LEARNING MONOTONE INFORMATION TIME FOR FIXED-BUDGET INTRADAY REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study representations of intraday market and order-book data for predicting stocks' relative next-day close-to-close returns. Uniform tokenization partitions 240 one-minute records at a fixed temporal resolution, although trading activity varies across intervals, stocks, and trading dates. We introduce a tokenizer that learns sample-dependent temporal coverage under a fixed token budget. A local causal encoder assigns a positive information rate to each minute; normalized cumulative rates define a monotone time coordinate. Soft aggregation around a fixed number of equally spaced centers in this coordinate yields tokens with adaptive spans in natural time while preserving chronological order. The rates are learned jointly with the prediction objective, without supervision of discrete boundaries. A causal convolutional Mixer and a constrained pooling head integrate these representations to predict next-day stock-ranking scores. Historical validation results support the tested adaptive encoding configuration and its learned rate-based allocation. A system built on the proposed method achieved champion status in an external quantitative modeling competition. The organizers executed the submitted training code and evaluated the resulting model on hidden test data using the submitted inference code; test labels and scores were not used for training or model selection.

1 INTRODUCTION

1.1 FROM EQUAL-WIDTH PATCHES TO ADAPTIVE INFORMATION TIME

This study begins with a practical question: how can a stock's daily time series be represented as tokens for a Transformer or another sequence model? The competition supplies 240 minute records per stock-day. A natural starting point is equal partitioning: each five-minute window becomes one token, giving 48 tokens for convenient batching and a controlled computational budget.

Equal-width partitioning assigns a vector of the same dimension to every five-minute interval. We use *representation capacity* to mean the number and dimensionality of tokens describing an interval. Here the vector dimension is fixed; the allocation of tokens along natural time changes. The opening session incorporates overnight information, the closing session contributes to closing-price formation, and intraday activity varies with stock-specific events and trading conditions. A fixed window can compress a brief change together with quiet observations. This motivates testing sample-dependent temporal resolution.

We learn variable temporal coverage through a positive rate at each minute. Accumulating these rates defines a stretchable temporal coordinate in which 48 token centers are equally spaced. This construction preserves minute ordering while varying coverage in natural time, without directly predicting discrete boundaries that complicate gradient-based training.

The question originates in Transformer input representation, but the final model uses a causal convolutional Mixer for the short token sequence. Attention is not required to implement the temporal representation studied here.

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.

1.2 Research foundations and contributions

DeepLOB demonstrates deep learning of price dynamics from order books (Zhang et al., 2019); RevIN motivates within-sample normalization (Kim et al., 2022). PatchTST, iTransformer, and TimesFM motivate examining how tokens are defined and organized (Nie et al., 2023; Liu et al., 2024; Das et al., 2024). Section 5 distinguishes the tokenizer design from established backbone components.

Our contribution has two parts: a shared network for sample-dependent, monotone, differentiable temporal allocation under a fixed 48-token budget; and its integration with a causal backbone, constrained pooling, and a cross-sectional prediction objective. Structural comparisons assess these design choices. We do not claim that adaptive tokenization or learned temporal coordinates are unprecedented.

2 Model Construction

2.1 Input and prediction task

Each sample contains the 240 one-minute records available for a stock at the close of a trading day. The input $X \in \mathbb{R}^{240\times 38}$ includes market and order-book prices, quantities, and order counts. After the close on trading day t, the model predicts a score for stock i's relative return on the next trading day. The underlying close-to-close return is

$$r_{i,t}^{\mathrm{cc}} = \frac{C_{i,t+1}}{C_{i,t}} - 1, \tag{1}$$

where $C_{i,t}$ and $C_{i,t+1}$ are the current and next closing prices. Training labels adjust common risk exposures and numerical scale. Scores are used to rank stocks. Industry and style variables are not inputs to the prediction network.

2.2 Overall architecture

The model performs input normalization, temporal reallocation, sequence modeling, and score prediction (Figure 1). The batch size is denoted by B. The network has approximately 18.03 million parameters. Here *causal* refers to left-directed convolutional connections: later tokens can read earlier tokens. Normalization and clock construction use the complete observed day. The model is an after-close predictor, not a streaming predictor at arbitrary intraday times.

Appendix A lists symbols, dimensions, and abbreviations. Dilation is the spacing of convolution taps, RF denotes theoretical receptive field, and Concat denotes concatenation.

2.3 Core method: from natural time to information time

The trading day can be viewed as a stretchable ruler: 240 minute marks retain their order while intervals expand or contract. Forty-eight equally spaced centers on the transformed ruler map back to unequal natural-time spacing (Figure 2). Appendix B details the operators.

For example, a 20-minute interval receives about four tokens under uniform allocation. If its rate is three times that of other minutes, its information-time share is approximately $60/(220 + 60)$, corresponding to about ten centers. The total remains 48; the additional allocation comes from other intervals.

2.3.1 Learning a positive rate at every minute

After normalization, mask concatenation, and projection, three stride-one left-causal convolutions produce a 128-dimensional context h_t at every minute. Each connects the current and preceding 14 normalized minute vectors. A linear head and Softplus give

$$q_t = \mathrm{softplus}\left(w_q^{\mathsf{T}}\,\mathrm{LN}(h_t) + b_q\right) + 10^{-4}. \tag{2}$$

The rate controls the minute's relative width in information time. A higher relative rate usually allocates denser token coverage to an interval. It is learned through the prediction objective; it is

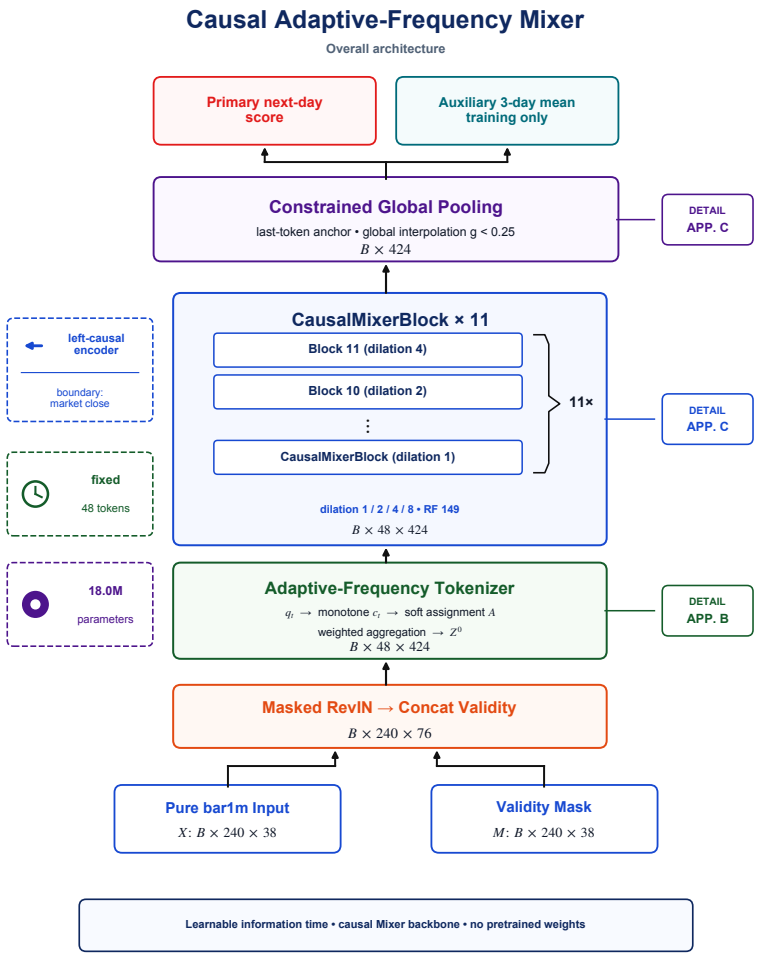


Figure 1: Overall architecture, read bottom to top. Adaptive tokenization maps 240 minutes to 48×424 representations; eleven Mixer blocks preserve sequence length. The primary head predicts the next-day score. Detailed operators appear in Appendices B and C.

not manually defined as volume or volatility, does not denote return direction, and is not a direct measurement of economic information. Layer normalization is denoted by LN; w_q, b_q are learned. Softplus smoothly maps a signed score to a positive value. The small floor protects against rates approaching zero; it is not a return or an importance threshold.

2.3.2 Positive rates define a monotone coordinate

Let $Q = \sum_{j=1}^{240} q_j$. The midpoint coordinate is

$$c_t = \frac{\sum_{j \le t} q_j - 0.5 q_t}{Q}, \qquad c_{t+1} - c_t = \frac{q_t + q_{t+1}}{2Q} > 0. \tag{3}$$

Thus the order cannot reverse. Larger rates increase local distances. Dividing by the full-day total places every stock-day in $(0, 1)$, allowing the same number of centers.

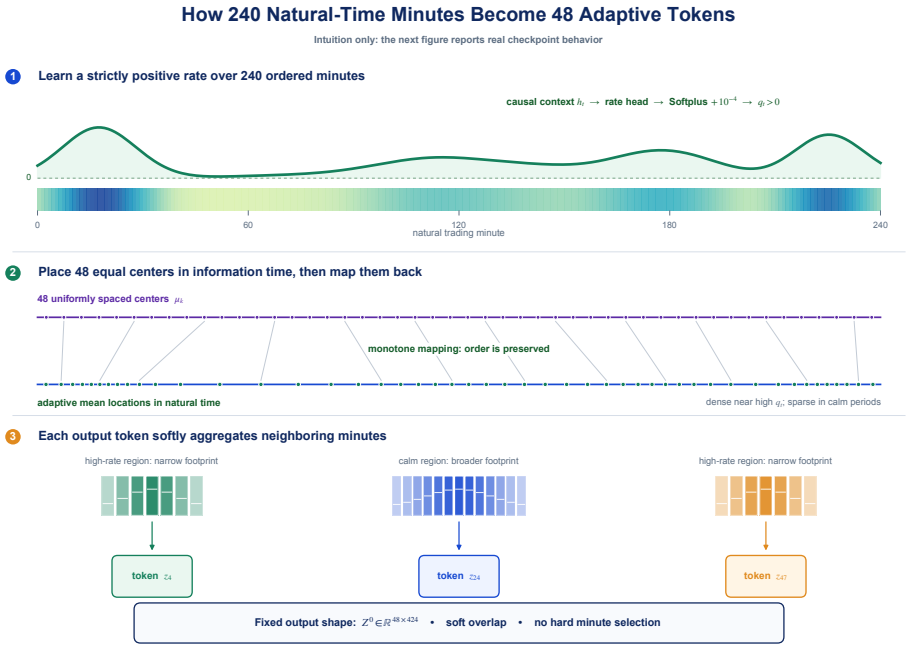


Figure 2: Temporal-budget reallocation, shown schematically. High-rate intervals occupy more information time, giving denser token centers when mapped back to natural time. Tokens are overlapping soft aggregates, not disjoint hard segments. The curves illustrate the mechanism, not measured model outputs.

2.3.3 Soft assignment produces a fixed number of tokens

Set $K = 48$ centers $\mu_k = (k - 0.5)/K$. Each minute distributes its weight according to distance:

$$A_{t,k} = \frac{\exp[-\tau(c_t - \mu_k)^2]}{\sum_{\ell=1}^{K} \exp[-\tau(c_t - \mu_\ell)^2]}, \qquad \bar{h}_k = \frac{\sum_t A_{t,k} h_t}{\sum_t A_{t,k}}. \tag{4}$$

The learned parameter τ controls concentration. Assignment sums to one across centers for each minute; pooling is a weighted average across minutes for each token. No discrete cut points or hard deletion are required, although compression can lose detail.

Each token also carries its weighted natural-time center and relative coverage mass before projection to 424 dimensions. This distinguishes early from late content and narrow from broad coverage. A learned fixed patch adapts its representation inside a fixed window; the information clock reallocates coverage for each stock-day. Appendix B gives the forward equations, numerical safeguards, and training interpretation.

2.4 Supporting backbone: causal Mixer

Tokenization sets the temporal resolution; the backbone combines 48 information fragments. Eleven blocks use the same structure with different dilations. Each applies temporal mixing followed by channel mixing (Figure 9).

Temporal mixing uses a width-five left-causal depthwise convolution, followed by a linear projection to exchange channel information. Channel mixing uses a $424 \to 1696 \to 424$ MLP at each position. Both branches use normalization, residual connections, and small-scale initialization for stable training.

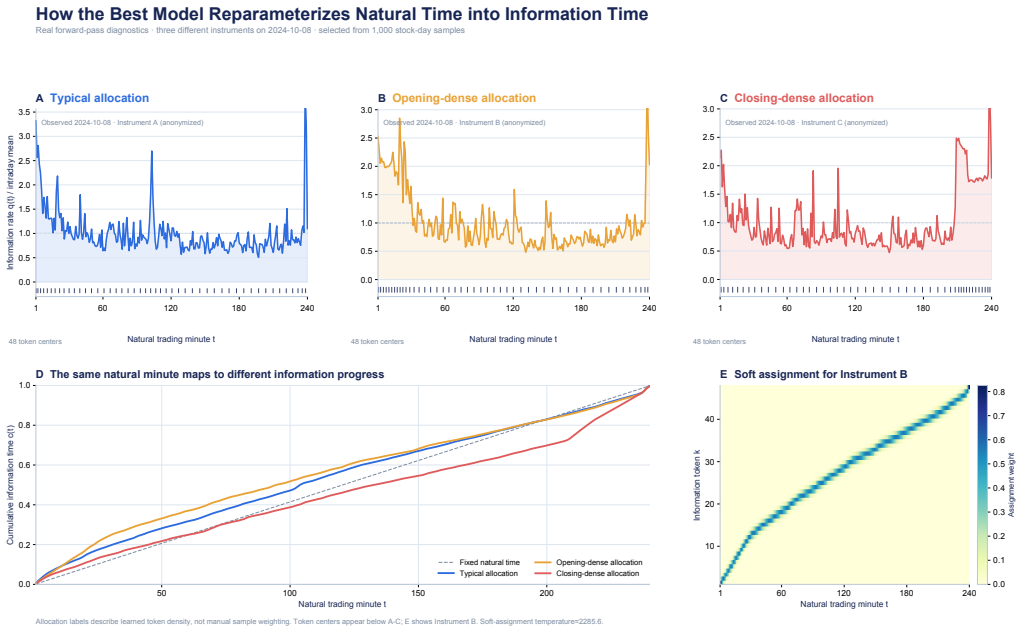


Figure 3: Real forward-pass diagnostics from the reference weights. A–C are three different anonymous securities on the same trading date: typical, opening-dense, and closing-dense allocations. D compares cumulative information time; E shows security B's assignment matrix. Dense refers to nearby token centers in natural time, not manual sample weighting.

Dilations are $(1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4)$. By block four, the last token can theoretically connect to all 48 input positions. Later blocks add nonlinear transformations and feature combinations. Section 4.1 examines depth; Appendix C gives operators and receptive fields.

2.5 Constrained Pooling and Prediction Heads

Let Z^{11} denote the sequence after eleven blocks, z_{last} its last token, and z_{global} a learned softmax-weighted average over all tokens. The readout is

$$z^* = (1 - g)z_{\text{last}} + g z_{\text{global}}, \tag{5}$$

$$g = 0.25\,\sigma\left(\text{Linear}([z_{\text{last}}; z_{\text{global}}])\right). \tag{6}$$

Hence $0 < g < 0.25$: the last representation receives more than 75% of the interpolation weight. It already integrates history, so this is not the independent contribution of raw closing minutes. Both prediction heads are LayerNorm plus a scalar linear layer. The primary head ranks next-day returns. The auxiliary target averages three consecutive one-period labels and is used only in training.

3 Training and Model Selection

3.1 Label Processing and Training Configurations

Labels start from next-day close-to-close returns. Industry and style exposures are removed cross-sectionally each day, followed by a Gaussian-rank transform. The private-stage code restores numerical scale using the ratio of robust standard deviations of neutralized returns and Gaussian-rank labels on the training set, then clips at the 1st and 99th percentiles. Scale and clipping bounds are determined from training data. The public-stage reference instead reuses a frozen label cache before clipping. Positive rescaling preserves ranks but changes error magnitude, so the two label paths are not numerically interchangeable.

The public-stage reference and private-stage code share the model structure but differ in training and selection (Table 10). SmoothL1 limits extreme-error influence; its threshold is measured in

each path's label units. The auxiliary mean of three one-period labels is standard-deviation matched and clipped, then weighted by 0.1. Samples with incomplete future labels are excluded from that term. Different labels, schedules, and seeds can produce different weights; public-stage performance cannot be assigned to the private-stage retraining result.

The optimizer is AdamW (Loshchilov & Hutter, 2019); the loss follows the SmoothL1 definition (PyTorch Contributors).

3.2 Selection of the public-stage model

Per-epoch weights were retained and compared using IC, ICIR, Sharpe proxy, and stress IR proxy on the public validation period. IC rose from 0.0341 at epoch 1 to 0.06838 at epoch 22, then fell to 0.04297 at epoch 40 while training loss continued to decrease. Epoch 22 was selected and obtained an approximately 0.892 public composite score. The report records a parameter-by-parameter reproduction of these public weights using a standalone training entry point.

Because checkpoint selection uses the public validation period, these results provide historical validation evidence, not independent-test estimates.

3.3 Automatic generation of private-stage weights

The organizer runs the frozen training and prediction code. Training follows the private configuration in Table 10. Each epoch, primary-head SmoothL1 is computed against raw close-to-close returns on the predefined 2024 validation set; the lowest-loss weights are saved automatically. Hidden-period inputs are used only for forward prediction with the resulting weights. Hidden-period labels and scores do not enter training or selection. This workflow is distinct from the public-stage multi-metric comparison.

4 Experimental Results and Core Ablations

The historical reference predictions cover 240,564 stock-date samples, 241 effective trading days, and 1,206 stocks in the 2024 public validation period. Two weight sets must be distinguished: structural comparisons use a reference with IC 0.067630; quarterly, regime, and cross-sectional diagnostics use the public-selected weights with IC 0.068378. Comparisons are made within their respective groups. Results involving 2024 model selection are not independent out-of-sample evidence.

4.1 Temporal grouping and backbone choices

Fixed windows versus sample-dependent coverage. Both schemes compress 240 minutes into 48 tokens. The fixed five-minute patch encoder concatenates normalized features and masks within non-overlapping windows (minutes 1–5, 6–10, and so on) and directly learns a token representation. Boundaries are fixed; it does not use the adaptive model's minute-context convolutions. Adaptive information time first encodes minute contexts, then softly pools them according to learned sample-specific rates; coverage can vary and overlap.

The downstream backbone, heads, and training configuration are shared. Each configuration trains for 40 epochs, selects its highest-IC epoch on 2024 under the same metric implementation, and reports all metrics at that epoch. Since local encoding also differs, this is a comparison of complete encoding schemes, not an isolated test of variable coverage.

Table 1: Complete temporal-encoding schemes; best-IC epoch per configuration.

Encoding	IC	ICIR	Sharpe proxy	Stress IR
Adaptive information time	.067630	1.153820	12.9676	.7421
Fixed five-minute patch	.054892	1.047335	10.8164	.6294

Matched-context clock controls across three seeds. We retain the minute-context encoder, output projection, Mixer, heads, and training recipe, and vary the allocation mechanism: sample-

dependent rates, uniform rates with soft pooling, learned position-only rates shared across samples, or means of five consecutive contexts. The last control differs from the direct patch encoder in Table 1. All four configurations train for 40 epochs on 2019–2023 without reading 2024 during training, using three paired seeds. All 40 checkpoints are subsequently evaluated on 2024; each run selects its own maximum-IC epoch, with other metrics taken from the same checkpoint.

Table 2: Matched-context controls: three-seed means at each run's historical IC-selected epoch. IC uncertainty is sample standard deviation, not a confidence interval.

Allocation	IC (mean $\pm$ SD)	ICIR	Sharpe proxy	Stress IR
Sample-dependent	$.065425 \pm .002248$	1.1884	13.0487	.6968
Uniform soft	$.057065 \pm .000425$	1.0094	11.8077	.6128
Shared position	$.058492 \pm .000528$	1.0360	11.7494	.6372
Fixed five-context mean	$.056313 \pm .000735$	1.0379	11.8836	.6151

Across all three seeds, sample-dependent allocation achieves higher IC than every paired control and the highest seed-mean values on all four metrics. The original seed reproduces IC .067631 at epoch 22; the other adaptive runs select epochs 18 and 15. At the fixed final epoch, mean ICs are .038369, .030127, .031215, and .027204 in table order: the ordering persists, but absolute performance is substantially lower. The paired comparisons support sample-dependent allocation within this family. Three seeds and retrospective checkpoint selection do not establish independent generalization or statistical significance.

Depth and later temporal connections. Six configurations vary depth or the last seven temporal branches under the same training and selection rule (Appendix C.3.1). Four to eight blocks attain ICs of 0.067228–0.067679. Setting the last seven dilations to one yields IC 0.068320 but lower ICIR and both proxies; no variant improves all four reference metrics.

Backbone configuration and parameter scale. With the adaptive encoder retained, the Mixer is compared with approximately 18M- and 100M-parameter Transformer configurations. Data, training recipe, and evaluation are aligned. Each model is evaluated over 40 epochs and selects its own highest-IC epoch; other metrics come from that same epoch.

Table 3: Backbone configurations and scale; all metrics are from each model's IC-selected epoch.

Metric	Causal Mixer	Transformer 18M	Transformer 100M
Parameters	$\sim$18.03M	$\sim$18.03M	$\sim$100.00M
IC	**.067630**	.056915	.057719
ICIR	**1.153820**	1.071515	1.037178
Sharpe proxy	**12.9676**	11.5187	10.9037
Stress IR proxy	**.7421**	.5802	.5254

The reference Mixer achieves the highest values on all four metrics. Replacing it with a Transformer also changes normalization, residual placement, and FFN settings; the shared training recipe may not be optimal for each architecture. These results favor this Mixer configuration for the evaluated task and recipe, without establishing a general advantage of convolutions over attention.

Finding. Adaptive encoding exceeds fixed-window direct encoding, and the matched-context controls support sample-dependent allocation across three seeds. The separate, single-seed backbone comparison favors the 18M Mixer over both tested Transformers on all four metrics. Depth results motivate further study of six- and eight-block lightweight candidates.

4.2 Quarterly and market-state consistency

Figure 12 in Appendix D provides the detailed plots. To assess concentration in particular periods, predictions are grouped by quarter and realized market state. Nonnegative mean stock return defines

an up market; a negative mean defines a down market. Cross-sectional return standard deviation at or above the year's median of 2.23% defines high dispersion; values below it define low dispersion. Dispersion concerns differences among stocks on the same day, not the time-series volatility of an index.

All four quarterly mean ICs are positive: Q3 is lowest (0.05094) and Q4 highest (0.08569). Down-market, high-dispersion days have the lowest state mean (0.05352); up-market, low-dispersion days have the highest (0.08506). Signal strength varies, without a reversal of group-mean direction.

Intervals use 5,000 independent bootstrap resamples of daily IC and the 2.5th/97.5th percentiles of the mean. The full-year interval is [0.061108, 0.075975]. These descriptive intervals do not correct for serial dependence, model selection, or multiple comparisons. Positive group means do not imply a positive signal every day.

4.3 Dependence on stock groups

Stocks are divided daily into three size groups. Industry IC is computed for 25 industries with at least 200 effective trading days (Figure 13). Large-, medium-, and small-cap group ICs are 0.07197, 0.06194, and 0.07352, respectively; all three intervals exclude zero. Mean IC is positive in all 25 industries, and 23 industry intervals have a positive lower bound. The intervals use the preceding subsection's procedure.

These diagnostics show no visible concentration of the signal in one size group or a few industries. They describe breadth within the same public validation year and do not constitute independent replication.

5 Contributions and Limitations

5.1 Contributions and their evidence boundaries

The method lets different stock-days use different natural-time coverage while retaining the same token count. At the method level, positive rates, monotone coordinates, soft assignment, and temporal metadata form a sample-adaptive $240 \rightarrow 48$ tokenizer. Ordering follows mathematically from the construction. The complete encoding scheme exceeds fixed-window direct encoding in historical validation; matched-context controls additionally favor sample-dependent allocation over uniform soft, shared-position, and fixed-mean alternatives across three seeds (Section 4.1). At the system level, the representation is jointly trained with the causal Mixer, constrained pooling, and cross-sectional targets, yielding positive historical ranking signals.

Dilated causal convolutions have established precedents in WaveNet and TCN (van den Oord et al., 2016; Bai et al., 2018). Token/channel mixing, depthwise convolution, expanded MLPs, and small residual scales are found in MetaFormer, ConvMixer, ConvNeXt, ConvFormer, and ModernTCN (Yu et al., 2022; Trockman & Kolter, 2023; Liu et al., 2022; Yu et al., 2024; Luo & Wang, 2024). These backbone operators are not claimed as original contributions.

Under the evaluated task and training recipe, the compact Mixer exceeds the tested 18M and 100M Transformers on all four historical metrics. Establishing runtime efficiency requires matched timing measurements. Comparison with prior adaptive allocation methods also remains necessary; we make no first-ever claim for learned rates or adaptive tokens.

5.2 Current limitations

Validation selection bias. Public weights and clock-control checkpoints were selected using per-epoch 2024 results. Clock controls now cover three seeds, whereas the earlier depth and backbone comparisons remain single-seed. Repeated use of one historical year does not establish independent generalization; seed variability does not quantify temporal uncertainty or correct selection bias.

Prediction-time boundary. Complete-day normalization and time coordinates restrict the model to after-close prediction. Intraday use requires redesigning information availability rather than reusing full-day calculations.

Compression and interpretation. Forty-eight tokens may omit detail. High rates describe relative representation allocation, not measured economic information or local causal contributions.

6 Conclusion and Future Directions

Learned positive rates, cumulative coordinates, and soft assignment produce ordered tokens with fixed count and sample-dependent temporal coverage. Historical comparisons favor the tested adaptive encoder over direct fixed-window encoding, while depth and backbone comparisons reveal configuration-specific trade-offs. With the minute-context encoder and downstream network retained, sample-dependent allocation exceeds all three paired clock controls in IC for each of three seeds. Its historical selected-epoch IC is 0.065425 ± 0.002248 (mean $\pm$ sample standard deviation), while the original seed reproduces 0.067631. The same ordering holds at epoch 40, although all configurations deteriorate relative to their selected peaks. Together, these comparisons support content-dependent allocation in the evaluated setting. Retrospective selection bias remains, and out-of-period performance is not established by these comparisons.

6.1 Future research

More robust model selection. Forward validation within the training period, frozen selection rules, plateau-based selection, weight averaging, and additional seeds across periods could reduce dependence on one validation peak.

Cross-day state and minute-token fusion. Earlier long-history, multiscale, and frequency-related explorations did not yield stable multi-metric gains. Future work could fuse the current intraday branch with a separately normalized slow state from completed prior days. A bounded gate that permits zero contribution would test whether history helps; history length and fusion should be varied separately.

Targets and trading constraints. Label alignment, turnover, slippage, capacity, and predictive uncertainty require controlled evaluation before claims about trading performance or transfer across markets.

AI USE STATEMENT

An AI coding agent assisted with literature discovery, implementation, debugging, experiment organization, analysis, and manuscript preparation, including translation and typesetting. The human authors are responsible for the research decisions, the accuracy of numerical results, citations, and all submitted materials. This version extends the research report with a completed three-seed matched-context clock-control study.

ETHICS STATEMENT

The authors confirm permission to use the competition data for research publication. Permission to publish research does not imply permission to redistribute raw data. The reported financial metrics are historical diagnostics, not investment recommendations or claims of realized returns.

REPRODUCIBILITY STATEMENT

Sections 2–4 specify the architecture, training paths, model selection, and comparison boundaries. Appendices A–D provide notation, forward computations, complete depth results, training configurations, and additional diagnostics. Figures are included as vector PDFs with their editable SVG sources. A public training reproduction also requires the permitted data, label pipeline, and evaluation dependencies.

REFERENCES

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. URL https://arxiv.org/abs/1803.01271.

Abhimanyu Das et al. A decoder-only foundation model for time-series forecasting. In *ICML*, 2024. URL https://arxiv.org/abs/2310.10688.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Yong Liu et al. itransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2024. URL https://arxiv.org/abs/2310.06625.

Zhuang Liu et al. A convnet for the 2020s. In *CVPR*, 2022. URL https://arxiv.org/abs/2201.03545.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *ICLR*, 2024. URL https://openreview.net/forum?id=vpJMJerXHU.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.14730.

PyTorch Contributors. Smoothl1loss. PyTorch documentation. URL https://docs.pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html.

Asher Trockman and J. Zico Kolter. Patches are all you need? *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2201.09792.

Aaron van den Oord et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Weihao Yu et al. Metaformer is actually what you need for vision. In *CVPR*, 2022. URL https://arxiv.org/abs/2111.11418.

Weihao Yu et al. Metaformer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. URL https://arxiv.org/abs/2210.13452.

Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 2019. URL https://arxiv.org/abs/1808.03668.

A NOTATION AND ABBREVIATIONS

The batch dimension is omitted in single-stock-day equations. Minute indices and trading-day indices are distinguished by context. Superscripts on token sequences indicate network depth, not powers.

A.1 SHAPES AND INDICES

Symbol	Meaning
B, T, F, K, C	Batch size, 240 minutes, 38 input fields, 48 tokens, and 424 backbone channels.
X, M	Input and elementwise validity mask, both $B \times 240 \times 38$.
H	Minute contexts, $B \times 240 \times 128$.
Z^0, Z^{11}	Token sequences before and after eleven blocks, $B \times 48 \times 424$.
z^*	Pooled representation, $B \times 424$.
i, t, k, ℓ	Stock, minute (or trading day in return equations), token, and zero-based block index.

A.2 TARGETS

Symbol	Meaning
$C_{i,t}, r^{cc}_{i,t}$	Closing price and next-day close-to-close return, $C_{i,t+1}/C_{i,t} - 1$.
$\alpha_t, \boldsymbol{\beta}_t, \boldsymbol{\gamma}_t$	Intercept, style coefficients and industry coefficients in daily cross-sectional neutralization.
$\boldsymbol{z}_{i,t}, \boldsymbol{d}_{i,t}$	Style exposures and industry indicators; label-construction inputs, not network inputs.
$\epsilon_{i,t}$	Return residual after neutralization.
$u_{i,t}, \Phi^{-1}, g_{i,t}$	Clipped rank percentile, inverse standard-normal CDF and Gaussian rank target.
$s, \widetilde{y}, y$	Scale adjustment, pre-clipping target and final training target.
P_1, P_{99}, W	Training-set percentile cutoffs and clipping operator.
$\mathcal{D}_{\text{train}}$	Training set used to estimate label scale and cutoffs.

A.3 INFORMATION TIME AND TOKENIZATION

Symbol	Meaning
v_t, h_t	Normalized feature–mask concatenation (76 dimensions) and local context (128 dimensions).
w_q, b_q, r_t	Shared linear rate-head parameters and its unconstrained scalar output.
q_t, Q	Positive rate and its sum over all minutes.
b_t, c_t	Normalized cumulative boundary and midpoint coordinate.
p_t, μ_k	Fixed natural-minute position and uniformly spaced information-time center.
ρ, τ	Learned log-temperature and bounded distance-kernel precision.
$A_{t,k}$	Fraction of minute t assigned to token k; sums to one across tokens.
$m_k, \bar{h}_k$	Column mass and mass-normalized context average.
$\bar{p}_k, d_k$	Weighted natural position and relative mass $m_k K/T$.
e_k, z^0_k	Learned token-position embedding and initial token representation.

A.4 BACKBONE AND READOUT

Symbol	Meaning
Z^ℓ, U^ℓ	Block input and intermediate state after temporal mixing.
d_ℓ	Dilation of block ℓ, cycling through $1, 2, 4, 8$.
LN, DWConv	Layer normalization and depthwise convolution.
α_ℓ, β_ℓ	Per-channel LayerScale parameters, initialized to 10^{-3}.
R_L	Receptive-field span after L temporal blocks.
$z_{\text{last}}, z_{\text{global}}$	Last-token state and learned weighted global summary.
g, z^*	Bounded gate, $0 < g < 0.25$, and combined representation.
$\widehat{y}, \widehat{a}$	Primary next-day and auxiliary prediction.

A.5 Losses and diagnostics

SmoothL1 uses prediction error e, threshold $\beta = 0.01$, and an auxiliary weight of 0.1. The auxiliary target averages three consecutive one-period targets; it is not a compounded three-day return. IC measures cross-sectional association; ICIR summarizes its daily mean relative to variability. Sharpe and stress IR are the report's historical proxies, not realized portfolio results. PIT denotes point-in-time data discipline; WSD denotes warmup–stable–decay; EMA and SWA denote possible future weight averaging methods.

Overloaded symbols. r_t in the rate head is not a return; $r^{cc}_{i,t}$ is a return. The Gaussian rank $g_{i,t}$ is not the pooling gate g. The token mass m_k is not the validity mask M. LayerScale β_ℓ, regression coefficients $\boldsymbol{\beta}_t$, and the SmoothL1 threshold β are distinct.

B From 240 Minutes to 48 Adaptive-Coverage Tokens

This appendix follows the forward computation: inputs, contexts, rates, coordinates, assignments, and tokens. Adaptive coverage changes the span summarized by each token; the output count and dimension remain fixed.

B.1 FORWARD OVERVIEW

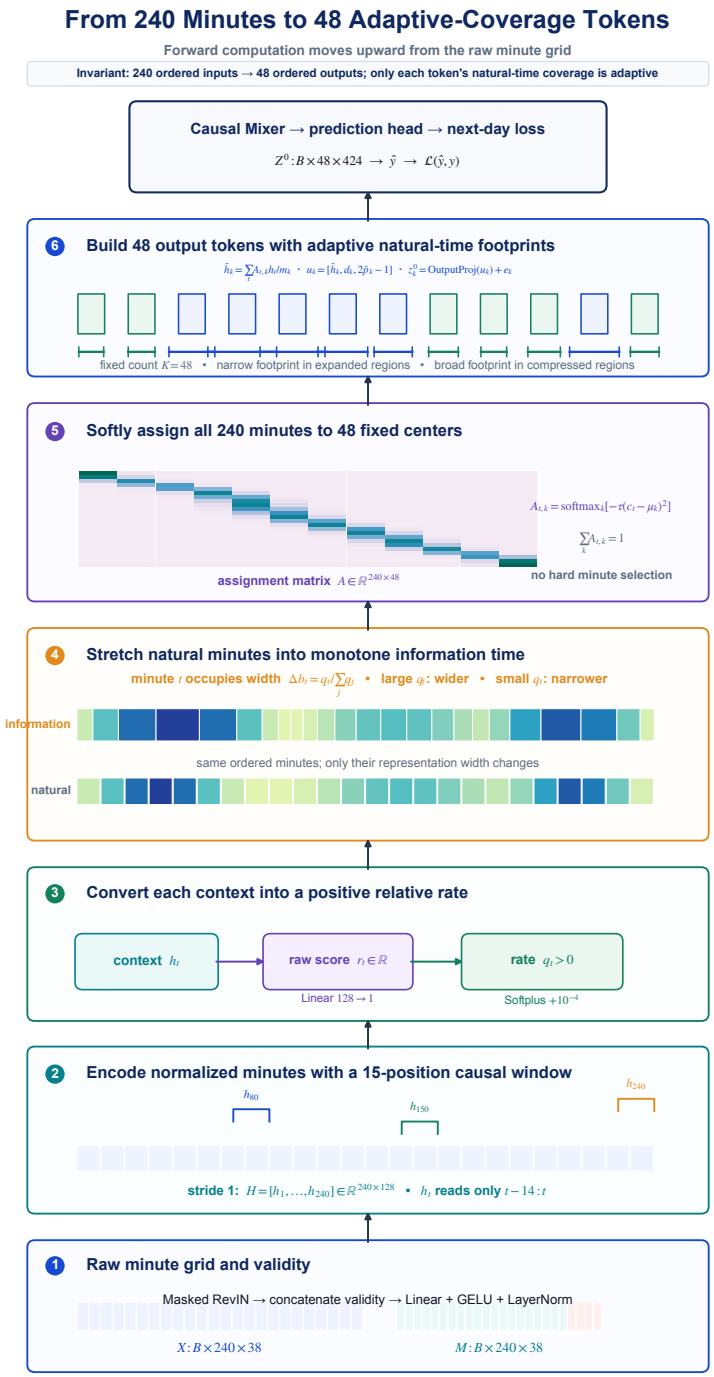


Figure 4: Forward path, read bottom to top. Minute cells and the heatmap illustrate the mechanism rather than measured values.

For a single stock-day, $H = [h_1, \ldots, h_{240}] \in \mathbb{R}^{240\times 128}$. Every minute has a context; the encoder stride is one. The six central quantities are

$$x_t \longrightarrow h_t \longrightarrow q_t \longrightarrow c_t \longrightarrow A_{t,k} \longrightarrow z_k^0.$$

They respectively denote observations, contextual features, relative representation rate, information-time location, soft allocation and output representation.

B.2 A NUMERICAL EXAMPLE

With uniform $q_t = 1$, each minute occupies $1/240$ of information time. Forty-eight centers correspond to approximately five-minute spacing, although overlapping soft pooling is still different from disjoint averaging. If a 20-minute interval has rate three and the remaining 220 minutes rate one, total rate is $Q = 280$. That interval covers $60/280 = 21.4\%$ of information time despite occupying only 8.3% of natural time. Approximately ten centers cover it, compared with four under uniform rates. The overall budget remains 48.

B.3 MINUTE-LEVEL CONTEXTS

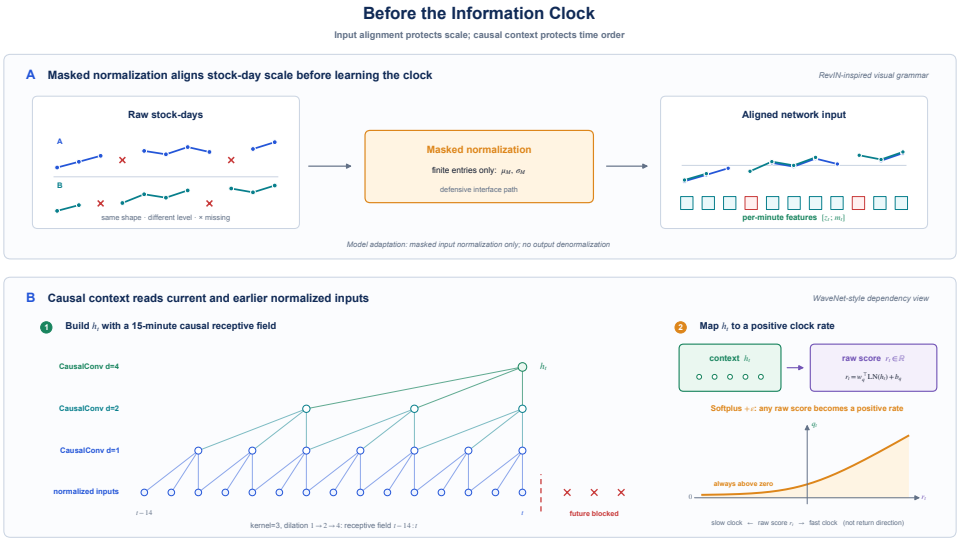


Figure 5: Normalization and local context formation. Positive rates encode relative clock speed, not the direction of returns.

B.3.1 NORMALIZATION AND PROJECTION

The standard input path clips fields to nonnegative values, applies `log1p`, and replaces nonfinite values with zero. The model constructs its mask from input finiteness, so upstream replacement affects which missingness remains visible. Masked normalization estimates each stock-day's feature means and variances from valid entries. Concatenating normalized values with the same-shaped mask gives 76 channels, projected to 128:

$$v_t = [\mathrm{MaskedRevIN}(X, M)_t, m_t], \qquad h_t = f_\theta(v_{t-14:t}).$$

The normalization motivation follows RevIN (Kim et al., 2022), without output denormalization for a ranking score. Full-day statistics and the clock denominator require the completed input day. Local causal convolutions do not make the entire system an intraday streaming predictor.

B.3.2 WHY FIFTEEN POSITIONS?

Three kernel-three convolutions use dilations $1, 2, 4$. A layer with dilation d directly reads $t, t - d, t - 2d$ from the preceding layer, adding $2d$ to its receptive-field span:

$$R = 1 + 2 \cdot 1 + 2 \cdot 2 + 2 \cdot 4 = 15.$$

The successive spans are $t - 2 : t$, $t - 6 : t$, and $t - 14 : t$. Intermediate positions are already represented by lower-layer nodes, so dilated connections do not omit those positions from the combined dependency. Thus h_{120} depends on normalized positions 106–120, and h_{240} on 226–240. Left padding handles the beginning of the sequence.

B.4 FROM CONTEXTS TO A MONOTONE CLOCK

B.4.1 SHARED SCALAR HEAD

$$r_t = w_q^\top \, \mathrm{LN}(h_t) + b_q.$$

The nonlinear encoder has already extracted context patterns; this shared low-parameter head maps them to the scalar degree of freedom needed for clock control. It learns a context-dependent budget score rather than a fixed importance for each minute index.

B.4.2 POSITIVE RATE

$$q_t = \log(1 + e^{r_t}) + 10^{-4} > 0.$$

Softplus is smooth, with derivative $\sigma(r_t)$, and approximately linear for large positive scores. Negative scores produce slower clocks, not predicted losses. Zero-initialized head weights and bias approximately 0.5413 start the rate near one, allowing the predictive objective to learn departures from uniform time.

B.4.3 BOUNDARIES AND MIDPOINTS

$$Q = \sum_t q_t, \quad b_0 = 0, \quad b_t = \frac{\sum_{j \le t} q_j}{Q}, \quad c_t = \frac{b_{t-1} + b_t}{2}.$$

Minute t occupies width q_t/Q. Multiplying all rates by the same positive constant leaves the coordinates unchanged. Moreover,

$$c_{t+1} - c_t = \frac{q_t + q_{t+1}}{2Q} > 0,$$

so chronological order is strictly preserved in exact arithmetic.

B.5 SOFT ALLOCATION AND COVERAGE METADATA

B.5.1 ASSIGNMENT AND AGGREGATION

$$\mu_k = \frac{k - 0.5}{K}, \quad \tau = \mathrm{clip}(e^\rho, 100, 20000), \quad A_{t,k} = \frac{e^{-\tau(c_t - \mu_k)^2}}{\sum_{j=1}^{K} e^{-\tau(c_t - \mu_j)^2}}.$$

Each row sums to one. Every minute distributes its context among centers; no hard boundary selection is required. High-rate intervals span more information-time centers and receive finer representation.

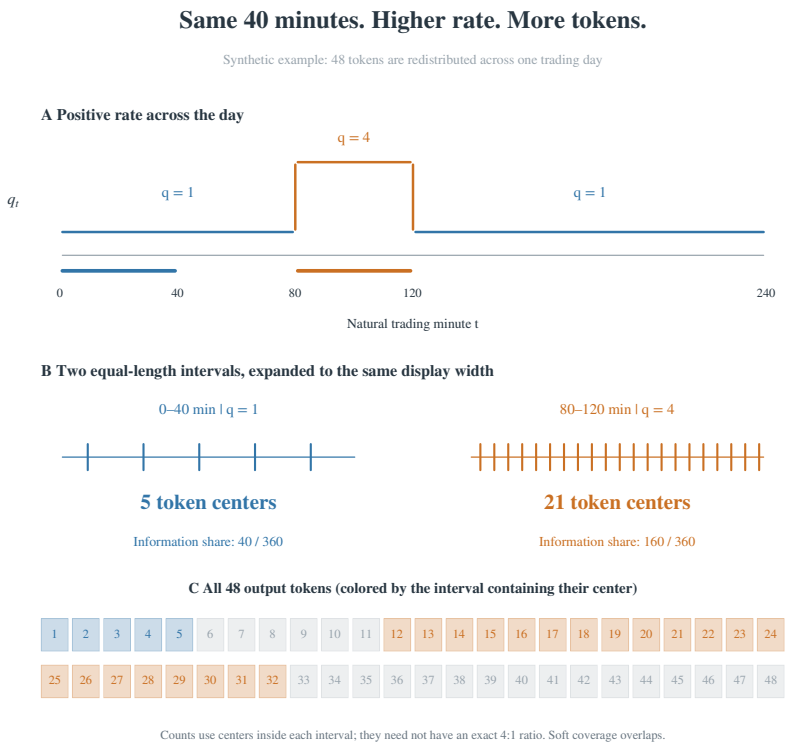


Figure 6: Illustrative equal-duration intervals with different rates. The two forty-minute regions receive five and twenty-one centers under the displayed cumulative coordinate.

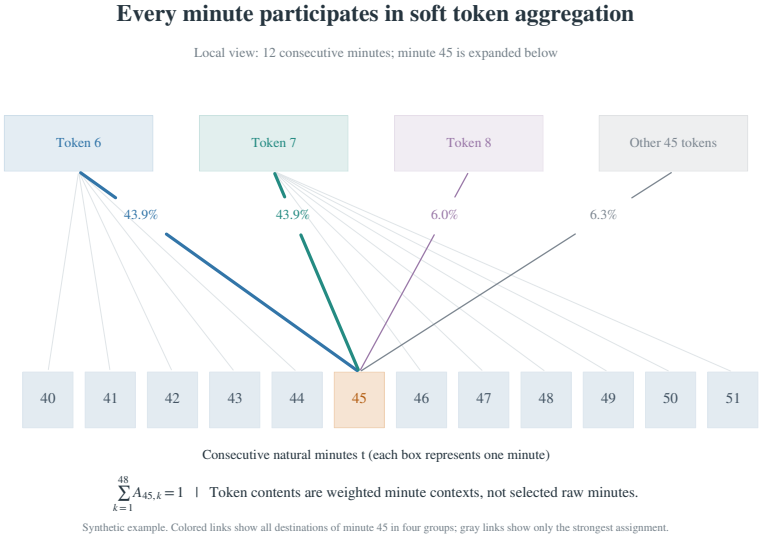


Figure 7: A minute contributes to several tokens. Displayed percentages are row assignment weights, not its final proportion of a token, which also divides by column mass. Gray lines display only the largest connection for clarity.

$$m_k = \sum_t A_{t,k}, \qquad \bar{h}_k = \frac{\sum_t A_{t,k} h_t}{m_k}.$$

Compression is not lossless. Distant assignments can also underflow numerically, despite positive mathematical kernel weights.

B.5.2 Mass, position, and relative duration

Row normalization does not imply unit column mass. Since $\sum_k m_k = T$, mean mass is $T/K = 5$. For example, weights $0.2, 0.7, 0.8, 0.3$ yield mass two, equivalent to two units of minute weight, not a hard two-minute window.

$$p_t = \frac{t - 0.5}{T}, \qquad \bar{p}_k = \frac{\sum_t A_{t,k} p_t}{m_k}, \qquad d_k = \frac{m_k K}{T}.$$

Here $\bar{p}_k$ locates the summary within the day, and d_k expresses its mass relative to five minutes. Masses 2.5, 5 and 10 correspond to relative durations 0.5, 1 and 2. Projection of $[\bar{h}_k, d_k, 2\bar{p}_k - 1]$, followed by the learned position embedding, produces the 424-dimensional token.

B.6 Interpretation and limits

The rate is learned from the downstream prediction loss, not supervised by a ground-truth information measure. Positive rates provide order preservation; soft allocation supplies a continuous training path. Content-dependent coverage reallocates a fixed budget across each stock-day, unlike fixed-window representation learning. This mechanism does not establish that a high-rate minute contains causal price information, nor does it guarantee improved prediction without empirical comparisons. The full-day coordinate is available after the input session closes.

B.7 DETAILED TOKENIZER OPERATORS

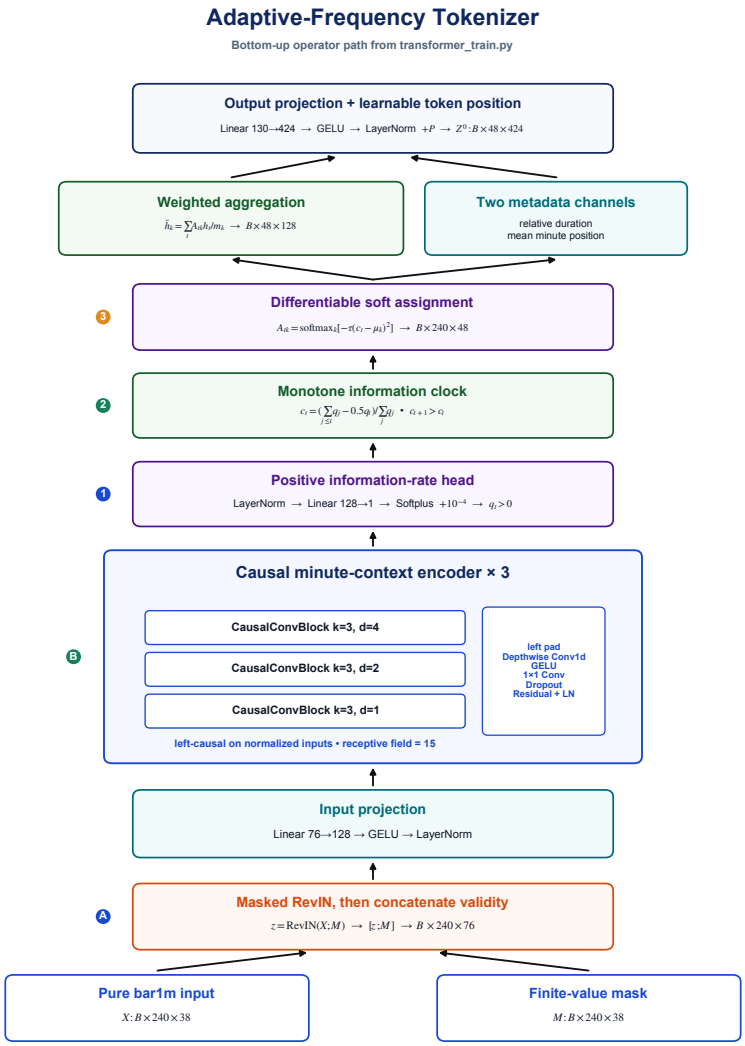


Figure 8: Bottom-up tokenizer computation. Minute contexts produce positive rates, rates define monotone coordinates, and continuous assignment pools 240 contexts into 48 tokens. Figure 2 illustrates the three numbered stages.

C CAUSAL MIXER STRUCTURE AND DEPTH COMPARISONS

C.1 ARCHITECTURE CHOICE

The tokenizer outputs 48×424 representations. Eleven blocks preserve this shape. The report's configuration also reflects the competition's serialized model-size constraint; eleven is an engineering choice, not a theoretically optimal depth. Causal dilated convolutions and channel mixing have established precedents (van den Oord et al., 2016; Bai et al., 2018; Yu et al., 2022; Trockman & Kolter, 2023).

C.2 BLOCK EQUATIONS

For $\ell = 0, \ldots, 10$, the temporal and channel branches are

$$U^{\ell} = Z^{\ell} + \alpha_{\ell} \odot \mathrm{Linear}\Big[\mathrm{GELU}\Big(\mathrm{DWConv}^{\mathrm{causal}}_{5,d_{\ell}}(\mathrm{LN}(Z^{\ell}))\Big)\Big], \tag{7}$$

$$Z^{\ell+1} = U^{\ell} + \beta_{\ell} \odot \mathrm{MLP}(\mathrm{LN}(U^{\ell})). \tag{8}$$

The MLP expands 424 channels to 1,696, applies GELU and dropout 0.1, projects back to 424, and applies another dropout. The temporal branch has no dropout. Per-channel LayerScale vectors start at 10^{-3}. Residual connections preserve the identity path; normalization is applied before each branch.

C.3 RECEPTIVE FIELD AND DEPTH

With kernel width five and stride one,

$$R_L = 1 + 4 \sum_{\ell=0}^{L-1} d_{\ell}.$$

The dilation sequence is $1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4$. Receptive-field spans at depths 1, 2, 3, 4, 8 and 11 are 5, 13, 29, 61, 121 and 149. The last token can access all 48 token positions by block four; earlier tokens still cannot read future positions. Later blocks add nonlinear transformations and repeated contextual mixing, rather than new available input positions for the last token. Receptive-field coverage alone cannot determine useful depth.

C.3.1 COMPLETE DEPTH COMPARISON

All rows use the same evaluation rule: choose the highest-IC epoch among forty epochs and take the remaining metrics from that epoch. These are single-seed historical comparisons.

Table 8: Depth and tail-connection ablations.

Configuration	Epoch	IC	ICIR	Sharpe	Stress IR
11 blocks, reference	22	.067630	1.153820	12.9676	.742106
4 blocks	20	.067228	1.238900	12.8302	.704804
6 blocks	19	.067679	1.168317	12.4698	.633283
8 blocks	23	.067318	1.177618	13.2234	.745880
11 blocks, dilation-one tail	22	.068320	1.133311	12.4789	.702931
11 blocks, channel-only tail	20	.065764	1.147961	12.4209	.624226

The shortened models retain the corresponding prefix of the dilation cycle. The dilation-one tail keeps eleven blocks but sets the last seven dilations to one; the channel-only tail removes their temporal branches. No tested configuration improves all four reference metrics. Six and eight blocks are plausible efficiency candidates, while the highest-IC dilation-one tail trades away stability and the two proxies. These results support treating depth and late dilation as tunable choices; repeated-seed comparisons and independent evaluation are needed before a general recommendation.

C.4 BLOCK AND READOUT DIAGRAMS

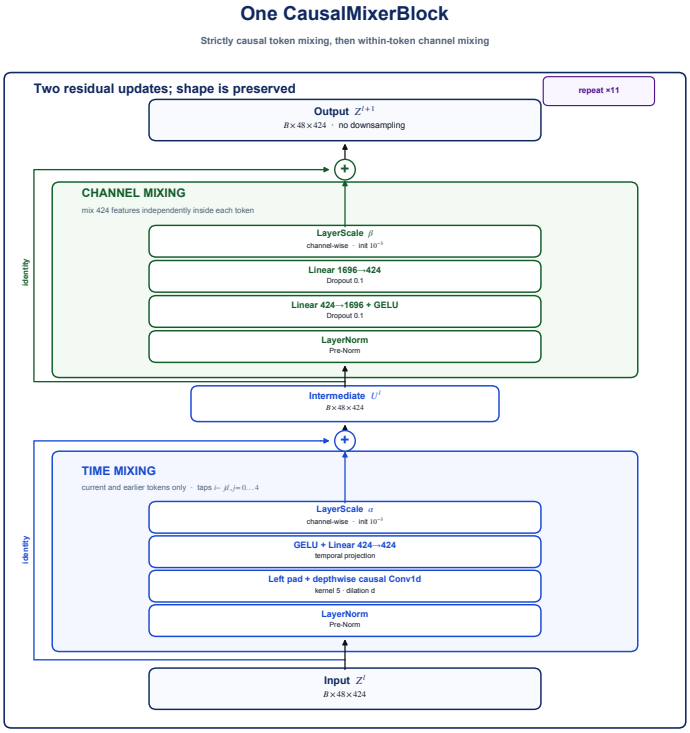


Figure 9: One Mixer block, executed bottom to top. Temporal mixing combines the current and earlier tokens; channel mixing combines features within a token. Two residual paths preserve the incoming representations.

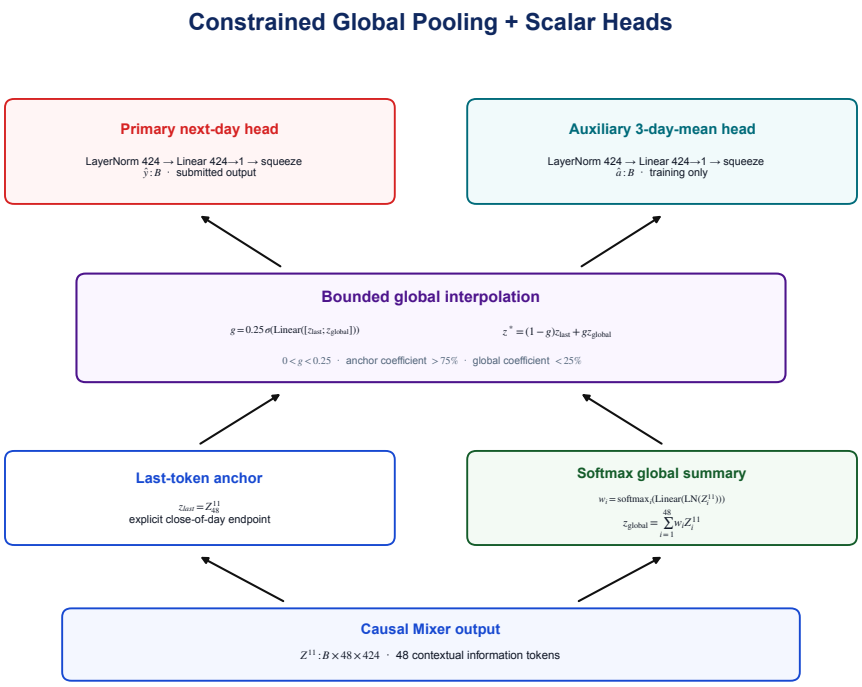


Figure 10: Constrained global pooling and two prediction heads. The last token anchors the readout; the global branch has a bounded mixing coefficient. Only the next-day head is retained for inference.

D Training Details and Additional Diagnostics

D.1 Component Roles

Table 9: Roles of the architectural components.

Component	Function	Output
Masked RevIN / mask	Normalize valid observations; concatenate validity indicators	Minute features and mask
Adaptive tokenizer	Adjust the temporal coverage of feature vectors	48 tokens, width 424
Causal Mixer block	Combine current and earlier token information	48 updated tokens
Constrained pooling	Fuse the last token with a bounded global contribution	One 424D vector
Primary / auxiliary heads	Main score and auxiliary training constraint	Main score at inference

D.2 Training Configurations

The two training paths differ in labels, schedules, random seed and checkpoint selection.

Table 10: Public-stage reference versus private-stage training code.

Item	Public reference	Private training code
Training period	2019-01-01 to 2023-12-31	Same
Fixed validation	2024-01-01 to 2024-12-31	Same
Input / structure	240×38; 48 tokens; 11 blocks	Same
Label path	Frozen neutralized Gaussian ranks; clipping	Rebuild; restore robust scale; clipping
AdamW	LR $8e{-5}$; decay $1e{-4}$	Same
SmoothL1	$\beta = 0.01$; auxiliary weight 0.1	Same; different label units
LR schedule	5% warmup, 75% stable, 20% linear decay	40-epoch cosine; no warmup
Epochs / batch	40 / 1024	Same
Dropout / channel mask	0.1 / 1.5%	0.1 / 0
Seed	20260828	20260820
Weight selection	Per-epoch validation metrics	Minimum validation loss

D.3 Public-stage selection trajectory

All epochs shown use the same public validation year used for selection.

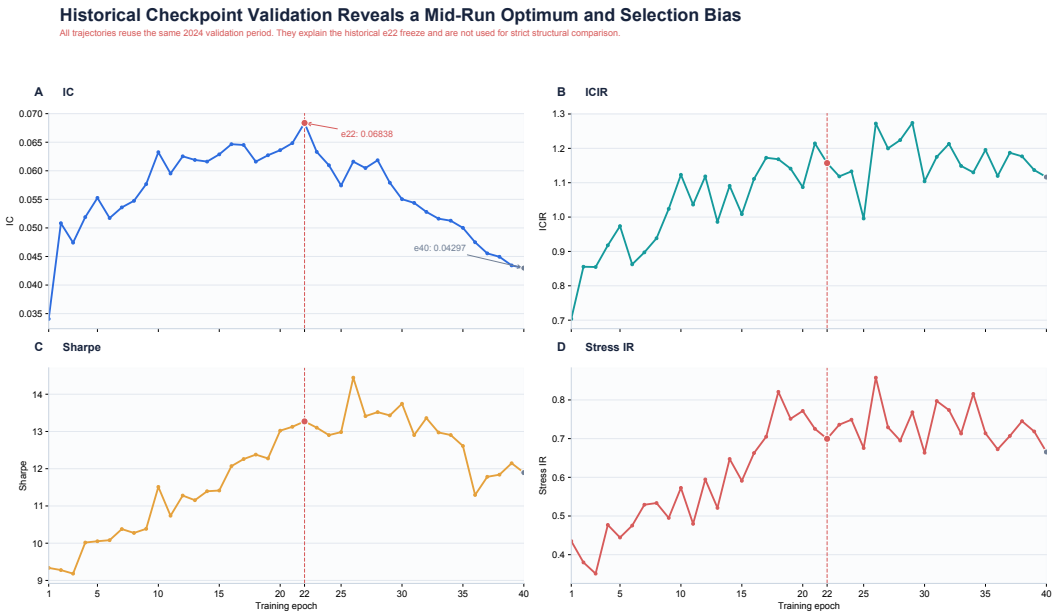


Figure 11: Forty-epoch validation trajectories of the public-stage reference. IC peaks at epoch 22; other metrics peak at different epochs. The final epoch is not validation-optimal. This period was used for selection.

D.4 Depth trade-offs

The following summary complements the full selected-epoch values in Appendix C.3.1.

Table 11: Depth and dilation choices relative to the eleven-block reference.

Configuration	IC	Other metric trade-offs
11 blocks, cyclic dilation	.067630	Reference
4 blocks	.067228	Higher ICIR; lower two proxies
6 blocks	.067679	Higher ICIR; lower two proxies
8 blocks	.067318	Higher ICIR and both proxies
11 blocks, dilation-one tail	.068320	Lower ICIR and both proxies
11 blocks, channel-only tail	.065764	Lower ICIR and both proxies

D.5 Market-state and cross-sectional diagnostics

These plots describe the public-selected weights, not the structural-comparison reference. Group definitions and interval limitations are given in Section 4.

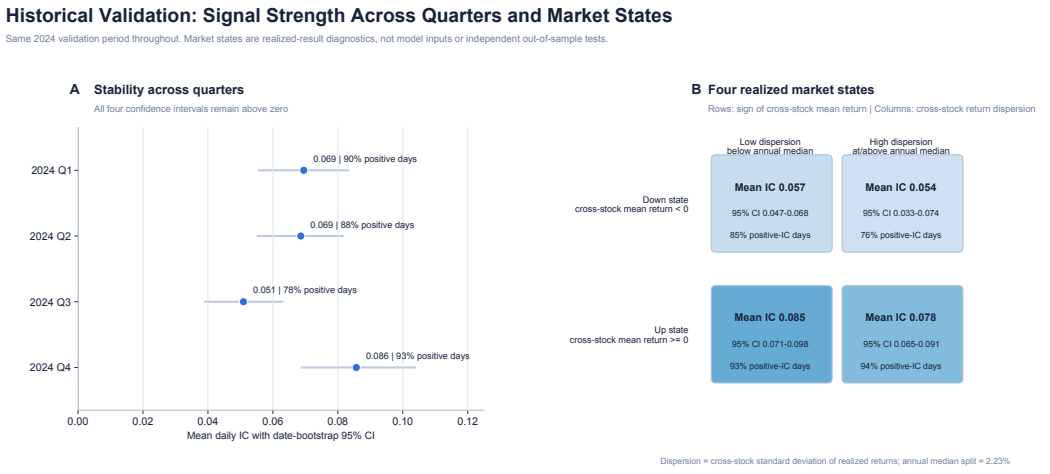


Figure 12: Quarterly and market-state diagnostics. Panel A gives mean daily IC and 95% intervals by quarter; panel B groups by the sign of market mean return and cross-sectional return dispersion. These are retrospective groups, not model inputs.

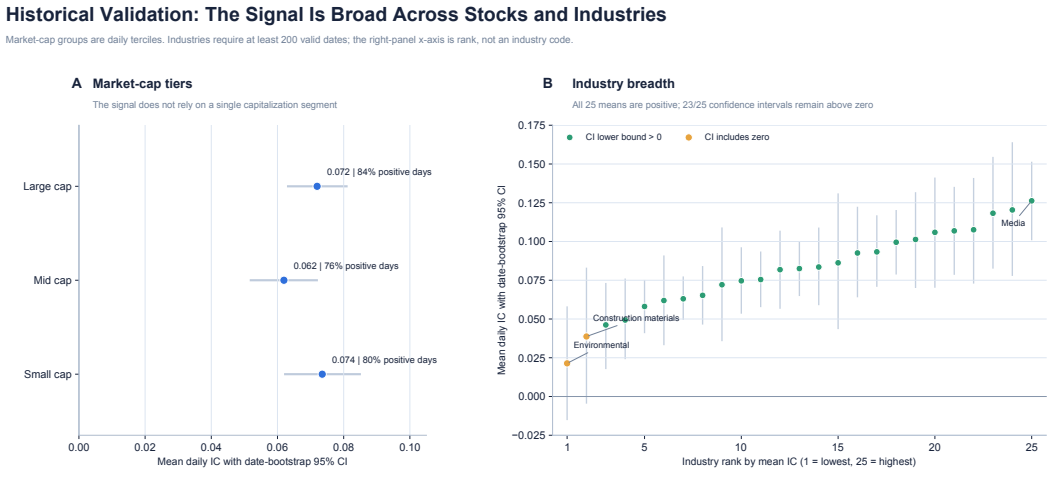


Figure 13: Size and industry diagnostics. Left: three size groups. Right: 25 industries ordered by mean IC; positions 1–25 denote ranks. Green indicates a positive interval lower bound, and orange an interval containing zero.