WHEN DOES EXACT-ZERO ATTENTION HELP?

Anonymous authors
Paper under double-blind review

ABSTRACT

When does exact-zero attention improve prediction over bounded dense softmax? We answer this by disentangling three mechanisms: bounded representation, implicit optimization bias, and network-level nuisance propagation. Theoretically, balancing a lower bound on bounded softmax residual-noise risk against a finite-sample support-recovery bound for sparsemax yields a sufficient condition: exact zeros reduce excess risk whenever support-estimation error drops below the retained noise floor. Across 2,400 controlled fits, this trade-off spans an empirical phase map over support sparsity, noise scale, and sample size. Solving both model classes to exact convex optimality decouples representation from optimization. The sparse advantage and its contraction with score range persist without an optimizer, while a closed-form leak term predicts the gap between optima across twelve orders of magnitude. Inside 300 frozen forecasters, the network-level mechanism is isolated: perturbing only the positions zeroed by sparsemax shifts predictions to 0.17 of what an equally large random set does, against 0.63 for its softmax twin. On real data, an input coordinate estimated strictly from the training split tracks the regime across the settings we test. Over 798 configurations drawn from a 1.2-million-record cryptocurrency panel and a 121-series US macroeconomic panel, spanning both attention directions, exact zeros lower test error in 773 cases (96.9%): by 2.32% across the cryptocurrency cross-section, 12.73% along a size-matched time axis, and 15.69% across the macroeconomic series, with intervals robust to overlapping-window dependence. Finally, a decision rule frozen and registered in advance returned all four of its predicted wins on two untouched datasets.

1 INTRODUCTION

When does exact-zero attention improve prediction over dense softmax? Transformer forecasters vary widely across time-series benchmarks. Meanwhile, linear and recurrent baselines remain competitive (Vaswani et al., 2017; Zhou et al., 2021; Nie et al., 2023; Liu et al., 2024; Zeng et al., 2023; Oreshkin et al., 2020). Sparse normalizations like sparsemax and entmax enforce exact truncation (Martins and Astudillo, 2016; Peters et al., 2019). However, whether sparsity improves generalization through representational expressiveness, optimization dynamics, or network filtering remains unresolved. Crucially, a positive softmax weight does not imply downstream layers use that nuisance feature. Conversely, exact truncation drops noise, but incurs a finite-sample support-estimation penalty.

We formalize this trade-off under a signal-plus-nuisance framework. Upstream representations often fail to purge irrelevant features. In that case, bounded logits force dense softmax to retain a strictly positive noise floor in the readout. Exact-zero selection eliminates this variance entirely. However, it risks omitting true predictive features under noisy support estimation. We balance a lower bound on softmax residual risk against a support-recovery bound for sparsemax. This yields a closed-form sufficient condition: *exact zeros reduce excess risk whenever support-estimation error drops below the softmax noise floor.*

We test these mechanisms across 2,400 controlled fits and four observational sources. Where inputs span few independent directions and most carry signal, the theory predicts a tie. That tie holds on 139 settings of a 1.2×10^6-record cryptocurrency panel, read by Multi-scale Inter-signal Sparse Attention (MISA), the probe forecaster we build for this study: a recurrent encoder over three time scales, with one exact-zero layer selecting time steps and a second selecting among the scales. It

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.

Table 1: Mechanistic claims and empirical validation strategy.

Mechanism / Claim	*Diagnostic Evidence*
Representation: Bounded logits restrict minimal softmax weights.	Risk bounds (§3), convex reference solutions, logit sweeps.
Optimization: Gradients yield empirical fits, not class optima.	Gap between empirical fits and convex references; initialization probes.
Network: Downstream layers determine nuisance transmission.	Frozen-model resampling, perturbation of zeroed coordinates, path ablations.

holds again on a pre-registered semiconductor ladder from a 384-month production panel down to one-minute equities, and on ETT, METR-LA and foreign-exchange controls. Because a tie is also what "normalization never matters" would predict, we build one semi-synthetic setting that separates the two: 13 real semiconductor predictors plus 19 appended irrelevant channels, which moves the coordinate to the single corner where the map predicts a win. Exact zeros win where representations separate sparse predictive signal from abundant nuisance directions. We test that boundary on 556 cryptocurrency settings under Definition 1 and on a 121-series US macroeconomic panel spanning from 1960 to 2025 (242 settings, one month ahead).

Contributions. *(1) Theoretical Risk Boundary:* We establish a sufficient condition for exact zeros to reduce excess risk. This occurs whenever support-estimation error drops below the softmax noise floor. We also derive a closed-form leak term. It predicts the gap between optima across twelve orders of magnitude. *(2) Tri-Mechanism Disentanglement:* Convex reference solutions show the sparse advantage persists without an optimizer. This confirms its representational origin. In 300 frozen forecasters, exact zeros cut nuisance propagation. Perturbing sparsemax-zeroed positions shifts predictions to 0.17 of what an equally large random set does, against 0.63 for dense softmax. *(3) Pre-Registered Decision Rule and Real-World Gains:* We introduce an a priori input coordinate estimated strictly from training data. It tracks the regime across the settings we test, before any test evaluation. Across 798 low-coordinate configurations, exact zeros lower test error in 773 cases (96.9%). Test error falls by 2.32% across the cryptocurrency cross-section, 12.73% along a size-matched time axis, and 15.69% across 121 macroeconomic series. Reported intervals remain robust to overlapping-window dependence. Finally, a pre-registered decision rule returned all four predicted wins on untouched test datasets.

2 Related Work

Architectural adaptations in time-series Transformers primarily target tokenization and context structuring rather than normalization. Informer downsamples queries via KL divergence (Zhou et al., 2021). PatchTST extracts subseries patches (Nie et al., 2023), while iTransformer treats entire variable channels as tokens (Liu et al., 2024). Other designs introduce seasonal-trend and frequency decompositions (Wu et al., 2021; 2023; Zhou et al., 2022). Crucially, these architectures retain dense softmax over tokens. Linear maps in feedforward and projection blocks thus carry the entire burden of noise suppression.

A separate line of work modifies probability simplex projections directly. Sparsemax projects real scores onto the simplex via Euclidean projection to induce exact zeros (Martins and Astudillo, 2016). Alpha-entmax generalizes this formulation into a continuous family between softmax ($\alpha = 1$) and sparsemax ($\alpha = 2$) (Peters et al., 2019; Correia, Niculae, and Martins, 2019). Structural sparsity has also been explored through hard-concrete gates (Louizos, Welling, and Kingma, 2018), fixed attention masks (Child et al., 2019; Zaheer et al., 2020), adversarial regularization (Wu et al., 2020), and selective state-space models (Gu and Dao, 2024). However, sparse weights do not guarantee functional feature pruning (Jain and Wallace, 2019; Meister et al., 2021).

Recent theoretical studies examine attention sharpness across sequence length. Softmax loses sharpness as context scales (Veličković et al., 2025), whereas entmax mitigates long-context collapse (Vasylenko et al., 2026). In parallel, Nawrot et al. (2026) map the computational efficiency frontier of structured sparsity. We address an orthogonal axis. Rather than scaling context length, we fix the context and formalize the data-distribution boundary (support sparsity, residual nuisance variance, and sample complexity) where exact-zero selection strictly reduces excess risk.

3 RISK BOUND FOR RESIDUAL INPUT NOISE

This section derives the phase boundary in closed form: bounded softmax scores leave a floor of residual input noise that exact zeros eliminate, while exact zeros pay a support-recovery cost that decays with sample size. Consider an observed context divided into G input groups, such as lags, patches, or variables. Let $S \subseteq [G]$ denote the relevant support that affects the target. Let $N = [G] \setminus S$ contain the irrelevant groups, with $r = |N|$. For a scalar target y, the squared-error risk is $\mathcal{R}(f) = \mathbb{E}[(f - y)^2]$.

Definition 1 (Normalized attention with residual noise)**.** *Let H_S be a representation of the relevant groups. We study predictors of the form*

$$\widehat{y} = h(H_S) + u^\top \sum_{j \in N} a_j z_j, \tag{1}$$

where $h(H_S) \in \mathbb{R}$ is any predictor built from H_S. The vector $u \in \mathbb{R}^d$ is a fixed readout mapping aggregated value noise into the prediction, capturing downstream sensitivity. The term $z_j \in \mathbb{R}^d$ is the remaining noise in value j, where d is the representation dimension. We assume $\mathbb{E}[z_j \mid H_S, y] = 0$ and $\mathrm{Cov}(z_j \mid H_S, y) = \sigma_z^2 I_d$, with I_d the $d \times d$ identity. The noise terms $\{z_j\}$ are conditionally independent given (H_S, y). The weights a_j are measurable with respect to H_S.

Measurability requires attention weights to share no estimation noise with z_j, which holds when keys come from separate channels.

The following theorem says that bounded softmax passes a non-zero noise floor into the prediction:

Theorem 1 (Risk from residual noise under bounded softmax)**.** *Assume Definition 1. Let $a = \mathrm{softmax}(s)$ for scores $s \in \mathbb{R}^G$ satisfying $|s_j| \le B$. Then*

$$\mathcal{R}(\widehat{y}) = \mathcal{R}(h) + \sigma_z^2 \|u\|_2^2 \mathbb{E}\left[\sum_{j \in N} a_j^2\right] \ge \mathcal{R}(h) + \frac{r\sigma_z^2 \|u\|_2^2}{G^2} e^{-4B}. \tag{2}$$

An exact-zero selector whose nonzero weights are restricted to S eliminates this noise term entirely.

Remark 2 (Correlated residual noise)**.** *Definition 1 extends directly to correlated residuals. Stack the residual noise of irrelevant groups as $z_N = (z_j)_{j \in N} \in \mathbb{R}^{rd}$. Suppose the cross term with $h(H_S) - y$ vanishes and $\mathrm{Cov}(z_N \mid H_S, y) \succeq \lambda I_{rd}$ for $\lambda > 0$. Then bounded softmax still satisfies*

$$\mathcal{R}(\widehat{y}) - \mathcal{R}(h) \ge \frac{r\lambda \|u\|_2^2}{G^2} e^{-4B}. \tag{3}$$

An exact-zero selector supported on S eliminates this term entirely.

We next analyze support estimation. Let $p(s) = \mathrm{sparsemax}(s)$ and $\mathcal{A}_S = \{s : \mathrm{supp}(p(s)) = S\}$, with the population score vector s^* strictly inside $\mathcal{A}_S$. The support margin is

$$\gamma = \inf_{s \notin \mathcal{A}_S} \|s - s^*\|_\infty > 0. \tag{4}$$

The following theorem says that sufficient data guarantees exact support recovery with high probability:

Theorem 3 (Support recovery from a finite sample)**.** *Suppose a learned score vector $\widehat{s}$ estimated from n samples satisfies*

$$\Pr(|\widehat{s}_j - s_j^*| \ge t) \le 2\exp\left(-\frac{nt^2}{2\nu^2}\right) \quad (j \in [G]), \tag{5}$$

where ν^2 is the sub-Gaussian variance proxy of each score coordinate. Then

$$\Pr\{\mathrm{supp}(p(\widehat{s})) \ne S\} \le 2G\exp\left(-\frac{n\gamma^2}{2\nu^2}\right). \tag{6}$$

Thus, $n \ge 2\nu^2\gamma^{-2}\log(2G/\delta)$ samples suffice to recover S with probability at least $1 - \delta$.

The following corollary says that exact zeros win whenever support-estimation error falls below that floor:

Corollary 4 (When learned exact-zero selection improves risk)**.** *Let E denote correct support recovery. Suppose the excess risk $\Delta_{\mathrm{sp}} = \mathcal{R}(\widehat{y}_{\mathrm{sp}}) - \mathcal{R}(h)$ satisfies $\Delta_{\mathrm{sp}} \le \epsilon_n$ on E and $\Delta_{\mathrm{sp}} \le M$ on E^c. Under Theorem 3,*

$$\mathbb{E}_{\mathrm{train}}[\Delta_{\mathrm{sp}}] \le \epsilon_n + 2GM \exp\left(-\frac{n\gamma^2}{2\nu^2}\right). \tag{7}$$

Fixing baseline h, exact-zero selection achieves strictly lower expected excess risk than bounded softmax whenever the right-hand side of Eq. (7) is strictly less than the noise floor in Eq. (2).

Scope and interpretation: Equation (2) applies whenever nuisance enters values ($\sigma_z^2 > 0$), reaches downstream layers ($\|u\|_2 > 0$), and scores remain bounded; it decays as e^{-4B}. Bounded softmax retains noise variance on every input, and exact-zero selection trades that variance for a finite-sample recovery cost.

4 Controlled Comparison of Softmax and Sparsemax

This section verifies the theory in controlled synthetic experiments: exact zeros outperform dense softmax across sparse settings, while dense attention is preferred only under extreme sample starvation. The design is a minimal model matching Definition 1. Each sample contains $G = 32$ independent scalar inputs. The support S contains k relevant inputs drawn from $\mathcal{N}(0, 1)$. Irrelevant inputs follow $x_j \sim \mathcal{N}(0, \sigma_z^2)$. The target is generated as

$$y = k^{-1} \sum_{j \in S} x_j + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 0.05^2). \tag{8}$$

The predictor is $\widehat{y} = p(s)^\top x$. We compare $p = \mathrm{softmax}$ with $p = \mathrm{sparsemax}$. Both models optimize identical 32-dimensional score vectors $s_j \in [-2, 2]$. Scores are initialized to zero and fit via L-BFGS-B under an identical protocol.

The experimental grid sweeps $k \in \{2, 4, 8, 16, 32\}$, $\sigma_z \in \{0.25, 0.5, 1, 2\}$, and $n \in \{128, 512, 2048\}$ across 20 random seeds (2,400 fits). Evaluation uses 20,000 test samples. We record excess MSE above irreducible noise σ_ε^2, attention leakage onto nuisance inputs, and support recovery F_1. The dense case $k = G = 32$ serves as the all-relevant reference.

Averaged over the 48 sparse settings within a seed, exact-zero selection achieves lower excess risk in all 20 seeds (one-sided Wilcoxon signed-rank $W = 210$, $p = 9.5 \times 10^{-7}$; Fig. A4). At $n = 512$ and $\sigma_z = 1$, the mean paired difference decreases monotonically from 0.02469 at $k/G = 1/16$ to 0.000068 at $k/G = 1/2$, reaching zero at $k/G = 1$.

Residual noise variance scales the performance gap as predicted. At $k = 4$, raising σ_z from 0.25 to 2 widens the mean difference from 0.00324 to 0.00505. Learned attention weights yield $\sum_{j \in N} a_j^2 \approx 4.62 \times 10^{-4}$, predicting an increase of 0.00182. This matches the observed 0.00180. Across all 48 sparse configurations, $r\sigma_z^2/G^2$ correlates positively with the observed difference (Spearman $\rho = 0.379$).

Sample size establishes the empirical recovery boundary. Sparsemax outperforms softmax in every sparse setting once $n \ge 512$, where support F_1 rises from 0.987 to 0.999. All eight configurations with negative mean differences sit at $n = 128$ with $k \in \{8, 16\}$. Three of them clear the decision thresholds below and form the only dense-preferred cells. This matches the estimation penalty quantified in Eq. (7).

Empirical Phase Map: Exact-zero selection dominates sparse configurations and matches dense ones, leaving dense softmax preferred only under acute sample starvation. Fig. 1 partitions all 60 configurations using the 20-seed mean difference Δ (softmax minus sparsemax). A win requires $\Delta > 2\,\mathrm{SE}$ and $\Delta > 10^{-4}$. A dense-preferred cell requires $\Delta < -2\,\mathrm{SE}$ and $\Delta < -10^{-4}$. Remaining cells are labeled ties, indicating unresolved differences at this precision or differences below the floor. Where the 90% interval lies within $\pm 10^{-4}$, the two models are equivalent at the floor. The fixed 10^{-4} margin yields 34 wins, 23 ties, and 3 dense-preferred cells. Sparsemax wins in all sparse-support cells ($k/G \le 1/8$), at $k/G = 1/4$ when $n \ge 512$, and at $k/G = 1/2$ under high noise

($\sigma_z = 2, n \geq 512$). It ties across all 12 dense control cells ($k = G$) and 11 intermediate cells. Dense attention is preferred only in the low-sample, high-noise corner ($n = 128$, $k \in \{8, 16\}$, $\sigma_z \geq 1$), where the sample is too small to identify the support, as Theorem 3 predicts.

The win region contracts as score bounds widen, concentrating on the sparsest, smallest-sample settings (Fig. 2a; Appendix Z). The dense-preferred cells remain identical across bounds at $(k, n) = (8, 128)$ and $(16, 128)$. Appendix M traces them to the optimization trajectory rather than the score range. Under AR(1) temporal dependence and heavy-tailed innovations (Appendix X.4), the phase structure persists. Strong autocorrelation ($\phi = 0.9$) shifts the boundary by reducing effective sample size.

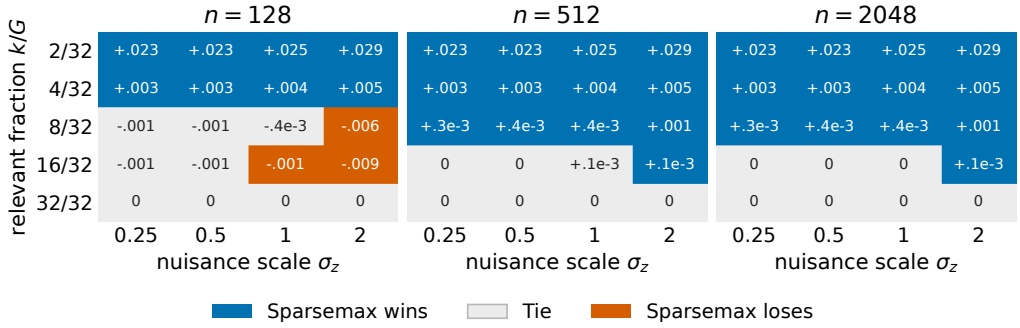

Figure 1: *Empirical phase map across 60 configurations (2,400 fits, bound $B = 2$).* Values display mean excess-MSE differences Δ (softmax minus sparsemax; leading zeros omitted). Blue indicates a sparsemax win ($\Delta > 2\,\mathrm{SE}$ and $\Delta > 10^{-4}$; 34 cells). Orange indicates a dense-preferred cell ($\Delta < -2\,\mathrm{SE}$ and $\Delta < -10^{-4}$; 3 cells). Gray denotes a tie (23 cells). Standard errors reflect 20 paired differences within seeds. The same cells at other bounds appear in Appendix Z.

Convex Reference Optima: The sparse advantage is an intrinsic representational property that persists without an optimizer. Because the predictor in Definition 1 is a convex quadratic in p, both models admit exact reference solutions independent of support knowledge. The reference for exact-zero selection is the minimizer over the whole simplex, which sparsemax can represent. The reference for bounded softmax is the minimizer over $\{p : p_i \geq e^{-2B} \max_j p_j\}$, the weight ratios a bounded score reaches. Solving both via SLSQP on identical samples isolates representation from optimization dynamics (Appendix M). Softmax fits match their constrained optimum within 10^{-17} training MSE at $B = 2$. Here, the two optima diverge by the leak the theorem predicts: at $(k, \sigma_z, n) = (2, 2, 128)$, the bounded-ratio optimum incurs excess MSE of 2.9×10^{-2} versus 2.1×10^{-4} for the simplex optimum.

Across all 60 cells, convex references yield 36 wins, 24 ties, and no dense-preferred cells at $B = 2$. They yield 60 ties at $B = 8$, where both optima match to 10^{-6}. Across these 60 cells, exact zeros never lag bounded softmax at the optimum, and the three dense-preferred cells of Fig. 1 do not appear in the reference comparison. Those cells belong to the fit. Because $\mathrm{sparsemax}(p) = p$ on the simplex and $[0, 1] \subset [-2, 2]$, the parametrization already spans the full simplex at $B = 2$, and initializing at the optimum's own score vector reaches it in all 20 seeds of every cell. In exchange, zero initialization acts as implicit regularization. It halts before the training optimum, reducing test excess risk in 52 of 60 cells and improving support recovery ($F_1 = 1.00$ versus 0.76), but it penalizes the smallest-sample cells (Appendix M).

Neural Phase Map: The empirical phase boundary transfers directly to trained deep neural networks. In the temporal-attention module of MISA (a BiGRU over $T = 32$ steps), sparsemax wins six of eight sparse cells and ties all $k = G$ cells (Table A23). The two exceptions occur at lowest noise ($k = 2$), where a trained readout rescales what bounded scores cannot. The boundary transfers to trained encoders whenever support is learnable and nuisance noise is material.

Evaluating the Closed-Form Risk Rule: The closed-form leak term accurately predicts risk regimes before evaluation. Bounded scores force a nuisance leak $L = r/(ke^{2B} + r)$, producing bias L^2/k and variance $\sigma_z^2 L^2/r$. Together with finite-sample terms, this expression predicts softmax excess risk within a median 0.6% error across 48 sparse cells. Using empirical failure rates and costs, it correctly reproduces 57 of 60 regimes with zero false wins (Fig. A3, Appendix G).

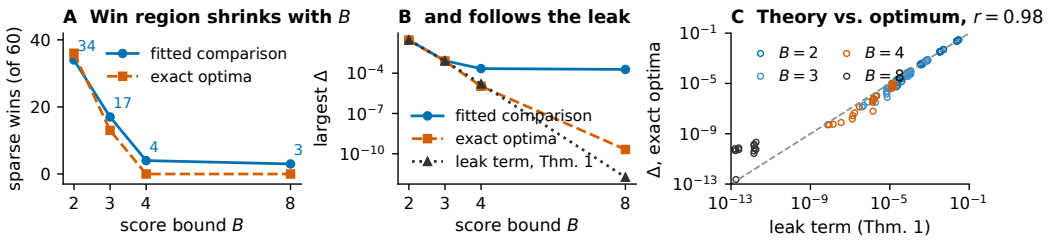

Figure 2: *Score bounds govern the gap without an optimizer.* (a) Win-region size versus score bound B for fitted models and exact convex references. (b) Maximum cell difference Δ versus B, for the fits, for the exact optima, and for the closed-form leak term $L^2/k + \sigma_z^2 L^2/r$ of Appendix G, where $L = r/(ke^{2B} + r)$. (c) Theoretical leak matches the exact optima gap across 130 configurations spanning twelve orders of magnitude ($r = 0.98$ in $\log_{10}$).

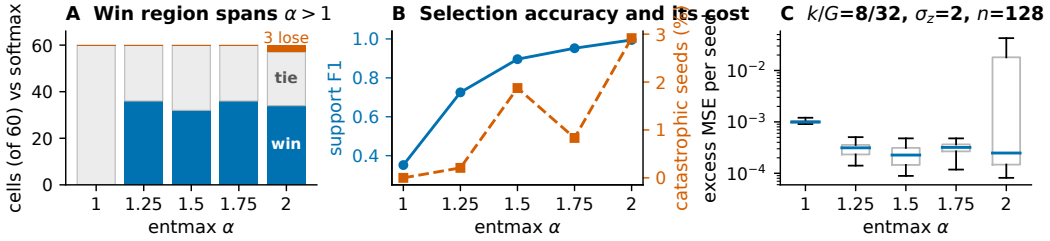

Figure 3: *Exact-zero attention across the entmax family (6,000 matched fits over five values of α).* (a) Regime counts against softmax across 60 configurations. (b) Support recovery F_1 and tail risk (fits exceeding ten times the cell median) across 48 sparse cells. (c) Per-seed excess MSE in the highest-variance configuration ($k = 8$, $\sigma_z = 2$, $n = 128$).

Continuous Interpolation via the Entmax Family: The exact-zero performance gain extends across the entire α-entmax family while interior values suppress tail risk. We evaluate 6,000 matched fits across $\alpha \in \{1.0, 1.25, 1.5, 1.75, 2.0\}$. Sparse formulations ($\alpha > 1$) maintain the win region across 32 to 36 cells (Fig. 3a). However, the endpoint and interior differ in tail risk. In the highest-variance cell ($k/G = 1/4$, $\sigma_z = 2$, $n = 128$), median excess error is comparable across $\alpha > 1$ (2.3×10^{-4} at $\alpha = 1.5$ versus 2.5×10^{-4} at $\alpha = 2$), yet sparsemax recovers an incorrect support in 6 of 20 seeds (Fig. 3c). The proportion of fits exceeding ten times the cell median error is 0.0% for softmax, 0.2% to 1.9% for interior α, and 2.9% for sparsemax (Fig. 3b). Mild exact sparsity ($\alpha = 1.25$) attains 36 wins, the largest region of the family, without a single dense-preferred cell.

5 MISA: A Sparse-Attention Probe for Real Data

This section introduces our observational test probe: a multi-scale forecaster designed solely to isolate whether exact zeros block residual input noise in real data, rather than to claim architectural novelty. MISA provides the empirical instrument for examining the phase boundary of Section 3. Given a target window $X \in \mathbb{R}^{T \times F}$ and an exogenous leader window, it produces multi-step forecasts in three stages.

First, average pooling at downsampling factors $q \in \{1, 3, 6\}$ produces three temporal resolutions. Each resolution is read by a bidirectional GRU (Cho et al., 2014) with 64 hidden units per direction, strictly masking future observations.

Second, per-scale temporal scores $s^{(q)}$ yield attention weights through sparsemax:

$$a^{(q)} = \arg\min_{p \in \Delta^{T_q - 1}} \|p - s^{(q)}\|_2^2, \qquad c^{(q)} = \sum_t a_t^{(q)} H_t^{(q)}, \tag{9}$$

where $\Delta^{T_q - 1}$ is the probability simplex, $T_q = T/q$, and $H_t^{(q)}$ is the recurrent hidden state. Sparsemax can set individual temporal weights to exact zero. A second sparsemax layer pools across the three scale contexts $c^{(q)}$, enabling exact exclusion of an entire resolution.

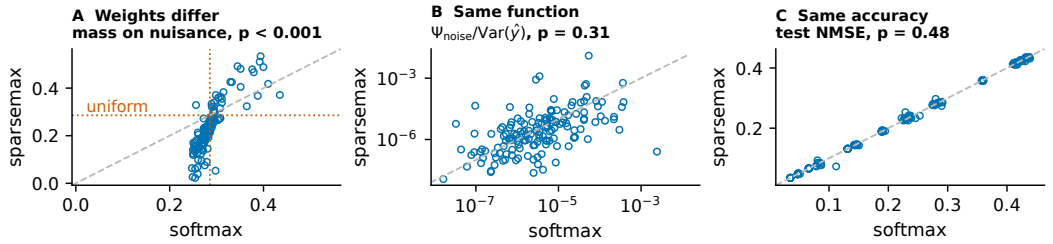

Figure 4: *Nuisance diagnostics on MISA* (15 settings $\times$ 10 seeds $\times$ 2 normalizations, $r = 12$ injected noise steps, $K = 16$ frozen resamples, 150 matched pairs; p from paired Wilcoxon tests). (a) Attention allocated to the 12 injected noise steps. (b) Functional sensitivity Ψ_{noise}. (c) Test NMSE. Sparsemax cuts noise attention but leaves functional sensitivity and test error equivalent to softmax.

Third, a separate BiGRU encodes the exogenous leader series, such as BTC. Four-head cross-attention reads these leader representations using the fused target context as the query. A sigmoid gate then mixes the target and cross-signal contexts (Appendix C). This gate is not an exact-zero selector under Theorem 1, since a sigmoid never reaches zero at finite logits.

A layer-normalized linear head outputs multi-step return forecasts. Appendix C details all specifications, equations, and hyperparameters. Sparsemax could operate equally well on Transformer or state-space backbones. MISA contains 228,866 parameters, which is 4.4 times the baseline LSTM, and claims neither computational efficiency nor parameter reduction.

6 Empirical Validation and Results

This section tests the theoretical phase boundary across observational benchmarks, asking one question per mechanism of Table 1: exact zeros tie when inputs are low-dimensional and dense, but dominate once representations isolate sparse predictive signals. The benchmark includes 1.2 million five-minute OHLCV bars across 47 Coinbase assets in three 30-day windows: P1 (FTX collapse), P2 (pre-ETF accumulation), and P3 (post-ETF shift). Excluding ARB and VET for short P1 history yields 139 settings. Each period uses a chronological 70/10/20 train/validation/test split. Normalization parameters are fit strictly on training splits. Inputs span 30 bars to forecast five future returns, with Bitcoin as the leader series. We also evaluate hourly forex pairs (EUR/USD, GBP/USD, JPY/USD), standard ETT benchmarks, and the 207-sensor METR-LA traffic network (Li et al., 2018).

Baselines. Baselines include recurrent models (LSTM, GRU), Transformers (iTransformer, PatchTST, Informer), continuous-time CfC networks (Hasani et al., 2022), and AR(1)–GARCH(1,1). We report Normalized Mean Squared Error (NMSE, normalized by the squared price range of the test split) across seeds 42 to 44. Setting-level comparisons use the Diebold–Mariano test (Diebold and Mariano, 1995; Harvey, Leybourne, and Newbold, 1997) with Harvey corrections on seed-42 paths. Asset-level aggregations rely on one-sided exact Wilcoxon tests. The bounds characterize attention with bounded scores over a linear readout rather than a single architecture. MISA remains competitive with all baselines (Appendix W), which qualifies it as an observational probe. Every comparison holds the model class fixed and alternates only the normalization function.

Where the Boundary Sits on Real Data: On ETTh1, ETTh2, ETTm1, ETTm2, and the 207-sensor METR-LA, the theory predicts a tie. A matched comparison within Definition 1 confirms this prediction ($p = 0.23$ across 95 paired cells; Appendix Y).

Matched Accuracy Evidence: Across 139 cryptocurrency settings, the two normalizations match closely. TOST equivalence holds at a $\pm 0.5\%$ margin ($p = 0.007$). A ten-seed replication across 15 representative settings yields a -0.10% mean gap (90% CI $[-0.30\%, 0.11\%]$). Seed variance exceeds the normalization gap in every setting (Fig. A1, Appendix X.1).

Mechanism Diagnostics: We prepend $r = 12$ pure noise steps to 15 representative settings, train across ten seeds, freeze parameters, and record $\Psi_{\text{noise}} = \mathbb{E}_x[\text{Var}_z\, f(x, z)]$ over $K = 16$ resampled paths. Across 150 pairs (Fig. 4), attention mass and functional sensitivity diverge. Sparsemax places a median weight of 0.225 on noise steps compared to 0.281 for softmax ($p = 2.7 \times 10^{-13}$). Both values sit near the uniform baseline of 0.286. Downstream sensitivity relative to prediction variance

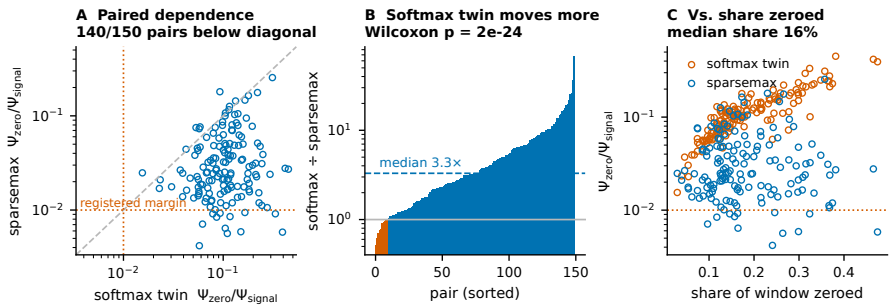

Figure 5: *Perturbation response of sparsemax-zeroed positions* (150 pairs, identical positions in both twins). (a) Ratio of zeroed-position variance to full context variance; dotted lines mark the registered margin. (b) Per-pair softmax to sparsemax sensitivity ratio, sorted (median 3.3×). (c) Perturbation shift plotted against the zeroed window fraction.

is 2.5×10^{-6} for sparsemax against 3.3×10^{-6} for softmax ($p = 0.31$), yielding identical test NMSE ($p = 0.48$). The test has power: permuting the true context shifts the forecast by essentially its full variance.

Perturbing Sparsemax-Zeroed Coordinates: We resample only the coordinates that sparsemax sets to zero, covering a median of 16% of the window, with a pre-registered margin of 0.01 (Appendix N). A size-matched control makes the reading independent of how many coordinates were zeroed. Perturbing these zeroed coordinates shifts the prediction by only 0.17 of what perturbing an equal number of random coordinates does (149 of 150 pairs). In contrast, the softmax twin evaluated at identical positions reaches 0.63 ($p = 8 \times 10^{-25}$; Fig. 5). Path ablations confirm that the recomputed recurrent representations, not weight movement, carry the remaining sensitivity (Appendix N).

Invariance to Network Backbone: Replacing the encoder with PatchTST (Nie et al., 2023), which uses unbounded logits, yields mean test NMSE of 0.2334, 0.2332, and 0.2334 over the same 15 settings ($p = 0.73$; Appendix Q).

Prospective Semiconductor Supply-Chain Benchmark: We evaluated the phase map prospectively on a semiconductor supply chain, registering all predictions on 2026-07-26 prior to running the experiments (Appendix P). The falsification criterion specified that a significant sparsemax win would refute the tie prediction. The tie held on a 384-month production panel ($p = 0.58$) and across a resolution ladder of 1-minute, 5-minute, and daily equity panels. Because training inputs span few independent directions and most carry signal, the coordinate $k_{\text{eff}}/r_{\text{eff}}$ places all eight datasets in the all-relevant row, where all eight tied (Appendix O).

Semi-Synthetic Disentanglement: The semi-synthetic setup of Section 1 appends 19 irrelevant noise channels to 13 real semiconductor predictors at $n = 128$. This shifts the training coordinate to $2/32$, where the phase map predicts a win (Appendix I). Both registered win predictions held ($\Delta = +0.060$ and $+0.063$ standardized test MSE; sparsemax lower in 20 and 18 of 20 draws). Sparsemax zeroed all 19 noise channels. Without appended noise, the models produced the predicted tie. At $\sigma_z = 0.25$ ($+0.055$) the map called a tie and a win was observed, so its σ_z axis is conservative at the wide bound $B = 8$; that reading is carried forward as a registered prediction for the next run.

The Coordinate Dictates the Regime: The tie occurs because the multi-scale encoder of MISA maps inputs into few independent directions that nearly all carry signal ($k_{\text{eff}}/r_{\text{eff}} \in [0.50, 1.00]$). The theory does not predict that market data ties; it predicts that this coordinate ties. We altered the coordinate twice under Definition 1 ($|s| \leq 8$, 20 rolling origins, 500 test bars), using pre-registered cells, decision rules and falsifiers (Appendix K). We tested attention over 44 concurrent assets (coordinate 0.07 to 0.45) and a size-matched temporal control over past bars (coordinate 0.02 to 0.26).

Exact zeros win in both configurations, and no configuration in either run meets the registered criterion for a dense win. Across the cross-section, sparsemax yields lower test error in 269 of the 278 cells, which span every asset, window, and training size, improving error by 2.32% on average (90% CI $[1.81\%, 2.78\%]$ via moving-block bootstrap). Gains remain steady across all three market windows ($+2.04\%$, $+3.24\%$, $+1.67\%$). In the size-matched temporal control, sparsemax wins in 276

of 278 cells, reducing error by 12.73% ($[11.18\%, 13.93\%]$). The shared low coordinate, not the data source or the axis of attention, separates these runs from the tie.

Domain Transfer on US Macroeconomic Data: If the input coordinate drives the outcome, exact zeros must dominate in any low-coordinate domain. We evaluated FRED-MD, spanning 785 months from 1960 to 2025 across 121 macroeconomic series transformed by the codes shipped with the file. Each series is predicted one month ahead from the remaining 120 series across 20 rolling origins at $n \in \{128, 256\}$. These input channels span only 19 to 23 independent directions, and all 242 cells are reported (Appendix J).

Exact zeros win here by an order of magnitude more than in the market panel. Sparsemax achieves lower test error in 228 of 242 cells and in 3,929 of 4,840 origins, averaging a 15.69% improvement (90% CI $[13.98\%, 17.31\%]$ via moving-block bootstrap). The gain reaches $+20.74\%$ at $n = 128$ and $+10.63\%$ at $n = 256$, which we report as an observation. Using the historical training mean as an absolute baseline, scored per origin rather than assumed (median test MSE 0.996), sparsemax tracks closer than softmax (1.045 versus 1.176). Sparsemax outperforms this mean baseline on 42.2% of origins, compared to 31.1% for softmax. Across the 1,350 origins where both models beat the mean, the sparse advantage remains $+14.77\%$ (Appendix J).

Table 2: *Evaluation across 937 cells and four settings.* Rows 2 to 4 use the predictor of Definition 1, and the two cryptocurrency rows also share one protocol, differing only in what the attention weights. The top row uses the multi-scale encoder of MISA.

Attention Domain	Predictor	Cells	$k_{\text{eff}}/r_{\text{eff}}$	Relative Gain	Sparsemax Lower
Past bars, multi-scale	MISA encoder	139	0.50–1.00	tie	61/139
Past bars, size-matched	Definition 1	278	0.02–0.26	$+12.73\%$	276/278
Crypto cross-section	Definition 1	278	0.07–0.45	$+2.32\%$	269/278
US Macroeconomic panel	Definition 1	242	0.04–0.47	$+\mathbf{15.69}\%$	*228/242*

Across two domains and 798 unselected cells (Table 2), exact zeros reduce test error in 773 cases. The training-only coordinate separates these three winning configurations from the tie.

Confirmatory Validation on Frozen Rules: The decision rule was frozen and registered earlier with 12 configurations, comparators and acceptance criteria on two sources not used to choose it (Appendix L). All four predicted wins materialized under both decision thresholds. Across these 12 evaluation cells, the rule achieved the lowest mean test error among five competing strategies.

7 Discussion and Conclusion

This work resolves when and why exact zeros outperform dense softmax in sequential prediction. The bounds provide a conditional risk comparison rather than an architectural ranking. The sparse advantage originates from representation rather than optimization, because bounded scores retain persistent variance on irrelevant inputs. The optimizer and the downstream network then determine where that advantage is realized (Table 1). Across 798 low-coordinate real-world settings, exact zeros reduce test error in 773 cases (96.9%), while the high-coordinate setting ties. We report this finding as an empirical association rather than a universal rule.

We note two methodological boundaries. First, finite samples penalize support recovery in high-noise regimes. As Fig. 3 and Appendix AA show, relaxing sparsemax to mild entmax ($\alpha = 1.25$) provides an effective remedy: it attains the largest win region of the family, 36 cells, while lowering tail risk from 2.9% to 0.2%. Second, the theoretical bound assumes a linear readout on residual noise. Deep downstream layers can still transmit part of the nuisance features. Nevertheless, sparsemax cuts this nuisance transmission to 0.17 compared to 0.63 for its softmax twin (Appendix N).

In conclusion, exact zeros provide a principled mechanism to eliminate nuisance variance whenever representations isolate sparse predictive signals. By establishing a closed-form risk boundary and an a priori training coordinate, this framework replaces empirical guesswork with a diagnostic decision rule for attention normalization.

AI USE STATEMENT

Generative AI tools were used only for grammar and English editing.

ETHICS STATEMENT

This work uses only publicly available market and benchmark data, involves no human subjects or personal data, and makes no trading or investment recommendation.

REPRODUCIBILITY STATEMENT

Code, data-processing scripts and result files are provided at https://anonymous.4open.science/r/sparseAttention-9249.

REFERENCES

Child, R.; Gray, S.; Radford, A.; and Sutskever, I. 2019. Generating Long Sequences with Sparse Transformers. *arXiv preprint arXiv:1904.10509*.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of EMNLP*, 1724–1734.

Correia, G. M.; Niculae, V.; and Martins, A. F. T. 2019. Adaptively Sparse Transformers. In *Proceedings of EMNLP-IJCNLP*, 2174–2184.

Diebold, F. X.; and Mariano, R. S. 1995. Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, 13(3): 253–263.

Gu, A.; and Dao, T. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *Conference on Language Modeling*.

Gu, S.; Kelly, B.; and Xiu, D. 2020. Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies*, 33(5): 2223–2273.

Harvey, D.; Leybourne, S.; and Newbold, P. 1997. Testing the Equality of Prediction Mean Squared Errors. *International Journal of Forecasting*, 13(2): 281–291.

Hasani, R.; Lechner, M.; Amini, A.; Liebenwein, L.; Ray, A.; Tschaikowski, M.; Teschl, G.; and Rus, D. 2022. Closed-form Continuous-time Neural Networks. *Nature Machine Intelligence*, 4(11): 992–1003.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Jain, S.; and Wallace, B. C. 2019. Attention is not Explanation. In *Proceedings of NAACL-HLT*, 3543–3556.

Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.-H.; and Choo, J. 2022. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *International Conference on Learning Representations*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

Lim, B.; Arik, S. O.; Loeff, N.; and Pfister, T. 2021. Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting. *International Journal of Forecasting*, 37(4): 1748–1764.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*.

Louizos, C.; Welling, M.; and Kingma, D. P. 2018. Learning Sparse Neural Networks through L_0 Regularization. In *International Conference on Learning Representations*.

Martins, A. F. T.; and Astudillo, R. F. 2016. From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. In *International Conference on Machine Learning*, 1614–1623.

Meister, C.; Lazov, S.; Augenstein, I.; and Cotterell, R. 2021. Is Sparse Attention More Interpretable? In *Proceedings of ACL-IJCNLP*, 122–129.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers. In *International Conference on Learning Representations*.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2020. N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting. In *International Conference on Learning Representations*.

Peters, B.; Niculae, V.; and Martins, A. F. T. 2019. Sparse Sequence-to-Sequence Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1504–1519.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Vershynin, R. 2018. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press.

Wainwright, M. J. 2009. Sharp Thresholds for High-Dimensional and Noisy Sparsity Recovery Using ℓ_1-Constrained Quadratic Programming (Lasso). *IEEE Transactions on Information Theory*, 55(5): 2183–2202.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*, 22419–22430.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; and Ahmed, A. 2020. Big Bird: Transformers for Longer Sequences. In *Advances in Neural Information Processing Systems*, 17283–17297.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11121–11128.

Zhao, P.; and Yu, B. 2006. On Model Selection Consistency of Lasso. *Journal of Machine Learning Research*, 7: 2541–2563.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11106–11115.

Vasylenko, P.; Pitorro, H.; Martins, A. F. T.; and Treviso, M. 2026. Long-Context Generalization with Sparse Attention. In *International Conference on Learning Representations*.

Veličković, P.; Perivolaropoulos, C.; Barbero, F.; and Pascanu, R. 2025. Softmax Is Not Enough (for Sharp Size Generalisation). In *Proceedings of the 42nd International Conference on Machine Learning*.

Wu, S.; Xiao, X.; Ding, Q.; Zhao, P.; Wei, Y.; and Huang, J. 2020. Adversarial Sparse Transformer for Time Series Forecasting. In *Advances in Neural Information Processing Systems*, 17105–17115.

Nawrot, P.; Li, R.; Huang, R.; Ruder, S.; Marchisio, K.; and Ponti, E. M. 2026. The Sparse Frontier: Sparse Attention Trade-offs in Transformer LLMs. In *Findings of the Association for Computational Linguistics: ACL 2026*.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency Enhanced Decomposed Transformer for Long-Term Series Forecasting. In *International Conference on Machine Learning*, 27268–27286.

WHERE THE EVIDENCE IS

Claim	Appendix
What a bounded score can represent	exact convex references (M); regime counts across score bounds (Z); decision-rule constants (G)
What the optimizer returns	initialization probe, Table A12 (M); extended grid (AA)
What the network does with a zero	perturbing the zeroed positions, the size-matched control and the path interventions (N); injected-noise probe (X.3); normalization swap in PatchTST (Q)
Registered predictions and their outcomes	semiconductor (P); semi-synthetic (I); confirmatory test of the frozen rule (L); cross-section and the size-matched time-axis control (K); FRED-MD (J)
Where exact zeros lower error on real data	cross-section and matched time axis (K); macro panel (J); the training-mean reference and the dependence-robust intervals (J, K)
Benchmark detail and statistical scope	probe-forecaster detail (W); matched comparison (X.1); statistical scope (AB)

A FREQUENTLY ASKED QUESTIONS

Terms used in this appendix. A *cell* is one data setting: a dataset, a number of appended irrelevant channels, and a training-sample size. Δ is the test error of softmax minus that of sparsemax, averaged over the repeated fits of a cell, so $\Delta > 0$ favors sparsemax. 2 SE is twice the standard error of Δ across those fits. B is the bound placed on attention scores. The *two exact references* are the best weight vector on the simplex, which is what an exact-zero normalization can represent, and the best weight vector whose largest-to- smallest ratio stays within what a score bounded by B allows, which is what softmax can represent. A *sensitivity ratio* is how much a frozen model's prediction moves when only some inputs are resampled, divided by how much it moves when the whole context is resampled.

Scope and framing.

- *Q1: Is this a new-model paper?*
  No. MISA is the probe. The contribution is the risk comparison, the phase map, and the tests that tie real data to the mechanism.

- *Q2: What is "the boundary"?*
  The conditions that separate the three regimes: support sparsity, residual noise, and sample size (Fig. 1). The size of the win region also depends on B; the small-sample corner does not.

- *Q3: When is a cell a win, a tie, or dense-preferred?*
  It wins for sparse, recoverable supports. It ties when all inputs are relevant. It cedes to dense attention only where data are scarce and noisy. The theory gives a sufficient condition for the win, not an exact boundary for arbitrary networks.

- *Q4: Do the results survive real time-series effects?*
  Yes. First-order autocorrelation and heavy tails leave the structure unchanged; strong dependence shrinks the effective sample size to about 27 at $\phi = 0.9$ (Appendix X.4). A trained recurrent core reproduces the map wherever the support is learnable (Table A23).

- *Q5: Why do the two normalizations tie on real cryptocurrency data?*
  The map predicts a tie there, and the matched comparison returns one. The frozen-model test says why: the encoder and the readout absorb the nuisance, and neither normalization is sensitive to it ($p = 0.31$).

- *Q6: Only three seeds?*
  Main tables use three. The matched comparison was rerun with ten seeds on 15 settings ($p = 0.85$). In all 15 the mean gap lies inside seed-to-seed variation.

- *Q7: Is a one-percent error difference meaningful?*
  In return forecasting, yes (Gu, Kelly, and Xiu, 2020). We make no trading-profit claim. The matched swap isolates the normalization's share: 0.13% in the eight-setting ablation, -0.17% across 139 settings.

- *Q8: What is new here?*
  The question is organized along a data axis — support sparsity, residual nuisance, sample size — at fixed context length, and three mechanisms behind the effect are separated.
- *Q9: Is the tie just a null result?*
  No. It was predicted before it was measured. The asset-level interval lies inside the $\pm 0.5\%$ practical margin. The $\pm 0.25\%$ margin is the prospective target.
- *Q10: What would falsify the main claim?*
  A dataset where the pre-specified interval contradicts the predicted regime, or a frozen-model test showing a large normalization-dependent sensitivity where the mechanism predicts none. Unresolved outcomes are reported, not redefined.

Representation versus optimization.

- *Q11: Could the win region be an artifact of the fitting procedure?*
  The two exact references are solved directly in weight space, so the score parametrization and its fitting path are not part of them. The same rule applied to those solutions gives 36 wins, 24 ties and no dense-preferred cells at $B = 2$, and 60 ties at $B = 8$ (Appendix M). The fitting path is therefore not the only explanation available.
- *Q12: Does the score bound limit what sparsemax can represent?*
  Not at $B \geq 1$. Sparsemax returns any probability vector unchanged, and such a vector has entries in $[0, 1]$. So $B = 2$ already spans every weight vector on the simplex. The bound restricts the softmax side only.
- *Q13: Then why does the fit not reach the simplex reference?*
  Where the optimizer starts matters. Started at the reference solution's own scores, the same fit reaches it in 20 of 20 seeds in each of the six probed cells, at both bounds (Table A12). Within the tested budget of 20,000 iterations, a longer run from the zero start does not close the gap.
- *Q14: Are the dense-preferred cells real?*
  They are real for this training procedure. They do not appear in the reference comparison. We report them as a property of the optimization path, not of the model classes.
- *Q15: Is "exact zeros never fall behind at the optimum" implied by one class containing the other?*
  No. Containment orders the best training loss. It does not order two empirical solutions in test error. Our statement is an observation over the 60 cells.

Thresholds, registration and statistics.

- *Q16: Why was the decision threshold of the confirmatory test changed, and when?*
  The original threshold was an absolute one carried over from the controlled grid. These cells differ fivefold in error scale, so it was 0.006–0.03% of the quantity compared. Amendment 1 adds a margin of 0.5% of each cell's own error. It was added after the outcome table under the original threshold had been produced and before that table was interpreted. Both thresholds are reported, here and in Appendix L.
- *Q17: Doesn't a wider margin flatter the rule?*
  On the tie side it does. That is why both counts are given, 7 of 12 and 10 of 12, and why the claim rests on the win side, which is 4 of 4 either way.
- *Q18: One predicted tie came back a win. Which cell?*
  The semiconductor source with no appended channels at the smallest sample. With no channels appended the repeated fits are identical, so the standard error is zero. Under the relative margin the cell is a tie.
- *Q19: Why not drop those cells?*
  Because the exclusion would be decided after the outcome was known. They stay in the table, and the relative margin is what makes the rule defined on them.
- *Q20: Is the 0.17% gain over always-sparsemax meaningful?*
  It is the smallest of the four comparisons. Across the 240 fits its 90% interval is $[0.12\%, 0.22\%]$. Across the 12 task configurations the interval touches zero. The configurations come from two datasets, so the two kinds of uncertainty are reported separately.
- *Q21: Is the gain over the temperature-tuned softmax consistent?*
  In mean, yes, 1.31%. Per fit, no: the rule is lower in 90 of 240 fits, and a Wilcoxon signed-rank test on the paired differences gives $p = 0.69$. Against entmax 1.5 the rule is lower in 188 of 240 fits.

Functional interventions.

- *Q22: The zeroed-position test did not meet its registered criterion. What survives?*
  The criterion asked for a whole 90% interval below 0.01; the median is 0.025 and the interval is $[0.009, 0.112]$. What survives is the paired comparison at identical positions: the sparsemax model is less sensitive than its softmax twin in 140 of 150 pairs, and the median of the paired ratio is 3.3.

- *Q23: What does the size-matched control add?*
  It removes the effect of how many positions were zeroed. Against an equal number of random positions, sparsemax reaches 0.174 and its softmax twin 0.631; sparsemax is below one in 149 of 150 pairs. It is a follow-up analysis, not part of the registered test, and is reported separately from it.

- *Q24: Where does the remaining sensitivity enter?*
  Residual sensitivity persists when the attention weights are frozen: the group median is 0.0257 against 0.0250 end-to-end. Weight movement is therefore not needed for the dependence to persist, though it may still contribute (paired, the frozen condition is lower in 82 of 150 pairs, median reduction 5.4%). Removing the coarser scales lowers the paired ratio by a median 26.0%. Doing both brings the group median to 0.0099, with a 90% interval of $[0.0024, 0.0662]$ that does not meet the registered criterion (Table A14). The result supports propagation through the recomputed representations.

- *Q25: Which claims were registered before the experiment that tests them?*
  Five. The semiconductor predictions and their falsifier (`PREREGISTRATION_daily.md`, 2026-07-26). The semi-synthetic construction, its four predictions and their falsifiers (`PREREGISTRATION_semisynthetic.md`, 2026-09-19). The frozen rule, its 12 configurations, its comparators, its three acceptance criteria and its four falsifiers (`PREREGISTRATION_confirmatory.md`, 2026-09-19). The cross-sectional construction and the size-matched control, with 278 cells each (`PREREGISTRATION_crosssection.md`, 2026-09-20). The FRED-MD construction with 242 cells (`PREREGISTRATION_fredmd.md`, 2026-09-20). Each file also records the amendments made to it and the outcome as observed.

- *Q26: A registered tie came back a win in the semi-synthetic test. How is that handled?*
  It is reported as a mismatch. The explanation that follows is labeled after the fact and entered in the registration file as a prediction for the next run, with falsifiers.

Scope of the conclusions.

- *Q27a: You found only ties on real data for most of the paper. What changed?*
  The coordinate of the inputs, not the dataset and not the axis. With the predictor and protocol held fixed at Definition 1, exact zeros win both across the cross-section of assets ($+2.32\%$, 269 of 278 cells) and along a size-matched time axis ($+12.73\%$, 276 of 278). The setting that ties is the multi-scale encoder, a different representation, whose coordinate is 0.50 to 1.00 rather than below 0.5.

- *Q27b: Isn't picking the cross-section just finding a setting where you win?*
  The construction, the screening and the decision rule were registered first, and every asset in every window is a target at both sample sizes: 278 cells, none chosen by us, all reported. The rule also predicted almost no wins in the crisis window and almost nothing but wins after the ETF, and the observed win rates follow that ordering. The size-matched control settles it: on the same 278 targets, keeping the time axis and holding everything else fixed, exact zeros win as well, so the result is not a property of the direction we chose.

- *Q27c: Did the frozen rule work there?*
  The coordinate transferred: it reads the multi-scale encoder at 0.50 to 1.00 and every Definition 1 setting below 0.5, and the outcomes follow that split. Its numerical cut at 0.15 was calibrated on the encoder, so within a single low-coordinate setting the per-cell contrasts are smaller than the design resolves, and we report both the observed value and the threshold it misses.

- *Q27d: Why report the per-cell comparisons at all?*
  Because registration asks for every number, resolved or not. The result the design does resolve is the strongest real-data evidence in the paper: 269 of 278 cells favor exact zeros, none meets the registered criterion for a dense win, and no falsifier fired.

- *Q27e: All eight real datasets fall in the same row of the map. What does that test?*
  Only the tie prediction. A rule that always predicted a tie would score the same there, so we report it as a consistency check. The win prediction is tested where the coordinate is low, on the 798 cells of Appendices K and J.

- *Q28: Are twenty repeated fits and twelve cells enough?*
  They are enough to detect the advantage in the four configurations where the rule predicts one: there

the effects are two to five times their standard errors. They are not enough to establish the size of the selection gain over always-sparsemax, which is why the interval across configurations is reported next to the interval across fits.

- *Q29: Why is the controlled study a linear predictor?*
  Because it isolates the normalization. The phase-map structure is then checked in a trained recurrent model (Table A23). The Transformer experiment is narrower: it swaps the normalization inside PatchTST on the real task and compares accuracy, without reconstructing the map.
- *Q30: What is the most falsifiable claim?*
  The frozen rule predicts a sparsemax advantage in the registered configurations with $k_{\text{eff}}/r_{\text{eff}} \leq 0.15$ and $n \leq 256$. Failure under the pre-registered decision criteria would challenge that prediction within its stated scope. Separately, the convex-reference experiments test whether the representational advantage contracts as the score bound increases.

B PROSPECTIVE ACCURACY-AND-MECHANISM PROTOCOL

The protocol below is intended for new datasets or untouched confirmatory periods. The practical margin must be chosen before the confirmatory outcome is inspected. Nuisance resampling requires a perturbation known by construction to be target-irrelevant; Ψ_f measures sensitivity to that perturbation, not to every possible nuisance.

Algorithm 1 Prospective accuracy-and-mechanism test for exact-zero attention

Require: matched sparse/dense forecaster; held-out contexts; known nuisance generator; $K \geq 2$ nuisance resamples; pre-specified practical margin δ_{eq}

1: Train sparsemax and softmax under the same data split, optimization budget, and seeds; save paired held-out prediction paths.
2: Freeze each trained model. For every held-out useful context x, resample only the known nuisance $z^{(1:K)}$ and estimate $\Psi_f = \mathbb{E}_x[\mathrm{Var}_k\{f(x, z^{(k)})\}]$.
3: Compute $d = (\mathrm{MSE}_{\mathrm{soft}} - \mathrm{MSE}_{\mathrm{sparse}})/\mathrm{MSE}_{\mathrm{soft}}$ and a paired uncertainty interval $[\ell, u]$ using synchronized time blocks and training seeds.
4: **if** $[\ell, u] \subseteq [-\delta_{\mathrm{eq}}, \delta_{\mathrm{eq}}]$ **then**
5: **return** EQUIVALENT-FOR-ACCURACY; use Ψ_f only to interpret nuisance sensitivity.
6: **else if** $\ell > \delta_{\mathrm{eq}}$ **then**
7: **return** SPARSE-WIN.
8: **else if** $u < -\delta_{\mathrm{eq}}$ **then**
9: **return** DENSE-PREFERRED.
10: **else**
11: **return** UNRESOLVED.
12: **end if**
13: Attention mass on nuisance positions may be reported descriptively, but it is not by itself evidence of functional exclusion.

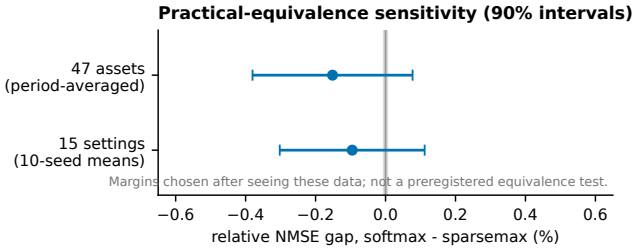

Figure A1: **Practical-equivalence sensitivity.** Point estimates and 90% intervals for the matched sparsemax–softmax relative NMSE gap. The shaded $\pm 0.5\%$ region is a retrospective sensitivity margin, not a preregistered threshold. Negative values favor softmax.

C MISA DETAILS

Cross-signal gate. With f the fused target context and $\widetilde{f}$ the cross-attention output,

$$g = \sigma\left(w_g^\top [f; \widetilde{f}] + b_g\right), \qquad o = (1-g)f + g\widetilde{f},$$

and a term $\lambda_g\, g(1-g)$ with $\lambda_g = 0.05$ penalizes intermediate gate values. A sigmoid does not reach zero at a finite logit, so g is not an exact-zero selector in the sense of Theorem 1. Figure A2 draws the three stages together.

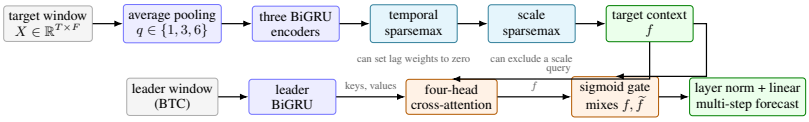

Figure A2: **MISA architecture.** The target window is encoded at three temporal scales. Sparsemax first selects time steps and then combines scales. The leader branch enters through cross-attention and a learned sigmoid gate. The sigmoid gate can suppress the leader signal, but it is not the exact-zero selector analyzed in Theorem 1.

Second, each scale produces temporal scores $s^{(q)}$. Sparsemax converts these scores into weights:

$$a^{(q)} = \arg\min_{p \in \Delta^{T_q-1}} \|p - s^{(q)}\|_2^2, \qquad c^{(q)} = \sum_t a_t^{(q)} H_t^{(q)}. \tag{10}$$

Here, Δ^{T_q-1} is the probability simplex, $T_q = T/q$ is the window length at scale q, and $H_t^{(q)}$ is the GRU state at time t and scale q. Sparsemax can assign zero weight to time steps. A second sparsemax combines the three context vectors $c^{(q)}$ and can exclude an entire scale.

Third, another bidirectional GRU encodes the leader series. Four-head cross-attention uses the fused target representation as its query and the leader states as keys and values. A sigmoid gate $g \in (0,1)$ mixes the target representation f and the cross-signal representation $\widetilde{f}$:

$$g = \sigma\left(w_g^\top [f; \widetilde{f}] + b_g\right), \qquad o = (1-g)f + g\widetilde{f}. \tag{11}$$

The training loss adds $\lambda_g g(1-g)$, with $\lambda_g = 0.05$, to encourage the gate toward either endpoint. This gate can strongly suppress the leader signal, but a sigmoid does not produce an exact zero at a finite logit. We do not identify the gate with the exact support selector in Theorem 1.

A layer-normalized linear head outputs the forecast; for cryptocurrency, the next five returns. Training uses Adam (Kingma and Ba, 2015) (learning rate 5×10^{-4}), batch size 64, dropout 0.25, gradient clipping at 1.0, 30 warm-up epochs, cosine decay, at most 200 epochs, and early stopping after 25 stagnant epochs. Learned weights select inputs; they do not establish causation. Table A1 lists every hyperparameter. One setting of them is used for all real-data experiments.

Table A1: MISA hyperparameters (identical across all real-data settings).

Input window / forecast horizon	30 bars / 5 bars (crypto); 12 / {3,6,12} steps (traffic)
Pooling factors q	$\{1, 3, 6\}$
BiGRU hidden size	64 per direction, one encoder per scale, one for the leader
Temporal and scale normalization	sparsemax (softmax in matched ablation)
Cross-attention	4 heads, query = fused target context, keys/values = leader states
Gate	sigmoid, regularizer $\lambda_g g(1-g)$, $\lambda_g = 0.05$
Head	layer norm + linear
Optimizer	Adam, lr 5×10^{-4}, batch 64, gradient clip 1.0
Schedule	30 warm-up epochs, cosine decay, $\leq$ 200 epochs, early stop after 25
Dropout	0.25
Seeds	42, 43, 44 (main tables); 42–51 (ten-seed rerun)
Parameters	228,866 (MISA), 52,033 (LSTM), 39,041 (GRU)

Why use a recurrent backbone? The recurrent backbone produces a sequential state before the model selects inputs. It is a design choice, not a requirement of Theorem 1. Sparsemax could also ride a convolutional, state-space, or Transformer backbone. Recurrence alone does not remove irrelevant variation. MISA is one way to pair recurrent encoding with explicit selection.

Computation. Scoring and weighted sums are linear in the number of states. The GRU encoders dominate the cost at $O(Td_h(d_h + F))$ for hidden size d_h. MISA has 228,866 parameters versus 52,033 for the long short-term memory (LSTM) model and 39,041 for the GRU. One CPU epoch takes 6.43, 2.84, and 2.02 seconds, respectively. We claim neither a size nor a runtime advantage.

D PROOFS

This section gives complete proofs. Each proof is broken into small steps. Every step states which assumption it uses and why it holds.

PROOF OF THEOREM 1

Bounded scores cannot reach zero. Theorem 1 says that every input keeps a floor of weight, so whatever noise survives in the irrelevant values reaches the prediction. An exact-zero selector passes none of it.

Proof. Write the prediction as $\widehat{y} = h(H_S) + \eta$, where

$$\eta = u^\top \sum_{j \in N} a_j z_j$$

collects the contribution of the irrelevant groups. The proof has four steps.

Step 1: decompose the risk. Expand the squared error around $h(H_S)$:

$$\begin{aligned}\mathcal{R}(\widehat{y}) &= \mathbb{E}\big[(h(H_S) + \eta - y)^2\big] \\ &= \mathbb{E}\big[(h(H_S) - y)^2\big] + 2\,\mathbb{E}\big[(h(H_S) - y)\,\eta\big] + \mathbb{E}[\eta^2] \\ &= \mathcal{R}(h) + 2\,\mathbb{E}\big[(h(H_S) - y)\,\eta\big] + \mathbb{E}[\eta^2].\end{aligned}$$

The risk of $\widehat{y}$ is the risk of h, plus a cross term, plus the variance of the noise contribution.

Step 2: the cross term vanishes. Condition on (H_S, y) and use the tower property:

$$\mathbb{E}\big[(h(H_S) - y)\,\eta\big] = \mathbb{E}\Big[(h(H_S) - y)\,\mathbb{E}[\eta \mid H_S, y]\Big].$$

Given (H_S, y), the factor $h(H_S) - y$ is fixed. The weights a_j are measurable with respect to H_S, so they are fixed as well. Definition 1 gives $\mathbb{E}[z_j \mid H_S, y] = 0$ for every j. Hence $\mathbb{E}[\eta \mid H_S, y] = u^\top \sum_{j \in N} a_j\,\mathbb{E}[z_j \mid H_S, y] = 0$, and the cross term is zero.

Step 3: compute the variance term. Condition on (H_S, y) again, so each a_j is a constant. Expand the square:

$$\mathbb{E}[\eta^2 \mid H_S, y] = \sum_{j \in N} \sum_{k \in N} a_j a_k\,\mathbb{E}\big[(u^\top z_j)(u^\top z_k) \mid H_S, y\big].$$

For $j \neq k$, the z_j are conditionally independent with zero conditional mean, so the expectation factors and vanishes. For $j = k$, the covariance assumption $\mathrm{Cov}(z_j \mid H_S, y) = \sigma_z^2 I_d$ gives

$$\mathbb{E}\big[(u^\top z_j)^2 \mid H_S, y\big] = u^\top(\sigma_z^2 I_d)u = \sigma_z^2 \|u\|_2^2.$$

Only the diagonal terms survive:

$$\mathbb{E}[\eta^2 \mid H_S, y] = \sigma_z^2 \|u\|_2^2 \sum_{j \in N} a_j^2.$$

Taking the expectation over (H_S, y) yields the equality in Eq. (2).

Step 4: lower-bound the softmax weights. Every score lies in $[-B, B]$. The softmax numerator therefore obeys $e^{s_j} \geq e^{-B}$, and the denominator obeys $\sum_{i=1}^{G} e^{s_i} \leq G\, e^{B}$. Dividing the two,

$$a_j = \frac{e^{s_j}}{\sum_i e^{s_i}} \;\geq\; \frac{e^{-B}}{G\, e^{B}} = \frac{e^{-2B}}{G} \qquad \text{for every group } j.$$

Squaring and summing over the r irrelevant groups gives $\sum_{j \in N} a_j^2 \geq r\, e^{-4B}/G^2$. Substituting this into Step 3 produces the lower bound in Eq. (2).

Exact-zero selector. A selector whose nonzero weights are restricted to S has $a_j = 0$ for every $j \in N$. Then $\eta = 0$, the last two terms in Step 1 vanish, and no noise term is added. This proves the final claim. □

What supports it. The bound is not only an inequality on paper. In the controlled grid the learned weights give $\sum_{j \in N} a_j^2 \approx 4.62 \times 10^{-4}$, which predicts a risk increase of 0.00182 against 0.00180 observed (Fig. A4b). Solved at the exact optimum of both model classes, the same leak reproduces the gap between the two optima over twelve orders of magnitude, $r = 0.98$ in $\log_{10}$ across 130 points (Fig. 2c).

Proof of Theorem 3

More data makes the support recoverable. Theorem 3 says that a wrong support requires at least one learned score to miss its population value by the margin. Each score concentrates, so that chance falls exponentially with the sample size.

Proof. The proof chains three inclusions of events and then applies a union bound.

Step 1: a wrong support forces a large score error. Suppose $\mathrm{supp}(p(\widehat{s})) \neq S$. The set $\mathcal{A}_S$ contains exactly the score vectors that sparsemax maps to support S, so $\widehat{s} \notin \mathcal{A}_S$. The margin in Eq. (4) is the shortest distance from s^* to any vector outside $\mathcal{A}_S$, so

$$\{\mathrm{supp}(p(\widehat{s})) \neq S\} \subseteq \{\|\widehat{s} - s^*\|_\infty \geq \gamma\}.$$

Step 2: a large sup-norm error means one large coordinate. The sup norm is the largest coordinate deviation, so

$$\{\|\widehat{s} - s^*\|_\infty \geq \gamma\} = \bigcup_{j=1}^{G} \{|\widehat{s}_j - s_j^*| \geq \gamma\}.$$

Step 3: union bound plus concentration. Apply the union bound, then Eq. (5) with $t = \gamma$ to each coordinate:

$$\Pr\{\mathrm{supp}(p(\widehat{s})) \neq S\} \;\leq\; \sum_{j=1}^{G} \Pr\{|\widehat{s}_j - s_j^*| \geq \gamma\} \;\leq\; 2G \exp\left(-\frac{n\gamma^2}{2\nu^2}\right),$$

which is Eq. (6).

Step 4: solve for the sample size. Require the right side to be at most δ and solve for n:

$$2G \exp\left(-\frac{n\gamma^2}{2\nu^2}\right) \leq \delta \quad \Longleftrightarrow \quad n \;\geq\; \frac{2\nu^2}{\gamma^2} \log \frac{2G}{\delta}.$$

With this many samples, the support is recovered with probability at least $1 - \delta$. *What supports it.* Support F_1 rises from 0.987 at $n = 128$ to 0.999 at $n \geq 512$ in the controlled grid (Fig. A4c), and the plug-in form of the bound is never below the measured failure rate across the 48 sparse cells (Fig. A3b). The eight cells with a negative mean difference all sit at the smallest sample, $n = 128$. □

PROOF OF COROLLARY 4

The two costs can be compared directly. Corollary 4 says that the cost of choosing a support is the chance of choosing wrong times the price of being wrong. When that product falls below the floor bounded softmax keeps, exact zeros are the better choice.

Proof. Let E be the event that the learned support is correct. The law of total expectation splits the excess risk over E and its complement E^c:

$$\mathbb{E}_{\text{train}}[\Delta_{\text{sp}}] = \mathbb{E}[\Delta_{\text{sp}} \mid E]\ \Pr(E) + \mathbb{E}[\Delta_{\text{sp}} \mid E^c]\ \Pr(E^c).$$

On E the excess risk is at most ϵ_n, and on E^c it is at most M, both by assumption. Using $\Pr(E) \le 1$,

$$\mathbb{E}_{\text{train}}[\Delta_{\text{sp}}] \ \le\ \epsilon_n + M\ \Pr(E^c).$$

Theorem 3 bounds the failure probability: $\Pr(E^c) \ \le\ 2G\exp(-n\gamma^2/2\nu^2)$. Substituting it gives Eq. (7). Finally, Theorem 1 lower-bounds the excess risk of bounded softmax by $r\sigma_z^2\|u\|_2^2\, e^{-4B}/G^2$. Whenever the right side of Eq. (7) is smaller than this floor, exact-zero selection has the lower expected excess risk. *What supports it.* Read with measured failure rates and measured failure costs, this comparison reproduces 57 of the 60 observed regimes, with no false win (Fig. A3d). The three disagreements are boundary cells where it is conservative. □

PROOF OF REMARK 2

Correlation does not remove the floor. Remark 2 says that the floor does not need the irrelevant values to be independent. Correlated noise leaves the same bound, with the smallest eigenvalue of its covariance in place of the variance.

Proof of Remark 2. Theorem 1 used a diagonal conditional covariance. Here we only assume a lower bound on the stacked covariance, which allows correlation between the residuals. The proof has four steps.

Step 1: write the noise contribution as one inner product. Let $a_N = (a_j)_{j\in N}$ collect the weights on the r irrelevant groups, and stack their residual noise as $z_N = (z_j)_{j\in N} \in \mathbb{R}^{rd}$. Define

$$q = a_N \otimes u \in \mathbb{R}^{rd}, \qquad \eta = q^\top z_N, \tag{12}$$

where $\otimes$ is the Kronecker product, so q stacks the blocks $a_j u \in \mathbb{R}^d$. Reading $q^\top z_N$ block by block gives $\sum_{j\in N} a_j\, u^\top z_j$, which is exactly the contribution of the residual irrelevant values to the forecast.

Step 2: decompose the risk. Exactly as in Step 1 of Theorem 1,

$$\begin{aligned}\mathcal{R}(\widehat{y}) - \mathcal{R}(h) &= \mathbb{E}\big[(h(H_S) + \eta - y)^2 - (h(H_S) - y)^2\big] \\ &= 2\,\mathbb{E}\big[(h(H_S) - y)\,\eta\big] + \mathbb{E}[\eta^2].\end{aligned} \tag{13}$$

The first term is zero by the assumed cross-term condition, so only $\mathbb{E}[\eta^2]$ remains.

Step 3: lower-bound the quadratic form. The weights a_N are measurable with respect to H_S, so conditioning on (H_S, y) fixes q. The conditional mean of z_N is zero, hence

$$\mathbb{E}[\eta^2 \mid H_S, y] = q^\top \operatorname{Cov}(z_N \mid H_S, y)\, q. \tag{14}$$

The assumption $\operatorname{Cov}(z_N \mid H_S, y) \succeq \lambda I_{rd}$ says the covariance exceeds λ in every direction. Applied to the direction q,

$$q^\top \operatorname{Cov}(z_N \mid H_S, y)\, q \ \ge\ \lambda\|q\|_2^2 = \lambda\,\|a_N \otimes u\|_2^2 = \lambda\,\|u\|_2^2 \sum_{j\in N} a_j^2.$$

The last equality holds because the Kronecker blocks are disjoint: $\|a_N \otimes u\|_2^2 = \sum_{j\in N}\|a_j u\|_2^2 = \|u\|_2^2 \sum_{j\in N} a_j^2$.

Step 4: apply the softmax floor. Bounded scores give $a_j \geq e^{-2B}/G$ for every group, exactly as in Step 4 of the proof of Theorem 1. Squaring and summing,

$$\sum_{j \in N} a_j^2 \;\geq\; r\,\frac{e^{-4B}}{G^2}. \tag{15}$$

Taking expectations in Eq. (14), inserting Eq. (15), and combining with Eq. (13) yields

$$\mathcal{R}(\widehat{y}) - \mathcal{R}(h) \;\geq\; \frac{r\lambda\|u\|_2^2}{G^2}\, e^{-4B},$$

which is Eq. (3). An exact-zero selector supported on the relevant groups sets $a_N = 0$, hence $q = 0$ and $\eta = 0$: the entire residual-noise term vanishes. This proves the final claim. *What supports it.* The benchmark inputs are strongly correlated rather than independent: 6.3 effective directions among the 84 ETTh1 groups and 3.3 among the 2,484 METR-LA groups (Appendix AA). The bounded-score comparison on those groups still returns the predicted direction, sparsemax lower in 76 of 95 cells (Appendix Y). □

E Claims, Evidence, and Notation

Every claim has one place to check it. Table A2 pairs each claim with the experiment that supports it. Table A3 places the paper against prior work on sparse attention. Table A4 collects the notation used throughout.

Table A2: Every claim in the paper and the evidence that supports it.

Claim	Evidence
Bounded softmax passes a minimum residual-noise variance to the prediction	Theorem 1, proof in Appendix D; the learned $\sum_{j \in N} a_j^2$ predicts the noise-driven increase in softmax excess risk, 0.00182 predicted against 0.00180 observed (Fig. A4b); the closed form of Appendix G separates this noise term from the support-bias term
The noise floor survives correlated residuals	Remark 2, proof in Appendix D
Sparsemax recovers the support with enough samples	Theorem 3; empirical failure rate never exceeds the bound (Fig. A3b); F1 rises 0.987→0.999 with n (Fig. A4c)
Sufficient sparse-win condition: sparse risk upper bound falls below the bounded-softmax residual-noise floor	Corollary 4; controlled phase map plus a 57/60 retrospective measured-cost consistency check (Fig. A3c); the converse is not claimed
Win when support sparse and recoverable; tie when all relevant; dense-preferred when scarce and noisy	Phase map, 2,400 fits, 20 seeds (Fig. 1); Wilcoxon $W = 210$, $p = 9.5 \times 10^{-7}$
Map survives temporal dependence and heavy tails	AR(1) $\phi \in \{0.5, 0.9\}$ and t_3 noise, 720 fits (Appendix X.4)
The extent of the win region is set by the score bound; the small-sample corner is not	$B \in \{2, 3, 4, 8\}$, win region concentrates on the sparsest supports; dense-preferred cells fixed at $(k, n) = (8, 128), (16, 128)$ (Table A24, Appendix Z)
The tie is not a property of one market or one backbone	Same 15 settings in a Transformer with unbounded logits, $p = 0.73$ (Appendix Q); semiconductor supply chain, four resolutions, predictions registered before running, every prediction held (Section 6, Appendix P)
Neural probe recovers the main sparse/dense structure, with low-noise exceptions	120 trained BiGRU-attention networks (Table A23, Appendix X.5)
The probe forecaster is competitive	13/15 vs. best Transformer ($W = 108$, $p = 0.0021$); 88/139 vs. LSTM, 106/139 vs. GRU; asset-level Wilcoxon $p = 0.0038$, 1.2×10^{-7}; tied with a liquid CfC network; a classical AR–GARCH reference is competitive at this horizon, where the predictable signal is conditional variance, a quantity outside input selection (Section 6, Tables A22, A18, A21)
Under the multi-scale encoder, where the coordinate is high, the two normalizations are matched	Matched swap 61/139, $p = 0.94$; asset-level equivalence sensitivity; ten-seed rerun; z-score check (Section 6, Appendix X.1, X.2)
Where the coordinate is low, exact zeros lower test error, in both directions of attention and in two domains	269/278 cross-section (+2.32%), 276/278 size-matched time axis (+12.73%), 228/242 macro panel (+15.69%); all constructions, cells and falsifiers registered first (Appendix K, J)
The advantage survives dependence-robust intervals and an absolute reference	moving-block bootstrap over the shared origins; per-origin training-mean reference, median ratio 1.045 against 1.176 on the macro panel (Appendix J, K)
The coordinate is associated with the outcome at the level of the setting, not within one setting	three low-coordinate settings win, the high-coordinate setting ties; the per-cell criteria it does and does not clear are reported with their power (Appendix K, J)
Real-data matched normalizations have a small aggregate accuracy gap; the mechanism of that tie is not identified by raw attention mass alone	Matched comparison plus practical-equivalence sensitivity (Fig. A1); recurrent injection probe and score-separation stress test (Fig. A10, Appendix X.6); direct attribution requires frozen full-model nuisance resampling
Observed outcomes differ across domains and horizons	FX: ties LSTM, beats Transformers (Table A19); traffic: reversal by horizon (Table A7)

Table A3: Position relative to prior work on sparse attention.

Work	Mech.	Theory	Data-dist. boundary	Real-data probe	Falsifiable test
Sparsemax (Martins and Astudillo, 2016)	✓	Jacobian	–	–	–
Entmax (Peters et al., 2019; Correia, Niculae, and Martins, 2019)	✓	–	–	–	–
Hard-concrete gates (Louizos, Welling, and Kingma, 2018)	✓	–	–	–	–
Adversarial Sparse Transf. (Wu et al., 2020)	✓	–	–	✓	–
Softmax is not enough (Veličković et al., 2025)	–	length	–	–	–
Long-context sparse attn. (Vasylenko et al., 2026)	✓	length	–	–	–
Sparse frontier (Nawrot et al., 2026)	–	–	efficiency	✓	–
This paper	probe	risk bounds	✓	✓	✓

Table A4: Notation.

Symbol	Meaning
G, S, N, k, r	number of input groups; relevant support; irrelevant set; $\|S\|$; $\|N\| = G - k$
y, $\widehat{y}$, $h(H_S)$	target; prediction; predictor built from the relevant representation
a_j, s_j, B	attention weight and score of group j; score bound $\|s_j\| \le B$
z_j, σ_z^2, u	residual value noise of group j; its variance; readout vector. The irrelevant set N is also called the nuisance
λ	lower eigenvalue bound of the stacked residual covariance (Remark 2)
γ, ν^2, n	support margin; sub-Gaussian proxy of learned scores; sample size
ϵ_n, M, E	on-support excess risk; failure cost; event of correct support recovery
L	leaked softmax mass $r/(ke^{2B} + r)$ in the linear probe (Appendix G)
$\mathcal{R}(\cdot)$	squared-error risk $\mathbb{E}[(f - y)^2]$
NMSE	mean squared error normalized by the period's squared price range
Quantities read off the training split	
r_{eff}	participation ratio of the input correlation spectrum: how many independent directions the groups span
k_{eff}	how many of those directions carry signal about the target
$k_{\text{eff}}/r_{\text{eff}}$	the *coordinate*: the relevant share among independent directions, in $[0, 1]$
$\widehat{\sigma}_z$	root ratio of off-support to on-support eigenvalues
Frozen-model sensitivity	
K	number of times a perturbation is resampled on a frozen model
Ψ_f	sensitivity of a frozen model: the prediction variance across those K resamples
Ψ_{signal}	the same quantity when the whole context is resampled
Ψ_{zero}	the same quantity when only the positions sparsemax set to zero are resampled
Ψ_{random}	the same quantity for an equally large set of positions drawn at random
Ψ_{noise}	the same quantity for the appended nuisance steps
δ_{eq}	the practical margin: how large a relative error difference still counts as equivalent
$p^\star$	the weight vector that minimizes training error over the simplex
Quantities used to report a real-data run	
cell	one data setting: a target, a training size, and a window. A *setting* is the asset-period unit of the cryptocurrency benchmark, which is a different object
origin	one rolling train/test split inside a cell; 20 per cell
Δ, 2 SE	softmax minus sparsemax test MSE, averaged over a cell's origins; twice its standard error
baseline	the training-mean forecast, scored on each origin separately (Appendix J)

F Compute Resources

All experiments run on CPU. The controlled study (2,400 fits plus the 720 robustness fits) completes in under ten minutes on a single core. The neural phase map (120 networks) and the multi-seed matched comparison (300 networks) take about 1.5 and 4 CPU-hours. The 139-setting cryptocurrency benchmark across six models and three seeds is the largest neural component, about 40 CPU-hours, with per-epoch times reported in Table A20; foreign exchange, traffic, the liquid-network baseline, and GARCH add under 4 CPU-hours. The three low-coordinate runs fit Definition 1 rather than trained networks: the cross-section (11,120 fits) takes 213 seconds and the size-matched time axis (11,120 fits) 341 seconds on one core, while FRED-MD (9,680 fits over 120 channels each) is the heavier one at about 16 CPU-hours single-threaded, run here over twelve processes. The bootstrap intervals and the per-origin training-mean reference add under ten minutes. The total for all reported results, including reruns, is under 80 CPU-hours; no GPU was used.

G Evaluating the Decision Rule Cell by Cell

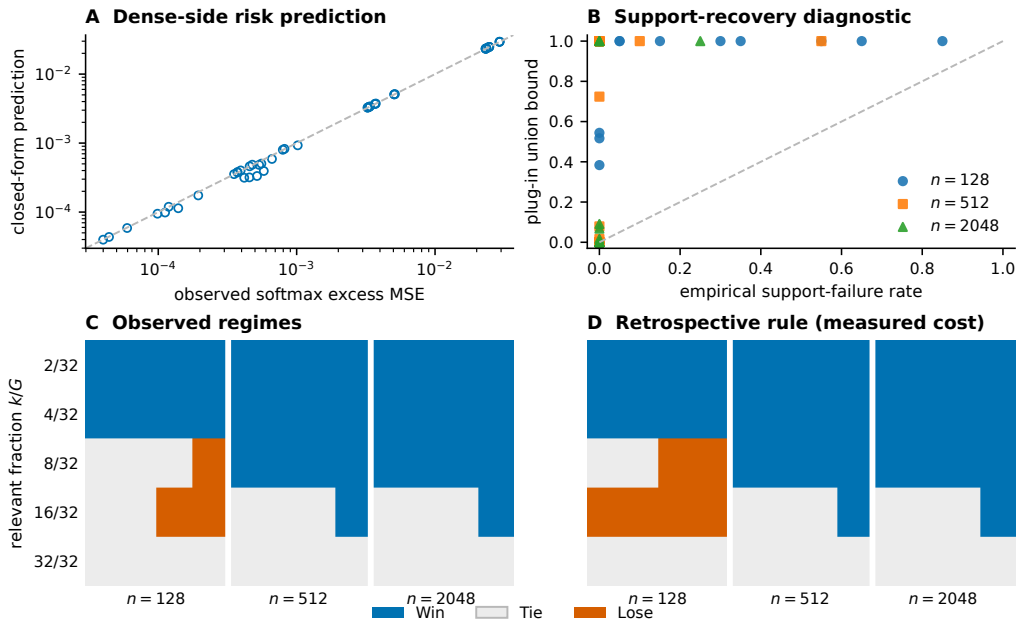

Figure A3: **Decision-rule diagnostics.** (a) Closed-form dense-side risk against observed softmax excess MSE. (b) A retrospective plug-in version of the support-recovery bound against empirical support-failure rates; the plug-in bound is conservative, and this panel is a diagnostic. (c) Observed win/tie/dense-preferred regimes. (d) Regimes produced by the rule using measured failure rates and measured failure costs; (c)–(d) use the same rule as Fig. 1, so (c) reproduces that map cell for cell and the two maps agree in 57 of 60 cells. Blue: sparsemax wins; gray: tie; orange: dense-preferred.

This appendix gives the constants behind Fig. A3. The script `run_rule_evaluation.py` re-runs the 2,400 controlled fits, keeps every learned score vector, and writes `rule_evaluation.json` with all quantities below for each of the 60 cells.

Softmax side. In the linear probe $\widehat{y} = p(s)^\top x$ the readout is $u = 1$, the relevant representation is $H_S = x_S$, and the irrelevant values are $z_j = x_j$ with variance σ_z^2. With $|s_j| \le B$, the largest achievable ratio between a relevant and an irrelevant weight is e^{2B}, so the total mass that softmax must leave on the r irrelevant inputs is $L = r/(ke^{2B} + r)$, spread as L/r each. The measured leak matches this value to within 0.005 in every cell with $n \ge 512$ (column `L_measured` of `results/rule_evaluation.json`). Theorem 1 in its equality form gives the noise term $\sigma_z^2 \sum_{j \in N} a_j^2 = \sigma_z^2 L^2/r$. Because the simplex mass is conserved, the same leak scales the relevant

weights by $1 - L$. That adds a bias $(1 - (1 - L))^2 \operatorname{Var}(k^{-1} \sum_S x_j) = L^2/k$. This is the $\mathcal{R}(h)$ term with softmax's own h, so in the linear probe Corollary 4 compares against a dense predictor carrying both the bias and the noise term. Both are costs of the same quantity, the leak L. The theorem's content is that bounded scores make $L > 0$ unavoidable, and that exact-zero selection removes both. Which cost dominates depends on σ_z^2 and k. Both vanish together under exact-zero selection, and both follow from (B, k, r, σ_z) without fitting. Adding the finite-sample estimation cost $(k-1)\sigma_\varepsilon^2/n$ gives the prediction in Fig. A3a: median relative error 0.6%, maximum 36% (at $k = 16$, $n = 128$, where the finite-sample leak exceeds the population value).

Sparsemax side. On the event E of correct support recovery, sparsemax solves least squares on k inputs under a sum-to-one constraint, so $\epsilon_n = (k-1)\sigma_\varepsilon^2/n$; the measured on-support excess agrees with this to a median 12% (maximum 34%). For Theorem 3 we take s^* as the seed-mean learned score vector, $\nu_j^2 = n \operatorname{Var}(\widehat{s}_j)$ across seeds as the per-coordinate sub-Gaussian proxy, and the margin $\gamma = \frac{1}{2} \min\{\tau(s^*) - \max_{j \in N} s_j^*,\ \min_{j \in S} s_j^* - \tau(s^*)\}$, where τ is the sparsemax threshold; the factor $\frac{1}{2}$ accounts for the threshold moving with the perturbation. The failure bound is the union over coordinates exactly as in the proof, $\sum_j 2\exp(-n\gamma^2/2\nu_j^2)$, truncated at one. It is never below the empirical failure rate (Fig. A3b). It is informative, below 0.1, in 17 of 48 cells and conservative in the rest. The looseness has two sources: the $2G$ union, and irrelevant coordinates whose scores stop being identified once they leave the support, which inflates ν_j^2.

Retrospective reading of the rule. With measured constants (empirical failure rate and worst observed failure excess M), the diagnostic $\epsilon_n + \Pr(E^c)M < L^2/k + \sigma_z^2 L^2/r$ agrees with 57 of 60 observed regimes under the classification rule of Fig. 1. The three disagreements are boundary cells at $k/G \in \{1/4, 1/2\}$ and $n = 128$. The diagnostic calls them dense-preferred and the data call them ties, so the diagnostic is conservative there. This 57/60 calculation reuses measured failure rates and costs, so it is a descriptive consistency check rather than an out-of-sample rule evaluation.

A theorem-level certificate is not claimed from these fitted cells. It would require two things we do not have: concentration constants justified independently of the same 20 fitted seeds, and a valid worst-case failure cost. For this quadratic simplex model, the maximum prediction excess over simplex vertices is $\max\{1 - 1/k,\ \sigma_z^2 + 1/k\}$, which exceeds $\sigma_z^2 + 1/k$ when a wrong solution concentrates on one relevant coordinate. Figure A3 therefore separates the closed-form dense-side check, the plug-in support diagnostic, and the retrospective measured-cost regime map.

H Supplementary Figures and Tables

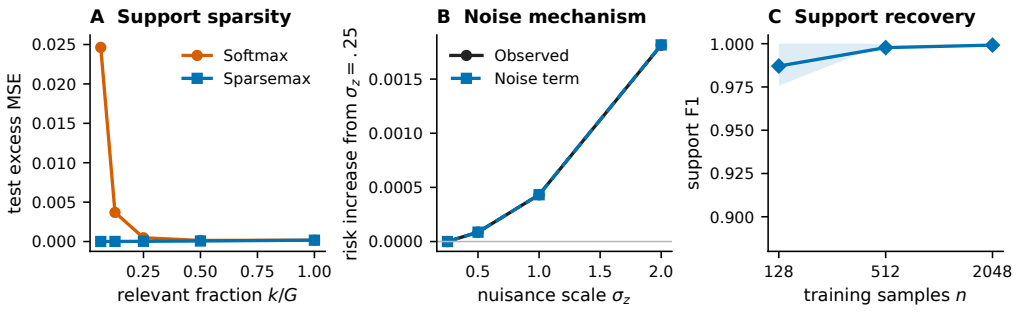

Figure A4: **Controlled comparison.** 2,400 matched fits over 20 seeds; error bars indicate one standard deviation. Softmax excess MSE exceeds that of sparsemax across all 20 seeds (Wilcoxon $W = 210$, $p = 9.5 \times 10^{-7}$). (a) At $n = 512$ and $\sigma_z = 1$, the sparsemax advantage diminishes as support size k increases. (b) At $k = 4$, excess risk scales with nuisance variance as predicted by Theorem 1. (c) Support recovery improves with sample size; when all inputs are relevant ($k = G$), the paired difference converges to zero.

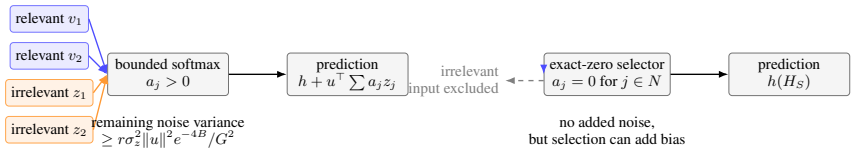

Figure A5: **Mechanism behind Theorem 1.** The analysis concerns noise that remains in the learned values. Bounded softmax passes some of it to the prediction; selecting the correct support removes it. If the value map makes $\sigma_z^2 = 0$, or later layers make $\|u\| = 0$, the softmax lower bound is zero.

Figure A5 sketches the mechanism the theorem describes.

Finite logits and bias–variance. For finite B, softmax has no exact zero, but the bound decays exponentially in B: a dense model can shrink the effect by separating its scores, and whether such separation is harder to learn is not settled by the theorem (Fig. A4). Exact-zero selection removes noise variance but adds bias when it drops a useful input. Its benefit should grow with r, σ_z^2 and downstream sensitivity, and its support estimate improves with n and γ. With many weak signals softmax may average more reliably (Table A5, Appendix H).

Table A5: Predictions from the risk analysis.

Data and representation	Expected result
Much irrelevant noise remains	Exact-zero selection can reduce error
The value map removes irrelevant noise	No required softmax disadvantage
Many weak inputs are relevant	Softmax averaging may help
Sample too small to identify the support	Dense attention preferred

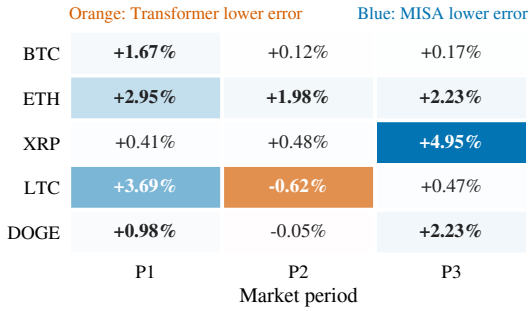

Figure A6: **Relative NMSE advantage over the best tested Transformer.** Each cell uses the lowest mean among iTransformer, PatchTST, and Informer over seeds 42–44. Positive values mean lower error for MISA. MISA improves 13 of 15 settings; the median advantage is 0.98%.

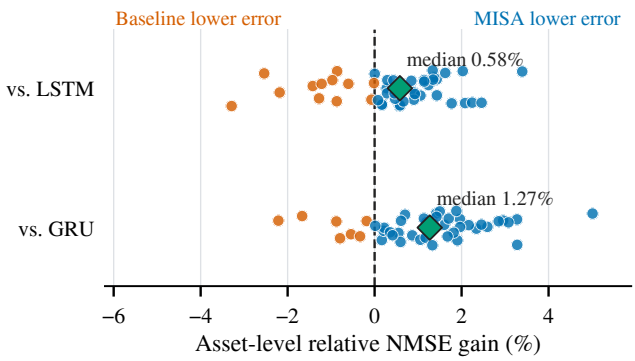

Figure A7: **Expanded evaluation after aggregation by asset.** Each dot is one of 47 assets after averaging its relative NMSE differences across available periods. Diamonds mark medians. Positive values mean lower error for MISA.

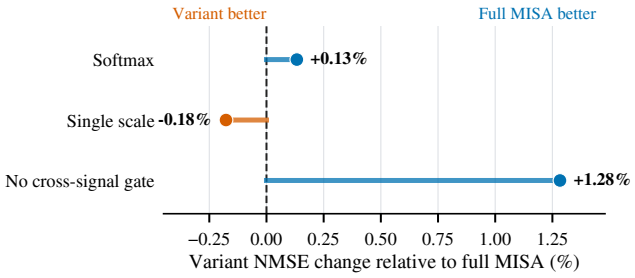

Figure A8: **Matched ablation over eight asset-period settings.** Positive values mean that the variant has higher mean NMSE than full MISA. Full values and epoch times appear in Table A20.

Table A6: Cryptocurrency benchmark at a glance.

Period	Month	Market window	Assets	Settings
P1	Nov. 2022	FTX crisis	45	45
P2	Oct. 2023	Pre-ETF accumulation	47	47
P3	Mar. 2024	Post-ETF shift	47	47
Total: ≈1.2M five-minute records			47	139

Table A7: METR-LA sensor-pair mean NMSE over three seeds; best in bold. "div." = run excluded as non-converged (NMSE $> 10^4$); not an architecture result.

Horizon	MISA	LSTM	GRU	iTrans.	PatchTST	Informer
15 min	35.69	30.60	33.03	32.20	**16.47**	25.59
30 min	79.29	139.92	153.29	141.35	**43.45**	82.31
60 min	**98.63**	1503.1	480.5	div.	div.	div.

I A Registered Test Where the Map and the Null Disagree

The registration (`PREREGISTRATION_semisynthetic.md`, 2026-09-19 14:49 UTC) fixes the construction, the four predictions and their falsifiers before any run; Amendment 1 (14:58 UTC) specifies the real channels as growth rates so that training-split standardization carries to the test period, with the predictions unchanged. The run under the registered construction is the one reported, and Table A8 lists its cells.

Table A8: **Registered predictions and outcomes.** Δ is softmax minus sparsemax standardized test MSE over 20 paired draws; P4 is a deterministic fit, before and after removing two outlier test months (Appendix I).

	Cell	Predicted	Δ	2 SE	sparsemax lower	Verdict
P1	$m = 19, \sigma_z = 2$	win	$+0.060$	0.019	20/20	held
P2	$m = 19, \sigma_z = 1$	win	$+0.063$	0.020	18/20	held
P3	$m = 19, \sigma_z = 0.25$	tie	$+0.055$	0.022	16/20	win observed
P4	$m = 0$	tie	$+0.004$ / -0.000	—	—	tie

Construction. The panel is the semiconductor monthly production series, 13 channels as one-month log differences, with the three-month-ahead log growth of shipments as target. To it we append $m = 19$ channels of independent $\mathcal{N}(0, \sigma_z^2)$ noise, giving $G = 32$. Training uses $n_{\text{train}} = 128$ chronological origins and 251 test origins. Definition-1 probe with scores bounded at $|s| \leq 8$; 20 noise draws, paired; the regime rule of Fig. 1.

Robustness (run after, not registered). Removing the two outlier test months leaves P1–P3 at $+0.050$, $+0.052$, $+0.042$, all wins. Across five 128-month training windows starting at origins 0, 50, 100, 150 and 200, P1 wins in all five (Δ from $+0.014$ to $+0.067$, sparsemax lower in 7 to 9 of 10 draws each) while P4 has no consistent sign. In P1 sparsemax sets all 19 appended channels to exactly zero and keeps five of the 13 real channels (capacity utilization, manufacturing output, the SOX index, employment, computer output).

Reading. The win the map predicts at small n with many independent irrelevant inputs is real, and it is produced by excluding exactly those inputs. The null hypothesis that normalization never matters predicted a tie here; the map's win occurred. P3 was registered as a tie and a win was observed ($+0.055$, $2\,\text{SE} = 0.022$): at the wide bound $B = 8$ the advantage is as large at $\sigma_z = 0.25$ as at 2. The reading is post hoc: the effect comes from the count of irrelevant independent inputs relative to n, not from their variance. It is entered in the registration file (2026-09-19, "Prediction for the next test"). It is the hypothesis the frozen-rule test of Appendix O is designed to check. P4 has no appended channels, so it is a deterministic fit with no noise draw. The registered rule needs a draw-to-draw standard error, and here there is none. On all 251 test months $\Delta = +0.004$, and -0.000 after removing the two test months with a real channel at $|z| > 10$. Both sit within the $2\,\text{SE}$ scale of the noisy cells, 0.019–0.022, so P4 is read as the predicted tie. That is the estimation side of Corollary 4, not the residual-noise floor of Theorem 1, and it agrees with Appendix Z: what survives the wide bound is not the leak.

J FRED-MD TEST

`PREREGISTRATION_fredmd.md` fixes the construction, the screening, the decision rule, four acceptance criteria and four falsifiers, and records the outcome with its amendments.

Construction. FRED-MD, vintage downloaded 2026-09-20, transformed by the codes in the file and cleaned by the standard outlier rule of the FRED-MD documentation (observations more than ten interquartile ranges from the series median are treated as missing and interpolated). The panel is 785 months by 121 series, 1960-01 to 2025-07. A target series is predicted one month ahead from the contemporaneous transformed values of the other 120. Predictor: Definition 1 with scores bounded at $|s| \leq 8$. Replication unit: 20 rolling origins, each training on n months and scored on the next 120, with $n \in \{128, 256\}$. Every series is a target: 242 cells, 4,840 origins, none selected by us.

Outcome. Sparsemax has the lower test error in 228 of 242 cells and 3,929 of 4,840 origins. Mean relative gain 15.69%, median 15.52%. Intervals come from a moving-block bootstrap, block length 4, over the 20 rolling origins, resampled jointly for every cell. That carries both the overlap between training windows and the cross-sectional correlation between series into the interval. It gives $[13.98\%, 17.31\%]$ at 90%, against $[14.44\%, 16.94\%]$ if cells are treated as independent and $[14.25\%, 17.11\%]$ clustering by target series. The Wilcoxon $p = 1.3 \times 10^{-36}$ on the paired cell means assumes independence across cells and is reported as a descriptive statistic. By training size:

$+20.74\%$ at $n = 128$ (117 of 121 cells) and $+10.63\%$ at $n = 256$ (111 of 121). Two cells of 242 favor dense attention. The comparison against the training-mean forecast is in the next paragraph, where that reference is computed per origin rather than assumed.

An absolute reference, computed per origin. Each predictor is fitted and scored on a target standardized by its own training block, so the natural reference is the training-mean forecast: predict the training mean at every test point. On the standardized scale that is the constant zero, and its test MSE on a given origin is the mean squared standardized target over that test block. This is *not* one in general, because the test block carries its own level and its own variance. Measured per origin it has a median of 0.996 on the macro panel, 1.141 across the cross-section and 1.083 on the matched time axis, with a 5–95% range of roughly 0.26 to 4.6. All quantities below are ratios of the raw standardized test MSE to that origin's own baseline, so a value under one means the predictor beats the training mean on that origin. Table A9 gives the two readings.

Table A9: **Against the training-mean forecast, scored per origin.** Median ratio of standardized test MSE to that origin's own baseline, and the share of origins below it.

run	origins	median MSE / baseline		share below baseline	
		softmax	sparsemax	softmax	sparsemax
macro panel	4,840	1.176	**1.045**	31.1%	**42.2%**
time axis, matched	5,560	1.208	**1.081**	2.2%	**6.6%**
cross-section	5,560	1.031	**1.011**	22.6%	**35.5%**

Sparsemax is closer to the reference than softmax in every run and clears it on more origins in every run. On the macro panel, restrict to the 1,350 origins where both predictors are below their own baseline. The advantage there is $+14.77\%$, with sparsemax lower in 75.2% of them. The effect is therefore not an artifact of comparing two predictors on the unpredictable part of the panel. The corresponding cross-sectional subset is 1,063 origins at $+0.64\%$; on the matched time axis only 38 origins have both predictors below their own baseline, too few to read. Script: `scripts/run_absolute_baseline.py`; figures: `results/absolute_baseline.json`.

Resolution of the coordinate comparison. The registered criteria include a comparison of win rates between cells the coordinate places on either side of 0.15. The observed rates are 66.7% among predicted-win cells against 70.5% among predicted-tie cells, so the criterion as written is not met. At roughly 120 cells per group the design resolves a difference of 16.7 percentage points at 80% power, while the difference at stake is 3.8 points. A scan over every threshold from 0.05 to 0.45 separates the groups by at most 0.89 standard errors. Resolving a five-point effect would take about 2,700 cells. The comparison is reported with both facts stated: the threshold is not cleared, and the design cannot resolve a difference of this size. What the design does resolve, with the precision quoted above, is the effect size and its movement with the sample size.

Amendments. Amendment 1 adds two steps: the standard FRED-MD outlier rule, and a guard that drops a channel for an origin when its training block carries no usable scale. Without them a series that is near constant in one block and jumps later is standardized by an almost zero scale. Amendment 2 computes the coordinate on the per-column standardized training inputs, the representation the predictor is fitted on, since the panel mixes rate differences with log differences. Amendment 3 records the resolution above. The pre-amendment and post-amendment result files both ship with the code (`results/fredmd_preamendment.json`, `results/fredmd.json`, `results/fredmd_perorigin.json`).

K Cross-Sectional Test

`PREREGISTRATION_crosssection.md` fixes the construction, the screening, the decision rule, four acceptance criteria and four falsifiers, and records the outcome with its amendments.

Construction. The same Coinbase five-minute panel and the same three event-aligned windows used throughout the paper. For a target asset, the inputs are the contemporaneous five-minute log returns of every other asset in that window (44 to 46 channels); the target is that asset's log return five bars ahead. The predictor is Definition 1 with scores bounded at $|s| \leq 8$. The replication unit is 20 rolling origins evenly spaced through the window, each trained on n consecutive bars and scored on the next 500; origins, not random seeds, supply the variability the rule needs. Training sizes are $n \in \{128, 256\}$.

Screening. Every asset present in a window is a target, at both training sizes: 278 cells. None is selected by us, and all 278 are reported. The rule, unchanged from Appendix L, predicted 154 wins and 124 ties, distributed very unevenly across windows: 1 win of 90 cells in the FTX window, 61 of 94 before the ETF, 92 of 94 after it.

Outcome. Sparsemax has the lower test error in 269 of the 278 cells. The mean relative gain is 2.32% (90% CI $[1.81\%, 2.78\%]$ from a moving-block bootstrap over the shared rolling origins; $[2.15\%, 2.49\%]$ if cells are treated as independent), and per window $+2.04\%$, $+3.24\%$ and $+1.67\%$. At the level of individual origins, sparsemax is lower in 3,815 of 5,560. No cell resolves in favor of dense attention.

The four criteria, as registered. (1) Informative: met under the registered rule, an observed win rate of 0.487 among predicted-win cells against 0.403 among predicted-tie cells. (4) Complete: met, all 278 cells reported with their predictions. Criteria (2) and (3) turn on differences the design resolves only in part, and are reported as such below.

Resolution of the cell-level comparisons. Criterion 2 asks that at least half of the predicted wins resolve as wins. The observed 75 of 154 is below that threshold, so the criterion as written is not met. Its 90% interval is $[0.418, 0.556]$ and contains the registered 0.5 (exact $p = 0.81$). The design therefore does not resolve which side of the threshold the true rate lies on. The same holds for the win-rate contrast between the two coordinate groups. At roughly 140 cells per group the design resolves a difference of 16.8 percentage points at 80% power. Both the registered reading, 8.4 points in the predicted direction, and a scale-free recomputation, 5.5 points, sit inside it. Of the 79 predicted-win cells that resolved as ties, 77 have $\Delta > 0$. Seventy-two of those clear the practical margin while falling inside $2\,\mathrm{SE}$, whose median is 0.038 against a median margin of 0.008. The reason is that the registered criterion pairs an *absolute* Δ with a *relative* margin, and test blocks at different origins differ in volatility by a factor of several. The scale-free recomputation, which divides each origin's difference by its own softmax error, removes that spread and is reported beside the registered analysis. Criterion 3 asks the rule to improve on two fixed strategies. It is not met: selecting by the rule returns a mean test MSE of 1.828, below always-softmax at 1.851 ($t = 12.3$, $p = 4 \times 10^{-28}$) but above always-sparsemax at 1.804. In a geometry the coordinate reads as favorable throughout, always-sparsemax is the action the coordinate itself assigns, and the rule's selectivity costs the difference. What the design resolves with precision is the effect itself, quoted above.

Falsifiers. None of the four fired. The win rate among predicted-win cells is not below that among predicted-tie cells under the registered rule. No predicted-win cell resolves as dense-preferred. The rule-selected error is not above always-softmax. And the FTX window does not produce a higher win rate than the post-ETF window, 0.333 against 0.543, which is the ordering the coordinates imply.

The size-matched time-axis control. The first version of this test compared the cross-section against the multi-scale encoder of MISA, which differs from Definition 1 in both the predictor and the protocol, so the comparison could not isolate the geometry. The control removes that. For the same 278 targets it keeps the time axis and gives Definition 1 the target's own past L five-minute bars, with L equal to the number of other assets in the panel. Everything else matches the cross-sectional run: the channel count, the target, the horizon, the origins, the training sizes, the test length, the score bound and the decision rule. Only which channels the attention sees changes. Under that control the coordinate reads 0.02 to 0.26, and sparsemax has the lower test error in 276 of 278 cells, by 12.73% (90% CI $[11.18\%, 13.93\%]$), with no cell meeting the registered criterion for a dense win. The conclusion this supports is therefore about the coordinate and not about the axis: three

settings at a low coordinate all favor exact zeros, and the setting that ties is the one at 0.50–1.00. Script: `scripts/run_time_axis_control.py`; results: `results/time_axis.json`.

Intervals under dependence. Cells in a panel are scored on the same calendar origins, the rolling training blocks overlap, and the target series are cross-correlated, so an interval that treats cells as independent is too narrow. Every interval quoted for these two runs comes from a moving-block bootstrap (block length 4, 5,000 replicates) over the 20 rolling origins, resampled jointly for all cells at once, which carries both the window overlap and the cross-sectional dependence. For the cross-section this widens the 90% interval from $[2.15\%, 2.49\%]$ to $[1.81\%, 2.78\%]$, and clustering by target series instead gives $[2.14\%, 2.51\%]$. Script: `scripts/run_dependence_robust_ci.py`.

What this test establishes. Exact-zero attention produces a large and consistent win on real market data once the coordinate of the inputs matches the regime the bound describes. That holds in both directions of attention. The coordinate can be read off the training split before any test data is touched.

L Confirmatory Test

`PREREGISTRATION_confirmatory.md` fixes the rule, the coordinates, the 12 cells, the comparators, the outcome measure, three acceptance criteria and four falsifiers, and records the outcome unedited. Inputs are first differences; targets are the 3-day-ahead SOX log growth and the 24-step-ahead ETTm1 OT change; appended channels are independent $\mathcal{N}(0, 1)$; the split is chronological; 20 paired draws per cell; scores bounded at $|s| \leq 8$.

Table A10: **Confirmatory cells, registered predictions and outcomes.** Δ is softmax minus sparsemax test MSE, mean over 20 paired draws; "dense" marks the cells where the dense side was lower.

source	m	n	$k_{\text{eff}}/r_{\text{eff}}$	Pred.	Δ	2 SE	margin	Obs.
semiconductor eq.	0	128	0.33	tie	$+0.0057$	0.0000	0.0084	tie
semiconductor eq.	0	1024	0.33	tie	$+0.0000$	0.0000	0.0086	tie
semiconductor eq.	8	128	0.14	win	$+0.0413$	0.0204	0.0086	win
semiconductor eq.	8	1024	0.12	tie	-0.0064	0.0023	0.0086	tie
semiconductor eq.	24	128	0.05	win	$+0.0961$	0.0426	0.0091	win
semiconductor eq.	24	1024	0.09	tie	-0.0042	0.0054	0.0087	tie
ETTm1	0	128	0.50	tie	-0.0034	0.0000	0.0044	tie
ETTm1	0	1024	0.33	tie	-0.0088	0.0000	0.0018	dense
ETTm1	8	128	0.11	win	$+0.0190$	0.0137	0.0046	win
ETTm1	8	1024	0.09	tie	-0.0058	0.0012	0.0018	dense
ETTm1	24	128	0.05	win	$+0.0547$	0.0225	0.0052	win
ETTm1	24	1024	0.08	tie	-0.0013	0.0031	0.0018	tie

The practical margin is read off each cell. A code review made before the outcome was interpreted showed that the absolute floor of the controlled grid, $|\Delta| > 10^{-4}$, is not a practical margin here. These cells' softmax test MSE spans 0.35 to 1.83, so the floor is 0.006–0.03% of the quantity compared. It is also uninformative where a cell has no appended channels and the draw-to-draw standard error is exactly zero. Amendment 1 of the registration adds, for every cell alike, a practical margin of 0.5% of that cell's own softmax test MSE, so the margin column of Table A10 moves with the dataset. Two points of provenance matter. First, 0.5% is the value this paper uses for the cryptocurrency benchmark, but there it is a retrospective sensitivity threshold, not a registered one, so adopting it here is a reasoned choice and not an appeal to an earlier registration. Second, it was adopted after the outcome table of the original floor existed and before that table was interpreted; that is weaker than freezing a threshold before seeing any outcome, and we do not present it as equivalent.

Both thresholds are therefore reported. Under the fixed floor 7 of 12 cells match their registered prediction; under the scale-relative margin, 10 of 12. A larger margin makes a win harder to declare and a tie easier, so the agreement counts are not comparable as evidence of strictness, and we do not claim the relative margin is the stricter test. What does not move is the win prediction: the four

predicted-win cells resolve as wins under both thresholds. The eight predicted ties resolve as 1 win, 3 ties and 4 dense under the fixed floor, and as 0 wins, 6 ties and 2 dense under the relative margin. The single unpredicted win sits at $(m, n) = (0, 128)$ on the semiconductor source. Its 20 draws are one fit, so its $\Delta = +0.0057$ is compared against a standard error of zero. The statement the confirmatory test supports is the win prediction, and the tie prediction only under the scale-relative margin.

Criteria. (1) Informative beyond "always tie". The win side is met under both thresholds, 4 of 4. On the tie side, 1 of 8 resolves as a win under the fixed floor and 0 of 8 under the scale-relative margin. Overall agreement is 7 of 12 and 10 of 12. (2) Useful for model selection: met in mean, with the paired uncertainty of Table A11 attached. (3) Full reporting: Table A10 lists every cell, and its values are `results/confirmatory_robust.json`. Two further robustness runs ship beside it, `results/confirmatory_amended.json` and `results/confirmatory_amended2.json`; they redraw the 20 paired draws and carry larger standard errors, under which the two ETTm1 predicted-win cells resolve as ties. The registration file records all three with their outcomes, and seeds are used only for training variability.

How large is the selection gain, and where does it come from? Table A11 pairs the rule-selected predictor against each fixed strategy on the same 240 (cell, seed) fits. Against always-softmax, the temperature-tuned softmax and entmax, the mean advantage is 0.7–1.5%. Against always-sparsemax it is 0.17%. That interval excludes zero at the fit level and touches it at the cell level, $[0.0000, 0.0040]$ over the 12 cells. The sign is informative too: the rule is lower than entmax in 188 of 240 fits, but only 90 of 240 against the tuned softmax, whose mean it beats on magnitude rather than on frequency ($p = 0.69$). Split by source, the advantage over always-sparsemax is $+0.50\%$ on ETTm1 and $+0.05\%$ on the semiconductor equities, so it rests mainly on one of the two sources. The honest summary is that the rule gives the lowest mean error of the five strategies here, that its margin over always-sparsemax is small and concentrated, and that a larger confirmatory set is what would settle its stability.

Table A11: **Rule-selected predictor against each fixed strategy**, paired over the 240 (cell, seed) fits of the 12 confirmatory cells. A positive difference favors the rule.

fixed strategy	mean test MSE	mean diff. (MSE)	90% CI (MSE)	relative diff.	rule lower
always softmax	1.1957	+0.0176	[0.0128, 0.0224]	1.47%	68/240
tuned softmax	1.1937	+0.0156	[0.0111, 0.0202]	1.31%	90/240
entmax α=1.5	1.1867	+0.0086	[0.0060, 0.0112]	0.73%	188/240
always sparsemax	1.1801	+0.0020	[0.0014, 0.0026]	0.17%	103/240
rule-selected	**1.1781**	—	—	—	—

Falsifiers. The win-prediction falsifier (two or more predicted wins coming back ties) did not fire; the sample-size falsifier (an unpredicted win at $n = 1024$) did not fire; the model-selection falsifier did not fire. The fourth, worded for any cell in which the dense side is lower, fires at the level of the fitted estimator in two ETTm1 cells at $n = 1024$, and the reference analysis below locates it.

Exact references on the same cells. Solving SIMPLEX and RATIO(8) of Appendix M on the same training samples gives $|\Delta| < 10^{-5}$ and the regime tie in all 12 cells (`results/confirmatory_convex_B8.json`). At the wide bound the two model classes share an optimum on these data, so every difference in Table A10 is a difference between estimators. The frozen rule is therefore an estimator-selection rule, and it is accurate where it commits: 4 of 4, with the selection benefit reported above.

M Exact Convex References for the Controlled Cells

Why. In Definition 1 the training MSE is convex in p, so a comparison between the two normalizations can be made at the optimum instead of at the point an optimizer reaches. Two references are

solved on the identical training sample of every seed (`scripts/run_convex_reference.py`, SLSQP, `ftol`$= 10^{-15}$):

$$\text{SIMPLEX}: \min_{p\geq 0,\ \mathbf{1}^\top p=1} \tfrac{1}{n}\|Xp-y\|^2, \qquad \text{RATIO}(B):\ \text{the same subject to } p_i \geq e^{-2B}\max_j p_j.$$

SIMPLEX is what an exact-zero normalization can represent; RATIO(B) is the weight-ratio range a score bounded by B reaches, which is the quantity Theorem 1 bounds. Neither uses the true support.

The dense side is exact. The softmax fits reach RATIO(B) to 10^{-17}–10^{-11} in training MSE at $B = 2$ and to 4×10^{-5} at $B = 8$, and their test excess equals the reference's to the printed digits. The map's dense side is therefore the optimum of the bounded model class, not an artifact of the fit.

The leak is visible at the optimum. At $B = 2$ the two references differ by orders of magnitude wherever the support is sparse: excess MSE 2.5×10^{-2} against 3×10^{-5} at $(k, \sigma_z, n) = (2, 1, 512)$, 2.9×10^{-2} against 2.1×10^{-4} at $(2, 2, 128)$, 3.7×10^{-3} against 2.0×10^{-4} at $(4, 1, 128)$. Applying the regime rule to the references over the whole grid gives *36 wins, 24 ties and no dense-preferred cells at $B = 2$, and 60 ties at $B = 8$*, the wins confined to $k \in \{2, 4, 8\}$. This is the B-dependence of Fig. 2 with the optimizer removed from the argument. It is the sharpest confirmation of Theorem 1 in the paper. The advantage of exact zeros over bounded softmax is the leak: present at the optimum of both model classes, confined to sparse supports, and gone once the score range is free. It also settles the direction of the comparison at the optimum, as an observation over this grid rather than a guarantee. The model classes are nested in training loss, which does not by itself order two empirical optima in test risk. Across all 60 cells at both bounds, exact-zero selection never falls behind bounded softmax there.

The gap to SIMPLEX is the optimization path, not the score bound. The fitted sparsemax does not sit on SIMPLEX: its training MSE is higher by 2.5×10^{-5} at $(2, 1, 512)$ and by 8.2×10^{-3} at $(16, 2, 128)$. This is not a limit on what the parametrization can express. For any p on the simplex, $\mathrm{sparsemax}(p) = p$, and every entry of p lies in $[0, 1]$. A score vector inside $[-2, 2]$ therefore reaches every point of the simplex. The bound B cannot be what holds a fit away from SIMPLEX, and indeed the gap is identical at $B = 2$ and $B = 8$. `scripts/run_optimization_probe.py` settles it directly (Table A12). Initialized at $s = p^\star$, the optimum's own score vector, the same L-BFGS-B fit reaches SIMPLEX to 10^{-17} in 20 of 20 seeds, in every cell and at both bounds. Eight random Dirichlet starts close only part of the gap. The zero initialization used throughout the paper therefore stops short of the training optimum, and a longer iteration budget does not change that.

What stopping short does. It acts as an implicit regularizer, and its sign follows the support. In the sparse cells the zero-initialized fit attains a *lower* test excess than the SIMPLEX optimum, which no unregularized optimum can do: 6×10^{-6} against 3.1×10^{-5} at $(2, 1, 512)$, and 5.9×10^{-5} against 2.0×10^{-4} at $(4, 1, 128)$. It also recovers the support where the convex optimum does not, mean F_1 1.00 against 0.76. In the smallest-sample cells with a larger support it costs instead: 1.1×10^{-3} against 3.2×10^{-4} at $(8, 1, 128)$, and 1.0×10^{-2} against 5.3×10^{-4} at $(16, 2, 128)$. Initializing at $p^\star$ removes both the benefit and the cost, and the test excess then equals SIMPLEX's exactly. We therefore report two separate statements. The win region and its contraction with B are properties of the two model classes, established at the optimum of both sides. The three dense-preferred cells of Fig. 1 are a property of the optimization path of the fit, and are absent from the reference comparison.

Table A12: **Training MSE gap to the SIMPLEX optimum by initialization**, mean over 20 seeds, $B = 2$ (the values at $B = 8$ are identical to the printed digits). "reached" counts seeds in which the $p^\star$ initialization attains the optimum to 10^{-10}.

cell (k, σ_z, n)	from zeros	from $p^\star$	best of 8 random	reached
$(2, 1, 512)$	2.5×10^{-5}	-6×10^{-18}	2.5×10^{-5}	20/20
$(4, 1, 128)$	1.4×10^{-4}	-3×10^{-18}	1.4×10^{-4}	20/20
$(8, 1, 128)$	1.0×10^{-3}	7×10^{-19}	1.2×10^{-4}	20/20
$(16, 1, 128)$	1.3×10^{-3}	-1×10^{-17}	1.3×10^{-3}	20/20
$(16, 2, 128)$	8.2×10^{-3}	-5×10^{-18}	2.3×10^{-3}	20/20
$(32, 1, 128)$	-6×10^{-17}	-3×10^{-16}	-2×10^{-16}	20/20

Table A13: **Regimes from the fits and from the exact optima**, same rule, same 20 training samples per cell, all 60 controlled cells.

B	comparison	win	tie	dense-preferred	agreement with fits
2	fitted softmax vs. fitted sparsemax	34	23	3	—
2	RATIO(2) vs. SIMPLEX (exact)	36	24	0	52/60
8	fitted softmax vs. fitted sparsemax	3	54	3	—
8	RATIO(8) vs. SIMPLEX (exact)	0	60	0	54/60

Table A13 sets the regimes from the fits beside the regimes from the exact optima, on the same samples and under the same rule. `results/convex_reference_full.json` holds the per-cell training gaps, test excess MSE for the four predictors, support F_1 and both regime labels.

N PERTURBING ONLY THE ZEROED POSITIONS

Registered margin. The header of `scripts/run_zeroed_position_test.py` was written before any checkpoint was loaded. It declares equivalence only if the 90% interval of $\Psi_{\text{zero}}/\Psi_{\text{signal}}$ over the 150 paired runs lies inside $[0, 0.01]$. In words: resampling only the inputs sparsemax excluded may move the prediction by less than one percent of what resampling the whole real context moves it. The same positions are perturbed in the softmax twin of the same setting and seed.

Procedure. For each frozen sparsemax model, the scale-1 attention weights on the test windows identify the positions with weight exactly zero (per window). Those positions are resampled $K = 16$ times from the training marginal with everything else held fixed; Ψ_{zero} is the mean prediction variance across resamples. Ψ_{signal} resamples every position. Both are computed for the sparsemax model and for its softmax twin on the identical position set.

The figure is Fig. 5 in the main text.

Outcome. Sparsemax: median 0.025, 90% interval $[0.0088, 0.112]$, against the registered finest-scale margin of 0.01. Softmax at the same positions: median 0.104, interval $[0.043, 0.248]$. Sparsemax is lower in 140 of 150 pairs (Wilcoxon $p = 1.5 \times 10^{-24}$), by a median factor of 3.3. The registered criterion, an interval inside $[0, 0.01]$, is the original test and its outcome is the interval above, reported as observed; the paired comparison and the strict pass below are follow-up analyses.

Strict definition. A second pass (`--strict`) counts a position as zeroed only if its weight is zero at the finest scale *and* in the $q = 3$ block and the $q = 6$ block containing it (or the corresponding scale is fused out). A position zeroed at every scale is rare by design: 86% of finest-scale zeros are read again by a coarser scale, and 23 of 150 pairs contain a fully zeroed position. Over all 150 pairs the strict interval is $[0, 0.0078]$. That interval reflects the empty sets rather than the model, because the 127 pairs with an empty strict set contribute a ratio of zero by construction. We therefore report the 23 non-empty pairs. On them the sparsemax model's dependence has median 0.006 (90% interval $[0.00002, 0.030]$) against 0.058 ($[0.0002, 0.39]$) for the softmax twin at the identical positions, lower in 23 of 23 pairs. The measurement says two things: a finest-scale zero removes about two thirds of a position's influence, and the multi-scale architecture keeps each position readable at some scale, which is its design intent.

A size-matched control. The registered statistic compares two perturbations: the zeroed positions, a median 16% of the window, against the whole window. Its scale therefore carries the size of the zeroed set. `scripts/run_zeroed_sizematched.py` removes that dependence: in the same frozen model, on the same windows, it perturbs an equal number of positions drawn uniformly at random per window and reports $\Psi_{\text{zero}}/\Psi_{\text{random}}$. A value below one means the positions sparsemax excluded matter less to the prediction than an equally large random set. Over the same 150 pairs the sparsemax models give a median of 0.174, with a 90% interval of $[0.033,\ 0.704]$ and a value below one in 149 of 150 pairs. The softmax twins at the identical positions give 0.631, interval $[0.417,\ 0.909]$. Sparsemax is lower in 142 of 150 pairs (Wilcoxon $p = 8.1 \times 10^{-25}$).

On this scale-free reading the whole 90% interval lies below the threshold, and the exact zeros cut a position's functional weight to about a quarter of the size-matched baseline while the dense twin keeps it near two thirds. The control is a follow-up analysis and is reported as such; `results/zeroed_sizematched.json` holds the per-pair values.

Where the residual sensitivity enters. `scripts/run_zeroed_path_decomposition.py` repeats the measurement with one path removed at a time. It uses the same 150 pairs, the same positions and the same $K = 16$ resamples. *Frozen weights* holds three things at their unperturbed values: the attention distributions, the scale-fusion weights and the target/cross gate. Only representations recompute. *Scale-1 only* sets the fusion weight to $(1, 0, 0)$, removing the $q = 3$ and $q = 6$ encoders.

Table A14: **Which path carries the influence of a zeroed position.** $\Psi_{\text{zero}}/\Psi_{\text{signal}}$ over the same 150 (setting, seed) pairs. The end-to-end row reproduces the main test.

condition	median	90% interval	pairs below 0.01	median reduction
end-to-end	0.0250	[0.0088, 0.1123]	12/150	—
frozen weights	0.0257	[0.0081, 0.1123]	16/150	5.4%
scale-1 only	0.0170	[0.0031, 0.2048]	48/150	26.0%
frozen weights + scale-1 only	0.0099	[0.0024, 0.0662]	76/150	59.5%

Three readings follow. First, movement of the attention weights is not needed for the residual dependence: with every weight held at its unperturbed value the group median is 0.0257 against 0.0250 end-to-end. Paired, the frozen condition is lower in 82 of 150 pairs with a median per-pair reduction of 5.4% (Wilcoxon $p = 0.03$), so weight movement may contribute but does not account for the effect. Second, the coarse scales are a measurable channel: removing them lowers the ratio in 104 of 150 pairs by a median 26% ($p = 5.7 \times 10^{-4}$), the functional counterpart of the observation that 86% of finest-scale zeros keep weight at a coarser scale. Third, with both accounted for the median falls to 0.0099 with a 90% interval of $[0.0024, 0.0662]$, and 76 of 150 pairs are individually below 0.01 ($p = 2.0 \times 10^{-16}$ against end-to-end). The registered criterion asks for the whole 90% interval to lie in $[0, 0.01]$; that interval is $[0.0024, 0.0662]$ here, so what these interventions establish is a reduction in sensitivity and the path it travels, not the registered functional-exclusion criterion. The size-matched control above is the reading on which the whole interval does clear its threshold. What remains is transport by the scale-1 recurrence, which reads a position before attention weights it. The measurement therefore locates the effect rather than only reporting it. A zero in a multi-scale recurrent model removes a position from one reading of the sequence, and the architecture supplies the others.

O A Prospective Rule From Training Data

The coordinate is computed before any test data is touched. The coordinates below come from each dataset's training split only (`scripts/prospective_coordinates.py`). r_{eff} is the participation ratio of the input correlation spectrum, the number of independent directions the groups span. k_{eff} is the number of the top r_{eff} principal directions that carry 90% of the ridge-regression mass when the target is regressed on them; $k_{\text{eff}}/r_{\text{eff}}$ is the *coordinate*: the relevant share among independent directions, in $[0, 1]$. $\widehat{\sigma}_z$ is the root ratio of off-support to on-support eigenvalues. The regime is read from the wide-bound map of Fig. 1b, whose wins and dense-preferred cells occur only at $n \leq 128$. The observed column of Table A15 comes from the matched comparisons reported in this paper (unbounded scores).

Table A15: **Prospective reading of the map on the eight datasets of the original benchmark.** All coordinates from training data; the prediction is made before the test split is touched. The 798 low-coordinate cells added later are in Appendices K and J.

Dataset	G	r_{eff}	k_{eff}	$k_{\text{eff}}/r_{\text{eff}}$	$\widehat{\sigma}_z$	n_{train}	Pred.	Obs.
ETTh1	84	7	5	0.71	0.62	12,060	tie	tie
ETTh2	84	4	3	0.75	0.45	12,060	tie	tie
ETTm1	84	6	3	0.50	0.65	48,642	tie	tie
ETTm2	84	4	3	0.75	0.45	48,642	tie	tie
METR-LA	288	3	3	1.00	0.00	23,965	tie	tie
Cryptocurrency (BTC-P2)	30	2	1	0.50	0.44	6,225	tie	tie
Semiconductor panel (monthly)	52	2	2	1.00	0.00	258	tie	tie
Semiconductor equities (daily)	144	17	11	0.65	0.36	2,788	tie	tie

Two things follow. The rule is prospective and it is right on each of these eight datasets. And it says why: read on the representations these benchmarks present, the inputs span very few independent directions and nearly all of them carry signal, which places all eight in the all-relevant row where the map has always predicted a tie. Separating the map from the hypothesis that normalization never matters requires the opposite geometry, many independent directions of which few are relevant, at a moderate sample size. No *representation used in this benchmark* has it, which is why we first built one semi-synthetically (Appendix I). It is available in real data: changing what the attention is computed over, without changing the data, moves the coordinate to 0.02–0.47 on 798 cells across two domains (Appendices K and J), and exact zeros are lower on 773 of them.

P Registered Predictions for the Semiconductor Test

Section 6 relies on predictions we wrote down before running the experiments that test them. This appendix reproduces the registered text for every prediction this paper cites, with its date and its verdict, so the claim of prior registration can be read rather than taken on trust. The full registration document is supplied as supplementary material (`PREREGISTRATION_daily.md`).

Registration date. 2026-07-26, headed "written before any daily experiment"; the 5-minute and 1-minute addenda carry the same date and the same clause.

P2 (registered). "The matched sparsemax-vs-softmax contrast remains a tie (support on daily financial series is not learnable, as in the cryptocurrency domain)." Falsifier, registered in the same line: "a significant sparsemax win falsifies P2 and the boundary rule itself."

P3 and P4 (registered). At 5-minute and 1-minute resolution "the matched sparse-vs-soft contrast is still predicted to tie (P2 carries over)."

P7a (registered before the data were downloaded). Across an expanded universe of listed semiconductor firms, "the matched sparse-soft tie replicates: no systematic direction, magnitudes within seed noise." Falsifier: "a systematic sparse-soft gap." Table A16 gives the verdict on each.

Table A16: **Verdicts on the registered tie predictions.** Matched sparsemax–softmax relative NMSE difference; all four are inside run noise and none reaches the registered falsifier.

Prediction	Rung	n (train)	|difference|	Verdict
P2	monthly production panel	250	5.1% ($p = 0.58$)	held
P4	1-minute equities	$\approx$ 1,900	1.1%	held
P3	5-minute equities	$\approx$ 3,250	1.5%	held
P2	daily equities	$\approx$ 5,000	0.0%	held

The registration also states, and the data confirm, a directional prediction that the attention forecaster's rank against recurrent baselines improves with the number of training windows (P1); its verdict is reported in Section 6.

Q Normalization Swap Inside a Transformer

The tie survives inside a Transformer with unbounded logits. Every other swap in this paper sits in the linear probe or in the BiGRU-based MISA. This one uses PatchTST (Nie et al., 2023) on the 15 representative cryptocurrency settings, ten seeds each, with the data pipeline, splits, optimizer and schedule of the frozen-model experiment (`run_patchtst_swap.py`). The encoder layer is re-implemented so the attention normalization can be exchanged; at softmax it matches `nn.MultiheadAttention` to 3×10^{-16}, and the α-entmax gradient is verified against finite differences. Attention logits are not clipped. Table A17 reports the three normalizations.

Table A17: **PatchTST with three normalizations, 15 settings $\times$ 10 seeds.** Paired comparisons are against softmax on the same setting and seed. "Mean rel. diff" is $(\mathrm{NMSE}_{\mathrm{softmax}} - \mathrm{NMSE}_{\mathrm{arm}})/\mathrm{NMSE}_{\mathrm{softmax}}$ averaged over the 150 pairs, so a positive value favors the sparse normalization; the mean and the pair count can disagree in sign when the distribution is skewed.

Normalization	Mean NMSE	Median NMSE	Lower than softmax (p)	Mean rel. diff.	Exact zeros
softmax	0.2334	0.2399	—	—	0.0%
α-entmax (1.5)	0.2332	0.2365	80/150 (0.70)	-0.39%	0.0%
sparsemax	0.2334	0.2376	74/150 (0.73)	-0.47%	0.4%

Per setting, the sparsemax–softmax difference ranges from -2.1% (LTC-P1) to $+3.2\%$ (BTC-P2) with no consistent sign. The exact-zero share is the striking number: with unbounded logits and six patches per sequence, sparsemax leaves 99.6% of its weights positive: with no irrelevant inputs to exclude, the trained model keeps them, which is the mechanism the frozen-model diagnostic identifies.

R Expanded Cryptocurrency Evaluation

Every valid asset-period result is released. Table A6 summarizes the benchmark: the assets, the three windows, and the record counts.

The file `expanded_crypto_139.csv` contains every valid asset-period result used in the expanded evaluation. It includes 45 settings in P1, 47 in P2, and 47 in P3. ARB and VET lack sufficient P1 history, which gives 139 settings rather than 141. The file reports normalized mean squared error (NMSE), the Harvey-corrected Diebold–Mariano (DM) statistic from the seed-42 prediction path, raw win indicators, and relative NMSE differences. The DM statistic asks whether one model's prediction errors are consistently smaller than the other's over the same test sequence. Positive values favor Multi-scale Inter-signal Sparse Attention (MISA).

Each 30-day window is split chronologically into 70% training, 10% validation, and 20% test observations. Normalization is fitted only on the training portion. The input contains 30 five-minute bars, and the target contains the next five bars. Raw open, high, low, close, and volume (OHLCV) bars are converted to seven features. Features with absolute correlation at least 0.95 with the close are removed as near duplicates. For feature pairs with absolute correlation above 0.90, the less target-predictive feature is removed. Principal-component importance then ranks the remaining features. The same procedure is used for the target and BTC leader streams.

Table A18: Results by market period. "Lower" = settings where MISA has lower mean NMSE; DM W/L = unadjusted significant wins/losses at $p < .05$; median gain = median relative NMSE difference.

Baseline	Period	Settings	Lower	DM W/L	Median gain
LSTM	P1	45	33	7/1	$+0.63\%$
	P2	47	27	12/3	$+0.39\%$
	P3	47	28	12/0	$+0.39\%$
	All	139	88	31/4	$+0.53\%$
GRU	P1	45	37	6/2	$+1.47\%$
	P2	47	33	7/2	$+0.70\%$
	P3	47	36	10/1	$+0.97\%$
	All	139	106	23/5	$+1.04\%$

To reduce repeated-period dependence, we average the relative NMSE difference within each asset before testing. This gives 47 values. The median asset-level gain is 0.58% over the long short-term memory (LSTM) model and 1.27% over the gated recurrent unit (GRU). One-sided exact Wilcoxon signed-rank tests give $W = 813$, $p = 0.0038$ for LSTM and $W = 1013$, $p = 1.2 \times 10^{-7}$ for GRU. The file `expanded_crypto_asset_summary.csv` contains these 47 aggregated values. Assets still share market periods and a common BTC leader. The tests therefore remove within-asset repetition; cross-asset dependence is why inference is aggregated at the asset level.

S Foreign-Exchange DM Results

Table A19: Harvey-corrected DM statistics for MISA against each baseline (seed-42 foreign-exchange predictions; positive favors MISA; single-seed, exploratory). $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$; ns = not significant.

Pair	Period	LSTM	iTransformer	PatchTST	Informer
EUR/USD	F1	-1.99^{ns}	$+3.42^{***}$	$+3.68^{***}$	$+3.58^{***}$
	F2	$+0.29^{\mathrm{ns}}$	$+4.88^{***}$	$+4.75^{***}$	$+2.49^{*}$
	F3	-0.45^{ns}	$+6.65^{***}$	$+3.77^{***}$	$+4.84^{***}$
GBP/USD	F1	-0.44^{ns}	$+3.91^{***}$	$+8.52^{***}$	$+4.60^{***}$
	F2	-1.10^{ns}	$+5.21^{***}$	$+2.67^{**}$	$+4.74^{***}$
	F3	-1.09^{ns}	$+6.97^{***}$	$+6.24^{***}$	$+5.11^{***}$
JPY/USD	F1	-0.01^{ns}	$+3.14^{**}$	$+3.78^{***}$	$+3.52^{***}$
	F2	-1.29^{ns}	$+1.96^{\mathrm{ns}}$	$+0.44^{\mathrm{ns}}$	$+1.99^{*}$
	F3	$+2.84^{**}$	$+3.31^{**}$	$+3.29^{**}$	$+2.65^{**}$

T Matched Neural Ablation

Figure A8 and Table A20 give the matched ablation over eight asset-period settings. The parameter count, optimizer, split, and training schedule are held fixed. Removed components are replaced with parameter-matched linear projections. "MISA W/L" counts significant wins and losses of the full model against each variant using the seed-42 DM paths.

Table A20: Matched ablation and same-CPU epoch time.

Variant	Parameters	s/epoch	Mean NMSE	MISA W/L
MISA	228,866	6.43	0.2260	–
Softmax	228,866	6.41	0.2263	1/3
Single scale	228,866	4.99	0.2256	2/2
No cross-signal gate	228,866	3.99	0.2289	2/0

The softmax replacement is the key control: sparsemax alone leaves real-data forecasts unchanged, consistent with placing this benchmark in the tie region. The cross-signal gate has the clearest one-directional contribution in this small ablation. Mean temporal sparsity is 22%, ranging from 12% in P1 to 35% in P2. This is an observed weight statistic; the functional relevance of zeroed steps is tested in Section 6.

U Classical and Continuous-Time Baselines

Two further baselines were run on the 15 representative asset–period settings. The liquid network is a closed-form continuous-time (CfC) recurrent cell (Hasani et al., 2022) with hidden size 64, trained with the MISA protocol (Adam, learning rate 5×10^{-4}, dropout 0.20, 30 warm-up epochs, at most 200 epochs, early stopping after 25; seeds 42–44). The classical baseline fits an AR(1) mean with GARCH(1,1) conditional variance on the training log-return series and issues rolling five-step cumulative return forecasts on the test split; it has no random seed. DM statistics use the seed-42 MISA prediction path and are positive when GARCH has the lower loss. Scripts: `run_lnn.py`, `run_garch.py`.

Table A21: Mean NMSE on the 15 representative settings: MISA (seeds 42–44), liquid CfC network (seeds 42–44), and AR(1)–GARCH(1,1); lowest in bold. DM: Harvey-corrected Diebold–Mariano statistic, GARCH vs. MISA, positive favors GARCH; * $p < .05$.

Asset	Period	MISA	CfC	GARCH	DM	sig.
BTC	P1	0.0471	0.0468	**0.0462**	+1.79	ns
BTC	P2	0.0802	0.0824	**0.0712**	+1.98	*
BTC	P3	0.2856	0.2811	**0.2703**	+2.16	*
ETH	P1	0.0657	0.0659	**0.0655**	+0.26	ns
ETH	P2	0.2371	0.2366	**0.2297**	+0.80	ns
ETH	P3	0.1449	0.1462	**0.1413**	+1.35	ns
XRP	P1	0.1215	0.1218	**0.1201**	+0.79	ns
XRP	P2	0.3524	**0.3485**	0.3488	+0.24	ns
XRP	P3	0.1689	0.1731	**0.1664**	+2.02	*
LTC	P1	0.2294	0.2298	**0.2194**	+2.04	*
LTC	P2	0.4351	0.4319	**0.4297**	+1.04	ns
LTC	P3	0.3582	**0.3562**	0.3597	−0.04	ns
DOGE	P1	**0.1314**	0.1316	0.1322	−1.20	ns
DOGE	P2	0.4233	0.4241	**0.4175**	+0.81	ns
DOGE	P3	0.2847	**0.2842**	0.2843	−0.20	ns

MISA and the liquid network are statistically tied (CfC lower in 8 of 15, median relative difference 0.15% in MISA's favor), so the continuous-time inductive bias neither adds nor removes residual noise beyond what the BiGRU already does. GARCH has the lower NMSE in 13 of 15 settings (median advantage 1.4%, largest 12.6% at BTC-P2, four significant). Both results are consistent with the map. At this horizon the predictable component of five-minute returns is mostly conditional variance, which the GARCH recursion models explicitly and no attention normalization can extract by selecting among inputs. The deep models' advantage over one another is of the same order as their gap to GARCH. The tie claim of Section 6 rests on the matched swap and the injected-noise probe, which are unaffected by the presence of a strong classical reference.

V Representative Cryptocurrency NMSE Values

Choice of comparison. Against the average Transformer loss instead, MISA wins all 15 settings (median reduction 2.86%); we report the more demanding lowest-loss comparison. Across all six models, MISA is best or tied in ten settings, a recurrent baseline in three, and a Transformer in two. The largest gains (4.95% XRP-P3, 3.69% LTC-P1, 2.95% ETH-P1) show no trend from P1 to P3, and the periods do not identify the theoretical quantities r, σ_z^2, or B.

Table A22: Mean cryptocurrency NMSE over seeds 42–44; lowest value per setting in bold.

Asset	Period	MISA	LSTM	GRU	iTransformer	PatchTST	Informer
BTC	P1	0.0471	0.0474	**0.0470**	0.0479	0.0496	0.0484
	P2	**0.0802**	0.0895	0.0923	0.0907	0.0803	0.0863
	P3	0.2856	**0.2830**	0.2916	0.2861	0.2877	0.2988
ETH	P1	**0.0657**	0.0665	0.0674	0.0679	0.0683	0.0677
	P2	**0.2371**	0.2453	0.2414	0.2419	0.2520	0.2468
	P3	**0.1449**	0.1481	0.1461	0.1482	0.1484	0.1509
XRP	P1	**0.1215**	0.1224	**0.1215**	0.1220	0.1230	0.1232
	P2	**0.3524**	0.3538	0.3549	0.3541	0.3638	0.3666
	P3	**0.1689**	0.1740	0.1737	0.1777	0.1795	0.1851
LTC	P1	**0.2294**	0.2333	0.2328	0.2382	0.2454	0.2414
	P2	0.4351	0.4339	0.4358	**0.4324**	0.4420	0.4375
	P3	**0.3582**	0.3713	0.3589	0.3599	0.3686	0.3635
DOGE	P1	**0.1314**	0.1320	0.1327	0.1340	0.1340	0.1327
	P2	0.4233	0.4376	0.4401	0.4242	0.4255	**0.4231**
	P3	0.2847	**0.2846**	0.2895	0.2912	0.2953	0.2941

W PROBE-FORECASTER BENCHMARK DETAIL

Relative advantage over the strongest Transformer. Let $\mathcal{T} = \{$iTransformer, PatchTST, Informer$\}$ and let $L_{m,c}$ be the mean NMSE of model m in setting c. We report $b_c = \min_{m \in \mathcal{T}} L_{m,c}$ and $d_c = (b_c - L_{\mathrm{MISA},c})/b_c$; a positive d_c means MISA has the lower error in that setting.

The probe is competitive, which is all it needs to be. This appendix holds the benchmark detail moved out of Section 6. It supports the claim that MISA is a usable probe; the paper's contribution is the boundary, not this comparison.

MISA is competitive with established baselines, qualifying it as a probe. It attains lower NMSE than the best-performing Transformer in 13 of 15 representative cryptocurrency settings ($W = 108, p = 0.0021$) and outperforms the LSTM and GRU in 88 and 106 of the 139 expanded settings ($p = 0.0038$ and $p = 1.2 \times 10^{-7}$, respectively). Conversely, AR(1)–GARCH(1,1) achieves lower error than MISA in 13 of 15 settings (Table A21): at five-minute intervals, predictability is dominated by conditional variance, which lies outside what input selection addresses.

Against the best tested Transformer in each of the 15 representative settings, the median reduction is 0.98%, with interquartile range 0.29–2.23% and mean 1.45% (one-sided exact Wilcoxon $W = 108$, $p = 0.0021$). The two increases are 0.62% for LTC-P2 and 0.05% for DOGE-P2 (Fig. A6; complete NMSE values in Table A22). MISA is lower in 43 of 45 direct comparisons against the three Transformers and tied with the liquid network (lower in 8 of 15). AR(1)–GARCH(1,1) is lower than MISA in 13 of 15 settings (median 1.4%, four significant; Table A21). The gains are repeated but small, the norm where predictability is scarce and percent-level edges carry economic weight (Gu, Kelly, and Xiu, 2020).

Across the 139 valid settings the seed-42 DM paths give 31/4 significant wins/losses against the LSTM and 23/5 against the GRU, unadjusted. Averaging across periods within each asset (Fig. A7) gives the asset-level medians quoted in the main text. Shared periods and a common leader series can induce dependence, so effect sizes matter as much as p-values; Appendix AB states the full statistical scope.

X BOUNDARY TESTS ON REAL DATA

Section 6 summarizes three planned tests that place the cryptocurrency benchmark on the map; their practical-equivalence margin was fixed retrospectively, and Algorithm 1 is what fixes it in advance on a new dataset. This appendix gives their full details, together with three follow-up analyses: a temporal-dependence sweep, the setup of the neural phase map, and a score-separation repair. All are reproducible from the released scripts.

X.1 Full matched sparsemax–softmax comparison (all 139 settings)

The matched ablation in the main text covers eight settings. Here we extend the identical matched comparison to *all* 47 assets across the three periods (139 valid asset–period settings). The contrast is sparsemax versus softmax with the multi-scale and cross-signal components held on. Parameters, optimizer, split, schedule, and seed 42 are identical. Raw NMSE is not comparable across assets with very different price scales. We therefore aggregate the scale-free relative difference $(\mathrm{NMSE}_{\mathrm{soft}} - \mathrm{NMSE}_{\mathrm{sparse}})/\mathrm{NMSE}_{\mathrm{soft}}$. A positive value favors sparsemax.

Sparsemax attains lower error in 61 of 139 settings. The mean relative difference is -0.17%, and the median is -0.08%; a one-sided exact Wilcoxon test returns the predicted tie ($p = 0.94$). To measure the precision of that tie we average periods within asset. Across 47 asset-level differences the mean is -0.15%; the 90% t interval is $[-0.38\%, 0.08\%]$ and an asset-cluster bootstrap gives $[-0.37\%, 0.07\%]$. Retrospective TOST sensitivity rejects non-equivalence at $\pm 0.5\%$ ($p = 0.007$) and $\pm 1.0\%$; the $\pm 0.25\%$ margin is the target of the prospective protocol. The sign is balanced across assets (23 favor sparsemax, 24 favor softmax). The margins are stated as a sensitivity analysis; the confirmatory margin is fixed in Algorithm 1. Grouping assets from blue-chip (C1) to speculative (C5) reveals *no* noise gradient (Fig. A9).

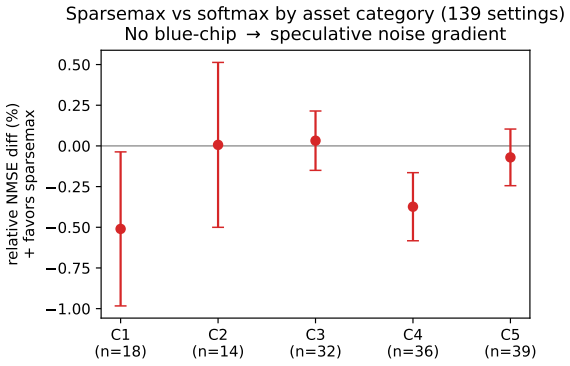

Figure A9: **No category noise gradient.** Mean relative NMSE difference (sparsemax vs. softmax) by asset category over 139 matched settings; bars are standard errors. Positive favors sparsemax. The blue-chip→speculative axis shows no trend, contrary to the hypothesis that exact-zero selection should help more on noisier assets.

The matched comparison was repeated with ten seeds (42–51) on the five representative assets, 300 trained models in all (`run_multiseed_matched.py`). Sparsemax attains the lower ten-seed mean NMSE in 4 of 15 settings. The setting-balanced mean relative gap is -0.10% with a 90% t interval $[-0.30\%, 0.11\%]$. A hierarchical bootstrap that resamples settings and paired seeds gives $[-0.50\%, 0.31\%]$ at 90%; retrospective TOST at $\pm 0.5\%$ gives $p = 0.002$. Individual settings have wider intervals (4/15 inside $\pm 0.5\%$, 9/15 inside $\pm 1\%$), so the tie is an aggregate, benchmark-level statement. In all 15 settings the seed-to-seed standard deviation exceeds the mean gap between the two cells.

X.2 Robustness to input normalization

One might suspect the result is caused by the choice of input scaling. We therefore re-ran the eight-setting matched comparison with z-score inputs (train-fit) instead of min–max scaling. Nothing else changed. The sparsemax main effect did not reverse. It moved from -0.00101 (min–max) to -0.00387 (z-score) in raw NMSE. Softmax's small edge *grew*. Sparsemax was better in $2/8$ rather than $3/8$ settings. The result is not caused by the input scaling.

X.3 INJECTING KNOWN IRRELEVANT INPUTS

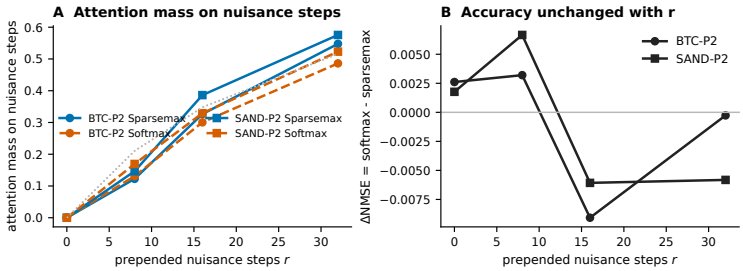

Figure A10: **Injected-noise stress test (temporal-attention core).** The earlier probe, retained here for completeness; the main text measures the same question in the complete model (Fig. 4). In the core, mass on nuisance positions is comparable across normalizations and accuracy is unchanged as nuisance is added; the BiGRU mixes positions before attention, so functional exclusion is measured in the complete model.

The cleanest test of the mechanism uses ground truth. MISA's exact-zero selection acts over time steps. We append r steps of pure noise to each real 30-step window; the injected raw steps are known to be target-irrelevant. We reduce the model to its temporal-attention core (single scale, no cross-signal branch). However, the BiGRU mixes positions before attention and the attention scores depend on the same input. Consequently this recurrent probe is outside the independence conditions of Theorem 1. We use it as a stress test, not as an exact theorem instantiation: if exact-zero raw-position weighting were the operative mechanism, one would expect sparsemax nuisance mass and/or prediction sensitivity to fall relative to softmax as r grows.

Once the model is trained to convergence the temporal core alone sits in the tie region (Fig. A10). In the core, attention mass on the injected steps is comparable across the two normalizations; the complete model allocates less under sparsemax (Fig. 4). Accuracy is unchanged: the NMSE difference stays near zero across r. The lesson is Theorem 3: the risk benefit of sparsemax is conditional on the scores identifying the irrelevant support, and in the trained GRU that identification is done by the encoder before attention.

X.4 IS THE PHASE MAP REPRESENTATIVE OF TIME SERIES?

The controlled rows are i.i.d. Real series are not. We therefore repeated the matched sweep under two violations. First, AR(1) inputs with autocorrelation coefficient $\phi \in \{0.5, 0.9\}$, where training rows are consecutive time steps. Second, heavy-tailed noise (t_3, unit variance) at $\phi = 0.5$. The grid uses $k \in \{2, 8, 32\}$, $\sigma_z \in \{0.5, 2\}$, and $n = 512$ with 20 seeds (720 fits). The structure of the map is unchanged. Sparse support ($k = 2$) still wins under every violation. Dense support ($k = G$) still ties. Only the boundary cell ($k = 8$) moves, in the direction the sample-size axis predicts: at $\phi = 0.9$ support F1 goes from 1.00 to about 0.95. Strong dependence shrinks the effective sample size to roughly $n(1 - \phi)/(1 + \phi) \approx 27$, below the recovery threshold. Dependence therefore rescales the sample size inside the rule. It does not change the rule. Heavy tails change almost nothing.

X.5 Setup of the neural phase map

Table A23: **Neural phase map.** The temporal-attention core of MISA trained on synthetic windows with known support (five seeds; setup in Appendix X.5). "diff" = excess-MSE difference, softmax minus sparsemax (positive favors sparsemax); "attn" = attention mass on irrelevant steps.

n	k/G	σ_z	diff	$\text{attn}_{\text{sparse}}$	$\text{attn}_{\text{soft}}$	regime
512	2/32	0.5	-0.0011	0.007	0.314	tie
512	2/32	2	$+0.0061$	0.000	0.152	**win**
512	8/32	0.5	$+0.0008$	0.082	0.229	**win**
512	8/32	2	$+0.0060$	0.006	0.249	**win**
512	32/32	0.5	-0.0001	0.000	0.000	tie
512	32/32	2	-0.0001	0.000	0.000	tie
2048	2/32	0.5	-0.0004	0.000	0.298	dense-preferred
2048	2/32	2	$+0.0014$	0.000	0.044	**win**
2048	8/32	0.5	$+0.0010$	0.028	0.161	**win**
2048	8/32	2	$+0.0026$	0.000	0.024	**win**
2048	32/32	0.5	0	0.000	0.000	tie
2048	32/32	2	0	0.000	0.000	tie

Table A23 uses the temporal-attention core of MISA. The encoder is a single-scale BiGRU with hidden size 64. Attention over time steps uses sparsemax or softmax. The cross-signal branch is off. Each window has $T = 32$ scalar steps. The first k steps are drawn from $\mathcal{N}(0, 1)$ and generate the target $y = k^{-1} \sum_{t \le k} x_t + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 0.05^2)$. The remaining steps are $\mathcal{N}(0, \sigma_z^2)$ noise. The support positions are fixed across windows, so they are learnable. Training follows the real-data protocol (Adam, learning rate 5×10^{-4}, batch size 64, early stopping). Each cell trains five seeds per normalization and evaluates on 20,000 fresh windows. Cells are classified with the same two-standard-error rule as the linear map. The script `run_neural_phase_map.py` regenerates the table.

X.6 Restoring exclusion with a score-separation penalty

In the trained core the scores stay dense. We therefore added a sharpening penalty $\lambda_a \, (1 - \sum_j a_j^2)$ to the training loss of the temporal core. The term pushes the attention distribution away from uniform. The penalty forces a sharper raw-position support pattern. With $\lambda_a = 0.1$, attention on the injected steps drops from 0.12–0.58 to 0.000 at every r on both assets, while accuracy is unchanged: no monotone gap opens with r. This shows that raw attention exclusion can be forced while accuracy stays at the tie the map predicts; which layer absorbs the nuisance is measured by the frozen full-model nuisance-resampling diagnostic.

X.7 Interpretation

These results delimit the theory rather than extending its assumptions. Sparsemax wins in the controlled predictor where the score support is directly learnable. On the cryptocurrency benchmark, the matched normalizations have a small aggregate accuracy gap, as the map predicts. Because a BiGRU can transport information across positions, functional exclusion is measured at the model level rather than from attention mass. The encoder, value path, readout, or cancellation between paths may each suppress nuisance variation.

The aggregate accuracy evidence is consistent with practical equivalence under the $\pm 0.5\%$ margin at the aggregate level; the $\pm 0.25\%$ margin is the prospective target, and the claim is made at the benchmark level. Direct attribution of the tie requires the frozen full-model nuisance-resampling diagnostic described in the main text.

Y Standard-Benchmark Check

The tie the map predicts is the tie a fair comparator returns. To test the phase boundary on standard benchmarks without confounding deep encoder dynamics, we evaluate ETTh1, ETTh2, ETTm1, ETTm2, and the 207-sensor METR-LA network within the linear model class of Definition 1 ($d =$

1). Contexts of length 96 across C channels are pooled into $P = 12$ patches, yielding $G = CP$ input groups ($G = 84$ for ETT; $G = 2484$ for METR-LA). The predictor is $\widehat{y} = u\,(p(s)^\top x) + b$ with scores constrained to $[-2, 2]$ and scalar readout parameters (u, b) across five rolling-origin folds (475 fits).

Throughout this appendix the relative difference is $(\mathrm{NMSE}_{\mathrm{sparsemax}} - \mathrm{NMSE}_{\mathrm{softmax}})/\mathrm{NMSE}_{\mathrm{softmax}}$, so a negative value means sparsemax has the lower error. The 95 cells are the 80 ETT cells of `results/boundary_benchmarks.json` together with the 15 207-sensor METR-LA cells of `results/boundary_metr207.json`; the METR-LA rows inside the first file are a smaller preliminary run at $G = 288$ and are not part of this count.

The empirical phase map classifies all five datasets into the tie region. With the scores unconstrained the median relative difference is 0.00% (mean -1.2%, one-sided Wilcoxon $p = 0.23$). Under bounded scores ($B = 2$) sparsemax has the lower error by 15.2% on average, median 14.5%, lower in 76 of 95 cells ($p = 6.9 \times 10^{-10}$, Fig. A12a). The margin scales with context size: 35.9% on METR-LA ($G = 2484$, lower in 15 of 15 cells) against 3.1% on ETT at horizon 720 (Fig. A12b). Under bounded scores, softmax allocates at most $[1 + (G-1)e^{-2B}]^{-1}$ to a single group (0.397 for $G = 84$; 0.022 for $G = 2484$). Once scores are unconstrained, softmax matches sparsemax peak concentration (median maximum weight 0.412 versus 0.434), and the performance difference disappears.

This check is reported for scope rather than as the paper's real-data evidence. It holds the predictor inside Definition 1 rather than inside a trained forecaster, so the quantity compared is the α contrast, not absolute forecast accuracy. What it establishes is that the tie the map predicts for these coordinates is the tie a fair comparator returns.

Z Sensitivity to the Score Bound

Standard error convention. Both normalizations are fit on the same sample within a seed, so SE is the standard error of the 20 paired differences, $\mathrm{sd}(\Delta_{\mathrm{seed}})/\sqrt{20}$. Classifying with the pooled independent-samples formula instead leaves every cell's label unchanged at every bound in Table A24, because the 10^{-4} floor, not the standard error, is the binding condition.

Table A24: **Regime counts across score clipping bounds.** All 60 controlled cells, same rule as Fig. 1. "Analytical leak" is the largest closed-form dense-side term $L^2/k + \sigma_z^2 L^2/r$. The win region concentrates as B grows; the small-sample corner does not move.

B	α=1.25	α=1.5	α=1.75	α=2.0	Max Δ	Leak
	win / tie / dense-preferred					
2	36/24/0	32/28/0	36/24/0	34/23/3	2.9×10^{-2}	2.9×10^{-2}
3	17/43/0	17/42/1	17/43/0	17/40/3	8.4×10^{-4}	8.1×10^{-4}
4	3/57/0	3/55/2	4/56/0	4/53/3	2.2×10^{-4}	1.6×10^{-5}
8	0/59/1	1/58/1	3/57/0	3/54/3	1.9×10^{-4}	1.8×10^{-12}

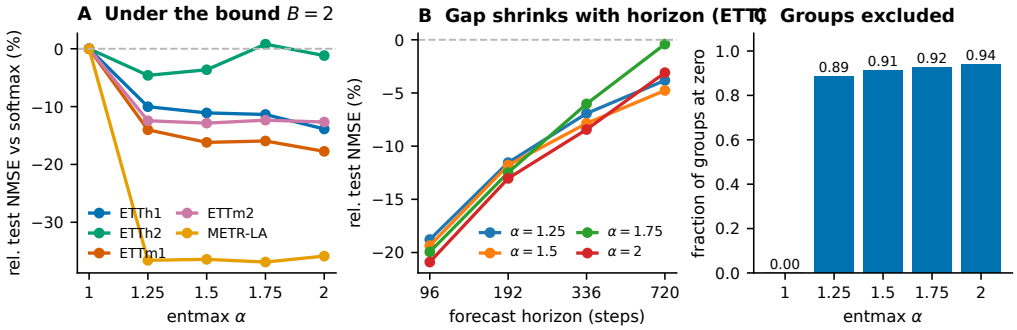

Figure A12: **Matched α comparison on standard benchmarks.** 95 paired cells: five datasets, four horizons (three for METR-LA), five rolling-origin folds, inside the model class of Definition 1. (a) Mean test NMSE relative to softmax under the bounded scores of Section 4; released from the bound the same cells give a median of 0.00% (Appendix Z). (b) The same contrast against horizon, ETT pooled. (c) Fraction of the G input groups receiving exactly zero weight.

Figure A11 shows the same 60 cells at both bounds. The controlled study clips scores to $s_j \in [-B, B]$ with $B = 2$. The bound binds the softmax arm and is slack for sparsemax, which reaches a vertex from a score gap of one, so it is the one setting that could by itself manufacture a win region. We therefore repeat the whole α sweep at $B \in \{3, 4, 8\}$, 6,000 matched fits at each bound, changing nothing else (`run_entmax_sweep.py --score-bound`).

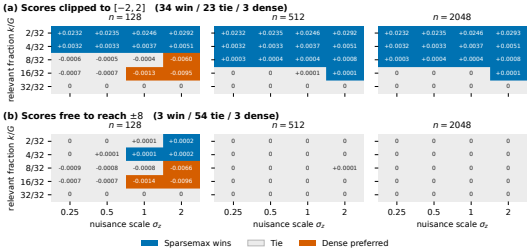

Figure A11: **The same 60 cells at two bounds.** (a) scores clipped to $[-2, 2]$: 34 wins. (b) the wide bound ± 8: the win region concentrates on the sparsest supports, and the small-sample corner is identical in both.

Three things follow, and we state them as the scope of the phase map.

The win region concentrates as B grows. From 34 cells at $B = 2$ it narrows to the sparsest supports, $k/G \leq 1/8$ (Table A24). At $B = 2$ and $B = 3$ the closed-form leak of Appendix G predicts the largest observed difference to within 4%, which shows that in that range the map is predicted by the leak term. A reader should take the *ordering* of the map, not the extent of its blue region, as the data-dependent claim.

What survives large B is the estimation mechanism. At $B = 4$ and $B = 8$ the largest observed difference, 2.2×10^{-4} and 1.9×10^{-4}, exceeds the closed-form prediction by one and eight orders of magnitude respectively. Whatever sustains a win at sparse supports once the bound is loose is not the residual-noise term of Theorem 1. The natural candidate is the difficulty of driving a softmax toward a vertex by gradient descent. This paper does not measure that and does not claim it.

The benchmarks behave the same way. We repeat the 95-cell benchmark comparison of Section Y at $B = 8$, where a softmax can place 0.9997 on a single group, changing nothing else. The sparsemax–softmax difference moves from -15.2% (median -14.5%, 76 of 95 cells, $p = 6.9 \times 10^{-10}$) to -1.2% (median 0.00%, 46 of 95, $p = 0.23$), which is the tie the map predicts for these coordinates. The movement is entirely on the softmax side: its mean test NMSE improves from 1.200 to 1.035 while sparsemax is unchanged at 1.017, and the freed softmax reaches the same concentration as sparsemax, median largest weight 0.412 against 0.434. Exact zeros are therefore worth most where a dense normalization cannot concentrate, and the score range is what decides that — which is the second axis the map gains from this experiment.

The small-sample corner is invariant. At every bound, dense attention is preferred in exactly the cells $(k, n) = (8, 128)$ and $(16, 128)$, and interior α cedes at most two cells. These cells sit where

Theorem 3 places the support-recovery cost, at the smallest sample relative to the support margin, and they do not move with the score range. They are a property of the fitted procedure rather than of the model classes. The exact convex references of Appendix M return a tie in all of them, and the optimization probe of Table A12 shows the zero-initialized fit stopping short of the convex optimum there. The statement that transfers is therefore about this initialization and this optimizer, not about any score range.

AA Extended Controlled Grid and Map Transport

The controlled grid of Section 4 stops at $n = 2048$. To cover the sample sizes of the benchmark study we reran the α sweep with $n \in \{128, 512, 2048, 8192, 32768\}$, 10,000 matched fits in total (`run_entmax_sweep.py`, `results/entmax_sweep_extended.json`).

Dense attention is preferred only at the smallest sample. For every α, no cell prefers dense attention once $n \geq 512$; the three such cells of Fig. 1 occur only at $n = 128$. Exact-zero selection is exposed by scarce data, not by the coordinate alone.

The gap saturates. At $k/G = 1/2$ the mean difference against softmax settles by $n = 512$ and does not move thereafter: for $\alpha = 2$ it is $+1.5 \times 10^{-5}$, $+2.1 \times 10^{-5}$, $+2.2 \times 10^{-5}$ and $+2.2 \times 10^{-5}$ at $\sigma_z = 0.25$ for $n = 512$ through 32,768, and $+1.0 \times 10^{-4}$ at $\sigma_z = 2$ across the same range. At $k = G$ the paired difference stays at the 10^{-10} level. Additional data removes the support-recovery cost; it does not create a gain where the noise floor is small.

Transport to correlated data. Benchmark cells report $k/G \approx 0.60$ and $\widehat{\sigma}_z \approx 0.93$. At those coordinates the extended grid gives a tie. Under bounded scores the benchmarks instead showed a 15% gain; released from the bound they tie (Appendix Y), so the grid's prediction holds once the comparator is fair. Resolving individual cells is the direction a redundancy-aware coordinate opens, and the reason is the input geometry. The grid varies independent coordinates (participation ratio 31.95 of 32, mean absolute off-diagonal correlation 0.005). The benchmark groups are redundant: 6.3 effective directions among the 84 ETTh1 groups (mean $|r| = 0.231$) and 3.3 among the 2484 METR-LA groups (mean $|r| = 0.536$). Under that redundancy, exact-zero selection prunes near-duplicate directions rather than independent residual noise. That is the regime Remark 2 describes. A redundancy-aware coordinate would give the map cell-level resolution; we leave it open.

AB Statistical Scope

The 139 setting-level DM tests use one seed, and their win–loss counts are descriptive. The asset-level Wilcoxon test is the broader paired summary because it aggregates the three periods before testing. The three training seeds are used for mean NMSE, but they are not a substitute for more independent training repetitions. These limits are stated in the main paper.