ROSKV: ROBUST SPECTRAL KEY-VALUE CACHE COMPRESSION FOR TIME-SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive time-series foundation models (TSFMs) enable zero-shot long-horizon forecasting, but real-world deployment on resource-constrained hardware is bottlenecked by a growing key–value (KV) cache. While caching past states eliminates quadratic recomputation, persistent memory scales linearly with look-back length and horizon. Furthermore, TSFM prefill importance is *delocalized in time* yet *localized in frequency*: seasonal cycles span many look-back tokens, so token eviction and uniform time-domain quantization are ill-matched, whereas a small fraction of RFFT bins carries most K/V energy. We present **ROSKV**, a memory-first, training-free spectral compressor for frozen TSFMs designed for memory-constrained deployments where persistent cache capacity is the primary barrier. RosKV (i) applies a fixed head-dimension rotation so NF4/NF2 block scales are not dominated by outliers on the few kept bins, (ii) transforms along the look-back with an RFFT, (iii) filters K and V separately by spectral energy, and (iv) stores adaptive mixed-precision coefficients, reconstructing an approximate dense prefill at every decode step before attention. Across nine datasets, four horizons, and four backbones (TimeMoE, Chronos, Moirai, TimesFM), RosKV consistently outperforms NLP token eviction (SnapKV, PyramidKV), coordinate quantization (KIVI), and frequency baselines (FreqKV), staying within $+3.6\%$ of Full-Cache aggregate MSE (with matching rankings confirmed on GIFT-Eval). On dense-token architectures where persistent KV storage forms a critical inference bottleneck (TimeMoE), RosKV slashes persistent cache memory by 55% on average (27.5 MB vs. 60.5 MB) and up to 78% at short horizons (9.5 MB vs. 42.5 MB), while maintaining competitive decoding latency (383 ms vs. No-Cache 810 ms).

1 INTRODUCTION

Long-horizon forecasting is vital across energy grids, industrial automation, finance, and edge-sensing infrastructure (Nie et al., 2022; Zeng et al., 2023). Time-series foundation models (TSFMs), including TimeMoE (Shi et al., 2025), Chronos (Ansari et al., 2024), Moirai (Woo et al., 2024), TimesFM (Das et al., 2023), and Timer (Liu et al., 2024a), deliver superior accuracy through pretraining on large corpora and zero-shot transfer across domains. However, deploying TSFMs on resource-constrained edge devices and real-time inference systems faces a severe memory wall. In edge deployments, high-bandwidth memory (DRAM/SRAM) is strictly limited and expensive, yet applications simultaneously demand fast autoregressive decoding. While maintaining an autoregressive Key–Value (KV) cache avoids quadratic historical recomputation across decode steps, the persistent KV cache scales linearly with both look-back length L and forecast horizon H. In dense-token architectures, at $L{=}512$ and $H{=}720$, caching states for a single TimeMoE-200M sequence persists over 90 MB of KV memory per batch element (> 900 MB at batch size 10). This footprint rapidly saturates on-device memory budgets, competing directly with model weights and severely bottlenecking edge inference throughput and serving concurrency.

Existing KV-cache compression methods largely originate from natural language processing (NLP) and fall into two primary paradigms: token eviction (Zhang et al., 2023; Li et al., 2024; Cai et al., 2024) and coordinate-wise quantization (Liu et al., 2024b; Hooper et al., 2024). However, transferring NLP-centric heuristics directly to TSFMs reveals a fundamental architectural mismatch. Unlike natural language, where attention concentrates around sparse semantic keywords or prompt sinks, time series are governed by multi-scale periodicity, macro seasonality, and short-term look-back

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.

fluctuations. Periodic cycles span dozens or hundreds of look-back timestamps, making importance inherently *delocalized in time*; pruning historical tokens breaks temporal phase coherence and destroys cyclical dependencies. Conversely, uniform coordinate quantization treats every timestamp equally, failing to exploit the extreme *spectral sparsity* (frequency concentration) of temporal signals while corrupting crucial recent look-back transitions with quantization noise.

Along the temporal dimension, KV energy is instead sharply *localized in frequency*: a small fraction of RFFT bins accounts for the vast majority of total key and value energy (e.g., 80–88% within the top 10% of bins on TimeMoE). This concentration suggests that compressing in the spectral domain—retaining only the dominant energy modes that govern broad seasonality and trend while discarding high-frequency noise—offers a far more natural representation. Existing frequency-domain methods for language models do not address the unique outlier characteristics of time-series attention states, while end-to-end frequency-based architectures (Zhou et al., 2022; Yi et al., 2023; **?**; Wu et al., 2022) require retraining from scratch rather than compressing an already-deployed, frozen foundation model.

Once high-energy modes are selected via top-k truncation, nearly all critical context is concentrated within a sparse set of d-dimensional spectral coefficient vectors. In low-bit quantization regimes (such as NF4/NF2), shared block scales along the head dimension become vulnerable to localized coordinate outliers: a single extreme value can artificially expand the dynamic range, degrading numerical fidelity across the entire block and corrupting global seasonal carriers. To safeguard these retained bins, a fixed orthogonal rotation $\mathcal{H}$ (e.g., Walsh–Hadamard or QR) is applied along the head dimension prior to the temporal RFFT. This operation disperses localized coordinate spikes uniformly across the head basis, stabilizing block scales and allowing the surviving spectral coefficients to be stored reliably at ultra-low precision.

Building on these insights, we introduce **RosKV** (Figure 2) as a training-free, drop-in KV-cache compressor for frozen TSFMs. Following the prefill pass, RosKV rotates the head dimension, performs a temporal RFFT, independently allocates energy-ranked frequency budgets to keys and values under a joint retention target, and quantizes the retained coefficients into adaptive mixed-precision NF4/NF2 tiers. During decoding, an inverse transform reconstructs historical context on the fly before attention, combining it with an exact buffer of newly generated tokens without modifying underlying model weights.

Our contributions are as follows:

- To the best of our knowledge, this is the first work investigating training-free Key–Value (KV) cache compression specifically tailored for time-series foundation models, identifying that temporal attention states are delocalized in time but concentrated in frequency.
- We design **RosKV**, an efficient rotation-aware spectral KV-cache compressor that combines head-dimension rotation, temporal RFFT filtering with separate K/V energy budgets, and adaptive low-bit quantization into a plug-in cache for frozen TSFMs without retraining.
- Across nine benchmarks, four horizons, and four foundation backbones, we show that RosKV stays within $+3.6\%$ of Full-Cache aggregate MSE, reducing cache memory by 55% on average (27.5 MB vs. 60.5 MB) and up to 78% at short horizons (9.5 MB vs. 42.5 MB) while running $> 2.1\times$ faster than No-Cache.

2 Preliminary

2.1 Autoregressive TSFM Inference and the KV Cache

We consider a frozen autoregressive Transformer decoder with N_L layers and N_h heads of dimension d. At each layer,

$$\mathrm{Attn}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}}\right)\boldsymbol{V}, \quad \boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{n\times d}. \tag{1}$$

Given a look-back window $\boldsymbol{x}_{1:L}$, the prefill pass materializes per-layer states $\boldsymbol{K}_{\text{pre}}, \boldsymbol{V}_{\text{pre}} \in \mathbb{R}^{L\times d}$. During decoding, newly generated tokens append $\boldsymbol{k}_t, \boldsymbol{v}_t$, and attention reads

$$\boldsymbol{K}_{\text{eff}} = [\boldsymbol{K}_{\text{pre}}; \boldsymbol{K}_{\text{dec}}], \qquad \boldsymbol{V}_{\text{eff}} = [\boldsymbol{V}_{\text{pre}}; \boldsymbol{V}_{\text{dec}}]. \tag{2}$$

Persistent memory and per-step attention work therefore scale with $L+t$. RosKV leaves weights and attention kernels untouched and compresses only the *persistent prefill* store; newly generated states remain in an exact FP32 decode buffer. The same interface covers decoder self-attention (TimeMoE, Moirai, TimesFM) and decoder cross-attention (Chronos): the compressor only needs read/write access to a per-layer $(\boldsymbol{K}_{\text{pre}}, \boldsymbol{V}_{\text{pre}})$ pair. Full notation is collected in Appendix A.1.

2.2 Compression Objective

A training-free prefill compressor is a map $(\boldsymbol{K}_{\text{pre}}, \boldsymbol{V}_{\text{pre}}) \mapsto \mathcal{C}$ that yields a compact persistent code $\mathcal{C}$, together with a reconstruction $\mathcal{C} \mapsto (\boldsymbol{K}_r, \boldsymbol{V}_r)$ used in place of $(\boldsymbol{K}_{\text{pre}}, \boldsymbol{V}_{\text{pre}})$ at attention time. The design goals are (i) small $|\mathcal{C}|$ relative to the dense FP32 prefill, (ii) forecast MSE close to Full-Cache under the frozen backbone, and (iii) no modification of pretrained weights. Token eviction realizes $\mathcal{C}$ by dropping rows of $\boldsymbol{K}_{\text{pre}}$ and $\boldsymbol{V}_{\text{pre}}$; coordinate quantization realizes $\mathcal{C}$ by reducing bit-width of every time-domain entry. RosKV instead builds $\mathcal{C}$ from a *sparse spectral support* of a rotated prefill, as motivated next.

2.3 Where Prefill Importance Lives

Two structural facts about frozen TSFM prefills (look-back L=512) drive the form of $\mathcal{C}$.

Importance is delocalized in time, concentrated in frequency. Seasonal modes are spread across many look-back positions, so row-level scores are a poor proxy for what the forecaster needs. Along the time axis, however, energy concentrates in few RFFT bins: for each layer and head, an RFFT of each KV column yields $F = \lfloor L/2 \rfloor + 1$ complex bins. Figure 1 plots the measured fraction of total K/V energy captured by the top-r energy-ranked bins, directly from prefill KV states of each frozen backbone: on TimeMoE the top 10% of bins retain 80–88% of KV energy (ETTh1, ETTm2, Weather, ECL), and at a comparable top-50% budget the retained fractions are 0.97, 0.83, 0.91, and 0.89 on TimeMoE, Chronos-1, Moirai, and TimesFM respectively, all above the uniform 0.50 baseline shown by the diagonal. Measured dataset coverage varies by backbone (4, 2, 8, and 8 of the nine main datasets; Appendix B.3). The resulting code keeps high-energy bins, discards the rest, and spends bits only on the survivors.

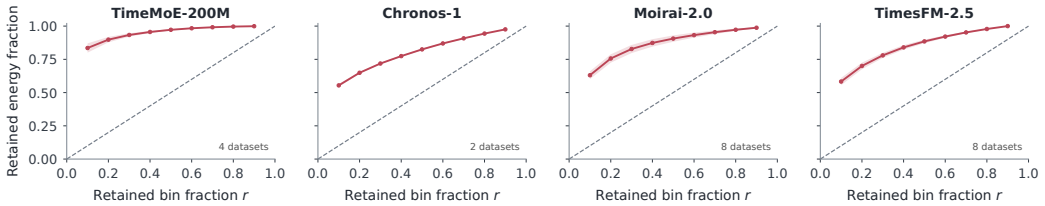


Figure 1: **Measured spectral energy concentration.** Mean fraction of total K/V RFFT energy captured by the top-r energy-ranked bins, per backbone (shaded: ± 1 std across the datasets noted in each panel; dashed: uniform baseline, i.e. energy $= r$).

The four backbones also differ sharply in how many bins there are to compress: TimeMoE, Chronos, Moirai, and TimesFM expose 257, 20, 20, and 9 RFFT bins respectively at L=512. The persistent-memory motivation in Section 1 therefore applies most directly to TimeMoE, whose long spectral footprint is where compression yields a materially smaller store. On the three short-footprint backbones a 9–20-bin cache is small in absolute terms regardless of compression. RosKV is primarily an accuracy stress test on these backbones: it tests whether the spectral contract tracks Full-Cache when only a handful of bins are available, rather than adding a separate memory-savings claim.

Kept spectral bins are sensitive to head-dimension outliers. After TopK, almost all useful signal sits in a few vectors in $\mathbb{R}^d$ (or $\mathbb{C}^d$). Blockwise NF4/NF2 codecs share a scale along the head axis, so one large coordinate inflates the scale and coarsens the entire block, corrupting a seasonal carrier rather than a single timestamp. A fixed orthogonal rotation $\mathcal{H}$ on the head axis (Hadamard with a shared Rademacher sign vector when d is a power of two; a seeded QR factor otherwise, e.g. TimesFM with d=80) is an isometry that spreads outliers before transform and coding. It does not

discard information; it only redistributes it. Keys and values also need not share the same spectral support: under one joint retention target r, RosKV allocates bin counts to K and V independently rather than imposing a fixed split (Section 3.3), so a more concentrated tensor can claim extra bins without exceeding the shared budget. We have not isolated this choice's effect against a tied split in ablation, so we present it as a design choice motivated by the asymmetry visible in Figure 1's per-tensor measurements, not as a validated comparative result.

These facts specify the RosKV contract: rotate the head basis, transform once along time, keep separate energy-ranked K/V supports, and store adaptive mixed-precision coefficients without touching TSFM weights.

3 RosKV Design

RosKV is a training-free, drop-in realization of the compression objective in Section 2.2. It compresses along the frequency axis where energy concentrates, after a fixed rotation that makes low-bit coding of the kept bins safer, and then reconstructs a dense prefill for attention. Because autoregressive decode attends at every step, the default path reconstructs every step unless an eager or fused variant is used. Figure 2 previews the full pipeline. The four operators are rotation, RFFT filtering, mixed-precision coding, and reconstructive decode.

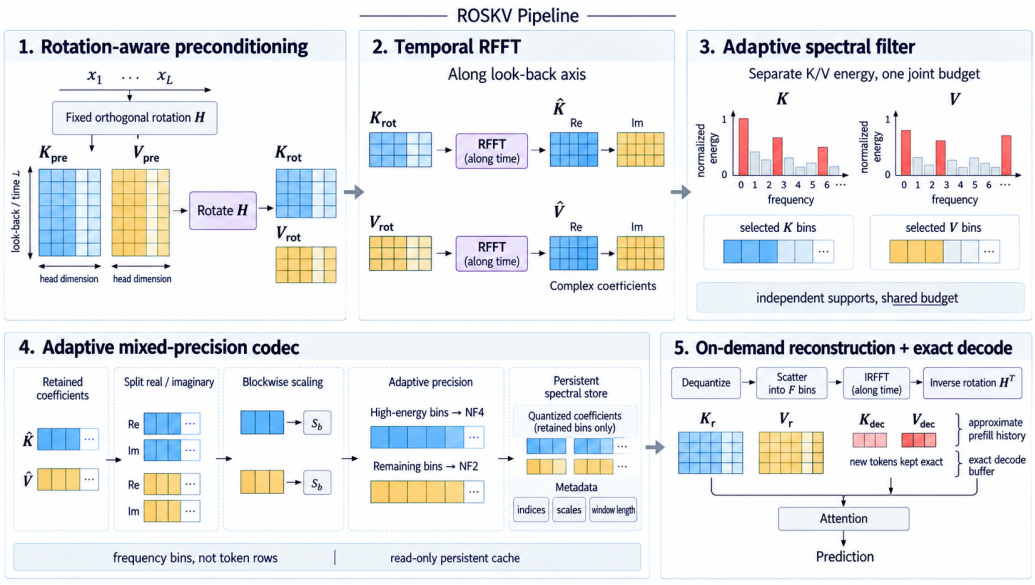


Figure 2: **RosKV pipeline.** Rotate prefill K/V along the head dimension, apply a temporal RFFT, select bins with the adaptive K/V energy filter, store adaptive NF4/NF2 coefficients, then invert the codec, scatter, IRFFT, and inverse-rotate before attention. Newly generated tokens are appended in exact precision.

3.1 Pipeline Overview

Using the prefill/decode interface of Section 2.1, RosKV maps each prefill tensor through four inference-only operators: a fixed head-dimension rotation $\mathcal{H}$, a temporal RFFT, an adaptive support $(\mathcal{F}_K^*, \mathcal{F}_V^*)$, and a mixed-precision codec $\mathcal{Q}$. It inverts these operators at decode time:

$$\hat{\boldsymbol{K}} = \mathrm{RFFT}\big(\mathcal{H}\boldsymbol{K}_{\mathrm{pre}}\big), \qquad \hat{\boldsymbol{V}} = \mathrm{RFFT}\big(\mathcal{H}\boldsymbol{V}_{\mathrm{pre}}\big), \tag{3}$$

$$\boldsymbol{K}_r = \mathcal{H}^{\mathsf{T}}\,\mathrm{IRFFT}\big(\mathrm{Scatter}_F\big(\mathcal{Q}^{-1}(\mathcal{Q}(\hat{\boldsymbol{K}}_{\mathcal{F}_K^*}))\big),\, L\big), \tag{4}$$

and likewise for $\boldsymbol{V}_r$ with support $\mathcal{F}_V^*$. The forward path runs once after prefill; the inverse path runs whenever attention needs dense history. The ordering is deliberate: rotating before the temporal transform conditions the head basis that later shares NF block scales; filtering before quantizing

ensures precision is spent only on coefficients that survive selection. Appendix A.2 gives the storage procedure and pseudocode.

3.2 Rotation-Aware Preconditioning

In standard attention states, individual feature coordinates along the head dimension often exhibit pronounced magnitude outliers. In an unrotated coordinate system, these isolated spikes force a blockwise quantizer's dynamic range to expand dramatically, coarsening the effective numerical resolution for all neighboring coordinates that share the block scale. To stabilize subsequent low-bit coding without discarding information, we precondition the head dimension with a fixed orthogonal rotation $\mathcal{H}$ prior to the temporal transform. When the head dimension d is a power of two, this rotation is formulated as:

$$\mathcal{H}X \;=\; d^{-1/2} H_d \,(X \,\mathrm{diag}(\mathbf{s})), \tag{5}$$

where H_d is the normalized Walsh–Hadamard matrix and $\mathbf{s} \in \{\pm 1\}^d$ is a fixed Rademacher vector shared across layers; because $\mathcal{H}$ is orthogonal, its exact inverse is simply $\mathcal{H}^{-1} = \mathcal{H}^{\mathsf{T}}$. For architectures where d is not a power of two (such as TimesFM with d=80), $\mathcal{H}$ is set to the orthogonal matrix $\mathbf{Q}$ obtained from the QR decomposition of a fixed, seeded Gaussian matrix. Operating as an exact norm-preserving isometry on the uncompressed KV states, $\mathcal{H}$ introduces zero learnable parameters while uniformly dispersing localized outlier energy across all d coordinates, ensuring that subsequent spectral ranking and mixed-precision NF4/NF2 block quantization operate over well-conditioned distributions (Section 4.4).

3.3 Adaptive Spectral Filter

After rotation, we transform along the look-back axis (Eq. (3)), obtaining $F = \lfloor L/2 \rfloor + 1$ complex bins. Following Section 2.3, we score keys and values *separately*,

$$e_f^K = \tfrac{1}{N_h d} \sum_{h,j} |\hat{K}_{hfj}|^2, \qquad e_f^V = \tfrac{1}{N_h d} \sum_{h,j} |\hat{V}_{hfj}|^2, \tag{6}$$

and normalize to probability masses $\bar{e}_f^K$ and $\bar{e}_f^V$. Let r be the target retained fraction and δ=0.02 a per-tensor slack. With joint budget $B = 2\lceil Fr \rceil$ and admissible counts in $[\ell, u] = [\lceil F(r-\delta) \rceil, \lfloor F(r+\delta) \rfloor]$, the filter maximizes retained joint energy under one storage target:

$$(k_K, k_V) \;=\; \arg \max_{\substack{k_K + k_V = B \\ k_K, k_V \in [\ell, u]}} \sum_{i=1}^{k_K} \bar{e}_{(i)}^K + \sum_{i=1}^{k_V} \bar{e}_{(i)}^V, \tag{7}$$

then sets $\mathcal{F}_K^* = \mathrm{TopK}(e^K, k_K)$ and $\mathcal{F}_V^* = \mathrm{TopK}(e^V, k_V)$. K and V therefore receive independent supports: a layer may keep more high-energy key bins and fewer value bins (or vice versa) without changing the total coefficient count. For instance, if a layer's key spectrum is markedly more concentrated than its value spectrum, the filter can retain more key bins and fewer value bins while leaving the total stored coefficient count, and therefore the persistent memory footprint, unchanged. The default operating point is r=0.5 (each side in $[0.48, 0.52]$); Section 4.4 isolates this filter's contribution by comparing RosKV against variants with the rotation, the filter, or the codec removed in turn.

3.4 Adaptive Mixed-Precision Quantization

Retained complex coefficients still dominate storage if kept in FP32. Within each retained spectrum of k bins we therefore assign an NF4 tier and a compact NF2 tier, following the normal-float codebook design of QLoRA (**?**): codebook levels are placed so that, for coefficients approximately normally distributed within a block, each quantization bin receives roughly equal probability mass, which spends less of the available bit budget on rarely visited tail values than a uniformly spaced codebook of the same width would. Let $\mathrm{CV} = \sigma(e)/\mu(e)$ be the coefficient of variation of retained energies. The NF4 fraction is

$$p \;=\; \mathrm{clip}_{[p_{\min}, p_{\max}]}\big(p_0 + \alpha \tanh(\log(1 + \mathrm{CV}) - \tau)\big), \qquad n_4 = \mathrm{round}(kp), \tag{8}$$

so concentrated spectra receive more NF4 slots while flatter spectra stay near the default p_0 (constants in Table 7). The n_4 highest-energy retained bins use NF4; the remainder use a 4-level normal-float

codebook $\{\pm 1, \pm\gamma\}$. The persistent store holds $\mathcal{Q}(\hat{\boldsymbol{K}}_{\mathcal{F}_K^*})$ and $\mathcal{Q}(\hat{\boldsymbol{V}}_{\mathcal{F}_V^*})$, together with frequency indices, block scales, and L. This design separates *which* bins survive (the filter) from *how accurately* they are coded (the codec), enabling the ablations in Section 4.4.

3.5 Reconstructive Decoding

At each decode step the default path rebuilds prefill K/V via Eq. (4) and does *not* cache the dense result: the packed spectral code is the only persistent history, while a temporary reconstruction is materialized for that step's attention. Newly generated tokens stay in an exact FP32 decode buffer. An *eager-recon* variant rebuilds once and caches dense tensors (latency closer to Full-Cache, no packed win); chunked reuse and fused reconstruct-and-attend are possible extensions that we do not test. Appendix C.3 reports the default storage–latency trade-off, with peak memory including temporary reconstructions.

4 Experiments

4.1 Experimental Setup

Datasets and backbones. We evaluate nine long-horizon forecasting datasets (ETTh1/2, ETTm1/2, Exchange, Solar, ECL, Traffic, Weather) and four frozen TSFMs zero-shot: TimeMoE (Shi et al., 2025), Chronos (Ansari et al., 2024), Moirai (Woo et al., 2024), and TimesFM (Das et al., 2023) (Appendix B.3).

Baselines. Full-Cache, SnapKV (Li et al., 2024), PyramidKV (Cai et al., 2024), KIVI (Liu et al., 2024b), and FreqKV (Kai et al., 2026) transfer (Appendix B; Table 7).

Environment. PyTorch on NVIDIA A100 GPUs; primary metrics are MSE/MAE after inverse instance normalization. Cache Memory, and latency appear in Section 4.3 (with extended breakdowns in Appendix C.3).

4.2 Main Results

We evaluate RosKV at L=512 across horizons $H \in \{96, 192, 336, 720\}$. Table 1 reports MSE and MAE averaged over nine benchmark datasets; Δ is relative MSE versus Full-Cache (negative denotes an accuracy gain). Across the 36 individual backbone–dataset configurations, RosKV improves MSE over Full-Cache in 15 cells and remains within $\pm 5\%$ in 24, achieving a near-neutral median relative change of $+0.3\%$ (see full per-dataset breakdowns in Appendix C.1). Retention budgets are approximately matched across compressor families (Appendix B), providing operating-point comparisons under practical constraints. Three key insights emerge from Table 1:

(1) Universally near-lossless (and net-positive) aggregate accuracy. While substantially shrinking the persistent prefill representation, RosKV consistently tracks Full-Cache across all four backbones: $+3.6\%$ on TimeMoE (0.310 vs. 0.299), -0.5% on Chronos (0.522 vs. 0.524), -2.2% on Moirai (0.405 vs. 0.414), and $+1.6\%$ on TimesFM (0.407 vs. 0.400). On Chronos (evaluated on cross-attention) and Moirai, spectral truncation even yields mild net MSE improvements over Full-Cache, consistent with discarding noisy, high-frequency residuals while preserving macro seasonality.

(2) Superiority over alternative compression paradigms. RosKV is the best-performing compressed method on TimeMoE, Chronos, and Moirai. On TimesFM, where all methods reside in a compact error corridor (-3.5% to $+4.2\%$), RosKV remains within $+1.6\%$ of Full-Cache. In contrast, NLP-derived token eviction (SnapKV and PyramidKV) suffers severe accuracy degradation across TimeMoE ($+24.1\%$ to $+30.4\%$), Chronos ($+24.9\%$ to $+61.5\%$), and Moirai ($+29.5\%$ to $+38.9\%$). Dropping temporal rows breaks periodic phase continuity across the look-back window, whereas RosKV's frequency-domain selection retains global temporal envelopes.

(3) Robustness where coordinate quantization collapses. Although coordinate-wise quantization (KIVI) preserves accuracy on patch-based models (Chronos $+0.6\%$, Moirai $+0.3\%$), it degrades sharply on the dense token representations of TimeMoE ($+33.4\%$). RosKV successfully closes this gap ($+3.6\%$ vs. $+33.4\%$), demonstrating that combining frequency-domain energy ranking with outlier-rotated low-bit coding provides stability where time-domain coordinate quantizers fail.

Table 1: **Main results** at look-back L=512, averaged over nine datasets. Each cell reports MSE, MAE, and relative MSE change (Δ) versus Full-Cache; lower is better and non-positive Δ is highlighted.

Method	Horizon	TimeMoE			Chronos			Moirai			TimesFM		
		MSE↓	MAE↓	Δ_{MSE} ↓	MSE↓	MAE↓	Δ_{MSE} ↓	MSE↓	MAE↓	Δ_{MSE} ↓	MSE↓	MAE↓	Δ_{MSE} ↓
Full-Cache	96	0.173	0.265	—	0.297	0.294	—	0.205	0.266	—	0.199	0.252	—
	192	0.281	0.336	—	0.518	0.413	—	0.327	0.349	—	0.342	0.362	—
	336	0.344	0.386	—	0.562	0.456	—	0.452	0.424	—	0.424	0.424	—
	720	0.397	0.431	—	0.721	0.551	—	0.672	0.542	—	0.635	0.548	—
	Avg	0.299	0.354	—	0.524	0.428	—	0.414	0.395	—	0.400	0.397	—
SnapKV	96	0.208	0.295	+20.2%	0.468	0.351	+57.5%	0.319	0.323	+55.6%	**0.199**	**0.252**	+0.0%
	192	0.375	0.405	+33.5%	0.655	0.461	+26.5%	0.524	0.443	+60.2%	0.335	0.362	−2.0%
	336	0.446	0.461	+29.7%	0.616	0.507	+9.7%	0.526	0.471	+16.4%	0.412	**0.419**	−2.8%
	720	0.529	0.529	+33.2%	0.880	0.633	+22.2%	0.776	0.581	+15.5%	**0.599**	**0.531**	−5.7%
	Avg	0.390	0.423	+30.4%	0.655	0.488	+24.9%	0.536	0.454	+29.5%	**0.386**	**0.391**	−3.5%
KIVI	96	0.255	0.350	+47.3%	0.299	0.297	+0.6%	0.216	0.278	+5.4%	**0.199**	**0.252**	+0.2%
	192	0.362	0.411	+28.9%	0.529	0.416	+2.2%	0.338	0.367	+3.3%	0.352	0.369	+2.9%
	336	0.438	0.464	+27.3%	0.558	0.456	−0.6%	**0.441**	0.430	−2.4%	0.454	0.443	+7.0%
	720	0.516	0.518	+29.9%	0.724	**0.554**	+0.4%	**0.638**	0.539	−5.1%	0.678	0.571	+6.7%
	Avg	0.393	0.436	+33.4%	0.528	0.431	+0.6%	0.408	0.403	+0.3%	0.421	0.409	+4.2%
PyramidKV	96	0.201	**0.290**	+16.2%	0.465	0.435	+56.8%	0.353	0.363	+72.2%	**0.199**	**0.252**	+0.0%
	192	0.361	0.397	+28.5%	0.759	0.585	+46.6%	0.543	0.481	+66.1%	0.333	**0.360**	−2.6%
	336	0.422	0.445	+22.7%	0.886	0.660	+57.8%	0.599	0.523	+32.5%	0.412	0.421	−2.8%
	720	0.502	0.510	+26.4%	1.276	0.820	+77.2%	0.806	0.627	+19.9%	0.616	0.541	−3.0%
	Avg	0.371	0.410	+24.1%	0.847	0.625	+61.5%	0.575	0.498	+38.9%	0.390	0.394	−2.5%
FreqKV	96	0.245	0.306	+41.6%	0.338	0.331	+13.7%	0.343	0.336	+67.3%	**0.199**	**0.252**	+0.0%
	192	0.370	0.397	+31.7%	0.595	0.464	+14.8%	0.468	0.426	+43.1%	**0.329**	**0.360**	−3.8%
	336	0.472	0.464	+37.2%	0.641	0.506	+14.1%	0.588	0.494	+30.1%	**0.410**	0.423	−3.3%
	720	0.624	0.566	+57.2%	0.880	0.619	+22.2%	0.950	0.644	+41.4%	0.669	0.564	+5.4%
	Avg	0.428	0.433	+43.1%	0.613	0.480	+17.0%	0.587	0.475	+41.8%	0.402	0.400	+0.5%
RosKV (ours)	96	**0.198**	0.299	+14.5%	**0.294**	**0.294**	−0.9%	**0.206**	**0.266**	+0.3%	**0.199**	**0.252**	+0.2%
	192	**0.288**	**0.355**	+2.6%	**0.515**	**0.412**	−0.5%	**0.321**	**0.348**	−1.9%	0.337	0.363	−1.6%
	336	**0.346**	**0.403**	+0.5%	**0.556**	**0.455**	−1.1%	0.449	**0.423**	−0.7%	0.441	0.436	+4.0%
	720	**0.406**	**0.451**	+2.3%	**0.723**	**0.554**	+0.3%	0.642	**0.531**	−4.5%	0.650	0.562	+2.4%
	Avg	**0.310**	**0.377**	+3.6%	**0.522**	**0.429**	−0.5%	**0.405**	**0.392**	−2.2%	0.407	0.403	+1.6%

4.3 Memory and Efficiency Benefits of RosKV

To quantify the resource footprint of autoregressive decoding across short and extended generation budgets, we benchmark multi-task forecasting on TimeMoE across $H \in \{96, 192, 336, 720\}$ (10 windows $\times$ 5 channels across ETTh1, ETTm2, and Weather; details in Appendix C.3). Table 2 exposes the trade-offs across the memory–compute continuum:

(1) Persistent memory savings across all horizons. Across every forecast horizon, RosKV drastically compresses persistent KV cache storage compared to Full-Cache and token-eviction baselines. At short horizons (H=96), RosKV slashes cache storage from 42.5 MB to just 9.5 MB—a 78% **memory reduction** ($> 4.4\times$ smaller than Full-Cache and $2.5\times$ smaller than SnapKV's 23.7 MB). Across intermediate horizons, RosKV sustains major memory advantages (H=192: 14.2 MB vs. 47.2 MB, −70%; H=336: 29.0 MB vs. 61.9 MB, −53%; H=720: 57.3 MB vs. 90.2 MB, −36%), averaging 27.5 MB (−55%). In edge environments where on-chip memory limits parallel stream capacity, this sustained reduction enables hosting significantly higher concurrency without OOM faults.

(2) Superior error profile across horizons. At short horizons ($H \in \{96, 192\}$), RosKV consistently improves over Full-Cache (0.235 vs. 0.246, and 0.381 vs. 0.413). As generation scales to $H \in \{336, 720\}$, token-eviction baselines degrade substantially, whereas RosKV achieves the lowest MSE

(0.510 and 0.514), outperforming Full-Cache (0.535 and 0.525) and yielding the best overall average MSE (0.410).

(3) Memory-first trade-off with competitive latency. Omitting the cache entirely (No-Cache) eliminates persistent state (0.0 MB) but triggers costly recomputation (810 ms). With optimized NF4 dequantization, RosKV's average decoding latency reaches 383 ms, running $> 2.1\times$ faster than No-Cache (810 ms), faster than SnapKV (483 ms) and PyramidKV (767 ms), and outperforming all alternative compressed baselines (KIVI 392 ms, FreqKV 396 ms), while providing substantial persistent cache compression across all deployment scenarios.

Table 2: **Multi-task efficiency and memory footprint**, complete per-horizon breakdowns appear in Appendix C.3 (Table 11).

Metric (Averaged over H)	No-Cache	Full-Cache	SnapKV	KIVI	PyramidKV	FreqKV	RosKV (ours)
Cache Memory (MB)↓	**0.0**	60.5	41.7	37.0	41.7	41.7	**27.5** (-55%)
MSE↓	0.430	0.430	0.463	0.434	0.445	0.462	**0.410**
Latency (ms)↓	810	240	483	392	767	396	**383**

4.4 Ablation Study

Next, to study the individual contribution of each algorithmic component in RosKV, we isolate spectral filtering, randomized rotation, and coefficient quantization by removing one component at a time under matched look-back, decode buffer, and precision settings. As shown in Table 3, the complete RosKV pipeline attains the lowest compressed MSE and MAE across every individual backbone architecture as well as the lowest overall Total (0.744 vs. Full-Cache 0.752), establishing that its core components operate synergistically.

Removing RFFT spectral filtering causes the largest degradation across backbones, elevating Total MSE to 0.815 ($+9.5\%$ relative over RosKV). This degradation is particularly severe on TimeMoE (0.965 vs. 0.760, $+27.0\%$) and Moirai (0.802 vs. 0.773), confirming that energy-ranked frequency truncation distills salient periodic dynamics while discarding high-frequency noise. In turn, eliminating the randomized Hadamard/QR rotation raises Total MSE to 0.777 ($+4.4\%$), with marked regressions on TimeMoE (0.808), Chronos (0.816), and Moirai (0.817), demonstrating that rotation disperses heavy-tailed channel kurtosis before low-bit quantization. Finally, retaining unquantized coefficients yields 0.760 Total MSE, slightly above RosKV (0.744), showing that non-uniform NF4/NF2 quantization preserves fidelity while providing mild regularization.

Table 3: **Component ablation** on ETTh1/ETTm2/Exchange. Backbone averages over $H \in \{96, 192, 336, 720\}$; bold: best among compressed variants. Details in Appendix C.4.

Method	TimeMoE		Chronos		Moirai		TimesFM		Total	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Full-Cache	0.853	0.643	0.808	0.587	0.700	0.577	0.645	0.542	0.752	0.587
w/o spectral filtering	0.965	0.694	0.829	0.616	0.802	0.598	0.662	0.550	0.815	0.615
w/o rotation	0.808	0.636	0.816	0.591	0.817	0.596	0.665	0.553	0.777	0.594
w/o quantization	0.798	0.625	0.799	0.585	0.790	0.588	0.653	0.546	0.760	0.586
RosKV	**0.760**	**0.610**	**0.798**	**0.585**	**0.773**	**0.581**	**0.646**	**0.544**	**0.744**	**0.580**

4.5 Evaluation on GIFT-Eval

To evaluate out-of-distribution (OOD) generalization, we stress-test the cache settings on two real-world, high-noise hourly series from GIFT-Eval: air-quality measurements (KDD-Cup-2018) and freeway traffic loop detectors (LOOP-Seattle). Table 4 highlights three primary observations:

(1) Consistent leadership. RosKV achieves the lowest MSE and MAE among all compressed methods on every evaluated backbone without exception, attaining Total MSE 0.485 (vs. Full-Cache 0.550 and next-best KIVI 0.583).

(2) Out-of-distribution denoising. On Chronos, Moirai, and TimesFM, RosKV strictly outperforms uncompressed Full-Cache: 0.542 vs. 0.549 on Chronos, 0.470 vs. 0.657 on Moirai (28.5% reduction), and 0.543 vs. 0.674 on TimesFM (19.4% reduction), confirming that energy-ranked spectral filtering eliminates sensor noise while retaining periodic modes.

(3) Robustness across architectures. Token-eviction baselines degrade on unfamiliar OOD series (PyramidKV MSE rises to 0.723, SnapKV to 0.651), while coordinate quantization degrades on TimeMoE (KIVI 0.700 vs. RosKV 0.384). RosKV maintains stability across dense-token and patch-based backbones under distribution shift.

Table 4: **GIFT-Eval** backbone averages (KDD-Cup-2018, LOOP-Seattle). Bold: best among compressed methods. Per-horizon results in Appendix C.5.

Backbone	Full-Cache		SnapKV		KIVI		PyramidKV		FreqKV		RosKV	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeMoE	0.319	0.354	0.465	0.454	0.700	0.572	0.428	0.433	0.452	0.451	**0.384**	**0.411**
Chronos	0.549	0.514	0.645	0.539	0.573	0.519	0.948	0.698	0.702	0.583	**0.542**	**0.510**
Moirai	0.657	0.490	0.792	0.558	0.517	0.506	0.816	0.567	0.995	0.625	**0.470**	**0.460**
TimesFM	0.674	0.506	0.700	0.514	0.544	0.504	0.699	0.514	0.761	0.534	**0.543**	**0.503**
Total	0.550	0.466	0.651	0.516	0.583	0.525	0.723	0.553	0.728	0.548	**0.485**	**0.471**

5 Related Work

KV-cache quantization and rotation. Coordinate quantizers (KIVI (Liu et al., 2024b), KVQuant (Hooper et al., 2024)) reduce bit-width without changing stored positions. RosKV applies that preconditioning to temporal TSFM KV and pairs it with spectral filtering.

Frequency-domain compression and forecasting. FreqKV (Kai et al., 2026) truncates DCT coefficients for LLMs; FEDformer/FreTS/FITS/TimesNet (Zhou et al., 2022; Yi et al., 2023; Xu et al., 2024; Wu et al., 2022) inject spectral bias by retraining. Our FreqKV transfer (Appendix B) keeps lowest DCT bins in FP32 without rotation or mixed precision; RosKV uses RFFT, energy-ranked separate K/V supports, and NF4/NF2.

Time-series foundation models. TSFMs including TimeMoE, Chronos, Moirai, and TimesFM (Shi et al., 2025; Ansari et al., 2024; Woo et al., 2024; Das et al., 2023) transfer zero-shot via attention and inherit expanding KV caches; RosKV compresses this persistent historical store training-free.

6 Conclusion

We presented **RosKV**, a training-free spectral compressor for frozen TSFM KV caches: rotate, RFFT, separate energy-ranked K/V filtering, and adaptive NF4/NF2. Across nine datasets, four horizons, and four backbones, RosKV stays within $+3.6\%$ of Full-Cache aggregate MSE while outperforming token eviction, coordinate quantization, and frequency baselines. On long-context dense-token architectures, RosKV slashes persistent cache memory by 55% on average and up to 78% at short horizons (9.5 MB vs. 42.5 MB), offering a principled foundation for memory-efficient time-series foundation model inference.

AI Use Statement. The authors used automated drafting and programming tools during paper preparation; all design choices, code changes, executions, interpretations, and claims were reviewed and controlled by the authors.

Reproducibility Statement. Implementation details, notation, and algorithmic procedures are provided in Appendix A. Complete experimental setups, full benchmark tables, and code will be released publicly upon acceptance.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

Jushi Kai, Yixuan Wang, Boyi Zeng, Haoli Bai, Bo Jiang, Ziwei He, and Zhouhan Lin. Freqkv: Key-value compression in frequency domain for context window extension. In *International Conference on Learning Representations*, volume 2026, pp. 105568–105587, 2026.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. *arXiv preprint arXiv:2402.02368*, 2024a.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*, 2024b.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *International conference on learning representations*, volume 2025, pp. 34635–34667, 2025.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, volume 2024, pp. 26295–26318, 2024.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in neural information processing systems*, 36:76656–76679, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in neural information processing systems*, 36:34661–34710, 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 *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

A Implementation Details

We collect notation, algorithms, and backbone-specific implementation notes for RosKV below.

A.1 Notation

Table 5 summarizes the mathematical notation used throughout this work. We distinguish between the look-back context length L and the forecast horizon H. The retention budget controls the trade-off between memory footprint and reconstruction fidelity, while the per-layer allocator distributes that budget between K and V.

Table 5: **Summary of Notation.** Symbols and their corresponding definitions across sequence architecture, spectral operations, and memory evaluation.

Symbol	Definition	Typical Value / Space
Architecture and Dimensions		
L	Look-back length (input context)	512 (default)
H	Forecast horizon (output length)	$\{96, 192, 336, 720\}$
N_L, N_h	Number of layers and attention heads	12 (TimeMoE-200M)
d	Attention head dimension	64
t	Autoregressive decode step	$t \in \{1, \dots, H\}$
Tensors and Transforms		
$\mathbf{Q}, \mathbf{K}, \mathbf{V}$	Query, Key, and Value matrices	$\mathbb{R}^{L \times d}$
$\mathbf{K}_{\text{pre}}, \mathbf{V}_{\text{pre}}$	Prefill KV states (to be compressed)	$\mathbb{R}^{L \times d}$
$\mathbf{K}_{\text{dec}}, \mathbf{V}_{\text{dec}}$	Decode KV states (exact/uncompressed)	$\mathbb{R}^{t \times d}$
$\mathcal{H}, \mathbf{s}$	Head-dimension rotation and Rademacher signs	$\mathcal{H}^\top \mathcal{H} = I, \mathbf{s} \in \{\pm 1\}^d$
$\hat{\mathbf{K}}, \hat{\mathbf{V}}$	RFFT of rotated KV states	$\mathbb{C}^{F \times d}$
F	Total RFFT bins	$F = \lfloor L/2 \rfloor + 1$
Spectral Compression		
r, δ	Target retained fraction and per-tensor slack	$r \in (0, 1], \delta \geq 0$
B	Joint K+V bin budget	$B = 2\lceil Fr \rceil$
k_K, k_V	Retained frequency-bin counts for K and V	$k_K + k_V = B$, each in $[r - \delta, r + \delta]$
$\mathcal{F}_K^*, \mathcal{F}_V^*$	Selected K/V frequency-index sets	Independent top-energy sets
e_f^K, e_f^V	K/V spectral energy at bin f	$\mathbb{R}_{\geq 0}$
$\mathcal{Q}, p, n_4$	Codec, NF4 fraction, NF4 count	$p \in [0, 1]$
Performance Metrics		
M_{FC}	Full-Cache memory footprint	$8 \times N_L N_h L d$ (bytes)
M_{RosKV}	RosKV memory footprint	Compressed store + buffer
MSE	Mean Squared Error	Instance-normalized

A.2 The RosKV Pipeline

Overview. The RosKV lifecycle consists of two phases integrated into the frozen TSFM inference logic, following the four operators of Section 3: the rotation $\mathcal{H}$ (Section 3.2), the retention target r and slack δ of the adaptive filter (Section 3.3), and the codec $\mathcal{Q}$ (Section 3.4) are all run-level configuration inputs to Algorithm 1, not fixed constants baked into the procedure. $\mathcal{Q}$ is instantiated three ways in the paper, all special cases of the single mixed-precision codec of Eq. (8): with $(p_0, \alpha, \tau, p_{\min}, p_{\max})$ left at their adaptive default; with $p_{\min}{=}p_{\max}{=}p_0{=}1$, so every retained bin is stored in NF4; and with $\mathcal{Q} = \text{id}$, which stores the selected support losslessly and isolates the codec's contribution from the rotation, transform, and filter (the w/o-quantization ablation, Table 3). The instantiated $(r, \delta, p_0, \alpha, \tau, p_{\min}, p_{\max})$ for every reported experiment are in Table 7 and the relevant table captions in Section 4; no default is assumed here.

1. **Prefill Compression:** During the initial forward pass, the model computes dense $\mathbf{K}_{\text{pre}}, \mathbf{V}_{\text{pre}}$ states. RosKV applies the rotation $\mathcal{H}$, a temporal RFFT, the joint energy filter of Eq. (7), and the codec $\mathcal{Q}$. It stores the coded real/imaginary coefficients, block scales when $\mathcal{Q} \neq \text{id}$, $\mathcal{F}_K^*, \mathcal{F}_V^*$, and L. 2. **Reconstructive Decoding:** For each autoregressive step t, Eq. (4) applies $\mathcal{Q}^{-1}$ (the identity when $\mathcal{Q} = \text{id}$), scatters, applies IRFFT, and inverse-rotates. Newly generated tokens $\mathbf{k}_t, \mathbf{v}_t$ are stored in

an exact `float32` buffer. Attention reads the concatenation of reconstructed historical and exact recent KV states.

Algorithms. Algorithm 1 details the per-layer compression logic, while Algorithm 2 provides the end-to-end inference execution.

Algorithm 1 RosKV Prefill Compression (Per-Layer)

Require: KV tensors $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{N_h \times L \times d}$, target r, slack δ, codec $\mathcal{Q}$

$F \leftarrow \lfloor L/2 \rfloor + 1$; $B \leftarrow 2\lceil Fr \rceil$

$\hat{\mathbf{K}}, \hat{\mathbf{V}} \leftarrow \mathrm{RFFT}(\mathcal{H}\mathbf{K}), \mathrm{RFFT}(\mathcal{H}\mathbf{V})$ {Eq. (3)}

$e_f^K, e_f^V \leftarrow \frac{1}{N_h d} \sum_{h,j} |\hat{K}_{hfj}|^2, \frac{1}{N_h d} \sum_{h,j} |\hat{V}_{hfj}|^2$

$(k_K, k_V) \leftarrow$ Eq. (7) with (r, δ); $(\mathcal{F}_K^*, \mathcal{F}_V^*) \leftarrow \mathrm{TopK}(e^K, k_K), \mathrm{TopK}(e^V, k_V)$

Store: $\mathcal{Q}(\hat{\mathbf{K}}_{\mathcal{F}_K^*}), \mathcal{Q}(\hat{\mathbf{V}}_{\mathcal{F}_V^*})$, block scales when $\mathcal{Q} \neq \mathrm{id}$, $\mathcal{F}_K^*, \mathcal{F}_V^*$, and L

return reconstructed $\mathbf{K}_r, \mathbf{V}_r$ via Eq. (4)

Algorithm 2 RosKV End-to-End Inference

Require: r, δ, and codec $\mathcal{Q}$, fixed for the run

Initialize compressed spectral store and empty decode buffers

for each layer ℓ **do**

  $\mathbf{K}_\ell, \mathbf{V}_\ell \leftarrow \mathrm{PrefillForward}(\mathrm{Input})$

  $\mathbf{K}_r, \mathbf{V}_r \leftarrow$ Algorithm 1$(\mathbf{K}_\ell, \mathbf{V}_\ell, r, \delta, \mathcal{Q})$ {Compress and get initial cache}

end for

for step $t = 1$ **to** H **do**

  for each layer ℓ **do**

    $[\mathbf{k}_t, \mathbf{v}_t] \leftarrow \mathrm{StepForward}(\mathrm{token}_{t-1})$

    $\mathbf{K}_{\mathrm{dec}}, \mathbf{V}_{\mathrm{dec}} \leftarrow \mathrm{Append}([\mathbf{k}_t, \mathbf{v}_t])$ {Update exact buffer}

    $\mathbf{K}_r, \mathbf{V}_r \leftarrow$ Eq. (4) for layer ℓ with codec $\mathcal{Q}$ {Apply $\mathcal{Q}^{-1}$, scatter, IRFFT, inverse-rotate}

    $\mathbf{K}_{\mathrm{kv}}, \mathbf{V}_{\mathrm{kv}} \leftarrow [\mathbf{K}_r; \mathbf{K}_{\mathrm{dec}}], [\mathbf{V}_r; \mathbf{V}_{\mathrm{dec}}]$ {Temporal concatenation}

    Compute Attention using $\mathbf{K}_{\mathrm{kv}}, \mathbf{V}_{\mathrm{kv}}$

    Release reconstructed K/V and temporary concatenations after attention

  end for

end for

Memory lifecycle. Reconstructed prefill states are temporary in the standard configuration. Coded coefficients, metadata, and exact decode buffers persist. $\mathcal{Q} = \mathrm{id}$ retains the same selected support as a quantized $\mathcal{Q}$ but a larger persistent footprint, since nothing is coded down from `float32`. Encoder experiments reconstruct within a forward pass and have no persistent autoregressive cache.

B Experimental Details

We document backbones, hyperparameters, and datasets used to evaluate RosKV.

B.1 Backbone Models and Implementations

Table 6 lists the versions used wherever the main text writes TimeMoE, Chronos, Moirai, or TimesFM. All models are instantiated in full precision (`float32`) and evaluated in a zero-shot regime without further fine-tuning. For integration with the library, RosKV extends the HuggingFace Transformers library by subclassing `DynamicCache`, maintaining compatibility with standard KV-cache APIs while implementing our spectral compression logic.

Table 6: **Backbone versions** used in the main text. All checkpoints are pretrained, frozen, and evaluated zero-shot.

Model Family	Model Variant	Params	HuggingFace Identifier
TimeMoE	TimeMoE-200M	200 M	`Maple728/TimeMoE-200M`
Chronos	Chronos-1 (T5-small)	46 M	`amazon/chronos-t5-small`
Moirai	Moirai-2.0-R-small	14 M	`Salesforce/moirai-2.0-R-small`
TimesFM	TimesFM-2.5-200M	200 M	`google/timesfm-2.5-200m-pytorch`

B.2 Hyperparameter Configurations

The experimental configuration for ROSKV is summarized in Table 7. We maintain a consistent look-back length of $L = 512$ across all models to ensure a fair comparison of KV-cache efficiency. The main configuration uses adaptive K/V allocation and adaptive mixed-precision coefficient storage; unquantized and fixed-precision ablations are identified separately.

Table 7: **Hyperparameter Summary.** Unless otherwise specified in ablation studies, these default values are used across all experiments.

Hyperparameter	Value
Look-back length (L)	512
Target joint retention (r)	0.5
Eq. (8) constants	p_0=0.20, α=0.10, τ=0.7
NF2 codebook level	$\gamma \approx 0.277$ (codebook $\{\pm 1, \pm\gamma\}$)
Quantization block size	64
Test samples per (dataset, horizon)	20 (main), 10 (ablation/cross-model)
Test-window protocol	First n held-out windows; backbone-specific split. Table 10 adds two more TimeMoE blocks starting 20 and 40 windows later
Max channels per dataset	5 (Table 1); varies in other studies
Numerical Precision	`float32` weights/decode buffer; adaptive NF4/NF2 coefficient tiers with block scales (Section 3)
Coefficient preconditioning	Fixed orthogonal $\mathcal{H}$ (Hadamard+Rademacher, or QR if d is not a power of two); inverse after IRFFT. Table 1 Moirai H=720 omits $\mathcal{H}$ (identity)
Software Environment	PyTorch 2.x, Transformers 4.52.1, CUDA 12.x

B.3 Dataset Profiles and Statistics

Table 8 provides the statistical properties and spectral characteristics of the nine benchmark datasets. These datasets represent a diverse range of domains, including energy, finance, and meteorology, with varying degrees of seasonality and noise.

Table 8: **Dataset Statistics and Spectral Profiles.** 'Freq.' denotes the sampling interval. The spectral profile describes the dominant signal characteristics identified via Fourier analysis.

Dataset	Channels	Length	Freq.	Domain	Spectral Profile
ETTh1	7	17,420	1h	Electricity	Daily/Weekly periodicity, moderate noise
ETTh2	7	17,420	1h	Electricity	Smooth, low-frequency dominant
ETTm1	7	69,680	15min	Electricity	Multi-scale seasonal harmonics
ETTm2	7	69,680	15min	Electricity	High volatility, broadband components
ECL	321	26,304	1h	Electricity	Sharp daily spikes, high periodicity
Weather	21	52,696	10min	Meteorology	Heterogeneous, multi-variate trends
Exchange	8	7,588	1day	Finance	Long-range trends, regime shifts
Traffic	862	17,544	1h	Transport	Complex, non-stationary patterns
Solar	137	52,560	10min	Energy	Bimodal, strong diurnal cycles

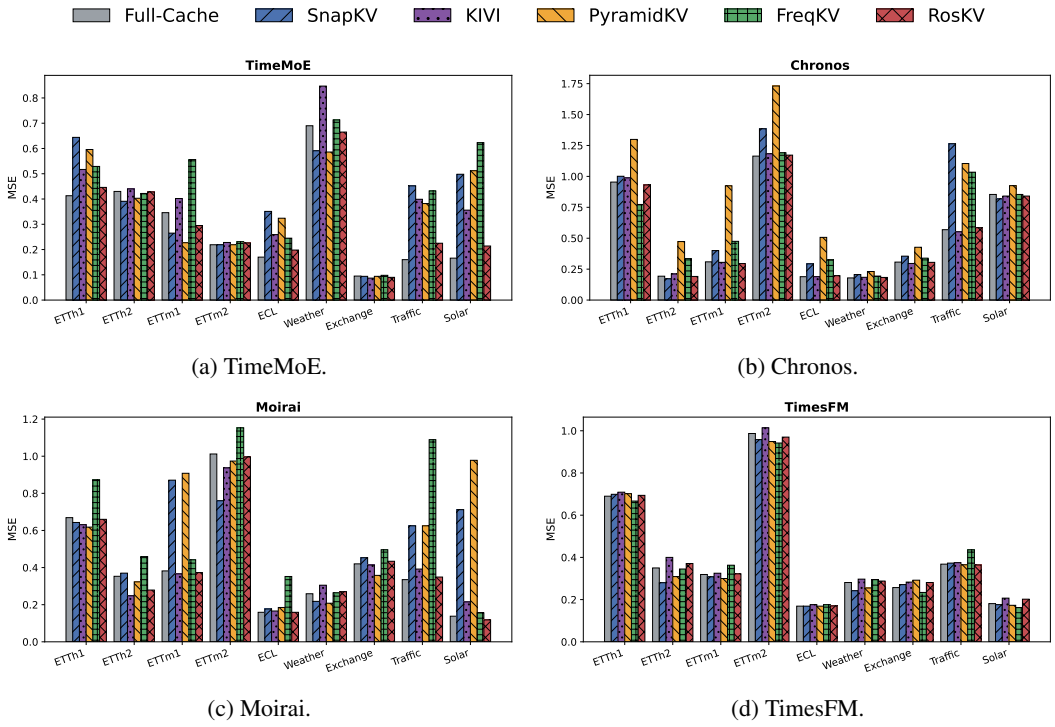
C Full Experiment Results

C.1 Per-Dataset Robustness of RosKV Across Backbones

We provide a granular evaluation of RosKV against four compressors (SnapKV, KIVI, PyramidKV, FreqKV†) across nine benchmark datasets and four TSFM backbones. All experiments utilize a look-back window of L=512, with results in Table 9 representing the average MSE/MAE across horizons $H \in \{96, 192, 336, 720\}$. Figure 3 visualizes these cross-model performance envelopes.

Heterogeneous outcomes. On Moirai/Solar, RosKV reports MSE 0.118 versus Full-Cache's 0.138, illustrating the gains obtained by retaining the dominant spectral components. Results across the remaining datasets are reported in full in Table 9.

Scope of the comparison. The eviction results use the token-eviction baselines, and the encoder results concern attention modification. TimesFM's $H = 96$ predictions precede compressed-cache access; longer-horizon means include the same initial predictions.



(a) TimeMoE. (b) Chronos.

(c) Moirai. (d) TimesFM.

Figure 3: **Per-dataset MSE comparison across the four backbones.**

C.2 Multi-seed TimeMoE Evaluation

Table 1's TimeMoE column uses one 20-window block (seed 2027). We rerun all six cache settings on the same nine datasets and four horizons on two further held-out blocks that start 20 and 40 windows later (seeds 2028 and 2029). Look-back, r=0.5 eviction matching, channels, and normalization are unchanged; RosKV's Rademacher signs use the same seed as the window block. Table 10 reports the nine-dataset MSE on every horizon of every seed. Seed 2027 reproduces Table 1's TimeMoE numbers. Across the three blocks, RosKV stays the lowest compressed average (0.342, +5.5% vs. Full-Cache 0.325), versus PyramidKV +17.8%, SnapKV +24.6%, KIVI +28.0%, and FreqKV +35.8%.

Table 9: **Per-dataset MSE comparison across the four backbones.**

Method	Dataset	TimeMoE MSE	TimeMoE MAE	Chronos-1 MSE	Chronos-1 MAE	Moirai MSE	Moirai MAE	TimesFM MSE	TimesFM MAE	$\overline{\Delta}$%↓
Full-Cache	ETTh1	0.413	0.432	0.954	0.665	0.669	0.604	0.690	0.613	—
	ETTh2	0.430	0.499	0.193	0.329	0.353	0.397	0.350	0.449	—
	ETTm1	0.346	0.400	0.309	0.398	0.382	0.438	0.319	0.430	—
	ETTm2	0.219	0.283	1.164	0.776	1.012	0.737	0.987	0.715	—
	ECL	0.170	0.241	0.189	0.240	0.159	0.217	0.169	0.226	—
	Weather	0.690	0.692	0.179	0.315	0.259	0.397	0.281	0.395	—
	Exchange	0.095	0.232	0.307	0.321	0.420	0.391	0.257	0.299	—
	Traffic	0.160	0.154	0.569	0.307	0.335	0.230	0.368	0.240	—
	Solar	0.166	0.256	0.854	0.504	0.138	0.147	0.181	0.203	—
	Avg	0.299	0.354	0.524	0.428	0.414	0.395	0.400	0.397	—
SnapKV	ETTh1	0.644	0.564	1.001	0.684	0.643	0.613	0.699	0.618	+14.6%
	ETTh2	0.391	0.462	0.172	0.319	0.370	0.419	0.280	0.405	-8.8%
	ETTm1	0.265	0.370	0.400	0.460	0.871	0.581	0.308	0.419	+32.7%
	ETTm2	0.219	0.280	1.385	0.885	0.760	0.656	0.958	0.707	-2.2%
	ECL	0.351	0.367	0.293	0.301	0.177	0.228	0.169	0.226	+43.2%
	Weather	0.591	0.634	0.206	0.346	0.218	0.362	0.242	0.371	-7.2%
	Exchange	0.094	0.229	0.355	0.340	0.453	0.398	0.271	0.305	+7.0%
	Traffic	0.452	0.370	1.264	0.429	0.625	0.340	0.373	0.254	+98.1%
	Solar	0.498	0.527	0.818	0.629	0.712	0.491	0.176	0.216	+152.2%
	Avg	0.390	0.423	0.655	0.488	0.536	0.454	**0.386**	**0.391**	+20.3%
KIVI	ETTh1	0.517	0.544	0.988	0.682	0.631	0.605	0.709	0.623	+6.5%
	ETTh2	0.440	0.491	0.211	0.341	0.248	0.361	0.400	0.476	-1.0%
	ETTm1	0.402	0.483	0.305	0.395	0.366	0.430	0.325	0.430	+3.1%
	ETTm2	0.228	0.294	1.183	0.787	0.937	0.714	1.014	0.731	+0.3%
	ECL	0.259	0.325	0.189	0.239	0.166	0.224	0.176	0.234	+15.2%
	Weather	0.847	0.775	0.184	0.317	0.305	0.427	0.297	0.404	+12.3%
	Exchange	0.087	0.218	0.296	0.319	0.414	0.385	0.283	0.309	-0.8%
	Traffic	0.399	0.353	0.552	0.305	0.392	0.260	0.375	0.247	+41.3%
	Solar	0.356	0.439	0.841	0.492	0.215	0.223	0.207	0.224	+45.8%
	Avg	0.393	0.436	0.528	0.431	0.408	0.403	0.421	0.409	+9.6%
PyramidKV	ETTh1	0.596	0.545	1.299	0.848	0.618	0.616	0.702	0.618	+18.6%
	ETTh2	0.403	0.471	0.473	0.509	0.323	0.420	0.309	0.424	+29.6%
	ETTm1	0.227	0.344	0.924	0.677	0.908	0.680	0.300	0.418	+74.1%
	ETTm2	0.219	0.278	1.731	1.002	0.974	0.720	0.949	0.707	+10.3%
	ECL	0.324	0.350	0.507	0.477	0.185	0.235	0.169	0.225	+68.8%
	Weather	0.586	0.633	0.231	0.363	0.208	0.340	0.255	0.379	-3.7%
	Exchange	0.094	0.228	0.427	0.360	0.358	0.362	0.292	0.312	+9.2%
	Traffic	0.382	0.327	1.104	0.638	0.625	0.347	0.365	0.249	+79.6%
	Solar	0.512	0.517	0.925	0.752	0.978	0.765	0.173	0.211	+205.3%
	Avg	0.371	0.410	0.847	0.625	0.575	0.498	0.390	0.394	+30.5%
FreqKV	ETTh1	0.529	0.516	0.773	0.596	0.873	0.611	0.667	0.602	+9.1%
	ETTh2	0.422	0.484	0.335	0.426	0.459	0.430	0.345	0.444	+25.1%
	ETTm1	0.556	0.511	0.475	0.449	0.442	0.497	0.363	0.431	+36.0%
	ETTm2	0.231	0.290	1.191	0.788	1.153	0.782	0.942	0.708	+4.3%
	ECL	0.245	0.288	0.324	0.347	0.352	0.364	0.176	0.231	+60.3%
	Weather	0.713	0.695	0.194	0.330	0.264	0.400	0.295	0.388	+4.7%
	Exchange	0.098	0.230	0.339	0.347	0.496	0.408	0.234	0.290	+5.7%
	Traffic	0.432	0.329	1.034	0.529	1.089	0.587	0.436	0.285	+123.8%
	Solar	0.623	0.554	0.854	0.511	0.156	0.195	0.162	0.222	+69.5%
	Avg	0.428	0.433	0.613	0.480	0.587	0.475	0.402	0.400	+25.6%
RoSKV	ETTh1	0.446	0.488	0.931	0.655	0.660	0.604	0.694	0.617	+1.2%
	ETTh2	0.428	0.502	0.191	0.331	0.279	0.373	0.371	0.460	-4.1%
	ETTm1	0.295	0.373	0.295	0.387	0.373	0.432	0.322	0.428	-5.2%
	ETTm2	0.226	0.292	1.172	0.779	0.996	0.731	0.970	0.711	+0.1%
	ECL	0.198	0.277	0.197	0.245	0.159	0.217	0.170	0.226	+5.3%
	Weather	0.665	0.673	0.183	0.318	0.270	0.403	0.288	0.395	+1.3%
	Exchange	0.090	0.223	0.305	0.322	0.434	0.391	0.281	0.306	+1.7%
	Traffic	0.225	0.248	0.585	0.318	0.349	0.238	0.364	0.250	+11.6%
	Solar	0.214	0.317	0.841	0.503	0.118	0.142	0.202	0.237	+6.1%
	Avg	**0.310**	**0.377**	**0.522**	**0.429**	**0.405**	**0.392**	0.407	0.403	+0.6%

Table 10: **Multi-seed baseline comparison evaluation.**

Seed	H	Full-Cache	SnapKV	KIVI	PyramidKV	FreqKV	RosKV
2027	96	0.173	0.208	0.255	0.201	0.245	**0.197**
	192	0.281	0.375	0.362	0.361	0.370	**0.288**
	336	0.344	0.446	0.438	0.422	0.472	**0.345**
	720	0.397	0.529	0.516	0.502	0.623	**0.404**
2028	96	0.222	0.257	0.293	0.243	0.278	**0.225**
	192	0.327	0.392	0.405	0.373	0.387	**0.339**
	336	0.364	0.456	0.455	0.427	0.466	**0.379**
	720	0.409	0.543	0.522	0.511	0.619	**0.435**
2029	96	0.212	0.260	0.318	**0.248**	0.270	0.255
	192	0.351	0.397	0.424	**0.375**	0.418	0.375
	336	0.386	0.447	0.466	0.418	0.485	**0.407**
	720	0.429	0.540	0.532	0.506	0.656	**0.461**
Mean		$0.325_{\pm 0.023}$	$0.404_{\pm 0.013}$	$0.415_{\pm 0.021}$	$0.382_{\pm 0.009}$	$0.441_{\pm 0.015}$	$\mathbf{0.342_{\pm 0.033}}$
Δ		—	+24.6%	+28.0%	+17.8%	+35.8%	**+5.5%**

C.3 Extended Multi-Horizon Efficiency and Memory Analysis

This appendix provides extended details and the complete multi-horizon breakdown for the efficiency and memory footprint evaluation presented in Section 4.3 (Table 2).

TimeMoE exposes a persistent decoder self-attention cache and serves as our efficiency probe (Chronos uses cross-attention; Moirai and TimesFM use self-attention with compressed prefill plus an exact decode buffer). We evaluate TimeMoE on ETTh1, ETTm2, and Weather across horizons $H \in \{96, 192, 336, 720\}$ (10 windows $\times$ 5 channels; per-chunk decode traces logged on an exclusive NVIDIA A100 GPU) to measure three operational quantities: forecast MSE, end-of-window cache memory, and wall-clock latency. Table 11 presents the full per-horizon evaluation across all seven methods (including KIVI).

Table 11: **Complete multi-horizon efficiency and memory footprint** on TimeMoE (look-back L=512, averaged over ETTh1, ETTm2, and Weather). Measured on an exclusive A100 GPU.

Horizon	No-Cache	Full-Cache	SnapKV	KIVI	PyramidKV	FreqKV	RosKV (ours)
Cache Memory (MB ↓)							
$H = 96$	**0.0**	42.5	23.7	17.7	23.7	23.7	**9.5** (-78%)
$H = 192$	**0.0**	47.2	28.5	24.2	28.5	28.5	**14.2** (-70%)
$H = 336$	**0.0**	61.9	43.2	38.9	43.2	43.2	**29.0** (-53%)
$H = 720$	**0.0**	90.2	71.5	67.2	71.5	71.5	**57.3** (-36%)
Avg	**0.0**	60.5	41.7	37.0	41.7	41.7	**27.5** (-55%)
MSE ↓							
$H = 96$	0.246	0.246	**0.199**	0.289	0.204	0.226	0.235
$H = 192$	0.413	0.413	0.441	0.397	0.425	0.429	**0.381**
$H = 336$	0.535	0.535	0.620	0.526	0.592	0.583	**0.510**
$H = 720$	0.525	0.525	0.591	0.522	0.560	0.609	**0.514**
Avg	0.430	0.430	0.463	0.434	0.445	0.462	**0.410**
Latency (ms ↓)							
$H = 96$	103	**81**	86	86	187	170	148
$H = 192$	208	**163**	191	188	172	195	185
$H = 336$	418	270	278	**251**	253	281	338
$H = 720$	2512	**447**	1376	1041	2454	937	861
Avg	810	**240**	483	392	767	396	**383**

Across horizons, Table 11 demonstrates how persistent cache savings scale with generation length. At short horizons (H=96), where the compressed prefill dominates total storage, RosKV reduces persistent cache memory from 42.5 MB to 9.5 MB (a 78% reduction, $> 4.4\times$ smaller than Full-Cache and $2.5\times$ smaller than token eviction). At intermediate horizons, RosKV sustains major memory advantages (H=192: 14.2 MB vs. 47.2 MB, -70%; H=336: 29.0 MB vs. 61.9 MB, -53%; H=720: 57.3 MB vs. 90.2 MB, -36%), averaging 27.5 MB (55% reduction).

Disabling the KV cache entirely (No-Cache) eliminates persistent cache storage (0.0 MB) but triggers costly quadratic recomputation (2512 ms at H=720) while inflating peak memory due to expanding forward activation graphs. Conversely, uncompressed Full-Cache incurs an unsustainable memory footprint (90.2 MB at H=720).

RosKV operates at an advantageous point on this Pareto frontier: across all horizons, it achieves an average MSE of 0.410 (outperforming Full-Cache 0.430 and all compressed baselines), slashes average cache memory by 55% (27.5 MB vs. 60.5 MB), and attains an average latency of 383 ms, running $> 2.1\times$ faster than No-Cache (810 ms) and outperforming all compressed baselines (KIVI 392 ms, FreqKV 396 ms, SnapKV 483 ms, PyramidKV 767 ms). In our unfused reference PyTorch implementation, prefill states are reconstructed via separate operator calls at each decode step (Section 3.5), running competitively while slashing memory by more than half (27.5 MB vs. 60.5 MB). Where DRAM capacity forms a hard constraint, trading modest arithmetic compute for persistent cache savings enables running long-horizon models that would otherwise fail to fit on-chip. Furthermore, systems-level kernel fusions (e.g., fused inverse-spectral reconstruction and FlashAttention) or chunked periodic reconstruction represent direct paths to eliminate PyTorch kernel launch overheads in production deployments.

C.4 Component Ablations Across Backbones

Table 12: **Cross-backbone component ablation, per-horizon. Avg** is the unweighted mean over horizons for that backbone. **Bold** marks the lowest MSE and MAE among w/o spectral filtering, w/o rotation, w/o quantization, and RosKV.

Method	H	TimeMoE		Chronos		Moirai		TimesFM	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Full-Cache	96	0.379	0.413	0.472	0.389	0.329	0.373	0.341	0.351
	192	0.859	0.626	0.748	0.541	0.627	0.536	0.610	0.513
	336	0.990	0.703	0.864	0.629	0.780	0.623	0.680	0.573
	720	1.184	0.829	1.150	0.791	1.067	0.777	0.947	0.732
	Avg	0.853	0.643	0.808	0.587	0.700	0.577	0.645	0.542
w/o spectral filtering	96	0.610	0.531	0.472	0.411	0.562	0.440	**0.341**	**0.351**
	192	0.931	0.679	0.791	0.576	0.735	0.566	**0.595**	**0.509**
	336	1.031	0.725	0.883	0.657	0.817	0.623	0.717	0.592
	720	1.287	0.842	**1.170**	0.822	**1.092**	**0.764**	0.995	0.746
	Avg	0.965	0.694	0.829	0.616	0.802	0.598	0.662	0.550
w/o rotation	96	0.406	0.439	0.419	0.380	0.514	0.401	**0.341**	**0.351**
	192	0.789	0.624	0.747	0.547	0.589	0.508	0.610	0.516
	336	0.895	0.679	0.872	0.633	0.751	0.604	0.709	0.590
	720	1.142	**0.800**	1.225	0.803	1.413	0.871	0.999	0.754
	Avg	0.808	0.636	0.816	0.591	0.817	0.596	0.665	0.553
w/o quantization	96	0.330	0.386	0.408	0.374	0.501	0.395	0.341	0.351
	192	0.752	0.589	0.736	0.544	0.554	0.503	0.604	0.512
	336	0.936	0.693	0.868	0.629	0.749	0.610	0.685	0.576
	720	1.176	0.830	1.183	0.792	1.356	0.846	0.982	0.744
	Avg	0.798	0.625	0.799	0.585	0.790	0.588	0.653	0.546
RosKV	96	**0.319**	**0.386**	**0.404**	**0.374**	**0.499**	**0.394**	**0.341**	**0.351**
	192	**0.739**	**0.582**	**0.728**	**0.543**	**0.533**	**0.490**	0.602	0.513
	336	**0.867**	**0.664**	**0.858**	**0.628**	**0.727**	**0.599**	**0.688**	**0.579**
	720	**1.118**	0.808	1.201	**0.795**	1.333	0.841	**0.955**	**0.733**
	Avg	**0.760**	**0.610**	**0.798**	**0.585**	**0.773**	**0.581**	**0.646**	**0.544**

This appendix reports the per-dataset component ablations summarized in Table 3 (Section 4.4) on ETTh1, ETTm2, and Exchange, a three-dataset subset. Full-Cache absolute MSE is not comparable to the nine-dataset averages in Table 1. We use the same look-back length L=512, the four forecast

horizons in the main study, paired held-out windows, and the same normalization and scoring procedure. TimeMoE exposes a persistent prefill cache, whose stored representation is shortened by compression. Chronos-1 exposes a persistent decoder cross-attention cache. Moirai and TimesFM expose a persistent decoder self-attention cache in the main-study harness; the component-ablation cells below report MSE only.

The comparison includes Full-Cache, complete RosKV, RosKV without the RFFT, and RosKV without the Hadamard rotation. The no-rotation condition retains the same adaptive frequency selection and mixed precision. The without-RFFT condition retains the rotation, the adaptive energy-ranked K/V selection, and the adaptive NF4/NF2 tiers, but ranks and stores temporal positions instead of frequency bins; a retained bin costs two real coefficients while a retained position costs one, so its budget is twice as many positions and it is iso-memory with the complete codec to within one percent. Within each backbone the three compressed variants share one target retention.

Table 12 gives the per-horizon breakdown underlying the backbone averages in Table 3.

Backbone-level averages in Table 12 show the two iso-memory removals separate cleanly by backbone. On TimeMoE, removing the RFFT costs $+27.0\%$ relative to complete RosKV, versus $+6.3\%$ for removing the rotation; Chronos-1 shows the same ordering but a milder gap ($+3.9\%$ vs. $+2.3\%$). On Moirai-2.0 and TimesFM-2.5 the ordering flips: removing the rotation costs slightly more than removing the RFFT ($+5.7\%$ vs. $+3.8\%$ on Moirai; $+2.9\%$ vs. $+2.5\%$ on TimesFM). The transform's contribution is therefore backbone-dependent: RFFT-based spectral filtering matters most where the KV axis is long (TimeMoE, Chronos), while the Hadamard rotation contributes comparatively more once the KV axis is already short (Moirai, TimesFM).

C.5 Evaluation on GIFT-Eval

Table 13: **GIFT-Eval summary, per-horizon.** Each horizon row is the unweighted mean of KDD-Cup-2018 and LOOP-Seattle. **Avg** is the unweighted mean over the eight dataset–horizon cells of that backbone; **Total** is the unweighted mean over all 32 cells. **Bold** marks the lowest MSE and MAE among SnapKV, KIVI, PyramidKV, FreqKV, and RosKV.

Backbone	H	Full-Cache		SnapKV		KIVI		PyramidKV		FreqKV		RosKV	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeMoE	96	0.366	0.341	0.451	0.395	0.910	0.590	**0.428**	**0.382**	0.573	0.457	0.444	0.399
	192	0.311	0.352	0.378	0.415	0.658	0.556	**0.348**	**0.394**	0.434	0.444	0.359	0.400
	336	0.255	0.338	0.414	0.461	0.581	0.556	0.367	0.433	0.369	0.434	**0.321**	**0.398**
	720	0.346	0.384	0.617	0.546	0.652	0.585	0.568	0.524	0.434	0.468	**0.415**	**0.445**
	Avg	0.319	0.354	0.465	0.454	0.700	0.572	0.428	0.433	0.452	0.451	**0.384**	**0.411**
Chronos	96	0.508	0.511	0.568	0.524	0.504	0.507	0.774	0.643	**0.485**	**0.505**	0.501	0.507
	192	0.513	0.506	0.566	0.511	0.516	0.500	0.820	0.649	0.681	0.563	**0.500**	**0.498**
	336	0.506	0.488	0.601	0.513	0.526	0.492	0.973	0.710	0.733	0.594	**0.490**	**0.480**
	720	0.668	0.552	0.847	0.607	0.745	0.579	1.227	0.790	0.907	0.671	**0.676**	**0.555**
	Avg	0.549	0.514	0.645	0.539	0.573	0.519	0.948	0.698	0.702	0.583	**0.542**	**0.510**
Moirai	96	0.666	0.491	0.748	0.535	0.513	0.502	0.776	0.545	0.895	0.588	**0.473**	**0.472**
	192	0.679	0.496	0.836	0.561	0.506	0.504	0.862	0.570	1.014	0.622	**0.460**	**0.465**
	336	0.589	0.461	0.744	0.539	0.465	0.480	0.784	0.558	0.953	0.611	**0.398**	**0.417**
	720	0.695	0.512	0.841	0.594	0.583	0.536	0.843	0.594	1.119	0.680	**0.549**	**0.486**
	Avg	0.657	0.490	0.792	0.558	0.517	0.506	0.816	0.567	0.995	0.625	**0.470**	**0.460**
TimesFM	96	0.679	0.507	0.679	0.507	**0.526**	**0.492**	0.679	0.507	0.679	0.507	**0.526**	**0.492**
	192	0.699	0.513	0.745	0.525	0.528	0.501	0.749	0.526	0.776	0.530	**0.514**	**0.495**
	336	0.625	0.488	0.659	0.500	**0.512**	0.489	0.662	0.498	0.728	0.520	0.527	0.493
	720	0.693	0.517	0.718	0.525	0.610	0.533	0.709	**0.524**	0.861	0.581	**0.606**	0.531
	Avg	0.674	0.506	0.700	0.514	0.544	0.504	0.699	0.514	0.761	0.534	**0.543**	**0.503**
Total		0.550	0.466	0.651	0.516	0.583	0.525	0.723	0.553	0.728	0.548	**0.485**	**0.471**

To test whether the cache codec transfers beyond the curated long-term forecasting datasets, we add an evaluation on GIFT-Eval (Aksu et al., 2024). We use the hourly *KDD-Cup-2018* air-quality series and *LOOP-Seattle* traffic series, retaining the public train/test ordering and applying the same

look-back (L=512), horizons ($H \in \{96, 192, 336, 720\}$), and instance normalization as the main study. Each dataset is evaluated with TimeMoE, Chronos-1, Moirai-2.0, and TimesFM-2.5. The comparison covers Full-Cache, SnapKV, PyramidKV, the transferred FreqKV baseline, and RosKV where valid results are available. We report MSE together with persistent cache-object MB whenever the backbone exposes a persistent cache; otherwise we report the attention-input result and mark memory as unavailable. Per-cell MAE values are retained in the experiment CSVs.

Table 13 gives the per-horizon breakdown underlying the backbone averages in Table 4.

D Forecast Visualizations and Qualitative Analysis

To interpret the error profiles observed in the aggregate benchmarks, we visualize representative forecast trajectories across the six KV-cache regimes (Full-Cache, SnapKV, KIVI, PyramidKV, FreqKV, and RosKV). Every figure in this section uses the same layout: 6 rows (one per method) $\times$ 4 columns (one per horizon $H \in \{96, 192, 336, 720\}$), drawn from TimeMoE-200M at look-back L=512 and compression ratio r=0.5. Within each panel the grey curve shows the tail of the look-back, the black curve is the ground-truth future, and the coloured curve is the method's prediction.

D.1 Additional Benchmark Datasets

We visualize six benchmarks (ETTh1, ETTm1, ETTm2, ECL, Traffic, Solar) under the same 6-method $\times$ 4-horizon protocol (Full-Cache, SnapKV, KIVI, PyramidKV, FreqKV, RosKV), covering every dataset reported in Table 1 other than ETTh2, Exchange, and Weather. Each figure draws a single representative test window per horizon, showing one segment per dataset rather than an average across windows.

High-frequency periodic data (ETTh1, ETTm1, ETTm2; Figs. 4–6). The hourly and 15-minute ETT splits share strong daily and weekly seasonality. RosKV tracks the dominant cycles cleanly at every horizon, while SnapKV and PyramidKV introduce visible phase drift at longer horizons.

Spectrally rich data (ECL, Traffic; Figs. 7, 8). Electricity load and traffic occupancy contain both broad diurnal cycles and fine-grained transients. All five compressors retain the macro pattern, but Full-Cache and RosKV reproduce sharp peaks more faithfully than the token-eviction baselines.

Strong daily periodicity (Solar; Fig. 9). The displayed Solar window illustrates a daily pattern that is preserved by the spectral representation across horizons.

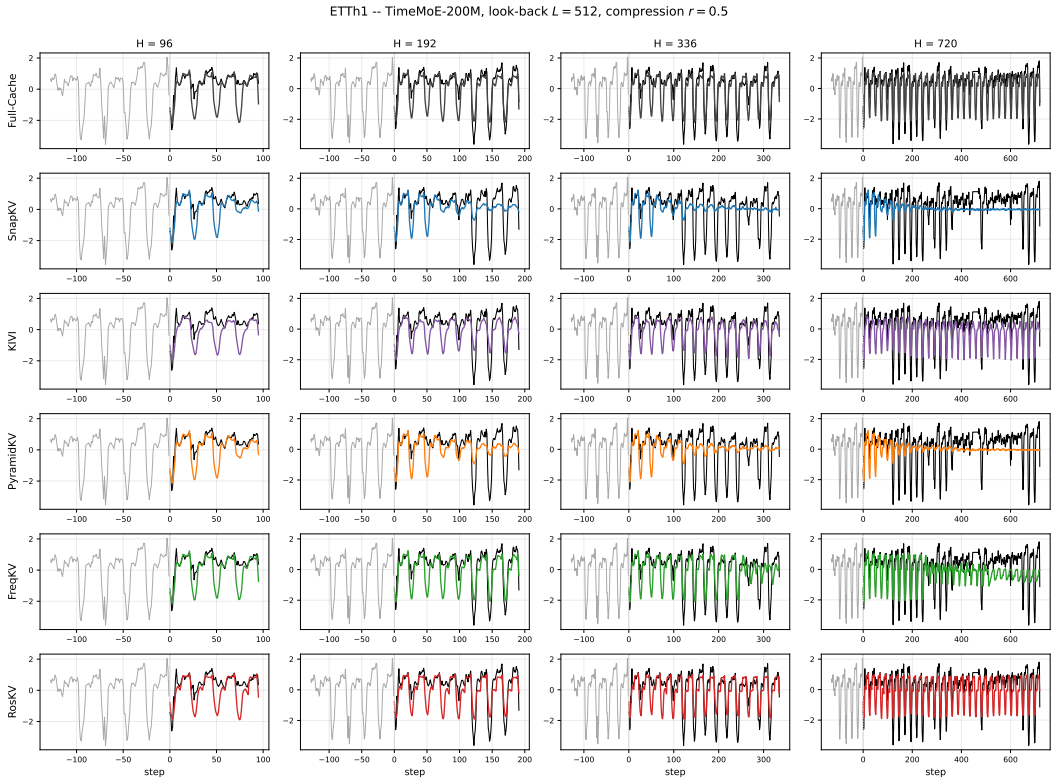


Figure 4: **ETTh1: hourly oil temperature.** 6 methods $\times$ 4 horizons on a single representative test window.

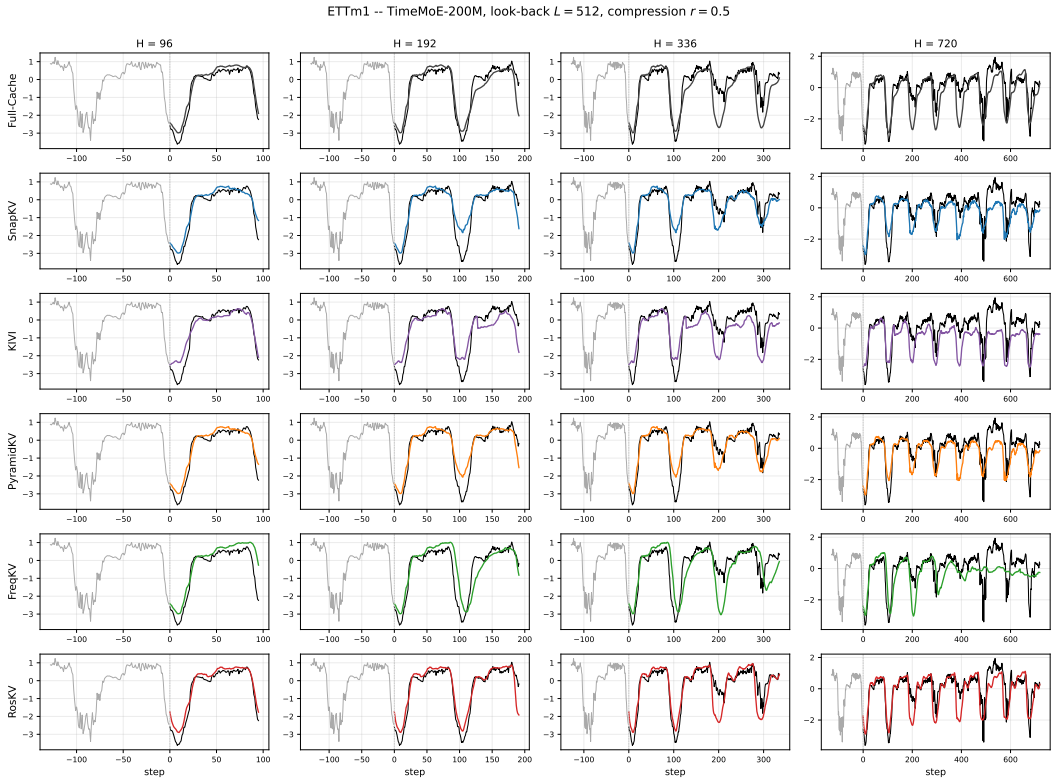


Figure 5: **ETTm1: 15-minute oil temperature.** The shorter sampling interval makes the within-day cycle more visible; RoSKV preserves it across horizons.

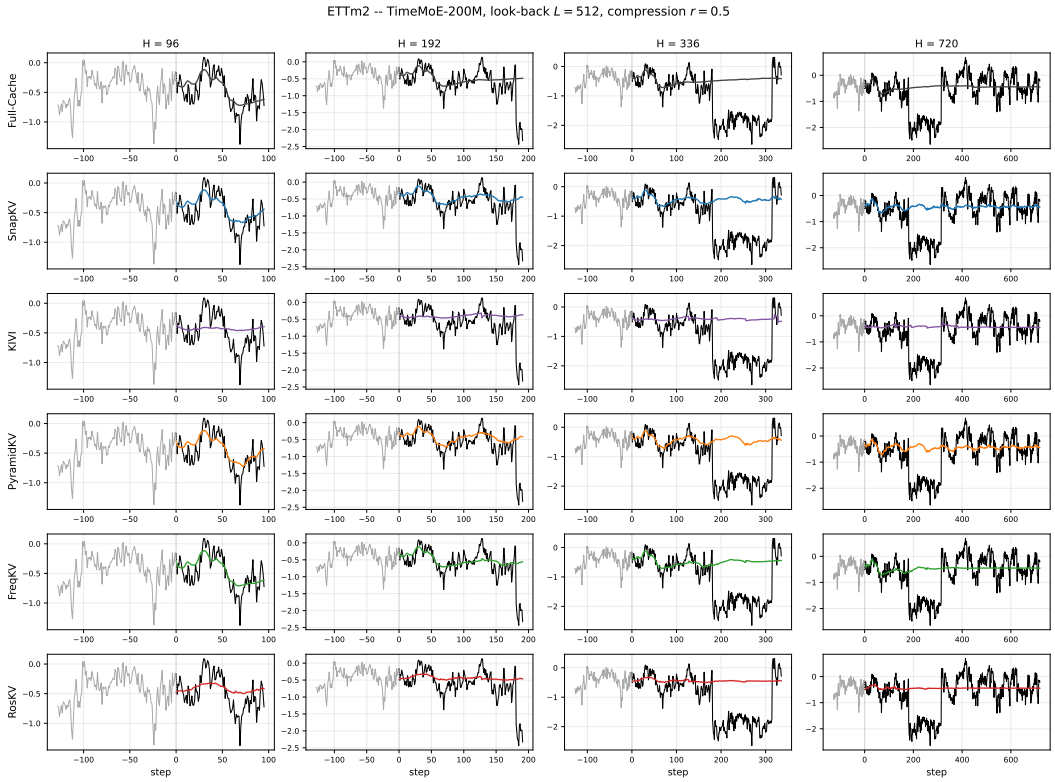


Figure 6: **ETTm2: 15-minute oil temperature.** A noisier sister dataset to ETTm1; spectral truncation changes the forecast, illustrating one forecast window rather than an aggregate accuracy guarantee.

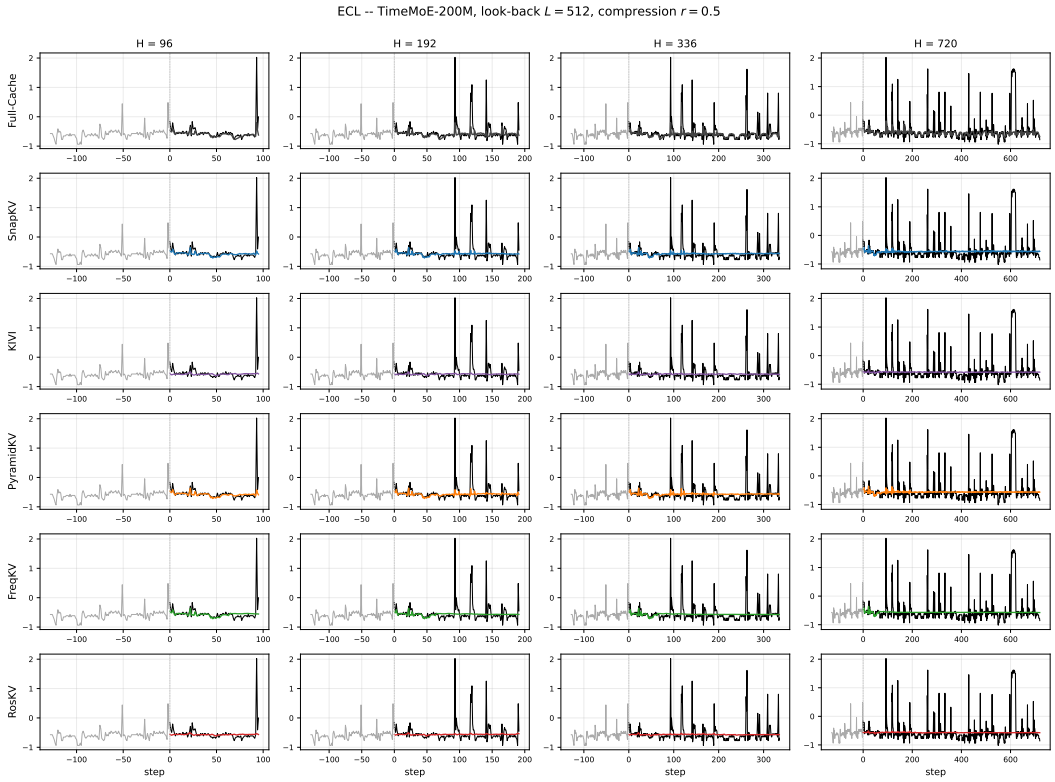


Figure 7: **ECL: hourly electricity load.** A spectrally rich signal with both diurnal and sub-daily transients; all methods retain the macro pattern, with RoSKV preserving sharper peaks than the token-eviction baselines.

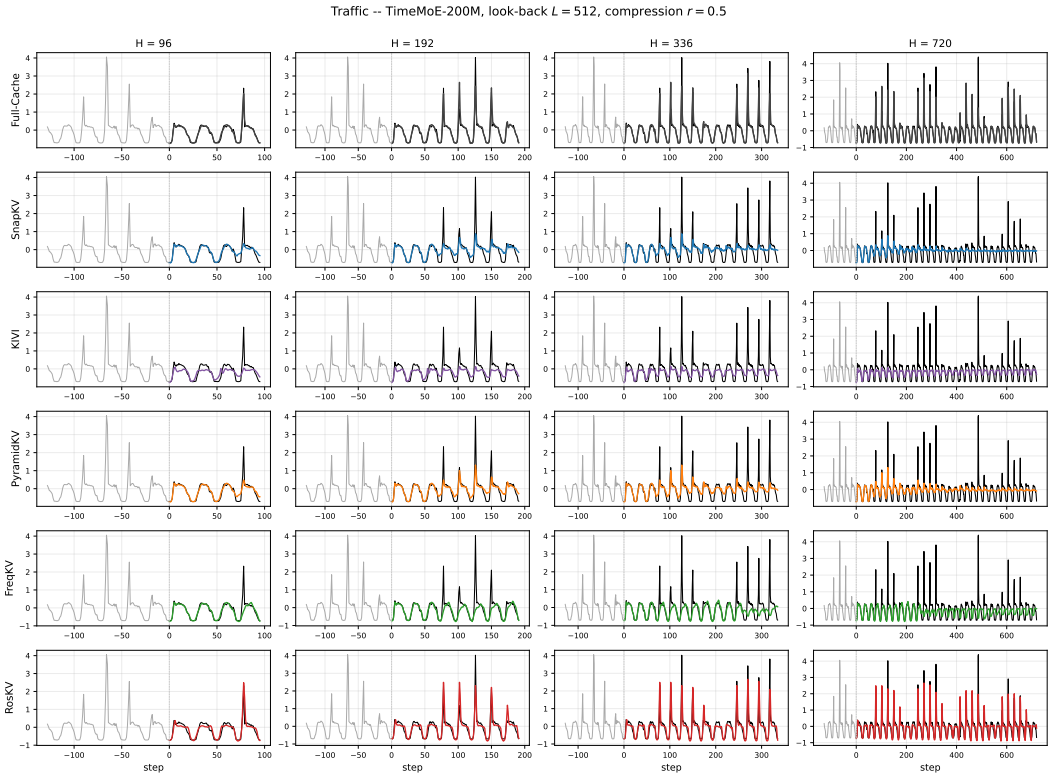


Figure 8: **Traffic: hourly road occupancy.** Strong commute-time peaks at 07:00 and 17:00; RosKV reproduces both reliably across all horizons.

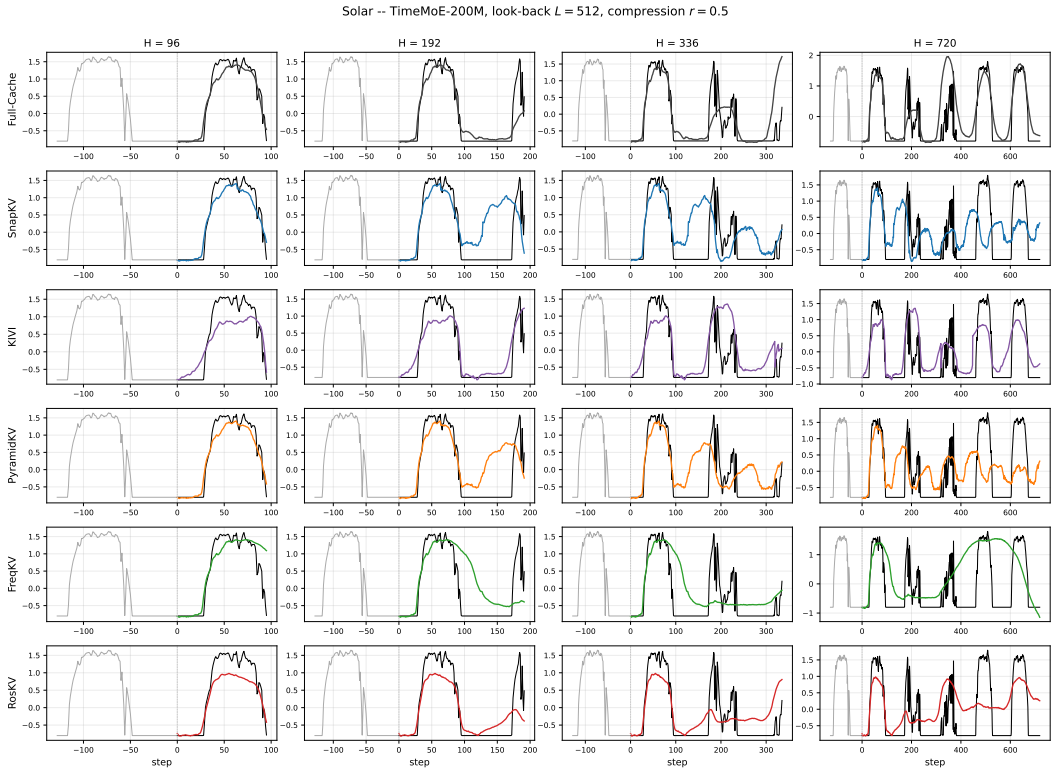


Figure 9: **Solar: 10-minute photovoltaic generation.** A representative forecast window illustrating the daily pattern across horizons.