Temporal Validation and Incremental News-Text Analysis of a Lightweight Cross-Sectional Equity Ranking Model

Model Construction, Matched Comparisons, and Cost Sensitivity for M8 N0

Jiexiao Zhang

discoveryopen.org

Corresponding author: Jiexiao Zhang | Email: jiexiaozhang1@outlook.com

Abstract

Cross-sectional equity prediction requires explicit information-timing rules, consistent treatment of differences in samples across trading dates, and careful separation of model selection from final evaluation. This study examines whether structured market features and news-event metadata support same-day open-to-close return ranking when inputs are frozen before the market opens. It also assesses the incremental contribution of news text in a matched comparison. M8 N0 combines 23 numerical features, their missingness indicators, and two log-transformed event counts into a 48-dimensional input. A multilayer perceptron with 5,345 trainable parameters produces cross-sectional ranking scores. Training uses within-date rank-normalized targets, a joint SmoothL1 and pairwise ranking objective, and optimization batches organized around complete trading dates. Three expanding-window folds use preprocessing statistics fitted only on their respective training samples, temporal gaps, and controls for duplicate events across partitions.

Across three development-validation folds with seed 42, the model covers 298,769 symbol-day observations and 1,350 trading dates. Mean daily Spearman IC is 0.02088, with positive IC on 55.48% of dates. Fold-level mean IC values are 0.01633, 0.02948, and 0.01668. In the matched comparison on fold 0, the numerical baseline, additive text-residual model, and text branch with relation-weighted pooling achieve IC values of 0.01633, 0.01587, and 0.00396, respectively. The zero-text control matches the numerical baseline. These findings document an observable historical ranking relationship in the structured inputs within the completed configurations and sample scope, while the tested text addition does not improve mean IC on this fold. Post-hoc moving-block bootstrap analyses characterize uncertainty in the saved daily sequences under temporal dependence, and yearly summaries describe temporal variation. An exploratory portfolio analysis demonstrates that ranking metrics and returns after costs require separate interpretation: an equal-weight long-short portfolio selecting 20% of each tail has approximately -3.11% arithmetic annualized return under a simplified cost of 2 basis points per one-way execution. Reload and release-consistency checks support reproducible inference. Because validation data were used for checkpoint selection, the results are developmental rather than an untouched out-of-sample test. The study provides an auditable research workflow connecting information timing, a numerical baseline, incremental-text controls, and released artifacts, establishing an explicit starting point for independent temporal evaluation and multi-seed research.

Keywords: cross-sectional equity ranking; event metadata; lightweight neural network; temporal validation; pairwise ranking; reproducibility

1 Introduction

1.1 Research Problem and Application Context

Financial prediction is often framed as forecasting future prices, but many investment studies first need to address relative ordering. Within a trading date, researchers seek to identify securities whose subsequent returns are more likely to rank near the top or bottom of that date's sample. A model may provide useful ranking information even when it cannot accurately predict the numerical magnitude of returns. Conversely, a lower average prediction error does not necessarily imply an ability to distinguish return rankings. The task objective, training loss, evaluation metrics, and final portfolio construction therefore need to address the same underlying question.

The prediction horizon in this study runs from the market open to the close on the same day. The decision time is fixed at 09:15 in New York local time. Inputs consist of statistics from previously completed market history and news-event metadata admitted under the information-timing rules. The model assigns a continuous score to each eligible vendor symbol on a trading date, and daily cross-sectional ranking measures evaluate its relationship with subsequent returns. The study population is subject to conditions on price, liquidity, history length, event coverage, and data completeness. Consequently, the findings apply to the filtered, event-covered sample rather than to the population of arbitrary US equities.

News text can describe market events, but the existence of textual information and a text branch's ability to deliver a stable incremental contribution are distinct propositions. A text model may identify sentiment yet lose its advantage when the task objective, temporal resolution, or training budget changes. Market history, event counts, and publisher structure may already contain some information related to the text. Accordingly, this study treats the numerical and metadata model as an independent object of investigation and examines the contribution of an additional text branch through within-fold controls, rather than assuming that more complex inputs necessarily outperform a smaller model.

1.2 Research Questions and Study Design

The empirical investigation addresses three questions. First, does a lightweight numerical model exhibit a positive historical daily ranking relationship when training-only preprocessing, forward temporal splits, and an explicit information-timing contract are applied? Second, in the available matched setup for fold 0, does adding a text residual change mean daily IC, and can a zero-text control verify consistency of the numerical pathway? Third, how should ranking results be interpreted across years, statistical treatments, and simplified transaction-cost assumptions? These questions concern predictive relationships, incremental information, and the boundaries of practical use, respectively.

The study uses two layers of evidence. The first comprises saved training and validation artifacts, including best checkpoints, daily metrics, training histories, fold-level sample statistics, and release-consistency checks. The second comprises post-hoc supplementary descriptive analyses of those saved daily sequences, including annual summaries and block bootstrapping. These supplementary analyses do not add model training or create independent test samples. Their role is to characterize existing results more fully, not to change the status of the original validation data.

1.3 Contributions and Organization

The first contribution is a complete description of an equity-ranking implementation with an explicit model size and a fixed input specification. Its 48-dimensional input, 5,345 parameters, training-period scaling, missing-value treatment, and within-date target construction can be checked individually, making the origin of prediction scores transparent. The second contribution is the joint presentation of numerical results across three folds and a matched text comparison on fold 0, exposing the relationships among the numerical pathway, text pathway, and zero-text control. The third contribution is the separate reporting of statistical ranking, exploratory portfolios, and reproducibility of the released model, so that executable files do not substitute for predictive evidence and gross returns do not substitute for economic significance after costs.

These contributions primarily concern empirical research and research engineering. Multilayer perceptrons, robust losses, and pairwise ranking have established foundations in the literature and are not presented here as new general-purpose algorithms. The paper proceeds through related work, the sample and timing protocol, the model and training methods, experimental results, cost analysis, and reproducibility and research limitations. Detailed fields and verification procedures are provided in the appendices to preserve a coherent main argument while supporting subsequent replication.

2 Related Work

Financial prediction research must address information representation, learning objectives, and experimental validation together: how market information is converted into inputs, how models learn relationships relevant to investment decisions, and what conclusions historical results can support. This section organizes related work around numerical modeling, text representations, learning to rank, and validation design to establish the research context of M8 N0.

2.1 Numerical Features and Equity Return Prediction

Gu, Kelly, and Xiu systematically compare multiple machine-learning methods for asset return prediction, examine nonlinear feature interactions, and evaluate predictions using chronologically ordered samples [2]. Their study provides context for combining price, liquidity, and volatility information and demonstrates that model value must be assessed within a specific prediction task. Its asset coverage, prediction frequency, and evaluation protocol differ from those used here. The present study therefore draws on its modeling and comparison approach rather than treating its return estimates as direct performance benchmarks for M8 N0. A smaller parameter count makes model scale explicit and facilitates reproduction of the computation; whether this implementation property translates into better predictions requires comparison under matched conditions.

2.2 Tabular Learning and Model Complexity

The 48-dimensional numerical input is structured tabular data, placing the neural network within the wider context of tabular learning methods. LightGBM introduces computational-efficiency techniques for gradient-boosted decision trees and is a candidate baseline for this type of task [10]. Gorishniy et al. emphasize consistent training and tuning protocols and compare residual networks, Transformers, and tree models [11]. Grinsztajn et al. examine the competitiveness of tree models across tabular benchmarks from multiple domains and the effects of factors such as uninformative features on neural networks [12]. These studies support a common experimental principle: architectural complexity cannot substitute for performance comparison. The lightweight network examined here has an explicit parameter count and an inspectable inference procedure; any relative advantage must be established using the same samples, the same available information, and comparable tuning budgets.

2.3 Financial Text Representations and Incremental Information

BERT learns text representations from bidirectional context, providing a foundation for downstream fine-tuning [13]. In finance, Araci's FinBERT investigates pretrained language models for financial sentiment classification [3], whereas the identically named work by Yang et al. performs domain pretraining on financial corpora and evaluates financial sentiment tasks [4]. The authors, model sources, and training configurations differ between these studies, so citations must correspond to the encoder actually used. FNSPID organizes news and stock prices temporally, providing a data resource for studying their joint use [1]. These studies show that financial text can be encoded computationally, but a task mismatch remains between textual sentiment and future equity ranking: positive news may already be reflected in prices, while publication timing, coverage, and market conditions can influence its incremental value. The present study therefore asks whether the text branch adds a reproducible contribution beyond the specified numerical information.

2.4 Ranking Objectives and Robust Optimization

Cross-sectional equity selection requires comparisons among securities on the same trading date. RankNet formulates pairwise ordering as a probabilistic learning problem, providing methodological context for pairwise ranking losses [5]. When both prediction values and relative order matter, regression and ranking terms can be combined, but their weights, pair-construction rules, and temporal scope affect the resulting learning behavior. Huber's work on robust estimation provides a classical foundation for reducing the influence of large residuals [6]; the distinction between Huber and SmoothL1 must follow the piecewise definition and scaling used in the implementation. AdamW decouples weight decay from adaptive gradient updates [7]. These references explain the origins of the training components. The contribution here is to describe how those components are integrated with symbol-day observations, feature processing, and validation, and to document their observed performance.

2.5 Temporal Validation, Selection Bias, and Research Scope

Financial experiments generally observe a finite period of realized market history, within which researchers may compare many features and models. White examines data snooping arising from repeated data use for selection and proposes a framework for testing the best model after a search [14]. Bailey et al. further study backtest overfitting and its probability assessment [8], while Harvey et al. discuss statistical thresholds in the presence of many factor tests [9]. These studies indicate that a sample occurring after the training period is not necessarily uninvolved in research selection. When validation results inform early stopping, checkpoint selection, or portfolio-rule selection, their developmental role must be explicit. Subsequent independent evaluation requires frozen rules and samples that did not participate in selection. Accordingly, this study distinguishes model-ranking results, incremental-text comparisons, and portfolio sensitivity analysis, stating the scope supported by each type of evidence.

In summary, the empirical study centers on a reproducible lightweight numerical model and treats the text branch as a source of incremental information requiring separate verification. Its focus is a complete record from information-time alignment and feature processing through the joint loss and period-specific evaluation, allowing parameter scale, performance metrics, and the scope of conclusions to be interpreted together. Under this framing, a result observed in one configuration motivates further research; repetition across periods and random seeds, standardized baselines, and a final independent test provide a path toward stronger conclusions.

3 Data Organization and Information-Timing Protocol

3.1 Data Sources and Unit of Observation

The market-data pipeline records Zihan1004/FNSPID as its source, while the event pipeline uses records associated with ashraq/financial-news and locally processed artifacts. The research background of FNSPID is described in [1]. These sources are reported separately: the original dataset's overall size, news sources, and organization cannot be directly equated with this project's cleaned training sample. Sample statistics in this paper are calculated from the actual run manifests and the effective members remaining after filtering.

Each observation consists of a vendor security symbol and a decision date. Its inputs include historical open-to-close returns, volatility, volume and liquidity statistics, market-condition statistics, and event-structure information available by the cutoff. During training or evaluation, the observation is also joined to that date's opening and closing prices to construct the target. Symbol identifiers support data joins and output alignment but are not included in the 48-dimensional network input.

A vendor symbol is a data identifier; linking it to a stable security identity across years still requires security-master information. Symbol changes, symbol reuse, delistings, and corporate actions can all affect historical joins. This study retains the vendor-symbol convention used by the executed pipeline and explicitly acknowledges this data-level limitation. The training procedure constrains information timing, but timing constraints alone cannot establish that every historical security mapping and data revision has been reconstructed on a point-in-time basis.

3.2 Decision Cutoff and Date-Level News Mapping

The decision time is set to 09:15 in the America/New_York time zone. A regional time-zone identifier with daylight-saving rules is used instead of assuming a constant UTC offset throughout the year. Historical market features must originate from trading dates strictly before the decision date. Events must have an availability time no later than the decision time and must also satisfy the requirement that their source date precedes the decision date.

For news with only a calendar date and no reliable hour or minute, the pipeline assigns the record to the first XNYS trading session strictly after its source date. For example, if an event is known only to have a Friday source date, its earliest decision-date assignment is the next trading session, rather than an assumption that it was available before Friday's market open. This rule forgoes some information that might have been available earlier in exchange for a timing definition supported by the recorded fields. The example explains the rule and is not an additional observation or experiment.

The event source provides calendar-date precision only. The available_at field is a research timestamp constructed under the preceding rule, not a vendor-recorded historical delivery or first-capture time. Mapping to a later trading session constrains the information window used in the study but does not verify when the report was actually accessible historically. Thus, availability before 09:15 means compliance with the specified operational timing contract, not independent evidence of real-time delivery of the original information.

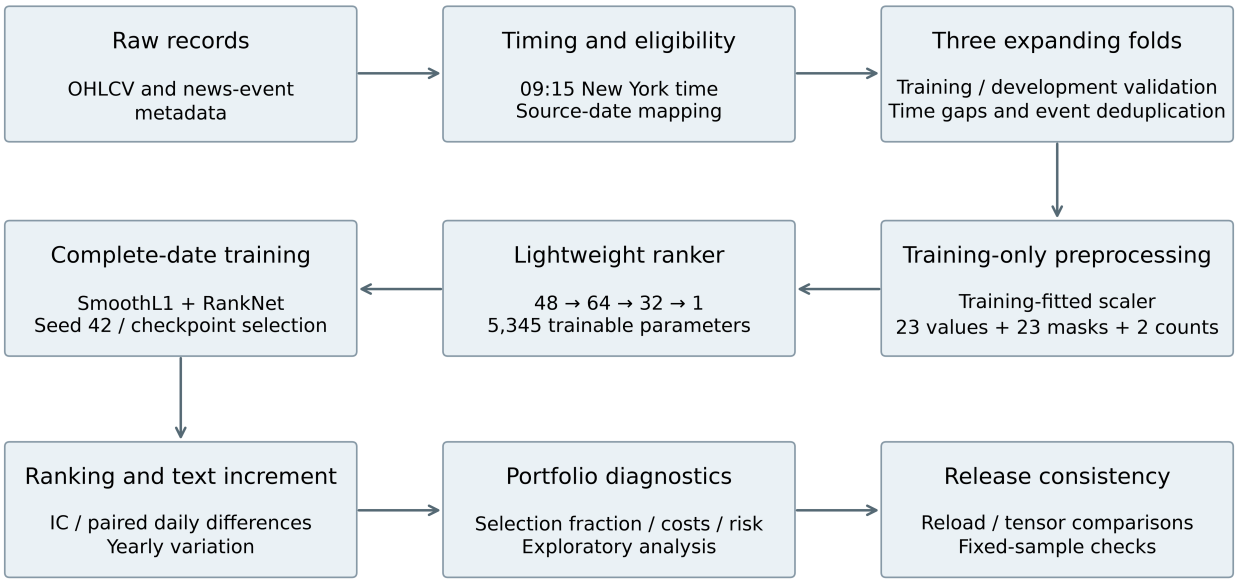


Figure 1. Research data and model-processing workflow. Solid connections show the implemented sequence of data processing, training, and evaluation. Development validation also serves checkpoint selection. Release consistency is assessed in a separate engineering verification stage.

3.3 Universe Filtering and Conditional Coverage

The main filters require valid OHLCV on the preceding trading date, a previous closing price of at least USD 5, raw median dollar volume of at least USD 1 million over the preceding 20 trading dates, the required history over the preceding 120 XNYS sessions, no suspected adjustment-factor change during the five trading dates before the decision date, and at least one eligible event. An individual event may be linked to no more than 20 vendor symbols, and event and publisher linkages must meet completeness requirements. Training accepts samples by date, with at least 30 eligible symbols required on each accepted date.

These rules produce an event-covered cross section with specified historical and liquidity support, while also defining the scope of extrapolation. Securities without news coverage, with insufficient liquidity, or without the required history have not been evaluated under the same conditions. Applying the model to the entire market would require checking the distribution of the additional samples and consistency of input construction, rather than relying only on matching field names.

Filtering also changes the relationship between nominal and effective windows. The nominal training start for all three folds is February 4, 2010, whereas the first date actually admitted to training and scaler estimation is March 1, 2010. Sample counts are reported for effective members. An intermediate selection table before training can differ from the observations ultimately used in optimization. For example, fold 0 contains 19,847 rows across 318 dates before the minimum-30-symbols-per-date rule; the final accepted sample contains 19,512 rows across 306 dates.

3.4 Expanding Windows and Duplicate-Event Controls

The experiment uses three expanding training windows ordered forward in time. Later folds can use a longer training history, while each validation window remains after its corresponding training window. Earlier dates can be reused in different roles across adjacent folds: a validation date in an earlier fold may become a training date in a later fold. This is consistent with expanding-history training and does not place the same date in both training and validation within an individual fold.

Table 1. Effective training and development-validation samples across three temporal folds

Fold	Effective training period	Training rows / dates	Validation period	Validation rows / dates
0	2010-03-01 to 2011-12-09	19,512 / 306	2011-12-27 to 2013-10-17	65,036 / 448
1	2010-03-01 to 2013-10-17	89,297 / 775	2013-11-01 to 2015-08-24	107,457 / 455
2	2010-03-01 to 2015-08-24	201,797 / 1,240	2015-09-09 to 2017-07-10	126,276 / 447

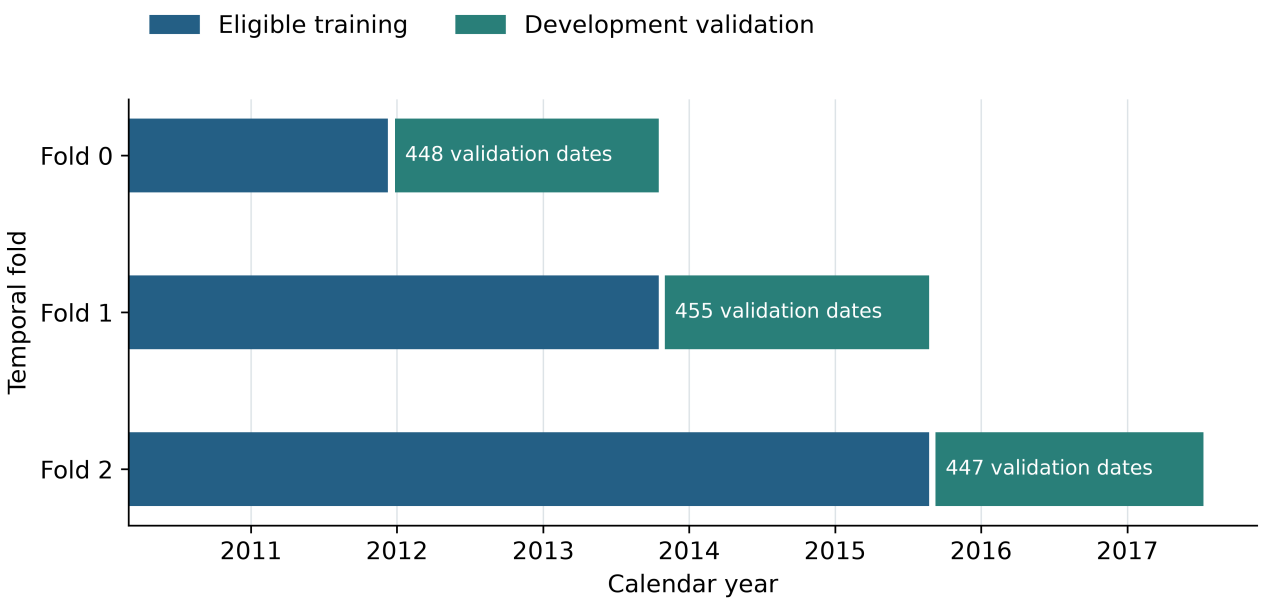


Figure 2. Temporal structure of the expanding windows. Dark segments indicate effective training periods and light segments indicate development-validation periods. Date intervals are drawn on the actual time scale. No additional evaluation of the sealed interval was conducted for this paper.

Each fold is configured with five purge trading dates and five embargo trading dates between training and validation. If an event or headline cluster spans that fold's temporal partitions, its associated members are removed from the fold to reduce the propagation of near-duplicate information across partitions. These measures address temporal proximity and content duplication, respectively. They do not establish that every possible source of data leakage has been eliminated; historical revisions and security identities, for example, still require independent data audits.

The existing M8 release records contain zero inference passes on development_holdout. The later sealed window from June 7, 2019 to June 5, 2020 was also not used for additional experiments in this paper. However, a period inspected during an earlier development stage does not become an unseen test merely because it is renamed. This paper therefore consistently uses development validation to describe the windows involved in best-checkpoint selection and does not report independent-test conclusions.

4 Feature Representation and Target Construction

4.1 Composition of Structured Information

The 23 base numerical features comprise 10 describing asset-level market and liquidity conditions, six describing historical market and QQQ conditions, and seven describing news-event associations and publisher structure. Adding a missingness indicator for each base field and log1p transforms of the raw and deduplicated event counts gives a final input dimension of 23 + 23 + 2 = 48. The input file supplies 25 fields; the inference program generates missingness indicators automatically from the base fields.

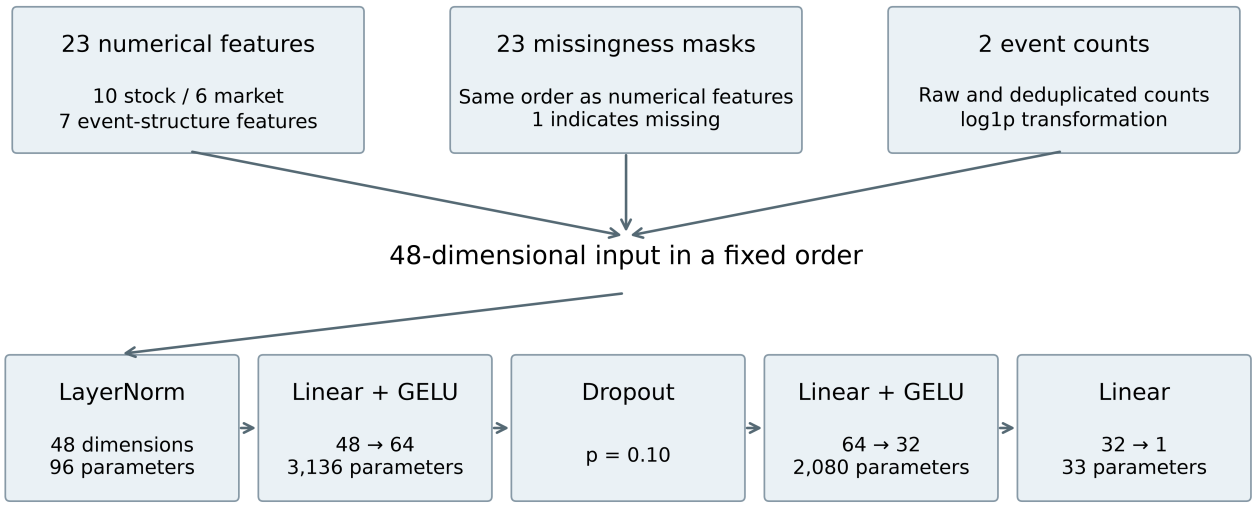


Figure 3. The 48-dimensional input and N0 network architecture. Base numerical features, missingness indicators, and event counts are concatenated in a fixed order. The network produces a single score for within-date ranking.

Event-structure fields include the maximum and mean numbers of symbols associated with an event, mean relation weight, the share of multi-symbol headlines, publisher count, publisher concentration, and the share of records with missing publisher information. Publisher concentration is the sum of squared shares, or HHI; it generally increases as a larger share of events is concentrated among fewer publishers. Event counts measure the number of records linked to an observation, while deduplicated counts capture structural information distinct from the raw total. The numerical model therefore includes event metadata, and its results cannot be interpreted as results from a market-data-only model.

4.2 Training-Period Scaling and Missingness Indicators

Each fold estimates a median and an interquartile range for every field using only that fold's accepted training observations. The interquartile range is the 75th percentile minus the 25th percentile; when invalid or too small, it falls back first to the training standard deviation and then to 1. Validation data are not used to estimate these statistics. Missingness is determined before imputation: unparseable or nonfinite base values are marked as missing and replaced with the training median.

$$z_{itj} = \mathrm{clip}\left(\frac{\tilde{x}_{itj} - m_j}{a_j}, -10, 10\right), \quad \tilde{x}_{itj} = \begin{cases} x_{itj}, & x_{itj}\ \text{finite}, \\ m_j, & \text{otherwise}. \end{cases}$$

Equation (1). Training-period scaling and clipping of base numerical features

In Equation (1), m_j and a_j denote the training median and training scale, respectively. Clipping to the interval from -10 to 10 limits the magnitude of extreme inputs far from the training distribution. It does not imply that clipped observations have become ordinary, so the clipping rate should be monitored in practical use. After median imputation, a missing value has a numerical channel of 0 and a missingness channel of 1. An observed value exactly equal to the median instead has a missingness channel of 0, preserving the distinction between these cases.

The two event counts receive separate log1p transforms, require finite nonnegative inputs, and do not use the median-based scaling described above. The log1p function is defined at 0 and reduces order-of-magnitude differences between large and small counts. Each fold's weights must be paired with its own scaler. Otherwise, identical raw features are mapped into a different coordinate system, and scores lose their training-time interpretation.

4.3 From Same-Day Returns to Rank-Normalized Targets

The raw label is the same-day closing price divided by the opening price, minus 1. For each eligible trading date, average ranks of these returns are first calculated within that date's symbol set in the main filtered model_context; tied returns receive average ranks. Targets are constructed before fold-membership filtering and cross-partition cluster filtering, and within-date target ranks are not recomputed after those filters. This affects the target scale while preserving the monotonic ordering of same-day returns. Ranks and the date's sample size then define quantiles between 0 and 1, clipped to the interval from 0.0001 to 0.9999. Applying the inverse standard-normal cumulative distribution function yields rank-normalized scores.

$$r_{it} = \frac{C_{it}}{O_{it}} - 1, \quad p_{it} = \mathrm{clip}\left(\frac{\mathrm{rank}_t(r_{it}) - 0.5}{N_t}, 10^{-4}, 1 - 10^{-4} \right), \quad q_{it} = \Phi^{-1}(p_{it})$$

Equation (2). Same-day open-to-close returns and rank-normalized targets

The rank transformation emphasizes relative ordering, aligning the target more directly with cross-sectional ranking rather than requiring the network to fit the numerical magnitudes of a small number of extreme returns. It also changes the interpretation of the output: a prediction is a fitted score for a standardized ranking target, not a percentage return. Training additionally standardizes the target using the mean and standard deviation of that fold's training targets. Any inverse transformation must use the target-scaling statistics from the same fold.

For example, two trading dates can each contain a top-ranked security even when market-wide returns differ substantially between those dates. The rank target treats both as highly ranked observations within their respective dates without requiring identical realized returns. This property makes within-date comparisons consistent and also explains why the model score alone cannot determine position sizes across dates or predict absolute profits.

5 Model Architecture and Training Methods

5.1 Network Architecture and Parameter Count

N0 comprises an input LayerNorm, a linear layer from 48 to 64 dimensions, GELU, Dropout with probability 0.10, a linear layer from 64 to 32 dimensions, GELU, and an output layer from 32 to 1 dimension. LayerNorm normalizes each observation's input representation and learns dimension-specific affine parameters. The two nonlinear transformations allow interactions among historical returns, liquidity, and event structure. Dropout randomly masks some activations during training and is disabled during inference.

Table 2. Layer-by-layer calculation of N0 trainable parameters

Layer	Calculation	Parameters
LayerNorm(48)	48 + 48	96
Linear(48,64)	48 × 64 + 64	3,136
Linear(64,32)	64 × 32 + 32	2,080
Linear(32,1)	32 × 1 + 1	33
Total	Includes biases and LayerNorm affine terms	5,345

This architecture allows both the computational pathway and the parameter count to be checked directly. A small parameter count makes the model component of saving, loading, and batch scoring relatively lightweight, but end-to-end system costs also include upstream market-data access, event joins, and feature computation. Online latency, serving throughput, and order-submission latency were not measured. Parameter count is therefore reported as a measure of network size, not as an unmeasured performance guarantee.

5.2 Pointwise Fitting and Pairwise Ranking Losses

The training objective combines SmoothL1 and a within-date pairwise ranking loss. SmoothL1 uses a quadratic penalty for small residuals and a linear penalty for large residuals, with beta set to 1. It constrains the numerical relationship between scores and standardized targets, while the pairwise ranking term directly constrains the relative order of symbols on the same date. These terms provide pointwise and pairwise supervision, respectively, drawing on robust estimation and RankNet [5–6].

$$\ell_{\mathrm{SL1}}(e) = \begin{cases} \frac{1}{2}e^2, & |e| < 1, \\ |e| - \frac{1}{2}, & |e| \geq 1. \end{cases} \qquad e = \hat{q} - \tilde{q}$$

Equation (3). SmoothL1 loss with beta equal to 1

$$p_{ij} = \sigma(s_i - s_j), \quad y_{ij} = \mathbf{1}[\tilde{q}_i > \tilde{q}_j],$$
$$L_{\mathrm{rank},t} = -\frac{1}{|P_t|} \sum_{(i,j)\in P_t} [y_{ij} \log p_{ij} + (1 - y_{ij}) \log(1 - p_{ij})],$$
$$L_{\mathrm{N0},t} = 1.10\,(0.25 L_{\mathrm{SL1},t} + 0.75 L_{\mathrm{rank},t}).$$

Equation (4). Within-date pairwise ranking loss and the joint objective

In Equation (4), P_t denotes the sampled pair set for trading date t, y_ij is determined by the relative target values, sigma is the sigmoid function, and the temperature is 1. Sampled pairs exclude equal targets and contain observations from the same trading date. They are stratified by the true rank difference and drawn deterministically under fixed random settings, with at most 4,096 distinct pairs per date. Stratification exposes training to both nearby and widely separated ranks, although the sampled loss remains an approximation to the objective over all possible pairs.

The primary loss assigns weights of 0.25 to SmoothL1 and 0.75 to the ranking loss; the numerical auxiliary loss has weight 0.10. For N0, the final prediction and numerical auxiliary prediction are the same tensor, so the auxiliary loss equals the primary loss. The per-date objective is therefore equivalent to multiplying the primary loss by 1.10. This implementation supplies neither a second set of labels nor an independent source of information; it changes the overall gradient scale.

5.3 Complete-Date Optimization and Update Weights

The number of available symbols varies across dates. Averaging all rows together would naturally assign greater weight to dates with larger cross sections. Instead, the trainer first calculates losses for complete dates and then aggregates the date-level losses within the current update using a fixed denominator of 3. Complete dates are packed in a deterministic order until at least 64 rows have accumulated, at which point an update group ends. Dates are not truncated to meet the target row count, and the last group of an epoch can contain fewer than 64 rows. The denominator is the maximum number of dates in an update group across all planned epochs. It equals 3 in the completed runs rather than being a universal constant prescribed for arbitrary data.

$$L_{\text{update}} = \frac{1}{3} \sum_{t \in \mathcal{D}_{\text{update}}} L_{\text{N0},t}$$

Equation (5). Update-level aggregation of complete-date losses

Equation (5) assigns the same coefficient to each date's loss within an update group rather than weighting it by the number of observations on that date. Because the denominator is fixed at 3, the operation is not exactly the arithmetic mean within each update group: update magnitude changes when groups contain different numbers of dates. Replication must preserve this implementation detail instead of merely setting a batch size of 64 rows and randomly splitting observations.

5.4 Optimization, Checkpoint Selection, and Recovery

N0 is trained with AdamW, a peak learning rate of 0.0001, and weight decay of 0.01. Linear warmup covers the first 6% of planned steps, followed by linear decay. Training is limited to 8 epochs, and the global gradient norm is clipped to 1.0. Runs use CUDA, FP16 automatic mixed precision, and GradScaler. AdamW decouples weight decay from adaptive gradients, as described in [7], but its contribution in this project would require a separate ablation experiment.

Table 3. Training and selection settings

Item	Setting in the reported runs
Optimizer / learning rate / weight decay	AdamW / 0.0001 / 0.01
Training precision / gradient clipping	CUDA FP16 AMP + GradScaler / norm 1.0
Training plan / schedule	Up to 8 epochs / 6% warmup followed by linear decay
Date groups / ranking pairs	Complete dates packed to a target of 64 rows / at most 4,096 pairs per date
Loss	0.25 SmoothL1 + 0.75 RankNet; N0 objective multiplied by 1.10
Loss beta / ranking temperature	1 / 1
Validation frequency	Every 750 successful optimization steps and at the end of each epoch
Selection and early stopping	Mean daily IC / minimum improvement 0.0001 / patience 4
Seed	Completed runs use seed 42; three folds are not three seeds

Validation is triggered every 750 successful optimization steps and at the end of each epoch. Selection uses mean daily Spearman IC, and a new candidate must exceed the current best value by at least 0.0001. Early stopping occurs after 4 consecutive checks without sufficient improvement. Final optimization steps for folds 0, 1, and 2 are 1,715, 4,500, and 9,003, respectively; the selected best steps are 1,715, 3,000, and 7,500. Released weights come from the best step, which need not be the final step.

Every 250 optimization attempts, the trainer saves a recoverable state comprising the model, optimizer, scheduler, mixed-precision scaler, random states, and data-iteration position, binding that state to the data, configuration, code, fold, and variant. All three released-run records contain zero AMP-skipped steps and no unrecovered out-of-memory events. Recovery support improves traceability after an interruption, but it provides a different type of evidence from out-of-sample predictive ability.

6 Evaluation Design and Statistical Methods

6.1 Daily Ranking Metrics

The primary metric is the Spearman correlation between prediction scores and targets within each trading date, termed daily Rank IC. Because the target is a monotonic transformation of within-date return ranks, this metric measures relative ordering. Aggregation weights evaluable dates equally and reports the median, share of positive-IC dates, and conventional t statistic for the mean. Observation counts describe coverage, while trading dates form the units of the statistical time series. The saved checkpoint-selection validation uses a minimum of 30 symbols per date. The three folds contain 15, 1, and 25 dates with fewer than 100 symbols, respectively. All metrics in this paper follow the stated minimum-30-symbols-per-date evaluation convention.

$$IC_t = \mathrm{Spearman}(s_{it}, r_{it})_{i \in U_t}, \qquad \overline{IC} = \frac{1}{T}\sum_{t=1}^{T} IC_t, \qquad t_{\mathrm{ordinary}} = \frac{\overline{IC}}{s_{IC}/\sqrt{T}}$$

Equation (6). Daily IC and the conventional t statistic for its mean

The conventional t statistic uses the sample standard deviation of daily IC. Its interpretation depends on conditions including temporal dependence and research selection, so it is presented as a descriptive statistic alongside supplementary block analyses. Approximately 299,000 records do not constitute approximately 299,000 independent experiments: securities on the same trading date face common market shocks, and adjacent dates can also be correlated. Symbol-day row counts are not substituted for the effective temporal sample size.

A supplementary ranking metric is the difference in mean targets between the highest and lowest predicted quintiles. It measures the relationship between prediction-based groups and the rank-normalized target, in target-score units rather than returns. Reporting this measure separately from actual long-short portfolio returns avoids interpreting standardized target differences as tradable returns.

6.2 Matched Models and Scope of Comparison

Four core variants are saved for fold 0. N0 uses numerical features and event metadata; F0-zero retains the comparison framework but disables the text pathway; F0-additive introduces an additive text residual; and T1 uses a text pathway with fold-local domain adaptation. Comparisons use seed 42 and the corresponding saved validation members, with differences calculated on common dates. Weight sources and text implementations for each configuration are documented in the saved manifests. Short names are used here to explain the comparison without conflating distinct FinBERT studies.

N0 and F0-zero have identical numerical-state hashes and validation IC, providing a pathway-alignment control. This demonstrates that the comparison framework reproduces the numerical baseline under the implemented zero-text setting. The difference between F0-additive and N0 is the direct observation of the implemented incremental text contribution in this fold, while T1 describes the ranking information provided by the text pathway on its own. The text comparison covers only one fold and one seed, so it cannot estimate general effects across all periods or text-modeling approaches.

The text encoder is ProsusAI/finbert, corresponding to Araci's work [3], fixed at revision 4556d13015211d73dccd3fdd39d39232506f3e43. BERT has 12 layers, a hidden dimension of 768, and 12 attention heads, with a maximum input length of 128 tokens. Fold 0 domain adaptation reads only training-period headlines from 2010-02-04 to 2011-12-09 and does not read return labels. Retaining the earliest event versions and deduplicating headline clusters reduces 29,773 candidates to 18,619 headlines, of which 16,769 are used for masked-language-model training and 1,850 for date-separated validation.

Domain adaptation uses 15% dynamic masking and an 80/10/10 replacement rule. The microbatch size is 32, with 8 accumulation steps and an effective batch size of 256. The learning rate is 0.0002, training runs for 1 epoch, and 66 optimization updates are completed. Query/value LoRA settings are rank 8, alpha 16, and dropout 0.05. Both LoRA and the MLM prediction head are trained at this stage, with 24,358,458 recorded trainable parameters. Only 48 LoRA tensors are transferred to the supervised stage, excluding the MLM head. The supervised learning rates are 0.00002 for LoRA and 0.0001 for newly added heads.

The supervised text pathway applies LayerNorm, a linear transformation from 768 to 128 dimensions, GELU, and 0.1 Dropout to headline CLS representations, followed by normalized pooling with positive relation weights. Each relation weight is the reciprocal of the number of symbols associated with the event. T1 is therefore a relation-weighted text branch that also uses relation metadata. F0 adds the numerical score to a text residual multiplied by a sigmoid numerical gate; text-modality Dropout during training is 0.15. T1 and F0 have 395,009 and 400,387 trainable parameters, respectively, with total parameter counts of 109,877,249 and 109,882,627. Although F0-zero trains only 5,345 numerical parameters, it still instantiates the complete model, so its runtime cannot be treated as the standalone deployment time of N0.

Matched refers here to alignment of validation observations, targets, date schedules, and seeds, not equality of parameter counts or computational budgets. In the fusion model, the numerical parameter block and the text/gating parameter block are each separately clipped to norm 1. The times in Table 5 are recorded for those particular runs. They exclude the full cost of external data preparation and do not support speed rankings across hardware platforms.

6.3 Supplementary Statistical Analyses

Daily IC is extracted from the saved validation manifests and summarized by calendar year to describe the direction and magnitude of temporal variation. For the N0 mean and paired model differences, moving blocks of 5, 10, and 20 adjacent eligible evaluation dates are sampled with replacement. Missing evaluation trading dates are not imputed, so block length is not equivalent to a count of consecutive trading sessions on the complete exchange calendar. Each block preserves the internal order of neighboring observations. Resampling is repeated 2,000 times, and the 2.5th and 97.5th percentiles of the bootstrap distribution are reported. The workbook records the algorithm convention, replication count, and block lengths; interval-specific random seeds are listed in Table 7 and the analysis metadata.

For incremental-text comparisons, the comparison model's IC minus N0 IC is first calculated on each common trading date, and this paired-difference sequence is then resampled. The algorithm draws ceiling(T/L) blocks with replacement from all overlapping, non-circular blocks, concatenates them, and truncates the result to the original sequence length T. Pairing controls for the shared date environment faced by the models and is preferable to treating the two daily sequences as independent samples. For pooled N0 results, resampling is performed within each fold before concatenation, avoiding treatment of fold boundaries as continuous time. Years with short samples retain their actual date counts rather than being presented as complete annual observations.

These intervals describe sampling variation in saved results under the corresponding block assumptions and are post-hoc robustness summaries. They do not retrain the network or incorporate the entire sequence of checkpoint selection, hyperparameter trials, and portfolio selection into resampling. Consequently, even an interval entirely on one side of zero supports only a conditional interpretation of the current development results, not a claim that an independent test has been passed.

7 Experimental Results

7.1 Ranking Performance across Folds

The three N0 development-validation windows contain a total of 298,769 symbol-day observations across 1,350 dates. The pooled mean IC is 0.020877, the median IC is 0.019575, the proportion of dates with positive IC is 55.48%, and the conventional t-statistic is 5.067. Mean IC is positive in all three folds. Fold 1 has a mean of 0.02948, compared with 0.01633 in Fold 0 and 0.01668 in Fold 2.

Table 4. N0 ranking results on development-validation data

Fold	Dates	Mean IC	Median IC	Conventional t	Positive IC	Target-group spread
0	448	0.01633	0.01710	2.711	54.91%	0.05705
1	455	0.02948	0.02857	3.762	58.46%	0.08180
2	447	0.01668	0.00896	2.255	53.02%	0.05748
Pooled	1350	0.02088	0.01957	5.067	55.48%	0.06554

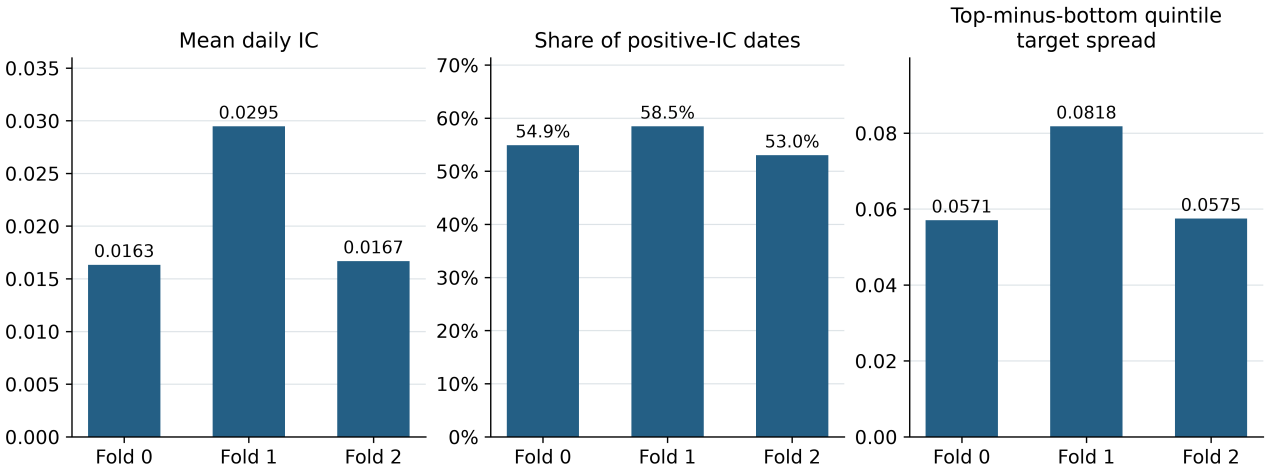


Figure 4. Mean IC, proportion of dates with positive IC, and target-quintile spread by fold. IC and target-score spreads are ranking-related quantities rather than percentage returns; the positive-IC proportion is calculated across dates.

Positive means across folds provide directionally consistent observations over three historical periods, but differences between folds are also important. The higher mean in Fold 1 indicates that the pooled result does not arise from an identical relationship in every period. Further research should jointly examine feature distributions, cross-sectional coverage, market conditions, and target noise to determine whether the differences reflect sample coverage or model adaptability. The three means alone cannot identify the cause.

A mean IC of approximately 0.02 indicates a small correlation between predictions and return rankings. In financial ranking tasks, small correlations can produce portfolio differences worth investigating, but this possibility depends on risk exposures, execution, and costs. Accordingly, this study characterizes the result as a positive ranking relationship in development validation and directly examines the effect of simplified costs in Section 8, rather than interpreting positive IC as a promise of profitability.

Published 26 September 2026 | CC BY-NC 4.0 discoveryopen.org

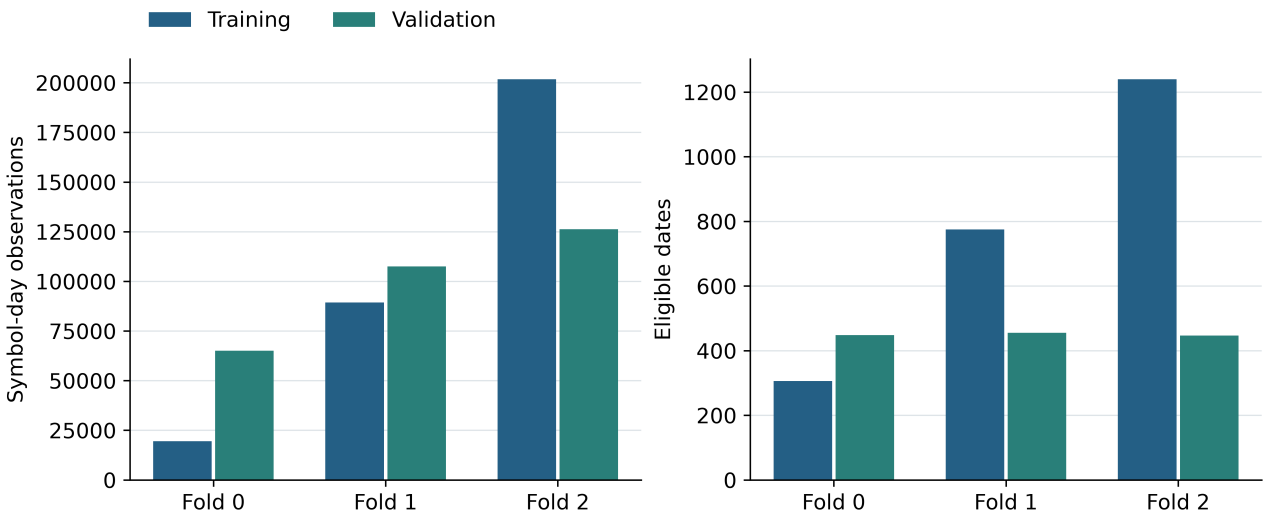


Figure 5. Effective training and development-validation sample sizes. The training window expands across folds; the counts correspond to accepted training observations rather than the size of the raw dataset.

The pooled mean IC gives equal weight to trading dates and cannot be obtained by simply weighting folds by their numbers of validation rows. A cross-sectional correlation is calculated separately for each trading date, so dates with more observations do not receive more weight in the final mean solely because they contain more rows. The accompanying workbook retains the number of dates in each fold and uses formulas to verify the pooled result obtained by weighting the three fold means by their date counts.

7.2 Training Progress and Checkpoint Trajectories

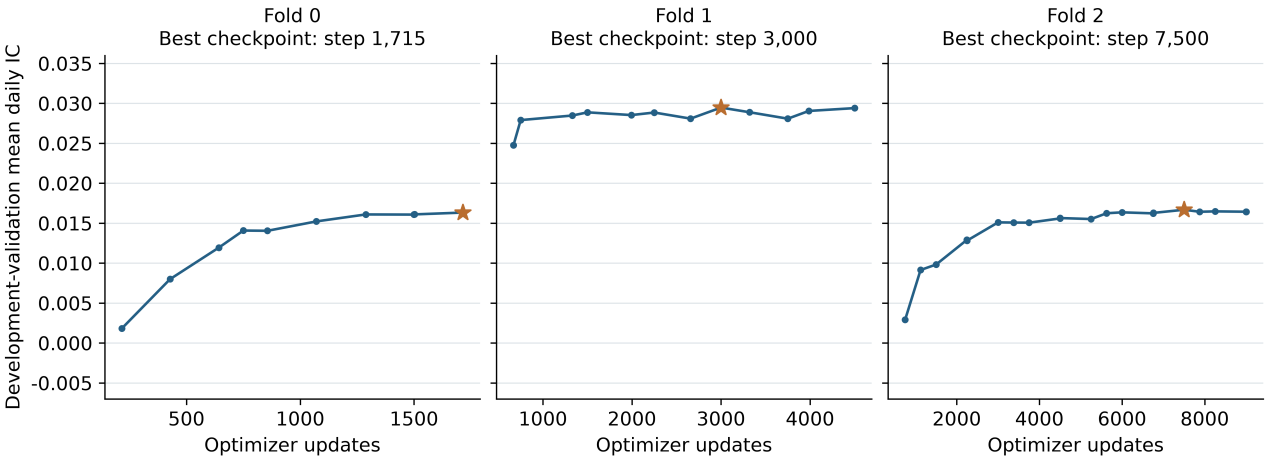


Figure 6. Development-validation IC trajectories for the three N0 folds. Points represent saved validation events, and highlighted markers identify the best steps ultimately selected. Connecting lines are visual aids and do not imply that validation was performed at every step.

The training trajectories help explain why the released checkpoints differ from the final training steps. Fold 1 reached its selected best value at step 3,000 and continued training through step 4,500. Fold 2 selected step 7,500 and ended at step 9,003. Subsequent training did not produce an improvement meeting the threshold, triggering the early-stopping rule. The best value in Fold 0 occurred at the last saved training stage.

Published 26 September 2026 | CC BY-NC 4.0 discoveryopen.org

These trajectories document the selection process while also showing that validation metrics were used to select the models. Treating the results at the best points as unbiased final estimates would overlook this selection. A subsequent independent evaluation should fix preprocessing, the network, stopping rules, and portfolio rules before applying a predefined evaluation once to a period not involved in development, rather than continuing to choose the best training step within the new window.

7.3 Matched Comparison of Incremental Text Information

Table 5. Matched comparison of four variants in Fold 0 with seed 42

Variant	Mean IC	Relative to N0	Best step	Training seconds
N0	0.01633	+0.000000	1715	274.0
F0-zero	0.01633	+0.000000	1715	1309.5
F0-additive	0.01587	-0.000461	1715	1462.9
T1	0.00396	-0.012366	1071	1105.0

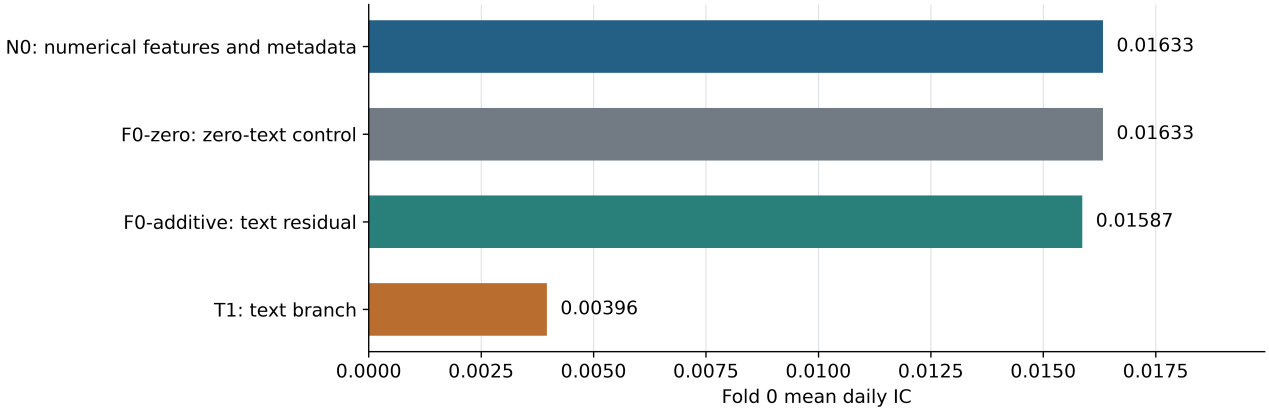


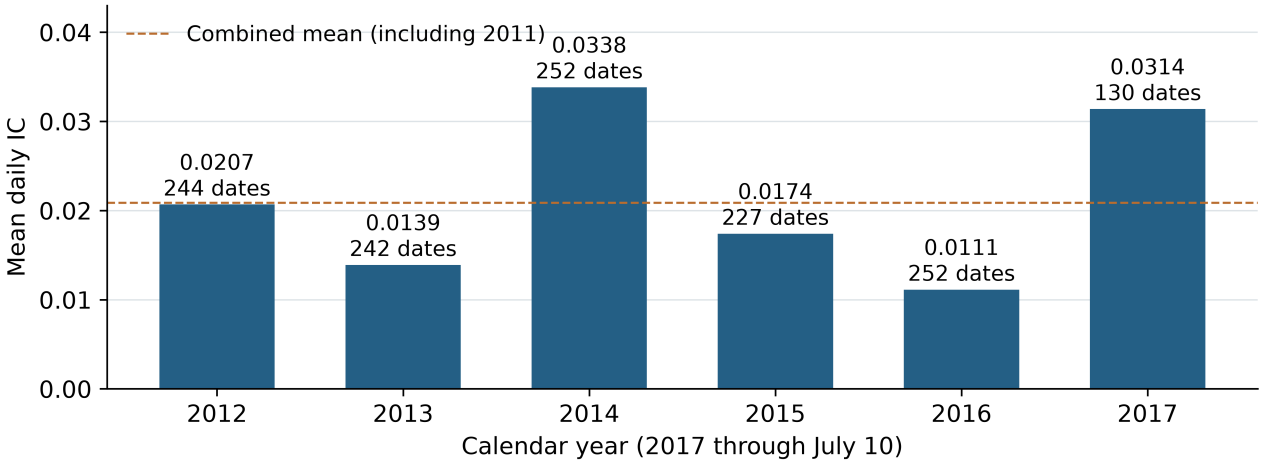
Figure 7. Mean daily IC of the four variants in Fold 0. N0 matches the zero-text control; the point estimate for the additive text residual is close to, and slightly below, N0. The figure describes results for one fold and one seed.

The additive text-residual model has an IC of 0.015869, approximately 0.000461 below N0. This difference is small relative to daily fluctuations, so the point estimates alone do not establish general statistical superiority of the numeric model over text fusion. T1 has a mean IC of 0.003964, approximately 0.012366 below N0. Together, these results show that the current text pathway and fusion configuration have not delivered a higher mean than the numeric baseline in the completed matched fold.

Several mechanisms could generate this result: structured fields may already capture some event information; text representations may be imperfectly aligned with the intraday return-ranking objective; limited samples or training budgets may affect fusion; and event-time mapping may attenuate information. This study did not conduct controlled experiments that isolate each mechanism. These possibilities are therefore testable explanations rather than causes established by the experiment. Identifying the causes requires more detailed ablations of text quality, event types, and fusion structures.

The zero-text control has a separate methodological value: it aligns the numeric pathway in the comparison framework with N0. If the control itself produced different results, the apparent text increment could be confounded with differences in numeric initialization, sample membership, or optimization. The observed agreement reduces this particular source of confounding but does not replace replication across folds and seeds.

7.4 Annual Variation and Block-Based Uncertainty



2011 has only 3 dates (mean IC 0.1378; Table 6) and is not plotted as a comparable yearly group.

Figure 8. N0 daily IC aggregated by calendar year. The initial and final years, together with gaps between folds, result in incomplete coverage for some years. The actual number of evaluation dates is annotated for each year.

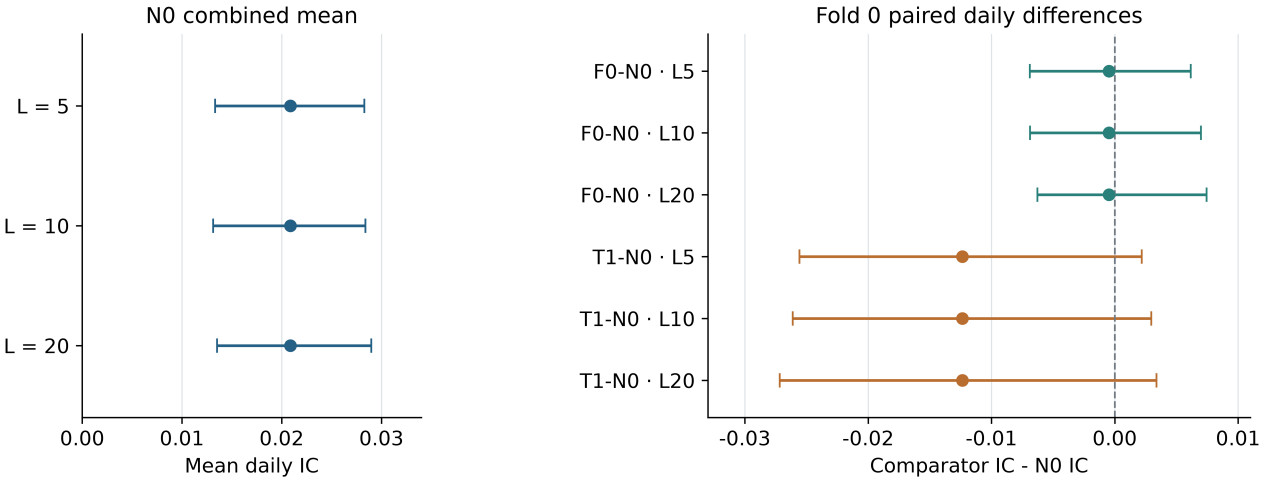
Table 6. Annual N0 ranking results and date coverage

Year	Dates	Mean IC	Positive-IC dates	Actual coverage
2011	3	0.13780	66.67%	2011-12-27 — 2011-12-30
2012	244	0.02068	55.74%	2012-01-03 — 2012-12-31
2013	242	0.01390	55.79%	2013-01-02 — 2013-12-31
2014	252	0.03383	57.14%	2014-01-02 — 2014-12-31
2015	227	0.01740	55.07%	2015-01-02 — 2015-12-31
2016	252	0.01113	52.78%	2016-01-04 — 2016-12-30
2017	130	0.03139	56.92%	2017-01-03 — 2017-07-10

The annual decomposition examines whether the pooled mean conceals local variation. Compared with fold-level summaries, the calendar-year view makes it easier to locate periods of changing performance, but it cannot independently identify the causal role of market conditions. In particular, when sample coverage changes over time, differences between years reflect both changes in the environment and changes in the set of symbols eligible for the study. These annual results are not used to reselect models or

Published 26 September 2026 | CC BY-NC 4.0 discoveryopen.org

exclude less favorable years.



95% moving-block bootstrap percentile intervals; 2,000 replicates. L denotes eligible-date block length.

Figure 9. Moving-block bootstrap 95% percentile intervals. The left panel shows pooled mean IC for N0, and the right panel shows paired IC differences in Fold 0. Block lengths are 5, 10, and 20 adjacent valid evaluation dates. The intervals apply only to the saved series of the selected models.

Table 7. Supplementary block-bootstrap results

Quantity	Block length	Mean	95% percentile interval	Random seed
N0 pooled	5	0.02088	[0.01332, 0.02829]	20270923
N0 pooled	10	0.02088	[0.01313, 0.02840]	20270928
N0 pooled	20	0.02088	[0.01352, 0.02898]	20270938
F0 - N0	5	-0.00046	[-0.00689, 0.00617]	20262923
F0 - N0	10	-0.00046	[-0.00687, 0.00699]	20262928
F0 - N0	20	-0.00046	[-0.00627, 0.00744]	20262938
T1 - N0	5	-0.01237	[-0.02556, 0.00219]	20261923
T1 - N0	10	-0.01237	[-0.02612, 0.00296]	20261928
T1 - N0	20	-0.01237	[-0.02718, 0.00338]	20261938

With a block length of 10 adjacent valid dates, the interval for N0 pooled mean IC is [0.01313, 0.02840]. The paired-difference interval for F0-additive relative to N0 is [-0.00687, 0.00699], and that for T1 relative to N0 is [-0.02612, 0.00296]. Both model differences include zero at all three block lengths. The results therefore support the statement that no improvement from incremental text information was observed in this matched experiment. They do not establish statistical superiority of N0 over the text variants or equivalence between F0 and N0.

The intervals for N0 mean IC lie above zero at all three block lengths, consistent in direction with the positive fold means. However, these intervals are conditional on the selected models and do not eliminate development-selection bias or cross-fold dependence arising from expanding training histories. The zero difference and zero-width interval for the zero-text control follow from the verified agreement between pathways. They should be interpreted as an implementation control rather than an independent replication.

8 Exploratory Portfolios and Transaction Costs

8.1 From Ranking Scores to Portfolios

To examine the economic implications of the ranking relationship, the existing development analysis selects securities from both tails of the daily score distribution and forms equally weighted long and short positions. Long notional exposure is 1, absolute short notional exposure is 1, and gross exposure is 2. The holding period runs from the market open to the close. Gross return equals the mean long-side return minus the mean short-side return. This defines a research portfolio without margin requirements, stock-borrow eligibility, or actual execution constraints.

Daily returns were saved for the quintile portfolio selecting 20% of securities on each side. A separate scan covers selection fractions of 5%, 10%, 15%, 20%, and 30% per side. The 10% fraction attracted attention after inspection of the scan and is therefore an exploratory selection. This study presents the complete scan and the original 20% portfolio separately, neither relabeling the 20% daily return series as a 10% portfolio nor reporting only the more favorable fractions.

8.2 Consistent Cost Units

One basis point (1 bp) equals 0.01%, or 0.0001 in decimal form. The cost parameter b denotes the cost of each one-way execution, charged separately on entry and exit. Because both the long and short sides require one entry and one exit, a complete intraday round trip deducts 4b basis points. Under arithmetic annualization using 252 trading days per year, a one-way execution cost of 1 bp corresponds to a fixed annual deduction of 10.08 percentage points.

$$R_t^{\mathrm{net}}(b) = R_t^{\mathrm{long}} - R_t^{\mathrm{short}} - \frac{4b}{10{,}000}, \qquad R_{\mathrm{ann}}^{\mathrm{arith}}(b) = 252\,\overline{R^{\mathrm{net}}(b)}$$

Equation (7). Simplified costs and annualized arithmetic return

Equation (7) defines a transparent and reproducible cost scenario rather than precise execution prices in an actual market. Costs are not modeled as functions of trading volume, spreads, order size, or volatility, and stock-borrow and financing costs are excluded. If actual costs vary across securities and trading dates, they will affect both mean return and volatility rather than merely shifting the daily mean return.

8.3 Selection-Fraction Scan and Break-Even Costs

Table 8. Exploratory cost scan across tail-selection fractions

Fraction per side	Gross annualized	Gross Sharpe	Net at 1 bp	Net at 2 bp	Break-even bp
5.00%	29.28%	1.109	19.20%	9.12%	2.905
10.00%	26.73%	1.336	16.65%	6.57%	2.651
15.00%	20.89%	1.224	10.81%	0.73%	2.072
20.00%	17.05%	1.143	6.97%	-3.11%	1.692
30.00%	11.15%	0.916	1.07%	-9.01%	1.106

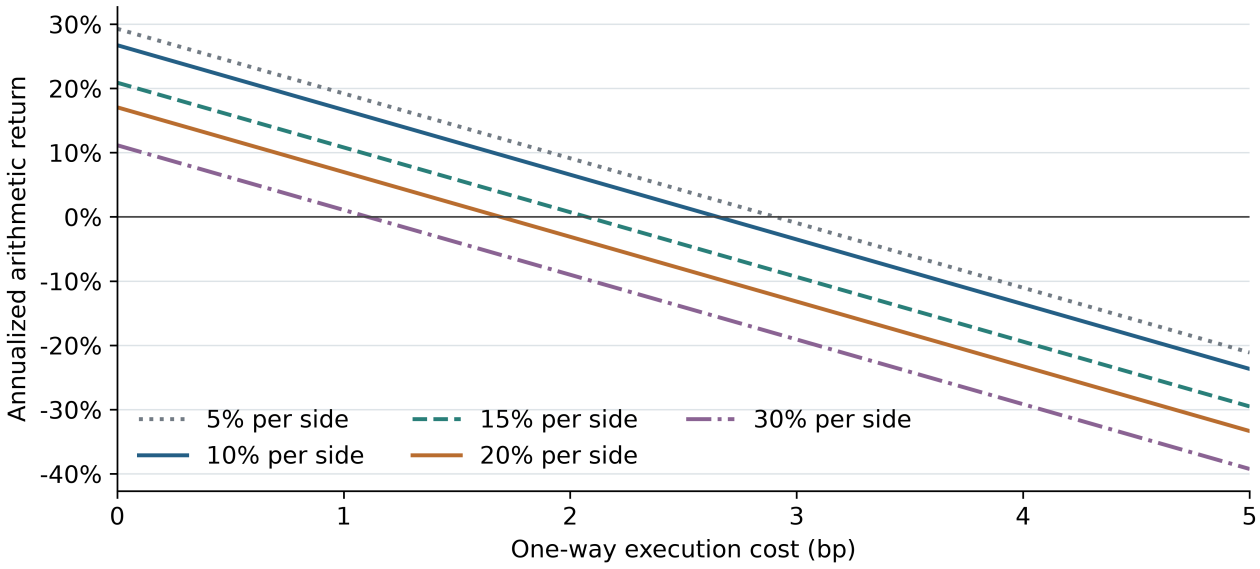


Figure 10. Annualized arithmetic returns across selection fractions and one-way execution costs. All fractions are shown together. Each line's slope follows from the same assumption of 4 daily one-way execution charges, not from retraining the model.

The portfolio selecting 10% per side has a gross annualized arithmetic return of approximately 26.73% and a gross Sharpe ratio of approximately 1.34. Its returns under costs of 1 bp and 2 bp are approximately 16.65% and 6.57%, respectively. The portfolio selecting 20% per side has a gross annualized arithmetic return of approximately 17.05%, declining to approximately 6.97% and -3.11% under the same cost assumptions. Positive ranking relationships and positive gross returns therefore do not eliminate cost sensitivity.

The break-even cost is obtained by dividing the gross mean daily return by 4 and converting the result to basis points. It is approximately 2.65 bp for the 10% fraction and 1.69 bp for the 20% fraction. These quantities describe the cost allowance under the current simplified rules and inherit the effects of model and portfolio selection during development. They do not establish whether a particular broker or trading venue can achieve those costs.

8.4 Additional Interpretation of Risk and Drawdown

Table 9. Costs, volatility, and maximum drawdown for the 20%-per-side portfolio

One-way cost	Annualized arithmetic return	Annualized volatility	Sharpe	Maximum drawdown
0 bp	17.05%	14.92%	1.143	-19.64%
1 bp	6.97%	14.92%	0.467	-26.15%
2 bp	-3.11%	14.92%	-0.208	-34.32%
5 bp	-33.35%	14.92%	-2.235	-84.47%

The 20%-per-side portfolio has gross annualized volatility of approximately 14.92% and a gross maximum drawdown of approximately 19.64%. Under the simplified 2 bp cost scenario, its Sharpe ratio is approximately -0.21 and maximum drawdown is approximately 34.32%. A fixed daily deduction preserves the volatility of daily returns but changes their mean, the compounded wealth path, and drawdown. Reporting only gross Sharpe or mean IC therefore omits information relevant to practical usefulness.

This study does not decompose portfolio returns into industry, style-neutral, or other risk-premium components. Because controls for industry, market capitalization, volatility, beta, and liquidity exposures have not been completed, the observed long-short differences may also reflect known risk characteristics. Further research seeking an investment interpretation should undertake risk attribution alongside more realistic trading constraints, rather than only continuing to optimize the net-return curve.

9 Release Reproducibility and Practical Use

9.1 From Training Checkpoints to the Inference Package

The recovery file resume.pt contains optimizer, scheduler, random-state, and recovery information, whereas best_model.pt stores the selected trainable weights and their associated binding information. The files serve different purposes. The public inference package extracts 8 network tensors from each best checkpoint, stores them in Safetensors format, and includes fold-specific scalers, a consistent field order, network configuration, and inference code. All 24 trainable tensors across the three folds match the selected training artifacts tensor by tensor. The model and documentation releases have separate roles: weights and scalers are hosted on Hugging Face, while methodological descriptions, aggregate evaluations, and lightweight usage code are hosted on GitHub.

For each fold, a new model was reconstructed after saving, its state was loaded strictly, and inference was repeated on validation observations. The maximum absolute prediction difference was 0 in every fold. Fold 2 was additionally checked on 2,048 fixed validation observations by comparing released and training-model inference along the canonical CUDA path. The recorded differences in scores and target-scale reconstruction were both 0. These checks cover the specific saving and release implementation used in this study.

Table 10. Model-release verification checks

Verification item	Result	Scope of interpretation
Reloading after saving the training model	Maximum absolute prediction difference of 0 in all three folds	The same canonical execution path
Public-tensor comparison	All 3 folds × 8 tensors match	Released weights and selected checkpoints
Fixed-sample inference in Fold 2	2,048 rows; recorded difference of 0	The canonical CUDA precision path
Matched membership in Fold 0	65,036 row keys, dates, and targets match	Identical validation membership for paired comparisons
Daily IC recalculation	Difference from the saved series below 1e-12	Daily series used for supplementary statistics
Reconstruction of 10% / 20% portfolios	Verification differences in scan statistics below 1e-12	Consistent portfolio definitions; not evidence of execution feasibility

CPU float32 and CUDA automatic mixed precision can differ in the least significant digits, so zero differences in the canonical environment do not imply bitwise identity across all hardware and software combinations. Reproduction requires a fixed field order, correct pairing of weights and scalers, consistent inference precision, and consistent temporal semantics for the inputs. Successfully loading the model with synthetic inputs verifies that the interface runs, but does not establish that externally generated real features have the same semantics as the training inputs.

9.2 Inference Procedure and Interpretation of Outputs

A compatible input file should provide all 23 base columns and 2 nonnegative count columns, while retaining a date or local row identifier for joining the outputs. The program parses the base numeric values, constructs missingness masks, imputes and scales values, concatenates a 48-dimensional tensor, and loads the specified fold for scoring. The default release entry point uses Fold 2, but the three folds are research artifacts trained over different periods and do not automatically constitute an evaluated three-model ensemble.

Scores should be ranked within the same decision date and the same eligible equity universe. Raw scores from different folds or dates should not be interpreted as comparable probabilities or return magnitudes without calibration. Converting scores into actual positions requires separately fixing the candidate universe, position mapping, risk budget, execution rules, and exit conditions, followed by a new evaluation of the complete system.

9.3 Data and Code Availability

The model page is [source link], and the documentation and usage repository is [source link]. The public release covers model weights, preprocessing parameters, configuration, the inference interface, and aggregate evaluations. Source-data terms and versions must be checked against their respective sources; the model-file license does not automatically grant redistribution rights for the underlying news or market data.

The accompanying research workbook contains aggregate results, calculation definitions, supplementary analyses, and a source index. The model is distributed under the CC BY-NC 4.0 noncommercial license. The underlying data, model, and code are each subject to their respective license terms.

10 Discussion

10.1 Research Value Supported by the Current Evidence

The most direct finding is that a network with 5,345 parameters achieves positive mean daily ranking relationships in the currently filtered, event-covered sample under fixed information-cutoff rules and across three forward development windows. This observation provides a baseline with an explicit parameter count, input specification, preprocessing procedure, and training rules. Its value lies in an empirically inspectable starting point, rather than in using model size to imply predictive ability.

The value of the text comparison likewise does not depend on obtaining a positive increment. Reporting only the IC of a complex model makes it difficult to determine whether its performance comes from text or other inputs. Introducing a matched numeric baseline and a zero-text control reformulates the question in terms of same-date paired differences. The current results suggest that future work should first establish the incremental contribution of text relative to structured information before intensifying text modeling, while retaining runs that do not improve performance.

Engineering checks connect the public artifacts to the object studied in the paper. If reported metrics came from one checkpoint but the release contained different weights or an incorrect scaler, external users would struggle to reproduce them. The reloading, tensor, and fixed-sample checks recorded here directly address this issue. They improve research traceability but do not substitute for independent temporal evaluation.

10.2 Statistical and Experimental Boundaries

First, development validation was used for early stopping and best-checkpoint selection. The conventional t-statistics, annual means, and supplementary block intervals in this study are all based on already-selected models and do not cover the full model-search process. Second, the public numeric experiments use only seed 42. Three folds represent three periods, not three independent random seeds. The matched text comparison is also currently confined to Fold 0, so it cannot characterize training randomness or the temporal stability of incremental text information.

Third, complete results are currently unavailable for simple reversal or momentum factors, linear or Ridge models, and gradient-boosted trees evaluated on the same sample. Without these baselines, the contribution of the nonlinear network relative to simpler structures cannot be determined. Fourth, the separate contributions of event metadata, missingness indicators, market conditions, and pairwise loss have not been established through ablation. The study can report the performance of the overall system but cannot attribute that performance to any individual design choice that has not been isolated.

Fifth, block-based treatment of temporal dependence relies on block length and conditions such as approximate stationarity. Multiple block lengths provide a degree of sensitivity analysis but do not establish robustness to every possible dependence structure. Differences across years and changes in sample coverage also complicate interpretation of a single mean. Statistical reporting should therefore be read together with data coverage and the precise experimental configuration.

Published 26 September 2026 | CC BY-NC 4.0 discoveryopen.org

10.3 Data and Trading Boundaries

The extent to which historical vendor symbols, corporate actions, data revisions, delistings, and news-source timestamps can be reconstructed determines how closely model inputs approximate information available at the time. Existing filters and temporal rules reduce some risks but do not establish a complete security-master system. Requiring at least one eligible event also conditions the sample on news coverage, a condition that must be retained when interpreting the findings.

On the trading side, opening and closing prices underpin the research labels and return calculations, whereas actual execution additionally depends on order types, queue position, executable quantity, and market impact. Short portfolios particularly require consideration of stock-borrow availability, fees, and restrictions. The fixed-cost scenarios illustrate sensitivity; actual execution feasibility, strategy capacity, and independent sources of risk-adjusted returns have not yet been verified.

10.4 Directions for Further Research

The next stage should first freeze the research protocol and then extend the evidence. Simple factors, Ridge, and LightGBM can be added under identical sample membership, information cutoffs, and preprocessing. Event metadata, market conditions, missingness indicators, and the ranking term can then be removed separately to form ablations that answer specific questions. Model comparisons should share an interpretable tuning budget and retain all candidates and selection logs.

Subsequent work should complete prespecified replication across folds and seeds, consistently reporting daily paired differences and the treatment of their temporal dependence. Only after the model, data filters, and portfolio rules have been frozen should an untouched period meeting the established eligibility criteria, or a newly established prospective paper-trading period, be used for final evaluation. If performance does not persist in that evaluation, the outcome should be reported as a research finding rather than continuing to tune on the same period while retaining its designation as a test set.

If the focus shifts toward economic value, the analysis should add industry and style-risk controls, execution constraints, stock-borrow and financing costs, and an impact model that varies with order size. If the focus shifts toward text, the numeric baseline should remain fixed while event types, text-timestamp precision, domain adaptation, and fusion structures are compared. These two directions address different questions, and experimental priorities should follow a prespecified primary metric.

11 Conclusion

This study describes the data organization, temporal constraints, 48-dimensional features, rank-normal targets, 5,345-parameter network, and complete-date training procedure of M8 N0. Saved experiments are used to report ranking performance across three development-validation windows. The pooled mean daily IC is 0.02088, with positive means in all three folds. In the matched Fold 0 comparison, the additive text residual does not exceed the mean performance of the numeric baseline, while the zero-text control agrees with the numeric pathway. Annual and block-based analyses provide supplementary descriptions of temporal variation and sampling fluctuations in the existing results.

Exploratory portfolio results show that after-cost returns are sensitive to the selection fraction and one-way execution costs. The 20%-per-side portfolio has a negative annualized arithmetic return under the

simplified 2 bp cost scenario, demonstrating the need to evaluate ranking relationships, gross portfolio returns, and executable net returns separately. Reloading and release-consistency checks establish a verifiable connection between the research checkpoints and the public model files.

The study establishes a reproducible, extensible development baseline and identifies the next questions to test: the incremental contribution of the lightweight network relative to simple baselines; the contributions of different structured information sources; the stability of incremental text information across periods and random seeds; and independent predictive performance after the protocol is frozen. Further evidence should be organized around these questions rather than solely increasing model size or extending descriptions of existing results.

References

[1] Zihan Dong, Xinyu Fan, Zhiyuan Peng. FNSPID: A Comprehensive Financial News Dataset in Time Series. arXiv preprint arXiv:2402.06698, 2024. DOI 10.48550/arXiv.2402.06698. [source link]

[2] Shihao Gu, Bryan Kelly, Dacheng Xiu. Empirical Asset Pricing via Machine Learning. The Review of Financial Studies, 33(5): 2223–2273, 2020. DOI 10.1093/rfs/hhaa009. [source link]

[3] Dogu Araci. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. arXiv preprint arXiv:1908.10063; University of Amsterdam master's thesis, 2019. DOI 10.48550/arXiv.1908.10063. [source link]

[4] Yi Yang, Mark Christopher Siy Uy, Allen Huang. FinBERT: A Pretrained Language Model for Financial Communications. arXiv preprint arXiv:2006.08097, 2020. DOI 10.48550/arXiv.2006.08097. [source link]

[5] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, Greg Hullender. Learning to Rank Using Gradient Descent. Proceedings of the 22nd International Conference on Machine Learning (ICML): 89–96, 2005. DOI 10.1145/1102351.1102363. [source link]

[6] Peter J. Huber. Robust Estimation of a Location Parameter. The Annals of Mathematical Statistics, 35(1): 73–101, 1964. DOI 10.1214/aoms/1177703732. [source link]

[7] Ilya Loshchilov, Frank Hutter. Decoupled Weight Decay Regularization. International Conference on Learning Representations (ICLR), 2019. DOI 10.48550/arXiv.1711.05101. [source link]

[8] David H. Bailey, Jonathan M. Borwein, Marcos López de Prado, Qiji Jim Zhu. The Probability of Backtest Overfitting. Journal of Computational Finance, 20(4): 39–69, 2017. DOI 10.21314/JCF.2016.322. [source link]

[9] Campbell R. Harvey, Yan Liu, Heqing Zhu. … and the Cross-Section of Expected Returns. The Review of Financial Studies, 29(1): 5–68, 2016. DOI 10.1093/rfs/hhv059. [source link]

[10] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems, 30, 2017. [source link]

[11] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, Artem Babenko. Revisiting Deep Learning Models for Tabular Data. Advances in Neural Information Processing Systems, 34, 2021. [source link]

[12] Léo Grinsztajn, Edouard Oyallon, Gaël Varoquaux. Why Do Tree-Based Models Still Outperform Deep Learning on Typical Tabular Data?. Advances in Neural Information Processing Systems, Datasets and Benchmarks Track, 35, 2022. DOI 10.52202/068431-0037. [source link]

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019, Volume 1 (Long and Short Papers): 4171–4186, 2019. DOI 10.18653/v1/N19-1423. [source link]

[14] Halbert White. A Reality Check for Data Snooping. Econometrica, 68(5): 1097–1126, 2000. DOI 10.1111/1468-0262.00152. [source link]

Appendix A. Feature Dictionary

The field order below matches the release configuration. All historical market features use only trading dates completed before the decision date, while event-structure features use the eligible event set. The descriptions in the table explain field semantics; complete window boundaries, data-cleaning rules, and vendor mappings remain defined by the corresponding training version.

Table A1. The 23 base fields and two count fields

No.	Field name	Meaning
1	asset_oc_ret_lag1	Previous-day open-to-close return
2	asset_oc_ret_mean5	Mean return over the previous 5 trading days
3	asset_oc_ret_mean10	Mean return over the previous 10 trading days
4	asset_oc_ret_mean20	Mean return over the previous 20 trading days
5	asset_oc_ret_vol5	Return volatility over the previous 5 trading days
6	asset_oc_ret_vol20	Return volatility over the previous 20 trading days
7	asset_range_lag1	Previous-day price range
8	asset_log_volume_z20	Log trading volume standardized over a 20-day window
9	asset_log_median_dollar_volume20	Log of 20-day median dollar volume
10	asset_log_amihud20	Log of 20-day Amihud-style illiquidity
11	market_median_oc_ret_lag1	Previous-day market median return
12	market_median_oc_ret_mean5	5-day mean of the market median return
13	market_median_oc_ret_vol20	20-day volatility of the market median return
14	qqq_oc_ret_lag1	Previous-day QQQ open-to-close return
15	qqq_oc_ret_mean5	Mean QQQ return over the previous 5 trading days
16	qqq_oc_ret_vol20	QQQ return volatility over the previous 20 trading days
17	event_vendor_symbol_count_max	Maximum number of symbols linked to a single event
18	event_vendor_symbol_count_mean	Mean number of symbols linked to a single event
19	relation_weight_mean	Mean event-to-symbol relation weight
20	multi_ticker_headline_share	Share of events associated with multiple symbols

No.	Field name	Meaning
21	publisher_count	Number of publishers
22	publisher_hhi	Publisher concentration measured by HHI
23	publisher_missing_share	Share with missing publisher information
24	raw_headline_count	Number of eligible raw events, log1p
25	unique_headline_count	Number of eligible deduplicated events, log1p

Appendix B. Verification Procedures and Metric Interpretation

B.1 From Original Runs to Paper Figures and Tables

The paper's data lineage begins with run manifests, public aggregate evaluations, and the existing portfolio scan. The extraction script reads fold-level dates, sample counts, best steps, validation histories, and daily IC. After checking duplicate dates, row counts, and consistency of summaries, it saves a consolidated experiment-evidence file. Annual statistics and the block bootstrap use only these daily series. The plotting program reads values from that evidence file, while the workbook uses the same source values and retains formulas for derived relationships such as costs.

The relative path and SHA-256 of each source file are recorded to identify the data version used in the analysis and support file-integrity checks. Original records, extraction scripts, figures, tables, and the workbook are linked through a shared data lineage. Statistics in the text and tables are checked consistently against this data version.

B.2 Interpreting IC, t-Statistics, and Portfolio Returns

IC is a correlation coefficient ranging from -1 to 1. A mean IC of 0.02 indicates neither a 2% return nor 52% security-level accuracy in classifying upward and downward price movements. The proportion of positive-IC dates is the share of dates on which the daily correlation coefficient exceeds zero; it is not the proportion of profitable trades. The target-quintile spread in this study is measured in rank-normal scores and likewise does not equal a return.

The conventional t-statistic measures the size of a mean relative to its estimated standard error. It does not automatically account for repeated trials, temporal dependence, or model selection. Bootstrap intervals provide another description of the saved series but do not turn data previously used in development into unseen data. Portfolio returns additionally depend on selection fractions, weights, holding periods, actual execution, and costs. Two return figures are comparable only after these elements have been fixed.

B.3 Algorithmic Steps

The training procedure can be summarized in the following order: fix sources and time zones; generate events available on each decision date and lagged market features; construct eligible symbol-day membership; apply temporal and duplicate-event filters within each fold; fit numeric and target scaling on training members; compute forward scores and the joint loss using complete dates; apply gradient clipping and AdamW updates; validate at the predefined cadence and save the best state; reload and verify; and extract public weights and check consistency. This ordering imposes substantive constraints. For example, fitting a scaler after pooling all periods and then splitting the data would change the experiment defined here.

Inference begins with a fixed input contract and proceeds through field checks, count-validity checks, missingness indicators, training-median imputation, training-scale transformation, clipping, tensor concatenation, and forward scoring. For same-date ranking, outputs must also be joined to the original dates and security identifiers. Upstream feature generation is responsible for temporal and membership semantics, while the inference program is responsible for the interface and numeric transformations. Both determine whether the final inputs are compatible.

Publication declarations

Jiexiao Zhang. Discovery 1(1), e1 (2026).

Author contributions

Jiexiao Zhang is the sole author and was responsible for conceptualization, methodology, software implementation, data curation, model training, experimental analysis, visualization, and manuscript preparation.

Funding

This research received no external funding.

Data and code availability

Date-level aggregate evidence, daily information coefficients, portfolio diagnostics, training-validation trajectories, and the calculation workbook accompany this article as supplementary materials. Access to underlying news and market data is governed by the terms and versions of the respective sources. Model weights, fold-specific preprocessing parameters, configuration, and the inference interface are available at huggingface.co/NeoZJX/m8-n0-numeric-equity-ranking. Methodological documentation, aggregate evaluations, and usage code are available at github.com/JiexiaoZhang1/m8-n0-numeric-equity-ranking. Third-party datasets and code remain subject to their respective terms.

Use of generative AI and AI-assisted tools

OpenAI Codex was used to assist with manuscript drafting, English translation, editorial organization, figure and table preparation, and consistency checks. Reported experimental results are drawn from the saved research artifacts identified in the article and supplementary evidence. AI-assisted translation and document preparation did not generate new experimental observations.

Copyright and reuse

Article record: https://discoveryopen.org/index.php/discovery/article/view/m8-n0-equity-ranking