VOLATILITY-CLUSTERING ADAPTATION FOR FINANCIAL TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series foundation models are increasingly adapted to new domains through fine-tuning on target data, under the implicit assumption that more target data yields better forecasts. We show that this assumption can fail in financial forecasting, where individual price changes are difficult to predict, but large moves tend to cluster, creating alternating calm and turbulent periods. Using financial foundation models trained on price bars of open, high, low, close, and volume, we argue that adapting to financial domains requires training signals beyond next-token prediction. We introduce **Volatility-Clustering Adaptation (VCA)**, which augments next-token cross-entropy with a differentiable penalty on the autocorrelation of squared returns, the standard statistical signature of volatility clustering. This additional objective provides a multi-step training signal by matching the resulting dependence structure of autoregressive rollouts to those of the realized future. Across three asset sets and two evaluation conventions, VCA improves adaptation over the pre-trained model, with the strongest gains under the primary evaluation (FORE), driven primarily by reduced variance error. Overall, our results suggest that effective financial adaptation requires objectives that capture domain-specific temporal structure beyond token-level prediction.

1 INTRODUCTION

Time-series foundation models are pre-trained on large and diverse corpora and then applied to previously unseen time series (Rasul et al., 2023; Das et al., 2024; Woo et al., 2024; Goswami et al., 2024; Ansari et al., 2024). Financial foundation models such as Kronos (Shi et al., 2026) extend this paradigm to finance by tokenizing open, high, low, close, and volume (OHLCV) bars and autoregressively generating future price paths. Adapting such models to a target market typically involves fine-tuning on historical data, with the implicit assumption that using more target data should improve performance. That assumption can fail because fine-tuning can distort useful pre-trained features and reduce out-of-distribution robustness (Kumar et al., 2022), or overwrite capabilities acquired during pre-training (Kirkpatrick et al., 2017). The problem is particularly relevant in finance, where returns, defined as relative changes in asset prices, are weakly predictable but volatility is strongly autocorrelated and varies across regimes (Engle, 1982; Bollerslev, 1986; Cont, 2001). Long historical windows can therefore mix very different volatility regimes, so adding more data may introduce patterns that are less relevant to the regime at deployment.

This raises two separate questions for adaptation: *what structure the training objective should enforce*, and, secondarily, *which data to fine-tune on*. For the training objective, we encode volatility clustering, a classical stylized fact of financial returns, as an auxiliary signal, since a next-token prediction loss gives no explicit signal about the shape of a multi-step rollout used for test-time evaluation. In particular, we introduce **Volatility-Clustering Adaptation (VCA)**, which matches the *autocorrelation function of squared returns (ACF^2)* between autoregressive rollouts and the corresponding realized future paths (Figure 1). This directly trains the model to reproduce volatility clustering over the evaluation horizon. To make this objective differentiable, we use a Gumbel-softmax straight-through rollout (Jang et al., 2017; Bengio et al., 2013). Additionally, we compute ACF statistics at the batch level rather than from individual trajectories, providing a more stable training signal. This formulation brings stylized-fact-driven objectives, analogous to physics-informed learning (Raissi et al., 2019; Wiese et al., 2020), into the adaptation of an existing financial foundation model rather than training one from scratch. Regarding data selection, we ask a retrospective

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

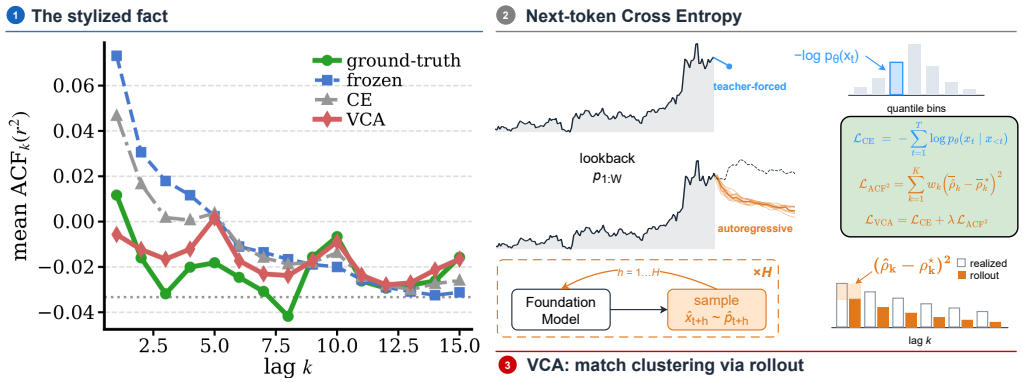


Figure 1: Overview of Volatility-Clustering Adaptation (VCA). ❶ Mean ACF of squared returns against lag k on CSI300: the realized (ground-truth) path stays positive and well above the dotted zero-clustering null, the signature of volatility clustering. The frozen model and CE fine-tuning deviate from the ground-truth curve, while VCA matches its shape more closely. ❷ The baseline objective $\mathcal{L}_{\text{CE}}$ is single-step and teacher-forced: it scores the next token against the ground truth but gives no signal about the shape of the multi-step path the model generates at test time. ❸ VCA rolls out autoregressively over the horizon and adds $\mathcal{L}_{\text{ACF}^2}$, which matches the *batch-mean* ACF of squared returns $\bar{\rho}_k$ to the realized $\bar{\rho}_k^\star$. VCA's total objective is $\mathcal{L}_{\text{VCA}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{ACF}^2}$.

question: does fine-tuning on a chronological slice whose realized volatility matches the test period outperform fine-tuning on the full historical set? We hypothesize that long historical windows mix volatility regimes that may differ from the one at deployment, so a smaller matched slice could supply more relevant signal than the full set. This setup examines whether regime alignment drives adaptation gains beyond data volume. To verify, we use the realized volatility of the test period for matching, making this a diagnostic study of volatility-regime alignment.

Across major financial benchmarks such as CSI300, CSI500, and Crypto, VCA improves adaptation under both evaluation conventions, with gains driven primarily by lower variance error. In contrast, cross-entropy fine-tuning can be harmful, particularly on Crypto, while full-data VCA causes greater weight drift from the frozen backbone, which we use as a proxy for potential forgetting. Regime-matched historical data improves data efficiency at longer forecasting horizons, matching or outperforming full-data fine-tuning with less weight drift, but this benefit does not extend to shorter horizons. We observe similar results when adapting Chronos-T5-small, suggesting that these challenges are not specific to Kronos, highlighting the value of domain-specific path-level objectives and horizon-aware data selection for financial foundation-model adaptation.

Our contributions are:

- We introduce **Volatility-Clustering Adaptation (VCA)**, a differentiable objective that matches volatility clustering in autoregressive rollouts, with a controlled ablation isolating the contribution of ACF^2 from that of rollout-based training itself.
- We show that volatility-matched data can match or outperform full-data fine-tuning at the longer horizon with substantially less weight drift, while broader data coverage remains more important at the shorter horizon.
- We identify a volatility-trend direction that the ACF^2 loss cannot constrain and confirm it in both simulation and empirical results, providing a partial explanation for VCA's instability.

2 Related Work

Time-series foundation models. Lag-Llama (Rasul et al., 2023), TimesFM (Das et al., 2024), Moirai (Woo et al., 2024), MOMENT (Goswami et al., 2024), and Chronos (Ansari et al., 2024) are pre-trained on large heterogeneous corpora and applied zero-shot across domains, using the same weights without modeling domain-specific structure. Kronos (Shi et al., 2026) instead targets

finance, tokenizing OHLCV bars into hierarchical discrete tokens and pre-training a decoder-only transformer on billions of K-lines. All of these works focus on optimizing forecasting accuracy, but none constrain adaptation to preserve domain-specific properties such as volatility clustering.

Adaptation, domain knowledge, and data selection for time series. Several approaches have been proposed to adapt time-series foundation models to target domains. Parameter-efficient methods use LoRA (Hu et al., 2022; Gupta et al., 2024) or specialized adaptation schemes such as TRACE (Li & Zhu, 2025) to update only a small fraction of model parameters. In-context fine-tuning adapts to a target distribution through related time-series examples without updating model weights at inference time (Faw et al., 2025), while online adaptation updates forecasts as new observations arrive (Lee et al., 2025). Other work moves beyond conventional forecast loss, for example by fine-tuning toward downstream decision objectives (Beichter et al., 2025). In contrast, our adaptation changes the training objective itself by encoding a financial stylized fact, volatility clustering, through a differentiable path-level loss. Data selection provides a complementary route, with evidence that carefully selected subsets can outperform full-data fine-tuning, while simply matching the training data to the target distribution does not always improve transfer (Xia et al., 2024; Kang et al., 2024). In finance, DoubleAdapt (Zhao et al., 2023) adapts a stock-forecasting model to shifting regimes through a learned adapter. We instead use volatility regime to select historical data and find that such alignment can improve data efficiency at longer forecasting horizons.

Stylized facts in financial modeling. Autocorrelation in squared returns, together with weak autocorrelation in raw returns, is a classical financial stylized fact (Engle, 1982; Bollerslev, 1986; Cont, 2001). Their strength depends on sampling frequency (Drost & Nijman, 1993), motivating our evaluation across forecast horizons. Quant GANs reproduce financial stylized facts directly (Wiese et al., 2020), while physics-informed learning encodes known structure in differentiable objectives (Raissi et al., 2019). Exposure bias highlights the gap between teacher-forced training and autoregressive evaluation (Bengio et al., 2015; Lamb et al., 2016), and soft-DTW and DILATE address related gaps with differentiable path-shape losses (Cuturi & Blondel, 2017; Le Guen & Thome, 2019). These methods target generic path geometry rather than any domain-specific structure. To our knowledge, we are the first to use a financial stylized fact as a differentiable path-level objective for adapting a pretrained foundation model.

3 METHOD

We propose *Volatility-Clustering Adaptation (VCA)*, which augments next-token prediction with a multi-step signal for volatility dependence. It matches the squared-return ACF of differentiable rollouts to that of the realized future, while preserving the original CE objective. We build on Kronos (Shi et al., 2026), a decoder-only foundation model for financial time series that tokenizes each open, high, low, close, and volume (OHLCV) bar into hierarchical discrete tokens and predicts the next bar autoregressively. Given a lookback window of W bars, Kronos rolls out an H-bar forecast by sampling tokens one step at a time and decoding them back to prices.

3.1 PRELIMINARIES

Problem Formulation. A training sequence consists of the tokenized $W+H$ bars, $x_{1:W+H}$, spanning the lookback window and the forecast horizon. The model p_θ is trained to predict each token autoregressively, giving the next-token cross-entropy (CE) objective

$$\mathcal{L}_{\mathrm{CE}} \;=\; -\sum_{t=1}^{W+H} \log p_\theta(x_t \mid x_{<t}). \tag{1}$$

This is the standard training objective for autoregressive time-series foundation models, and for financial foundation models such as Kronos in particular. The CE objective is teacher-forced and one step ahead; that is, it sharpens next-token accuracy, but gives no signal about the shape of the H-step rollout that is actually produced and evaluated at test time. We next describe one such shape property, *volatility clustering*, which Eq. 1 fails to model. Our method (§3.2) explicitly optimizes for this property.

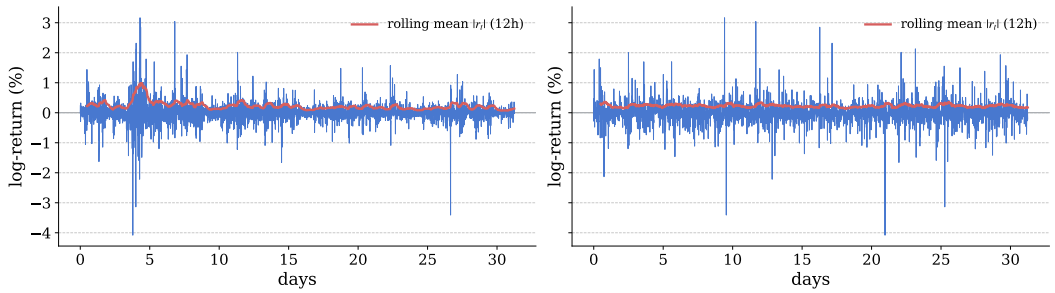


Figure 2: Meaning of $\hat{\rho}_k > 0$ on a real trajectory (BTCUSDT, 15-minute bars). Blue is the log-return r_t; red is the rolling mean $|r_t|$ (12h), a realized-volatility proxy. **(a)** Real order: volatility moves in sustained stretches, so a large $|r_t|$ predicts a large $|r_{t+k}|$. **(b)** Same returns, time-shuffled: the returns have the same values and distribution, but once their order is shuffled, the clustering disappears, and the envelope flattens. The ordering impact is the structure that $\hat{\rho}_k$ measures and what Eq. 5 trains the model to match.

The volatility-clustering stylized fact. Financial returns exhibit *volatility clustering*. The autocorrelation function (ACF) of *squared* returns decays slowly and stays positive, while raw returns are near white noise (Cont, 2001). For a price path $p_{1:H}$, we form log-returns $r_t = \log(p_t/p_{t-1})$ and let $\overline{r^2}$ denote their mean squared return. The sample ACF of squared returns at lag k is

$$\hat{\rho}_k \;=\; \frac{\sum_t (r_t^2 - \overline{r^2})(r_{t-k}^2 - \overline{r^2})}{\sum_t (r_t^2 - \overline{r^2})^2}, \qquad k = 1, \ldots, K. \tag{2}$$

K introduces a trade-off between capturing longer-range dependence and maintaining reliable estimates. Larger K captures slower-decaying clustering, but longer-lag estimates of $\hat{\rho}_k$ are noisier and sit closer to the no-clustering null. Concretely, $\hat{\rho}_k > 0$ means a large move tends to be followed by another large move k steps later, regardless of direction. Figure 2 illustrates this temporal dependence directly: the same returns exhibit sustained volatility clustering in their original order, but not after time-shuffling.

3.2 Volatility-Clustering Adaptation

Standard fine-tuning adapts foundation models using the objective in Eq. 1. We augment it with a signal for the volatility dependence of the multi-step rollout.

A differentiable ACF^2 loss. Computing the ACF loss on generated trajectories requires gradients to propagate through the autoregressive sampling process. Because Kronos predicts discrete tokens, naive sampling would break this gradient path. We therefore replace the categorical draw with a Gumbel-softmax straight-through estimator (Jang et al., 2017; Bengio et al., 2013). Specifically, letting ℓ_t denote the model logits over the token vocabulary, the forward pass feeds the $\arg\max$ token, matching test-time discretization, while the backward pass uses the soft relaxation

$$q_t \;=\; \mathrm{softmax}\big((\ell_t + g_t)/\tau\big), \qquad g_t \sim \mathrm{Gumbel}(0, 1), \tag{3}$$

with Gumbel-softmax temperature τ annealed over training. This yields a fully differentiable autoregressive rollout $\hat{p}_{1:H}$ (one sample per window at train time, versus an N-rollout average at test time), from which we compute the per-window $\hat{\rho}_k$ of Eq. 2 and penalize its discrepancy to the realized future squared-return ACF $\rho_k^\star$ of the ground-truth path.

A per-window loss $(\hat{\rho}_k - \rho_k^\star)^2$ is a high-variance target even for a perfectly calibrated model, because $\hat{\rho}_k$ is a sample ACF computed from as few as $H-k$ squared returns. We therefore average *before* squaring, over a batch of B windows indexed by b,

$$\bar{\rho}_k \;=\; \frac{1}{B}\sum_{b=1}^{B} \hat{\rho}_k^{(b)}, \qquad \bar{\rho}_k^\star \;=\; \frac{1}{B}\sum_{b=1}^{B} \rho_k^{\star(b)}, \tag{4}$$

which reduces the sampling variance of each term before the squared difference is taken. The *aggregate* training loss is then

$$\mathcal{L}_{\mathrm{ACF}^2} = \sum_{k=1}^{K} w_k \left(\bar{\rho}_k - \bar{\rho}_k^{\star}\right)^2, \qquad w_k = e^{-k/\gamma}. \tag{5}$$

We retain the general K-lag form as the natural statement of the stylized fact where near lags carry most of the stylized-fact signal and empirical persistence decays with lag (Cont, 2001) (Figure 1, panel 1), so w_k is a decay kernel rather than a normalized distribution over lags. When $K{=}1$ (our default), only $w_1 = e^{-1/\gamma}$ enters the loss, so λ and γ affect the loss only through the product $\lambda e^{-1/\gamma}$, while normalizing w_k would simply rescale λ. The total objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda\, \mathcal{L}_{\mathrm{ACF}^2}, \tag{6}$$

with plain CE recovered at $\lambda = 0$. The ACF^2 term is thus the only difference from the CE baseline, so any gap between the two is attributable to this signal. However, this signal constrains only the *dependence structure* of squared returns, not their overall *level*. In particular, it cannot distinguish opposite trends in volatility level. We formalize this property below.

Proposition 1 (The volatility-trend channel)**.** *Let $r_t = \sigma_0 e^{gt}\varepsilon_t$, where g controls the exponential trend in volatility, for $t = 1, \ldots, n$ with ε_t i.i.d. and no clustering. Then (a) the law of the sample ACF $\hat{\rho}_k(r^2)$ is invariant under $g \mapsto -g$, while (b) $\mathbb{E}\big[\sum_t r_t^2\big] = \sigma_0^2 \sum_{t=1}^{n} e^{2gt}$ is strictly increasing in g, with $\mathbb{E}\big[\sum_t r_t^2\,; +g\big] \,/\, \mathbb{E}\big[\sum_t r_t^2\,; -g\big] = e^{2g(n+1)}$.*

Proof. See Appendix B. □

A volatility trend can substantially change the overall level of squared returns without changing the distribution of their squared-return ACF. Matching the ACF therefore does not control the overall volatility level, leaving the objective insensitive to the direction of a volatility trend. This provides a mechanistic explanation for a failure mode observed in our VCA runs, where some runs end with sharply inflated forecast variance despite a small ACF^2 validation gap (Section 4.2). In practice, this effect is minor on our primary backbone, Kronos: VCA improves on the frozen model across all three datasets and horizons (Table 1), with variance inflation remaining modest relative to the magnitude possible in Proposition 1. The effect is more visible on Chronos-T5-small (Appendix Table 9), a smaller and more general backbone. We discuss this limitation further in Appendix D.

Why target the ACF gap, and not volatility level directly? Realized-variance level is a natural alternative target for a volatility-aware objective, but we find out empirically that its gap to a naive baseline is much less consistent across datasets than the ACF^2 gap. In contrast, ACF^2 shows a stable $\approx 2\times$ gap and is the only consistent performer across all three datasets, motivating our choice of ACF^2 as the training signal (Appendix Table 5). Intuitively, variance level is *scale-dependent* and can be matched reasonably well by a naive baseline that tracks the unconditional scale, whereas ACF^2 captures *scale-invariant* temporal dependence that such baselines cannot reproduce. This makes its headroom more stable across datasets and regimes.

4 Experiments

We evaluate VCA across datasets and forecasting horizons against alternative adaptation objectives and classical volatility models, using both finance-specific and general-purpose time-series foundation models. Our ablations then study three aspects of the method: how regime matching affects fine-tuning data efficiency, how adaptation changes model weights, and how the main design choices (K, W, and λ) affect performance.

4.1 Setup

Datasets. We evaluate on three asset sets in the per-symbol OHLCV format consumed by the Kronos inference pipeline: **CSI300** and **CSI500**, standard equity benchmarks in prior return-forecasting work (Li et al., 2024; Xu et al., 2021; Zhao et al., 2023) sourced from Qlib (Yang et al., 2020), and **Crypto**, 20 USDT pairs from the Binance API (Binance, 2026). More details are in Appendix Table 3.

Table 1: Main comparison under FORE. The Frozen row (for reference) is raw RankIC and MAE; GARCH(1,1) and HAR-RV forecast only variance, so RankIC and Score are not applicable to them (-). Every other row is a relative delta against Frozen in percentage (higher is better), and the last group averages both deltas and the Score over horizons. Bold marks the best fine-tuned method in each column. **WR** denotes win rate, computed as the fraction of the eight per-horizon columns in which a method has the best result. Per-horizon values are in Appendix Tables 15–18.

		H=8		H=16		H=32		H=48		average over H			
DS	Method	Δ_{RankIC}	Δ_{MAE}	Δ_{RankIC}	Δ_{MAE}	Δ_{RankIC}	Δ_{MAE}	Δ_{RankIC}	Δ_{MAE}	Δ_{RankIC}	Δ_{MAE}	Score	WR
CSI300	Frozen	+0.291	3.4e-3	+0.317	6.6e-3	+0.333	1.4e-2	+0.333	2.3e-2	-	-	-	-
	GARCH	-	-10.5	-	-13.3	-	-7.3	-	-0.6	-	-7.9	-	-
	HAR-RV	-	-6.2	-	-5.2	-	+7.9	-	+15.8	-	+3.1	-	-
	CE	-12.6	+6.7	**+7.5**	**-6.4**	-11.8	-11.8	-8.8	-8.5	-6.4	-5.0	-5.7	2/8
	MSE	-16.8	-5.3	+5.9	-13.0	-6.5	-11.8	**-1.7**	-4.7	-4.8	-8.7	-6.7	1/8
	VCA	**-6.0**	**+10.0**	+3.6	-19.1	**+8.0**	**-4.8**	-6.2	**+7.2**	**-0.2**	**-1.7**	**-0.9**	**5/8**
CSI500	Frozen	+0.849	4.5e-3	+0.853	9.0e-3	+0.864	2.0e-2	+0.880	3.1e-2	-	-	-	-
	GARCH	-	-11.4	-	-13.9	-	-6.7	-	-1.4	-	-8.3	-	-
	HAR-RV	-	-13.1	-	-9.0	-	+4.2	-	+12.0	-	-1.5	-	-
	CE	-3.4	**+6.1**	-4.5	+3.1	**-0.6**	-5.9	-8.3	-1.2	-4.2	+0.5	-1.9	2/8
	MSE	-4.0	-5.0	-2.7	-8.1	-3.3	-5.4	-5.7	-2.3	-3.9	-5.2	-4.6	0/8
	VCA	**-0.6**	-5.7	**+1.0**	**+4.5**	-1.3	**+3.9**	**-4.9**	**+3.4**	**-1.4**	**+1.5**	**+0.0**	**6/8**
Crypto	Frozen	+0.055	1.4e-4	+0.051	3.1e-4	+0.051	6.8e-4	+0.044	1.1e-3	-	-	-	-
	GARCH	-	-0.7	-	+9.2	-	+8.0	-	+12.6	-	+7.3	-	-
	HAR-RV	-	-13.1	-	-4.1	-	-0.8	-	+0.4	-	-4.4	-	-
	CE	-12.0	**+5.7**	-14.3	+6.2	-50.7	**+6.2**	-11.3	+4.5	-22.1	**+5.7**	-8.2	2/8
	MSE	-42.1	-0.5	-5.7	+0.4	**-2.7**	-6.6	+6.5	+3.6	-11.0	-0.8	-5.9	1/8
	VCA	**+3.4**	+3.9	**+7.3**	**+11.0**	-6.1	-4.8	**+10.2**	**+6.0**	**+3.7**	+4.0	**+3.9**	**5/8**

Metrics. For the path-wise quantity, we report price RankIC (**RankIC**), the per-window Spearman correlation between the predicted and realized OHLC price path, averaged over windows. For the volatility side, we report σ^2**-MAE**, the mean absolute error of realized variance. Both metrics follow the evaluation protocol of prior work (Shi et al., 2026). We use RankIC and σ^2-MAE for our main evaluation, while QLIKE (Patton, 2011) is reported as a complementary volatility metric in Appendix Table 6. The overall score is

$$\text{Score} \;=\; \tfrac{1}{2}\big(\Delta_{\text{RankIC}}\% + \Delta_{\text{MAE}}\%\big), \tag{7}$$

where Δ is relative to the frozen model, with both terms signed so that positive means improvement. Definitions of RankIC, σ^2-MAE and additional metrics (QLIKE, price IC, return IC/RankIC, and the ACF^2 gap, Eq. 5) are in Appendix C.

Baselines. We compare our method, **VCA**, with three methods: (1) the frozen pre-trained model **Frozen**; (2) plain cross-entropy fine-tuning **CE**; and (3) **MSE**, which keeps the identical differentiable AR rollout and replaces only $\mathcal{L}_{\text{ACF}^2}$ with a relative squared error on the rolled-out path, $\mathcal{L}_{\text{MSE}} = \frac{1}{H}\sum_{t=1}^{H}\big((\hat{p}_t - p_t^\star)/p_t^\star\big)^2$. We additionally compare VCA with two classical volatility models, GARCH(1,1) (Bollerslev, 1986) and HAR-RV (Corsi, 2009), which fit per symbol and forecast at the same windows.

Protocol. All methods are fully fine-tuned without adapters for the same number of epochs and learning rate over three seeds on 4 V100 GPUs, with window size fixed at W=160 and horizon H swept over $\{8, 16, 32, 48\}$. Checkpoints are selected by lowest teacher-forced validation cross-entropy for CE, and by lowest aggregate ACF^2 gap on autoregressive validation rollouts for VCA and MSE. Rollouts follow two sampling conventions: FORE (T=0.6, N=10) and VOL (T=0.9, N=1), the latter matching the setting Kronos uses for its own variance metric. The two share a single RankIC, computed once from the FORE rollout; only σ^2-MAE is recomputed per convention, so the FORE and VOL scores differ through that term alone. We report FORE in the main tables and VOL in the appendix (Table 7); QLIKE also uses FORE. More details are in Appendix A.

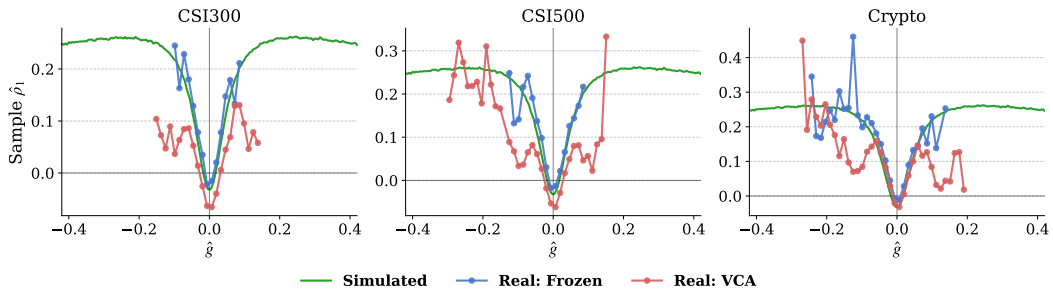


Figure 3: Volatility-trend channel, simulated vs. real rollouts (H=32), one panel per dataset. *Simulated* is $\mathbb{E}[\hat{\rho}_1]$ from $r_t = \sigma_0 e^{gt}\varepsilon_t$ (no genuine clustering) as g varies; *Real* shows the mean $\hat{\rho}_1$ within bins of the per-window trend rate $\hat{g}$ from the predicted rollout, computed as the log-ratio of second-half to first-half squared returns. We use second-half over first-half so that $\hat{g}$ has the same sign as g, following Proposition 1(b).

4.2 Main Results

VCA is the strongest fine-tuning objective in aggregate. VCA achieves the best average Score among fine-tuned methods and the highest win rate on all three datasets (Table 1). While aggregate results are reported for comprehensiveness, practical deployment typically targets a fixed horizon. At H=16, VCA consistently improves RankIC and achieves a higher average Score than the frozen model across datasets. Across horizons, VCA is also the only fine-tuned method to improve over the frozen model under σ^2-MAE. Compared with MSE, which shares the same rollout, VCA's margin isolates the gain to the ACF^2 loss itself rather than to rollout-based training in general. The margin over CE further shows that the ACF^2 signal adds information that next-token cross-entropy does not provide. It does so by moving the model further from the pre-trained weights, not closer (Section 4.3), indicating an additional training signal rather than a regularizer. Moreover, VCA outperforms both GARCH(1,1) and HAR-RV on CSI500, while each classical baseline is better on one of the other two datasets.

Under VOL (Appendix Table 7), VCA remains best on CSI300 and Crypto, with CSI500 as the only exception. Under QLIKE (Appendix Table 6), VCA shows the largest average improvement on CSI300 and CSI500 but not Crypto, where its residual under-dispersion is penalized more heavily. All fine-tuned methods trail both classical baselines on all three datasets under this metric. These per-symbol-fit models do not provide a price path, and hence no RankIC or Score, or a shared pre-trained backbone. VCA also achieves the lowest average ACF^2 gap at each horizon across datasets (Appendix Table 8).

Evaluation on new backbone. We further evaluate VCA on **Chronos-T5-small** (Ansari et al., 2024), a smaller and less-trained backbone, using the same fine-tuning setup at H=16 and H=32 across all three datasets (Appendix Table 9). VCA produces improvements on Crypto at H=16 and CSI500 at H=32, with the latter improving the average Score by 69.1% over the frozen baseline. The results are less stable than with Kronos, with rollout instability on CSI300 and Crypto at H=32, suggesting that VCA benefits more from domain-specific fine-tuning data than from a general-purpose backbone.

Empirical evidence for the volatility-trend channel. Figure 3 plots the sample ACF_1 of squared returns against the per-window trend rate $\hat{g}$ for both simulated and real rollouts. In the simulation, a deterministic trend with no true volatility clustering already produces a non-zero ACF when $|\hat{g}|$ is large, showing that trend alone can mimic the signal optimized by VCA. The Frozen model's real rollouts closely follow the same relationship across all three datasets, confirming that this effect also occurs in real model outputs. VCA's rollouts instead lie below the curve at matched $\hat{g}$ values and show substantially more scatter. This suggests that VCA does not simply increase $\hat{g}$ to exploit the objective, although the relationship does not fully explain the observed instability.

Table 2: Regime-matched definition. vol(·) is the standard deviation of log returns on close price (Appendix C). test/first and test/last are the ratios used to pick the matched slice (bold: closer to one); val/first and val/last are the same ratios computed from the validation period.The shaded column identifies whether the *first* or *last* slice is selected.

Dataset	Slice	vol(first)	vol(last)	vol(test)	vol(val)	test/first	test/last	val/first	val/last	matched
CSI300	10%	0.0298	0.0192	0.0253	0.0241	**0.85**×	1.32×	**0.81**×	1.26×	first
	1%	0.0287	0.0182	0.0253	0.0241	**0.88**×	1.39×	**0.84**×	1.33×	first
CSI500	10%	0.0299	0.0213	0.0282	0.0267	**0.94**×	1.32×	**0.89**×	1.25×	first
	1%	0.0235	0.0196	0.0282	0.0267	**1.20**×	1.44×	**1.13**×	1.36×	first
Crypto	10%	0.0105	0.0042	0.0051	0.0050	0.48×	**1.21**×	0.47×	**1.18**×	last
	1%	0.0133	0.0051	0.0051	0.0050	0.38×	**0.99**×	0.38×	**0.97**×	last

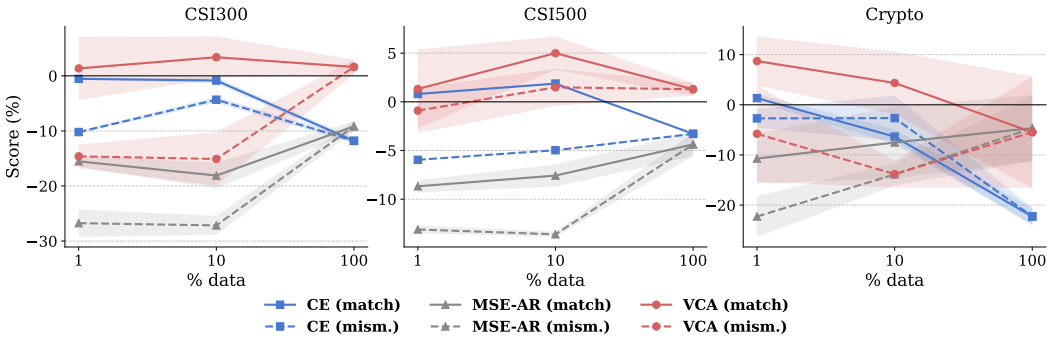


Figure 4: Data efficiency across all three methods, Score against the frozen model (H=32, FORE). Solid lines are the *regime-matched* slice and dashed the *mismatched* one, with bands of one standard deviation over 3 seeds. The regime-matched slices are identified in Table 2. Detailed values are in Appendix Table 11.

4.3 Ablations

A regime-matched slice makes fine-tuning data-efficient. Fine-tuning on a chronological slice (first or last) of the training data can outperform full-data fine-tuning when the slice's volatility regime matches the test period, suggesting an *alignment* effect rather than a purely *recency* effect. Table 2 determines the matched slice without evaluating the test period. The validation-based rule picks the same slice as the test-based rule in all six (dataset, slice-size) cells, showing that the selection can be made before deployment rather than after observing the test period. Volatility vol(·) is more discriminative than an ACF-based statistic, with raw lag-1 ACF of squared returns disagreeing across slice sizes on CSI300 (Appendix Table 10), consistent with noisier ACF estimates at this sample size (Section 3.2). The matched slice outperforms full-data fine-tuning across datasets and evaluation conventions at the tested slice sizes, while full-data fine-tuning is negative on Crypto (Figure 4). Mismatch is costly on Crypto and CSI300 but not on CSI500, so regime matching should be viewed as a useful selection rule rather than a universal penalty for mismatch.

This data-efficiency effect is specific to H=32 and does not transfer to H=16, where small-data fine-tuning remains substantially below the frozen model until the full dataset is used (Appendix Figure 6). The degradation reflects a drop in RankIC and affects CE as well as VCA, suggesting that broader data coverage is more important at shorter horizons, while regime alignment becomes more useful at longer horizons. We therefore view the two factors as complementary rather than treating regime alignment as a general substitute for data volume.

Does adaptation forget? The regime-matched slice matches full-data fine-tuning on the target, raising the question of how much additional weight drift the unused data introduces. We measure relative weight drift, $|\theta - \theta_0|_2/|\theta_0|_2$ from the frozen model θ_0, as a proxy for how much a model *could* have forgotten (Appendix Table 14). At full data and H=32, VCA produces larger weight drift

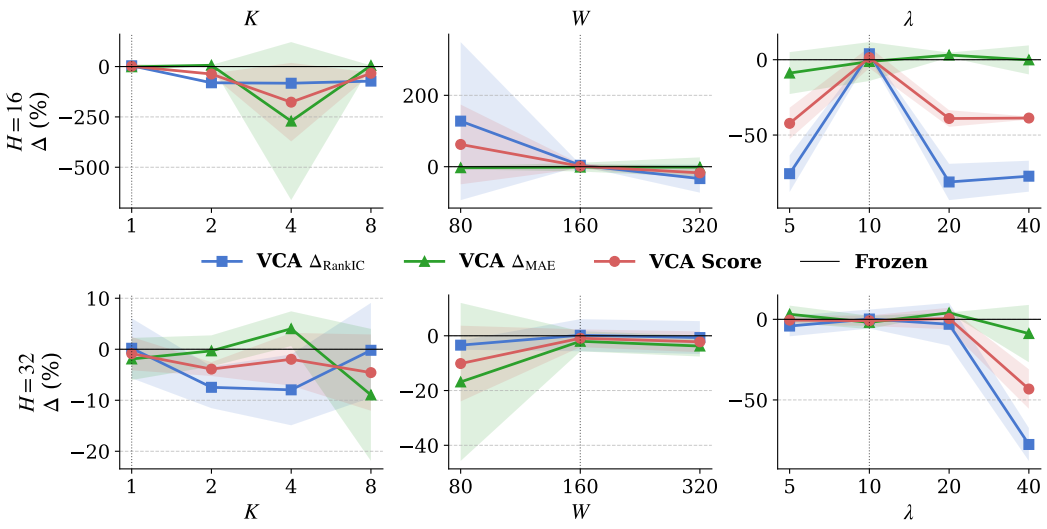


Figure 5: Ablation sweeps under FORE, mean over the three datasets, band is one standard deviation of the per-dataset delta. Δ_{RankIC} and Δ_{MAE} are the two halves Score averages; the black line at 0 is the frozen backbone. Detailed values are in Appendix Tables 12 and 13.

than CE, with a smaller difference at $H{=}16$. The regime-matched 10% slice reduces VCA's weight drift at both horizons while matching or outperforming full-data fine-tuning on the target. However, its drift remains comparable to or higher than CE, indicating that regime matching reduces, but does not eliminate, the additional drift associated with VCA. At $H{=}16$, full-data drift is larger in five of the six method–dataset combinations, with VCA on CSI300 as the only exception.

Effect of lag count K. The lag count has a stronger effect at $H{=}16$. The choice $K{=}1$ produces a sharp peak, while larger lag counts substantially reduce RankIC and increase seed-to-seed variation. The same ordering remains at $H{=}32$, but the performance gap is much smaller across the sweep, with $K{=}1$ still leading on all three datasets (Figure 5-left).

Effect of lookback W. The lookback shows a similar horizon-dependent pattern. At $H{=}16$, raw RankIC varies sharply with W and declines on both sides of the best-performing setting ($W{=}160$). The Δ-based Score in Figure 5 makes this contrast appear even larger because it is measured relative to the frozen backbone at the same W, whose performance is much lower with shorter context. At $H{=}32$, this dependence largely disappears, with raw RankIC remaining close across W values.

Effect of weight λ. The ACF^2 weight behaves differently from the other two hyperparameters. At $H{=}16$, the recipe value gives a narrow peak: reducing λ weakens the auxiliary signal mildly, whereas increasing it causes much larger RankIC losses before the effect saturates. This sharp sensitivity persists at $H{=}32$, where the largest tested weight again causes severe degradation. Thus, unlike W and K, λ remains sensitive across horizons.

5 Conclusion

We introduce *Volatility-Clustering Adaptation (VCA)*, which augments next-token prediction with a differentiable signal for the autocorrelation of squared returns over autoregressive rollouts. Across three asset sets and two evaluation conventions, VCA delivers the best average performance, primarily reflected in variance-related metrics. We further find that volatility-matched data can improve data efficiency at longer horizons, while broader data coverage remains important at shorter horizons. These findings highlight the value of adapting time-series foundation models with temporal objectives rather than relying on token-level prediction. Limitations are discussed in Appendix D.

AI USE STATEMENT

We used generative AI tools solely to support writing, including language polishing and clarity, not to develop ideas, design methods, or run experiments. All AI-assisted text was reviewed and verified by the authors, who take full responsibility for the final content.

ETHICS STATEMENT

VCA is a research recipe, not a validated trading system, so deploying it on live capital without independently verifying stability could turn a research-scale gain into a real loss. We use only public data and open pre-trained backbones, with no human subjects.

REPRODUCIBILITY STATEMENT

Datasets and implementation details are in Section 4, Table 3 and Appendix Table 4. Proof of Proposition 1 is in Appendix B. We will release the code upon publication.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Maximilian Beichter, Nils Friederich, Janik Pinter, Dorina Werling, Kaleb Phipps, Sebastian Beichter, Oliver Neumann, Ralf Mikut, Veit Hagenmeyer, and Benedikt Heidrich. Decision-focused fine-tuning of time series foundation models for dispatchable feeder optimization. *Energy and AI*, 21:100533, 2025. ISSN 2666-5468. doi: https://doi.org/10.1016/j.egyai.2025.100533. URL https://www.sciencedirect.com/science/article/pii/S2666546825000655.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in Neural Information Processing Systems*, 2015.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Binance. Binance spot API documentation. https://developers.binance.com/docs/binance-spot-api-docs, 2026. Accessed: 2026-09-15.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

Rama Cont. Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236, 2001.

Fulvio Corsi. A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196, 2009.

Marco Cuturi and Mathieu Blondel. Soft-DTW: a differentiable loss function for time-series. In *International Conference on Machine Learning*, 2017.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, 2024.

Feike C Drost and Theo E Nijman. Temporal aggregation of GARCH processes. *Econometrica*, 61 (4):909–927, 1993.

Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

Matthew Faw, Rajat Sen, Yichen Zhou, and Abhimanyu Das. In-context fine-tuning for time-series foundation models. In *International Conference on Machine Learning*, 2025.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *International Conference on Machine Learning*, 2024.

Divij Gupta, Anubhav Bhatti, Suraj Parmar, Chen Dan, Yuwei Liu, Bingjie Shen, and San Lee. Low-rank adaptation of time series foundational models for out-of-domain modality forecasting. In *Proceedings of the 26th International Conference on Multimodal Interaction*, ICMI '24, pp. 382–386, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704628. doi: 10.1145/3678957.3685724. URL https://doi.org/10.1145/3678957.3685724.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Feiyang Kang, Hoang Anh Just, Yifan Sun, Himanshu Jahagirdar, Yuanzhi Zhang, Rongxing Du, Anit Kumar Sahu, and Ruoxi Jia. Get more for less: Principled data selection for warming up fine-tuning in LLMs. In *International Conference on Learning Representations*, 2024.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

Alex M Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. *Advances in Neural Information Processing Systems*, 2016.

Vincent Le Guen and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. *Advances in Neural Information Processing Systems*, 2019.

Thomas L Lee, William Toner, Rajkarn Singh, Artjom Joosen, and Martin Asenov. Lightweight online adaption for time series foundation model forecasts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33736–33764. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/lee25ag.html.

Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. MASTER: Market-guided stock transformer for stock price forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Yuze Li and Wei Zhu. TRACE: Time series parameter efficient fine-tuning. *Neurocomputing*, pp. 132098, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Andrew J Patton. Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1):246–256, 2011.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

Yu Shi, Zongliang Fu, Shuo Chen, Bohan Zhao, Wei Xu, Changshui Zhang, and Jian Li. Kronos: A foundation model for the language of financial markets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 25366–25373, 2026.

Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant GANs: Deep generation of financial time series. *Quantitative Finance*, 20(9):1419–1440, 2020.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning*, 2024.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning*, 2024.

Wentao Xu, Weiqing Liu, Lewen Wang, Yingce Xia, Jiang Bian, Jian Yin, and Tie-Yan Liu. HIST: A graph-based framework for stock trend forecasting via mining concept-oriented shared information. *arXiv preprint arXiv:2110.13716*, 2021.

Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. Qlib: An ai-oriented quantitative investment platform. *arXiv preprint arXiv:2009.11189*, 2020.

Lifan Zhao, Shuming Kong, and Yanyan Shen. DoubleAdapt: A meta-learning approach to incremental learning for stock trend forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

APPENDIX

A ADDITIONAL IMPLEMENTATION DETAILS

We summarize dataset details in Table 3 and hyperparameters in Table 4.

GARCH/HAR-RV implementation. Both baselines are fit once per symbol on the combined training and validation split, as they have no checkpoint to select and therefore require no separate validation-based calibration. Each baseline then forecasts at exactly the same strided test windows Kronos uses (same lookback=160, stride, and test period per dataset). GARCH(1,1) is fit by maximum likelihood estimation; HAR-RV regresses realized variance on its own lags $\{1, 5, 22\}$ by ordinary least squares. Lags are bar-level, not calendar day/week/month, matching how QLIKE and σ^2 are defined throughout this paper (Appendix C).

B PROOF OF PROPOSITION 1

Proof. Write

$$X_t^{(g)} := r_t^2 = \sigma_0^2 e^{2gt} \varepsilon_t^2, \qquad t = 1, \dots, n, \tag{8}$$

let $X^{(g)} := X_{1:n}^{(g)}$, and let $\mathcal{R}x_{1:n} := (x_n, \dots, x_1)$ denote time reversal. Since ε_t^2 is non-degenerate, the denominator of Eq. 2 evaluated at $X^{(g)}$ is a.s. positive, so $\hat{\rho}_k(X^{(g)})$ is a.s. well defined.

We first record two elementary invariances of Eq. 2, valid for every $x_{1:n}$ with positive sample variance, every $k \in \{1, \dots, n-1\}$ and every $c \neq 0$:

$$\hat{\rho}_k(\mathcal{R}x_{1:n}) = \hat{\rho}_k(x_{1:n}), \tag{9}$$

$$\hat{\rho}_k(c\, x_{1:n}) = \hat{\rho}_k(x_{1:n}). \tag{10}$$

Eq. 9 holds because reversal is a permutation of the entries, hence leaves $\bar{x}$ and $\sum_t (x_t - \bar{x})^2$ unchanged, while in the numerator the substitution $s = n + 1 - t - k$ is a bijection of $\{1, \dots, n-k\}$ onto itself carrying the pair $\big((\mathcal{R}x)_t, (\mathcal{R}x)_{t+k}\big) = (x_{n+1-t}, x_{n+1-t-k})$ to (x_{s+k}, x_s), so that reversal merely swaps the two coordinates of each lag-k product. Eq. 10 holds because rescaling by c multiplies the numerator and the denominator of Eq. 2 by the same factor c^2.

(a) Reversing Eq. 8 gives the pathwise identity

$$\big(\mathcal{R}X^{(g)}\big)_t = X_{n+1-t}^{(g)} = e^{2g(n+1)}\, \sigma_0^2 e^{-2gt} \varepsilon_{n+1-t}^2, \qquad t = 1, \dots, n, \tag{11}$$

that is, $\mathcal{R}X^{(g)} = e^{2g(n+1)} Y$ with $Y_t := \sigma_0^2 e^{-2gt} \varepsilon_{n+1-t}^2$ and $e^{2g(n+1)} > 0$. Combining Eq. 11 with Eqs. 9–10,

$$\hat{\rho}_k\Big(X^{(g)}\Big) = \hat{\rho}_k\Big(\mathcal{R}X^{(g)}\Big) = \hat{\rho}_k(Y) \qquad \text{a.s., for every } k. \tag{12}$$

Since the ε_t are i.i.d., they are exchangeable, so

$$(\varepsilon_n^2, \dots, \varepsilon_1^2) \overset{d}{=} (\varepsilon_1^2, \dots, \varepsilon_n^2); \tag{13}$$

applying the deterministic map $(u_1, \dots, u_n) \mapsto (\sigma_0^2 e^{-2gt} u_t)_{t=1}^n$ to both sides of Eq. 13 yields $Y \overset{d}{=} X^{(-g)}$ as random vectors in $\mathbb{R}^n$. As $\hat{\rho}_k$ is measurable, Eq. 12 gives

$$\hat{\rho}_k\Big(X^{(g)}\Big) \overset{d}{=} \hat{\rho}_k\Big(X^{(-g)}\Big), \tag{14}$$

jointly over any finite set of lags, proving (a).

(b) Since $\mathbb{E}[\varepsilon_t^2] = 1$,

$$\mathbb{E}\left[\sum_{t=1}^n r_t^2\right] = \sigma_0^2 \sum_{t=1}^n e^{2gt} =: S_n(g), \tag{15}$$

which is strictly increasing in g because

$$S_n'(g) = 2\sigma_0^2 \sum_{t=1}^{n} t\, e^{2gt} > 0 \qquad \text{for all } g \in \mathbb{R}. \tag{16}$$

Finally, setting $q := e^{2g} > 0$, the substitution $s = n + 1 - t$ gives

$$\sum_{t=1}^{n} q^{-t} = \sum_{s=1}^{n} q^{-(n+1-s)} = q^{-(n+1)} \sum_{s=1}^{n} q^{s}, \tag{17}$$

and hence

$$\frac{\mathbb{E}\big[\sum_t r_t^2\,;\, +g\big]}{\mathbb{E}\big[\sum_t r_t^2\,;\, -g\big]} = \frac{\sum_{t=1}^{n} q^t}{\sum_{t=1}^{n} q^{-t}} = q^{\,n+1} = e^{2g(n+1)}, \tag{18}$$

proving (b). □

C METRIC DEFINITIONS

Notation: $p_{1:H}$ is a close-price path over the H-bar test window, p_0 the anchor bar, $r_t = \log(p_t/p_{t-1})$ the log-return, $\hat{}$ predictions, $\star$ realized values, and $\mathcal{W}$ the set of (symbol, test-window) pairs.

Rank IC. Spearman correlation between predicted and realized paths over the H steps, averaged across the four OHLC channels and all windows:

$$\text{RankIC} \;=\; \frac{1}{|\mathcal{W}|} \sum_{w \in \mathcal{W}} \frac{1}{4} \sum_{c \in \{O,H,L,C\}} \text{Spearman}\big(\hat{p}^{(c)}_{1:H,w},\, p^{\star(c)}_{1:H,w}\big). \tag{19}$$

Replacing Spearman with Pearson correlation creates *price IC*.

σ^2-MAE. Realized variance of a path is the sum of squared log-returns, with no annualization or square root:

$$\sigma^2(p_{1:H}) \;=\; \sum_{t=1}^{H-1} \big(\log p_{t+1} - \log p_t\big)^2, \tag{20}$$

and the metric is its absolute error against the realized value, with the prediction averaged over the N test-time rollouts:

$$\sigma^2\text{-MAE} \;=\; \frac{1}{|\mathcal{W}|} \sum_{w \in \mathcal{W}} \big|\sigma^2(\hat{p}_{1:H,w}) - \sigma^2(p^{\star}_{1:H,w})\big|. \tag{21}$$

Parkinson volatility estimator. Used only in Table 5, alongside the same $\sigma^2(\cdot)$ already defined above ("Vol. level (close)"), as an alternative realized-variance proxy from each bar's intra-bar high H_t and low L_t rather than close-to-close returns ("Vol. level (Parkinson)"):

$$\sigma^2_{\text{park}}(p_{1:H}) \;=\; \sum_{t=1}^{H} \Big(\log \tfrac{H_t}{L_t}\Big)^2. \tag{22}$$

For both proxies, Table 5 scores the model with Model/Naive: the model's squared error against always predicting the training-set mean $\overline{\sigma^{2\star}}$, equivalently $1 - R^2$:

$$\text{Model/Naive} \;=\; \frac{\sum_{w \in \mathcal{W}} \big(\hat{\sigma}^2_w - \sigma^{2\star}_w\big)^2}{\sum_{w \in \mathcal{W}} \big(\sigma^{2\star}_w - \overline{\sigma^{2\star}}\big)^2} \;=\; 1 - R^2. \tag{23}$$

QLIKE. The standard robust loss for a noisy realized-variance proxy (Patton, 2011): writing $\sigma^{2\star} = \sigma^2(p^{\star}_{1:H,w})$ and $\hat{\sigma}^2 = \sigma^2(\hat{p}_{1:H,w})$ for the realized and predicted variance of a window (prediction again averaged over the N test-time rollouts),

$$\text{QLIKE} \;=\; \frac{1}{|\mathcal{W}|} \sum_{w \in \mathcal{W}} \left(\frac{\sigma^{2\star}_w}{\hat{\sigma}^2_w} - \log \frac{\sigma^{2\star}_w}{\hat{\sigma}^2_w} - 1 \right), \tag{24}$$

which is 0 at $\hat{\sigma}^2_w = \sigma^{2\star}_w$ and grows asymmetrically faster as $\hat{\sigma}^2_w \to 0$ (under-prediction) than as $\hat{\sigma}^2_w \to \infty$ (over-prediction). As with σ^2-MAE, both terms are winsorized at the 99th percentile per (dataset, convention) cell before averaging.

Regime-match volatility, $\mathrm{vol}(\cdot)$**.** Used to define the regime-matched slice (Table 2, Section 4.3). For a period $\mathcal{P}$ (train-first-10%, train-last-10%, validation, or test) let $r_t^{(s)} = \log(p_t^{(s)}/p_{t-1}^{(s)})$ be the close-price log-return of symbol s at bar t, pooled over all symbols and bars in $\mathcal{P}$, with mean $\bar{r}_{\mathcal{P}}$. Then

$$\mathrm{vol}(\mathcal{P}) \;=\; \sqrt{\frac{1}{|\mathcal{P}|}\sum_{(s,t)\in\mathcal{P}} \big(r_t^{(s)} - \bar{r}_{\mathcal{P}}\big)^2}, \tag{25}$$

where $\bar{r}_{\mathcal{P}}$ is the mean return over the same observations. We do not annualize this quantity because only within-dataset ratios of $\mathrm{vol}(\cdot)$ are used, so the common annualization factor would cancel.

Return IC / Return RankIC. The end-of-window simple return is $\hat{r}_{s,t} = \hat{p}_{H,s,t}/p_{0,s,t} - 1$ for the prediction and $r^\star_{s,t} = p^\star_{H,s,t}/p_{0,s,t} - 1$ for the realized path. Both metrics correlate these returns *cross-sectionally* over the symbols S_t with a valid window anchored at test timestamp t (a calendar date for daily equity bars, a 15-min bar for crypto), then average over timestamps:

$$\mathrm{IC}_t = \mathrm{Pearson}\big(\{\hat{r}_{s,t}\}_{s\in S_t},\, \{r^\star_{s,t}\}_{s\in S_t}\big), \qquad \mathrm{IC} = \frac{1}{|T|}\sum_{t\in T}\mathrm{IC}_t, \tag{26}$$

$$\mathrm{RankIC}_t = \mathrm{Spearman}\big(\{\hat{r}_{s,t}\}_{s\in S_t},\, \{r^\star_{s,t}\}_{s\in S_t}\big), \qquad \mathrm{RankIC} = \frac{1}{|T|}\sum_{t\in T}\mathrm{RankIC}_t. \tag{27}$$

ACF^2 gap. Let $\hat{\rho}_{k,w}$ and $\rho^\star_{k,w}$ be the lag-k sample autocorrelations within window w (Eq. 2). The gap squares the per-lag discrepancy, averages over the K lags, then over windows:

$$\mathrm{ACF}^2\text{-gap} \;=\; \frac{1}{|\mathcal{W}|}\sum_{w\in\mathcal{W}}\frac{1}{K}\sum_{k=1}^{K}\big(\hat{\rho}_{k,w} - \rho^\star_{k,w}\big)^2. \tag{28}$$

The training loss differs, where the ACFs are averaged over a batch before the squared difference is taken.

D LIMITATIONS

First, VCA's advantage appears conditional on a well-pre-trained backbone, and the gains largely disappear on Chronos-t5-small, whose zero-shot RankIC is an order of magnitude lower.

Second, although VCA is strongest on average, its gains are not consistent across benchmarks, and the scaling trend remains unclear, leaving room for future work.

Table 3: Dataset details. Equity universes are the survivorship-free point-in-time membership union sourced via Qlib, and Crypto is from the Binance API.

Dataset	Source	Freq.	#Symbols	Splits	#Samples ($n_{\mathrm{tr}}/n_{\mathrm{val}}/n_{\mathrm{te}}$)
Crypto top-20	Binance API	15-min	20	2021–23 / 2024H1 / 2024H2–25	63514 / 10860 / 19308
CSI300	Qlib	daily	515	2015–17 / 2018 / 2019–20	3991 / 1836 / 3557
CSI500	Qlib	daily	937	2015–17 / 2018 / 2019–20	6246 / 2991 / 5749

Table 4: Hyperparameter details.

	Setting	Value
Backbone	Model	Kronos-base (102M), decoder-only
	Adaptation	full fine-tuning, no adapters
Optimization	Epochs	10
	Learning rate	5×10^{-5}
	Optimizer	$\beta = (0.9,\ 0.95)$, AdamW (Loshchilov & Hutter, 2019)
	Weight decay	0.1
	Gradient clipping	3.0
	Seeds	$\{1, 2, 3\}$
	Effective batch	32
Data	Lookback W	160 bars
	Horizon H	$\{8, 16, 32, 48\}$
	Max context	512
	Return clipping	$\pm 5.0\sigma$
	Eval stride	16 bars
VCA objective	Weight λ_{acf}	10
	Lags K	1
	Lag weights	$w_k = e^{-k/\gamma}$, $\gamma = 2.0$
	Target form	aggregate (batch-mean ACF matched)
	Rollout	fully differentiable autoregressive
	Gumbel-softmax	straight-through, τ decay $0.5 \to 0.1$
Selection	CE	lowest teacher-forced validation CE
	VCA, MSE	lowest AR-validation aggregate ACF^2 gap
Evaluation	FORE convention	T=0.6, N=10 rollouts
	VOL convention	T=0.9, N=1 rollout
	ACF lags (metric)	15

Table 5: Headroom on the frozen backbone at H=16, model error against a naive baseline. ACF^2: model gap vs. a $\hat{\rho}$=0 (no-clustering) floor. Vol. level (close/Parkinson): model realized-variance error vs. always predicting the training-set mean. Detailed metrics are in Appendix C.

	ACF^2 gap		Vol. level (close)		Vol. level (Parkinson)	
Dataset	gap	Model/Naive	R^2	Model/Naive	R^2	Model/Naive
CSI300	0.0392	1.96×	−4.62	5.62×	−0.70	1.70×
CSI500	0.0394	1.95×	−3.17	4.17×	−1.60	2.60×
Crypto	0.0417	1.93×	+0.02	0.98×	+0.17	0.83×

Table 6: Secondary comparison, QLIKE, computed under the FORE sampling convention. Frozen's row is the raw QLIKE; every other row is a relative delta against Frozen, $\Delta_{\text{QLIKE}}\%$ signed so positive means improvement, with the last column averaging over horizons. Bold marks the best fine-tuned method in each column, and **WR** is the win rate over the 4 per-horizon columns to its left.

		H=8	H=16	H=32	H=48	average over H	
DS	Method	Δ_{QLIKE}	Δ_{QLIKE}	Δ_{QLIKE}	Δ_{QLIKE}	Δ_{QLIKE}	WR
CSI300	Frozen	3.13	3.36	4.65	6.04	-	-
	GARCH	+84.3	+88.5	+92.6	+94.8	+90.1	-
	HAR-RV	+84.3	+88.6	+93.2	+95.3	+90.3	-
	CE	-27.0	-49.4	-90.5	-69.9	-59.2	0/4
	MSE	-257.1	-128.4	-97.1	-14.7	-124.3	0/4
	VCA	**-3.7**	**+7.8**	**-11.4**	**+46.3**	**+9.8**	**4/4**
CSI500	Frozen	3.32	3.66	4.91	6.35	-	-
	GARCH	+84.1	+88.5	+92.3	+94.4	+89.8	-
	HAR-RV	+83.1	+87.9	+92.6	+94.8	+89.6	-
	CE	-3.4	**+5.5**	-52.9	-7.2	-14.5	1/4
	MSE	-184.9	-89.7	-65.7	-16.6	-89.2	0/4
	VCA	**+7.4**	+4.2	**+18.6**	**+28.9**	**+14.8**	**3/4**
Crypto	Frozen	4.84	6.77	10.14	11.72	-	-
	GARCH	+91.5	+94.8	+96.9	+97.3	+95.1	-
	HAR-RV	+88.3	+92.7	+95.5	+96.1	+93.1	-
	CE	**+47.5**	+47.8	**+52.6**	**+47.7**	**+48.9**	**3/4**
	MSE	-5.6	-1.3	-787.4	+14.8	-194.9	0/4
	VCA	+27.3	**+61.8**	+38.3	+0.2	+31.9	1/4

Table 7: Same layout as Table 1, with σ^2-MAE recomputed under VOL in place of FORE. RankIC is unchanged from Table 1, since it is computed once from the FORE rollout and shared across conventions; only Δ_{MAE} and the resulting Score differ. Frozen row is the same as in Table 1.

		H=8		H=16		H=32		H=48		average over H			
DS	Method	Δ_{RankIC}	Δ_{MAE}	Δ_{RankIC}	Δ_{MAE}	Δ_{RankIC}	Δ_{MAE}	Δ_{RankIC}	Δ_{MAE}	Δ_{RankIC}	Δ_{MAE}	$\text{Score}_{\text{MAE}}$	WR
CSI300	Frozen	+0.291	3.4e-3	+0.317	6.1e-3	+0.333	1.3e-2	+0.333	2.0e-2	-	-	-	-
	GARCH	-	-11.1	-	-22.4	-	-22.1	-	-17.3	-	-18.2	-	-
	HAR-RV	-	-6.8	-	-13.7	-	-4.7	-	+1.8	-	-5.8	-	-
	CE	-12.6	+8.7	**+7.5**	**-1.0**	-11.8	-8.3	-8.8	-9.8	-6.4	**-2.6**	-4.5	2/8
	MSE	-16.8	-1.9	+5.9	-15.6	-6.5	-19.7	**-1.7**	-12.8	-4.8	-12.5	-8.7	1/8
	VCA	**-6.0**	**+8.8**	+3.6	-32.6	**+8.0**	**-2.6**	-6.2	**-2.8**	**-0.2**	-7.3	**-3.8**	**5/8**
CSI500	Frozen	+0.849	4.5e-3	+0.853	8.4e-3	+0.864	1.7e-2	+0.880	2.7e-2	-	-	-	-
	GARCH	-	-10.9	-	-22.1	-	-22.0	-	-17.9	-	-18.2	-	-
	HAR-RV	-	-12.6	-	-16.9	-	-9.6	-	-2.3	-	-10.3	-	-
	CE	-3.4	**+6.2**	-4.5	**+3.9**	**-0.6**	**-3.4**	-8.3	**-2.4**	-4.2	**+1.1**	**-1.6**	**5/8**
	MSE	-4.0	-1.8	-2.7	-11.0	-3.3	-14.0	-5.7	-12.2	-3.9	-9.8	-6.9	0/8
	VCA	**-0.6**	-35.0	**+1.0**	+2.6	-1.3	-4.4	**-4.9**	-23.2	**-1.4**	-15.0	-8.2	3/8
Crypto	Frozen	+0.055	1.2e-4	+0.051	2.7e-4	+0.051	5.8e-4	+0.044	9.0e-4	-	-	-	-
	GARCH	-	-11.2	-	-5.4	-	-8.4	-	-2.8	-	-7.0	-	-
	HAR-RV	-	-24.8	-	-20.8	-	-18.8	-	-17.2	-	-20.4	-	-
	CE	-12.0	-1.8	-14.3	**+3.4**	-50.7	**+7.4**	-11.3	**+5.0**	-22.1	**+3.5**	-9.3	3/8
	MSE	-42.1	-9.1	-5.7	-10.1	**-2.7**	-21.5	+6.5	-7.5	-11.0	-12.1	-11.6	1/8
	VCA	**+3.4**	**+0.6**	**+7.3**	-30.2	-6.1	-12.4	**+10.2**	-8.7	**+3.7**	-12.7	**-4.5**	**4/8**

Table 8: Aggregate ACF^2 gap by horizon and dataset (lower is better). Bold marks the lowest (best) arm in each row. $^\dagger H{=}8$ gives only 7 squared returns per window, short of the $K{=}15$ lags used here (needs at least $2K$ squared returns), reported for completeness.

H	Dataset	CE	MSE	VCA
$8^\dagger$	CSI300	.05772±.00042	.05949±.00082	**.05749±.00028**
	CSI500	.05651±.00037	.05821±.00091	**.05419±.00070**
	Crypto	**.06436±.00007**	.06773±.00110	.06456±.00006
16	CSI300	**.03517±.00015**	.03663±.00013	.03523±.00053
	CSI500	.03804±.00020	.04019±.00061	**.03730±.00045**
	Crypto	.03958±.00008	.04275±.00091	**.03885±.00026**
32	CSI300	.01771±.00004	.01850±.00014	**.01749±.00003**
	CSI500	.01773±.00005	.01843±.00017	**.01755±.00007**
	Crypto	**.01847±.00003**	.02076±.00116	.01852±.00021
48	CSI300	.01466±.00003	.01489±.00014	**.01417±.00018**
	CSI500	.01478±.00008	.01541±.00001	**.01452±.00036**
	Crypto	**.01692±.00001**	.01872±.00056	.01754±.00192
wins		4/12	0/12	**8/12**
mean		.03264	.03431	**.03228**

Table 9: Chronos backbone, $H{=}16, 32$. **Bold** indicates best score across fine-tuning methods, underlined marks a cell where σ^2-MAE (FORE) is orders of magnitude above Frozen; ΔRankIC does not depend on that blown-up price scale. Full per-cell details in Tables 19–20.

		$H{=}16$			$H{=}32$			
DS	Method	Δ_{RankIC}	Δ_{MAE}	Score	Δ_{RankIC}	Δ_{MAE}	Score	WR
CSI300	Frozen	+0.0181	7.4e-3	-	+0.0042	1.6e-2	-	-
	CE	-70.7	**-2.2**	-36.4	**+364.3**	**-2.2**	**+181.1**	**4/6**
	MSE	**+18.8**	-3.9	**+7.4**	+181.0	-2.5	+89.2	2/6
	VCA	-248.4	-1649079.1	-824663.8	-512.9	-133064.2	-66788.5	0/6
CSI500	Frozen	+0.0537	9.9e-3	-	+0.0534	2.2e-2	-	-
	CE	**-34.6**	**-0.6**	**-17.6**	-126.6	**-0.7**	-63.7	**4/6**
	MSE	-235.9	-2.9	-119.4	-241.8	-2.0	-121.9	0/6
	VCA	-91.2	-265545.6	-132818.4	**+138.8**	-0.5	**+69.1**	2/6
Crypto	Frozen	-0.0017	2.7e-4	-	+0.0049	6.0e-4	-	-
	CE	**+970.6**	**-5.9**	**+482.3**	**+230.6**	-6.5	**+112.0**	**5/6**
	MSE	+747.1	-8.9	+369.1	-173.5	**-6.2**	-89.9	1/6
	VCA	+370.6	-6.7	+181.9	-77.5	-2512388.9	-1256233.2	0/6

Table 10: Same regime-match exercise as Table 2, with the raw lag-1 sample ACF of squared returns $\hat{\rho}_1(\cdot)$ in place of $\mathrm{vol}(\cdot)$. Bold marks a cell where $\hat{\rho}_1$ disagrees with $\mathrm{vol}(\cdot)$'s match in Table 2.

Dataset	Slice	$\hat{\rho}_1$(first)	$\hat{\rho}_1$(last)	$\hat{\rho}_1$(test)	$\hat{\rho}_1$(val)	test-closer / val-closer
CSI300	10%	+0.099	+0.102	+0.041	+0.054	first / first
	1%	+0.070	+0.058	+0.041	+0.054	**last / last**
CSI500	10%	+0.172	+0.127	+0.047	+0.050	**last / last**
	1%	+0.148	+0.074	+0.047	+0.050	last / last
Crypto	10%	+0.306	+0.134	+0.207	+0.210	last / last
	1%	+0.247	+0.179	+0.207	+0.210	last / last

Table 11: Full per-cell numbers behind Figure 4, under both evaluation conventions. Bold marks the best method in each column within a (convention, dataset). Shaded cells indicate *matched regime*.

	Dataset	Method	1%-first	1%-last	10%-first	10%-last	100%
RankIC	CSI300	CE	**+1.6**±0.2	**-18.3**±0.4	**-0.1**±1.1	**+0.7**±1.1	-11.8±1.1
		MSE	-16.5±1.7	-32.9±4.5	-22.0±6.4	-34.7±4.1	-6.5±1.5
		VCA	-1.3±1.3	-23.0±4.2	-0.4±12.4	-28.7±9.7	**+8.0**±1.0
	CSI500	CE	**+2.0**±0.1	-5.9±0.1	-0.1±0.1	-3.6±0.1	**-0.6**±0.1
		MSE	-5.1±0.6	-10.2±0.1	-6.5±0.8	-10.0±0.2	-3.3±0.8
		VCA	+1.0±1.0	**-0.5**±2.6	**+0.2**±1.1	**-1.8**±3.0	-1.3±2.1
	Crypto	CE	**-10.7**±4.0	-1.0±1.6	**-8.7**±9.0	-16.7±2.9	-50.7±3.5
		MSE	-41.1±8.4	-23.8±10.7	-25.8±4.1	-17.3±14.5	**-2.7**±12.0
		VCA	-15.2±19.0	**+12.2**±13.5	-33.5±4.6	**+7.0**±12.9	-6.1±19.0
FORE	CSI300	CE	-0.5±0.1	-10.2±0.2	-0.8±0.5	**-4.3**±0.5	-11.8±0.5
		MSE	-15.5±1.0	-26.8±2.5	-18.1±2.3	-27.2±1.7	-9.2±0.9
		VCA	**+1.3**±5.8	**-14.6**±2.1	**+3.4**±3.8	-15.1±4.9	**+1.6**±1.3
	CSI500	CE	+0.8±0.0	-6.0±0.0	+1.9±0.0	-5.0±0.1	-3.3±0.1
		MSE	-8.7±0.6	-13.1±0.4	-7.6±1.1	-13.6±0.3	-4.4±1.2
		VCA	**+1.3**±4.1	**-0.9**±2.3	**+5.0**±1.7	**+1.5**±1.9	**+1.3**±0.7
	Crypto	CE	**-2.7**±1.9	+1.3±0.8	**-2.7**±4.5	-6.4±1.5	-22.3±1.7
		MSE	-22.3±4.1	-10.7±5.0	-13.9±2.2	-7.5±7.1	**-4.7**±6.6
		VCA	-5.8±9.6	**+8.7**±4.9	-13.8±2.7	**+4.4**±6.3	-5.5±11.1
VOL	CSI300	CE	**+1.0**±0.1	**-8.6**±0.2	-1.6±1.2	**-1.8**±0.6	-10.0±0.5
		MSE	-18.6±1.0	-30.4±2.9	-21.7±1.8	-30.0±1.4	-13.1±0.9
		VCA	-1.3±0.9	-12.8±1.8	**+2.5**±5.6	-13.6±4.7	**+2.7**±4.0
	CSI500	CE	**+1.5**±0.2	-4.1±0.0	-0.2±0.1	-2.4±0.2	**-2.0**±0.2
		MSE	-12.5±0.8	-16.7±0.7	-11.5±1.6	-17.4±0.7	-8.7±1.3
		VCA	+0.1±2.1	**-1.2**±2.8	**+3.1**±1.0	**+3.4**±3.1	-2.9±4.1
	Crypto	CE	**-1.4**±1.6	+2.9±0.9	**-2.4**±4.5	-5.8±1.5	-21.7±1.7
		MSE	-25.9±3.7	-12.0±5.0	-17.9±2.1	-9.4±7.3	-12.1±7.3
		VCA	-4.7±9.4	**+6.4**±3.4	-21.6±10.6	+1.7±6.6	**-9.3**±12.8

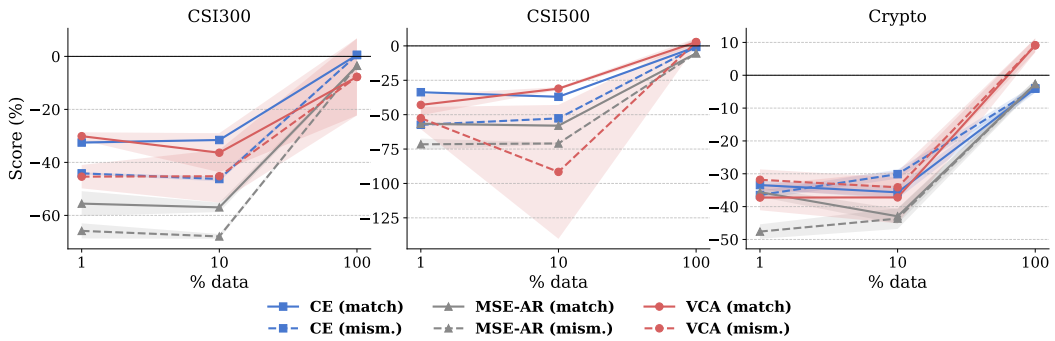


Figure 6: Same layout as Figure 4, at H=16: every fine-tuning method stays well below the frozen model at 1% and 10% data, matched or mismatched, recovering only at 100%.

Table 12: Per-cell values for ablation at H=16, under FORE, mean over three seeds. *No setting is uniformly optimal across all three datasets.*

Ablation	Value	CSI300		CSI500		Crypto	
		RankIC	σ^2-MAE	RankIC	σ^2-MAE	RankIC	σ^2-MAE
Lags K	**1**	+0.3280	0.00788	+0.8615	0.00860	+0.0552	0.00028
	2	+0.0480	0.00614	+0.1242	0.00834	+0.0147	0.00030
	4	+0.0447	0.00626	+0.1437	0.00830	+0.0106	0.00289
	8	+0.0818	0.00626	+0.1582	0.00859	+0.0199	0.00030
Lookback W	80	+0.0902	0.00652	+0.0774	0.00824	+0.0072	0.00031
	160	+0.3280	0.00754	+0.8615	0.00852	+0.0552	0.00028
	320	+0.0773	0.01192	+0.1930	0.00849	+0.0073	0.00029
Weight λ	5	+0.1312	0.00669	+0.1178	0.00874	+0.0092	0.00040
	10	+0.3280	0.00788	+0.8615	0.00860	+0.0552	0.00028
	20	+0.0557	0.00652	+0.0405	0.00852	+0.0176	0.00030
	40	+0.0478	0.00631	+0.1343	0.00823	+0.0191	0.00035

Table 13: Same ablations as Table 12, at H=32.

Ablation	Value	CSI300		CSI500		Crypto	
		RankIC	σ^2-MAE	RankIC	σ^2-MAE	RankIC	σ^2-MAE
Lags K	**1**	+0.3596	0.01501	+0.8522	0.01882	+0.0482	0.00071
	2	+0.3162	0.01494	+0.8280	0.01955	+0.0445	0.00066
	4	+0.2739	0.01441	+0.8473	0.01813	+0.0491	0.00065
	8	+0.2931	0.01414	+0.8683	0.02492	+0.0569	0.00069
Lookback W	80	+0.3105	0.01427	+0.8272	0.02659	+0.0413	0.00058
	160	+0.3596	0.01454	+0.8522	0.01858	+0.0482	0.00071
	320	+0.2761	0.01604	+0.8607	0.01942	+0.0493	0.00070
Weight λ	5	+0.3435	0.01489	+0.8345	0.01784	+0.0450	0.00065
	10	+0.3596	0.01501	+0.8522	0.01882	+0.0482	0.00071
	20	+0.2632	0.01378	+0.8719	0.01877	+0.0569	0.00065
	40	+0.0520	0.01364	+0.1267	0.02624	+0.0189	0.00066

Table 14: Relative weight drift $\|\theta - \theta_0\|_2 / \|\theta_0\|_2$ from the frozen model, whole-model (*all*) weights, in units of 10^{-2}, mean over three seeds. Ratio is VCA/CE. The 10% rows (shaded) are the regime-matched slice.

		H=16			H=32		
%data	Dataset	CE	VCA	Ratio	CE	VCA	Ratio
100%	CSI300	2.34	3.38	1.4×	0.90	3.78	4.2×
	CSI500	4.70	5.77	1.2×	1.38	4.95	3.6×
	Crypto	5.08	6.95	1.4×	1.94	5.77	3.0×
10%	CSI300	0.49	0.85	1.7×	0.60	0.81	1.3×
	CSI500	1.18	1.29	1.1×	0.96	1.08	1.1×
	Crypto	1.57	1.68	1.1×	1.39	1.43	1.0×
1%	CSI300	0.54	0.42	0.8×	0.25	0.45	1.8×
	CSI500	0.71	0.38	0.5×	0.42	0.50	1.2×
	Crypto	0.37	0.55	1.5×	0.35	0.39	1.1×

Table 15: Raw per-cell results at $H=8$, mean±sd over 3 seeds. No quantity is normalized against the frozen model. σ^2-MAE columns are reported to 5 decimal places; all other columns to 4.

DS	Arm	price RankIC	price IC	ret. RankIC	ret. IC	ACF^2 gap	σ^2-MAE$_{\text{FORE}}$	σ^2-MAE$_{\text{VOL}}$
CSI300	Frozen	+0.2906	+0.0630	+0.0537	+0.0142	0.05812	0.00345	0.00342
	CE	+0.2539	+0.0044	-0.0092	-0.0100	0.05772	0.00322	0.00313
	MSE	+0.2418	-0.0096	+0.0031	-0.0025	0.05949	0.00363	0.00349
	VCA	+0.2731	+0.0331	+0.0115	-0.0067	0.05749	0.00310	0.00312
CSI500	Frozen	+0.8485	+0.0985	+0.0411	+0.0444	0.05537	0.00446	0.00448
	CE	+0.8196	-0.0787	+0.0168	+0.0185	0.05651	0.00419	0.00420
	MSE	+0.8148	-0.1105	-0.0416	-0.0551	0.05821	0.00469	0.00456
	VCA	+0.8437	+0.0569	+0.0122	+0.0030	0.05419	0.00472	0.00605
Crypto	Frozen	+0.0546	+0.0178	+0.0127	+0.0104	0.06489	0.00014	0.00012
	CE	+0.0480	+0.0099	+0.0089	+0.0127	0.06436	0.00013	0.00013
	MSE	+0.0316	+0.0057	+0.0120	+0.0140	0.06773	0.00014	0.00014
	VCA	+0.0564	+0.0195	+0.0148	+0.0130	0.06456	0.00013	0.00012

Table 16: Raw per-cell results at $H=16$, mean±sd over 3 seeds. No quantity is normalized against the frozen model. σ^2-MAE columns are reported to 5 decimal places; all other columns to 4.

DS	Arm	price RankIC	price IC	ret. RankIC	ret. IC	ACF^2 gap	σ^2-MAE$_{\text{FORE}}$	σ^2-MAE$_{\text{VOL}}$
CSI300	Frozen	+0.3167	+0.0817	+0.0240	-0.0304	0.03641	0.00662	0.00612
	CE	+0.3405	+0.1118	+0.0304	+0.0183	0.03517	0.00704	0.00618
	MSE	+0.3352	+0.1045	+0.0424	+0.0463	0.03663	0.00747	0.00708
	VCA	+0.3280	+0.0990	+0.0219	+0.0030	0.03523	0.00788	0.00812
CSI500	Frozen	+0.8529	+0.1210	+0.0642	+0.0314	0.03759	0.00901	0.00840
	CE	+0.8146	-0.1335	+0.0138	+0.0373	0.03804	0.00873	0.00807
	MSE	+0.8300	-0.0373	+0.0219	+0.0688	0.04019	0.00973	0.00932
	VCA	+0.8615	+0.1723	+0.0480	+0.0438	0.03730	0.00860	0.00819
Crypto	Frozen	+0.0514	+0.0139	+0.0147	+0.0053	0.04047	0.00031	0.00027
	CE	+0.0441	+0.0042	+0.0078	+0.0136	0.03958	0.00029	0.00026
	MSE	+0.0485	+0.0079	+0.0083	+0.0120	0.04275	0.00031	0.00030
	VCA	+0.0552	+0.0167	+0.0143	+0.0067	0.03885	0.00028	0.00035

Table 17: Raw per-cell results at $H=32$, mean±sd over 3 seeds. No quantity is normalized against the frozen model. σ^2-MAE columns are reported to 5 decimal places; all other columns to 4.

DS	Arm	price RankIC	price IC	ret. RankIC	ret. IC	ACF^2 gap	σ^2-MAE$_{\text{FORE}}$	σ^2-MAE$_{\text{VOL}}$
CSI300	Frozen	+0.3328	+0.1078	+0.0164	-0.0309	0.01848	0.01430	0.01260
	CE	+0.2936	+0.0496	+0.0380	+0.0517	0.01771	0.01600	0.01360
	MSE	+0.3111	+0.0759	+0.0163	+0.0262	0.01850	0.01600	0.01510
	VCA	+0.3596	+0.1472	+0.0595	+0.0507	0.01749	0.01500	0.01290
CSI500	Frozen	+0.8638	+0.1946	+0.0933	+0.0321	0.01815	0.01960	0.01710
	CE	+0.8583	+0.1606	+0.0029	-0.0089	0.01773	0.02080	0.01770
	MSE	+0.8349	-0.0004	-0.0273	+0.0147	0.01843	0.02060	0.01950
	VCA	+0.8522	+0.1285	+0.0132	-0.0120	0.01755	0.01880	0.01790
Crypto	Frozen	+0.0513	+0.0136	+0.0031	-0.0035	0.01965	0.00068	0.00058
	CE	+0.0253	-0.0125	-0.0125	-0.0100	0.01847	0.00064	0.00054
	MSE	+0.0499	+0.0108	+0.0100	+0.0018	0.02076	0.00073	0.00070
	VCA	+0.0482	+0.0091	-0.0024	-0.0031	0.01852	0.00072	0.00065

Table 18: Raw per-cell results at $H=48$, mean$\pm$sd over 3 seeds. No quantity is normalized against the frozen model. σ^2-MAE columns are reported to 5 decimal places; all other columns to 4.

DS	Arm	price RankIC	price IC	ret. RankIC	ret. IC	ACF^2 gap	σ^2-MAE$_{\text{FORE}}$	σ^2-MAE$_{\text{VOL}}$
CSI300	Frozen	+0.3329	+0.1110	-0.0155	-0.0548	0.01618	0.02290	0.01970
	CE	+0.3035	+0.0656	+0.0150	+0.0129	0.01466	0.02490	0.02160
	MSE	+0.3274	+0.1065	+0.0211	+0.0328	0.01489	0.02400	0.02220
	VCA	+0.3122	+0.0842	+0.0246	+0.0182	0.01417	0.02130	0.02020
CSI500	Frozen	+0.8800	+0.2900	+0.1163	+0.0429	0.01583	0.03110	0.02680
	CE	+0.8068	-0.0934	+0.0007	-0.0420	0.01478	0.03150	0.02740
	MSE	+0.8300	+0.0201	+0.0198	+0.0322	0.01541	0.03180	0.03010
	VCA	+0.8370	+0.0666	+0.0461	+0.0532	0.01452	0.03010	0.03300
Crypto	Frozen	+0.0440	+0.0008	-0.0088	-0.0159	0.01938	0.00105	0.00089
	CE	+0.0390	-0.0005	-0.0053	-0.0029	0.01692	0.00101	0.00085
	MSE	+0.0468	+0.0051	+0.0095	+0.0019	0.01872	0.00102	0.00096
	VCA	+0.0485	+0.0078	-0.0049	-0.0042	0.01754	0.00099	0.00097

Table 19: Raw per-cell results at $H=16$ (Chronos). σ^2-MAE columns are reported to 5 decimal places, 1 for the Chronos-VCA rows; all other columns to 4.

DS	Method	price RankIC	price IC	ret. RankIC	ret. IC	ACF^2 gap	σ^2-MAE$_{\text{FORE}}$	σ^2-MAE$_{\text{VOL}}$
CSI300	Frozen	+0.0181	+0.0189	+0.0221	+0.0159	0.03561	0.00745	0.00650
	CE	+0.0053	+0.0055	+0.0498	+0.0397	0.03556	0.00761	0.00670
	MSE	+0.0215	+0.0220	+0.0654	+0.0610	0.03551	0.00774	0.00703
	VCA	-0.0269	-0.0265	-0.0059	+0.0092	0.03369	123.0	189.0
CSI500	Frozen	+0.0537	+0.0589	+0.0451	+0.0206	0.03777	0.00994	0.00868
	CE	+0.0351	+0.0425	+0.0740	+0.0601	0.03736	0.01000	0.00883
	MSE	-0.0730	-0.0698	+0.0763	+0.0504	0.03709	0.01020	0.00932
	VCA	+0.0047	+0.0193	+0.0474	+0.0488	0.03691	26.4	47.2
Crypto	Frozen	-0.0017	-0.0023	-0.0145	-0.0165	0.03828	0.00027	0.00025
	CE	+0.0148	+0.0155	+0.0150	+0.0113	0.03827	0.00028	0.00024
	MSE	+0.0110	+0.0117	+0.0152	+0.0089	0.03824	0.00029	0.00024
	VCA	+0.0046	+0.0053	+0.0083	+0.0070	0.03814	0.00029	0.00025

Table 20: Raw per-cell results at $H=32$ (Chronos). σ^2-MAE columns are reported to 5 decimal places, 1 for the Chronos-VCA rows; all other columns to 4.

DS	Method	price RankIC	price IC	ret. RankIC	ret. IC	ACF^2 gap	σ^2-MAE$_{\text{FORE}}$	σ^2-MAE$_{\text{VOL}}$
CSI300	Frozen	+0.0042	+0.0040	+0.0223	+0.0200	0.01808	0.01630	0.01410
	CE	+0.0195	+0.0192	+0.0873	+0.0933	0.01779	0.01660	0.01500
	MSE	+0.0118	+0.0124	+0.0721	+0.0493	0.01763	0.01670	0.01530
	VCA	-0.0173	-0.0197	+0.0058	-0.0035	0.01719	21.7	51.7
CSI500	Frozen	+0.0534	+0.0458	+0.0990	+0.0622	0.01798	0.02180	0.01900
	CE	-0.0142	-0.0127	+0.0856	+0.0922	0.01768	0.02190	0.01970
	MSE	-0.0757	-0.0755	+0.0758	+0.0420	0.01759	0.02220	0.02060
	VCA	+0.1274	+0.1295	+0.0449	+0.0474	0.01766	0.02190	0.01930
Crypto	Frozen	+0.0049	+0.0051	-0.0160	-0.0212	0.01764	0.00060	0.00046
	CE	+0.0162	+0.0147	-0.0131	-0.0162	0.01745	0.00064	0.00046
	MSE	-0.0036	-0.0048	-0.0024	-0.0045	0.01752	0.00063	0.00047
	VCA	+0.0011	-0.0012	+0.0027	-0.0021	0.01728	15.0	93.4