FOMO: FORECASTING WITH OTHER MARKETS' OBSERVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series foundation models (TSFMs) increasingly accept multivariate inputs, yet adding related series as context can make their forecasts worse, a problem we call **context degradation**. In financial markets where contracts often have several related ones, this effect can be severe and unpredictable. The same kind of context that degrades one model by up to 15.6% can improve another by 6.8%. We find that the outcome depends on where series interact inside the network, and that early cross-series attention moves the target's representation away from the state it has without context. We introduce FOMO (Forecasting with Other Markets' Observations), a training-free algorithm that supplies related market histories as context and restricts cross-series attention to the later layers of a TSFM. Our experiments on four TSFMs and five datasets spanning prediction markets, equities, and crypto show that FOMO never falls below target-only forecasting in any setting where it changes the model, whereas native context mixing significantly degrades forecasts in half of the held-out settings. FOMO also reduces the worst-decile loss relative to native mixing in every held-out setting, by a median of $5.4\times$.

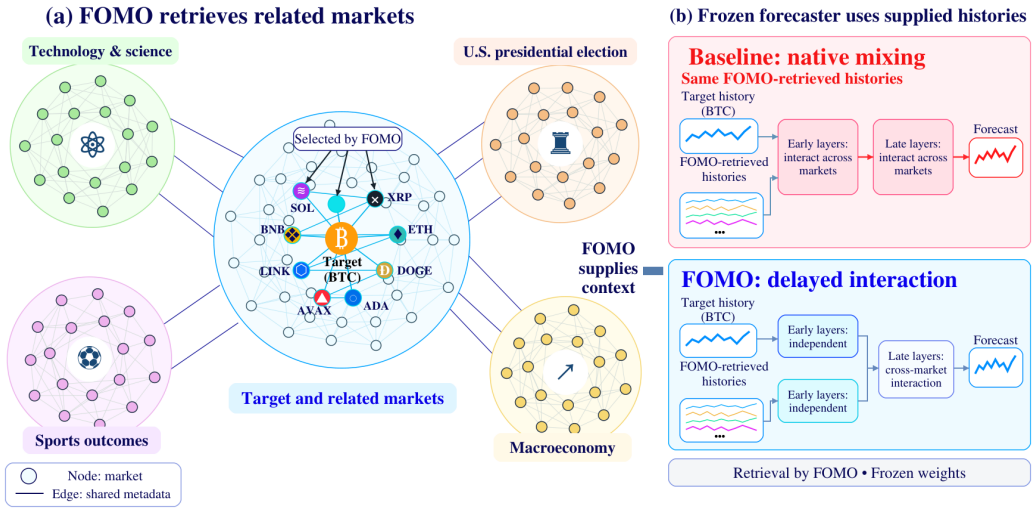

Figure 1: **FOMO overview.** **(A)** Related market histories are retrieved through shared metadata and passed with the target to a frozen forecaster, which processes each series independently in early layers and lets markets interact only in later layers. **(B)** Out-of-sample C_Q and relative gain over target-only forecasting on later-period Polymarket (Table 1).

1 INTRODUCTION

Time-series foundation models (TSFMs) have shown the ability to forecast unseen series zero-shot (Ansari et al., 2024; Das et al., 2024), and recent models increasingly accept multivariate inputs,

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.

allowing related series to serve as context for a target (Ansari et al., 2025; Jain & Sen, 2026; Khwaja et al., 2026; Woo et al., 2024). Financial markets are the obvious place to exploit this context, as prediction markets such as Polymarket and Kalshi list many contracts on the same event with different thresholds or deadlines (Wolfers & Zitzewitz, 2004; Arrow et al., 2008; Hanson, 2003; Manski, 2006), even equities move with their industry, and coins move with their sector. For example, a contract on whether Bitcoin exceeds $100,000 by December 31 and one on whether it exceeds $120,000 by December 31 likely share similar price movements.

In this paper, we explore the methods to use such related contracts to forecast a target's future price distribution (Koenker & Bassett, 1978). The naive way to do this is to supply related contracts as additional variates and let the model's cross-series attention operate throughout the network, which we refer to as *native mixing*. Under native mixing, the target's representation is exposed to other series from the first layer onward.

However, in native mixing, adding related series often makes forecasts *worse* than using the target alone. Our activation-patching experiments and representation analyses suggest that early cross-series interaction disrupts target representations (Section 5.4). This context degradation has been observed broadly in multivariate forecasting, where target-only models frequently outperform multivariate ones (Han et al., 2024; Nie et al., 2023; Zeng et al., 2023), and several foundation models forgo cross-variate modeling entirely (Ansari et al., 2024; Das et al., 2024). We find that context degradation depends on how early or late cross-series interaction occurs in the network. Keeping the retrieved histories and model weights fixed, the same context that degrades Moirai-2 by 15.6% under native mixing improves it by 2.2% when cross-market interaction is restricted to the later half of the network. The damage is also unpredictable: across models and markets, native mixing sometimes helps and sometimes significantly degrades forecasts, so it is usually unsafe to use native mixing.

To address this, we introduce FOMO (Forecasting with Other Markets' Observations), a training-free algorithm that constructs neighborhoods of related markets from shared contract metadata and moves cross-series attention to the later half of a model's network. FOMO requires no parameter updates, and because it constrains only where series interact, it applies to any retrieved context. Retrieval serves mainly to supply realistic context without look-ahead: under FOMO, randomly drawn admissible neighbors help about as much as metadata neighbors, so what makes context safe is where it enters the network rather than which series are chosen.

We evaluate FOMO on four TSFMs (Toto-4M, Moirai-2, Chronos-2, TimesFM-3) across five datasets: three from Polymarket and Kalshi, and two beyond prediction markets, S&P 500 equities and crypto. Our contributions are:

- We show that context degradation in multivariate TSFMs depends on *where* cross-series attention acts: early mixing degrades forecasts, while the same context mixed late helps (Section 5.3).
- We introduce FOMO, a training-free fix that never falls below target-only forecasting in any setting where it changes the model, and shrinks the worst-decile loss relative to native mixing in every held-out setting, by a median of $5.4\times$ (Section 5.2).
- We provide evidence for the cause. Early mixing displaces the target's representation 1.7–2.7 times as far as late mixing in every held-out domain, and restoring the target's mid-network state recovers 64–82% of the error gap between native mixing and FOMO (Section 5.4).

2 Related Work

Channel-independent forecasters often match or outperform channel-dependent ones (Nie et al., 2023; Zeng et al., 2023; Han et al., 2024), while iTransformer attends across variates (Liu et al., 2024b) and graph models learn or fix cross-series adjacency (Li et al., 2018; Wu et al., 2019; 2020). Foundation models have moved from univariate forecasting (Ansari et al., 2024; Das et al., 2024) to multivariate inputs with cross-series attention (Woo et al., 2024; Ansari et al., 2025; Jain & Sen, 2026; Khwaja et al., 2026). FOMO leaves these models' weights unchanged and controls *where* their existing cross-series attention operates. Retrieval augmentation supplies external evidence to a frozen model (Lewis et al., 2020), and recent work retrieves similar series for time-series foundation models (Tire et al., 2026; Zhang et al., 2025). That work focuses on *what* to retrieve; we show that *where* retrieved series interact can decide whether they help, echoing findings that language models are distracted by

poorly placed context (Shi et al., 2023; Liu et al., 2024a). Our interventions follow causal mediation and activation patching (Vig et al., 2020; Geiger et al., 2021; Meng et al., 2022); Appendix A extends this discussion.

3 PRELIMINARIES

3.1 PROBABILISTIC FORECASTING WITH FROZEN MODELS

Prediction markets trade contracts whose payoffs depend on the outcome of future events (Wolfers & Zitzewitz, 2004). Let $i \in \mathcal{V}$ index a contract with metadata m_i and observed prices in $[0, 1]$. We forecast a contract's future *price* (Fama, 1970; Manski, 2006).

At forecast origin t, the target's history is

$$\boldsymbol{x}_i(t) = [x_i(t - (T-1)\Delta), \ldots, x_i(t)] \in [0,1]^T, \tag{1}$$

where Δ is the sampling interval and each $x_i(u)$ is the last observation at or before u. We use $T = 256$ hourly values and forecast the next $H = 24$ hours.

A pretrained forecaster f_θ comprising L layers takes the observed history as input and predicts quantiles of future values.

$$\hat{\boldsymbol{q}}_i(t) = f_\theta(\boldsymbol{x}_i(t)) \in \mathbb{R}^{H \times |\mathcal{Q}|}, \qquad \mathcal{Q} = \{0.1, 0.2, \ldots, 0.9\}, \tag{2}$$

where $\hat{q}_{i,h,\alpha}(t)$ estimates the α-quantile of the price at horizon h. Forecasts are scored with the pinball loss $\rho_\alpha(e) = e\,(\alpha - \mathbb{1}\{e < 0\})$ (Koenker & Bassett, 1978), averaged over horizons and quantile levels:

$$C_Q(i,t) = \frac{2}{H|\mathcal{Q}|} \sum_{h=1}^{H} \sum_{\alpha \in \mathcal{Q}} \rho_\alpha(y_{i,h}(t) - \hat{q}_{i,h,\alpha}(t)), \tag{3}$$

where $y_{i,h}(t)$ is the realized price. This is the finite-grid approximation of the CRPS (Gneiting & Raftery, 2007; Bracher et al., 2021), and lower is better.

3.2 CONTEXT-AUGMENTED FORECASTING

Context-augmented forecasting supplies related histories alongside the target's own. This involves deciding *which* contracts to include in the context and *how* their context is allowed to interact within the model.

Context selection. A retrieval policy π selects an ordered neighborhood $\mathcal{N}_i^\pi(t)$ containing up to K other contracts, using only information available at time t. Their histories form a target-first panel

$$\boldsymbol{X}_i^\pi(t) = [\boldsymbol{x}_i(t);\, \boldsymbol{x}_{j_1}(t);\, \ldots;\, \boldsymbol{x}_{j_k}(t)]^\top \in \mathbb{R}^{(k+1)\times T}, \qquad k = |\mathcal{N}_i^\pi(t)| \le K, \tag{4}$$

on a shared temporal grid. We use $K = 7$. The model receives only price histories: metadata is used to determine which contracts are retrieved, but is not provided as input to f_θ.

Context integration. Multivariate foundation models differ in how cross-series information is integrated across layers. Toto-4M permits cross-series attention (Vaswani et al., 2017) only in its final block, whereas Moirai-2, Chronos-2, and TimesFM-3 permit it at every layer (Khwaja et al., 2026; Ansari et al., 2025; Jain & Sen, 2026; Liu et al., 2025). Let $\mathcal{S} \subseteq \{1, \ldots, L\}$ denote the set of layers in which information may be exchanged across series, and let $f_\theta^{\mathcal{S}}$ denote the corresponding model with cross-series attention restricted to those layers. Processing within each individual series remains unchanged at all layers. The model's own configuration is $\mathcal{S}_{\text{native}}$, and *native mixing* forecasts the target as

$$\hat{\boldsymbol{q}}_i^{\pi,\text{native}}(t) = \left[f_\theta^{\mathcal{S}_{\text{native}}}(\boldsymbol{X}_i^\pi(t)) \right]_{\text{target}}. \tag{5}$$

With $\mathcal{N}_i^\pi(t) = \emptyset$ this reduces to target-only forecasting, $\hat{\boldsymbol{q}}_i^{\varnothing}(t) = f_\theta(\boldsymbol{x}_i(t))$.

Context degradation. Adding context is not guaranteed to help. Define the context effect of a policy π under interaction set $\mathcal{S}$ as

$$\delta^{\pi,\mathcal{S}}(i,t) = C_Q\left(\hat{\boldsymbol{q}}_i^{\pi,\mathcal{S}}(t)\right) - C_Q\left(\hat{\boldsymbol{q}}_i^{\varnothing}(t)\right). \tag{6}$$

Context *degrades* the forecaster when $\delta > 0$ on average, meaning that the augmented model performs worse than the same model given only the target history. This degradation has been observed in multivariate forecasting settings (Han et al., 2024; Zeng et al., 2023). The related market retrieval policy π determines which contracts are provided to the model, and $\mathcal{S}$ determines where that information may interact across the network. We notice that even with π and θ fixed, the sign of δ can change with $\mathcal{S}$. Hence, FOMO uses a metadata-based retrieval policy π together with a late-layer interaction set $\mathcal{S}$, as described in Section 4. Degradation is uneven across targets, so we also report two risk measures. *Harmed* is the share of targets whose mean C_Q is more than 0.1% worse than under target-only forecasting. *Worst-10%* is the total excess error of the worst 10% of targets, expressed as a percentage of the target-only mean.

4 METHOD

FOMO has two parts. A retrieval step picks the related contracts (Section 4.1), and a masking step controls where in the network they can interact with the target (Section 4.2). Neither touches the model's weights.

4.1 METADATA-GUIDED CONTEXT SELECTION

We describe each contract i by a record m_i (platform, category, area, entity, event family, threshold, expiry date) and the set of tokens A_i in its title. From these records we build one graph of related markets ahead of time and look up neighbors in it at forecast time. Equities and crypto have no contract records, and there we use sector and sub-industry labels instead (Appendix H).

Candidate generation. Two contracts become candidates for each other when they share a field value, for example the same entity, the same threshold ladder, the same event, or the same expiry date. Most contracts share values with many others. We rank candidates by the groups they share, giving more weight to groups that point to the same event, and keep the top 512 (Appendix P.2).

Pairwise ranking. We score each candidate j with

$$\begin{aligned} w(i,j) = {} & 0.25\mathbb{1}_{\text{platform}} + 1.5\mathbb{1}_{\text{area}} + 0.75\mathbb{1}_{\text{expiry}} + 2.5\mathbb{1}_{\text{family}} \\ & + 4\,J(A_i,A_j)\mathbb{1}\{J(A_i,A_j) \geq 0.2\} + w_{\text{card}}(i,j). \end{aligned} \tag{7}$$

Here J is the Jaccard overlap of the two token sets (Jaccard, 1901), and w_{card} collects relations parsed from the contract text, such as a shared entity, two rungs of one threshold ladder, or two outcomes of one event (weights in Appendix P.2). Every contract stores its 16 best-scoring candidates.

No look-ahead. When ranking candidates for target i, we only consider contracts first observed no later than i itself. The filter runs before either cap, which means a contract listed after the target cannot affect which candidates the target keeps.

Assembly at forecast time. At origin t we walk down the stored list and take the first $K = 7$ contracts with usable histories. Each neighbor value $x_j(u)$ is its last observation at or before u, gaps with no earlier observation are masked, and all histories end at t. The result is the panel $\boldsymbol{X}_i^{\pi_{\text{meta}}}(t)$ from Section 3.2. If none of the stored candidates is usable, the target is forecast alone.

4.2 LATE-LAYER CONTEXT ALLOCATION

Let $\mathcal{S}_{\text{native}} \subseteq \{1,\dots,L\}$ be the layers in which the pretrained model lets series attend to each other. We keep the ones in the second half of the network,

$$\mathcal{S}_{\text{late}} = \{\ell \in \mathcal{S}_{\text{native}} : \ell > \lceil L/2 \rceil\}, \tag{8}$$

and in every other layer of $\mathcal{S}_{\text{native}}$ we add a mask to the cross-series attention logits before the softmax,

$$M_{ab}^{(\ell)} = \begin{cases} 0 & a = b, \\ -\infty & a \neq b, \end{cases} \tag{9}$$

where a and b index series in the panel. The FOMO forecast is $\left[f_\theta^{\mathcal{S}_{\text{late}}}(\boldsymbol{X}_i^{\pi_{\text{meta}}}(t))\right]_{\text{target}}$. With the mask in place a series can only attend to itself, so up to layer $\lceil L/2 \rceil$ each market is processed from its own history. The target's hidden state at that layer is the same as when it is forecast alone. Related markets start to interact in the next layer.

For our models, this removes cross-series attention from the first 3 of 6 layers in Moirai-2, 6 of 12 in Chronos-2, and 10 of 20 in TimesFM-3. Toto-4M only mixes series in its last block (layer 4 of 4), so FOMO leaves it unchanged and its forecast equals native mixing. The allocations we compare in Section 5 all use the same retrieved panel and differ only in $\mathcal{S}$. Any difference in their accuracy therefore comes from where context enters the network.

5 Experiments

We start with accuracy and safety across models and markets (RQ1). We then vary where interaction happens while holding the context fixed (RQ2), look inside the models to see what early interaction does to the target (RQ3), and finally give FOMO context that carries no useful information to find where it breaks (RQ4).

5.1 Experimental Setup

Base Models and Baselines. We evaluate four foundation models with cross-series attention: Toto-4M (Khwaja et al., 2026), Moirai-2 (Liu et al., 2025), Chronos-2 (Ansari et al., 2025), and TimesFM-3 (Jain & Sen, 2026). Moirai-2, Chronos-2, and TimesFM-3 have L=6, 12, and 20 layers respectively; for Toto-4M native mixing and FOMO coincide (Section 4.2). Architecture details are in Appendix B.

Datasets and Evaluation Metrics. Prediction-market data are hourly contract prices (T=256, H=24), retrieved with the graph of Section 4.1 (K=7): (1) a Polymarket *development dataset* (977 windows); (2) a Polymarket *later-period dataset* (2,000 windows, 100 targets); and (3) a *Kalshi dataset* (555 windows, 179 targets). Beyond prediction markets, (4) *equities* use daily prices of 100 S&P 500 stocks (2,000 monthly windows, 2013–2015) and (5) *crypto* uses daily prices of 41 coins (820 windows, 2025–2026), both with T=256 days and H=24 days, each series scaled by its price at the forecast origin (Appendix H). The primary metric is C_Q (Eq. 3), with the harmed and worst-10% risk measures of Section 3.2. We report paired 95% bootstrap intervals clustered over targets (Efron & Tibshirani, 1993; Cameron et al., 2008).

5.2 Safety and Accuracy Across Markets (RQ1)

To evaluate forecasting performance and safety, we compare target-only forecasting, native mixing, and FOMO on all five datasets (Table 1; raw scores in Appendices F and H). We draw the following observations:

Obs 1: FOMO never falls below target-only forecasting. For Moirai-2, Chronos-2, and TimesFM-3, whose cross-series attention FOMO changes, FOMO improves on target-only inputs in all 15 model–dataset settings, and Toto-4M improves in four of five. On equities every improvement is significant, including for models that native mixing degrades. Every other use of the same context we test falls below target-only in at least one setting. Native mixing does so in 10 of 15 settings, early-half mixing in 11, forecast averaging in 7, recalibration in 3, and a development-tuned weighted average in 2 (Table 1; Appendix J).

Obs 2: Native mixing is unpredictable, and FOMO is safe. Native mixing significantly degrades six of the twelve held-out settings, e.g., Moirai-2 on Kalshi (-15.3%) and all three models on equities, yet helps elsewhere. FOMO lowers the worst-10% loss in all twelve, by a median of $5.4\times$ (Table 3; e.g., 15.0% $\rightarrow$ 0.9% for TimesFM-3 on equities). Where early correlations happen to help,

Table 1: Gain over target-only forecasting (%; higher is better) on all five datasets. *Early half* allows cross-series attention in as many layers as FOMO, but in the first half of the network. *Weighted avg.*: $(1-w)$ target-only $+\, w$ native-mixing quantiles. *Recalibration*: target-only quantiles rescaled around the median by one factor. Both w and the factor are chosen per model on the development set. Best per model and dataset in bold; ‡: paired, target-clustered 95% bootstrap interval excludes zero. Toto-4M mixes only in its last layer, so native mixing and FOMO coincide. Raw C_Q values are in Appendix F.

		Prediction markets			Other markets	
Model	Input	Development	Later-period	Kalshi	Equities	Crypto
Toto-4M	Native = FOMO	$+4.37^{\ddagger}$	$+4.41^{\ddagger}$	$+2.06$	$+0.95^{\ddagger}$	-0.05
Moirai-2	Native mixing	$-15.57^{\ddagger}$	-1.93	$-15.33^{\ddagger}$	$-3.22^{\ddagger}$	$-3.55^{\ddagger}$
	Early half	$-4.95^{\ddagger}$	$+0.54$	$-4.72^{\ddagger}$	$-2.09^{\ddagger}$	$-2.67^{\ddagger}$
	Weighted avg.	$+0.07$	$+0.55^{\ddagger}$	-0.03	$+0.13^{\ddagger}$	$+0.07^{\ddagger}$
	Recalibration	0.00	0.00	0.00	0.00	0.00
	FOMO	$\mathbf{+2.25}^{\ddagger}$	$\mathbf{+3.99}^{\ddagger}$	$\mathbf{+1.69}$	$\mathbf{+0.84}^{\ddagger}$	$\mathbf{+0.85}$
Chronos-2	Native mixing	-1.07	$+1.86$	$\mathbf{+2.45}$	$-1.15^{\ddagger}$	$\mathbf{+3.11}^{\ddagger}$
	Early half	$-3.23^{\ddagger}$	$\mathbf{+1.96}$	-0.59	$-1.87^{\ddagger}$	$+1.54^{\ddagger}$
	Weighted avg.	$+0.61$	$+1.77^{\ddagger}$	$+2.16^{\ddagger}$	$+0.10$	$+1.88^{\ddagger}$
	Recalibration	$+0.16$	$+0.21^{\ddagger}$	$+0.26$	$-0.43^{\ddagger}$	$+0.01$
	FOMO	$\mathbf{+0.87}^{\ddagger}$	$+1.15^{\ddagger}$	$+0.83^{\ddagger}$	$\mathbf{+0.55}^{\ddagger}$	$+0.61^{\ddagger}$
TimesFM-3	Native mixing	-1.81	$\mathbf{+6.77}^{\ddagger}$	$-4.32^{\ddagger}$	$-2.16^{\ddagger}$	$\mathbf{+2.90}^{\ddagger}$
	Early half	$-3.53^{\ddagger}$	$+4.62^{\ddagger}$	$-6.05^{\ddagger}$	$-4.41^{\ddagger}$	$-1.99^{\ddagger}$
	Weighted avg.	$+0.14$	$+2.11^{\ddagger}$	-0.33	$+0.02$	$+1.42^{\ddagger}$
	Recalibration	$+0.08$	$+0.03$	$+0.15^{\ddagger}$	$-0.23^{\ddagger}$	$-0.08^{\ddagger}$
	FOMO	$\mathbf{+0.56}$	$+1.47^{\ddagger}$	$\mathbf{+0.78}$	$\mathbf{+0.60}^{\ddagger}$	$+1.42^{\ddagger}$

native mixing can score higher on average (Chronos-2 and TimesFM-3 on later-period Polymarket and crypto, Chronos-2 on Kalshi); FOMO trades this upside for protection against degradation. Recalibrating the target-only quantiles gains at most 0.27%, and FOMO exceeds it in all 15 settings (Appendix J). FOMO's 80% intervals stay within 0.98–1.08 times the target-only width in 11 of the 12 held-out settings (Table 20).

5.3 Impact of Interaction Depth (RQ2)

To evaluate how interaction depth influences forecasting accuracy, we hold the retrieved panel $\boldsymbol{X}_i^{\pi_{\text{meta}}}(t)$ and model weights fixed while varying $\mathcal{S}$ with Eq. 9 across Moirai-2, Chronos-2, and TimesFM-3. Beyond *native* and *late half* (FOMO), we test nested suffixes (*last only*, last 25%, last 75%), *early half* (first $\lceil L/2 \rceil$ layers), *interleaved* (equally spaced layers), single-layer placements (first, middle, or last layer only), and three random-depth schedules. Based on Figure 2, we draw the following observations:

Obs 3: Unrestricted early cross-market attention degrades forecasting performance. Native mixing (all layers active) degrades all three models relative to target-only inputs, by up to 15.6% on Moirai-2, and early-half mixing degrades them as well (-3.2% to -5.0%); interleaving, which also touches early layers, degrades TimesFM-3 by 4.7% here and on all four held-out datasets (Appendix K). The same holds beyond prediction markets: early-half mixing degrades all three models on equities (-1.9% to -4.4%; Table 1).

Obs 4: Confining cross-series attention to later layers converts degradation into positive gains. Restricting cross-market attention to the later half improves all three models (+2.2% on Moirai-2, +0.9% on Chronos-2, +0.6% on TimesFM-3). Because early-half and late-half schedules use the same number of interacting layers, interaction position, not count, governs degradation. On the development dataset, the late half beats the other suffix schedules we test (last only, last 25%, last 75%) and every single-layer placement for all three models (Table 6). We also sweep the boundary over every layer on all five datasets (Figure 3). Tail loss falls steeply over the first masked layers and

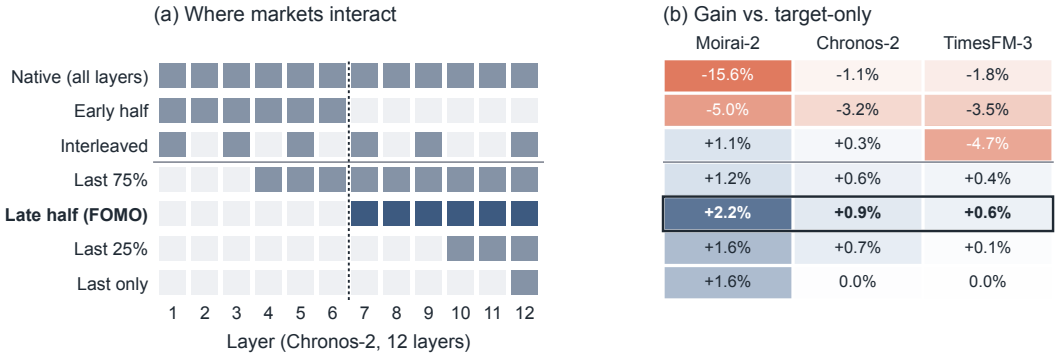

Figure 2: **Where cross-market interaction happens decides whether context helps. (a)** Layers with active cross-market attention under each schedule, shown for Chronos-2; the other models use the same relative depths. **(b)** Relative C_Q reduction versus target-only (development dataset, 977 windows; blue is better, orange is worse). Native and early-half mixing hurt all three models, and the late half (FOMO, outlined) scores best, or within 0.1% of the best schedule, for every model. For Moirai-2 (6 layers), last 25% equals last only. Full results are in Appendix C.

then flattens. Half depth captures a median 90% of the achievable tail reduction and keeps the gain positive in all 15 settings (Appendix I). Allowing interaction in the last layer only gains less than FOMO in all 15 (Appendix K). In Moirai-2, the target's volatility is as decodable from its target-only state at layer 3 of 6 as at the final layer ($R^2 = 0.45$ vs. 0.46; Table 9), consistent with interaction starting once the target is encoded.

5.4 Representation Analysis and Activation Interventions (RQ3)

We hypothesize that early mixing hurts because it alters the target's own representation before the model has finished encoding it, and we test this in three ways on Moirai-2, Chronos-2, and TimesFM-3.

Activation patching. We run each window twice on the same panel, once with native mixing and once with FOMO, and pause both runs at the midpoint layer $b = L/2$, where FOMO's masking ends (Vig et al., 2020; Meng et al., 2022; Zhang & Nanda, 2024). Up to this layer the FOMO target has not seen any other series, so its hidden state is exactly its target-only state. We then swap the target's hidden state between the two runs, and let both runs finish. In the *rescue*, the native run continues from FOMO's target state. If early mixing hurts by corrupting the target, this swap should recover most of the error gap between native mixing and FOMO. In the *reverse* swap, the FOMO run continues from the native target state and should lose accuracy. As a control, we replace the target's state with Gaussian noise of the same norm, which shows whether any change to the state would have the same effect.

Representation drift and probes. We measure how far context moves the target's representation as the standardized distance between its final-layer state under each allocation and under target-only processing, on 600 windows per model and domain. To test whether the target's own information survives mixing, we train ridge probes on the target's hidden states at each layer to predict its volatility over the previous 72 hours (Alain & Bengio, 2016; Belinkov, 2022); details are in Appendix E. From these experiments (Table 2; Appendix D), we draw the following observations:

Obs 5: Intermediate target representations largely carry the depth allocation effect. The rescue recovers 64%, 73%, and 82% of the error gap between native mixing and FOMO in Moirai-2, TimesFM-3, and Chronos-2, and the reverse swap brings back 23%, 51%, and 77% of it (Table 8). Both swaps are significant in Moirai-2, where native mixing hurts most, and the rescue in TimesFM-3.

Obs 6: Early mixing displaces the target's representation in every domain. Under native mixing the target's final-layer state moves 1.7–2.7 times as far from its target-only state as under FOMO, for

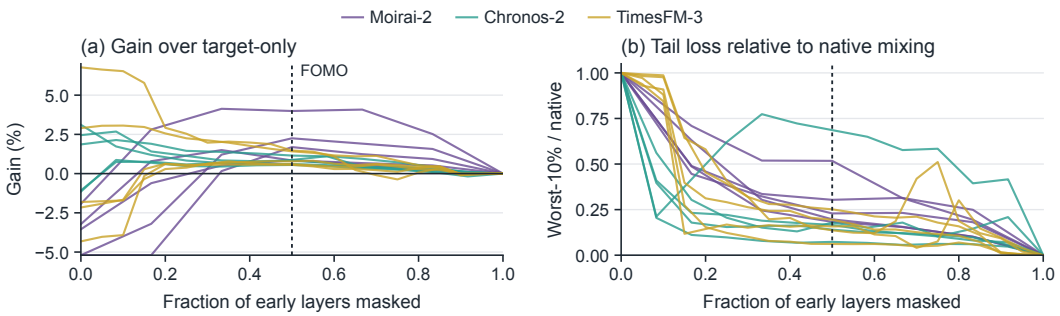

Figure 3: **Choosing where interaction begins.** (a) Gain over target-only and (b) worst-10% loss relative to native mixing as more early layers are masked, for every model (color) and dataset (line). FOMO (dashed) masks half.

every model on Kalshi, equities, and crypto (Table 2; all paired intervals exclude zero). On Kalshi, Moirai-2 moves 0.83 under native mixing and 0.44 under FOMO, alongside its largest held-out degradation (-15.3%). Early-half mixing uses as many interacting layers as FOMO and displaces the target further in all nine cells, so the displacement depends on where interaction happens.

Obs 7: Late-layer allocation preserves the linear decodability of target volatility. In Moirai-2, trailing volatility decodability under late routing remains high ($R^2 \approx 0.39$–0.45) from layer 2 onward, tracking target-only states; under native mixing it drops to 0.18–0.23. The decodability gap tracks forecast degradation across architectures (smaller gaps in TimesFM-3 and Chronos-2), providing descriptive support that early cross-attention alters target dynamics before later layers can process them (Figure 7, Appendix E).

Table 2: Representation drift and robustness to uninformative context, as *native / FOMO* (drift also reports early-half mixing; best in bold). Drift: standardized distance of the target's final-layer state from its target-only state (555 windows per cell on Kalshi and 600 elsewhere); every native−FOMO paired 95% interval excludes zero. Remaining columns: gain over target-only (%) with retrieved neighbors, or with neighbors replaced by assets drawn from the same admissible pool (Random), or synthetic random walks matching each neighbor's start value and step volatility (Walks). Random and Walks are means of five draws.

		Drift ↓	Gain over target-only (%) ↑		
Domain	Model		Retrieved	Random	Walks
Prediction markets (Kalshi)	Moirai-2	0.83 / 0.75 / **0.44**	−15.3 / **+1.7**	−13.9 / **+3.1**	−18.0 / **+1.4**
	Chronos-2	0.37 / 0.30 / **0.20**	**+2.5** / +0.8	+0.1 / **+0.5**	**+3.8** / +0.7
	TimesFM-3	0.82 / 0.76 / **0.38**	−4.3 / **+0.8**	−5.1 / **+0.7**	−5.1 / **+0.3**
Equities	Moirai-2	0.97 / 0.92 / **0.54**	−3.2 / **+0.8**	−3.8 / **+1.2**	−21.6 / **−2.3**
	Chronos-2	0.88 / 0.73 / **0.34**	−1.1 / **+0.6**	−1.0 / **+0.7**	−4.6 / **+0.5**
	TimesFM-3	1.18 / 1.09 / **0.61**	−2.2 / **+0.6**	−1.9 / **+0.5**	−0.3 / **+0.3**
Crypto	Moirai-2	0.85 / 0.80 / **0.46**	−3.5 / **+0.8**	−4.1 / **+1.0**	−33.1 / **−6.3**
	Chronos-2	0.71 / 0.51 / **0.26**	**+3.1** / +0.6	**+3.3** / +0.6	**+1.5** / +0.3
	TimesFM-3	1.34 / 1.08 / **0.78**	**+2.9** / +1.4	**+2.8** / +1.0	**+2.6** / +0.7

5.5 Robustness to Uninformative Context (RQ4)

A safe method should not be hurt when the supplied context carries no useful information. Holding the target fixed, we replace the retrieved neighbors with random assets from the same admissible pool or with synthetic random walks matched to each neighbor's start value and volatility, under both native mixing and FOMO (Table 2; development-set arms, including neighbors frozen at their last value, in Appendix L).

Obs 8: FOMO is safe with realistic but uninformative context; native mixing is not. Random neighbors help FOMO about as much as retrieved ones in every domain (e.g., $+3.1\%$ vs. $+1.7\%$ for Moirai-2 on Kalshi), so FOMO's benefit comes from safely using realistic context rather than from uniquely informative neighbors. The same random neighbors degrade native Moirai-2 by 13.9% and native TimesFM-3 by 5.1% on Kalshi, and synthetic random walks degrade them by 18.0% and 5.1% while FOMO stays positive on average. On equities and crypto, walks also push Moirai-2 below target-only under FOMO, although far less than under native mixing (-2.3% vs. -21.6% and -6.3% vs. -33.1%). One failure mode remains: neighbors frozen at a constant value break Moirai-2 under FOMO (Appendix L), so FOMO assumes context with realistic temporal dynamics.

Cost. FOMO adds a graph lookup and a static attention mask; on an NVIDIA H100 with seven neighbors its single-request p50 latency overhead over target-only forecasting is 0.2–7.4 ms across the four models (Appendix M).

Table 3: Per-target risk on the twelve held-out settings, as *native mixing* $\rightarrow$ *FOMO* (%; lower is better). Worst-10%: excess error of the worst decile of targets, as a share of the target-only mean. Harmed: share of targets made more than 0.1% worse than target-only. FOMO lowers the worst-10% loss in all twelve settings and the harmed share in ten.

	Model	Later-period	Kalshi	Equities	Crypto
Worst-10%	Moirai-2	31.3 $\rightarrow$ **16.2**	115.7 $\rightarrow$ **22.8**	20.3 $\rightarrow$ **6.2**	20.9 $\rightarrow$ **4.8**
	Chronos-2	13.9 $\rightarrow$ **2.5**	40.6 $\rightarrow$ **5.7**	12.6 $\rightarrow$ **0.9**	1.4 $\rightarrow$ **0.9**
	TimesFM-3	12.2 $\rightarrow$ **1.7**	45.7 $\rightarrow$ **8.9**	15.0 $\rightarrow$ **0.9**	5.7 $\rightarrow$ **0.9**
Harmed	Moirai-2	70 $\rightarrow$ **41**	76 $\rightarrow$ **54**	62 $\rightarrow$ **41**	66 $\rightarrow$ **32**
	Chronos-2	38 $\rightarrow$ **29**	39 $\rightarrow$ **35**	53 $\rightarrow$ **30**	17 $\rightarrow$ 22
	TimesFM-3	23 $\rightarrow$ 26	60 $\rightarrow$ **42**	65 $\rightarrow$ **22**	27 $\rightarrow$ **7**

6 Discussion, Limitations, and Conclusion

FOMO separates what context a frozen model receives from where that context can act, and the second choice decides whether context is safe. Delaying interaction trades some upside for safety: where early correlations happen to help, native mixing can score higher on average (Chronos-2 and TimesFM-3 on later-period Polymarket and crypto, Chronos-2 on Kalshi), and masking fewer early layers recovers part of this upside, at a higher worst-case loss in three of these five settings (Appendix I). On Polymarket, where prices are often flat, a residual-quantile baseline is competitive with every foundation model (Appendix B); on Kalshi, equities, and crypto, all foundation models beat a persistence forecast by 9–30%. On electricity load without shared metadata, delaying mixing reduces but does not remove Moirai-2's degradation (Appendix H). Equity sectors use current S&P 500 membership (survivorship bias), crypto sectors are author-assigned, and undisclosed pretraining data prevent ruling out overlap. We evaluate one checkpoint per model family. Future work could select the masking boundary per model, test larger checkpoints, and extend late interaction to other domains with related series. In this work, we showed that context degradation in multivariate time-series foundation models depends on where cross-series attention acts, and introduced FOMO, a training-free method that supplies related market histories while restricting their interaction to later layers. Across four foundation models and five datasets spanning prediction markets, equities, and crypto, FOMO never falls below target-only forecasting in any setting where it changes the model and reduces the worst-decile loss in every held-out setting, while native mixing degrades forecasts unpredictably. Representation analysis and activation patching suggest why. Early mixing displaces the target's own representation, and restoring it recovers 64–82% of the error gap between native mixing and FOMO.

Reproducibility Statement

All algorithms, candidate retrieval functions, attention-masking equations, and activation-patching protocols are fully specified in Section 4 and Section 5. Model checkpoint identifiers, layer configurations, and checkpoint hashes are recorded in Appendix N. Deterministic replay scripts, self-patch identity assertions, and artifact verification fixtures reproducing reported results are archived in the

accompanying code submission, together with scripts that build the equity and crypto benchmarks from public price data and compute every risk measure.

Ethics Statement

This research uses publicly available, aggregated time-series price histories from public blockchain (Polymarket) and regulated (Kalshi) prediction venues. No personally identifiable information, user profiles, or private transaction records were collected or processed. Our work investigates probabilistic forecasting on public foundation models and does not involve human subjects, automated trading execution, or market manipulation.

AI Use Statement

In this work, we used generative AI tools for assisting in the implementation of attention-masking and activation-patching experiment scripts, reformatting evaluation metric summaries, formatting LaTeX tables, preparing figure plotting scripts, and generating the illustrative market-graph schematic. Quantitative figures are rendered directly from saved results or explicitly identified manuscript table values. We have not used generative AI tools for developing theoretical proofs or formulating scientific hypotheses, and tasks involving human survey analysis or translation are not applicable to this work. All AI-assisted code was tested with deterministic assertions and numerical checks, and all numerical values in tables were verified directly against saved experiment artifacts. The authors take full responsibility for the entire content of this work, including all claims, analyses, and results.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

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. *Transactions on Machine Learning Research*, 2024. arXiv:2403.07815.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Kenneth J. Arrow, Robert Forsythe, Michael Gorham, Robert Hahn, Robin Hanson, John O. Ledyard, et al. The promise of prediction markets. *Science*, 320(5878):877–878, 2008. doi: 10.1126/science.1157679.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Johannes Bracher, Evan L. Ray, Tilmann Gneiting, and Nicholas G. Reich. Evaluating epidemic forecasts in an interval format. *PLOS Computational Biology*, 17(2):e1008618, 2021.

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics*, 90(3):414–427, 2008.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 10148–10167, 2024. arXiv:2310.10688.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.

Eugene F. Fama. Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2):383–417, 1970.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 9574–9586, 2021.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):7129–7142, 2024. doi: 10.1109/TKDE.2024.3400008.

Robin Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1): 107–119, 2003.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019. doi: 10.18653/v1/D19-1275.

Paul Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:241–272, 1901.

Ayush Jain and Rajat Sen. TimesFM-3: A zero-shot foundation model for multivariate forecasting. Google Research Blog, aug 2026. URL https://research.google/blog/timesfm-3-a-zero-shot-foundation-model-for-multivariate-forecasting/.

Emaad Khwaja, Chris Lettieri, Gerald Woo, Eden Belouadah, Marc Cenac, Guillaume Jarry, Enguerrand Paquin, Xunyi Zhao, Viktoriya Zhukov, Othmane Abou-Amal, Chenghao Liu, Ameet Talwalkar, and David Asker. Toto 2.0: Time series forecasting enters the scaling era. *arXiv preprint arXiv:2605.20119*, 2026. URL https://arxiv.org/abs/2605.20119.

Roger Koenker and Gilbert Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024a.

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 (ICLR)*, 2024b.

Charles F. Manski. Interpreting the predictions of prediction markets. *Economics Letters*, 91(3): 425–429, 2006. doi: 10.1016/j.econlet.2006.01.004.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning (ICML)*, 2023.

Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Retrieval augmented time series forecasting. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 300 of *PMLR*, pp. 4294–4302, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Justin Wolfers and Eric Zitzewitz. Prediction markets. *Journal of Economic Perspectives*, 18(2): 107–126, 2004. doi: 10.1257/0895330041371321.

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 (ICML)*, 2024. arXiv:2402.02592.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for deep spatial-temporal graph modeling. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2020.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence*, 2023.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations (ICLR)*, 2024.

Huanyu Zhang, Chang Xu, Yi-Fan Zhang, Zhang Zhang, Liang Wang, and Jiang Bian. TimeRAF: Retrieval-augmented foundation model for zero-shot time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 37(9):5654–5665, 2025. doi: 10.1109/TKDE.2025.3579137.

A Extended Related Work

Temporal representations and cross-series interaction. PatchTST splits each series into patches and runs every series through the same Transformer separately, with no attention between series (Nie et al., 2023). iTransformer takes the opposite approach: each variate becomes a single token, and attention runs only across variates (Liu et al., 2024b). Together, these models show that temporal modeling and cross-series interaction can be handled separately. FOMO keeps both inside one pretrained model and controls the depth at which series begin to interact.

Graph-based and multivariate forecasting. Early graph models for time series, such as DCRNN (Li et al., 2018), use a fixed directed graph between series. Graph WaveNet (Wu et al., 2019) and MTGNN (Wu et al., 2020) learn the adjacency matrix from data. FOMO instead builds each target's neighbors from contract metadata at forecast time, with no training. Its main role is to limit candidates to markets observable before the forecast origin; within that pool, random neighbors work about as well as metadata neighbors (Section 5.5).

Time-series foundation models. Early foundation models forecast one series at a time, whether by tokenizing values (Chronos; Ansari et al., 2024) or by decoder-only modeling of patches (TimesFM; Das et al., 2024). More recent models accept a variable number of related series, including Moirai (Woo et al., 2024; Liu et al., 2025), Chronos-2 (Ansari et al., 2025), TimesFM-3 (Jain & Sen, 2026), and Toto 2.0 (Khwaja et al., 2026). We find that passing market context through these models' native cross-series attention can reduce accuracy. FOMO uses the same attention layers but restricts them to the later part of the network, without changing any weights.

Retrieval augmentation and context sensitivity. Retrieval-augmented generation gives a frozen model extra evidence at inference time (Lewis et al., 2020), and recent forecasting work retrieves similar series for a time-series foundation model in the same way (Tire et al., 2026; Zhang et al., 2025). That work is mostly about *what* to retrieve. Our results suggest that, for models with cross-series attention, *where* the retrieved series enter the network can matter more. Language models show a similar sensitivity: irrelevant or badly placed context can hurt their answers (Shi et al., 2023; Liu et al., 2024a).

Interventions and probes. Our rescue and reverse runs follow causal mediation analysis and causal tracing (Vig et al., 2020; Geiger et al., 2021; Meng et al., 2022), and we report them following the recommendations of Zhang & Nanda (2024). We use linear probes (Alain & Bengio, 2016) only as descriptive evidence. As Hewitt & Liang (2019) and Belinkov (2022) caution, a probe's accuracy shows what can be linearly decoded from a representation, not what the model actually uses.

Probabilistic forecasting and scoring rules. Proper scoring rules favor evaluating the full predictive distribution rather than a point forecast (Gneiting & Raftery, 2007). We score nine quantiles with the pinball loss from quantile regression (Koenker & Bassett, 1978); averaged over the quantiles, this approximates the continuous ranked probability score (CRPS).

B Foundation Model Architectures and Baseline Details

The four models differ mainly in which layers let series attend to each other, and this determines what FOMO changes in each one.

Toto-4M (Toto 2.0). Toto-4M (Khwaja et al., 2026) has four Transformer layers. The first three attend only over time within each series, and series interact only in the last layer. Because layer 4 already lies in the later half of the network, Toto-4M satisfies FOMO's allocation without any masking ($\mathcal{S}_{\text{late}} = \mathcal{S}_{\text{native}} = \{4\}$), and native mixing and FOMO give identical forecasts. We include it as a model that mixes late by design. It gains from context on all three prediction-market datasets (Table 12), as FOMO's premise would predict.

Moirai-2. Moirai-2 (Liu et al., 2025) is a 6-layer patch-based Transformer in which series attend to each other in every layer ($\mathcal{S}_{\text{native}} = \{1, \ldots, 6\}$). It is the model most hurt by native mixing (-15.57% on the development dataset and -15.33% on Kalshi), and the one where early mixing moves the target's representation furthest (Section 5.4). Masking cross-series attention in layers 1–3 ($\mathcal{S}_{\text{late}} = \{4, 5, 6\}$) turns these losses into gains of $+2.25\%$, $+1.69\%$, and $+3.99\%$ on the development, Kalshi, and later-period Polymarket datasets.

Chronos-2. Chronos-2 (Ansari et al., 2025) is a 12-block encoder-only model. Each block combines attention over time with group attention across the series in the panel, so series interact at every depth ($\mathcal{S}_{\text{native}} = \{1, \ldots, 12\}$). Native mixing affects it less than Moirai-2: it loses 1.07% on the development dataset, gains 2.45% on Kalshi, and gains 1.86% on later-period Polymarket. With group attention restricted to blocks 7–12, FOMO improves on target-only on all three datasets ($+0.83\%$ to $+1.15\%$).

TimesFM-3. TimesFM-3 (Jain & Sen, 2026) is a 20-layer decoder-only model with a variate-attention module in every layer ($\mathcal{S}_{\text{native}} = \{1, \ldots, 20\}$). Its response to native mixing depends on the dataset: it loses 4.32% on Kalshi and 1.81% on the development dataset but gains 6.77% on later-period Polymarket. Restricting variate attention to layers 11–20 gives smaller but consistently positive gains ($+0.56\%$, $+0.78\%$, and $+1.47\%$), trading the large upside on one dataset for no losses on the others.

Statistical baselines. We also compare against four statistical baselines, each fit only on data available at the forecast origin:

- **Past residual quantiles:** empirical quantiles of the target's past price changes over the history window, held fixed across the 24-hour horizon.

- **Volatility-conditioned residual quantiles:** the same quantiles, rescaled by the ratio of trailing 72-hour volatility to volatility over the full history.
- **Linear quantiles (target features):** linear quantile regressions on the target's lagged hourly prices, one for each of the nine levels $\tau \in \{0.1, \ldots, 0.9\}$.
- **Linear quantiles (neighbor summary):** the same regressions with added summary features of the $K{=}7$ retrieved neighbors (mean price, trailing trend, and volatility).

Table 4 reports their scores, with gains relative to Toto-4M's target-only forecast as in Table 12. On later-period Polymarket, where prices are often flat, past residual quantiles are competitive with the foundation models; on Kalshi, every baseline is clearly worse.

Table 4: Statistical baselines under the quantile score ($C_Q \downarrow$; $\times 10^{-4}$ for later-period Polymarket, $\times 10^{-2}$ for Kalshi). Gains are relative to Toto-4M's target-only forecast. The linear regressions are reported for Kalshi only; on a separate later-period replication draw they score 61.8% and 85.3% worse than Toto-4M's target-only forecast.

	Later-period Poly. (2,000w)		Kalshi (555w)	
Baseline	C_Q	Gain (%)	C_Q	Gain (%)
Past residual quantiles	3.310	$+5.37\%$	1.274	-13.18%
Volatility-conditioned residuals	3.546	-1.39%	1.292	-14.82%
Linear quantiles (target features)	–	–	1.186	-5.41%
Linear quantiles (neighbor summary)	–	–	1.254	-11.42%

C Depth Allocation: Full Results

Table 5 adds the three random-depth schedules (seeds 7, 13, 29; each with the same number of cross-series layers as late half) and paired 95% bootstrap intervals behind Section 5.3 (evaluated with 10,000 clustered resamples). For Moirai-2 two of the three random draws coincide with the late-half set. Table 7 gives the full metric breakdown for all four models on the development dataset.

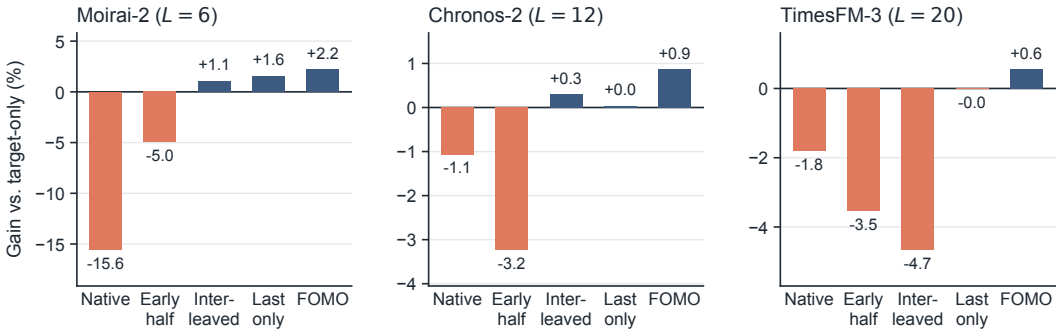

Figure 4: **Relative C_Q change versus target-only across allocation schedules** (development dataset, 977 windows), complementing Figure 2.

Table 5: Development dataset ($C_Q \downarrow$) for all depth schedules, with paired 95% intervals (target-clustered / day-clustered) for the late-half contrasts. Negative deltas favor late half.

	Moirai-2	Chronos-2	TimesFM-3
Target only	0.014375	0.014826	0.014368
Native	0.016613	0.014984	0.014628
Early half	0.015087	0.015299	0.014876
Interleaved	0.014219	0.014783	0.015038
Last only	0.014148	0.014819	0.014372
Late half (FOMO)	0.014052	0.014697	0.014288
Random depth, seed 7	0.014052	0.014734	0.014431
Random depth, seed 13	0.014052	0.014704	0.014490
Random depth, seed 29	0.014049	0.014786	0.014667
Late − native, target CI	$[-3.14, -1.99] \times 10^{-3}$	$[-6.39, +0.68] \times 10^{-4}$	$[-6.04, -0.82] \times 10^{-4}$
Late − native, day CI	$[-4.67, -1.79] \times 10^{-3}$	$[-11.8, +1.25] \times 10^{-4}$	$[-12.0, +0.67] \times 10^{-4}$
Late − target only, target CI	$[-5.77, -0.90] \times 10^{-4}$	$[-2.38, -0.29] \times 10^{-4}$	$[-1.89, +0.31] \times 10^{-4}$
Late − target only, day CI	$[-5.84, -0.93] \times 10^{-4}$	$[-3.50, -0.43] \times 10^{-4}$	$[-2.13, +1.13] \times 10^{-4}$

Table 6 compares schedules that each allow cross-series attention in exactly one layer, holding the count of interacting layers fixed at one.

Table 6: Single-layer placement on the development dataset (relative C_Q change vs. target-only; positive is better). Each single-layer arm has exactly one interacting layer. In Moirai-2, last-only outperforms first-only under both target- and day-clustered 95% intervals.

Schedule	Moirai-2	Chronos-2	TimesFM-3
First layer only	-0.08	$+0.33$	-0.31
Middle layer only	$+1.05$	-0.09	-0.06
Last layer only	$+1.57$	$+0.05$	-0.03
Late half (FOMO)	$\mathbf{+2.25}$	$\mathbf{+0.87}$	$\mathbf{+0.56}$

Table 7: Frozen foundation model performance on the Polymarket development dataset (977 origins).

Model	Configuration	$C_Q \downarrow$	Gain (%)	MAE ↓	Cov80	Width80
Toto-4M	Target only	0.014855	–	0.017066	0.8194	0.05526
	Native	0.014206	+4.37	0.016312	0.8119	0.05020
	FOMO	**0.014206**	**+4.37**	**0.016312**	**0.8119**	**0.05020**
Chronos-2	Target only	0.014826	–	0.016973	0.7952	**0.05022**
	Native	0.014984	-1.07	0.017153	0.7869	0.05064
	FOMO	**0.014697**	**+0.87**	**0.016785**	**0.8046**	0.05121
TimesFM-3	Target only	0.014368	–	0.016517	0.7763	**0.04999**
	Native	0.014628	-1.81	0.016519	0.7939	0.06088
	FOMO	**0.014288**	**+0.56**	**0.016351**	**0.7828**	0.05133
Moirai-2	Target only	0.014375	–	0.016626	0.7674	**0.04329**
	Native	0.016613	-15.57	0.019276	0.8481	0.07017
	FOMO	**0.014052**	**+2.25**	**0.016095**	**0.8069**	0.04632

D ACTIVATION INTERVENTIONS: SCORES AND CONTROLS

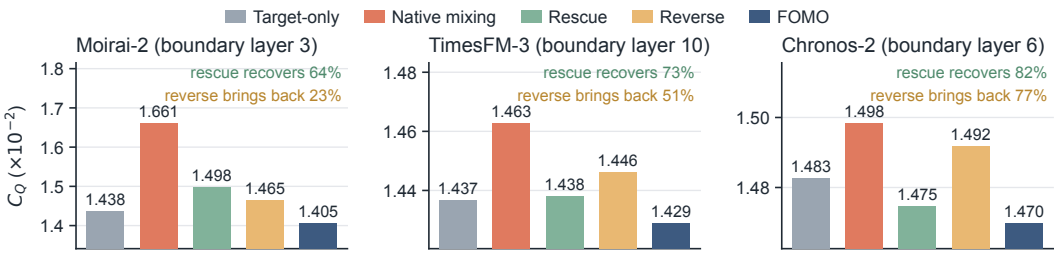

Figure 5: **Activation patching across network midpoint.** Quantile scores ($C_Q \downarrow$). Swapping FOMO's target state into the native run (green) recovers 64–82% of the error gap between native mixing and FOMO; moving the native target state into the FOMO run (orange) brings back 23–77% of it.

Table 8: Midpoint activation interventions (development dataset, $C_Q \downarrow$). Rescue: native pass receives late target states. Reverse: late pass receives native target states (equivalently, native pass receives late neighbor states). Rescue recovery: fraction of the native–late gap closed. Advantage retained: $(C_Q^{\text{native}} - C_Q^{\text{reverse}})/(C_Q^{\text{native}} - C_Q^{\text{late}})$. †: target-resampled 95% interval excludes zero; ‡: target and day intervals exclude zero. Native/late rows are the intervention runs' own passes.

	Moirai-2 (b=3)	TimesFM-3 (b=10)	Chronos-2 (b=6)
Target only	0.014375	0.014368	0.014826
Native	0.016613	0.014628	0.014984
Late (FOMO)	0.014052	0.014288	0.014697
Rescue: native ← late target states	0.014975‡	0.014379†	0.014747
recovery of native–late gap	64%	73%	82%
Reverse: late ← native target states	0.014651‡	0.014463	0.014917
advantage retained after reverse swap	77%	49%	23%
Control: native ← norm-matched noise	0.018421	0.021062	0.014876

E REPRESENTATION PROBES: FULL RESULTS AND MULTI-SEED ROBUSTNESS

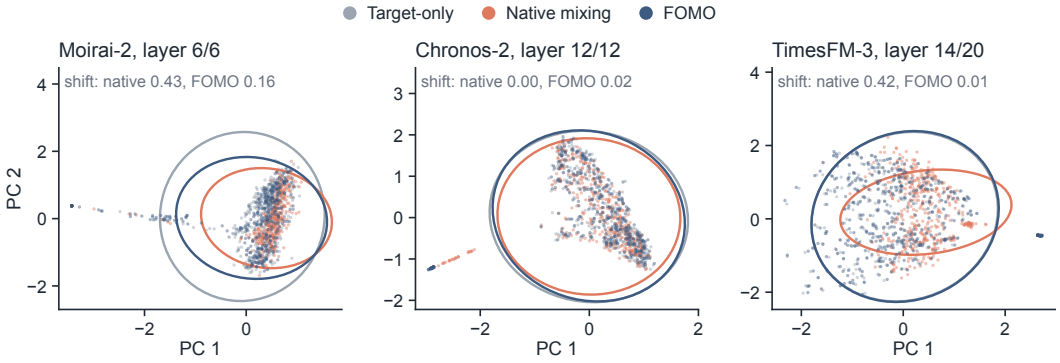

Figure 6: **Distribution of target hidden states with and without context.** Target history-token states (development dataset) are standardized and projected onto the top two principal components of the target-only states; ellipses show two standard deviations. **(a)** Native mixing displaces the target's states from their target-only distribution. **(b)** Under FOMO they remain close to it.

Figure 6 visualizes on the development set the displacement that Table 2 measures on the held-out domains.

Tables 9–11 report held-out R^2 for both probe targets under the four schedules, using the target's last history-token state (the specification plotted in Figure 7). To ensure that these results are not artifacts of a single fold partition, we repeated 5-fold cross-validation across 10 independent random seeds. Across all 10 seeds, Moirai-2's late-routing advantage is strictly positive at every layer (10-seed midpoint delta $+0.246 \pm 0.023$; Table 9 shows a single seed). Under a pooled specification that averages all history-token states, the Moirai-2 volatility gap remains positive at all layers (late 0.38–0.48 vs. native 0.31–0.41), Chronos-2 favors late routing at 10 of 12 layers by 0.02 on average, and TimesFM-3 shows modest reversals (-0.03 on average). Regularization sweeps across $\alpha \in [10, 1000]$ across all 10 seeds confirm this hierarchy: Moirai-2's gap remains large and positive across all α values ($+0.176$ to $+0.310$), TimesFM-3 remains positive across all α under last-token probing ($+0.005$ to $+0.021$), and Chronos-2 remains near neutral. Last-price decodability is higher under native mixing in Chronos-2 and TimesFM-3. These results are consistent with representation preservation tracking forecast degradation across architectures.

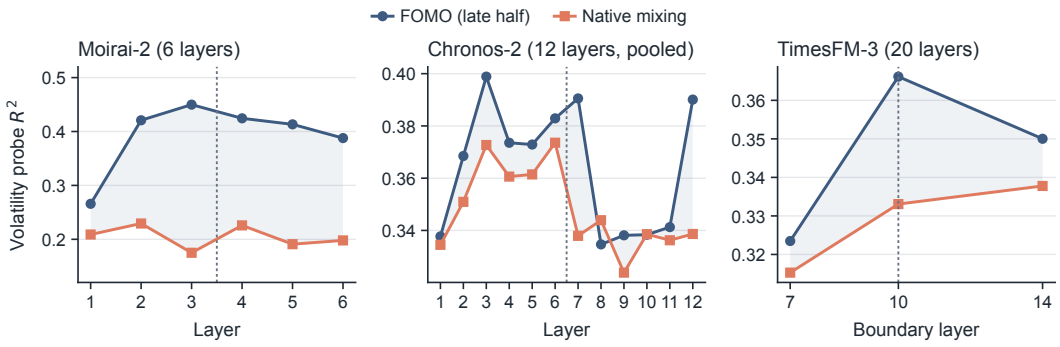

Figure 7: **Target volatility decodability across depth.** Held-out ridge probe R^2 predicting trailing 72-hour volatility from target hidden states. Native mixing degrades decodability early, while late routing preserves target dynamics (shaded gap), tracking the forecast degradation hierarchy across models.

Table 9: Moirai-2 probes (977 windows, last history token). Late and target-only coincide through layer 3, native and early through layer 3, since the schedules differ only from layer 4.

Target	Schedule	L1	L2	L3	L4	L5	L6
Volatility	Native	0.209	0.229	0.175	0.226	0.191	0.198
	Early half	0.209	0.229	0.175	0.282	0.270	0.281
	Late half	0.266	0.421	0.450	0.424	0.413	0.388
	Target only	0.266	0.421	0.450	0.437	0.446	0.456
Last price	Native	0.199	0.193	0.254	0.229	0.228	0.238
	Early half	0.199	0.193	0.254	0.253	0.260	0.286
	Late half	0.205	0.242	0.256	0.281	0.307	0.300
	Target only	0.205	0.242	0.256	0.259	0.272	0.266

Table 10: Chronos-2 volatility probes (977 windows, last history token). Schedules differ from layer 7.

Schedule	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
Native	0.123	0.225	0.234	0.302	0.242	0.206	0.203	0.152	0.179	0.210	0.115	0.175
Early half	0.123	0.225	0.234	0.302	0.242	0.206	0.198	0.153	0.167	0.144	0.126	0.179
Late half	0.126	0.218	0.241	0.296	0.255	0.203	0.181	0.192	0.145	0.141	0.139	0.230
Target only	0.126	0.218	0.241	0.296	0.255	0.203	0.188	0.174	0.157	0.121	0.163	0.239

Table 11: TimesFM-3 probes (977 windows, last history token) at the three intervention boundaries. Schedules differ from layer 11.

Target	Schedule	L7	L10	L14
Volatility	Native	0.315	0.333	0.338
	Early half	0.315	0.333	0.332
	Late half	0.323	0.366	0.350
	Target only	0.323	0.366	0.350
Last price	Native	0.003	0.127	0.097
	Early half	0.003	0.127	0.084
	Late half	−0.021	0.037	−0.008
	Target only	−0.021	0.037	−0.012

F Raw Scores on Prediction Markets

Table 12 reports the quantile scores behind the prediction-market columns of Table 1.

Table 12: Foundation model performance across evaluation datasets under quantile score ($C_Q \downarrow$, lower is better; reported $\times 10^{-4}$ for later-period Polymarket and $\times 10^{-2}$ otherwise). Gain (%) indicates relative error reduction over target-only input for each model; best input per model in bold. Worst: lowest gain across the three datasets. ‡: paired bootstrap 95% interval excludes zero.

Model	Input	Later-period (2,000w)		Kalshi (555w)		Development (977w)		Worst
		C_Q	Gain (%)	C_Q	Gain (%)	C_Q	Gain (%)	Gain (%)
Toto-4M	Target only	3.498	–	1.125	–	1.486	–	–
	Native mixing	3.343	$+4.41\%^{\ddagger}$	1.102	$+2.06\%$	1.421	$+4.37\%^{\ddagger}$	$+2.06\%$
	FOMO	**3.343**	$\mathbf{+4.41\%}^{\ddagger}$	**1.102**	$\mathbf{+2.06\%}$	**1.421**	$\mathbf{+4.37\%}^{\ddagger}$	$\mathbf{+2.06\%}$
Moirai-2	Target only	3.522	–	1.124	–	1.438	–	–
	Native mixing	3.590	-1.93%	1.297	$-15.33\%^{\ddagger}$	1.661	$-15.57\%^{\ddagger}$	-15.57%
	FOMO	**3.381**	$\mathbf{+3.99\%}^{\ddagger}$	**1.105**	$\mathbf{+1.69\%}$	**1.405**	$\mathbf{+2.25\%}^{\ddagger}$	$\mathbf{+1.69\%}$
TimesFM-3	Target only	3.570	–	1.108	–	1.437	–	–
	Native mixing	**3.328**	$\mathbf{+6.77\%}^{\ddagger}$	1.156	$-4.32\%^{\ddagger}$	1.463	-1.81%	-4.32%
	FOMO	3.517	$+1.47\%^{\ddagger}$	**1.099**	$\mathbf{+0.78\%}$	**1.429**	$\mathbf{+0.56\%}$	$\mathbf{+0.56\%}$
Chronos-2	Target only	3.596	–	1.145	–	1.483	–	–
	Native mixing	**3.529**	$\mathbf{+1.86\%}$	**1.117**	$+2.45\%$	1.498	-1.07%	-1.07%
	FOMO	3.555	$+1.15\%^{\ddagger}$	1.135	$+0.83\%^{\ddagger}$	**1.470**	$\mathbf{+0.87\%}^{\ddagger}$	$\mathbf{+0.83\%}$

G KALSHI CROSS-PLATFORM DATASET

Table 13 reports the Kalshi dataset (555 windows, 179 targets, 108 provider events, 47 calendar days) under the same protocol and target-weighted aggregation as Table 12. Only Chronos-2's FOMO-vs-target-only interval excludes zero (target-clustered $[-2.04, -0.06] \times 10^{-4}$; day-clustered $[-2.01, -0.38] \times 10^{-4}$); the Toto-4M, Moirai-2, and TimesFM-3 intervals include zero under both clusterings. Native mixing degrades Moirai-2 by 15.3% and TimesFM-3 by 4.3%, both of which FOMO reverses, while native Chronos-2 ($+2.5\%$) outscores FOMO ($+0.8\%$).

Table 13: Kalshi dataset ($C_Q \downarrow$; gain vs. the same model's target-only forecast; coverage is target-weighted). ‡: target- and day-clustered 95% intervals exclude zero.

Model	Input	C_Q	Gain	MAE	Cov80
Toto-4M	Target only	0.011252	–	0.013346	0.853
	Native = FOMO	0.011021	$+2.1\%$	0.013221	0.846
Moirai-2	Target only	0.011242	–	0.013608	0.805
	Native mixing	0.012966	-15.3%	0.014978	0.840
	FOMO	0.011052	$+1.7\%$	0.013271	0.824
TimesFM-3	Target only	0.011081	–	0.013152	0.827
	Native mixing	0.011559	-4.3%	0.013650	0.830
	FOMO	0.010994	$+0.8\%$	0.013084	0.826
Chronos-2	Target only	0.011449	–	0.013765	0.835
	Native mixing	0.011168	$+2.5\%$	0.013479	0.824
	FOMO	0.011354	$+0.8\%^{\ddagger}$	0.013714	0.846

H EQUITY, CRYPTO, AND EXTERNAL BENCHMARKS

Equities. We take daily adjusted closes for the 110 tickers that appear both in a 2009–2015 US equity panel and in the current S&P 500 constituent list, and sample 100 targets. Each target has 20 forecast origins spaced 21 trading days apart (July 2013 to February 2015), with 256 trading days of context and a 24-day horizon. Neighbors are ranked by shared GICS sub-industry (score 3) and sector (score 1), with ties broken by a fixed seeded order; on average 6.3 of 7 neighbors share the target's sector and 1.7 its sub-industry. Random neighbors are seeded draws from the same universe,

and every series is divided by its own close at the origin. Using current membership introduces survivorship bias.

Crypto. We use daily closes of 41 large-cap coins (July 2024 to July 2026), with 20 origins spaced 21 days apart and the same context and horizon. Sector and sub-category labels (e.g., smart-contract platform/layer-1, currency/privacy, DeFi/lending) are author-assigned from common market classifications; on average 5.8 of 7 neighbors share the target's sector and 3.4 its sub-category.

Table 14 reports the equity and crypto gains with the harmed and worst-10% measures.

Table 14: Beyond prediction markets: equities (100 S&P 500 stocks, 2,000 monthly windows, 2013–2015; GICS sector/sub-industry neighbors) and crypto (41 coins, 820 windows, 2025–2026; sector/sub-category neighbors). Gain over target-only (%); ‡: target-clustered 95% interval excludes zero. Harmed: share of targets made >0.1% worse. Worst-10%: excess error of the worst decile of targets, % of the target-only mean.

		Gain (%)			Harmed (%)	Worst-10% (%)
Domain	Model	Native	Early half	FOMO	native → FOMO	native → FOMO
Equities	Moirai-2	$-3.22^{\ddagger}$	$-2.09^{\ddagger}$	$\mathbf{+0.84}^{\ddagger}$	62 → **41**	20.3 → **6.2**
	Chronos-2	$-1.15^{\ddagger}$	$-1.87^{\ddagger}$	$\mathbf{+0.55}^{\ddagger}$	53 → **30**	12.6 → **0.9**
	TimesFM-3	$-2.16^{\ddagger}$	$-4.41^{\ddagger}$	$\mathbf{+0.60}^{\ddagger}$	65 → **22**	15.0 → **0.9**
Crypto	Moirai-2	$-3.55^{\ddagger}$	$-2.67^{\ddagger}$	$\mathbf{+0.85}$	66 → **32**	20.9 → **4.8**
	Chronos-2	$+3.11^{\ddagger}$	$+1.54^{\ddagger}$	$\mathbf{+0.61}^{\ddagger}$	17 → 22	1.4 → **0.9**
	TimesFM-3	$+2.90^{\ddagger}$	$-1.99^{\ddagger}$	$\mathbf{+1.42}^{\ddagger}$	27 → **7**	5.7 → **0.9**

Risk measures on prediction markets. Table 15 reports the harmed and worst-10% measures for the Kalshi runs of Table 12, and Table 16 reports them for the later-period Polymarket runs.

Table 15: Kalshi (555 windows, 179 targets): risk measures for the runs in Table 12. ‡: target-clustered 95% interval excludes zero.

	Gain over target-only (%)			Targets harmed (%)	Worst-10% excess (%)	
Model	Native	Early half	FOMO	native → FOMO	native → FOMO	Persistence
Moirai-2	$-15.33^{\ddagger}$	$-4.72^{\ddagger}$	$\mathbf{+1.69}$	76 → **54**	115.7 → **22.8**	−21.13
Chronos-2	+2.45	−0.59	$\mathbf{+0.83}^{\ddagger}$	39 → **35**	40.6 → **5.7**	−18.94
TimesFM-3	$-4.32^{\ddagger}$	$-6.05^{\ddagger}$	$\mathbf{+0.78}$	60 → **42**	45.7 → **8.9**	−22.89

Table 16: Later-period Polymarket (2,000 windows, 100 targets): risk measures for the runs in Table 12. ‡: target-clustered 95% interval excludes zero. Persistence: gain of a no-change forecast that places every quantile at the last observed price.

	Gain over target-only (%)			Targets harmed (%)	Worst-10% excess (%)	
Model	Native	Early half	FOMO	native → FOMO	native → FOMO	Persistence
Moirai-2	−1.93	+0.54	$\mathbf{+3.99}^{\ddagger}$	70 → **41**	31.3 → **16.2**	+1.69
Chronos-2	+1.86	+1.96	$\mathbf{+1.15}^{\ddagger}$	38 → **29**	13.9 → **2.5**	+3.71
TimesFM-3	$+6.77^{\ddagger}$	$+4.62^{\ddagger}$	$\mathbf{+1.47}^{\ddagger}$	23 → 26	12.2 → **1.7**	+3.01

Electricity. On 321 electricity load series (GluonTS), with neighbors chosen by past-change similarity and no shared metadata, native mixing degrades Moirai-2 by 26.3% and FOMO by 18.0%, while the intervals for Chronos-2 and TimesFM-3 include zero under both allocations.

I MASKING-BOUNDARY SWEEP

We sweep the masking boundary k: cross-series attention is blocked in the first k layers and left unchanged afterwards (k=0 is native mixing; FOMO masks the first half of the layers). Table 17 compares native mixing, the *minimal safe prefix* (the smallest k with a positive gain on the development dataset, selected once per model), and FOMO. Masking only the first few layers already removes the average degradation on every dataset and can recover part of native mixing's upside (e.g., Chronos-2 on Kalshi and later-period Polymarket), while FOMO's half-depth boundary has the smallest worst-10% loss in most settings.

Table 17: Masking-boundary sweep: gain over target-only (%) and worst-10% excess error (%) for native mixing, the minimal safe prefix k (chosen on the development dataset), and FOMO.

		Native		Minimal safe prefix		FOMO (half)	
Model	Dataset	Gain	Worst-10%	Gain	Worst-10%	Gain	Worst-10%
Moirai-2 (k=2)	Development	−15.57	163.9	+1.21	40.2	+2.25	30.0
	Later-period	−1.93	31.3	+4.13	16.2	+3.99	16.2
	Kalshi	−15.33	115.7	+0.16	35.1	+1.69	22.8
	Equities	−3.22	20.3	+1.50	6.8	+0.84	6.2
	Crypto	−3.55	20.9	+0.52	6.8	+0.85	4.8
Chronos-2 (k=1)	Development	−1.07	63.9	+0.74	26.2	+0.87	10.6
	Later-period	+1.86	13.9	+2.14	7.8	+1.15	2.5
	Kalshi	+2.45	40.6	+2.68	16.1	+0.83	5.7
	Equities	−1.15	12.6	+0.87	2.6	+0.55	0.9
	Crypto	+3.11	1.4	+1.73	0.3	+0.61	0.9
TimesFM-3 (k=4)	Development	−1.81	56.2	+0.29	25.5	+0.56	14.1
	Later-period	+6.77	12.2	+2.92	7.1	+1.47	1.7
	Kalshi	−4.32	45.7	+0.61	14.3	+0.78	8.9
	Equities	−2.16	15.0	+0.63	2.3	+0.60	0.9
	Crypto	+2.90	5.7	+2.56	0.8	+1.42	0.9

J SIMPLE ALTERNATIVES FOR SAFE CONTEXT USE

We compare FOMO with three simpler ways to make context safe, all computed from the same saved forecasts (Table 18). *Recalibration* widens or narrows the target-only quantiles around their median by one factor per model, chosen on the development set. As a reference, we also choose the factor on each dataset itself (*oracle*). *Averaging* takes the mean of the target-only and native-mixing quantiles, and *weighted averaging* uses $(1 - w)$ target-only $+ w$ native quantiles with w chosen per model on the development set ($w = 0.05$ for Moirai-2, 0.4 for Chronos-2, and 0.2 for TimesFM-3). *Selection* uses native mixing or target-only per model, whichever scores better on the development set. Native mixing is worse there for all three models, so selection always returns the target-only forecast (gain 0). Recalibration gains at most 0.27% even with the oracle factor, and FOMO exceeds it in all 15 settings. Averaging keeps some of native mixing's upside where early correlations help (e.g., TimesFM-3 on later-period Polymarket) and degrades 7 of the 15 settings, by up to 5.2%. Weighted averaging is the strongest alternative. Its development-set weight keeps it close to target-only, which gives it a smaller worst-10% loss than FOMO in most held-out settings. It gains less than FOMO in 10 of 15 settings, mostly because it barely uses context for Moirai-2, and it scores higher than FOMO only where native mixing itself helps (Chronos-2 on later-period Polymarket, Kalshi, and crypto; TimesFM-3 on later-period Polymarket and crypto).

Table 18: FOMO versus simpler safeguards: gain over target-only (%, best in bold), with the worst-10% excess error in parentheses. Recal.: target-only quantiles rescaled around the median, with the factor chosen on the development set or on each dataset (oracle). Average: mean of target-only and native-mixing quantiles. Weighted: $(1 - w)$ target-only $+ w$ native, with w chosen per model on the development set. Selection on the development set returns target-only for every model (gain 0) and is omitted.

Dataset	Model	FOMO	Recal. (dev)	Recal. (oracle)	Average	Weighted (dev)
Development	Moirai-2	**+2.25** (30.0)	0.00 (0.0)	0.00 (0.0)	−5.19 (73.6)	+0.07 (5.5)
	Chronos-2	**+0.87** (10.6)	+0.16 (6.3)	+0.16 (6.3)	+0.56 (29.0)	+0.61 (22.7)
	TimesFM-3	**+0.56** (14.1)	+0.08 (3.5)	+0.08 (3.5)	−0.19 (26.6)	+0.14 (10.2)
Later-period	Moirai-2	**+3.99** (16.2)	0.00 (0.0)	+0.27 (1.6)	+2.21 (9.5)	+0.55 (0.6)
	Chronos-2	+1.15 (2.5)	+0.21 (1.3)	+0.23 (2.2)	**+2.03** (3.3)	+1.77 (2.2)
	TimesFM-3	+1.47 (1.7)	+0.03 (0.7)	+0.03 (0.7)	**+4.55** (4.3)	+2.11 (1.5)
Kalshi	Moirai-2	**+1.69** (22.8)	0.00 (0.0)	0.00 (0.0)	−5.06 (48.8)	−0.03 (3.2)
	Chronos-2	+0.83 (5.7)	+0.26 (4.7)	+0.18 (2.1)	**+2.46** (14.8)	+2.16 (10.9)
	TimesFM-3	**+0.78** (8.9)	+0.15 (1.9)	+0.20 (4.0)	−1.42 (20.9)	−0.33 (7.8)
Equities	Moirai-2	**+0.84** (6.2)	0.00 (0.0)	+0.06 (0.4)	−0.24 (7.2)	+0.13 (0.5)
	Chronos-2	**+0.55** (0.9)	−0.43 (1.7)	+0.10 (0.9)	+0.01 (5.4)	+0.10 (4.2)
	TimesFM-3	**+0.60** (0.9)	−0.23 (0.8)	+0.24 (1.2)	−0.38 (6.3)	+0.02 (2.3)
Crypto	Moirai-2	**+0.85** (4.8)	0.00 (0.0)	+0.17 (0.5)	−0.63 (6.8)	+0.07 (0.3)
	Chronos-2	+0.61 (0.9)	+0.01 (0.8)	+0.05 (0.3)	**+2.22** (0.3)	+1.88 (0.2)
	TimesFM-3	+1.42 (0.9)	−0.08 (0.4)	0.00 (0.0)	**+2.74** (1.2)	+1.42 (0.1)

K Held-Out Depth and Context Ablations

Table 19 repeats the development-set depth and content ablations of Appendices C and L on the twelve held-out settings. Interaction in the last layer only gains less than FOMO in every setting. Interleaved schedules, which include early layers, degrade TimesFM-3 on every held-out dataset, and FOMO gains the most with all seven neighbors in every setting. Interleaving scores higher than FOMO in two settings (Moirai-2 on later-period Polymarket, Chronos-2 on Kalshi), and its losses on TimesFM-3 make it unsafe as a default. Table 20 reports the empirical coverage and width of the 80% interval (the 0.1 and 0.9 quantiles). FOMO keeps interval widths within 0.98–1.08 times target-only in 11 of 12 settings, so its gains are not driven by wider intervals.

Table 19: Held-out depth and context ablations: gain over target-only (%). Depth schedules use the metadata neighbors; content arms use late-half allocation (FOMO). Last: last layer only. Inter.: equally spaced layers. Early: first half. K: number of neighbors (FOMO uses K=7). Dup.: seven copies of the target. Noise: half of the neighbors replaced by random admissible ones. Miss.: half of the neighbor observations masked. Polymarket: later-period dataset.

		Depth schedule				Context content (FOMO)				
Dataset	Model	Last	Inter.	Early	FOMO	K=1	K=3	Dup.	Noise	Miss.
Polymarket	Moirai-2	+2.52	+5.65	+0.54	**+3.99**	+1.31	+1.27	0.00	+4.39	+4.13
	Chronos-2	−0.18	+1.04	+1.96	**+1.15**	+0.05	+0.67	0.00	+1.29	+1.12
	TimesFM-3	−0.02	−1.17	+4.62	**+1.47**	+1.05	+1.21	0.00	+1.53	+1.55
Kalshi	Moirai-2	+0.93	+0.37	−4.72	**+1.69**	+1.03	+1.42	0.00	+2.14	+1.33
	Chronos-2	+0.11	+1.56	−0.59	**+0.83**	+0.25	+0.80	0.00	+0.82	+0.79
	TimesFM-3	0.00	−4.35	−6.05	**+0.78**	−0.02	+0.68	0.00	+0.71	+0.88
Equities	Moirai-2	+0.52	+0.46	−2.09	**+0.84**	+0.17	+0.35	0.00	+1.31	−0.24
	Chronos-2	+0.07	+0.34	−1.87	**+0.55**	+0.25	+0.45	0.00	+0.63	+0.35
	TimesFM-3	0.00	−4.06	−4.41	**+0.60**	+0.13	+0.35	0.00	+0.59	+0.42
Crypto	Moirai-2	+0.22	−0.18	−2.67	**+0.85**	−0.58	+0.63	0.00	+1.07	+0.96
	Chronos-2	+0.10	+0.61	+1.54	**+0.61**	+0.21	+0.49	0.00	+0.64	+0.27
	TimesFM-3	−0.03	−3.07	−1.99	**+1.42**	+0.29	+0.96	0.00	+1.15	+0.31

Table 20: Empirical coverage of the nominal 80% interval (pooled over windows) and its mean width relative to target-only on the held-out settings. Polymarket: later-period dataset.

		Target only		Native mixing		FOMO	
Dataset	Model	Cov.	Width	Cov.	Width	Cov.	Width
Polymarket	Moirai-2	0.837	1.00	0.856	1.00	0.843	0.73
	Chronos-2	0.890	1.00	0.885	1.01	0.880	0.98
	TimesFM-3	0.702	1.00	0.693	1.04	0.702	1.04
Kalshi	Moirai-2	0.818	1.00	0.850	1.23	0.832	0.99
	Chronos-2	0.843	1.00	0.833	0.98	0.850	1.00
	TimesFM-3	0.831	1.00	0.835	1.09	0.829	1.01
Equities	Moirai-2	0.772	1.00	0.839	1.27	0.802	1.08
	Chronos-2	0.762	1.00	0.721	0.92	0.769	1.01
	TimesFM-3	0.731	1.00	0.746	1.05	0.737	1.01
Crypto	Moirai-2	0.820	1.00	0.873	1.25	0.840	1.06
	Chronos-2	0.790	1.00	0.779	0.92	0.797	1.00
	TimesFM-3	0.783	1.00	0.806	1.08	0.795	1.01

L Retrieval and History-Content Controls: Full Arms

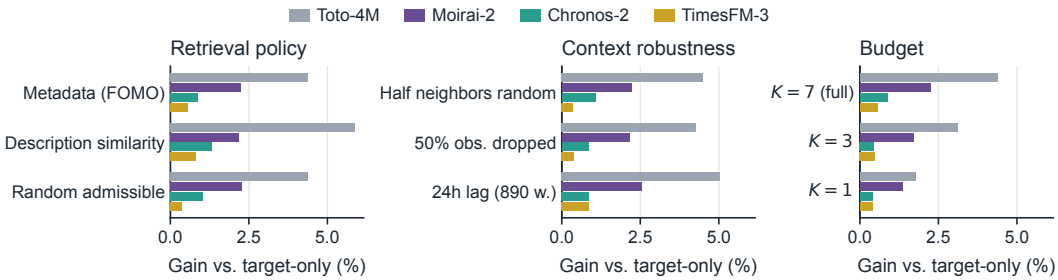

Figure 8: **Retrieval sensitivity, robustness, and budget scaling across four models.** Relative C_Q reduction versus target-only (development dataset, 977 windows). Full ablations are in Appendix L.

Table 21: Matched lag-control scores ($C_Q \downarrow$, 890 windows in each row). These absolute scores report the matched lag evaluation.

Context	Toto-4M	Moirai-2	Chronos-2	TimesFM-3
Unlagged metadata	0.012441	0.012351	0.012859	0.012620
Lagged 24 hours	0.012436	0.012382	0.012877	0.012616

Table 22 lists every retrieval and content arm on the development dataset as relative change in C_Q versus target-only, including the individual random seeds and the 100% noise and 50% missingness arms. In the *neighbor noise* arms, the stated fraction of retrieved neighbors is replaced by random admissible neighbors (so the 100% arm coincides with random seed 17); in the *missing* arms, the stated fraction of neighbor observations is masked after panel assembly and filled with the last observed value. The lag arms are scored on the 890-window subset where a 24-hour-lagged neighbor panel exists.

Table 22: All retrieval and content arms on the development dataset (relative change vs. target-only; positive is better). Allocation is late half for Moirai-2, Chronos-2, TimesFM-3 and native for Toto-4M.

Arm	Toto-4M	Moirai-2	Chronos-2	TimesFM-3
Metadata	+4.37	+2.25	+0.87	+0.56
Description similarity	+5.86	+2.17	+1.32	+0.80
Random, seed 17	+4.81	+2.47	+1.05	+0.11
Random, seed 27	+3.84	+2.38	+0.89	+0.53
Random, seed 37	+4.45	+2.07	+1.08	+0.53
Random, seed 47	+5.24	+2.44	+0.97	+0.38
Random, seed 57	+3.62	+1.99	+1.11	+0.36
Budget K=1	+1.77	+1.36	+0.39	+0.40
Budget K=3	+3.10	+1.70	+0.45	+0.48
Neighbors replaced by random (100%)	+4.81	+2.47	+1.05	+0.11
Neighbor missing 50%	+4.27	+2.16	+0.85	+0.39
Constant latest value	+3.20	−66.89	+0.46	−0.41
Copies of target	0.00	0.00	0.00	0.00

M LATENCY

Table 23 reports single-request latency on an NVIDIA H100 (two test fixtures, one timed request each) with seven exact neighbor histories. FOMO's p50 overhead over target-only forecasting ranges from 0.2 ms (Toto-4M) to 7.4 ms (TimesFM-3). Peak GPU memory per request on the development dataset is 65 MB for Toto-4M, 84 MB for Moirai-2, 613 MB for Chronos-2, and 1.44 GB for TimesFM-3 under FOMO, within 10% of the corresponding target-only figures for all models except Chronos-2 (+17%).

Table 23: Single-request inference latency (p50 / p95 over two fixtures, ms) on an NVIDIA H100.

Model	Target only	Native mixing	FOMO	Overhead (p50)
Toto-4M	11.70 / 11.79	12.06 / 12.20	11.88 / 11.97	+0.18
Chronos-2	13.67 / 13.68	14.48 / 14.50	14.65 / 14.68	+0.97
TimesFM-3	62.80 / 63.38	69.02 / 74.10	70.18 / 75.01	+7.38
Moirai-2	9.74 / 9.80	9.80 / 9.84	15.56 / 17.95	+5.82

N REPRODUCIBILITY AND AUDIT RECORDS

Frozen-model checkpoint hashes are verified before and after every evaluation and intervention run; self-patch interventions reproduce the unpatched forecast with zero error in all three models. Dataset inputs, model weights, and analysis scripts are SHA-256 hashed and archived in the repository audit log.

O ADDITIONAL DISCUSSION AND SCOPE

What the evidence supports. We view the contribution as a design rule for context integration: a frozen model with cross-series attention should encode each series alone before interacting with others. FOMO reverses degradation where it occurs, but native mixing scores better for Chronos-2 and TimesFM-3 on later-period Polymarket and crypto, and for Chronos-2 on Kalshi. The half-depth cut was never tuned per model, and we have not explored learned schedules.

Retrieval matters less than its integration. Under late routing, metadata retrieval does not beat matched random draws from the admissible pool, and the gain survives 100% noise, 50% missingness, and a 24-hour lag on the neighbors while collapsing when they lose their temporal shape (Figure 8). The models appear to benefit from realistic market trajectories in context without extracting neighbor-

specific signals. The metadata graph establishes a causal candidate set without look-ahead, without isolating uniquely informative individual contracts.

Domain limitations of the metadata graph. Retrieval relies on structure that many markets lack. Entities, event families, thresholds, and expiries are parsed from contract slugs against a fixed alias vocabulary; unlisted entities receive only weak platform- or area-level edges, expiry dates parse for fewer than half of corpus contracts, and threshold-ladder relations abstain on Kalshi's native ticker identifiers. For equities and crypto we substitute sector labels for contract metadata (Appendix H); transfer to a third prediction venue, Manifold, failed. Evaluated datasets require at least one admissible neighbor, so the share of raw markets that would receive no context is unmeasured. FOMO's depth allocation is architecture-general; its contract-level retrieval policy is prediction-market specific.

Statistical scope and provenance. The causal past-residual quantile baseline beats every FOMO-augmented model on the later-period dataset, so FOMO makes context safe for a frozen foundation model without surpassing this baseline there. On Kalshi only Chronos-2's FOMO-vs-target interval excludes zero. The mechanism analyses use a dataset inspected during development, report many paired contrasts without multiplicity correction, and rely on probes whose gaps in TimesFM-3 and Chronos-2 are small; we have not identified a complete mechanism. Finally, pretraining corpora for the evaluated models are undisclosed, so historical market prices may have been seen during pretraining, and Kalshi's calendar overlaps Polymarket's, so the Kalshi dataset tests transfer across venues and contracts within the same period.

P Visualizing Prediction Market Datasets Through Examples

To provide concrete intuition for prediction market forecasting and FOMO's retrieval policy, this section details the structure of raw contract data and illustrates a representative retrieval neighborhood from our evaluation datasets.

P.1 Contract Representation and As-Of Alignment

Each prediction market contract $i \in \mathcal{V}$ corresponds to a specific verifiable future event, with hourly quoted trade prices in $[0, 1]$ representing risk-neutral probability assessments. Alongside observed prices, each contract record contains structured metadata fields:

- **Platform:** Trading venue (e.g., Polymarket, Kalshi).
- **Area and Category:** Domain classification (e.g., crypto, finance, elections, sports).
- **Entity:** Underlying subject or asset (e.g., Bitcoin, Ethereum, CPI, Federal Reserve).
- **Event Family:** Cluster of interrelated contracts sharing common structural conditions (e.g., Bitcoin price-threshold ladder, presidential election outcome).
- **Expiry Date:** Timestamp at which the contract resolves to 0 or 1.
- **Description Tokens:** Normalized unigram tokens extracted from the contract title and specification text.

At each forecast origin t, FOMO constructs an input panel $\boldsymbol{X}_i^{\pi_{\text{meta}}}(t) \in \mathbb{R}^{(k+1)\times 256}$ containing the target history and up to $K = 7$ neighbor series. Strictly causal alignment enforces two rules:

1. **Causal Candidate Admissibility:** A candidate contract j is admitted into target i's ranking only if j's initial listing timestamp occurred at or before i's first observation. This prevents future contracts from contaminating historical context graphs.
2. **Strict As-Of Availability:** Every price entry $x_j(u)$ in the panel represents the last valid transaction recorded at or before hourly grid timestamp $u \leq t$. Allowed staleness thresholds are 2 hours for the target and 6 hours for context neighbors. Positions lacking valid observations are masked.

P.2 Retrieval Details

Candidate groups are formed from area, event family, expiry, description tokens, entity, entity–threshold pairs, event type, the first two levels of a taxonomy path, and mutually exclusive outcome sets. A candidate's priority is the sum of the weights of the groups it shares with the target (outcome set 6; entity, threshold, and family 4; taxonomy and event type 2; expiry and token 1; area 0.5). Each group contributes at most 512 members, and the 512 highest-priority candidates are scored with Eq. 7. The card term w_{card} adds 0.1 for the same platform, 0.8 for the same area, 0.8 for the same event type, 0.7 for the same expiry, 1.5 for a shared entity, 1.0 for a shared taxonomy branch, 2.5 for a threshold ladder, and 3.0 for a cross-platform duplicate, with a further 2.5 when the two contracts are mutually exclusive outcomes of one event.

P.3 Example of Retrieved Context

Table 24 shows the metadata neighborhood retrieved for one target in the development dataset.

Table 24: Retrieved neighborhood ($K=7$) for one development-set target, listed in retrieval order. Contract titles are paraphrased from their slugs.

Role	Contract
Target	**Will the Pittsburgh Pirates win 100 or more games in the 2026 MLB regular season?**
Neighbor 1	Will the Arizona Diamondbacks win 100 or more games in the 2026 MLB regular season?
Neighbor 2	Will the Baltimore Orioles win 100 or more games in the 2026 MLB regular season?
Neighbor 3	Will the Seattle Mariners win 100 or more games in the 2026 MLB regular season?
Neighbor 4	Will the Cincinnati Reds have the longest winning streak of the 2026 MLB regular season?
Neighbor 5	Will the Houston Astros have the longest winning streak of the 2026 MLB regular season?
Neighbor 6	Will the Golden State Valkyries make the 2026 WNBA playoffs?
Neighbor 7	Will the Seattle Mariners clinch a spot in the 2026 MLB postseason?

The first three neighbors ask the same question about other teams, and neighbors 4, 5, and 7 are other markets on the same MLB season. Neighbor 6 is a weaker match that shares only the sport category and season. The graph finds contracts on the same topic, which does not guarantee that their prices move together (Appendix O).