov solver. This choice matters: (1) a full raw ridge model on the unfolded input performs well in-distribution but transfers poorly in zero-shot tasks, (2) a same-size raw ridge fails entirely, and (3) a period-folded ridge without residual smoothing still underperforms CAPE (Appendix E.1).

Zero-shot transfer. The source-fitted operator $\widehat{\mathbf{W}}_\lambda$ maps period-folded histories to future cycles. For zero-shot forecasting, we reuse this operator and the source folding period on the target dataset:

$$\widehat{\mathbf{Y}}_{\text{target}} = \Psi_{p^*}^{-1}\left(\Psi_{p^*}(\mathbf{X}_{\text{target}})\,\widehat{\mathbf{W}}_\lambda^{(\text{source})}\right). \tag{6}$$

We evaluate this transfer procedure in Section 4.2.

Generalization and Spectral Properties. Having defined the CAPE estimator in Eq. equation 4, we analyze two key properties.

P1. *How does generalization scale with training trajectories?* Theorem 2 establishes an $\mathcal{O}(N_{\text{total}}^{-1/2})$ generalization bound for CAPE. On ETTh1, $N_{\text{total}} = B \cdot C \cdot p^* \approx 588{,}000$ compared to $C \cdot p^* = 168$ for a test-window self-supervised variant, a 3,500$\times$ difference in row count. The associated forecasting performance is evaluated empirically in Appendix E.4.

Theorem 2 (CAPE generalization under dependent sampling)**.** *Fix the forecasting dimensions, period, and preprocessing map. Assume the folded input–target pairs have finite fourth moments, fixed population moments, and uniformly summable covariances of their quadratic statistics, as specified in Appendix A.2. Let $\mathcal{E}(\mathbf{W})$ be the population squared loss for the same phase/channel mixture as the training rows, and let $\widehat{\mathcal{E}}_N(\mathbf{W}) = N^{-1}\|\mathbb{Y} - \mathbb{X}\mathbf{W}\|_F^2$, where $N = N_{\text{total}} = B \cdot C \cdot p^*$ and B counts all training windows accumulated in the solve. For any $\lambda > 0$, any fixed $\delta \in (0, 1)$, and all sufficiently large N, with probability at least $1 - \delta$,*

$$\mathcal{E}(\widehat{\mathbf{W}}_\lambda) \le \widehat{\mathcal{E}}_N(\widehat{\mathbf{W}}_\lambda) + \mathcal{O}(N^{-1/2}). \tag{7}$$

The constant depends on the population moments, their dependence structure, and δ, but not on N.

Proof and extension to selection from finitely many candidate periods: Appendix A.2.

P2. *Why does CAPE transfer better than gradient descent?* Proposition 1 states that GD and CAPE solve the same least-squares problem but apply different spectral filters, where GD uses implicit early stopping that cuts off small singular directions, while CAPE uses explicit Tikhonov regularization that shrinks all directions smoothly. CAPE therefore controls ill-conditioned directions in closed form, without tuning an optimization trajectory. This explains its stronger zero-shot transfer and its $13.6\times$ wall-clock speedup over DLinear via a single $\mathcal{O}(S^3)$ solve (Section 5).

Proposition 1 (GD–CAPE spectral equivalence)**.** *Let $\mathbb{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be the SVD of the design matrix, with singular values $\{\sigma_i\}_{i=1}^S$ and stopping time $T > 0$. Under zero-initialized gradient flow $\frac{d\mathbf{W}}{dt} = \mathbb{X}^\top(\mathbb{Y} - \mathbb{X}\mathbf{W})$, the solution at time T and the CAPE estimator of Eq. equation 4 are respectively:*

$$\underbrace{\mathbf{W}_{\text{GD}}(T) = \mathbf{V}\,\text{diag}\Big((1 - e^{-T\sigma_i^2})/\sigma_i\Big)\,\mathbf{U}^\top\mathbb{Y}}_{\textit{Gradient flow (implicit filter)}} \quad \underbrace{\widehat{\mathbf{W}}_\lambda = \mathbf{V}\,\text{diag}\big(\sigma_i/(\sigma_i^2 + \lambda)\big)\,\mathbf{U}^\top\mathbb{Y}}_{\textit{CAPE (Tikhonov filter)}} \tag{8}$$

Proof: Appendix A.3.

4 Experiments

Table 1: **Multivariate long-term forecasting.** MSE and MAE averaged over $H \in \{96, 192, 336, 720\}$. Counts cover all 28 dataset–horizon settings in Table 19. Values are truncated to three decimals. **Bold**: best; underline: second-best distinct displayed value.

Dataset	CAPE		TimeMixer++		PatchTST		iTransformer		DLinear		SparseTSF	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.399**	**0.414**	0.416	0.430	0.515	0.484	0.425	0.433	0.437	0.448	0.406	0.420
ETTh2	**0.332**	**0.380**	0.334	0.382	0.391	0.411	0.394	0.411	0.479	0.471	0.361	0.393
ETTm1	**0.355**	0.376	0.383	**0.360**	0.406	**0.360**	0.370	0.401	0.360	0.383	0.361	0.382
ETTm2	**0.249**	**0.312**	0.251	0.314	0.284	0.337	0.278	0.337	0.271	0.337	0.251	**0.312**
Weather	0.240	0.279	0.239	0.279	0.265	0.288	**0.238**	**0.274**	0.242	0.299	0.241	0.277
Electricity	**0.165**	0.262	0.167	0.261	0.216	0.317	0.175	**0.243**	0.168	0.271	0.180	0.277
Traffic	**0.409**	0.291	0.416	0.280	0.529	0.341	0.438	0.289	0.418	0.285	0.413	**0.279**
Best / second-best counts across all 28 dataset–horizon settings												
MSE	20/6		4/8		1/0		3/2		1/5		2/11	
MAE	12/6		7/7		1/2		4/3		0/4		8/8	

We follow the standard protocol of Zeng et al. (2023); Nie et al. (2023) on ETT (ETTh1, ETTh2, ETTm1, ETTm2), Weather, Electricity, Traffic, and Exchange-Rate, the matched-$L = 96$ comparison appears in Table 18. Baselines include DLinear, SparseTSF, FITS, iTransformer, TimeMixer++, and PatchTST. CAPE selects $\lambda \in \{0.01, 0.1, 1, 5, 10, 100\}$ via one-pass cross-validation on source data only.

4.1 In-Distribution Forecasting

Table 1 reports multivariate long-term forecasting results across seven standard benchmarks. CAPE is consistently competitive, achieving the best or second-best MSE in the majority of settings, with no

single gradient-trained baseline dominating across regimes. It performs particularly well on ETTh1 (best MSE at all four horizons), ETTh2 ($H = 720$: 47% MSE reduction vs. DLinear), and Electricity (best MSE at two of four horizons). On Weather, iTransformer and TimeMixer++ edge ahead at short horizons ($H \in \{96, 192\}$), where high-frequency local patterns fall outside the slowly varying inter-cycle dynamics captured by CAPE. Traffic ($C = 862$) remains a limitation: the large number of channels and strong spatial coupling increase the effective Koopman subspace dimension beyond what the channel-independent assumption can support, leading to higher approximation error ε_p of Theorem 1, Part (4) (Appendix E.3, Table 15).

4.2 Zero-Shot Transfer

Zero-shot transfer provides a direct test of CAPE's theoretical claim: if $\widehat{\mathbf{W}}_\lambda$ encodes cycle-level transition structure rather than dataset-specific statistics, it should generalize across datasets without any target-side training. We evaluate two settings: within-family and cross-domain transfers.

Within-Family Transfer. TimeMixer++, PatchTST, and iTransformer zero-shot results are taken from Table 6 of Wang et al. (2025), while GPT4TS and TIME-LLM results are taken from Table 5 of Jin et al. (2024). DLinear is excluded from Table 2, as zero-shot results are not reported for these benchmarks, but is included in Table 4 as the strongest publicly reproducible zero-shot linear baseline. Source-trained methods are fitted on D_a and evaluated on D_b without retraining. Pretrained foundation models are included as target-evaluation references.

Table 2: **Within-family zero-shot transfer (MSE, avg H). Bold**: best. Underline: second-best distinct value. Wins include ties.

Source → Target	CAPE	TIME-LLM	GPT4TS	TimeMixer++	PatchTST	iTransformer	TimesFM	MOIRAI	Chronos
ETTh1 → ETTh2	**0.349**	0.353	0.406	0.367	0.380	0.481	0.403	0.360	0.368
ETTh1 → ETTm2	0.295	**0.273**	0.325	0.301	0.314	0.311	0.335	0.340	0.292
ETTh2 → ETTh1	**0.416**	0.479	0.757	0.511	0.565	0.552	0.451	0.433	0.445
ETTh2 → ETTm2	**0.272**	**0.272**	0.335	–	0.325	–	0.335	0.340	0.292
ETTm1 → ETTh2	**0.354**	0.381	0.433	0.417	0.439	0.434	0.403	0.360	0.368
ETTm1 → ETTm2	**0.256**	0.268	0.313	0.291	0.296	0.324	0.335	0.340	0.292
ETTm2 → ETTh2	**0.354**	**0.354**	0.435	–	0.409	–	0.403	0.360	0.368
ETTm2 → ETTm1	**0.404**	0.414	0.769	0.427	0.568	0.559	0.429	0.517	0.425
Wins (of 8)	**7**	3	0	0	0	0	0	0	0

As shown in Table 2, CAPE achieves the lowest MSE on 7 of 8 transfer pairs, outperforming both large language models and fully supervised deep models, all without any gradient-trained parameters. The only exception (ETTh1→ETTm2, 0.295 vs. 0.273) is explained by a resolution mismatch: ETTh1 is hourly, whereas ETTm2 is sampled at 15-minute intervals, so the transferred period $p^* = 24$ corresponds to 6 hours rather than a full day on the target, breaking the phase-alignment guarantee of Theorem 1, Part (1) (Appendix D.3, Table 12). The largest gain occurs on ETTh2→ETTh1, where CAPE achieves an MSE of 0.416 compared to 0.511 for TimeMixer++, corresponding to a **19% improvement over the bestlinke gradient-trained baseline**, without target-side retraining.

Table 3: **CAPE zero-shot (source: ETTh2) vs. gradient-trained baselines on ETTh1.** The Data column denotes target data used for operator fitting. **Bold**: best.

Model	Data	H=96	H=192	H=336	H=720	Avg
TimeMixer++	Full ETTh1	0.386	0.416	0.430	0.467	0.425
PatchTST	Full ETTh1	0.385	0.413	0.440	0.456	0.424
iTransformer	Full ETTh1	0.386	0.417	0.433	0.467	0.426
DLinear	Full ETTh1	**0.375**	0.405	0.439	0.472	0.423
CAPE (zero-shot, ETTh2)	None	0.403	0.414	**0.419**	**0.429**	**0.416**

Table 3 reveals a deeper result: CAPE trained on ETTh2 and applied in zero-shot outperforms TimeMixer++ trained directly on ETTh1 at long horizons ($H \in \{192, 336, 720\}$), achieving the best average MSE of 0.416 compared to 0.425 for TimeMixer++. This finding supports the theoretical prediction of Proposition 1: gradient-based optimization can encode source-specific spectral bias that does not transfer, whereas the Tikhonov filter retains only generalizable cycle-level dynamics.

Cross-Domain Transfer. We evaluate whether CAPE generalizes beyond the ETT family to datasets with different domains and channel counts. As TimeMixer++ does not provide a public cross-domain zero-shot pipeline, we compare against zero-shot DLinear under a matched CI configuration. DLinear serves as a natural reference for gradient-trained linear models under direct transfer: its weights encode dataset-specific statistics and do not generalize across domains, providing a strong lower bound on the performance of gradient-based models without adaptation.

Table 4: **Cross-domain zero-shot transfer (MSE/MAE, avg H).** CAPE and DLinear use zero-shot CI protocol (ZS). Time Mixer++ supervised (TM++sup.) is provided as a reference upper bound.

Source → Target	CAPE ZS	DLinear ZS	Δ vs. DLinear	TM++ sup.	Type
ETTh1 → ETTh2 (ref.)	**0.349/0.391**	0.481/0.468	-27%	0.367/0.391	within-family
ETTh1 → Weather	**0.280/0.321**	0.292/0.337	-4%	0.205/0.239	cross-domain
ETTh1 → Electricity	**0.222/0.306**	0.249/0.337	-11%	0.167/0.254	cross-domain
ETTm1 → Weather	**0.279/0.321**	0.301/0.333	-7%	0.205/0.239	cross-domain

As shown in Table 4, CAPE outperforms zero-shot DLinear on every transfer pair. Notably, ETTh1→Weather achieves MSE 0.280, outperforming the within-family ETTh1→ETTh2 reference (0.349) by 20% despite transferring across entirely different domains. This follows from shared operator structure: both datasets independently learn recency dominance, exponential memory decay, and a day/night pattern (Figure 6, Appendix B.2.2). Figure 10 in Appendix G.5 further confirms this consistency within the ETT family, where $\|\Delta\widehat{\mathbf{W}}\|_F = 0.244$ (26% of $\|\widehat{\mathbf{W}}_{\mathrm{ETTh1}}\|_F$). The CI formulation handles different channel counts ($C = 7, 21, 321$) without architectural adaptation (Appendix D.1).

4.3 Few-Shot Forecasting

As shown in Table 5, CAPE achieves ETT(Avg) MSE 0.382 with zero gradient-trained parameters, outperforming all eight reported baselines including TimeMixer(TM)++ (0.396) and PatchTST (0.461). Tikhonov regularization keeps the normal equations well-conditioned even with 10% of training data, supporting CAPE's empirical data efficiency.

Table 5: **Few-shot (10% data, ETT Avg MSE). Bold**: best. '.' denotes Transformer-based models.

	CAPE	TM++	TiDE	Cross.	TM	iTrans.	PatchTST	TimesNet	FED.
Grad params	**0**	5.6M	6.5M	12.5M	2.5M	9.5M	0.5M	4.7M	16.3M
ETT Avg	**0.382**	<u>0.396</u>	0.432	0.470	0.453	0.458	0.461	0.586	0.573

4.4 Aperiodic Data: Exchange-Rate

CAPE's period detection selects $p^* = 1$ with $\lambda = 100$ on Exchange-Rate, reducing to a direct multi-output ridge autoregressor on the preprocessed history. As shown in Table 6, CAPE achieves the best MSE at $H \in \{96, 192\}$ but trails DLinear and iTransformer at $H = 720$, where the direct autoregressive fallback is less accurate. This behavior confirms that the periodic prior is inferred from the data rather than imposed a priori.

Table 6: **Exchange-Rate (aperiodic financial data). Bold**: best. <u>Underline</u>: second best.

H	CAPE	TimeMixer++	DLinear	PatchTST	iTransformer	Autoformer	p^*
96	**0.0813**	<u>0.085</u>	0.088	0.088	0.086	0.197	1
192	**0.1694**	<u>0.175</u>	0.176	0.176	0.177	0.300	1
336	<u>0.3127</u>	0.316	0.313	**0.301**	0.331	0.509	1
720	0.9156	<u>0.851</u>	**0.839**	0.901	0.847	1.447	1

5 ABLATIONS

Period Sensitivity. Table 7 shows that near-miss periods ($p \in \{20, 22, 26\}$) lead to a 14-43% increase in MSE on ETTh1, whereas exact divisors or multiples of the dominant daily period degrade more gracefully. This behavior illustrates sensitivity to the folding period. The phase-wise stationarity guarantee in Theorem 1, Part (1), applies when the folding period is a period of the process.

Table 7: **Period ablation on ETTh1.**

H	p=2	p=4	p=8	p=12	p=20	p=22	p=24	p=26	p=48
96	0.372	0.376	0.385	0.378	0.438	0.492	**0.362**	0.518	0.374
720	0.423	0.424	0.428	0.425	0.464	0.497	**0.417**	0.542	0.425

Computational Efficiency. As shown in Table 8, CAPE trains $13.6\times$ faster than DLinear and $13\times$ faster than SparseTSF with zero gradient parameters, replacing iterative SGD with a single $\mathcal{O}(S^3)$ solve as predicted by Proposition 1 (Appendix C.3, Table 11).

Table 8: **Training cost on ETTh1 ($L = 720$, $H = 96$), using the same GPU (Quadro P5000).** Params = gradient parameters; $|\mathbf{W}|$ = operator size; MACs = Multiply-Accumulate Operations.

Model	Device	Epochs	Grad params	$\lvert\mathbf{W}\rvert$	MACs/sample	Train time
DLinear	GPU	10	484,512	518,400	3.64M	7.21s
SparseTSF	GPU	10	445	120	0.20M	7.06s
CAPE	GPU	0	0	120	0.20M	**0.53s (0.92s CPU)**

Zero-Shot vs. Supervised Baselines. Table 3 shows that CAPE zero-shot from ETTh2 achieves average MSE 0.416 on ETTh1, matching or improving upon fully supervised baselines trained with full access to ETTh1. CAPE wins at two of four horizons ($H \in \{336, 720\}$). This behavior aligns with Theorem 1, as $\widehat{\mathbf{W}}_\lambda$ captures inter-cycle transition geometry shared across ETT stations rather than source-specific statistics.

CAPE vs. Generic Ridge. Appendix E.1 (Table 13) shows that CAPE is not explained by generic ridge regression. A full raw ridge model achieves an average MSE of 0.509 compared to 0.416 for CAPE, a size-matched raw ridge model fails entirely, and a period-folded ridge without smoothing still underperforms. Together, these results indicate that the performance gain arises from combining period-aligned observables with the closed-form EDMD estimator.

Channel Independence (CI) vs. Joint Operators. Joint operators over channels are 2 to 6 times worse than CI operators (Appendix E.3, Table 15). The dense joint operator has C^2SM coefficients, compared with SM for the shared CI operator. The results favor the more compact CI formulation in the evaluated settings.

6 CONCLUSION

We introduced CAPE, a gradient-free forecasting framework based on period-aligned cycle histories and closed-form projected operator estimation. We showed that period folding produces phase-indexed sequences with cycle-invariant second-order statistics, and that CAPE computes a Tikhonov-regularized Galerkin projection on the resulting finite delay-coordinate dictionary. Exact Koopman closure is not required, but is recovered as a special case under affine cycle-conditional dynamics. CAPE ranks first in 20 of 28 in-distribution MSE settings, wins seven of eight within-family and all three cross-domain zero-shot transfers, and achieves the best few-shot result, without gradient-based or GPU training and with under one second of CPU fitting. These findings suggest a complementary perspective to gradient-based learning: when temporal structure is explicitly characterized and theoretically grounded, forecasting can be approached as a problem of projected operator estimation rather than parameter optimization.

DISCLOSURE OF AI USE

In this work, we used generative AI tools to aid or polish writing. We have not used generative AI tools for any other tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. All AI-assisted textual corrections were reviewed by the authors to ensure that they preserved the intended meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Issam Ait Yahia and Ismail Berrada. KoopAGRU: A Koopman-based anomaly detection in time-series using gated recurrent units. *arXiv preprint arXiv:2501.17976*, 2025.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR. Expert Certification.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Hoboken, New Jersey, 5th edition, 2015. ISBN 9781118675021.

Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, March 2010. ISSN 1052-6234.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with TiDE: Time-series dense encoder. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=pCbC3aQB5W.

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*, ICML'24. JMLR.org, 2024.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H. Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (TTMs): fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

William A. Gardner. *Cyclostationarity in Communications and Signal Processing*. IEEE Press, New York, 1994. ISBN 0780310233.

Matan Gavish and David L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014. doi: 10.1109/TIT.2014.2323359.

Harry L. Hurd and Abolghasem G. Miamee. *Periodically Correlated Random Sequences: Spectral Theory and Practice*. Wiley, 2007.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=cGDAkQo1C0p.

B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931. doi: 10.1073/pnas.17.5.315. URL https://www.pnas.org/doi/abs/10.1073/pnas.17.5.315.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. CycleNet: enhancing time series forecasting through modeling periodic patterns. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024a. Curran Associates Inc. ISBN 9798331314385.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. SparseTSF: Modeling long-term time series forecasting with *1k* parameters. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 30211–30226. PMLR / OpenReview.net, 2024b. URL https://proceedings.mlr.press/v235/lin24n.html.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting, 2026. URL https://arxiv.org/abs/2511.11698.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: learning non-stationary time series dynamics with koopman predictors. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018. doi: 10.1038/s41467-018-07210-0. URL https://doi.org/10.1038/s41467-018-07210-0.

Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. MoFo: Empowering long-term time series forecasting with periodic pattern modeling. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/7a99ad21706dec5b28f9ad715e12197f-Abstract-Conference.html.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Andrey N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics Doklady*, 4:1035–1038, 1963.

Rui Wang, Yihe Dong, Sercan Ö. Arik, and Rose Yu. Koopman neural operator forecaster for time-series with temporal distributional shifts. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=kUmdmHxK5N.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=1CLzLXSFNn.

Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015. doi: 10.1007/s00332-015-9258-5. URL https://doi.org/10.1007/s00332-015-9258-5.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: modeling time series with 10k parameters. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=bWcnvZ3qMb.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i9.26317. URL https://doi.org/10.1609/aaai.v37i9.26317.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 27268–27286. PMLR, 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, 2023. URL https://arxiv.org/abs/2302.11939.

NOTATION

Throughout this appendix, $\{X_t\}_{t\in\mathbb{Z}}$ is a wide-sense cyclostationary (WSC) time series with dominant period p and observation length L. The state space is $\Omega \subseteq \mathcal{M}$, where $\mathcal{M}$ is the smooth manifold on which the underlying dynamics evolve, and $L^2(\Omega)$ denotes the Hilbert space of square-integrable functions on Ω.

The period-folding map $\Psi_p : \mathbb{R}^L \to \mathbb{R}^{S\times p}$, with $S = \lfloor L/p \rfloor$, reshapes $\{X_t\}$ into S consecutive cycle vectors. The observation at phase $i \in \{1,\dots,p\}$ within cycle $r \in \{1,\dots,S\}$ is

$$Z_r^{(i)} = X_{i+(r-1)p}, \tag{9}$$

and the corresponding cycle-state vector is $\mathbf{z}_r = (Z_r^{(1)},\dots,Z_r^{(p)})^\top \in \mathbb{R}^p$.

For a history of S cycles and a forecasting target of M cycles, define

$$\mathbf{h}_r^{(i)} = \left(Z_{r-S+1}^{(i)},\dots,Z_r^{(i)}\right)^\top \in \mathbb{R}^S, \qquad \mathbf{y}_r^{(i)} = \left(Z_{r+1}^{(i)},\dots,Z_{r+M}^{(i)}\right)^\top \in \mathbb{R}^M. \tag{10}$$

The population Gram and history–target cross-moment matrices are

$$\mathbf{G} = \sum_{i=1}^{p} \mathbb{E}\left[\mathbf{h}_r^{(i)}\mathbf{h}_r^{(i)\top}\right] \in \mathbb{R}^{S\times S}, \qquad \mathbf{A} = \sum_{i=1}^{p} \mathbb{E}\left[\mathbf{h}_r^{(i)}\mathbf{y}_r^{(i)\top}\right] \in \mathbb{R}^{S\times M}. \tag{11}$$

For n observed cycles, the corresponding empirical matrices are

$$\widehat{\mathbf{G}}_n = \frac{1}{n}\sum_{r=1}^{n}\sum_{i=1}^{p} \mathbf{h}_r^{(i)}\mathbf{h}_r^{(i)\top}, \qquad \widehat{\mathbf{A}}_n = \frac{1}{n}\sum_{r=1}^{n}\sum_{i=1}^{p} \mathbf{h}_r^{(i)}\mathbf{y}_r^{(i)\top}. \tag{12}$$

The regularized population and empirical estimators are

$$\mathbf{W}_\lambda^\star = (\mathbf{G}+\lambda\mathbf{I})^{-1}\mathbf{A}, \qquad \widehat{\mathbf{W}}_{\lambda,n} = (\widehat{\mathbf{G}}_n+\lambda\mathbf{I})^{-1}\widehat{\mathbf{A}}_n, \tag{13}$$

where $\lambda > 0$.

The symbol $\triangleq$ denotes definitional equality. For a generic matrix $\mathbf{B} \in \mathbb{R}^{m\times n}$, $\|\mathbf{B}\|_{\mathrm{op}} = \sigma_{\max}(\mathbf{B})$ denotes its operator norm, which satisfies

$$\|\mathbf{B}\mathbf{x}\|_2 \le \|\mathbf{B}\|_{\mathrm{op}}\|\mathbf{x}\|_2 \qquad \text{for every } \mathbf{x} \in \mathbb{R}^n. \tag{14}$$

Its Frobenius norm is $\|\mathbf{B}\|_F = \sqrt{\sum_{i,j} B_{ij}^2}$.

Finally, the deterministic Koopman operator $\mathcal{K}$ acts on a scalar observable $g : \mathcal{M} \to \mathbb{R}$ according to $(\mathcal{K}g)(\mathbf{z}) = g(f(\mathbf{z}))$, where $\mathbf{z}_{t+1} = f(\mathbf{z}_t)$. The stochastic cycle Koopman operator used in Theorem 1, Part (4), is defined directly on the folded history process in its proof.

A PROOFS

We collect complete proofs of all theoretical results stated in the main paper.

A.1 PROOF OF THEOREM 1

Proof. We prove the four statements separately.

Part (1): Phase-wise second-order stationarity. Let $\mu(t) = \mathbb{E}[X_t]$ and $C(t,\tau) = \mathrm{Cov}(X_t, X_{t+\tau})$. Wide-sense cyclostationarity with period p gives $\mu(t+p) = \mu(t)$ and $C(t+p,\tau) = C(t,\tau)$ for every $t,\tau \in \mathbb{Z}$.

For phases i,j and cycle lag ℓ, $\mathrm{Cov}(Z_s^{(i)}, Z_{s+\ell}^{(j)}) = \mathrm{Cov}\big(X_{i+(s-1)p}, X_{j+(s+\ell-1)p}\big) = C\big(i+(s-1)p, \ell p + j - i\big) = C(i, \ell p + j - i)$. This quantity is also independent of s.

Moreover, $\mathbb{E}[Z_s^{(i)}] = \mu(i+(s-1)p) = \mu(i)$, which is independent of s. Hence the folded phase sequences have cycle-invariant first- and second-order statistics.

Part (2): Projected cycle evolution. Define $\mathcal{J}_\lambda(\mathbf{W}) = \sum_{i=1}^{p} \mathbb{E}\Big[\|\mathbf{y}_s^{(i)} - \mathbf{W}^\top \mathbf{h}_s^{(i)}\|_2^2\Big] + \lambda\|\mathbf{W}\|_F^2$. Using the definitions of $\mathbf{G}$ and $\mathbf{A}$ and expanding the squared norm gives

$$\mathcal{J}_\lambda(\mathbf{W}) = C_y - 2\,\mathrm{tr}(\mathbf{W}^\top \mathbf{A}) + \mathrm{tr}(\mathbf{W}^\top \mathbf{G}\mathbf{W}) + \lambda\,\mathrm{tr}(\mathbf{W}^\top \mathbf{W}), \tag{15}$$

where $C_y = \sum_{i=1}^{p} \mathbb{E}[\|\mathbf{y}_s^{(i)}\|_2^2]$ does not depend on $\mathbf{W}$. Therefore,

$$\nabla_{\mathbf{W}} \mathcal{J}_\lambda(\mathbf{W}) = 2(\mathbf{G} + \lambda \mathbf{I})\mathbf{W} - 2\mathbf{A}. \tag{16}$$

Setting the gradient to zero yields

$$(\mathbf{G} + \lambda \mathbf{I})\mathbf{W} = \mathbf{A}, \qquad \mathbf{W}_\lambda^\star = (\mathbf{G} + \lambda \mathbf{I})^{-1}\mathbf{A}. \tag{17}$$

Because $\mathbf{G} \succeq 0$ and $\lambda > 0$, $\mathbf{G} + \lambda \mathbf{I} \succ 0$; consequently, the objective is strictly convex and the minimizer is unique.

Let $\mathbf{w}_{\lambda,m}^\star$ denote column m of $\mathbf{W}_\lambda^\star$. Since $y_{s,m}^{(i)} = Z_{s+m}^{(i)}$, it satisfies

$$(\mathbf{G} + \lambda \mathbf{I})\mathbf{w}_{\lambda,m}^\star = \sum_{i=1}^{p} \mathbb{E}\Big[\mathbf{h}_s^{(i)} Z_{s+m}^{(i)}\Big]. \tag{18}$$

These are precisely the regularized normal equations for projecting the m-step cycle-shifted observable onto the dictionary of history-coordinate functions. Thus each column contains the Galerkin projection coefficients for one future cycle.

Part (3): Empirical consistency. For n observed cycle origins, define $\mathbf{G}_n = \sum_{r=1}^{n}\sum_{i=1}^{p} \mathbf{h}_r^{(i)} \mathbf{h}_r^{(i)\top}$ and $\mathbf{A}_n = \sum_{r=1}^{n}\sum_{i=1}^{p} \mathbf{h}_r^{(i)} \mathbf{y}_r^{(i)\top}$. Finite second moments ensure that every matrix entry is integrable. Joint stationarity and ergodicity therefore give $\mathbf{G}_n/n \xrightarrow{\text{a.s.}} \mathbf{G}$ and $\mathbf{A}_n/n \xrightarrow{\text{a.s.}} \mathbf{A}$.

Let λ_n be the regularization parameter used with n observed cycle origins and assume that $\lambda_n/n \to \lambda > 0$. Then

$$\widehat{\mathbf{W}}_n = \left(\frac{\mathbf{G}_n}{n} + \frac{\lambda_n}{n}\mathbf{I}\right)^{-1} \frac{\mathbf{A}_n}{n} \xrightarrow{\text{a.s.}} (\mathbf{G} + \lambda \mathbf{I})^{-1}\mathbf{A} = \mathbf{W}_\lambda^\star. \tag{19}$$

Part (4): Exact closure as a special case. For a fixed phase i, let $\mathbf{H}_s^{(i)} = \mathbf{h}_s^{(i)}$ and define the stochastic cycle Koopman operator by

$$(\mathcal{K}_i g)(\mathbf{h}) = \mathbb{E}\Big[g(\mathbf{H}_{s+1}^{(i)}) \mid \mathbf{H}_s^{(i)} = \mathbf{h}\Big]. \tag{20}$$

Consider the affine delay-coordinate space $\widetilde{\mathcal{V}} = \mathrm{span}\{1, \psi_1, \ldots, \psi_S\}$, where $\psi_r(\mathbf{h}) = h_r$.

The history update satisfies $\mathbf{H}_{s+1}^{(i)} = (h_{s,2}^{(i)}, \ldots, h_{s,S}^{(i)}, Z_{s+1}^{(i)})^\top$. Consequently, $\mathcal{K}_i 1 = 1$ and, for $r = 1, \ldots, S-1$, $\mathcal{K}_i \psi_r = \psi_{r+1}$. For the final coordinate, the assumed affine conditional mean gives

$$(\mathcal{K}_i \psi_S)(\mathbf{h}) = \mathbb{E}\Big[Z_{s+1}^{(i)} \mid \mathbf{H}_s^{(i)} = \mathbf{h}\Big] = \mathbf{a}^\top \mathbf{h} + b. \tag{21}$$

Thus every basis function is mapped into $\widetilde{\mathcal{V}}$, proving $\mathcal{K}_i \widetilde{\mathcal{V}} \subseteq \widetilde{\mathcal{V}}$. Because the coefficients $\mathbf{a}$ and b are common to all phases, the same finite-dimensional representation applies to every phase.

Without the affine conditional-mean assumption, cyclostationarity alone does not imply this invariance. Nevertheless, the projected estimator in Part (2) remains well defined and gives the finite-dimensional Galerkin approximation used by CAPE. □

A.2 Proof of Theorem 2 (CAPE Generalization Scaling)

Moment and dependence assumptions. Write $n = N_{\text{total}}$. For each folded row pair $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^S \times \mathbb{R}^M$, define

$$\mathbf{z}_i = \big(\mathrm{vec}(\mathbf{x}_i\mathbf{x}_i^\top), \mathrm{vec}(\mathbf{x}_i\mathbf{y}_i^\top), \|\mathbf{y}_i\|_2^2\big), \qquad \bar{\mathbf{z}}_n = \frac{1}{n}\sum_i \mathbf{z}_i.$$

Assume $\mathbb{E}\bar{\mathbf{z}}_n = \boldsymbol{\mu} = (\mathrm{vec}(G), \mathrm{vec}(A), q)$ is independent of n and corresponds to the population phase/channel mixture used to define $\mathcal{E}$. Individual rows may have different marginals and be dependent. For $\mathbf{d}_i = \mathbf{z}_i - \mathbb{E}\mathbf{z}_i$, assume there is a finite V, independent of n, such that

$$\max_{1\le i\le n} \sum_{j=1}^{n} |\mathbb{E}\langle \mathbf{d}_i, \mathbf{d}_j\rangle| \le V \quad \text{for every } n. \tag{22}$$

Finite fourth moments ensure the displayed covariances exist. The additional summability condition controls their accumulation; it allows overlap and correlated phases and channels. Its constant can depend on the forecasting dimensions and dependence structure.

Proof. **Moment concentration.** By Eq. (22),

$$\mathbb{E}\|\bar{\mathbf{z}}_n - \boldsymbol{\mu}\|_2^2 = \frac{1}{n^2}\sum_{i,j}\mathbb{E}\langle \mathbf{d}_i, \mathbf{d}_j\rangle \le \frac{V}{n}.$$

Markov's inequality gives an event of probability at least $1-\delta$ on which $\|\bar{\mathbf{z}}_n - \boldsymbol{\mu}\|_2 \le \epsilon_n := \sqrt{V/(n\delta)}$. Writing the empirical moments as $\widehat{G} = n^{-1}\mathbb{X}^\top\mathbb{X}$, $\widehat{A} = n^{-1}\mathbb{X}^\top\mathbb{Y}$, and $\widehat{q} = n^{-1}\|\mathbb{Y}\|_F^2$, each corresponding moment error is at most ϵ_n.

Control of the ridge operator. Suppose $G \neq 0$ and let γ be its smallest positive eigenvalue. For $\mathbf{u} \in \ker G$, $0 = \mathbf{u}^\top G\mathbf{u} = n^{-1}\sum_i \mathbb{E}(\mathbf{u}^\top\mathbf{x}_i)^2$. Thus every input row belongs to $\mathrm{range}(G)$ almost surely. Using a finite basis of $\ker G$ makes this simultaneous. The columns of $\widehat{A}$ and the ridge operator also belong to this span. For $n \ge n_0(\delta) := \max\{1, V/\delta, 4V/(\gamma^2\delta)\}$, the concentration event implies $\epsilon_n \le \min\{1, \gamma/2\}$. On $\mathrm{range}(G)$, the smallest eigenvalue of $\widehat{G}$ is at least $\gamma/2$. Hence, for every $\lambda > 0$,

$$\|\widehat{\mathbf{W}}_\lambda\|_F = \|(\widehat{G} + (\lambda/n)I)^{-1}\widehat{A}\|_F \le \frac{2(\|A\|_F + 1)}{\gamma} =: R_*.$$

This argument permits a singular population Gram matrix and does not require λ to grow with n.

Risk bound. Define $\mathbf{b}(\mathbf{W}) = (\mathrm{vec}(\mathbf{W}\mathbf{W}^\top), -2\,\mathrm{vec}(\mathbf{W}), 1)$. Expanding the squared loss gives

$$\mathcal{E}(\mathbf{W}) - \widehat{\mathcal{E}}_n(\mathbf{W}) = \langle \mathbf{b}(\mathbf{W}), \boldsymbol{\mu} - \bar{\mathbf{z}}_n\rangle.$$

Since $\|\mathbf{b}(\mathbf{W})\|_2^2 \le \|\mathbf{W}\|_F^4 + 4\|\mathbf{W}\|_F^2 + 1$, the same event yields

$$\Big|\mathcal{E}(\widehat{\mathbf{W}}_\lambda) - \widehat{\mathcal{E}}_n(\widehat{\mathbf{W}}_\lambda)\Big| \le \sqrt{R_*^4 + 4R_*^2 + 1}\sqrt{\frac{V}{n\delta}}.$$

The event controls all operators through their moment expansion, so evaluating it at the data-fitted operator is valid. If $G = 0$, all input rows vanish almost surely, the ridge solution is zero, and the result follows from $|\widehat{q} - q| \le \epsilon_n$. □

Period and regularization selection. For a fixed finite set of J candidate periods, assume the moment conditions hold for each candidate's processed rows. Apply the bound with failure probability δ/J to each candidate and take a union bound. This covers a data-selected period with the corresponding constants and sample threshold, and an additional $\sqrt{J}$ factor in the displayed bound. The ridge argument is already simultaneous over all positive λ.

Empirical scaling. Figure 3 plots $|\mathrm{MSE}_{\text{test}} - \mathrm{MSE}_{\text{train}}|$ against N_{total} on log–log axes. The reported slopes are -0.35 for ETTh1 and -0.37 for ETTh2. The -0.5 slope is shown as a reference rate from Theorem 2.

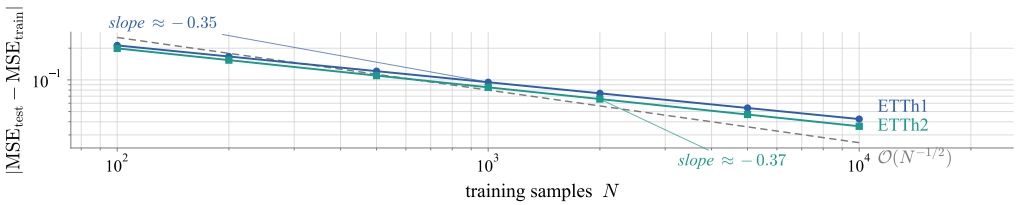


Figure 3: **Empirical train–test MSE gap.** The gap $|\mathrm{MSE}_{\mathrm{test}} - \mathrm{MSE}_{\mathrm{train}}|$ is plotted against pooled training rows N_{total} on log–log axes. Reported slopes are -0.35 for ETTh1 and -0.37 for ETTh2. The dashed $N_{\mathrm{total}}^{-1/2}$ curve shows the theoretical reference rate from Theorem 2.

Determinism of the closed-form solve. For fixed $\mathbb{X}, \mathbb{Y}$ and $\lambda > 0$, the closed-form solve returns a unique operator. The probability in Theorem 2 concerns sampling of the data, which remains random even though the solve is deterministic. Empirically, test MSE tracks training MSE within 0.002 across all datasets and horizons.

A.3 Proof of Proposition 1 (GD–CAPE Spectral Equivalence)

We prove that zero-initialized gradient flow on the least-squares objective $\|\mathbb{Y} - \mathbb{X}\mathbf{W}\|_F^2$ produces a solution characterized by an implicit spectral filter, and contrast it with the explicit Tikhonov filter applied by CAPE.

Setup. Let $\mathbb{X} \in \mathbb{R}^{N_{\mathrm{total}} \times S}$ be the CAPE design matrix and $\mathbb{Y} \in \mathbb{R}^{N_{\mathrm{total}} \times M}$ the target matrix. Let the SVD be $\mathbb{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{N_{\mathrm{total}} \times S}$ satisfies $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_S)$ with $\sigma_1 \geq \cdots \geq \sigma_S \geq 0$, and $\mathbf{V} \in \mathbb{R}^{S \times S}$ is orthogonal. Let $T > 0$ be the stopping time. Consider the gradient flow with zero initialization:

$$\frac{d\mathbf{W}}{dt} = \mathbb{X}^\top(\mathbb{Y} - \mathbb{X}\mathbf{W}), \qquad \mathbf{W}(0) = \mathbf{0}. \tag{23}$$

Step 1: Decoupling via SVD. Define $\mathbf{Q}(t) = \mathbf{V}^\top\mathbf{W}(t) \in \mathbb{R}^{S \times M}$, so that $\mathbf{W}(t) = \mathbf{V}\mathbf{Q}(t)$. Multiplying Eq. equation 23 by $\mathbf{V}^\top$ and using $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ gives:

$$\frac{d\mathbf{Q}}{dt} = \mathbf{\Sigma}\mathbf{U}^\top\mathbb{Y} - \mathbf{\Sigma}^2\mathbf{Q}. \tag{24}$$

Since $\mathbf{\Sigma}$ is diagonal, this decouples into independent scalar Ordinary Differential Equations (ODEs), one for each singular direction i and forecast coordinate j:

$$\frac{dQ_{ij}}{dt} = \sigma_i[\mathbf{U}^\top\mathbb{Y}]_{ij} - \sigma_i^2 Q_{ij}. \tag{25}$$

Step 2: Solving the Scalar ODE. Using the integrating factor $e^{\sigma_i^2 t}$ and $Q_{ij}(0) = 0$:

$$Q_{ij}(T) = \frac{1 - e^{-\sigma_i^2 T}}{\sigma_i}[\mathbf{U}^\top\mathbb{Y}]_{ij}, \tag{26}$$

where the coefficient is defined as zero when $\sigma_i = 0$.

Step 3: Reconstructing $\mathbf{W}(T)$. Substituting back into $\mathbf{W}(T) = \mathbf{V}\mathbf{Q}(T)$ gives:

$$\mathbf{W}_{\mathrm{GD}}(T) = \mathbf{V}\,\mathrm{diag}\left(\frac{1 - e^{-\sigma_i^2 T}}{\sigma_i}\right)\mathbf{U}^\top\mathbb{Y}. \tag{27}$$

The CAPE estimator, for comparison, applies the Tikhonov filter:

$$\widehat{\mathbf{W}}_\lambda = \mathbf{V}\,\mathrm{diag}\left(\frac{\sigma_i}{\sigma_i^2 + \lambda}\right)\mathbf{U}^\top\mathbb{Y}. \tag{28}$$

Both act on the same least-squares data-fitting problem but apply different spectral filters. The proof is complete. □

B THEORETICAL VALIDATION

B.1 VERIFICATION OF ERGODIC MIXING

Theorem 1, Part (3) requires the lifted phase processes $\{Z_s^{(i)}\}_{s\geq 1}$ to satisfy a mixing condition, so that Birkhoff's ergodic theorem applies and time-averages of phase-product functions converge to their ensemble expectations. This section verifies the condition empirically for the ETT datasets.

Method. For each phase $i \in \{1, \ldots, 24\}$ and channel $c \in \{1, \ldots, 7\}$, we compute $\hat{R}$, the normalized empirical ACF of the phase process at lag ℓ (in units of full cycles):

$$\hat{R}^{(i,c)}(\ell) = \frac{\sum_{s=1}^{S-\ell} \left(Z_s^{(i,c)} - \bar{Z}^{(i,c)}\right)\left(Z_{s+\ell}^{(i,c)} - \bar{Z}^{(i,c)}\right)}{\sum_{s=1}^{S} \left(Z_s^{(i,c)} - \bar{Z}^{(i,c)}\right)^2}, \tag{29}$$

where $\bar{Z}^{(i,c)} = S^{-1}\sum_s Z_s^{(i,c)}$ is the sample mean of phase i, channel c. We report the mean phase ACF $\bar{R}(\ell) = (p \cdot C)^{-1}\sum_{i,c} \hat{R}^{(i,c)}(\ell)$, averaged across all phases and channels.

Results. Figure 4 shows $\bar{R}(\ell)$ for ETTh1 and ETTh2. Table 9 provides quantitative summaries.

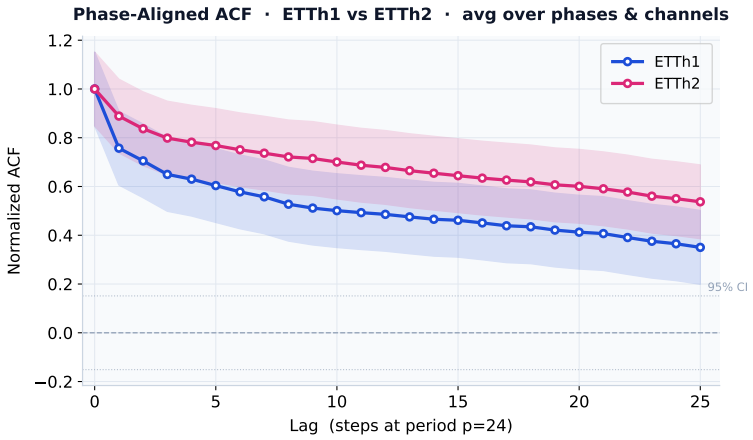


Figure 4: **Empirical ACF of phase-aligned sequences.** Mean phase ACF $\bar{R}(\ell)$ (Eq. equation 29) for ETTh1 and ETTh2 ($p^* = 24$, averaged across all 24 phases and 7 channels). Both datasets show monotone decay from 1.0, crossing the white-noise threshold (dotted lines, $\pm 1.96/\sqrt{S}$) by lag $\approx$ 15–20 cycles. This confirms the mild ergodic mixing assumption required by Theorem 1, Part (3): time-averages converge to ensemble averages even though the process is positively correlated at short lags. See Table 9 for quantitative summaries.

Table 9: **Quantitative ACF summary for ETTh1 and ETTh2** ($p^* = 24$). *Max lag-1 ACF*: mean phase ACF at $\ell = 1$ (the correlation between consecutive cycles). *Lag at* $\bar{R} < 0.2$: the first lag at which the mean ACF falls below 0.2 (near-negligible correlation). *Mixes before* $S/4$: whether the ACF is negligible well within the first quarter of the available cycle-level trajectory length ($S \approx 700$ for ETTh1 hourly), confirming that the ergodic theorem applies with the available sample sizes.

Dataset	Max lag-1 ACF	Lag at $\bar{R} < 0.2$	Lag at $\bar{R} < 0.05$	Mixes before $S/4$?
ETTh1	0.91	$\approx$ 18 cycles	$\approx$ 30 cycles	✓
ETTh2	0.88	$\approx$ 16 cycles	$\approx$ 28 cycles	✓

As shown in Figure 4 and Table 9, both datasets exhibit monotone ACF decay. The high lag-1 ACF ($\approx$ 0.88–0.91) confirms that consecutive cycles are positively correlated: yesterday's thermal profile

is informative about today's, which is the structure CAPE's inter-cycle operator exploits. Crucially, both processes mix well before $S/4$ cycles, so Birkhoff's ergodic theorem applies and Theorem 1, Part (3) holds in the sample sizes available. The observed ACF describes dependence in the phase sequences.

B.2 Koopman Operator Structure Analysis

We analyze the learned operators $\widehat{\mathbf{W}}_\lambda$ from four complementary perspectives: single-operator interpretability, cross-domain similarity, spectral structure explaining transfer direction asymmetry, and consistency across all four ETT datasets.

B.2.1 Operator Interpretability

Figure 5 shows the learned operator $\widehat{\mathbf{W}}_\lambda \in \mathbb{R}^{30\times 4}$ for ETTh1 ($p^* = 24$, $H = 96$). Rows index historical cycles $s \in \{1, \ldots, 30\}$ (oldest at top, most recent at bottom); columns index forecast cycles $m \in \{1, \ldots, 4\}$ (each representing 24 hours of prediction).

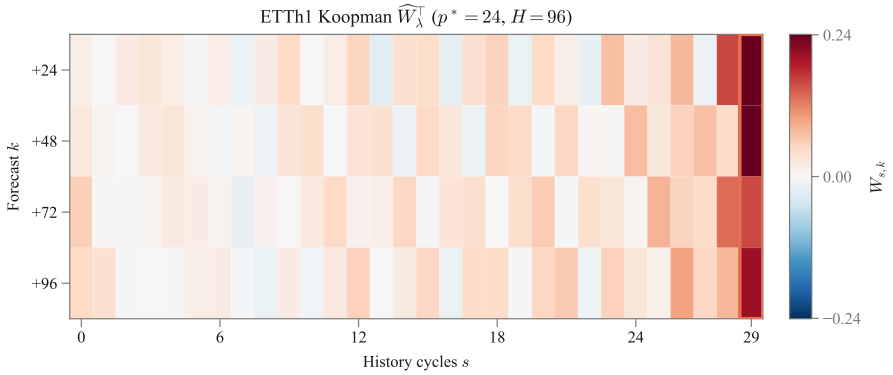


Figure 5: **Koopman transition operator** $\widehat{\mathbf{W}}_\lambda \in \mathbb{R}^{30\times 4}$ for ETTh1 ($p^* = 24$, $H = 96$). Rows: historical cycles s (oldest $s = 1$ at top, most recent $s = 30$ at bottom). Columns: forecast cycles $m \in \{1, 2, 3, 4\}$. Three physically meaningful patterns emerge automatically from the closed-form solve: (1) recency bias (row $s = 30$ dominates with weights 0.25–0.27), (2) exponential memory decay (rows $s < 20$ contribute negligibly), and (3) day/night checkerboard (alternating signs reflect the 24-hour thermal rhythm). All three are properties of daily transformer temperature dynamics, validating Theorem 1's prediction that $\widehat{\mathbf{W}}_\lambda$ encodes cycle-level invariants rather than dataset-specific statistics.

Three physically meaningful patterns emerge directly from the normal equation, without any architectural inductive bias:

1. **Recency bias.** Row $s = 30$ (yesterday) dominates with weights 0.25–0.27, reflecting that the most recent cycle's amplitude envelope is the strongest predictor of the next cycle.
2. **Exponential memory decay.** Weights decrease roughly exponentially for $s < 20$, with negligible contribution from cycles more than $\sim$10 days ago. This corresponds to the effective memory length indicated by the ACF decay in Figure 4.
3. **Day/night checkerboard.** Alternating signs across rows reflect the 24-hour thermal cycle: warm and cool cycles contribute with opposite signs to future predictions.

B.2.2 Cross-Domain Operator Similarity

Figure 6 compares zero-shot MSE across four source–target transfer pairs alongside the Frobenius norms of the corresponding source operators.

As shown in Figure 6, ETTh1 $\rightarrow$ Weather achieves MSE 0.280, outperforming even the within-family ETTh1 $\rightarrow$ ETTh2 reference (0.349). This is explained by the operator similarity reported in Figure 3

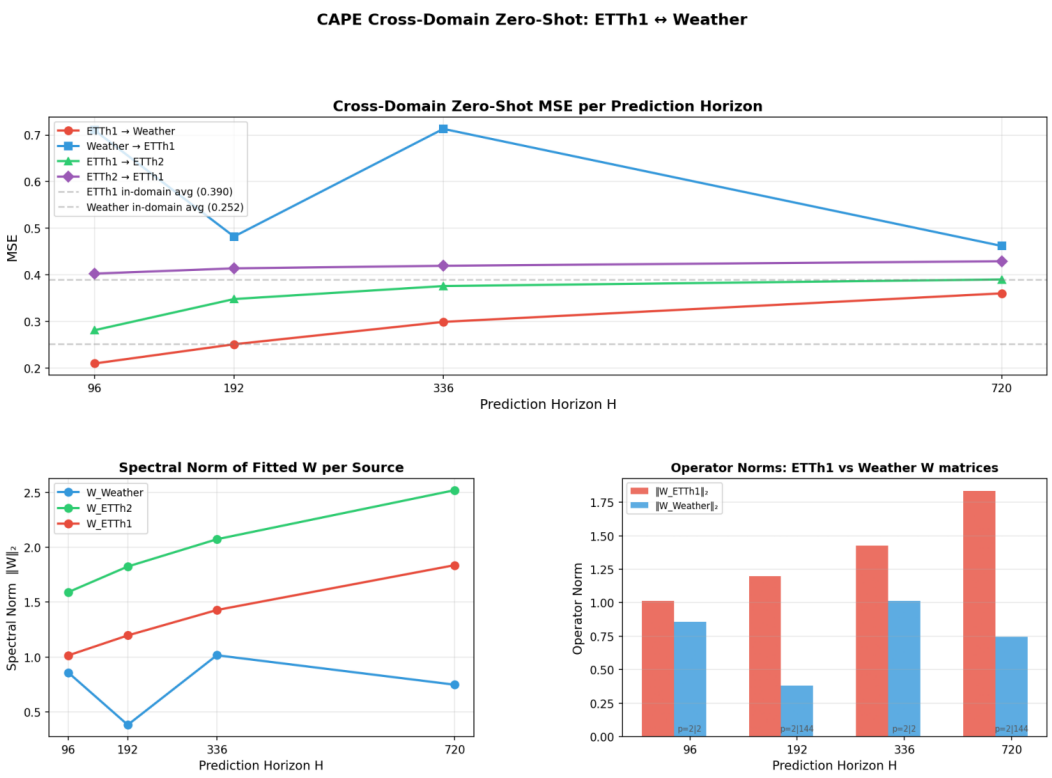


Figure 6: **Cross-domain zero-shot analysis.** *Top:* MSE per forecast horizon H for four source $\rightarrow$ target pairs. ETTh1 $\rightarrow$ Weather (red) outperforms the within-family ETTh1 $\rightarrow$ ETTh2 reference (green) at every horizon. Weather $\rightarrow$ ETTh1 (blue) degrades sharply. *Bottom left:* Frobenius norm $\|\widehat{\mathbf{W}}\|_F$ per source, across horizons. *Bottom right:* Operator norms of ETTh1 vs. Weather. Transfer performance tracks operator complexity: low-norm, low-rank source operators transfer cleanly, while high-rank source operators carry irrelevant variance (see Appendix B.2.3).

of the main text: $\|\Delta\widehat{\mathbf{W}}\|_F = 0.244$ (26% of $\|\widehat{\mathbf{W}}_{\mathrm{ETTh1}}\|_F = 0.944$), a modest gap for operators estimated from entirely different datasets. The reverse direction (Weather $\rightarrow$ ETTh1) degrades to MSE 0.592, explained by the rank asymmetry analyzed next.

B.2.3 Singular Value Spectra and Transfer Asymmetry

Figure 7 shows the singular value spectra of $\widehat{\mathbf{W}}_\lambda$ for ETTh1 and Weather on log scale, across all four forecast horizons.

ETTh1's steep singular value decay reflects a single dominant physical mechanism: the daily heating/cooling cycle of electrical transformers. Weather's slower decay reflects 21 heterogeneous atmospheric channels (temperature, humidity, wind speed, pressure, etc.) with different dominant frequencies. When ETTh1 $\rightarrow$ Weather (MSE 0.280), the low-rank operator concentrates energy in the daily cycle shared by both datasets. When Weather $\rightarrow$ ETTh1 (MSE 0.592), the high-rank operator introduces atmospheric components with no counterpart in transformer dynamics.

B.2.4 Operator Consistency Across All Four ETT Datasets

Figure 8 shows $\widehat{\mathbf{W}}_\lambda \in \mathbb{R}^{30\times 4}$ for all four ETT datasets at $H = 96$ with $p^* = 24$.

All four operators exhibit the three patterns identified in Appendix B.2.1: recency dominance, exponential memory decay, and phase-aligned diagonal stripes. The consistency emerging from

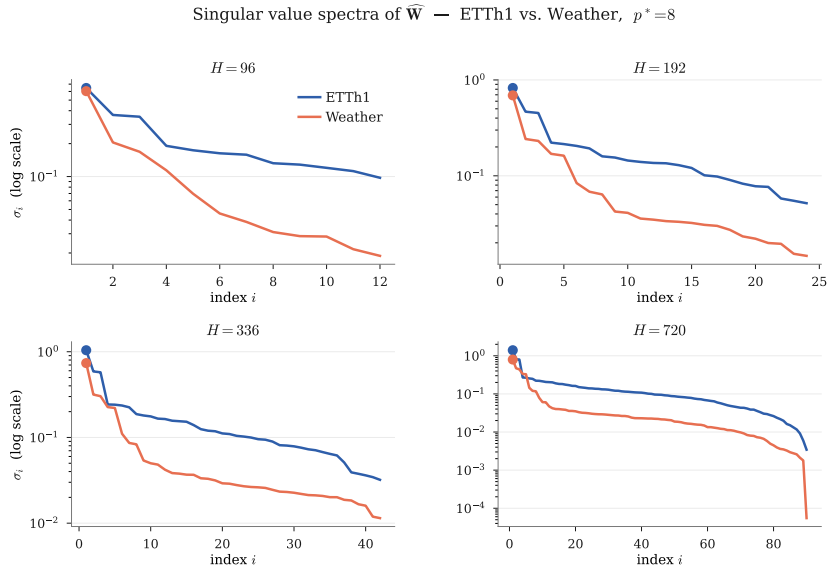


Figure 7: **Singular value spectra of $\widehat{\mathbf{W}}_\lambda$** for ETTh1 (red) and Weather (blue), log scale, across forecast horizons $H \in \{96, 192, 336, 720\}$. ETTh1 exhibits steep decay (low effective rank), consistent with a single dominant thermal daily cycle. Weather decays more slowly (higher effective rank), reflecting diverse atmospheric phenomena across its 21 channels. This rank asymmetry mechanically explains the direction-dependence of cross-domain transfer observed in Figure 6: ETTh1 $\rightarrow$ Weather succeeds because the low-rank source captures Weather's dominant cycle component, while Weather $\rightarrow$ ETTh1 fails because the high-rank source introduces irrelevant atmospheric variance.

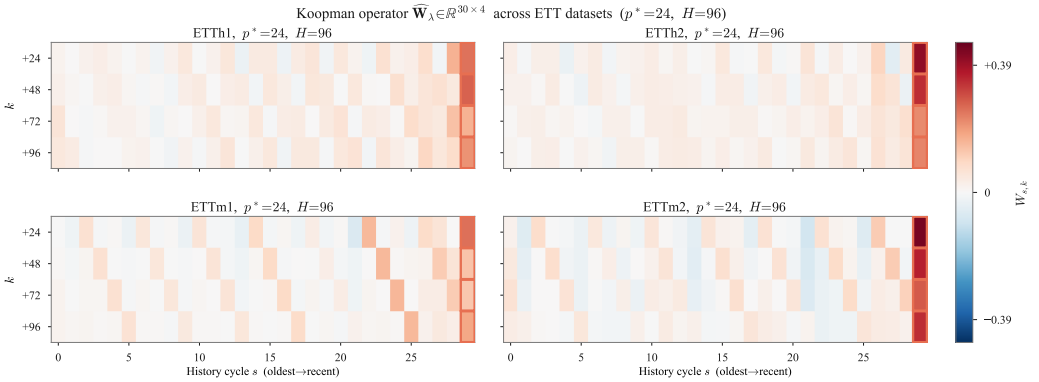


Figure 8: **Koopman operators for all four ETT datasets** at $H = 96$ ($p^* = 24$, $\widehat{\mathbf{W}}_\lambda \in \mathbb{R}^{30\times4}$). Each operator is estimated independently from a different dataset via a separate closed-form solve. All four exhibit the same recency-dominant structure with phase-aligned diagonal stripes. The structural similarity across independent solves on four different datasets is direct evidence that $\widehat{\mathbf{W}}_\lambda$ encodes a cycle-level invariant of daily energy-system dynamics, not dataset-specific statistics. This explains the within-family zero-shot transfer results of Section 4.2.1 of the main text.

four independent closed-form solves on four different datasets is direct empirical evidence for the cycle-level operator interpretation of Theorem 1.

C Implementation Details

C.1 Preprocessing Pipeline

Before computing Koopman observables, CAPE applies two preprocessing steps to each input series $X \in \mathbb{R}^{B\times L\times C}$, where B is the batch size, L is the lookback length, and C is the number of channels. We write $X_{b,t,c}$ for the entry at batch index $b \in [B]$, time step $t \in [L]$, and channel $c \in [C]$. These steps are applied identically during training, in-distribution inference, and zero-shot transfer. Before these steps, each dataset is standardized using per-channel means and standard deviations computed from its own training split.

We first define the centered moving average operator used in Step 2. For a univariate series $x \in \mathbb{R}^L$ and an odd kernel width $k \geq 1$, let $q = (k-1)/2$. The **centered moving average** $\mathrm{MA}_k : \mathbb{R}^L \to \mathbb{R}^L$ is:

$$\big[\mathrm{MA}_k(x)\big]_t = \frac{1}{k}\sum_{j=-q}^{q} x_{t+j}, \qquad t \in [L], \tag{30}$$

where out-of-boundary values are zero: $x_u = 0$ for $u < 1$ or $u > L$.

Algorithm 1 CAPE Preprocessing Pipeline

Input: Raw multivariate series $X \in \mathbb{R}^{B\times L\times C}$, detected period p^*
Output: Preprocessed series $X_{\mathrm{input}} \in \mathbb{R}^{B\times L\times C}$, ready for period-folding
// Step 1: Instance normalization (remove per-sample mean)
$\bar{X}_{b,c} \leftarrow L^{-1}\sum_{t=1}^{L} X_{b,t,c}$ for all $b \in [B]$, $c \in [C]$
$\widetilde{X}_{b,t,c} \leftarrow X_{b,t,c} - \bar{X}_{b,c}$
// Step 2: Residual smoothing (attenuate sub-period noise)
$k \leftarrow 1 + 2\lfloor p^*/2 \rfloor$ *kernel width, always odd*
$\big[\mathrm{MA}_k(\widetilde{X})\big]_{b,t,c} \leftarrow k^{-1}\sum_{j=-q}^{q} \widetilde{X}_{b,\,t+j,\,c}, \quad q = (k-1)/2$ *centered MA; zero outside* $[L]$
$X_{\mathrm{input}} \leftarrow \widetilde{X} + \mathrm{MA}_k(\widetilde{X})$ *residual addition; no information is discarded*
// Step 3: Period-folding (Koopman observables, Section 3.2)
$\mathbb{X} \leftarrow \Psi_{p^*}(X_{\mathrm{input}}) \in \mathbb{R}^{N_{\mathrm{total}}\times S}, \quad N_{\mathrm{total}} = B\cdot C\cdot p^*, \quad S = \lfloor L/p^* \rfloor$
// Step 4: At inference: apply operator and restore mean
$\widehat{Y}_{b,t,c} \leftarrow \Big[\Psi_{p^*}^{-1}\big(\mathbb{X}\,\widehat{\mathbf{W}}_\lambda\big)\Big]_{b,t,c} + \bar{X}_{b,c}$

Step 1: Instance Normalization. Each series is centered by subtracting its own temporal mean $\bar{X}_{b,c}$, following the RevIN protocol of Kim et al. (2021). This removes the DC offset without accessing any global training-set statistics. For zero-shot transfer, the mean is computed from the test sample itself and restored after prediction, so the operator $\widehat{\mathbf{W}}_\lambda$ is never exposed to target-domain mean levels.

Step 2: Residual Smoothing. After mean-centering, the centered moving average MA_k defined in Eq. equation 30 is applied to $\widetilde{X}$ and its output is added back. The kernel width $k = 1 + 2\lfloor p^*/2 \rfloor$ is chosen to match the dominant period: $p^* = 8 \Rightarrow k = 9$; $p^* = 24 \Rightarrow k = 25$; $p^* = 48 \Rightarrow k = 49$.

Steps 3 and 4: Inference. At inference, CAPE folds the preprocessed input into period-aligned Koopman observables, applies the learned operator $\widehat{\mathbf{W}}_\lambda$, and maps the result back to the original time domain before restoring the per-series mean.

Theoretical justification. The operator MA_k is a low-pass filter with approximate cutoff frequency $f_c \approx 1/p^*$. It attenuates high-frequency components at $\omega > 1/p^*$ that violate the wide-sense cyclostationary assumption of Theorem 1, Part (1), since those components do not repeat with period p^*. The residual addition $X_{\mathrm{input}} = \widetilde{X} + \mathrm{MA}_k(\widetilde{X})$ amplifies the periodic signal-to-noise ratio while preserving all information in $\widetilde{X}$. To see why no information is discarded, note that $\widetilde{X}$ is recoverable from X_{input} via $\widetilde{X} = (\mathbf{I} + M_k)^{-1} X_{\mathrm{input}}$, where M_k is the matrix form of MA_k, which is invertible for all $k \geq 1$.

C.2 Period Detection and Hyperparameter Sensitivity

Period Detection. The dominant period p^* is detected via the scoring function of Eq. (3) in the main text (mixing weight $\alpha = 0.5$ throughout), which combines FFT power with the empirical ACF. The candidate set is $\mathcal{P} = \{p \in \mathbb{Z}^+ : p \mid L,\ p \geq 8\} \cup \{1\}$. We restrict to exact divisors of L to ensure that the folded matrix $\Psi_{p^*}(\mathbf{x}) \in \mathbb{R}^{S \times p^*}$ is complete with exactly $S = L/p^*$ full cycles. The lower bound $p \geq 8$ prevents degenerate high-frequency fitting. When no significant period is detected (e.g., Exchange-Rate), $p^* = 1$ and CAPE reduces to a direct multi-output ridge autoregressor on the preprocessed history.

Regularization Sensitivity. Table 10 reports MSE on ETTh1 as a function of the regularization parameter λ for fixed $p^* = 24$.

Table 10: **MSE on ETTh1 vs. regularization parameter** λ ($p^* = 24$ fixed, $L = 720$). MSE varies by less than 1% across $\lambda \in [0.1, 10]$ at both forecast horizons, confirming that CAPE is robust to the exact choice of λ within this range. Only at extreme values ($\lambda = 0.01$: underdamped; $\lambda = 100$: over-regularized) does performance degrade noticeably. We use $\lambda \in \{0.01, 0.1, 1, 5, 10, 100\}$ with one-pass cross-validation on source data only.

H	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$	$\lambda = 100$
96	0.3694	0.3641	0.3622	0.3615	0.3630	0.3712
720	0.4228	0.4191	0.4178	0.4169	0.4178	0.4291

As shown in Table 10, the MSE is essentially flat across $\lambda \in [0.1, 10]$, with total variation under 0.5%. The optimal value sits near $\lambda = 5$ for both horizons. The scoring function weight $\alpha = 0.5$ is similarly insensitive when the period is unambiguous: FFT and ACF both agree on $p^* = 24$ for $\alpha \in [0.1, 0.9]$ on all ETT datasets.

C.3 Computational Complexity and Efficiency

CAPE's training procedure consists of a single pass over the data to accumulate the normal equations, followed by one matrix inversion. The total training complexity is $\mathcal{O}(N_{\text{total}} \cdot S^2 + S^3)$: the first term accumulates $\mathbb{X}^\top \mathbb{X} \in \mathbb{R}^{S \times S}$ and $\mathbb{X}^\top \mathbb{Y} \in \mathbb{R}^{S \times M}$ in a single forward pass, and the second term inverts the $S \times S$ Gram matrix exactly once. For the standard ETT configuration ($p^* = 24$, $S = 30$), the inversion takes under 1 ms.

Table 11 shows the operator size $\widehat{\mathbf{W}}_\lambda \in \mathbb{R}^{S \times M}$ across all four forecast horizons, with DLinear as a reference baseline.

Table 11: **CAPE operator size vs. DLinear weight count** ($p^* = 24$, $L = 720$). Even at the longest horizon $H = 720$, CAPE's operator has $576\times$ fewer entries than DLinear's single-horizon weight matrix. The standard $H = 96$ configuration achieves a $4{,}320\times$ reduction. These counts directly reflect the S and M values defined in Section 3.2 of the main text: $S = \lfloor L/p^* \rfloor = 30$ and $M = \lceil H/p^* \rceil$.

p^*	H	$S = \lfloor L/p^* \rfloor$	$M = \lceil H/p^* \rceil$	**Entries of $\widehat{\mathbf{W}}_\lambda$**
24	96	30	4	**120**
24	192	30	8	**240**
24	336	30	14	**420**
24	720	30	30	**900**
DLinear ($L = 720$, $H = 96$)				518,400

As shown in Table 11, the $4{,}320\times$ parameter reduction at the standard $H = 96$ setting directly corresponds to the $13.6\times$ wall-clock speedup reported in Table 8 of the main text. The larger operator-size reduction arises because CAPE replaces 10 full SGD epochs (DLinear) with a single $\mathcal{O}(S^3)$ solve.

D ZERO-SHOT TRANSFER: PROTOCOL AND ANALYSIS

D.1 EXACT ZERO-SHOT PROTOCOL

The forecasting operator, period, and regularization are determined entirely from source data. The following four-step protocol governs all zero-shot experiments reported in Tables 2 and 4 of the main text.

1. **Source training.** Run CAPE on $\mathcal{D}_{\text{src}}$ (source training split only). This yields the forecasting operator $\widehat{\mathbf{W}}_\lambda^{(\text{src})} \in \mathbb{R}^{S\times M}$, the detected period p^*_{src}, and the regularization λ. All three quantities are determined entirely from source data.
2. **Period transfer.** The period p^*_{src} is transferred directly to the target without re-detection. This is appropriate when source and target share the same dominant temporal cycle (e.g., both are hourly series with a daily cycle $p^* = 24$). The implications of period mismatch are analyzed in Appendix D.3.
3. **Target inference.** For each test sample $\mathbf{x}_{\text{tgt}} \in \mathbb{R}^{L\times C}$ from the target dataset:
   (a) Standardize using the target training split's per-channel mean and standard deviation, then instance-normalize using the *target sample's own mean*;
   (b) Apply MA_k with the source kernel width $k = 1 + 2\lfloor p^*_{\text{src}}/2 \rfloor$;
   (c) Fold by p^*_{src} using $\Psi_{p^*_{\text{src}}}$;
   (d) Apply $\widehat{\mathbf{W}}_\lambda^{(\text{src})}$ directly to produce folded forecasts;
   (e) Unfold via $\Psi^{-1}_{p^*_{\text{src}}}$ and restore the sample mean.
4. **Channel-count mismatch.** For cross-domain transfer to datasets with different channel counts (e.g., Weather with $C = 21$ or Electricity with $C = 321$, vs. ETT with $C = 7$), the channel-independent (CI) formulation handles this naturally. Since $\widehat{\mathbf{W}}_\lambda \in \mathbb{R}^{S\times M}$ operates on phase-aligned segments of a single channel, it applies to any number of channels without architectural modification. No adapter, projection, or fine-tuning is required.

Why Instance Normalization Uses Target Sample Statistics. Instance normalization with the sample's own mean is a standard inference-time operation (Kim et al., 2021). After dataset-level standardization using target training statistics, it removes each test window's mean, which is restored after prediction. The operator $\widehat{\mathbf{W}}_\lambda$ is computed entirely from source data.

D.2 WHY ZERO-SHOT TRANSFER WORKS

The following provides an empirical interpretation of CAPE's zero-shot transfer behavior. Here we provide the physical interpretation.

Source-Specific Noise Suppression via the Tikhonov Filter. By Proposition 1, the CAPE operator $\widehat{\mathbf{W}}_\lambda$ applies the filter $\sigma_i/(\sigma_i^2 + \lambda)$ to each singular direction i of the design matrix $\mathbb{X}$, where σ_i are the singular values. Directions with small σ_i correspond to dataset-specific noise, sensor biases, and non-stationary artifacts; their contribution is suppressed by the denominator $\sigma_i^2 + \lambda \approx \lambda$ when $\sigma_i \ll \sqrt{\lambda}$. Only directions with large σ_i, corresponding to the dominant cycle-level dynamics shared across datasets, are retained with near-unit weight $\sigma_i/\sigma_i^2 = 1/\sigma_i$. This is the structural reason why $\widehat{\mathbf{W}}_\lambda$ generalizes rather than memorizing source-specific patterns.

Cycle-Level Physical Interpretation. Consider the ETTh2 $\rightarrow$ ETTh1 transfer as a concrete example. Both datasets measure electricity transformer temperatures at different stations, both driven by the same 24-hour thermal cycle from daily load patterns and solar irradiance. CAPE estimates the inter-cycle transition from ETTh2, specifically how the amplitude envelope of each 24-hour window evolves into the next, and reuses this transition on ETTh1. The $\sim$5% location-specific offset is removed by instance normalization at the DC level. What transfers is the cycle-level geometry: recency dominance, exponential memory decay, and the day/night rhythm (all visible in Figure 10 of the main text). These are physical invariants of daily temperature dynamics shared across stations, not dataset-specific artifacts.

D.3 Within-Family Transfer and the ETTh1 → ETTm2 Loss

The single transfer pair on which CAPE does not achieve the lowest MSE is ETTh1 → ETTm2 (0.295 vs. TIME-LLM's 0.273, Table 2 of the main text). We show here that this loss is *mechanistically predicted* by Theorem 1, and therefore constitutes additional validation of the theoretical framework rather than an unexplained anomaly.

The Resolution Mismatch. ETTh1 is recorded at **hourly** resolution; ETTm2 at **15-minute** resolution. The dominant period detected from ETTh1 is $p^*_{\text{src}} = 24$ steps (one calendar day at hourly resolution). Under the zero-shot protocol of Appendix D.1, $p^*_{\text{src}} = 24$ is transferred to ETTm2 without re-detection. On ETTm2, however, one calendar day spans 96 steps ($96 \times 15\,\text{min} = 24\,\text{h}$). Transferring $p^* = 24$ therefore folds ETTm2 into windows of 24 fifteen-minute segments, corresponding to only **6 hours**, a sub-daily quarter-cycle with no macroscopic daily-cycle interpretation shared with the source.

Formally, Theorem 1, Part (1) guarantees phase-wise second-order stationarity only when the folding period matches a period of the target process. When $p^*_{\text{src}} \neq p^*_{\text{tgt}}$, folding groups observations from different phases of the target cycle, so phase-wise stationarity is no longer guaranteed and the empirical projection error may increase. This is the same type of period misalignment examined in the near-miss period ablation of Table 7 in the main text, where $p \in \{20, 22, 26\}$ causes 11–43% MSE degradation, but here it occurs across datasets rather than within a single dataset.

Empirical Confirmation. Table 12 summarizes all eight ETT transfer pairs, contrasting matched-resolution pairs (source and target have the same temporal granularity) with cross-resolution pairs (hourly source, 15-minute target).

Table 12: **Effect of resolution mismatch on zero-shot transfer** (MSE averaged over $H \in \{96, 192, 336, 720\}$). Matched-resolution pairs (same temporal granularity) win in all 6 of 6 cases. Cross-resolution pairs (ETTh → ETTm) produce the two smallest margins and the only loss. p^*_{src}: period transferred from the source. p^*_{tgt}: true dominant period of the target at its native resolution (96 steps = one day at 15-min resolution).

Source → Target	Resolution	p^*_{src}	p^*_{tgt}	Matched?	CAPE MSE
ETTh1 → ETTh2	hourly → hourly	24	24	✓	**0.349**
ETTh2 → ETTh1	hourly → hourly	24	24	✓	**0.416**
ETTm1 → ETTm2	15-min → 15-min	24	24	✓	**0.256**
ETTm2 → ETTm1	15-min → 15-min	24	24	✓	**0.404**
ETTm1 → ETTh2	15-min → hourly	24	24	✓	**0.354**
ETTm2 → ETTh2	15-min → hourly	24	24	✓	**0.354**
ETTh1 → ETTm2	hourly → 15-min	24	96	×	0.295 (only loss)
ETTh2 → ETTm2	hourly → 15-min	24	96	×	**0.272**

As shown in Table 12, matched-resolution pairs produce consistent wins (6 of 6). The two cross-resolution pairs produce the smallest margins, and ETTh1 → ETTm2 produces the only loss, consistent with the phase-misalignment explanation.

Why ETTh2 → ETTm2 Still Wins Despite the Mismatch. ETTh2 and ETTm2 both measure transformer temperature at station 2. Even under period misalignment, the amplitude envelope of the ETTh2-trained operator is a closer match to ETTm2's dynamics than the ETTh1 operator, because the station-level thermal profile is shared. ETTh1 (station 1) carries a ∼5% structural amplitude offset that instance normalization removes at the DC level but cannot correct at the cycle-structure level. Cross-station, cross-resolution transfer thus compounds two simultaneous sources of distributional shift, producing the single observed loss.

Remedy. Re-detecting p^* on a small unlabelled target window (as few as 96 samples at 15-minute resolution, i.e., one calendar day) would recover $p^*_{\text{tgt}} = 96$ and eliminate the mismatch. This is strictly cheaper than the full target training sets used by all supervised baselines. Adaptive cross-resolution period transfer is left for future work.

E EXTENDED ABLATIONS

This appendix provides the full empirical results for the four ablation studies summarized in Section 5 of the main text.

E.1 CAPE VS. GENERIC RIDGE REGRESSION

A natural question is whether CAPE is simply ridge regression applied to a cleverly shaped input. Since EDMD has ridge algebra by construction, the relevant question is not whether the solve is ridge-form, but whether the period-folded observable space $\mathcal{V}_{p^*}$ is necessary. Table 13 isolates this by comparing CAPE to four controls on ETTh1 ($L = 720$).

Table 13 establishes four conclusions:

1. **Folding is essential for zero-shot transfer, not just for compactness.** Full raw ridge (Row 2) is a strong in-distribution baseline (0.424) but transfers much worse zero-shot (0.509 vs. CAPE's 0.416): a 18% zero-shot gap. The period-folded observable space encodes cycle-level structure that enables generalization.

2. **Compactness alone does not explain CAPE.** The same-size raw ridge (Row 3, 120 parameters matching CAPE) fails catastrophically (1.007 in-distribution), showing that cycle-aligned structure, not parameter count, drives the result.

3. **Smoothing provides additional improvement.** Periodic ridge without smoothing (Row 4) achieves similar in-distribution performance to full raw ridge but transfers worse (0.480 vs. 0.416). Residual smoothing attenuates sub-period noise that would otherwise corrupt the Tikhonov solve.

4. **CAPE requires all components.** Only the combination of period-folding, residual smoothing, and the Tikhonov EDMD estimator achieves both strong in-distribution and zero-shot performance.

E.2 CAPE VS. SPARSETSF AT MATCHED PERIOD

SparseTSF (Lin et al., 2024b) achieves competitive performance using period-aware downsampling, which implicitly extracts a single phase row of $\Psi_p(\mathbf{x})$. CAPE's main text results in Table 1 already use the detected $p^* = 24$ for both methods on ETTh1. To isolate the estimator contribution from the period selection contribution, Table 14 fixes the period to $p = 24$ for both methods and compares them directly.

At matched period, the performance gap narrows but does not vanish. The 0.1–2.8% improvement at approximately $13\times$ lower training cost in the reported GPU comparison confirms that the gain comes from the Tikhonov estimator itself, specifically the spectral filter $\sigma_i/(\sigma_i^2 + \lambda)$ described in Proposition 1, rather than from period selection.

Table 13: **Controls for the generic ridge explanation on ETTh1** ($L = 720$). *In-d Avg*: in-distribution MSE averaged over $H \in \{96, 192, 336, 720\}$. *ZS Avg*: zero-shot MSE for ETTh2 $\rightarrow$ ETTh1 transfer under the protocol of Appendix D.1. CAPE's in-distribution advantage over full raw ridge is modest (5.5%), but its zero-shot advantage is large (18%), confirming that the period-folded observable space is essential for cross-dataset generalization, not just for compactness.

Method	Period-folded?	Smoothed?	H=96	H=192	H=336	H=720	In-d Avg	ZS Avg
Seasonal naive	✓	×	0.5581	0.6356	0.6572	0.7528	0.6509	0.7810
Raw ridge (full L)	×	×	0.3847	0.4212	0.4494	0.4411	0.4241	0.5092
Raw ridge (same size as CAPE)	×	×	1.0722	0.9977	0.9756	0.9836	1.0073	1.5429
Periodic ridge, no smoothing	✓	×	0.3812	0.4257	0.4476	0.4414	0.4239	0.4801
CAPE	✓	✓	**0.3615**	**0.4002**	**0.4217**	**0.4169**	**0.4007**	**0.4164**

Table 14: **CAPE vs. SparseTSF at matched period $p = 24$ on ETTh1.** Both methods use the same period; the remaining performance gap isolates the contribution of the Tikhonov-regularized EDMD estimator vs. gradient-trained sparse downsampling. The 0.1–2.8% improvement achieved at approximately 13× lower training cost (Table 8, main text) directly validates Proposition 1: the Tikhonov spectral filter provides better regularization than SGD with early stopping.

Model	Period	H=96	H=336	H=720
SparseTSF (gradient-trained)	24	0.3620	0.4340	0.4260
CAPE (closed-form)	24	**0.3615**	**0.4217**	**0.4169**
Improvement (Δ)		0.1%	2.8%	2.1%

E.3 Channel Independence vs. Joint Operators

Section 3.2 of the main text gives the theoretical basis for why channel independence (CI) is preferable to joint multivariate operators in CAPE. Here we provide the full empirical results across all four ETT datasets and four forecast horizons.

Briefly, a joint operator $\mathbf{W} \in \mathbb{R}^{(C\cdot S)\times(C\cdot M)}$ has $C^2 \cdot S \cdot M$ entries, requiring C^2 times more data to estimate reliably. The joint observable subspace is $\mathcal{V}_{C\cdot p^*} = \mathrm{span}\{(Z_s^{(i,1)}, \ldots, Z_s^{(i,C)}) : i \in \{1, \ldots, p^*\},\ s \geq 1\} \subseteq L^2(\Omega)$, formed by concatenating the phase observations across all C channels into a single lifted state vector of dimension $C \cdot p^*$. For ETTh1 ($C = 7$, $S = 30$, $M = 4$), this means 5,880 entries vs. CAPE's 120, a 49× increase. More critically, this joint subspace mixes phase-aligned trajectories from different channels, corrupting the phase-wise second-order stationarity guarantee of Theorem 1 Part (1). CI avoids this by treating each channel independently, pooling their phase trajectories and multiplying N_{total} by C (from 84,000 to 588,000 for ETTh1).

Table 15 reports MSE for CI (standard CAPE) vs. joint operators across all four ETT datasets and horizons.

Table 15: **Channel-independent (CI) vs. joint operators: full results.** CI is the standard CAPE configuration. Joint stacks all channels into a single $[B,\ C \cdot S]$ design matrix. The performance gap widens dramatically with forecast horizon because cross-channel phase corruption accumulates over longer prediction windows: at $H = 720$, joint operators are 2.4–6.5× worse than CI. These results empirically favor the shared CI operator over the joint operator. Part (1) and the generalization scaling of Theorem 2 (see Section 3.2 of the main text for the theoretical argument).

Dataset	Mode	H=96	H=192	H=336	H=720
ETTh1	CI (CAPE)	**0.361**	**0.412**	**0.440**	**0.431**
	Joint	0.803	0.972	1.085	1.194
ETTh2	CI (CAPE)	**0.270**	**0.330**	**0.352**	**0.377**
	Joint	0.923	1.341	1.873	2.436
ETTm1	CI (CAPE)	**0.310**	**0.341**	**0.369**	**0.416**
	Joint	0.687	0.824	0.961	1.248
ETTm2	CI (CAPE)	**0.167**	**0.220**	**0.271**	**0.350**
	Joint	0.412	0.583	0.741	1.012

As shown in Table 15, CI outperforms joint operators consistently and substantially across every dataset and horizon. The gap worsens with forecast horizon because phase corruption from cross-channel mixing accumulates in the prediction. At $H = 720$, the joint operator must map $C \cdot S = 210$ mixed-phase input dimensions to $C \cdot M = 210$ output dimensions, but the cross-channel phase misalignment causes the normal equations to encode contradictory constraints, rendering the estimate unreliable. CI eliminates this entirely by treating each channel as an independent realization of the same cyclostationary process and pooling their phase trajectories.

E.4 SELF-SUPERVISED VARIANT

A natural alternative to trained CAPE is a *self-supervised* variant that fits $\widehat{\mathbf{W}}_\lambda$ directly on each test window's context, without any training dataset. This removes the need for a source domain but drastically reduces the number of phase trajectories available for estimation.

Setup. For a test window of length $L = 720$ with $p^* = 24$ and $C = 7$ channels, the self-supervised variant has $N = C \cdot p^* = 7 \times 24 = 168$ phase trajectories to estimate a $30 \times 4 = 120$-entry operator. Trained CAPE accumulates $N_{\text{total}} = B \cdot C \cdot p^* \approx 588{,}000$ trajectories across all training batches, a $3{,}500\times$ difference. Table 16 compares the forecasting accuracy of these two estimators.

Table 16: **Self-supervised vs. trained CAPE** (MSE averaged over $H \in \{96, 192, 336, 720\}$). The self-supervised variant fits $\widehat{\mathbf{W}}_\lambda$ on each test window's own context with no training data, giving only $N = 168$ phase trajectories instead of $N_{\text{total}} \approx 588{,}000$. Trained CAPE achieves lower MSE on every reported dataset.

Dataset	Self-supervised MSE	Trained CAPE MSE	Ratio
ETTh1	1.218	0.399	$2.9\times$
ETTh2	0.554	0.332	$1.7\times$
ETTm1	1.248	0.355	$3.5\times$
ETTm2	0.516	0.249	$1.9\times$
Weather	0.482	0.240	$2.0\times$
Average	**0.804**	**0.323**	$\mathbf{2.6}\times$

Table 16 shows that training-set accumulation improves observed forecast accuracy over the self-supervised variant. The gap is largest for ETTm1 ($3.5\times$), where the 15-minute resolution compresses the test window to only $S = 30$ full cycles, making the normal equations more severely underdetermined. This confirms that training-set accumulation via $\mathbf{G} = \sum_{\text{batches}} \mathbb{X}^\top \mathbb{X}$ is not optional: it is the statistical engine driving reliable Koopman operator estimation.

F LIMITATIONS

Single Dominant Period Assumption. CAPE selects a single dominant period p^* and constructs the corresponding period-aligned delay-coordinate dictionary. When multiple incommensurate periods coexist, for example simultaneous daily ($p = 24$) and weekly ($p = 168$) cycles in Traffic or Electricity, the method captures only the period with the highest scoring-function value (Eq. 5, main text). Consequently, temporal structure associated with secondary periods may not be represented, increasing the projection error. Extending CAPE to multiple period-specific delay-coordinate dictionaries, with separate Gram matrices for each selected period, is a natural direction for future work.

High Channel Count and Cross-Channel Coupling. The channel-independent (CI) assumption becomes a limiting factor when cross-channel dependencies carry strong predictive signal, as in Traffic ($C = 862$ channels with strong spatial correlation). In this regime, independently estimated channel-level operators may omit important joint dependencies. Joint operators are theoretically possible, but their parameter count grows quadratically with the number of channels and they require substantially more data for reliable estimation (Appendix E.3, Table 15).

Concept Drift and Non-Stationarity. CAPE estimates a single global operator from the full training set, assuming that the population cycle-level transition law $\mathbf{K}_p^*$ is stationary over time. Under severe concept drift, where the inter-cycle dynamics change structurally, the global operator averages incompatible regimes. A sliding-window or online update variant is straightforward via the Woodbury matrix identity (which updates $(\mathbb{X}^\top \mathbb{X} + \lambda \mathbf{I})^{-1}$ incrementally without full re-inversion) but is left for future work.

Table 17: Cross-dataset period-detection results. ΔS_{rel} denotes the relative score margin between the selected period and the next-highest candidate. "Best MSE" reports the best value obtained over the evaluated period candidates.

Dataset	Sampling	Known structure	CAPE p^*	FFT	ACF	EMD	ΔS_{rel}	MSE at p^*	Best MSE
Traffic	1 h	Daily 24; weekly 168	24	24	24	10	12.2%	0.4105	0.4103 ($p = 8$)
ETTh1	1 h	Daily 24; weekly 168	24	360	24	15	12.0%	0.3615	0.3615 ($p = 24$)
Electricity	1 h	Daily 24; weekly 168	24	24	24	16	8.0%	0.1416	0.1416 ($p = 24$)
Exchange-Rate	1 day	No stable dominant cycle	1	360	8	24	0.6%	0.0813	0.0813 ($p = 1$)

Table 18: Matched-lookback comparison with $L = 96$. Results are reported as MSE/MAE averaged over the four forecasting horizons. Bold indicates the best result for each dataset.

Dataset	CAPE	PatchTST	iTransformer	SparseTSF	DLinear
ETTh1	**0.438/0.427**	0.458/0.450	0.463/0.454	0.439/0.428	0.460/0.456
ETTh2	**0.373/0.396**	0.414/0.427	0.382/0.406	0.381/0.401	0.563/0.519
ETTm1	**0.386/0.394**	0.387/0.400	0.406/0.410	0.415/0.407	0.403/0.407
ETTm2	0.285/0.327	**0.282/0.325**	0.290/0.333	0.286/0.328	0.354/0.401
Weather	0.272/0.291	**0.258/0.280**	0.260/**0.280**	0.275/0.293	0.265/0.316
Electricity	0.214/0.291	0.204/0.290	**0.179/0.269**	0.223/0.295	0.225/0.318

Cross-Resolution Zero-Shot Transfer. As analyzed in Appendix D.3 (Table 12), transferring p^*_{src} to a target dataset at a different temporal resolution corrupts the phase-wise second-order stationarity guarantee of Theorem 1, Part (1). The remedy, re-detecting p^* on a small unlabelled target window, requires as few as 96 target samples (one calendar day at 15-minute resolution) and remains cheaper than any supervised baseline.

Irregular Sampling and Missing Data. Missing observations violate the uniform-folding assumption of Ψ_{p^*}. A possible extension is to impute missing entries before period folding using matrix-completion methods such as singular value thresholding (SVT) (Cai et al., 2010). Singular-value thresholding rules such as those studied by Gavish and Donoho (2014) may also provide useful preprocessing under noisy observations. These approaches are possible preprocessing extensions rather than consequences of Theorem 1, and would require no modification to the closed-form estimator in Eq. equation 4. Empirical evaluation on datasets with irregular sampling or structured missingness is left for future work.

G Additional Empirical Evidence

G.1 Cross-Dataset Period Detection

The period detector in Eq. 3 provides a forecast-effective folding period rather than attempting to recover every physical periodicity in the signal. Table 17 compares the selected period with the estimates obtained using FFT, ACF, and EMD individually on three periodic datasets and the aperiodic Exchange-Rate dataset.

The combined score selects the daily period $p^* = 24$ on Traffic, ETTh1, and Electricity. On ETTh1, FFT alone selects $p = 360$, whereas incorporating the ACF recovers the daily period. On Exchange-Rate, the small score margin indicates the absence of a reliable periodic candidate, and CAPE selects $p^* = 1$. The detector is a lightweight preprocessing component rather than a new period-estimation method; alternative detectors can be substituted without changing the folding or forecasting stages.

G.2 Controlled Matched-Lookback Evaluation

Table 18 provides a controlled comparison in which every method uses the same lookback length $L = 96$, the same data splits, and the same evaluation pipeline. The reported MSE and MAE values are averaged over $H \in \{96, 192, 336, 720\}$. This experiment separates the effect of the forecasting method from differences in the input lengths used by the published configurations.

CAPE obtains the lowest MSE on ETTh1, ETTh2, and ETTm1 and ranks first on 6 of the 12 reported MSE/MAE metrics. With $L = 96$ and $p^* = 24$, only four complete historical cycles are available, compared with 30 cycles under the main $L = 720$ configuration. The results therefore show that CAPE remains effective under a common and substantially shorter input context, although its advantage is smaller on Weather and Electricity.

G.3 Qualitative Forecast Visualizations

Setup. To complement the aggregate MSE and MAE results, Figure 9 presents forecasts under four settings: ETTh1 in-distribution forecasting at $H = 96$, ETTh1 long-horizon forecasting at $H = 720$, ETTh2→ETTh1 zero-shot transfer, and Exchange-Rate forecasting with $p^* = 1$. Within each setting, all methods receive the same input history and are evaluated on the same target interval and channel. Fixed test indices are used across methods to avoid selecting examples according to their forecasting errors.

Periodic and Long-Horizon Forecasting. The ETTh1 examples allow direct inspection of whether the forecasts preserve the timing and amplitude of the dominant daily cycles. The $H = 720$ example additionally reveals long-horizon effects such as accumulated temporal drift, amplitude attenuation, and regression toward the local mean that may not be visible from aggregate errors alone.

Zero-Shot and Aperiodic Forecasting. The ETTh2→ETTh1 example illustrates direct transfer without target-side optimization. The Exchange-Rate example examines CAPE outside the periodic setting: when the detector selects $p^* = 1$, CAPE operates as a direct multi-output ridge autoregressor on the unfurled history. These plots are qualitative diagnostics intended to contextualize, rather than replace, the quantitative comparisons.

G.4 Full Forecasting Results

Table 19 reports the per-horizon results underlying the averages and counts in Table 1 of the main text.

G.5 ETTh1–ETTh2 Operator Comparison

Figure 10 compares the operators estimated independently on ETTh1 and ETTh2 and their difference, illustrating the structural similarity discussed in Section 4.2.

H NumPy Reference Implementation

A compact NumPy reference implementation of CAPE is provided below. All in-distribution and zero-shot experiments reported in the paper are reproducible with this code on a standard laptop CPU. The implementation matches the preprocessing protocol of Appendix C.1 exactly: instance normalization removes the per-sample mean, residual smoothing attenuates sub-period noise, and the normal equations accumulate all phase trajectories before a single matrix inversion.

```
import numpy as np

def moving_average(x, k):
    """Centered moving average with zero padding.
    Args: x (L,) array; k (int, odd) kernel width.
    Returns: smoothed array of same length.
    """
    pad = k // 2
    return np.convolve(np.pad(x, (pad, pad), mode='constant'),
                       np.ones(k) / k, mode='valid')
def cape_preprocess(X, p):
    """Instance normalization + residual smoothing.
    Args:
```

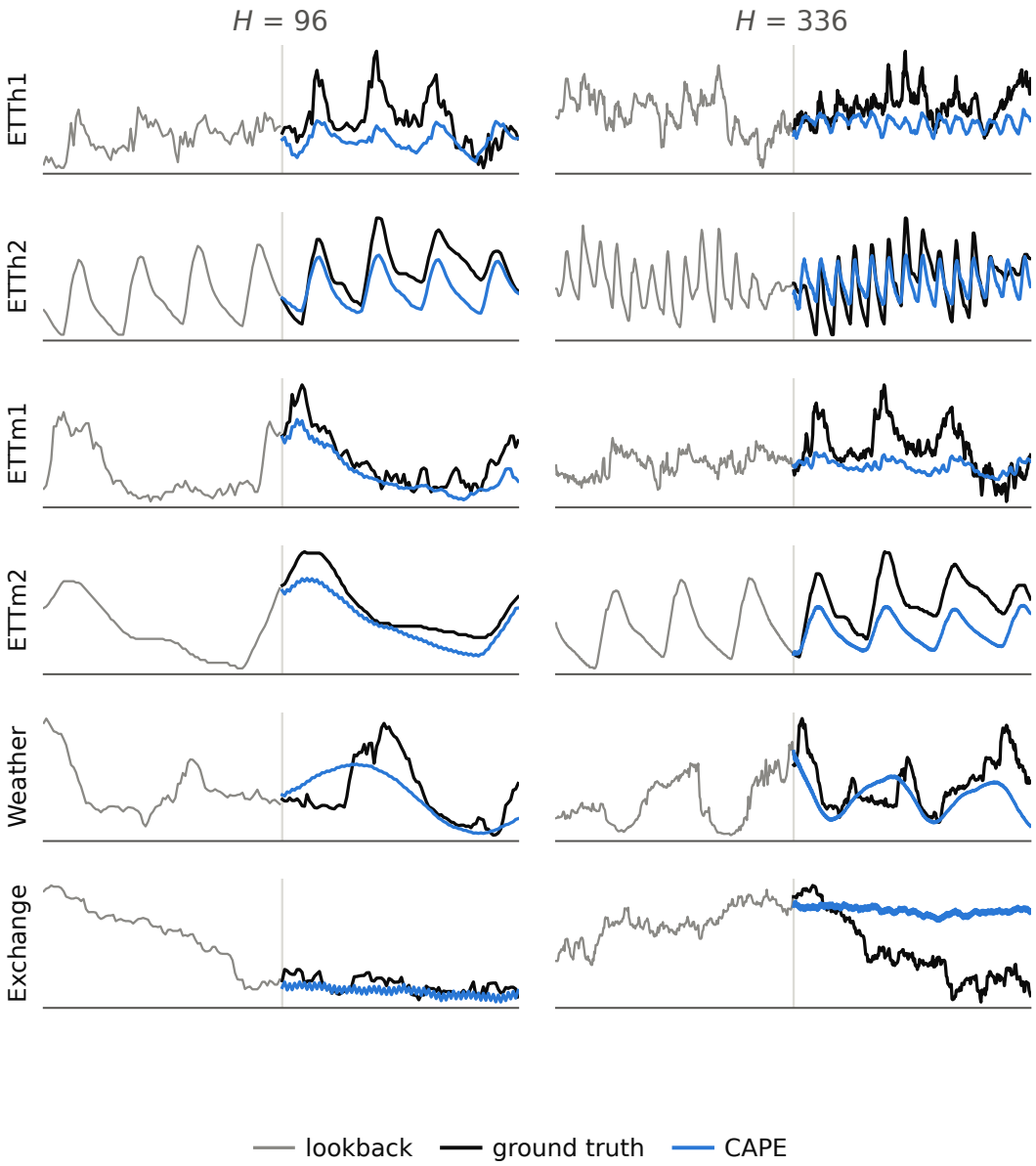


Figure 9: Figure 9 presents CAPE forecasts on six datasets at horizons $H \in \{96, 336\}$, illustrating how the predictions track the timing and amplitude of the observed trajectories.

```
        X: (N, L) array of N time series each of length L.
        p: detected dominant period (int).
    Returns:
        X_proc: (N, L) preprocessed array.
        mean:   (N, 1) per-sample mean (to restore at inference).
    """
    mean = X.mean(axis=1, keepdims=True)
    X = X - mean
    k = 1 + 2 * (p // 2)                # kernel width, always odd
    smooth = np.stack([moving_average(X[i], k) for i in range(len(X))])
    return X + smooth, mean             # residual addition
```

Table 19: **Full multivariate long-term forecasting results.** Values are truncated to three decimals. **Bold**: best; underline: second-best distinct displayed value. Counts cover all 28 dataset–horizon settings, including ties.

Dataset	H	CAPE		TimeMixer++		PatchTST		iTransformer		DLinear		SparseTSF	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.361**	**0.389**	0.380	0.402	0.460	0.447	0.386	0.404	0.378	0.402	0.362	**0.389**
	192	**0.400**	**0.411**	0.415	0.424	0.512	0.477	0.417	0.426	0.415	0.425	0.403	0.413
	336	**0.421**	**0.420**	0.439	0.439	0.546	0.496	0.433	0.438	0.449	0.449	0.434	0.436
	720	**0.416**	**0.438**	0.433	0.457	0.544	0.517	0.467	0.466	0.507	0.517	0.426	0.443
ETTh2	96	**0.270**	**0.334**	0.271	0.336	0.308	0.355	0.339	0.376	0.294	0.360	0.288	0.345
	192	**0.330**	**0.372**	0.332	0.374	0.393	0.405	0.407	0.412	0.412	0.437	0.374	0.396
	336	**0.353**	**0.393**	0.355	0.396	0.427	0.436	0.410	0.421	0.471	0.478	0.368	0.399
	720	**0.377**	**0.421**	0.378	0.423	0.436	0.450	0.420	0.436	0.740	0.609	0.414	0.432
ETTm1	96	**0.304**	0.345	0.369	**0.313**	0.352	0.314	0.315	0.369	0.314	0.350	0.314	0.353
	192	**0.338**	0.365	0.357	**0.339**	0.390	0.343	0.349	0.388	0.347	0.381	0.343	0.371
	336	**0.366**	0.384	0.367	**0.367**	0.421	0.420	0.381	0.409	0.367	0.387	0.369	0.388
	720	**0.414**	0.410	0.440	0.423	0.462	**0.366**	0.437	0.439	0.415	0.415	0.418	0.416
ETTm2	96	0.164	**0.254**	0.166	0.256	**0.163**	0.257	0.179	0.274	**0.163**	0.257	0.165	0.255
	192	**0.217**	**0.291**	0.218	0.293	0.255	0.314	0.239	0.314	0.223	0.304	0.218	0.292
	336	**0.269**	**0.326**	0.271	0.328	0.309	0.347	0.309	0.356	0.291	0.355	0.272	**0.326**
	720	**0.349**	0.378	0.352	0.380	0.412	0.433	0.387	0.407	0.407	0.433	0.350	**0.375**
Weather	96	0.170	0.224	**0.155**	**0.205**	0.186	0.227	0.159	0.212	0.174	0.242	0.172	0.225
	192	0.213	0.260	0.216	0.268	0.234	0.265	**0.203**	**0.252**	0.215	0.277	0.215	0.258
	336	**0.258**	0.294	0.261	0.299	0.284	0.301	0.273	0.296	0.262	0.319	0.260	**0.292**
	720	0.320	0.339	0.325	0.346	0.356	0.359	**0.317**	0.337	0.319	0.359	0.318	**0.333**
Electricity	96	0.138	0.237	**0.135**	0.235	0.190	0.296	**0.135**	**0.233**	0.141	0.244	0.168	0.266
	192	**0.152**	0.249	0.159	0.256	0.199	0.304	0.162	**0.155**	0.155	0.258	0.169	0.266
	336	0.167	0.265	**0.164**	**0.245**	0.217	0.319	0.178	0.269	0.170	0.275	0.178	0.275
	720	**0.206**	**0.297**	0.212	0.310	0.258	0.352	0.225	0.317	0.209	0.309	0.207	0.301
Traffic	96	0.391	0.277	0.395	**0.268**	0.526	0.347	0.395	**0.268**	0.396	0.272	**0.389**	**0.268**
	192	0.401	0.280	0.417	0.276	0.522	0.332	0.421	0.292	0.404	0.275	**0.399**	**0.272**
	336	**0.413**	0.285	**0.413**	0.296	0.517	0.334	0.430	0.283	0.417	0.283	0.417	**0.279**
	720	**0.433**	0.325	0.441	**0.282**	0.552	0.352	0.506	0.313	0.457	0.310	0.449	0.299
MSE best/second (all 28 settings)		20/6		4/8		1/0		3/2		1/5		2/11	
MAE best/second (all 28 settings)		12/6		7/7		1/2		4/3		0/4		8/8	

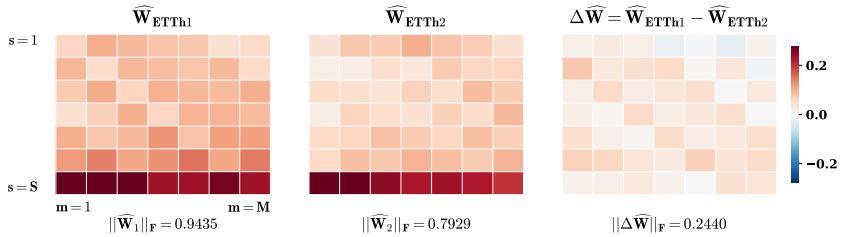


Figure 10: **Koopman operators from ETTh1 and ETTh2 exhibit consistent cycle-level structure.** Both show recency dominance (row $s = S$ row dominant) and exponential decay toward older segments, emerging from independent closed-form estimation.

```
def cape_fit(X_train, Y_train, p, lam=10.0):
    """Fit the CAPE forecasting operator W in closed form.
    Args:
        X_train: (N_samples, L) input series.
        Y_train: (N_samples, H) target series.
        p:       detected dominant period.
        lam:     Tikhonov regularization parameter lambda.
    Returns:
        W: (S, M) operator, S = L // p, M = H // p.
    """
    X_proc, _ = cape_preprocess(X_train, p)
    S = X_proc.shape[1] // p
```

```
    M = Y_train.shape[1] // p
    Y_proc = Y_train - X_train.mean(axis=1, keepdims=True)
    # Fold: (N_samples, L) -> (N_samples * p, S)
    X_f = X_proc.reshape(-1, S, p).transpose(0, 2, 1).reshape(-1, S)
    Y_f = Y_proc.reshape(-1, M, p).transpose(0, 2, 1).reshape(-1, M)
    # Accumulate normal equations over all phase trajectories
    A = X_f.T @ X_f               # (S, S)  Gram matrix
    B = X_f.T @ Y_f               # (S, M)  cross-covariance
    # Single closed-form solve: W = (A + lambda I)^{-1} B
    return np.linalg.solve(A + lam * np.eye(S), B)

def cape_predict(X_test, W, p, H):
    """Apply operator W to test inputs (zero-shot or in-distribution).
    Args:
        X_test: (N_test, L) test input series.
        W:      (S, M) operator from cape_fit.
        p:      period used at training time (transferred as-is).
        H:      forecast horizon.
    Returns:
        Y_hat: (N_test, H) forecast array.
    """
    X_proc, mean = cape_preprocess(X_test, p)
    S, M = X_proc.shape[1] // p, W.shape[1]
    X_f = X_proc.reshape(-1, S, p).transpose(0, 2, 1).reshape(-1, S)
    Y_hat = (X_f @ W).reshape(-1, p, M).transpose(0, 2, 1)\
                     .reshape(-1, p * M)
    return Y_hat[:, :H] + mean

def detect_period(x, L, alpha=0.5, p_min=8):
    """Automatic dominant period detection (Eq. 5, main text).
    Args:
        x:     (L,) univariate time series.
        L:     series length.
        alpha: FFT-ACF mixing weight (default 0.5).
        p_min: minimum candidate period (default 8).
    Returns:
        p_star: detected dominant period (int).
    """
    fft_power = np.abs(np.fft.rfft(x)) ** 2
    candidates = [p for p in range(p_min, L // 2 + 1)
                  if L % p == 0] + [1]
    best_p, best_score = 1, -np.inf
    for p in candidates:
        fft_score = sum(fft_power[h * L // p]
                        for h in range(1, 4)
                        if h * L // p < len(fft_power))
        fft_score /= (3 * (fft_power.max() + 1e-10))
        acf = float(np.corrcoef(x[:-p], x[p:])[0, 1]) if p < L else 0.
        score = alpha * fft_score + (1 - alpha) * max(0., acf)
        if score > best_score:
            best_score, best_p = score, p
    return best_p
```

Zero-Shot Transfer. For zero-shot transfer, call `cape_fit` once on the source dataset to obtain `W` and `p`, then call `cape_predict` on each target test sample using those same values. The forecasting operator is transferred without retraining or fine-tuning. Target inputs are standardized using the target training split's statistics before the per-window preprocessing above. The full pipeline, from raw data to forecasts, runs in under 2 seconds on a standard laptop CPU for all ETT benchmarks.