SPECSPIN: SPECTRAL TRUNCATION AND ROTATION FOR 4-BIT ZERO-SHOT TIME-SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series foundation models (TSFMs) forecast zero-shot across domains, but they are expensive to serve, and 4-bit weight quantization, the standard remedy borrowed from large language models, is not free for them: the autoregressive loop compounds rounding error over long horizons. We show that TSFM attention has spectral structure that a quantizer can exploit. Along the key/time axis, pre-softmax score rows are smooth and recency-biased, so most of their energy sits in the lowest DCT frequencies; along the input-feature axis, the q/k/v projections concentrate their energy in a small set of DCT directions. We introduce **SpecSpin** (Spectral Spin quantization), a training-free, calibration-free 4-bit recipe built on both observations. SpecSpin truncates each q/k/v projection to its highest-energy DCT directions, applies a Haar rotation inside the kept subspace, quantizes all linear layers to 4-bit NormalFloat (NF4), and low-pass filters every attention score row before the softmax. On three architecturally distinct TSFMs (Chronos, TimeMoE, Moirai), six benchmarks and four horizons, SpecSpin is the best 4-bit method on overall average for all three backbones and lowers average MSE below the BF16 model on each (-0.9%, -5.5%, -5.2%, versus -0.1%, -0.6%, -3.5% for uniform 4-bit quantization). Its advantage grows with the horizon, reaching -9.0% (TimeMoE) and -7.7% (Moirai) at $H{=}720$.

1 INTRODUCTION

Time-series forecasting underpins decisions in energy, finance, climate, transportation, healthcare and cloud operations. Time-series foundation models (TSFMs) such as Chronos (Ansari et al., 2024), TimeMoE, Moirai (Woo et al., 2024) and TimesFM are pretrained on large, diverse corpora and forecast new series zero-shot, so a single model can replace many task-specific forecasters. Much of this forecasting, however, has to run close to where the data are produced: on edge servers, gateways and embedded devices that monitor grids, vehicles or patients, where memory is scarce and data often cannot leave the device. There, storage is the binding constraint. In BF16, Chronos-2 occupies 228 MB and TimeMoE-200M 864 MB, and larger TSFMs keep appearing. Compressing the weights with post-training quantization, which needs no retraining, is therefore a prerequisite for putting TSFMs on the edge.

Post-training quantization has matured rapidly for large language models (LLMs). Weight-only methods such as GPTQ and AWQ (Lin et al., 2024) choose rounding or scaling with the help of calibration data, NormalFloat matches the quantization grid to bell-shaped weights, and rotation-based methods such as QuIP (Chee et al., 2023), QuaRot (Ashkboos et al., 2024) and SpinQuant (**?**) rotate weights to spread outliers before rounding. These methods are built around the statistics of language: they treat every weight matrix as an unstructured collection of numbers and every token as equally informative. Time series violate both assumptions, which raises three challenges for quantizing TSFMs. First, time series are *periodic and smooth*, so their energy concentrates in a few spectral components, and the attention computations that process them inherit this concentration; a quantizer that treats all directions alike spends much of its bit budget on directions the model barely uses. Second, forecasts depend most on *recent fluctuations*, so TSFM attention is strongly recency-biased and its score rows vary smoothly with the lag; rounding error perturbs these smooth patterns with high-frequency noise, and in autoregressive decoding the perturbed prediction is fed

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.

back as input, so the error compounds over the horizon. Third, real measurements are *noisy*, largely at high frequencies, so quantization noise adds to noise already present in the input rather than to a clean signal, which leaves less margin for rounding error than in language.

These gaps show up in practice: in our experiments, NLP-style quantization does not reliably preserve TSFM accuracy. Uniform 4-bit quantization raises MSE on four of six benchmarks for TimeMoE and by 10.7% on average for TimesFM, and at 3 bits the Chronos error on ETTh1 grows by 73%.

The same structure that NLP quantizers ignore is what makes TSFMs compressible. We find that pretrained TSFM attention is *spectrally sparse along two axes*. Along the key/time axis, score rows are smooth and recency-biased: across our six benchmarks, as little as 1% (Exchange) to 12% (Weather) of the lowest DCT frequencies retain 90% of a row's energy, and on ETTh1 a representative TSFM row needs 0.2% of its coefficients where a GPT-2 row needs 7.7%. Along the input-feature axis, the q/k/v projections concentrate their energy in a small, scattered set of DCT directions. Both axes can be compressed with a fixed, data-free transform.

Building on this observation, we propose **SpecSpin** (Spectral Spin quantization), a training-free and calibration-free 4-bit recipe that compresses a TSFM along its spectral structure in three stages, each aimed at one of the challenges above (Figure 1). (i) A *dual-axis frequency transform*. On the feature axis, each q/k/v projection moves to the DCT basis and keeps only its highest-energy directions, so the quantizer no longer spends bits on directions that carry little of the signal. On the temporal axis, every attention score row is low-pass filtered in the DCT domain before the softmax, targeting the high-frequency perturbations that rounding and noisy inputs add to smooth, recency-biased attention. (ii) A *spin*: a Haar-random rotation inside the retained subspace spreads the remaining energy evenly before rounding, following rotation-based LLM quantizers. (iii) *Quantization* of the spun projections and all other linear layers to 4-bit NormalFloat. The recipe needs no calibration data, no fine-tuning and no change to the KV cache, and it stores the model at about a quarter of its BF16 size.

We evaluate SpecSpin on three architecturally distinct TSFMs (Chronos-2, an encoder-only model with quantile outputs; TimeMoE, a mixture-of-experts autoregressive decoder; and Moirai 2.0, a compact decoder-only model), six long-term forecasting benchmarks and four horizons. SpecSpin has the lowest average MSE of the 4-bit methods on every backbone and is below BF16 on each, so compression to 4 bits costs no accuracy on average. Its advantage widens with the horizon, reaching 9.0% (TimeMoE) and 7.7% (Moirai) below BF16 at $H{=}720$, which is where compounding error matters most.

Our contributions are as follows:

- To the best of our knowledge, this is the first systematic study of post-training quantization for time-series foundation models. We identify the properties of time series that NLP quantizers ignore and show that they make TSFM attention spectrally sparse along both the temporal and the feature axis.
- We design SpecSpin, a training-free and calibration-free 4-bit quantization method for TSFMs that combines a dual-axis frequency transform, a subspace spin and NF4 quantization.
- Evaluations on three architecturally distinct TSFMs (Chronos, TimeMoE and Moirai), six long-term forecasting benchmarks and four horizons show that SpecSpin achieves the best accuracy among 4-bit methods on every backbone, 0.9%, 5.5% and 5.2% below BF16 on average at a quarter of the weight memory, with gains that grow with the forecast horizon.

2 METHOD: SPECSPIN

SpecSpin compresses a frozen TSFM in three stages that follow the spectral structure of its attention: a *dual-axis frequency transform* (a DCT along the feature axis of the q/k/v weights and along the key/time axis of the attention scores), a *spin* (a Haar rotation inside the retained feature subspace), and *quantization* to 4-bit NormalFloat. We first fix notation (§2.1), then measure the two spectral axes (§2.2), give the overall pipeline (§2.3), and describe each stage (§2.4–§2.6).

2.1 PRELIMINARIES

Zero-shot forecasting. A pretrained TSFM is a frozen map f_θ that turns a window $\mathbf{x}_{t-L+1:t} \in \mathbb{R}^{C\times L}$ into an H-step forecast

$$\hat{\mathbf{x}}_{t+1:t+H} \;=\; f_\theta(\mathbf{x}_{t-L+1:t}) \in \mathbb{R}^{C\times H}, \tag{1}$$

with no dataset-specific training (L context length, C channels, H horizon). Autoregressive backbones feed each prediction back as input, so any perturbation of f_θ is applied repeatedly over the horizon. We use $L{=}512$ and $H \in \{96, 192, 336, 720\}$.

Attention. Each Transformer block computes

$$S \;=\; \frac{QK^\top}{\sqrt{d_h}}, \qquad A \;=\; \mathrm{softmax}(S+M)\,V, \tag{2}$$

with $Q{=}xW_Q^\top$, $K{=}xW_K^\top$, $V{=}xW_V^\top$, projections $W_{\{Q,K,V\}} \in \mathbb{R}^{d\times d_{\text{in}}}$, layer input $x \in \mathbb{R}^{B\times L_q\times d_{\text{in}}}$, head dimension d_h and mask M. Two axes of this computation carry spectral structure: the *feature axis* d_{in} of the static weights W, and the *temporal (key) axis* L_k of the transient scores $S \in \mathbb{R}^{B\times h\times L_q\times L_k}$.

Weight quantization. Post-training quantization replaces each weight with a b-bit code and a per-group scale; we write $\mathcal{Q}(W)$ for the quantized weight. Our baseline is group-wise round-to-nearest on a uniform grid (group size 128; labeled AWQ-4bit in the tables). SpecSpin uses NormalFloat-4 (NF4) (**?**), whose 16 levels are quantiles of a standard normal and therefore suit bell-shaped weight distributions. Weight-only quantization leaves S in full precision, so rounding error reaches the forecast only through the attention pattern and, for autoregressive models, the feedback loop.

2.2 DUAL-AXIS SPECTRAL STRUCTURE OF TSFM ATTENTION

For a vector $\mathbf{u} \in \mathbb{R}^N$ with orthonormal DCT-II coefficients $\hat{\mathbf{u}}$, let $\rho^*_{\text{LPF}}(\mathbf{u};\tau)$ be the smallest fraction of *lowest-index* coefficients that keeps a fraction τ of $\|\hat{\mathbf{u}}\|_2^2$, and $\rho^*_{\text{topk}}(\mathbf{u};\tau)$ the same fraction for the *largest-magnitude* coefficients. Similar values mean the energy sits at low indices; $\rho^*_{\text{topk}} \ll \rho^*_{\text{LPF}}$ means it is concentrated but scattered. The two attention axes fall on opposite sides of this distinction, and the transform in §2.4 uses a different selection rule on each.

Temporal axis: score rows are low-frequency. Time series are autocorrelated, so the similarity between a query and a distant key changes slowly with the lag, and positional encodings and causal masking add a recency prior. Each score row is therefore a smooth function of key position whose DCT spectrum decays with index. We verify this on four pretrained TSFMs. Across our six benchmarks, $\rho^*(0.9)$ ranges from 0.010 (Exchange) to 0.121 (Weather), and the concentration survives removing the row mean (the DC term, to which softmax is invariant): after centering, TSFM rows keep over 88% of their AC energy in the lowest 10–15% of frequencies, whereas GPT-2 rows are broadband. We group benchmarks into G1 ($\rho^*{<}0.05$) and G2 ($0.05{\le}\rho^*{\le}0.15$); these thresholds are descriptive and were set after inspecting the suite.

Feature axis: weight energy is top-k concentrated. The input-feature axis of $W_{\{Q,K,V\}}$ has no temporal order, so no low-frequency bias is expected. Changing to the DCT basis, $W \to WF$, leaves the layer's function unchanged but concentrates column energy in a subset of bins that is not a low-index prefix: at matched retention, energy-ranked selection keeps more of $\|W\|_F^2$ than index-prefix selection (Appendix D.3). We use the DCT here as a fixed, data-free orthonormal basis, not as a frequency decomposition.

2.3 OVERVIEW: TRANSFORM, SPIN, QUANTIZE

Figure 1 summarizes SpecSpin (pseudocode in Appendix Algorithm 1). At load time, each q/k/v projection is transformed to the DCT basis and truncated to its $k{=}\lceil \rho_W d_{\text{in}} \rceil$ highest-energy directions $F_{\mathcal{S}}$, spun by a Haar rotation $R \in \mathbb{R}^{k\times k}$, and quantized to NF4; every other linear layer is quantized to NF4 directly. At run time, the input is mapped into the same spun subspace, and each score row is low-pass filtered in the DCT domain before the softmax:

$$\begin{aligned} \widetilde{W} &= \mathcal{Q}_{\text{NF4}}\big(WF_{\mathcal{S}}R\big), \qquad \widetilde{Q} = (xF_{\mathcal{S}}R)\,\widetilde{W}_Q^\top \;\;(\text{same for } K,V), \\ A_{\text{SpecSpin}} &= \mathrm{softmax}\big(\mathcal{T}_\rho(\widetilde{Q}\widetilde{K}^\top/\sqrt{d_h}) + M\big)\,\widetilde{V}. \end{aligned} \tag{3}$$

The method has two retention hyperparameters, feature retention ρ_W and score retention ρ, and needs no calibration data, fine-tuning or KV-cache change.

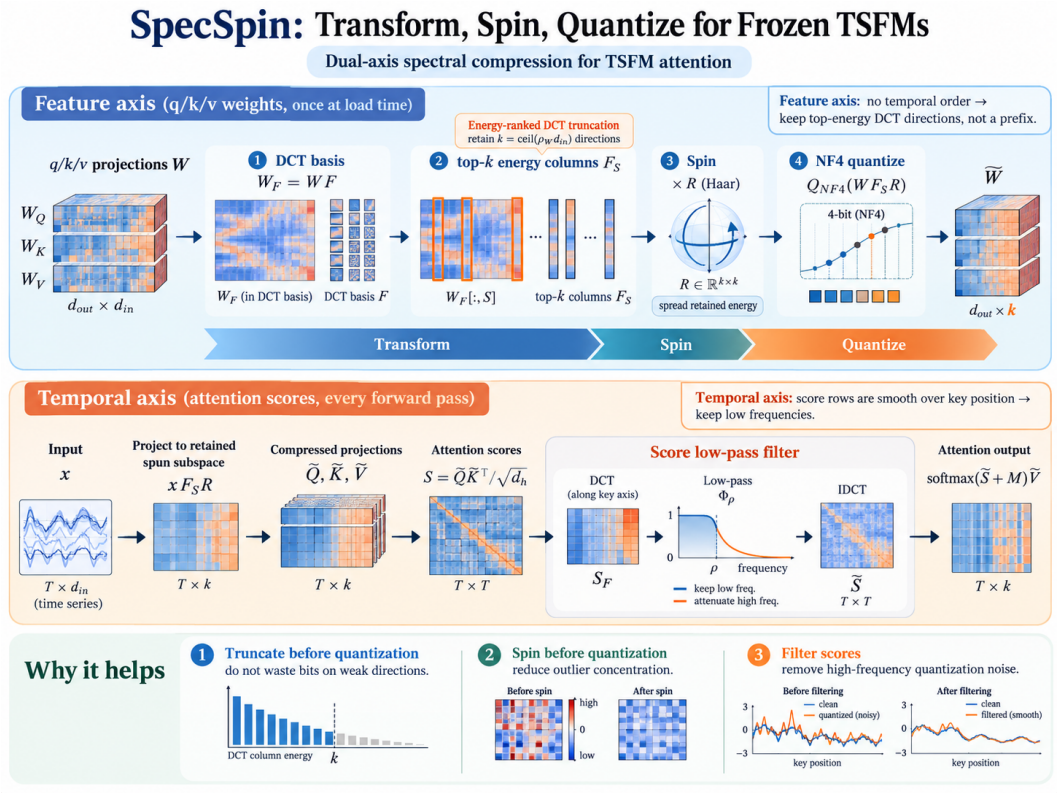


Figure 1: **SpecSpin: transform, spin, quantize.** *Top (feature axis, once at load time):* each q/k/v projection is moved to the DCT basis, truncated to its $k{=}\lceil \rho_W d_{\text{in}} \rceil$ highest-energy directions, spun by a Haar rotation R, and stored in NF4. *Middle (temporal axis, every forward pass):* the input is projected into the same spun subspace, and each score row is low-pass filtered along the key axis in the DCT domain before the softmax. *Bottom:* the intended role of each stage. All other linear layers are stored in NF4.

2.4 Stage 1: Dual-Axis Frequency Transform

Feature axis: energy-ranked DCT truncation. Let $F \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ be the orthonormal DCT-II matrix. For a projection W, let $\mathcal{S}_{\rho_W}(W)$ index the $k{=}\lceil \rho_W d_{\text{in}} \rceil$ columns of WF with the largest ℓ_2 norm and $F_{\mathcal{S}}{=}F[:, \mathcal{S}_{\rho_W}]$. Because F is orthonormal,

$$xW^\top = (xF)(WF)^\top \;\approx\; (xF_{\mathcal{S}})(WF_{\mathcal{S}})^\top, \qquad \left\|W - WF_{\mathcal{S}}F_{\mathcal{S}}^\top\right\|_F^2 = \textstyle\sum_{j \notin \mathcal{S}} \|(WF)_{:,j}\|_2^2, \quad (4)$$

so truncation discards exactly the least energetic directions, and it removes them before quantization, so the quantizer's grid is not spent on them. This step is load-time only and runs on the weights alone.

Temporal axis: score low-pass filter. On each score row we apply the DCT along the key axis, attenuate the high-frequency tail and invert:

$$\widetilde{S} \;=\; \mathcal{T}_\rho(S) \;=\; \mathbf{D}^{-1}(\Phi_\rho(\mathbf{D}S))\,, \qquad A = \mathrm{softmax}(\widetilde{S} + M)\,V. \quad (5)$$

The hard filter Φ_ρ^{LPF} zeroes coefficients beyond index $\lceil \rho L_k \rceil$; for short or bidirectional key axes we use a soft taper that scales coefficient j by $\exp(-j^2/2\sigma^2)$, $\sigma{=}\max(1, \rho L_k)$, keeping the DC term and rolling off gradually. Weight rounding perturbs the scores, $\widetilde{S} = S + \Delta S$; because the clean rows S are low-frequency (§2.2), the filter removes the part of ΔS that lies above the retained band while changing S only slightly.

Proposition 1 (Score filter perturbation bound; sanity check)**.** *If every row's truncation residual is bounded by ϵ (i.e. $\max_i \|\hat{\mathbf{s}}_i - \Phi_\rho(\hat{\mathbf{s}}_i)\|_2 \le \epsilon$), then* $\|\operatorname{softmax}(S{+}M)V - \operatorname{softmax}(\widetilde{S}{+}M)V\|_F \le \frac{L_q}{2}\,\epsilon\,\|V\|_F$.

The proof combines the orthonormality of $\mathbf{D}$, Cauchy–Schwarz over the row residuals and $\|J\|_2 \le \frac{1}{2}$ for the softmax Jacobian $J = \operatorname{diag}(\mathbf{q}) - \mathbf{q}\mathbf{q}^\top$. The bound is loose by about an order of magnitude; we use it only to check that the filter does not amplify error.

2.5 Stage 2: Spin

After truncation, SpecSpin rotates the retained subspace with a Haar-random orthogonal matrix $R \in \mathbb{R}^{k\times k}$ (QR of a seeded Gaussian matrix with sign correction):

$$W_{\text{spin}} = WF_{\mathcal{S}}R, \qquad (xF_{\mathcal{S}}R)\,W_{\text{spin}}^\top = (xF_{\mathcal{S}})\,RR^\top(WF_{\mathcal{S}})^\top = (xF_{\mathcal{S}})(WF_{\mathcal{S}})^\top. \tag{6}$$

The spin is exact before rounding. Its purpose is to change what the quantizer sees: energy-ranked truncation keeps a few high-energy DCT columns, and a random rotation spreads their mass across all k coordinates, so each group of 128 weights has a smaller range relative to its absmax scale, the mechanism behind rotation-based LLM quantizers (Chee et al., 2023; Ashkboos et al., 2024). R depends only on k and the seed, so it is shared across projections and can be regenerated rather than stored. To measure its contribution we also consider a partial spin that rotates only the $\lceil \alpha k \rceil$ highest-energy kept directions (α=0: no spin, α=1: full spin; Appendix D.2).

2.6 Stage 3: Quantize

The spun q/k/v weights and all remaining linear layers (output projections, feed-forward and expert layers) are quantized with NF4. For each group g of 128 consecutive weights with scale $s_g = \max_{i\in g}|w_i|$,

$$\mathcal{Q}_{\text{NF4}}(w_i) = s_g \arg\min_{c\in\mathcal{C}_{\text{NF4}}} \left|w_i/s_g - c\right|, \tag{7}$$

where $\mathcal{C}_{\text{NF4}} \subset [-1,1]$ is the fixed set of 16 normal-quantile levels. Embeddings, normalization layers and expert routers stay in BF16. The order of the stages matters: truncating first means the spin mixes only retained energy, and spinning before quantizing means the rounding error of Eq. (7) is spread over all k retained directions instead of concentrating on the few dominant ones. The residual rounding error then reaches the scores as ΔS, which the temporal-axis filter of Eq. (5) partly removes. As for the baselines, we simulate quantization (quantize, then dequantize) and report the analytical storage footprint.

2.7 Cost and Memory

Storage. Every linear layer is stored at 4 bits plus one BF16 scale per 128 weights, and the q/k/v projections additionally drop the $1-\rho_W$ truncated input columns. Measured on the checkpoints, this gives 65.5 MB for Chronos-2 (BF16 227.9 MB; uniform 4-bit 67.6 MB), 222.4 MB for TimeMoE-200M (864.4; 222.9) and 5.5 MB for Moirai-2-small (21.7; 5.6). SpecSpin is thus 3.5–3.9$\times$ smaller than BF16, and truncation adds at most a 3% saving over uniform 4-bit: its role is accuracy, not further compression. The spin needs one $k \times k$ matrix per distinct k, regenerated from the seed; our reference implementation materializes $F_{\mathcal{S}}R$ per projection in BF16 for simplicity, which a deployment kernel would avoid by applying an FFT-based DCT, an index gather and the shared R.

Compute. The score filter adds two FFT-based DCTs per row, under 1% of FLOPs at L=512 and +1.8% to +2.3% measured wall-clock on an unfused attention path. The input transform $xF_{\mathcal{S}}R$ roughly offsets the matmul saving from truncation. Transient score memory and the KV cache are unchanged, so SpecSpin composes with paged attention and KV-cache compression. The score filter needs the materialized score matrix and is therefore not yet compatible with FlashAttention; a fused DCT–filter–softmax kernel is future work.

3 EXPERIMENTS

Table 1: **Zero-shot MSE across three backbones (Chronos, TimeMoE, Moirai), per horizon and averaged over $H \in \{96, 192, 336, 720\}$, on six datasets.** $\Delta\%$ is vs. BF16; **bold**/underline = best/second-best Avg among the 4-bit methods {AWQ-4bit, FQ-4bit, SpecSpin}. "—" marks uncompleted cells. Per-horizon breakdowns: Appendix C.1.

Method	Dataset	Chronos						TimeMoE						Moirai					
		Horizon				Avg	Δ%	Horizon				Avg	Δ%	Horizon				Avg	Δ%
		96	192	336	720			96	192	336	720			96	192	336	720		
AWQ-16 (BF16)	ETTh1	0.371	0.414	0.439	0.449	0.418	—	0.354	0.395	0.417	0.445	0.403	—	0.378	0.420	0.454	0.501	0.438	—
	ETTh2	0.280	0.351	0.378	0.398	0.352	—	0.276	0.332	0.354	0.399	0.340	—	0.297	0.380	0.411	0.445	0.383	—
	ETTm1	0.308	0.362	0.395	0.484	0.387	—	0.306	0.392	0.457	0.541	0.424	—	0.338	0.403	0.459	0.547	0.436	—
	ETTm2	0.171	0.241	0.306	0.404	0.280	—	0.192	0.264	0.325	0.420	0.300	—	0.180	0.270	0.385	0.537	0.343	—
	Exchange	0.085	0.180	0.334	0.944	0.386	—	0.186	0.357	0.622	1.244	0.602	—	0.090	0.188	0.345	0.885	0.377	—
	Weather	0.156	0.211	0.281	0.383	0.258	—	0.161	0.217	0.276	0.357	0.253	—	0.168	0.216	0.283	0.383	0.263	—
	Avg	*0.228*	*0.293*	*0.356*	*0.510*	*0.347*	—	*0.246*	*0.326*	*0.409*	*0.568*	*0.387*	—	*0.242*	*0.313*	*0.389*	*0.549*	*0.373*	—
AWQ-4bit	ETTh1	0.370	0.412	0.438	0.448	0.417	-0.3%	0.368	0.403	0.423	0.459	0.413	+2.6%	0.373	0.411	0.436	0.448	0.417	-4.7%
	ETTh2	0.280	0.351	0.377	0.396	0.351	-0.2%	0.282	0.334	0.359	0.416	0.348	+2.3%	0.294	0.372	0.406	0.426	0.375	-2.2%
	ETTm1	0.307	0.361	0.396	0.486	0.388	+0.1%	0.305	0.385	0.444	0.527	**0.415**	-2.0%	0.338	0.396	0.444	0.526	0.426	-2.4%
	ETTm2	0.171	0.241	0.307	0.406	0.281	+0.3%	0.194	0.269	0.336	0.434	0.308	+2.6%	0.180	0.259	0.341	0.470	0.312	-8.9%
	Exchange	0.085	0.178	0.332	0.935	0.382	-0.9%	0.180	0.330	0.537	1.197	0.561	-6.9%	0.091	0.181	0.320	0.886	0.370	-1.9%
	Weather	0.157	0.212	0.284	0.391	**0.261**	+1.3%	0.167	0.227	0.286	0.366	**0.262**	+3.5%	0.171	0.219	0.282	0.371	0.261	-0.6%
	Avg	*0.228*	*0.293*	*0.355*	*0.510*	*0.347*	-0.1%	*0.249*	*0.325*	*0.398*	*0.567*	*0.385*	-0.6%	*0.241*	*0.307*	*0.372*	*0.521*	*0.360*	-3.5%
FQ-4bit	ETTh1	0.367	0.412	0.441	0.449	0.417	-0.2%	0.365	0.402	0.425	0.457	**0.412**	+2.2%	0.369	0.408	0.432	0.441	0.413	-5.8%
	ETTh2	0.282	0.355	0.382	0.399	0.355	+0.7%	0.278	0.334	0.359	0.411	**0.345**	+1.6%	0.293	0.371	0.406	0.427	**0.374**	-2.3%
	ETTm1	0.311	0.364	0.400	0.488	0.391	+1.0%	0.330	0.412	0.470	0.540	0.438	+3.2%	0.437	0.489	0.524	0.584	0.508	+16.6%
	ETTm2	0.172	0.242	0.306	0.401	0.280	+0.1%	0.196	0.269	0.327	0.417	**0.302**	+0.7%	0.179	0.259	0.340	0.466	0.311	-9.4%
	Exchange	0.084	0.176	0.325	0.914	**0.375**	-2.9%	0.130	0.278	0.505	1.186	0.525	-12.8%	0.091	0.181	0.320	0.892	0.371	-1.6%
	Weather	0.157	0.214	0.289	0.402	0.265	+2.9%	0.170	0.230	0.290	0.362	0.263	+4.1%	0.171	0.219	0.285	0.388	0.266	+1.1%
	Avg	*0.229*	*0.294*	*0.357*	*0.509*	*0.347*	+0.1%	*0.245*	*0.321*	*0.396*	*0.562*	*0.381*	-1.6%	*0.257*	*0.321*	*0.385*	*0.533*	*0.374*	+0.2%
SpecSpin (ours)	ETTh1	0.362	0.405	0.431	0.433	**0.408**	-2.5%	0.371	0.412	0.442	0.501	0.431	+7.0%	0.372	0.410	0.430	0.430	**0.410**	-6.3%
	ETTh2	0.278	0.344	0.371	0.389	**0.345**	-2.0%	0.280	0.341	0.363	0.405	0.347	+2.2%	0.295	0.369	0.400	0.435	0.375	-2.2%
	ETTm1	0.307	0.359	0.395	0.477	**0.384**	-0.7%	0.340	0.416	0.478	0.567	0.450	+6.1%	0.341	0.393	0.439	0.514	**0.422**	-3.3%
	ETTm2	0.170	0.238	0.302	0.395	**0.276**	-1.4%	0.208	0.280	0.346	0.433	0.317	+5.6%	0.178	0.249	0.320	0.430	**0.294**	-14.3%
	Exchange	0.100	0.195	0.336	0.919	0.388	+0.4%	0.105	0.204	0.358	0.832	**0.374**	-37.8%	0.086	0.182	0.332	0.876	**0.369**	-2.2%
	Weather	0.175	0.227	0.285	0.362	0.262	+1.6%	0.187	0.243	0.300	0.367	0.274	+8.4%	0.168	0.214	0.272	0.358	**0.253**	-3.8%
	Avg	*0.232*	*0.295*	*0.353*	*0.496*	***0.344***	*-0.9%*	*0.249*	*0.316*	*0.381*	*0.517*	***0.366***	*-5.5%*	*0.240*	*0.303*	*0.366*	*0.507*	***0.354***	*-5.2%*

3.1 SETUP

Datasets and backbones. We evaluate on six long-term forecasting benchmarks (ETTh1, ETTh2, ETTm1, ETTm2, Exchange, Weather) with input length L=512 and horizons $H \in \{96, 192, 336, 720\}$, using three pretrained TSFMs with different attention and decoding schemes: Chronos-2 (an encoder-only model with quantile outputs), TimeMoE-200M (a mixture-of-experts autoregressive decoder) and Moirai-2.0-R-small (a decoder-only model with quantile outputs). All runs are zero-shot from the official checkpoints (Appendix B).

Baselines. We compare SpecSpin with the uncompressed **BF16** model; **AWQ-4bit**, group-wise 4-bit round-to-nearest weight quantization (group size 128); and **FQ-4bit**, AWQ-4bit plus the score low-pass filter alone, which isolates the temporal-axis component of SpecSpin.

Environment. All methods run in PyTorch within the Time-Series-Library framework on NVIDIA A100 GPUs, with batch size 8 and simulated (quantize–dequantize) weight quantization; each cell is a single deterministic run.

3.2 MAIN RESULTS

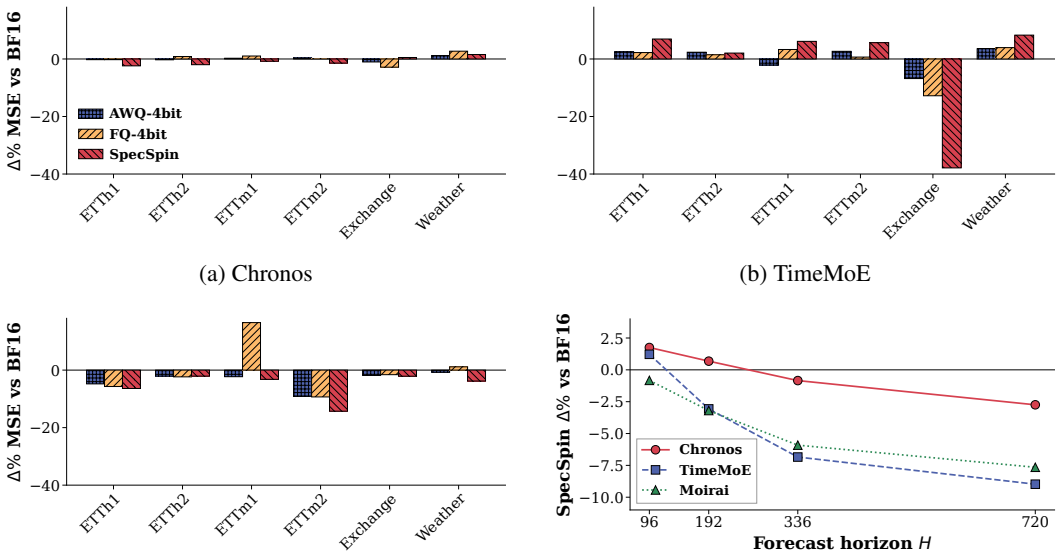
At 4 bits, SpecSpin matches or improves on the full-precision model across three architecturally different TSFMs, six datasets and four horizons, and it does so more consistently than either 4-bit

baseline. Table 1 reports the zero-shot MSE with its change ($\Delta\%$) versus BF16, and Figure 2 summarizes the per-dataset and per-horizon changes. Three observations stand out.

Best 4-bit accuracy on every backbone. SpecSpin has the lowest overall average of the 4-bit methods on Chronos, TimeMoE and Moirai: -0.9%, -5.5% and -5.2% relative to BF16, compared with -0.1%, -0.6% and -3.5% for AWQ-4bit and $+0.1\%$, -1.6% and $+0.2\%$ for FQ-4bit. Averaged over datasets and horizons, compressing to 4 bits with SpecSpin therefore costs no accuracy on any of the three backbones.

Larger gains at longer horizons. The advantage over BF16 widens with the forecast horizon on all three backbones (Figure 2d): from $+1.8\%$, $+1.2\%$ and -0.8% at $H{=}96$ (Chronos, TimeMoE, Moirai) to -2.7%, -9.0% and -7.7% at $H{=}720$. Long horizons are where perturbations of the attention pattern compound the most, and they are exactly where SpecSpin helps the most.

Broad gains across datasets and architectures. The improvement is not driven by a single benchmark. On Moirai, SpecSpin is the best 4-bit method on $5/6$ datasets and improves on BF16 on all six, most on ETTm2 (-14.3%). On Chronos, it is the best 4-bit method on all four ETT datasets, 0.7% to 2.5% below BF16.



(a) Chronos (b) TimeMoE

(c) Moirai (d) SpecSpin across H

Figure 2: $\Delta\%$ **MSE vs. BF16 of the 4-bit methods.** (a)–(c) Per dataset, averaged over H (Table 1). (d) SpecSpin per horizon, averaged over the six datasets.

Weight-side and score-side transforms are complementary. Filtering the attention scores alone on top of uniform 4-bit weights (FQ-4bit) is unstable: it improves on AWQ-4bit on TimeMoE, but on Moirai ETTm1 it raises MSE by 16.6%. SpecSpin, which pairs a gentler score filter with truncated and spun NF4 weights, reaches -3.3% on the same cell and has a lower overall average than FQ-4bit on all three backbones (0.344 vs. 0.347, 0.366 vs. 0.381, 0.354 vs. 0.374). The score filter is therefore most useful as part of the full recipe, where the weight-side transform has already removed the low-energy directions it would otherwise amplify.

The largest gains appear on the most spectrally concentrated data. The biggest single improvements are on Exchange with TimeMoE (-37.8%) and on ETTm2 with Moirai (-14.3%). These are among the most spectrally concentrated benchmarks in the suite ($\rho^*(0.9){=}0.010$ and 0.039), where discarding low-energy directions and high-frequency score content removes the least signal, consistent with the spectral view that motivates SpecSpin.

Architecture sets the headroom. Chronos-2 is already robust to plain 4-bit rounding (AWQ-4bit -0.1%), so there is little error left to recover; SpecSpin still lowers the average by a further 0.8 points. TimeMoE and Moirai 2.0 decode autoregressively, so a perturbed prediction is fed back as input for

later steps, and there SpecSpin gains the most (-5.5% and -5.2%). Across all 18 backbone–dataset pairs, SpecSpin is more accurate than AWQ-4bit on 11 and ties on one.

3.3 EFFICIENCY

To assess the deployment cost of SpecSpin, we measure its static weight memory and dynamic compute against BF16 and the 4-bit baselines (Table 2). SpecSpin runs at the same cost as standard 4-bit quantization, about a quarter of BF16 in both storage and compute, so its accuracy gains come without extra deployment cost.

$4\times$ **smaller weights.** For Π parameters at bit-width b_W, weight memory is $\Pi \cdot b_W/8$ plus per-group scales, so every 4-bit method stores a quarter of the BF16 weights (100 vs. 400 MB, -75.0%). SpecSpin is slightly smaller still: its q/k/v matrices drop the truncated $1-\rho_W$ fraction of input columns, the rotation is regenerated from a fixed seed rather than stored, and the score filter touches only transient scores.

Table 2: **Efficiency:** static weight memory and dynamic compute (BOPs) on ETTh1 (7 variables) and Weather (21 variables). Settings: $B = 1$, $L = 512$, $b_A = 16$. Percentages are relative to BF16. All 4-bit methods share the 4-bit row; SpecSpin's q/k/v truncation removes a further $1-\rho_W$ of those projections' weights, and its input rotation roughly offsets the corresponding MAC saving.

Method	Static Efficiency		ETTh1 BOPs (T) Horizon				Weather BOPs (T) Horizon			
	Wt (MB)	$\Delta\%$	96	192	336	720	96	192	336	720
BF16	400	ref.	204.7	227.4	298.5	435.0	614.1	682.3	895.5	1304.9
4-bit (AWQ-4bit, FQ-4bit, SpecSpin)	**100**	**−75.0%**	**51.2**	**56.9**	**74.6**	**108.7**	**153.5**	**170.6**	**223.9**	**326.2**

75% **fewer bit-operations, with negligible filter overhead.** Measured in bit-operations, $\mathrm{BOPs} = \mathrm{MACs} \times b_W \times b_A$ (b_A=16), the 4-bit methods cut dynamic compute by 75% in every setting. The score filter runs after the $QK^\top$ matmul and adds under 1% ($\approx 0.7\%$ at L=512), and SpecSpin's input rotation roughly offsets the MACs saved by truncation, so its compute matches the other 4-bit methods. BOPs grow linearly with the number of channels ($3\times$ from ETTh1 to Weather) but sublinearly with the horizon ($2.1\times$ for a $7.5\times$ longer horizon), because prefill is amortized over the decoding steps.

Consistent memory savings across backbones. Measured on the released checkpoints, SpecSpin shrinks Chronos-2 from 227.9 to 65.5 MB, TimeMoE-200M from 864.4 to 222.4 MB and Moirai 2.0 from 21.7 to 5.5 MB, a 3.5–3.9$\times$ reduction that holds across a $40\times$ range of model sizes. The largest backbone, TimeMoE-200M, drops from almost a gigabyte to under a quarter of one, which brings it within the memory budget of edge gateways that could not hold the BF16 model.

3.4 ABLATION STUDY

To identify which stage of SpecSpin drives its accuracy, we remove one stage at a time on Chronos (ETTh1, ETTh2 and Exchange, all four horizons): the feature-axis transform (ρ_W=1, so the spin and NF4 act on the full feature dimension) or the spin (α=0). Table 3 reports the results: the feature-axis spectral transform drives the accuracy, while the spin can be kept at no accuracy cost.

Spectral truncation is the key component. Removing the feature-axis transform raises average MSE by 2.4%, 2.9% and 2.4% on ETTh1, ETTh2 and Exchange, and it is worse than full SpecSpin in all 12 dataset–horizon cells (Appendix Table 6). Discarding the lowest-energy DCT directions before quantization therefore improves accuracy consistently rather than only saving parameters.

The spin is accuracy-neutral. Removing it changes the average by at most 0.1% on all three datasets, so the rotation can be kept as a safeguard for the quantizer without an accuracy cost. Appendix D shows how each hyperparameter moves the error.

Table 3: **Ablation of SpecSpin on Chronos** (test MSE averaged over $H \in \{96, 192, 336, 720\}$, one stage removed at a time; $\Delta\%$ vs. full SpecSpin). **Bold**: lowest Avg per dataset. Per-horizon results: Appendix Table 6.

Variant	ETTh1		ETTh2		Exchange	
	Avg	$\Delta\%$	Avg	$\Delta\%$	Avg	$\Delta\%$
SpecSpin (full)	**0.4081**	—	**0.3460**	—	**0.3889**	—
w/o feature-axis transform (ρ_W=1)	0.4178	+2.4%	0.3559	+2.9%	0.3984	+2.4%
w/o spin (α=0)	0.4086	+0.1%	**0.3460**	0.0%	0.3891	+0.1%

4 Related Work

Time-series foundation models. Pretrained TSFMs such as Chronos (Ansari et al., 2024), TimeMoE (Shi et al., 2025), Moirai (Woo et al., 2024) and Timer (Liu et al., 2024b) forecast unseen series zero-shot, in contrast to supervised forecasters such as PatchTST (Nie et al., 2022) and iTransformer (Liu et al., 2024a) that are trained per dataset. Their size makes deployment on memory-constrained devices difficult, and SpecSpin compresses a frozen pretrained TSFM without retraining.

Quantization. Post-training quantization for LLMs reduces weights to 4 bits or fewer: GPTQ (Frantar et al., 2022) and AWQ (Lin et al., 2024) use calibration data to choose rounding or per-channel scales matches the quantization grid to bell-shaped weights. Rotation-based methods, QuIP (Chee et al., 2023), QuaRot (Ashkboos et al., 2024) and SpinQuant (Liu et al., 2025), apply orthogonal transforms that spread outliers before rounding. These methods were designed for language models and leave the attention scores untouched. SpecSpin borrows the rotation, but applies it inside a DCT subspace selected by weight energy, and adds a filter on the attention scores; it needs no calibration data.

Frequency-domain methods. Frequency-domain modeling is well established in time-series forecasting: FEDformer (Zhou et al., 2022b), FiLM (Zhou et al., 2022a) and FreTS (Yi et al., 2023) build Fourier or Legendre blocks into supervised models trained from scratch. SpecSpin instead applies frequency-domain transforms to the weights and pre-softmax scores of a frozen pretrained TSFM, as part of quantization rather than of model design.

5 Conclusion

We presented SpecSpin, a training-free, calibration-free 4-bit quantization recipe for time-series foundation models built on one structural observation: TSFM attention is spectrally sparse, along the key/time axis of its scores and along the feature axis of its q/k/v projections. SpecSpin truncates the projections to their highest-energy DCT directions, rotates and NF4-quantizes them, quantizes the rest of the model to NF4, and low-pass filters the attention scores. With retention settings tuned per backbone (and on some datasets per dataset), it is the best 4-bit method on overall average for Chronos, TimeMoE and Moirai, -0.9%, -5.5% and -5.2% below BF16 at a quarter of the weight memory, and its advantage grows to -9.0% at H=720.

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

All datasets and pretrained backbones are public. SpecSpin is training-free, needs no calibration data and uses a fixed rotation seed; implementation details and hyperparameters are in Appendix A, and code is provided in the supplementary material.

REFERENCES

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.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in neural information processing systems*, 36: 4396–4429, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6: 87–100, 2024.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pages 11116–11140, 2024a.

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*, 2024b.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, volume 2025, pages 92009–92032, 2025.

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, pages 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.

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.

Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35:12677–12690, 2022a.

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*, pages 27268–27286. PMLR, 2022b.

A Implementation Details

A.1 Notation

We summarise all symbols used throughout the paper for quick reference (Table 4).

Table 4: SpecSpin notation summary.

Symbol	Meaning
Tensors and dimensions	
$\mathbf{x}$	Input multivariate time series, $\mathbf{x} \in \mathbb{R}^{C \times L}$
L	Lookback (input context) length
H	Forecast horizon
C	Number of channels (variates)
P	Patch length
B	Batch size; h = number of attention heads; d_h = per-head hidden dim
L_q, L_k	Query / key sequence length (post-patching)
Q, K, V	Query / key / value projections, each $\in \mathbb{R}^{B \times h \times L_q \times d_h}$
S	Pre-softmax score matrix $S{=}QK^\top/\sqrt{d_h} \in \mathbb{R}^{B \times h \times L_q \times L_k}$
$\widetilde{S}$	Score-LPF-filtered score matrix, $\widetilde{S}{=}\mathbf{D}^{-1}\Phi_\rho^{\mathrm{LPF}}(\mathbf{D}\,S)$ (Eq. 5)
M	Causal mask added to S before softmax
A	Attention output $A{=}\,\mathrm{softmax}(S{+}M)V$
W_Q, W_K, W_V, W_O	Attention projection weight matrices (quantized by AWQ)
SpecSpin operators	
$\mathbf{D}$	Orthonormal Type-II DCT on the score (key/temporal) axis
$\mathbf{D}^{-1}$	Inverse DCT (Type-III), $\mathbf{D}^{-1}{=}\mathbf{D}^\top$
F	Orthonormal Type-II DCT on the weight input-feature axis (Eq. 4)
Φ_ρ^{LPF}	Score-LPF operator: keeps top-$\lceil \rho L_k \rceil$ index-prefix DCT coefficients (Eq. 5)
$\mathcal{S}_{\rho_W}(W)$	Energy-ranked retention set on the weight axis: top-$\lceil \rho_W d_{\text{in}} \rceil$ columns of WF by energy
W_{LPF}	Truncated q/k/v projection $(WF)[:, \mathcal{S}_{\rho_W}(W)]$ used by Weight-LPF[topk]
$\hat{\mathbf{s}}$	DCT-II coefficients of score row $\mathbf{s}$, $\hat{\mathbf{s}}{=}\mathbf{D}\mathbf{s}$
Scalars and statistics	
ρ	Score-LPF retention, $\rho \in (0, 1]$ (Moirai 0.40, Chronos 0.87, TimesFM 0.28)
ρ_W	Feature (q/k/v input) retention of SpecSpin, $\rho_W \in (0, 1]$
R	Haar-random orthogonal rotation in the kept $k{=}\lceil \rho_W d_{\text{in}} \rceil$-dim. subspace (fixed seed)
α	Fraction of kept directions that are rotated (ablation only; $\alpha{=}1$ in SpecSpin)
$\mathcal{Q}_{\text{NF4}}$	NormalFloat-4 quantizer, per-group absmax scale, group size 128
$\rho^*(\tau)$	Index-prefix spectral concentration threshold
τ	Energy-capture threshold for ρ^* (default $\tau{=}0.9$)
$\gamma(\cdot)$	Mean autocorrelation function of the dataset
b	Weight-quantization bit-width, $b \in \{16, 8, 4, 3, 2\}$
ϵ	Worst-case row LPF residual norm in Proposition 1
Method names	
BF16	Backbone in bfloat16, no compression
AWQ-$b$bit	Group-wise b-bit weight-only PTQ (no spectral operator)
FQ-bf16	Score-LPF over BF16 backbone
FQ-$b$bit	Score-LPF over AWQ-$b$bit backbone (e.g., FQ-4bit)
SpecSpin	DCT top-k truncation + Haar rotation + NF4 on q/k/v, NF4 elsewhere, + score-LPF
G1, G2	Spectral groups: smooth/random-walk, mixed-spectrum

A.2 Algorithm, Patching, DCT-II Implementation, and Compatibility

This subsection collects the SpecSpin pseudocode (Algorithm 1) together with the deployment-side details: the forward-hook patching procedure, the FFT-based DCT-II kernel used inside the hook, and the compatibility envelope with common inference frameworks.

Score-LPF is a strict add-on between the formation of S and the softmax: the DCT, low-pass mask and inverse DCT are inserted as a forward hook on each attention block, and the rest of the attention

Algorithm 1 SpecSpin (per attention block)

Require: projections W_Q, W_K, W_V; retentions ρ_W, ρ; seed σ

Load time

for $W \in \{W_Q, W_K, W_V\}$ **do**
  $\mathcal{S} \leftarrow$ indices of the $\lceil \rho_W d_{\text{in}} \rceil$ largest column norms of WF ▷ feature-axis transform
  $R \leftarrow \mathrm{Haar}(|\mathcal{S}|; \sigma)$ ▷ spin
  $\widetilde{W} \leftarrow \mathcal{Q}_{\text{NF4}}(WF_{\mathcal{S}}R)$ ▷ quantize
end for
quantize every other linear layer with $\mathcal{Q}_{\text{NF4}}$

Forward pass

$\widetilde{Q}, \widetilde{K}, \widetilde{V} \leftarrow (xF_{\mathcal{S}}R)\widetilde{W}^{\top}_{\{Q,K,V\}}; \quad S \leftarrow \widetilde{Q}\widetilde{K}^{\top}/\sqrt{d_h}$

$\widetilde{S} \leftarrow \mathbf{D}^{-1}\big(\Phi_{\rho}(\mathbf{D}S)\big)$ along the key axis ▷ temporal-axis transform

return $\mathrm{softmax}(\widetilde{S} + M)\,\widetilde{V}$

computation is unchanged from the host TSFM. The DCT and IDCT use the FFT-based Type-II orthonormal transform, vectorised across the batch, head, and query dimensions; the LPF is a scalar index that requires no per-call allocation.

Causal masking and leakage semantics. Regarding the causal mask M, we distinguish three attention regimes:

1. *Autoregressive decoding ($L_q = 1$):* In step-by-step forecasting generation, the active query attends exclusively to historical key states accumulated in the KV cache ($L_k \leq t$). There are no future positions in the cache, and M is identically zero across all stored keys. Consequently, the row transform is executed strictly over past keys and no future information leakage is mathematically possible.
2. *Encoder-only backbones (Chronos-2):* time-axis attention is bidirectional, so the transform is taken over the full key sequence and the mask is added after the filter.
3. *Causal prefill ($L_q > 1$):* When processing an initial context sequence where key positions $j > i$ are masked ($M_{i,j} = -\infty$), a strictly causal transform applies the DCT over the valid prefix slice $\mathbf{s}_{i,1:i}$. In our LTSF benchmark evaluations, generation proceeds token-by-token in decoding mode or via non-autoregressive encoder blocks, ensuring that all reported metrics are strictly leak-free.

Patching procedure. Score-LPF patches any `nn.Module` whose forward function computes scaled dot-product attention. For Chronos, we replace `MHA._eager_attention` with a closure that intercepts the pre-softmax score matrix:

1. Compute $S = QK^{\top}/\sqrt{d_h}$ as in the original forward.
2. Call `dct_1d(S, dim=-1)` (FFT-based, defined in the next paragraph).
3. Zero high-frequency coefficients: `S_dct[..., k:] = 0` where $k = \lceil \rho \cdot L_k \rceil$.
4. Call `idct_1d(S_dct, dim=-1)` to recover filtered $\widetilde{S}$.
5. Continue with `softmax(`$\widetilde{S}$ `+ M) @ V`.

The patch touches 4 lines of the $\approx$50-line forward function and leaves Q, K, V, and the causal mask M unmodified.

FFT-based DCT-II implementation. The standard DCT-II of a sequence $\mathbf{s} \in \mathbb{R}^N$ can be computed in $\mathcal{O}(N \log N)$ via FFT by the following reordering trick: (i) construct the even-extended sequence $\mathbf{v} = [s_0, s_1, \ldots, s_{N-1}, s_{N-1}, \ldots, s_0]$ of length $2N$; (ii) compute $\hat{\mathbf{v}} = \mathrm{FFT}(\mathbf{v})$; (iii) the DCT-II coefficients are the real part of $e^{-j\pi k/(2N)}\hat{v}_k$ for $k = 0, \ldots, N-1$. The inverse DCT-III (IDCT) is implemented symmetrically using the conjugate phase correction and the IFFT. Round-trip error in float32 is $< 10^{-6}$ for $N = 512$.

HuggingFace Transformers. Backbones built on the standard `transformers` library expose attention as a discoverable submodule; patching walks the module tree with `model.named_modules()` and replaces the bound attention method of every matching instance (e.g. every `MHA` instance for Chronos) in place, so no changes to the surrounding model code are required.

vLLM and continuous batching. Score-LPF modifies only the score matrix computation, which occurs *after* the KV-cache read and *before* the attention output write. Neither the KV manager, the block allocator, nor the output gather is touched. vLLM compatibility requires only that the replaced forward function accepts the same signature as the original.

FlashAttention. FlashAttention fuses the QK matmul, softmax, and AV matmul into a single CUDA kernel, preventing interception of the score matrix. Score-LPF requires the unfused attention path (both AWQ and FQ baselines use unfused attention in our benchmarks, so the $+1.8\%$–$+2.3\%$ wall-clock comparison is internally consistent; a FlashAttention-accelerated AWQ baseline would widen this gap by 1.5–2$\times$). A custom fused DCT-filter-IDCT kernel would eliminate this; left to future work.

bfloat16 precision. The FFT-based DCT operates in float32 internally (via upcast) and downcasts the result. The round-trip precision in bfloat16 is $< 10^{-3}$, well below the magnitude of the score matrix entries and introduces no perceptible accuracy loss.

B Experimental Settings

This appendix collects the implementation choices that determine the results in Section 3: pretrained backbones (B.1), the post-training quantization recipe and other hyperparameters (B.2), and the datasets used for zero-shot evaluation (B.3).

B.1 Backbone Details

We evaluate SpecSpin on three pretrained TSFM backbones in the main table (Chronos, TimeMoE, Moirai) and report TimesFM separately as a limitation; together they span four distinct attention and decoding schemes. All backbones are loaded from their official public checkpoints; we apply no fine-tuning or task-specific adaptation.

Chronos-2 (?). A 120M-parameter encoder-only model inspired by the T5 encoder (`amazon/chronos-2`). It combines RoPE time-axis self-attention with attention across related series (group attention) and produces multi-step quantile forecasts; we use the mean forecast. SpecSpin transforms the q/k/v projections of all 24 attention layers and applies the score filter only to the time-axis (RoPE) attention layers.

TimeMoE (?). A decoder-only mixture-of-experts model (`Maple728/TimeMoE-200M`; 453M parameters in total, about 200M activated per token) with RoPE self-attention and continuous autoregressive outputs. SpecSpin truncates, rotates and NF4-quantizes the 36 q/k/v projections of its 12 attention layers and quantizes the remaining 364 linear layers (output projections and experts) to NF4; the expert router stays in BF16.

Moirai 2.0 (?). The small decoder-only successor of Moirai (Woo et al., 2024) (`Salesforce/moirai-2.0-R-small`, 11.4M parameters), with quantile outputs and recursive multi-step decoding; we use the mean forecast. Score matrices are intercepted by wrapping `F.scaled_dot_product_attention`, and each channel is forecast independently.

TimesFM (?). A 200M-parameter PaLM-style causal decoder with 20 layers and a fixed L_k=16 patch grid. The small L_k is significant for score-LPF: at ρ=0.28 only $\lceil 0.28 \cdot 16 \rceil = 5$ DCT coefficients are retained per row, which over long autoregressive chains accumulates unbounded error and makes the score filter unstable without a retention floor.

C Full Experimental Results

C.1 Full Comparison Across Bit-Widths

Table 5 extends the main comparison to more bit-widths and to TimesFM: it reports per-horizon zero-shot MSE for uniform quantization (AWQ-$\{16, 4, 3, 2\}$bit), the score filter on top of it (FQ-$\{16, 4, 3\}$bit) and SpecSpin on Chronos, Moirai and TimesFM. Across this wider grid, SpecSpin remains the most accurate compressed variant on Chronos and Moirai, and the spectral operators consistently recover part of the error that low-bit uniform quantization introduces. Figure 3 summarizes the table by bit-width and by horizon.

SpecSpin is the most accurate compressed variant on Chronos and Moirai. Averaged over datasets and horizons, SpecSpin is 0.8% below BF16 on Chronos, tied with FQ-3bit for the lowest error of any compressed row, and 5.2% below BF16 on Moirai, where it improves on BF16 on all six datasets; no other quantized row matches this on Moirai (AWQ-4bit reaches -3.6%).

The score filter rescues low-bit weights on Chronos. Uniform 3-bit quantization raises the Chronos error by 15.6%, driven by ETTh1, whereas adding the score filter to the same 3-bit weights (FQ-3bit) brings the average to 0.8% below BF16. Filtering the attention scores thus absorbs most of the rounding noise that 3-bit weights inject.

Spectral operators also narrow the gap on TimesFM. TimesFM is the most fragile backbone under uniform quantization ($+10.7\%$ at 4 bits, $+63.8\%$ at 3 bits). Both spectral variants reduce the 4-bit loss, to $+6.8\%$ for FQ-4bit and $+8.1\%$ for SpecSpin, even though TimesFM's short patch grid leaves the score filter little room to act.

Backbones differ widely in how much weight noise they absorb. Chronos-2 stays at BF16 accuracy even with 2-bit weights (-0.3%), whereas 2-bit quantization diverges on the autoregressive Moirai 2.0 (MSE above 1000 on ETTh1 at H=720) and TimesFM (marked $\times$). This matches the larger gains that SpecSpin brings to the autoregressive backbones in Table 1.

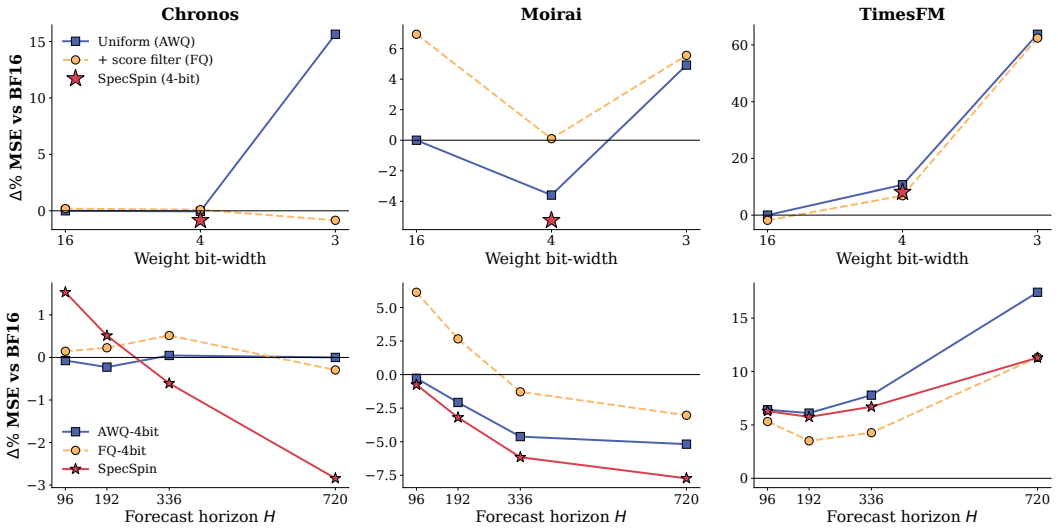


Figure 3: **Visual summary of Table 5** ($\Delta\%$ MSE vs. BF16). *Top:* error against weight bit-width for uniform quantization and with the score filter; the star marks SpecSpin at 4 bits (2-bit omitted, as it diverges on Moirai and TimesFM). *Bottom:* the 4-bit methods across forecast horizons, averaged over datasets.

D Key Parameter Study

This appendix studies the hyperparameters of SpecSpin: the feature retention ρ_W, the score retention ρ and the rotated fraction α (Section D.2), and the feature-axis operator (Section D.3).

Table 5: Zero-shot MSE per horizon for Chronos, Moirai and TimesFM under uniform quantization (AWQ-$\{16, 4, 3, 2\}$bit), the score filter on top of it (FQ-$\{16, 4, 3\}$bit) and SpecSpin, on six datasets. Avg: mean over horizons (columns) or datasets (rows). $\times$ marks TimesFM AWQ-2bit, which diverges by 6–13 orders of magnitude on every cell (round-to-nearest quantizer breakdown at 2 bits).

		Chronos					Moirai					TimesFM				
Method	Dataset	96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg
AWQ-16 (BF16)	ETTh1	0.371	0.414	0.439	0.449	0.418	0.378	0.420	0.454	0.501	0.438	0.421	0.472	0.510	0.514	0.479
	ETTh2	0.280	0.351	0.378	0.398	0.352	0.297	0.380	0.411	0.445	0.383	0.327	0.400	0.434	0.452	0.403
	ETTm1	0.308	0.362	0.395	0.484	0.387	0.338	0.403	0.459	0.547	0.437	0.357	0.411	0.441	0.507	0.429
	ETTm2	0.171	0.241	0.306	0.404	0.280	0.180	0.270	0.385	0.537	0.343	0.205	0.294	0.367	0.473	0.335
	Exchange	0.085	0.180	0.334	0.944	0.386	0.090	0.188	0.345	0.885	0.377	0.110	0.219	0.393	1.009	0.433
	Weather	0.156	0.211	0.281	0.383	0.258	0.168	0.216	0.283	0.383	0.263	0.122	0.169	0.243	0.392	0.232
	Avg	*0.229*	*0.293*	*0.355*	*0.510*	*0.347*	*0.242*	*0.313*	*0.390*	*0.550*	*0.373*	*0.257*	*0.328*	*0.398*	*0.558*	*0.385*
AWQ-4bit	ETTh1	0.370	0.412	0.438	0.448	0.417	0.373	0.411	0.436	0.448	0.417	0.445	0.500	0.537	0.540	0.506
	ETTh2	0.280	0.351	0.377	0.396	0.351	0.294	0.372	0.406	0.426	0.374	0.337	0.412	0.457	0.535	0.435
	ETTm1	0.307	0.361	0.396	0.486	0.388	0.338	0.396	0.444	0.526	0.426	0.394	0.452	0.496	0.581	0.481
	ETTm2	0.171	0.241	0.307	0.406	0.281	0.180	0.259	0.341	0.470	0.312	0.206	0.289	0.373	0.569	0.359
	Exchange	0.085	0.178	0.332	0.935	0.383	0.091	0.181	0.320	0.886	0.370	0.102	0.216	0.403	1.166	0.472
	Weather	0.157	0.212	0.284	0.391	0.261	0.171	0.219	0.282	0.371	0.261	0.157	0.216	0.308	0.539	0.305
	Avg	*0.228*	*0.292*	*0.356*	*0.510*	*0.347*	*0.241*	*0.306*	*0.371*	*0.521*	*0.360*	*0.274*	*0.347*	*0.429*	*0.655*	*0.426*
AWQ-3bit	ETTh1	0.744	0.743	0.720	0.693	0.725	0.406	0.448	0.489	0.715	0.514	0.601	0.720	0.874	1.422	0.904
	ETTh2	0.368	0.381	0.379	0.417	0.386	0.298	0.376	0.410	0.432	0.379	0.424	0.546	0.664	0.895	0.632
	ETTm1	0.306	0.359	0.392	0.478	0.384	0.354	0.413	0.465	0.577	0.452	0.623	0.742	0.868	1.202	0.859
	ETTm2	0.170	0.239	0.303	0.402	0.278	0.180	0.255	0.322	0.418	0.294	0.275	0.384	0.525	0.919	0.526
	Exchange	0.086	0.179	0.331	0.906	0.376	0.088	0.182	0.352	1.175	0.449	0.103	0.205	0.345	0.799	0.363
	Weather	0.155	0.210	0.282	0.384	0.258	0.170	0.218	0.284	0.377	0.262	0.243	0.330	0.490	0.938	0.500
	Avg	*0.305*	*0.352*	*0.401*	*0.547*	*0.401*	*0.249*	*0.315*	*0.387*	*0.616*	*0.392*	*0.378*	*0.488*	*0.628*	*1.029*	*0.631*
AWQ-2bit	ETTh1	0.373	0.413	0.430	0.426	0.410	4.511	11.3	33.1	1039	272	$\times$	$\times$	$\times$	$\times$	$\times$
	ETTh2	0.268	0.333	0.361	0.381	0.336	1.051	1.804	4.013	107	28.5	$\times$	$\times$	$\times$	$\times$	$\times$
	ETTm1	0.289	0.334	0.373	0.450	0.361	1.284	2.458	7.465	341	88.2	$\times$	$\times$	$\times$	$\times$	$\times$
	ETTm2	0.169	0.231	0.288	0.380	0.267	0.407	0.673	1.402	35.1	9.398	$\times$	$\times$	$\times$	$\times$	$\times$
	Exchange	0.112	0.265	0.472	1.049	0.474	0.177	0.342	0.650	12.1	3.317	$\times$	$\times$	$\times$	$\times$	$\times$
	Weather	0.143	0.191	0.247	0.322	0.226	0.291	0.416	0.690	10.6	2.992	$\times$	$\times$	$\times$	$\times$	$\times$
	Avg	*0.226*	*0.295*	*0.362*	*0.501*	*0.346*	*1.287*	*2.840*	*7.880*	*258*	*67.4*	$\times$	$\times$	$\times$	$\times$	$\times$
FQ-16 (FQ-bf16)	ETTh1	0.368	0.414	0.443	0.450	0.419	0.373	0.415	0.449	0.491	0.432	0.418	0.463	0.498	0.504	0.471
	ETTh2	0.282	0.355	0.383	0.402	0.356	0.297	0.379	0.410	0.448	0.383	0.332	0.406	0.437	0.447	0.406
	ETTm1	0.312	0.365	0.400	0.487	0.391	0.449	0.536	0.633	0.748	0.592	0.351	0.401	0.431	0.493	0.419
	ETTm2	0.172	0.242	0.306	0.400	0.280	0.180	0.270	0.386	0.541	0.344	0.198	0.277	0.343	0.442	0.315
	Exchange	0.085	0.177	0.328	0.922	0.378	0.090	0.187	0.345	0.887	0.377	0.110	0.217	0.390	1.028	0.436
	Weather	0.156	0.212	0.286	0.394	0.262	0.170	0.218	0.288	0.395	0.268	0.128	0.171	0.234	0.354	0.222
	Avg	*0.229*	*0.294*	*0.358*	*0.509*	*0.348*	*0.260*	*0.334*	*0.418*	*0.585*	*0.399*	*0.256*	*0.323*	*0.389*	*0.545*	*0.378*
FQ-4bit	ETTh1	0.367	0.412	0.441	0.449	0.417	0.369	0.408	0.432	0.441	0.412	0.444	0.492	0.524	0.524	0.496
	ETTh2	0.282	0.355	0.382	0.399	0.355	0.293	0.371	0.406	0.427	0.374	0.336	0.406	0.448	0.511	0.425
	ETTm1	0.311	0.364	0.401	0.488	0.391	0.437	0.489	0.524	0.584	0.508	0.385	0.437	0.474	0.534	0.458
	ETTm2	0.172	0.242	0.306	0.401	0.280	0.179	0.259	0.340	0.466	0.311	0.202	0.275	0.346	0.492	0.329
	Exchange	0.084	0.176	0.325	0.914	0.375	0.091	0.181	0.320	0.892	0.371	0.101	0.211	0.399	1.149	0.465
	Weather	0.157	0.214	0.289	0.402	0.265	0.171	0.219	0.285	0.388	0.266	0.156	0.213	0.299	0.516	0.296
	Avg	*0.229*	*0.294*	*0.357*	*0.509*	*0.347*	*0.257*	*0.321*	*0.385*	*0.533*	*0.374*	*0.271*	*0.339*	*0.415*	*0.621*	*0.411*
FQ-3bit	ETTh1	0.367	0.412	0.442	0.443	0.416	0.399	0.441	0.479	0.654	0.493	0.598	0.706	0.858	1.399	0.890
	ETTh2	0.281	0.353	0.381	0.394	0.352	0.298	0.375	0.410	0.431	0.379	0.435	0.560	0.687	0.935	0.654
	ETTm1	0.310	0.363	0.398	0.482	0.388	0.415	0.458	0.493	0.562	0.482	0.601	0.703	0.819	1.134	0.814
	ETTm2	0.171	0.241	0.303	0.397	0.278	0.180	0.255	0.321	0.415	0.293	0.274	0.377	0.522	0.910	0.521
	Exchange	0.085	0.177	0.326	0.891	0.370	0.088	0.182	0.353	1.175	0.450	0.104	0.207	0.354	0.838	0.376
	Weather	0.154	0.210	0.283	0.391	0.260	0.172	0.223	0.292	0.390	0.269	0.239	0.322	0.491	0.940	0.498
	Avg	*0.228*	*0.293*	*0.355*	*0.500*	*0.344*	*0.259*	*0.322*	*0.391*	*0.605*	*0.394*	*0.375*	*0.479*	*0.622*	*1.026*	*0.626*
SpecSpin	ETTh1	0.362	0.405	0.431	0.433	0.408	0.372	0.410	0.430	0.430	0.410	0.428	0.481	0.523	0.571	0.501
	ETTh2	0.278	0.344	0.371	0.389	0.346	0.295	0.369	0.400	0.435	0.375	0.337	0.414	0.455	0.476	0.420
	ETTm1	0.307	0.359	0.395	0.477	0.384	0.341	0.393	0.439	0.514	0.422	0.394	0.458	0.504	0.596	0.488
	ETTm2	0.170	0.238	0.302	0.395	0.276	0.178	0.249	0.320	0.430	0.294	0.209	0.288	0.365	0.488	0.338
	Exchange	0.100	0.195	0.336	0.919	0.388	0.086	0.182	0.332	0.876	0.369	0.099	0.209	0.398	1.166	0.468
	Weather	0.175	0.227	0.285	0.362	0.262	0.168	0.214	0.272	0.358	0.253	0.172	0.228	0.303	0.428	0.283
	Avg	*0.232*	*0.295*	*0.353*	*0.496*	*0.344*	*0.240*	*0.303*	*0.366*	*0.507*	*0.354*	*0.273*	*0.346*	*0.425*	*0.621*	*0.416*

D.1 Ablation: Per-Horizon Results

Table 6 breaks the ablation of Table 3 down by horizon. It shows where each stage of SpecSpin contributes.

The feature-axis transform helps everywhere, and most where it matters. Removing it raises MSE in all 12 dataset–horizon cells. On ETTh1 and ETTh2 its benefit grows with the horizon, from 1.5% and 0.8% at $H{=}96$ to 3.8% and 4.9% at $H{=}720$, so spectral truncation is most valuable exactly where forecasting error compounds. On Exchange, the most spectrally concentrated dataset, it gives the largest short-horizon gains (7.3% at $H{=}96$ and 9.0% at $H{=}192$).

The score filter pays off at the ends of the horizon range. At $H{=}720$, removing the temporal-axis transform raises MSE on ETTh1 ($+0.5\%$) and ETTh2 ($+0.4\%$), and on Exchange it helps at the short horizons ($+1.6\%$ at $H{=}96$ and $+0.8\%$ at $H{=}192$ without it). At the intermediate horizons of ETTh1 and ETTh2 its effect is within $\pm 0.8\%$.

The spin is a safe addition. Removing the spin changes MSE by less than 1.5% in every cell. On ETTh1 it slightly lowers the error at every horizon, so it can be kept as part of the recipe without trading off accuracy.

Table 6: **Ablation of SpecSpin on Chronos, per horizon** (test MSE; one stage removed at a time, all other settings as in Table 1). **Bold**: lowest MSE per row.

Dataset	Horizon	SpecSpin	w/o feature-axis	w/o temporal-axis	w/o spin
ETTh1	$H{=}96$	0.3627	0.3682	**0.3616**	0.3631
	$H{=}192$	0.4063	0.4139	**0.4049**	0.4068
	$H{=}336$	0.4326	0.4420	**0.4315**	0.4329
	$H{=}720$	**0.4306**	0.4470	0.4329	0.4316
	Avg	0.4081	0.4178	**0.4077**	0.4086
ETTh2	$H{=}96$	0.2782	0.2805	**0.2776**	0.2785
	$H{=}192$	0.3451	0.3540	**0.3435**	0.3452
	$H{=}336$	0.3737	0.3828	**0.3706**	0.3736
	$H{=}720$	0.3870	0.4061	0.3887	**0.3866**
	Avg	0.3460	0.3559	**0.3451**	0.3460
Exchange	$H{=}96$	0.0988	0.1060	0.1004	**0.0975**
	$H{=}192$	0.1932	0.2106	0.1947	**0.1917**
	$H{=}336$	0.3368	0.3422	0.3364	**0.3354**
	$H{=}720$	0.9268	0.9348	**0.9188**	0.9317
	Avg	0.3889	0.3984	**0.3875**	0.3891

D.2 Key Parameter Study

We vary one hyperparameter at a time around the Chronos default $(\rho_W, \rho, \alpha){=}(0.90, 0.87, 1)$ on ETTh1 at all four horizons (Figure 4, Table 7). **Feature retention** ρ_W has a clear optimum at 0.90, the best value at every horizon: keeping every direction ($\rho_W{=}1$) raises average MSE from 0.4081 to 0.4178, and truncating further degrades quickly (0.4100 at 0.8, 0.4259 at 0.7, 0.5604 at 0.5). **Score retention** ρ must stay high: the error is flat for $\rho \geq 0.87$ (0.4081 vs. 0.4077 at $\rho{=}1$) and rises sharply below 0.7 (0.5710 at 0.5, 0.7296 at 0.3). **Rotated fraction** α barely matters: average MSE stays between 0.4063 and 0.4086 across $\alpha \in [0, 1]$.

D.3 Energy-Ranked Feature Truncation

SpecSpin's feature-axis step (Eq. 4) keeps the energy-ranked DCT directions of the input feature dimension of the q/k/v projections, distinct from the score-LPF on the temporal axis; the Haar rotation and NF4 quantization are then applied inside the kept subspace.

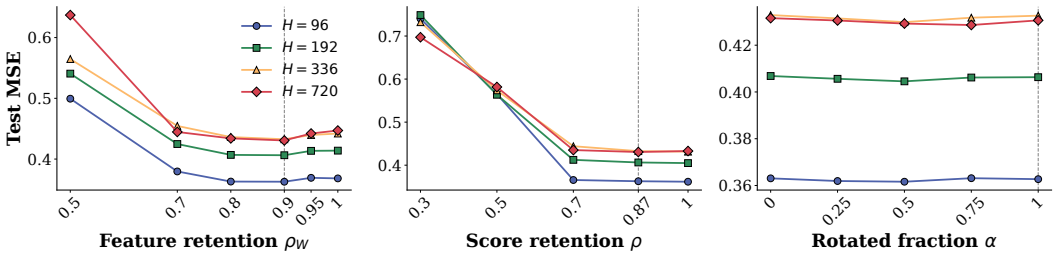


Figure 4: **Key parameter study on Chronos, ETTh1.** Test MSE as one of ρ_W, ρ, α is varied with the others at the default (dashed line), one line per horizon.

Table 7: Key parameter study on Chronos, ETTh1 (test MSE). One parameter varies at a time; the others stay at $(\rho_W, \rho, \alpha)=(0.90, 0.87, 1)$.

	Value	H=96	H=192	H=336	H=720	Avg
ρ_W	0.5	0.4994	0.5406	0.5647	0.6370	0.5604
	0.7	0.3796	0.4248	0.4545	0.4446	0.4259
	0.8	0.3630	0.4069	0.4362	0.4340	0.4100
	0.9 (default)	0.3627	0.4063	0.4326	0.4306	0.4081
	0.95	0.3691	0.4135	0.4397	0.4421	0.4161
	1	0.3682	0.4139	0.4420	0.4470	0.4178
ρ	0.3	0.7404	0.7487	0.7321	0.6972	0.7296
	0.5	0.5650	0.5636	0.5739	0.5816	0.5710
	0.7	0.3656	0.4124	0.4441	0.4351	0.4143
	0.87 (default)	0.3627	0.4063	0.4326	0.4306	0.4081
	1	0.3616	0.4049	0.4315	0.4329	0.4077
α	0	0.3631	0.4068	0.4329	0.4316	0.4086
	0.25	0.3619	0.4056	0.4315	0.4306	0.4074
	0.5	0.3616	0.4045	0.4299	0.4292	0.4063
	0.75	0.3631	0.4062	0.4318	0.4286	0.4074
	1 (default)	0.3627	0.4063	0.4326	0.4306	0.4081

Operator. For an attention projection $W \in \mathbb{R}^{\text{out}\times\text{in}}$ and a fixed retention $\rho_W \in (0, 1]$, define the energy-ranked truncation

$$W_{\text{LPF}} \;=\; W F\, M_{\text{topk}}, \qquad M_{\text{topk}} = \mathbf{1}\big[\, j \in \text{top-}\lceil \rho_W \cdot \text{in} \rceil \,(\, \|WF\|_{:,j}\,)\,\big],$$

where F is the orthonormal Type-II DCT matrix on the input axis and M_{topk} keeps the $\lceil \rho_W \cdot \text{in} \rceil$ DCT bins with largest column-energy $\|WF\|_{:,j} = \big(\sum_o (WF)^2_{o,j}\big)^{1/2}$. At inference,

$$y = (xF)_{\text{kept-bins}}\, W_{\text{LPF}}^{\top} + b$$

matches the original $y = xW^{\top} + b$ in the kept subspace. Because both bin-selection and basis change are deterministic functions of W, M_{topk} is computed once at load time; no calibration data or activation profiling is required.

Why *topk* (not lowpass). Pretrained q/k/v weights have approximately flat per-bin energy under DCT-II along the input dimension, but show non-uniform structure under measurement: energy-ranked selection keeps more column energy at matched retention than index-prefix selection, mirroring the score-side observation that informative DCT mass is concentrated but not always at low frequencies (Section 2.2). Energy-ranked selection therefore strictly dominates index-prefix selection for weight matrices.

Choice of ρ_W. SpecSpin uses $\rho_W \in \{0.90, 0.95\}$ (Section 3.1); the parameter study (Figure 4) shows a clear optimum at ρ_W=0.90 on Chronos, with accuracy degrading on both sides, and lowpass (index-prefix) collapses earlier than topk at every retention level. This is consistent with the informative bins on the input-feature axis being scattered rather than concentrated at low indices, which is why we use the energy-ranked operator.

FLOPs accounting (analytical). At $\rho_W{=}0.95$ each of the 3 q/k/v projections drops 5% of its input-axis weight count, so the attention-projection parameter footprint shrinks by $5\% \cdot \frac{3}{4} = 3.75\%$, or roughly 5% of the q/k/v parameters. The W-LPF wrapper also performs an input rotation $xF[:, \mathcal{S}] \in \mathbb{R}^{BL_q \times \lceil \rho_W d_{\text{in}} \rceil}$ each forward, which is not counted by a naive Linear-module MAC hook; the shared per-layer rotation matrix $F[:, \mathcal{S}] \in \mathbb{R}^{d_{\text{in}} \times \lceil \rho_W d_{\text{in}} \rceil}$ is reusable across q/k/v with the same input dimension (a single buffer suffices), so the rotation cost is paid once per attention block. Net: W-LPF[topk]'s practical compute benefit is bounded by its 5% q/k/v parameter-storage saving rather than by a simple linear-MAC reduction; it is overhead, not save, until a fused kernel folds the rotation into the matmul (future work).

Composability with score-LPF. The two LPFs act on independent subspaces (weight-input-feature versus score-key-time), so their reconstruction errors do not amplify each other (orthogonality argument analogous to Proposition 1).