Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082842

ARTICLE

Feature-Wise Linear Modulation for Heterogeneous-Frequency Multimodal Fusion in Temporal Sequence Encoders

Maurice Kyla Octaviano and Jin-Taek Seong*

Graduate School of Data Science, Chonnam National University, Gwangju, Republic of Korea

*Corresponding Author: Jin-Taek Seong. Email: jtseong@jnu.ac.kr

Received: 24 March 2026; Accepted: 27 May 2026; Published: 23 July 2026

ABSTRACT: Integrating high-frequency sequential signals with low-frequency contextual descriptors into a unified deep encoder is a recurring challenge in computational modelling, exemplified by cross-sectional stock ranking where price dynamics must be jointly modelled with quarterly accounting fundamentals. Existing approaches use late concatenation, where the contextual signal influences only the final prediction head and cannot shape upstream feature extraction. We propose Feature-wise Linear Modulation (FiLM) as an intermediate conditioning mechanism: fundamentals generate per-channel scaling (gamma) and shifting (beta) parameters that affinely transform the encoder's intermediate representations before aggregation. The same price sequence thus yields different temporal features depending on the firm's fundamental profile, which we hypothesise reduces signal variability across heterogeneous market regimes by allowing the encoder to amplify or suppress patterns based on contextual quality. We instantiate FiLM across recurrent (LSTM), convolutional (TCN), and attention-based (iTransformer) encoders, evaluated on China A-share equities (2010–2024). Across the convolutional and recurrent encoder families, the primary benefit of FiLM conditioning is improved signal stability—formally measured as the standard deviation of RankIC across rebalancing dates—and risk-adjusted performance, rather than mean predictive accuracy. The gain depends critically on where modulation is applied: pre-aggregation conditioning on temporally-rich representations produces the largest variance reduction. FiLM-TCN, which modulates the convolutional feature map before pooling, achieves RankIC of 0.1415, annualised Sharpe of 1.633, and IC hit rate of 80.4% net of transaction costs. The insight that intermediate conditioning improves signal stability rather than raw accuracy may inform analogous fusion problems in other sequential modelling domains.

KEYWORDS: Feature-wise linear modulation; temporal convolutional network; iTransformer; multimodal fusion; heterogeneous-frequency fusion; quantitative finance; stock ranking; China A-shares

1 Introduction

Integrating a low-frequency contextual descriptor with a high-frequency sequential signal—such that the former governs how the latter is encoded rather than merely how the resulting representation is interpreted—is a recurring architectural challenge in deep learning for sequential data. A natural instantiation of this problem arises in cross-sectional stock ranking [1], where sequential price dynamics must be jointly modelled with periodic accounting fundamentals disclosed at quarterly frequency. Unlike point forecasting of individual asset returns, cross-sectional ranking is concerned with the *relative ordering* of stocks at each rebalancing date, making it naturally suited to learning-to-rank formulations and robust to the absolute level of market returns.

Two broad categories of information are relevant to this task. Price and market microstructure data—open, high, low, close, volume, turnover, and their derived technical indicators—encode short- to medium-term behavioural dynamics available at daily or higher frequency. Quarterly accounting fundamentals—profitability ratios, earnings, and capital structure metrics—encode the underlying economic quality of the firm but are disclosed with a lag. These modalities are complementary: price data captures *when* a stock is moving, while fundamentals provide context for *why* a given pattern may be persistent or mean-reverting. Critically, they differ in temporal frequency by an order of magnitude, creating a heterogeneous-frequency fusion problem that is not well addressed by standard late concatenation.

The dominant approach to combining these modalities in deep learning-based stock prediction is *late fusion*: a temporal encoder processes the price sequence independently, and the resulting representation is concatenated with the fundamental vector before a prediction head [2]. While simple and effective, late fusion has a structural limitation—the fundamental context is never allowed to influence *how* temporal patterns are extracted. A model that has learned to recognise momentum patterns, for instance, cannot amplify those patterns for high-quality firms and suppress them for distressed ones without first seeing the fundamental signal, which under late fusion arrives only after the temporal encoding is complete.

Feature-wise Linear Modulation (FiLM) [3] offers a principled alternative. Originally proposed for visual question answering, FiLM applies a learned affine transformation—parameterised as a function of a conditioning input—to intermediate representations within a neural network. Applied to temporal sequence encoding, FiLM allows the low-frequency contextual signal to modulate *every layer* of the encoder, enabling the model to adapt its temporal pattern extraction to the contextual profile of each input at each time step.

In this paper, we investigate FiLM conditioning as a mechanism for heterogeneous-frequency multimodal fusion, instantiated in the context of cross-sectional stock ranking. We evaluate the approach across three temporally structured encoder families: a two-layer LSTM [4], a six-block dilated Temporal Convolutional Network [5], and a two-layer inverted Transformer [6]. For each family we construct a baseline variant using late-fusion concatenation and a FiLM-conditioned variant, yielding six deep models in total, supplemented by an XGBoost tabular baseline [7].

All models are evaluated on a large-scale longitudinal dataset of China A-share equities spanning January 2010 to December 2024, with monthly rebalancing and a strict walk-forward train/validation/test split. Chinese equity markets have distinctive structural properties—high retail participation, regulatory interventions, and a distinctive fundamental disclosure regime—and all conclusions are scoped to this single benchmark. The extent to which the findings and the conditioning framework generalise to other sequential prediction domains is an open and important direction for future work.

Main Contributions

1. **Intermediate conditioning as a fusion mechanism.** We propose and systematically evaluate FiLM as a mechanism for integrating a low-frequency contextual descriptor into a high-frequency temporal sequence encoder via intermediate affine modulation, providing a principled alternative to late-fusion concatenation across three encoder families—recurrent, convolutional, and attention-based.
2. **Architecture–fusion interaction.** The benefit of intermediate conditioning depends on where modulation is applied within the encoder: feature-map modulation (FiLM-TCN) produces the most consistent gains, hidden-state modulation (FiLM-LSTM) yields mixed results, and per-layer attention modulation (FiLM-iTransformer) produces intermediate but less parameter-efficient improvements—a finding that motivates more targeted architectural investigation in future work.

3. **Signal consistency as the primary locus of gain.** For the convolutional and recurrent encoder families, FiLM conditioning improves signal stability and risk-adjusted metrics more reliably than mean predictive accuracy, suggesting that intermediate conditioning reduces variance in the predictive signal rather than uniformly amplifying it.

The remainder of the paper is organised as follows. Section 2 reviews related work on deep learning for stock prediction and multimodal fusion. Section 3 describes the dataset construction and feature engineering pipeline. Section 4 presents the model architectures, training objective, and experimental setup. Section 5 reports and analyses the results. Section 6 discusses the findings, and Section 7 concludes.

2 Related Work

2.1 Cross-Sectional Ranking and Machine Learning

The application of machine learning to cross-sectional ranking has a substantial and growing history. Classical factor models identify firm characteristics—momentum, value, size—as predictors of future returns [1,8,9], but assume linear relationships and rely on hand-crafted features. Gu et al. [2] provide a large-scale empirical comparison of machine learning methods for equity return prediction, demonstrating that non-linear models including neural networks and gradient-boosted trees substantially outperform linear factor models, and establishing the rank-based cross-sectional evaluation protocol that we adopt in this work. Kelly et al. [10] further show that firm characteristics can be understood as covariances in a latent factor structure, providing theoretical grounding for the use of fundamental features as inputs to predictive models. Feng et al. [11] extend this line of work by showing that deep neural networks can generate latent pricing factors from firm characteristics, outperforming conventional characteristic-sorted portfolios on cross-sectional return prediction. Fan and Shen [12] demonstrate that a simple MLP-based architecture with channel-mixing operations achieves competitive stock ranking performance, highlighting that architectural simplicity with well-designed feature mixing can rival attention-based approaches.

The generalisability of these findings across markets has received increasing attention. Leippold et al. [13] conduct a comprehensive ML study of the China A-share market, applying a range of algorithms to 94 stock-level characteristics and demonstrating that ML models dominate linear factor benchmarks. Importantly for our work, they highlight that the China A-share market has distinctive structural properties—high retail investor participation, liquidity-driven predictability, and an opaque fundamental disclosure environment—that distinguish it from the US market studied in Gu et al. [2]. Ma et al. [14] further demonstrate using instrumented principal component analysis that firm characteristics condition latent factor loadings in the Chinese market—a result that motivates our hypothesis that fundamentals can act as a mechanism switcher for temporal price patterns rather than merely as additive predictors. Our work builds on this foundation by investigating whether deep temporal sequence encoders with intermediate conditioning via FiLM provide incremental value over tabular approaches on this benchmark.

2.2 Deep Sequential Encoders

Recurrent neural networks, particularly LSTMs [4], have been widely applied to sequential modelling owing to their ability to capture long-range dependencies in time-series data. Temporal Convolutional Networks [5] offer an alternative to recurrent architectures, using dilated causal convolutions to achieve large receptive fields with parallelisable training and competitive empirical performance on a range of sequence modelling benchmarks. The Transformer architecture [15], originally proposed for natural language processing, has subsequently been adapted for time-series forecasting in numerous variants. Among these, Nie et al. [16] propose PatchTST, which segments time series into subseries-level patches as input tokens, reducing

attention complexity and improving long-range forecasting accuracy. Liu et al. [6] further invert the standard attention mechanism to operate across the variate dimension rather than the time dimension, achieving strong results on multivariate forecasting benchmarks. In our work, we evaluate all three encoder families as backbones for cross-sectional sequential ranking.

Most closely related to our work, Li et al. [17] propose MASTER, a conditioning-based stock Transformer that uses an external market-index signal to gate and select features dynamically. Like our approach, MASTER is motivated by the observation that an external conditioning signal should modulate how individual features are processed, rather than being fused only at the output. However, MASTER conditions on market-index aggregates rather than firm-level contextual descriptors, operates at daily frequency with an 8-day lookback window, and explicitly models inter-stock correlations—design choices that reflect a different but complementary perspective on the conditioning problem.

Gradient-boosted trees, and XGBoost [7] in particular, remain highly competitive on tabular prediction tasks despite the proliferation of deep learning approaches. We include XGBoost as a strong tabular baseline to assess whether deep temporal encoders provide incremental value over well-tuned classical methods on this benchmark [2].

2.3 Multimodal Fusion Strategies

A recurring challenge in multimodal sequential prediction is the integration of heterogeneous data streams that differ in temporal frequency, dimensionality, and information content. In the setting studied here, these streams comprise a high-frequency sequential signal (price dynamics) and a low-frequency contextual descriptor (accounting fundamentals). Early work on multimodal fusion for sequential prediction focused predominantly on combining sequential signals with auxiliary text. Xu and Cohen [18] propose a deep generative model that jointly encodes tweet sentiment and historical price sequences, demonstrating that cross-modal fusion consistently outperforms unimodal baselines.

The dominant fusion approach in the broader machine learning literature remains *late fusion*: each modality is encoded independently and the resulting representations are concatenated before a prediction head [2]. While simple and widely adopted, late fusion has a structural limitation—the conditioning modality cannot influence how the primary modality is encoded, which may prevent the model from learning cross-modal interactions at the representational level rather than merely at the output stage.

Alternative fusion strategies have been explored in the broader multimodal learning literature. Early fusion concatenates raw features before any encoding, preserving all cross-modal interactions but making the encoder responsible for disentangling them from a high-dimensional joint input. Cross-attention mechanisms [15] allow each modality to attend over the other's representations, enabling richer interaction at the cost of increased model complexity and potential overfitting on smaller datasets. Gating-based approaches, such as the feature selection mechanism in MASTER [17], use an external signal to rescale input features before temporal encoding, which is closely related to FiLM but operates at the input layer rather than intermediate representations. What is absent from existing approaches is a mechanism that allows the contextual signal to interact with learned temporal representations at multiple levels of abstraction without requiring structural modification to the encoder—the gap that motivates the approach developed in Section 4.

2.4 Feature-Wise Linear Modulation

Feature-wise Linear Modulation (FiLM) was introduced by Perez et al. [3] for visual question answering, where a language conditioning signal modulates intermediate representations in a convolutional image

encoder via learned affine transformations. The original paper noted that FiLM is a generalisation of conditional normalisation techniques previously used for image stylisation and speech recognition, suggesting its applicability well beyond the vision domain.

Subsequent work has confirmed this generality across modalities and architectures. Birnbaum et al. [19] propose Temporal FiLM (TFiLM), which uses a recurrent network to modulate the activations of a convolutional sequence model, demonstrating improvements on text classification and audio super-resolution. Brockschmidt [20] extends FiLM to graph neural networks, showing that feature-wise modulation of message-passing enables target-node representations to gate incoming messages, outperforming standard GNN aggregation schemes on multiple graph learning benchmarks. Together, these works establish that FiLM's core mechanism—element-wise scaling and shifting of intermediate features conditioned on an external signal—is a flexible, parameter-efficient conditioning strategy that has been successfully applied across vision, language, audio, and graph-structured domains. However, all prior applications share the same temporal granularity between the conditioning signal and the primary signal. Our work introduces a qualitatively different setting: the conditioning signal (quarterly fundamentals) operates at a frequency an order of magnitude lower than the primary signal (daily price dynamics), requiring the generator to bridge two heterogeneous temporal scales rather than modulate within a single one—the core architectural problem that motivates the approach developed in Section 4.

2.5 Learning-to-Rank Methods

Learning-to-rank methods [21] frame prediction as an ordering problem rather than a regression or classification task, which is a natural fit for any setting where the goal is to identify the relative ordering of items in a cross-section rather than predict their absolute values. Listwise approaches, and ListMLE in particular [22], directly optimise the likelihood of the ground-truth permutation and have been shown to be effective for cross-sectional ranking tasks [2]. In our training objective we combine ListMLE with an MSE term to stabilise early training, following the common practice of mixing ranking and regression losses in sequential prediction tasks [2].

3 Data and Feature Engineering

3.1 Data Collection

To evaluate the proposed conditioning framework, we construct a large-scale longitudinal panel dataset using China A-share equities as a testbed. This market provides a natural evaluation setting for the heterogeneous-frequency fusion problem studied here: it offers two modalities that differ substantially in temporal frequency—daily price and microstructure data available at high frequency, and quarterly accounting fundamentals disclosed with a lag—over a sufficiently long horizon to cover multiple market regimes. The dataset spans January 2007 to December 2025 ($|\mathcal{T}| = 19$ years), sourced from the BaoStock financial data API (http://www.baostock.com).

The dataset comprises two raw modalities: (i) **daily market data**—open, high, low, close, volume, monetary turnover, daily return, and turnover rate for each stock—and (ii) **quarterly accounting fundamentals**—profitability ratios, earnings, and share structure metrics reported each fiscal quarter. From these two modalities, we engineer a 12-dimensional daily feature sequence and an 8-dimensional fundamental conditioning vector per stock, described in detail in Section 3.2.

The stock universe $\mathcal{S}_t$ is rebalanced annually at each year-end, excluding financially distressed and suspended securities to ensure that the cross-section reflects a realistic opportunity set. The resulting panel $\mathcal{D} = \{(s, t) \mid s \in \mathcal{S}_t,\ t \in \mathcal{T}\}$ is unbalanced, as $\mathcal{S}_t$ evolves over time due to new listings, delistings, and changes

in stock status. The cross-section grows over the sample period, expanding from approximately 1000 eligible stocks in the early years to over 3500 by the end of the sample, reflecting the substantial growth in A-share listings over this period.

3.2 Feature Engineering

Raw market data are transformed into a structured sequence dataset through a five-stage pipeline: feature engineering, fundamental alignment, monthly snapshot construction, sequence tensor assembly, and temporal train/validation/test splitting.

For each stock s and each trading day d, we compute a set of price-derived technical features from the raw Open-High-Low-Close-Volume (OHLCV) series. Let $c_{s,d}$ denote the forward-adjusted closing price and $r_{s,d} = c_{s,d}/c_{s,d-1} - 1$ the daily simple return. Let $r_{s,d}^{(k)}$, $\sigma_{s,d}^{(w)}$, $\mu_{s,d}^{(p/q)}$, $\rho_{s,d}^{v}$, $\bar{\tau}_{s,d}^{(20)}$, $\delta_{s,d}^{hl}$, $\mathrm{RSI}_{s,d}^{(14)}$, and $\tilde{r}_{s,d}$ denote the feature components defined in the equations below. The full 12-dimensional daily feature vector is:

$$\phi_{s,d} = \left(r_{s,d}^{(1)},\ r_{s,d}^{(5)},\ r_{s,d}^{(20)},\ \sigma_{s,d}^{(20)},\ \sigma_{s,d}^{(60)},\ \mu_{s,d}^{(5/20)},\ \mu_{s,d}^{(20/60)},\ \rho_{s,d}^{v},\ \bar{\tau}_{s,d}^{(20)},\ \delta_{s,d}^{hl},\ \mathrm{RSI}_{s,d}^{(14)},\ \tilde{r}_{s,d} \right), \tag{1}$$

where the components are defined as follows. Multi-horizon return features capture short- and medium-term price momentum [9], where k denotes the lookback horizon in trading days:

$$r_{s,d}^{(k)} = \frac{c_{s,d}}{c_{s,d-k}} - 1, \quad k \in \{1, 5, 20\}. \tag{2}$$

The rolling volatility $\sigma_{s,d}^{(w)}$ measures return dispersion over two horizons, where w denotes the window length in trading days, i is the lag index, and $\bar{r}_{s,d}^{(w)} = \frac{1}{w} \sum_{i=0}^{w-1} r_{s,d-i}$ is the rolling mean return over the same window:

$$\sigma_{s,d}^{(w)} = \sqrt{\frac{1}{w} \sum_{i=0}^{w-1} \left(r_{s,d-i} - \bar{r}_{s,d}^{(w)} \right)^2}, \quad w \in \{20, 60\}. \tag{3}$$

The moving-average momentum ratio $\mu_{s,d}^{(p/q)}$ encodes trend strength relative to different lookback scales, where p and q denote the short and long window lengths in trading days respectively, and $\frac{1}{p} \sum_{i=0}^{p-1} c_{s,d-i}$ is the p-day simple moving average of the closing price:

$$\mu_{s,d}^{(p/q)} = \frac{\frac{1}{p} \sum_{i=0}^{p-1} c_{s,d-i}}{\frac{1}{q} \sum_{i=0}^{q-1} c_{s,d-i}} - 1, \quad (p, q) \in \{(5, 20),\ (20, 60)\}. \tag{4}$$

The volume ratio $\rho_{s,d}^{v}$ and 20-day turnover moving average $\bar{\tau}_{s,d}^{(20)}$ capture liquidity and trading activity relative to recent history, where $v_{s,d}$ is the trading volume, $\tau_{s,d}$ is the daily turnover rate, and $\epsilon > 0$ is a small constant for numerical stability:

$$\rho_{s,d}^{v} = \frac{v_{s,d}}{\frac{1}{20} \sum_{i=1}^{20} v_{s,d-i} + \epsilon}, \tag{5}$$

$$\bar{\tau}_{s,d}^{(20)} = \frac{1}{20} \sum_{i=0}^{19} \tau_{s,d-i}. \tag{6}$$

The normalised high-low range $\delta_{s,d}^{hl}$ encodes intraday price dispersion, where $h_{s,d}$ and $l_{s,d}$ denote the daily high and low prices respectively:

$$\delta_{s,d}^{hl} = \frac{h_{s,d} - l_{s,d}}{c_{s,d} + \varepsilon}. \tag{7}$$

The 14-day Relative Strength Index $\mathrm{RSI}_{s,d}^{(14)}$ [23] is computed as follows, where $r_{s,d}^{+} = \max(r_{s,d}, 0)$ and $r_{s,d}^{-} = \max(-r_{s,d}, 0)$ denote the positive and negative components of the daily return respectively:

$$\mathrm{RSI}_{s,d}^{(14)} = 100 - \frac{100}{1 + \dfrac{\frac{1}{14}\sum_{i=0}^{13} r_{s,d-i}^{+}}{\frac{1}{14}\sum_{i=0}^{13} r_{s,d-i}^{-} + \varepsilon}}. \tag{8}$$

Finally, $\tilde{r}_{s,d}$ is the percentage change of the closing price as reported directly by the exchange. In all formulae, $\varepsilon = 10^{-8}$ is a small constant added for numerical stability.

To ensure year-boundary rolling statistics are computed over a full lookback window, the daily data for year t are prepended with a 30-trading-day buffer from year $t-1$ prior to feature computation; the buffer rows are discarded before saving.

3.2.1 Look-Ahead-Free Fundamental Alignment

Let $\mathbf{x}_{s,q}^{\mathrm{fund}} \in \mathbb{R}^{F_f}$ denote the fundamental feature vector of stock s reported in fiscal quarter q, and let $t_{s,q}^{\mathrm{pub}}$ denote the corresponding public announcement date. For a given rebalancing date d, the active fundamental conditioning vector $\mathbf{x}_{s,d}^{\mathrm{fund}}$ is taken from the most recently announced quarter q^* whose announcement date does not exceed d, ensuring that no future disclosure is visible to the model:

$$\mathbf{x}_{s,d}^{\mathrm{fund}} = \mathbf{x}_{s,\,q^*}^{\mathrm{fund}}, \quad q^* = \arg\max_{q} \left\{ t_{s,q}^{\mathrm{pub}} \le d \right\}. \tag{9}$$

In practice, $t_{s,q}^{\mathrm{pub}}$ is the actual filing date recorded in the BaoStock database, which reflects the regulatory disclosure deadline under CSRC rules: annual reports must be filed within four months of fiscal year-end, and interim reports within two months of the half-year end. The as-of join in Eq. (9) therefore incorporates the full regulatory reporting lag rather than assuming instantaneous disclosure.

This construction strictly prevents look-ahead bias, a critical requirement in ML-based asset pricing [2]: the model is never exposed to financial results before their public disclosure. Fundamental data are available from 2009 onward; stocks with no disclosed fundamental record prior to a given rebalancing date carry NaN values in the fundamental vector, which are subsequently imputed with zero after normalisation (see Section 3.2.3).

3.2.2 Monthly Snapshot and Target Construction

Rather than training on every trading day, we construct a monthly panel by selecting, for each stock s and calendar month m, the observation corresponding to the last trading day of m. This rebalancing-frequency design reduces temporal redundancy in the training data while matching the typical holding period of institutional quantitative strategies. Let $\mathcal{S}_d$ denote the eligible cross-section of stocks at rebalancing date d.

The raw 20-trading-day forward return $y_{s,d}$ is defined as:

$$y_{s,d} = \frac{c_{s,\,d+20}}{c_{s,d}} - 1, \tag{10}$$

where $c_{s,d+20}$ is the closing price 20 trading days ahead. This quantity is used solely for evaluation (IC computation) and to define the ground-truth cross-sectional ordering; it is not used directly as a training target due to its sensitivity to absolute market level.

Two derived training targets are constructed from $y_{s,d}$. The first is the *continuous cross-sectional rank* $\tilde{y}_{s,d} \in [0,1]$:

$$\tilde{y}_{s,d} = \frac{\mathrm{rank}\big(y_{s,d} \mid \mathcal{S}_d\big) - 1}{|\mathcal{S}_d| - 1}, \tag{11}$$

which serves as the regression target for the pointwise loss component. The second is the *quintile label* $z_{s,d} \in \{1,2,3,4,5\}$, obtained by applying cross-sectional equal-frequency binning to $y_{s,d}$ at each rebalancing date, which defines the ground-truth permutation π^* for the listwise ranking loss component. The model is trained jointly against both targets via the composite loss defined in Section 4.2.1. Both targets are computed within the eligible cross-section $\mathcal{S}_d$, making them invariant to absolute market direction.

3.2.3 Feature Normalisation

Price and Technical Features

Let $\phi_{s,d}^{(j)}$ denote the j-th scalar component of the feature vector $\phi_{s,d}$, where $j \in \{1,\ldots,12\}$ indexes the features defined in Eq. (1), and let $s' \in \mathcal{S}_d$ denote a generic stock in the eligible cross-section on day d. Each feature $\phi_{s,d}^{(j)}$ is independently rank-normalised across the cross-section:

$$\hat{\phi}_{s,d}^{(j)} = \frac{\mathrm{rank}\big(\phi_{s,d}^{(j)} \mid \{\phi_{s',d}^{(j)}\}_{s' \in \mathcal{S}_d}\big) - 1}{|\mathcal{S}_d| - 1}, \tag{12}$$

mapping all features to the interval $[0,1]$ while preserving relative ordinal relationships and providing automatic robustness to outliers and scale differences [10].

Fundamental Features

Fundamental features require additional treatment owing to their heavier-tailed distributions and sign-asymmetric magnitudes. A three-step normalisation is applied per rebalancing date d:

1. **Winsorisation.** Each feature is clipped to its 1st–99th percentile bounds within the cross-section $\mathcal{S}_d$ [2], suppressing the influence of extreme accounting values.
2. **Signed log-scaling.** Applied to the net profit feature only, owing to its heavy-tailed and sign-asymmetric distribution. Let $\Pi_{s,d}$ denote the raw net profit of stock s at rebalancing date d, and let $\mathrm{sgn}(\cdot)$ denote the sign function. The transformed value $\tilde{\Pi}_{s,d}$ is defined as:

$$\tilde{\Pi}_{s,d} = \mathrm{sgn}(\Pi_{s,d}) \cdot \log\big(1 + |\Pi_{s,d}|\big). \tag{13}$$

3. **Cross-sectional rank normalisation.** Eq. (12) is applied to each fundamental feature to yield a uniform $[0,1]$ distribution per feature per rebalancing date.

Missing values that remain after alignment (Eq. (9)) are imputed with 0.0 after normalisation, corresponding to the median of the normalised distribution and therefore encoding absence of information without introducing directional bias.

3.2.4 Sequence Tensor Construction

For each (stock, rebalancing-date) pair (s, d) that passes all filters, we extract a fixed-length historical window of $L = 60$ trading days ending on d, where $F_p = 12$ is the price feature dimension and $F_f = 8$ is the fundamental feature dimension. The resulting input tensors are:

$$\mathbf{X}_{s,d}^{\text{price}} \in \mathbb{R}^{L \times F_p}, \quad L = 60,\ F_p = 12, \tag{14}$$

$$\mathbf{x}_{s,d}^{\text{fund}} \in \mathbb{R}^{F_f}, \quad F_f = 8, \tag{15}$$

where $\mathbf{X}_{s,d}^{\text{price}}$ is the normalised price/technical feature matrix (Eqs. (1)–(12)) and $\mathbf{x}_{s,d}^{\text{fund}}$ is the normalised fundamental conditioning vector (Section 3.2.3). Samples with fewer than L historical trading days available prior to d are discarded. The FiLM conditioning vector $\mathbf{x}_{s,d}^{\text{fund}}$ is used to modulate the temporal encoder at each layer [3], as detailed in Section 4.1.

4 Methodology

4.1 Model Architecture

Figs. 1–3 provide an overview of all six deep model architectures evaluated in this work. Each encoder family—recurrent (LSTM), convolutional (TCN), and attention-based (iTransformer)—is instantiated in two variants: a *baseline* that incorporates the fundamental conditioning vector $\mathbf{x}^{\text{fund}} \in \mathbb{R}^{F_f}$ via late-fusion concatenation before the prediction head, and a *FiLM-conditioned* variant (ours) that instead injects $\mathbf{x}^{\text{fund}}$ into intermediate representations of the encoder via learned affine modulation. All six models share an identical prediction head architecture and hidden dimension $D_h = 64$, ensuring that observed performance differences are attributable to the fusion strategy rather than model capacity. The FiLM application point differs by encoder family: for the LSTM it modulates the final hidden state (Fig. 1b), for the TCN it modulates the full temporal feature map prior to pooling (Fig. 2d), and for the iTransformer it is applied inside each attention block after the self-attention sublayer and before the feed-forward sublayer (Fig. 3f).

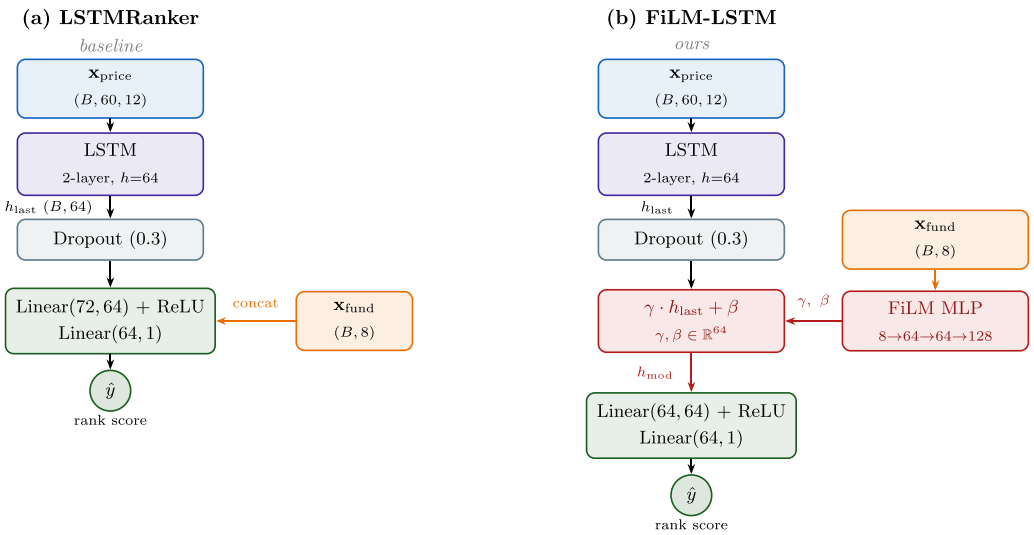

Figure 1: LSTM architecture variants. (**a**) LSTMRanker baseline—fundamentals concatenated to hidden state. (**b**) FiLM-LSTM—fundamentals modulate h_{last} via affine transformation before the prediction head.

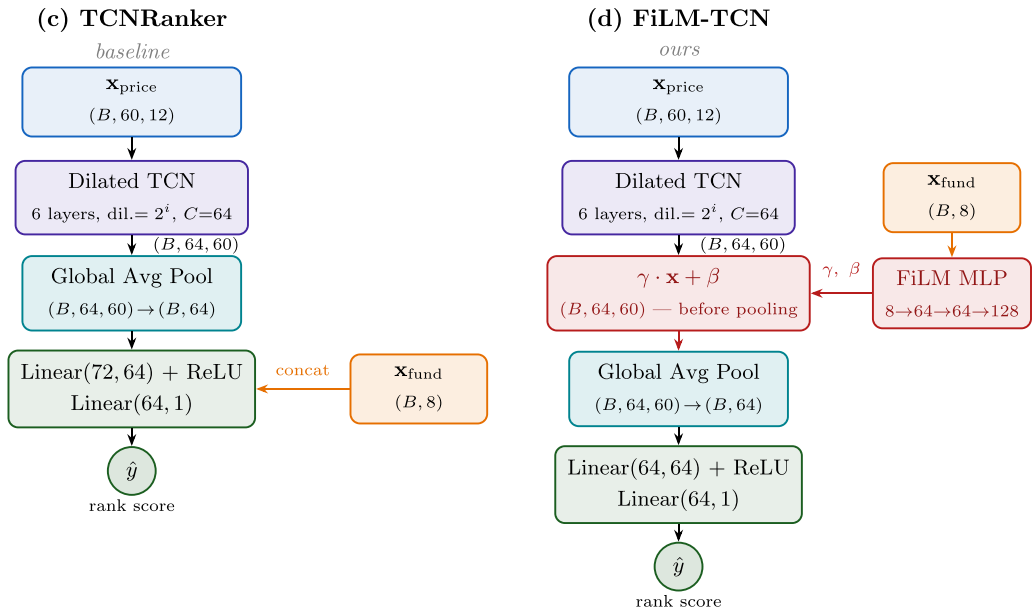

Figure 2: TCN architecture variants. (**c**) TCNRanker baseline—fundamentals concatenated after global average pooling. (**d**) FiLM-TCN—fundamentals modulate the full temporal feature map prior to pooling.

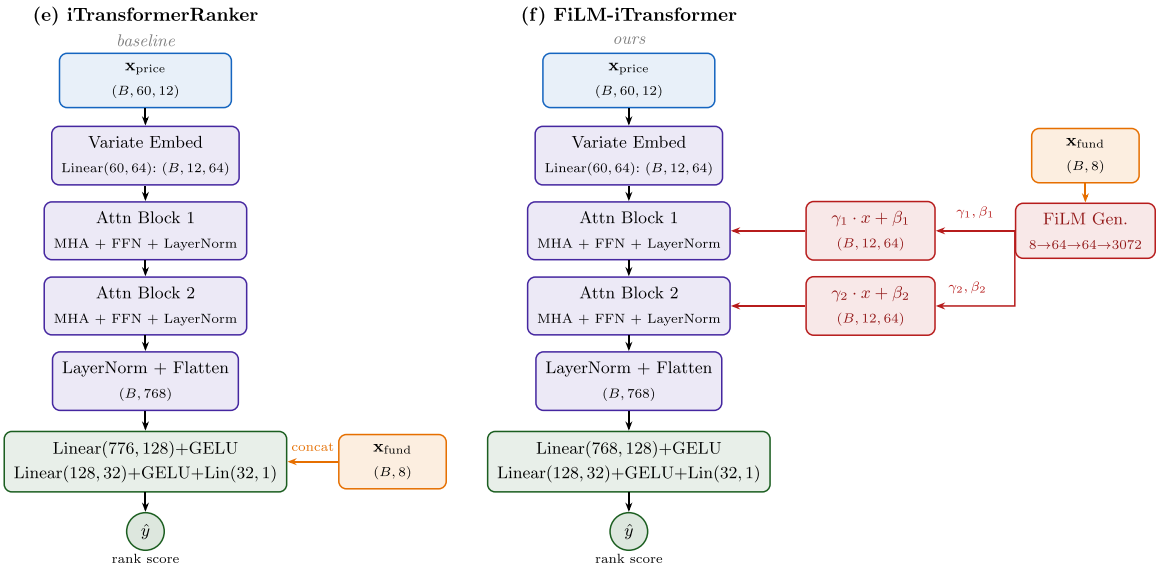

Figure 3: iTransformer architecture variants. (**e**) iTransformerRanker baseline—fundamentals concatenated to the flattened representation before the prediction head. (**f**) FiLM-iTransformer—a single FiLM generator produces layer-specific parameters (γ_l, β_l) applied inside each attention block after self-attention and before the feed-forward sublayer. $\mathbf{x}^{\text{fund}}$ is not concatenated to the prediction head.

4.1.1 Problem Formulation

Given the sequence tensor $\mathbf{X}_{s,d}^{\text{price}} \in \mathbb{R}^{L \times F_p}$ and the fundamental conditioning vector $\mathbf{x}_{s,d}^{\text{fund}} \in \mathbb{R}^{F_f}$ constructed in Section 3.2.4, the task is to learn a scoring function f_θ, where θ denotes the learnable model parameters:

$$f_\theta : \mathbb{R}^{L \times F_p} \times \mathbb{R}^{F_f} \longrightarrow \mathbb{R}, \tag{16}$$

such that, within each cross-section $\mathcal{S}_d$, the predicted score $\hat{y}_{s,d} = f_\theta(\mathbf{X}_{s,d}^{\text{price}}, \mathbf{x}_{s,d}^{\text{fund}})$ for stock s at rebalancing date d is monotonically consistent with the ground-truth cross-sectional rank $\tilde{y}_{s,d}$ (Eq. (11)). This is a *cross-sectional ranking* problem: the absolute level of $\hat{y}_{s,d}$ is irrelevant; only the relative ordering of stocks within a rebalancing date matters.

4.1.2 FiLM Conditioning Mechanism

Feature-wise Linear Modulation (FiLM) [3] is a general-purpose conditioning mechanism that applies a learned affine transformation to an intermediate representation as a function of an external conditioning signal. Let D_h denote the hidden dimension of the encoder, $\mathbf{h} \in \mathbb{R}^{D_h}$ an intermediate encoder representation, and $\mathbf{z} \in \mathbb{R}^{F_f}$ the conditioning input. The notation $\text{FiLM}(\mathbf{h}; \mathbf{z})$ uses a semicolon to separate the modulated representation $\mathbf{h}$ from the conditioning input $\mathbf{z}$. FiLM computes:

$$\text{FiLM}(\mathbf{h}; \mathbf{z}) = \boldsymbol{\gamma}(\mathbf{z}) \odot \mathbf{h} + \boldsymbol{\beta}(\mathbf{z}), \tag{17}$$

where $\odot$ denotes element-wise multiplication, and $\boldsymbol{\gamma} : \mathbb{R}^{F_f} \to \mathbb{R}^{D_h}$ and $\boldsymbol{\beta} : \mathbb{R}^{F_f} \to \mathbb{R}^{D_h}$ are learned feature-wise scaling and shifting functions respectively. Let $\text{ReLU}(x) = \max(0, x)$ denote the rectified linear unit activation applied element-wise, and let $[\cdot\,;\,\cdot]$ denote vertical vector concatenation. Both $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are implemented as a shared three-layer MLP with weight matrices $\mathbf{W}_1 \in \mathbb{R}^{D_h \times F_f}$, $\mathbf{W}_2 \in \mathbb{R}^{D_h \times D_h}$, and $\mathbf{W}_3 \in \mathbb{R}^{2D_h \times D_h}$, where the output dimension $2D_h$ accounts for the concatenated $[\boldsymbol{\gamma}; \boldsymbol{\beta}]$ vector:

$$\big[\boldsymbol{\gamma}(\mathbf{z}); \boldsymbol{\beta}(\mathbf{z})\big] = \mathbf{W}_3\, \text{ReLU}(\mathbf{W}_2\, \text{ReLU}(\mathbf{W}_1 \mathbf{z})) \in \mathbb{R}^{2D_h}. \tag{18}$$

The scaling output $\boldsymbol{\gamma}(\mathbf{z})$ is further passed through a scaled sigmoid activation $\sigma(\cdot)$ to constrain the gain to $[0, 2]$, preventing unstable amplification:

$$\boldsymbol{\gamma}(\mathbf{z}) = 2\,\sigma\Big(\mathbf{W}_3^{(\gamma)}\, \text{ReLU}(\mathbf{W}_2\, \text{ReLU}(\mathbf{W}_1 \mathbf{z}))\Big), \tag{19}$$

where $\mathbf{W}_3^{(\gamma)}$ denotes the $\boldsymbol{\gamma}$ partition of $\mathbf{W}_3$, and $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function. This parameterisation allows FiLM to selectively amplify, suppress, or shift each dimension of $\mathbf{h}$ conditioned on the fundamental context, without requiring the fundamental signal to be concatenated into the temporal processing stream.

In our setting, $\mathbf{z} = \mathbf{x}_{s,d}^{\text{fund}}$ is the F_f-dimensional normalised fundamental vector, with $F_f = 8$. Our instantiation departs from the original formulation of Perez et al. [3] in three specific ways motivated by the heterogeneous-frequency fusion setting. First, the original FiLM generator is a GRU over word token embeddings; ours is a three-layer MLP over eight normalised tabular accounting ratios, since fundamentals are tabular scalars rather than sequential tokens. Second, the original FiLM conditions inside every convolutional layer at the spatial level; ours conditions on complete encoder representations, because the conditioning signal is quarterly while the price sequence is daily—applying conditioning before a complete temporal summary is available would be meaningless for a static low-frequency signal. Third, the original FiLM's conditioning and primary signals share the same temporal granularity; ours deliberately bridges two different frequencies, which is the core architectural problem this paper addresses. The sigmoid constraint on $\boldsymbol{\gamma}$ to $[0, 2]$ is an additional design choice not present in the original formulation, introduced to prevent unstable amplification in the presence of financial data noise. We instantiate FiLM at different points in each encoder family, as described below.

4.1.3 Encoder Architectures

LSTM and FiLM-LSTM

The recurrent baseline (**LSTM Ranker**) encodes the price sequence $\mathbf{X}_{s,d}^{\text{price}} \in \mathbb{R}^{L \times F_p}$ with a two-layer LSTM [4], processing the sequence step by step and extracting the final hidden state $\mathbf{h}_L \in \mathbb{R}^{D_h}$, where L is the sequence length and D_h is the hidden dimension defined in Section 4.1.2. Let $\text{Head}(\cdot)$ denote the prediction head—a two-layer MLP with a scalar output that maps the encoder representation to the predicted score. Fundamentals are incorporated by late fusion—concatenation with the hidden state before the prediction head:

$$\hat{y}_{s,d} = \text{Head}\left([\mathbf{h}_L;\ \mathbf{x}_{s,d}^{\text{fund}}]\right). \tag{20}$$

The proposed **FiLM-LSTM** instead modulates the final hidden state via Eq. (17) before the prediction head:

$$\hat{y}_{s,d} = \text{Head}\left(\text{FiLM}(\mathbf{h}_L;\ \mathbf{x}_{s,d}^{\text{fund}})\right). \tag{21}$$

This allows the fundamental context to rescale and shift the learned temporal summary before any scoring decision is made. FiLM is applied at $\mathbf{h}_L$ rather than at each intermediate step $t < L$ for two reasons. First, quarterly accounting fundamentals are static within the 60-day window—injecting the same conditioning vector at every recurrent step would provide no additional temporal information. Second, intermediate hidden states $\mathbf{h}_t$ for $t < L$ encode only a partial sequential context; $\mathbf{h}_L$ is the earliest point at which the complete 60-day sequence has been processed, making it the natural insertion point for a low-frequency static signal [3,19]. The structural consequence of this placement—that FiLM-LSTM conditions a collapsed representation rather than a temporally-rich feature map—is discussed in Section 6.

TCN and FiLM-TCN

The convolutional baseline (**TCN Ranker**) uses a Temporal Convolutional Network [5] consisting of $N_c = 6$ causal dilated residual blocks, where N_c denotes the number of blocks. Let $i \in \{1, \ldots, N_c\}$ denote the block index (distinct from the lag index used in Section 3.2), $k_c = 3$ the convolutional kernel size, and $\delta_i = 2^{i-1}$ the exponentially increasing dilation factor of block i. The resulting receptive field is $1 + 2(k_c - 1)\sum_{i=0}^{N_c-1} 2^i = 127$ time steps, sufficient to cover the full $L = 60$-day input window.

Let $C = D_h$ denote the number of feature channels, $\mathbf{F}^{(i)} \in \mathbb{R}^{C \times L}$ the feature map after block i, and $t \in \{1, \ldots, L\}$ the time step index. The input feature map is $\mathbf{F}^{(0)} = \mathbf{X}_{s,d}^{\text{price}\top} \in \mathbb{R}^{F_p \times L}$, where $(\cdot)^\top$ denotes the matrix transpose. Let $\text{Conv1d}_{\delta_i}^{\text{causal}}(\cdot)$ denote a causal dilated 1D convolution with dilation δ_i, which ensures no future time steps are visible to the model. Each block applies this convolution followed by ReLU activation, dropout, and a residual shortcut:

$$\mathbf{F}^{(i)} = \text{ReLU}\left(\text{Conv1d}_{\delta_i}^{\text{causal}}(\mathbf{F}^{(i-1)})\right) + \mathbf{F}^{(i-1)}, \quad i = 1, \ldots, N_c. \tag{22}$$

The notation $\mathbf{F}_{:,t}^{(N_c)} \in \mathbb{R}^C$ denotes the t-th column of the final feature map, i.e., the feature vector at time step t. The baseline performs global average pooling over the time dimension and concatenates with fundamentals before the head:

$$\hat{y}_{s,d} = \text{Head}\left(\left[\frac{1}{L}\sum_{t=1}^{L}\mathbf{F}_{:,t}^{(N_c)};\ \mathbf{x}_{s,d}^{\text{fund}}\right]\right). \tag{23}$$

The proposed **FiLM-TCN** applies FiLM conditioning to the full temporal feature map $\mathbf{F}^{(N_c)} \in \mathbb{R}^{C\times L}$ prior to pooling, producing a modulated map $\tilde{\mathbf{F}}^{(N_c)} \in \mathbb{R}^{C\times L}$:

$$\tilde{\mathbf{F}}^{(N_c)}_{:,t} = \boldsymbol{\gamma}\left(\mathbf{x}^{\text{fund}}_{s,d}\right) \odot \mathbf{F}^{(N_c)}_{:,t} + \boldsymbol{\beta}\left(\mathbf{x}^{\text{fund}}_{s,d}\right), \quad \forall\, t \in \{1,\ldots,L\}, \tag{24}$$

followed by global average pooling and the prediction head. Crucially, conditioning occurs *before* aggregation rather than after, enabling the fundamental context to modulate how each temporal pattern is weighted across the entire sequence—for instance, amplifying momentum patterns for high-ROE stocks while suppressing them for low-quality firms [3].

iTransformer and FiLM-iTransformer

The attention-based baseline (**iTransformer Ranker**) [6] inverts the standard Transformer architecture by treating each variate (feature dimension) as a token rather than each time step. Let D_{emb} denote the embedding dimension, $l \in \{1,\ldots,N_l\}$ the layer index, and $N_l = 2$ the number of inverted self-attention layers. Let $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{L\times D_{\text{emb}}}$ denote the learned embedding projection matrix. The L-length time series of each of the F_p features is first projected to a D_{emb}-dimensional embedding, yielding the initial embedding matrix $\mathbf{E}^{(0)} \in \mathbb{R}^{F_p\times D_{\text{emb}}}$:

$$\mathbf{E}^{(0)} = \mathbf{X}^{\text{price}\top}_{s,d}\,\mathbf{W}_{\text{emb}} \in \mathbb{R}^{F_p\times D_{\text{emb}}}, \tag{25}$$

where $(\cdot)^\top$ denotes the matrix transpose as before. Let $\text{LayerNorm}(\cdot)$ denote layer normalisation, $\text{MHA}(\mathbf{Q},\mathbf{K},\mathbf{V})$ multi-head attention with query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ matrices and $H = 4$ attention heads, and $\text{FFN}(\cdot)$ a two-layer feed-forward network with hidden dimension $4D_{\text{emb}}$. Let $\mathbf{E}^{(l)} \in \mathbb{R}^{F_p\times D_{\text{emb}}}$ denote the embedding matrix after layer l, and $\mathbf{E}'$ the intermediate embedding after the self-attention sublayer. Each of the N_l layers models inter-variate dependencies:

$$\mathbf{E}' = \text{LayerNorm}\left(\mathbf{E}^{(l-1)} + \text{MHA}(\mathbf{E}^{(l-1)},\mathbf{E}^{(l-1)},\mathbf{E}^{(l-1)})\right), \tag{26}$$

$$\mathbf{E}^{(l)} = \text{LayerNorm}(\mathbf{E}' + \text{FFN}(\mathbf{E}')). \tag{27}$$

The baseline concatenates $\text{flatten}(\mathbf{E}^{(N_l)})$—where $\text{flatten}(\cdot)$ reshapes a matrix into a vector—with $\mathbf{x}^{\text{fund}}_{s,d}$ before the prediction head. The proposed **FiLM-iTransformer** introduces a shared FiLM generator that produces layer-specific scaling and shifting parameters $\boldsymbol{\gamma}_l, \boldsymbol{\beta}_l \in \mathbb{R}^{F_p\times D_{\text{emb}}}$ for each layer l, via a layer-specific MLP $\text{MLP}_l(\cdot)$:

$$\left[\boldsymbol{\gamma}_l;\, \boldsymbol{\beta}_l\right] = \text{MLP}_l\left(\mathbf{x}^{\text{fund}}_{s,d}\right) \in \mathbb{R}^{2\times F_p\times D_{\text{emb}}}, \tag{28}$$

$$\mathbf{E}^{(l)} = \text{FiLM}\left(\mathbf{E}';\, \boldsymbol{\gamma}_l,\, \boldsymbol{\beta}_l\right). \tag{29}$$

This design allows the fundamental context to modulate the inter-variate attention representations at every layer, giving the model direct control over which cross-feature interactions are amplified or suppressed conditional on the firm's financial profile. The prediction head uses only the modulated representation—$\mathbf{x}^{\text{fund}}_{s,d}$ is not concatenated—as the conditioning information has already been fully absorbed into $\mathbf{E}^{(N_l)}$:

$$\hat{y}_{s,d} = \text{Head}\left(\text{flatten}\left(\mathbf{E}^{(N_l)}\right)\right). \tag{30}$$

XGBoost Ranker

As a strong tabular baseline, we train an XGBoost gradient-boosted tree model [7] on the flattened union of all price/technical and fundamental features (20 features total), without any temporal sequence structure.

This model serves as the primary indicator of whether deep temporal encoders provide incremental value over well-tuned classical approaches.

4.1.4 Model Configurations

All deep models share a common hidden dimension $D_h = 64$ and a common prediction head architecture: a two-layer MLP with ReLU activation, dropout with rate $p_{\text{drop}} = 0.3$, and a scalar output. The three encoder families evaluated are: the two-layer LSTM [4], the six-block dilated TCN [5], and the two-layer iTransformer [6], each instantiated in a baseline and a FiLM-conditioned variant. Table 1 summarises the key architectural hyperparameters.

Table 1: Architectural hyperparameters and parameter counts. FiLM overhead is the additional parameters introduced by the FiLM generator MLP relative to the baseline.

Model	Encoder	Depth	Dim (D_h)	Parameters
XGBoost Ranker	GBT[a]	500 trees	—	—
LSTM Ranker	LSTM	2 layers	64	57,985
FiLM-LSTM	LSTM + FiLM	2 layers	64	70,529
TCN Ranker	Dilated conv[b]	6 blocks	64	69,697
FiLM-TCN	TCN + FiLM	6 blocks	64	82,241
iTransformer Ranker	Inv. attn.[c]	2 layers	64	207,617
FiLM-iTransformer	Inv. + FiLM	2 layers	64	411,009

Note: [a] GBT: Gradient-Boosted Trees [7]. [b] Dilated conv: causal dilated 1D convolution as used in the Temporal Convolutional Network [5]. [c] Inv. attn.: inverted self-attention operating across the variate dimension rather than the time dimension [6].

4.2 Training Objective

4.2.1 Loss Function

Let $\mathcal{L}$ denote the composite training objective, $\mathcal{L}_{\text{ListMLE}}$ the listwise ranking loss, $\mathcal{L}_{\text{MSE}}$ the pointwise regression loss, and $\alpha \in [0, 1]$ the weighting coefficient controlling their relative contribution. Model training optimises:

$$\mathcal{L} = \alpha\, \mathcal{L}_{\text{ListMLE}} + (1 - \alpha)\, \mathcal{L}_{\text{MSE}}, \quad \alpha = 0.1, \tag{31}$$

where the MSE term provides dense, stable gradients early in training and the ListMLE term directly optimises the ranking objective. The MSE loss is defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum_{s \in \mathcal{S}_d} \left(\hat{y}_{s,d} - \tilde{y}_{s,d} \right)^2, \tag{32}$$

where $\tilde{y}_{s,d}$ is the continuous cross-sectional rank defined in Eq. (11), serving as the regression target.

The **ListMLE** loss [22] maximises the log-likelihood of the ground-truth permutation of stocks ranked by their 20-day forward returns. Let $n = |\mathcal{S}_d|$ denote the number of stocks in the cross-section at rebalancing date d, $\hat{\mathbf{y}} = (\hat{y}_{1,d}, \ldots, \hat{y}_{n,d})$ the vector of predicted scores, and π^* the ground-truth permutation induced by ranking stocks by $y_{s,d}$ in descending order, where $\pi^*(r)$ denotes the index of the stock at rank r. Let r and r' denote summation indices over ranks (distinct from the lag and block indices used earlier), and let log and exp denote the natural logarithm and natural exponential respectively:

$$\mathcal{L}_{\text{ListMLE}} = -\frac{1}{n}\sum_{r=1}^{n}\left[\hat{y}_{\pi^*(r)} - \log\sum_{r'=r}^{n}\exp\left(\hat{y}_{\pi^*(r')}\right)\right], \tag{33}$$

where $\hat{y}_{\pi^*(r)}$ denotes the predicted score of the stock at ground-truth rank r. The log-sum-exp term $\log\sum_{r'=r}^{n}\exp(\hat{y}_{\pi^*(r')})$ is computed using the numerically stable log-sum-exp identity $\log\sum_i \exp(a_i) = c + \log\sum_i \exp(a_i - c)$, where $c = \max_i a_i$, to avoid floating-point overflow. ListMLE is only well-defined within a single cross-section; accordingly, each training batch is constructed to contain stocks from exactly one rebalancing date (Section 4.2.3).

4.2.2 Optimisation and Regularisation

All deep models are trained with the AdamW optimiser [24], a variant of Adam with decoupled weight decay, using weight decay coefficient $\lambda_w = 10^{-3}$ and gradient clipping: the ℓ_2 norm of the gradient $\|\nabla\mathcal{L}\|_2$ is clipped to a maximum of 1.0 at each update step to stabilise training.

Two learning rate schedules are used depending on encoder type. Let η_τ denote the learning rate at training step τ (distinct from the time step index t used in Section 4.1.3), $\eta_{\min}$ and $\eta_{\max}$ the minimum and maximum learning rates respectively, and $T_{\max}$ the total number of training steps. LSTM and TCN models use cosine annealing [24]:

$$\eta_\tau = \eta_{\min} + \tfrac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\frac{\pi_c\,\tau}{T_{\max}}\right), \tag{34}$$

where $\pi_c \approx 3.14159$ is the mathematical constant (distinct from the permutation π^* defined in Section 4.2.1), with $\eta_{\max} = 10^{-3}$ (LSTM) or 3×10^{-4} (TCN), and $\eta_{\min} = 0.01\,\eta_{\max}$.

Transformer-based models use a one-cycle learning rate schedule [25] consisting of three phases: a linear warmup over the first 10% of training steps from $\eta_{\max}/10$ to $\eta_{\max}$, followed by cosine annealing from $\eta_{\max}$ to $\eta_{\max}/100$ over the remaining steps, with $\eta_{\max} = 3 \times 10^{-4}$.

Early stopping monitors the validation Information Coefficient (IC), defined formally in Section 4.2.4, with a patience of 10 epochs for LSTM and TCN models and 15 epochs for Transformer variants.

4.2.3 Cross-Sectional Batch Construction

A training batch is defined as the set of all stock samples from a single rebalancing date, and one epoch is a full pass over all rebalancing dates in the training split. A custom grouped sampler, which we refer to as **DateGroupedSampler**, constructs batches such that all samples within a batch originate from the same rebalancing date. This is a hard requirement for the ListMLE loss (Eq. (33)): ranking stocks from different calendar months in the same gradient update is semantically meaningless, as their forward returns are not cross-sectionally comparable and no valid permutation π^* exists across months.

Batches with fewer than 10 stocks are discarded to ensure a minimum cross-sectional diversity for the ListMLE loss to produce a meaningful ranking gradient. The training split covers 96 calendar months (January 2010 to December 2017, Table 2), yielding mini-batches of 128 stocks drawn exclusively from the same rebalancing date, resulting in approximately 1500 gradient updates per epoch across the training split.

Table 2: Temporal train/validation/test split. Train covers bull/bear cycles and the circuit-breaker crash (2015–16); validation covers the US–China trade war and domestic deleveraging; test covers the COVID crash, recovery, and post-COVID normalisation.

Split	Period	Months	Sequences	Avg. Stocks/Month
Train	2010-01–2017-12	96	202,043	2105
Validation	2018-01–2019-12	24	76,132	3172
Test	2020-01–2024-12	60†	217,850	3631

Note: †The test window spans 60 calendar months (2020-01 to 2024-12). Evaluation metrics are computed over $T = 56$ rebalancing dates with valid 20-trading-day forward returns and sufficient cross-sectional coverage; the remaining four months did not meet these criteria and are excluded.

4.2.4 Evaluation Protocol

Model selection uses the **Information Coefficient** (IC), defined as the Spearman rank correlation between the vector of predicted scores $\hat{\mathbf{y}}_d = (\hat{y}_{1,d}, \ldots, \hat{y}_{n,d})$ and the vector of continuous cross-sectional ranks $\tilde{\mathbf{y}}_d = (\tilde{y}_{1,d}, \ldots, \tilde{y}_{n,d})$ (Eq. (11)) across all stocks in $\mathcal{S}_d$. Let $\rho_s(\cdot, \cdot)$ denote the Spearman rank correlation coefficient. The IC at rebalancing date d is:

$$\mathrm{IC}_d = \rho_s(\hat{\mathbf{y}}_d,\ \tilde{\mathbf{y}}_d)\,, \tag{35}$$

and the mean IC across all rebalancing dates $d \in \mathcal{D}_{\mathrm{val}}$ in the validation set is:

$$\overline{\mathrm{IC}} = \frac{1}{|\mathcal{D}_{\mathrm{val}}|} \sum_{d \in \mathcal{D}_{\mathrm{val}}} \mathrm{IC}_d. \tag{36}$$

The model checkpoint achieving the highest $\overline{\mathrm{IC}}$ on the validation set is selected for final evaluation.

The **IC Information Ratio** (ICIR) measures the consistency of the predictive signal and is defined as the ratio of the mean IC to its standard deviation σ_{IC} across rebalancing dates:

$$\mathrm{ICIR} = \frac{\overline{\mathrm{IC}}}{\sigma_{\mathrm{IC}}}, \quad \sigma_{\mathrm{IC}} = \sqrt{\frac{1}{|\mathcal{D}_{\mathrm{val}}| - 1} \sum_{d \in \mathcal{D}_{\mathrm{val}}} (\mathrm{IC}_d - \overline{\mathrm{IC}})^2}. \tag{37}$$

ICIR is logged per epoch but is not used for model selection: with only 24 validation months (Table 2), the standard error on σ_{IC} is too large for reliable model selection.

To quantify variance due to random weight initialisation, all models are trained under three independent random seeds $\mathcal{R} = \{42, 123, 777\}$; test results are reported as mean ± standard deviation across seeds. Mixed-precision training, which reduces memory footprint by storing activations in 16-bit floating point during the forward pass, is enabled throughout via `torch.amp.autocast` in PyTorch.

4.3 Experimental Setup

4.3.1 Dataset Splits

The dataset is partitioned into three non-overlapping temporal splits following a strict walk-forward protocol with no shuffling across time boundaries (Table 2).

The test set is intentionally held out until final evaluation to prevent indirect data snooping through hyperparameter tuning on test-period market regimes. The test split contains more total sequences than the training split despite covering fewer calendar months, reflecting the growth of the A-share cross-section over

the sample period; the number of unique stocks appearing in each split is 3185 (train), 3420 (validation), and 4423 (test), with the increase driven by new listings over the 15-year window.

4.3.2 Baseline Models

We compare the three proposed FiLM-conditioned models against four baselines spanning the classical-to-deep-learning spectrum:

- **XGBoost Ranker** [7]: a gradient-boosted tree model trained on the full 20-dimensional tabular feature vector (12 price/technical + 8 fundamental features), without any temporal sequence structure.
- **LSTM Ranker:** a two-layer LSTM [4] encoding the 60-day price sequence, with fundamental features incorporated via late-fusion concatenation to the final hidden state (Eq. (20)).
- **TCN Ranker:** a six-block dilated causal TCN [5] with fundamental features concatenated after global average pooling (Eq. (23)).
- **iTransformer Ranker:** a two-layer inverted Transformer with variate-level attention [6], with fundamental features concatenated to the flattened representation before the prediction head.

Each deep baseline shares identical hidden dimension ($D_h = 64$), dropout ($p_{\text{drop}} = 0.3$), and prediction head with its FiLM-conditioned counterpart, ensuring that performance differences are attributable solely to the conditioning mechanism rather than model capacity.

4.3.3 Evaluation Metrics

We evaluate all models on two complementary axes: predictive ranking quality and simulated portfolio performance.

Ranking Metrics

The **Information Coefficient** (IC) [1] measures the Pearson correlation between the predicted scores $\hat{y}_{s,d}$ and the realised 20-day forward returns $y_{s,d}$, computed per rebalancing date and averaged. Let $\rho_p(\cdot,\cdot)$ denote the Pearson correlation coefficient, $\mathcal{D}$ the set of all rebalancing dates in the evaluation period, and IC_d the per-date IC:

$$\text{IC}_d = \rho_p\left(\{\hat{y}_{s,d}\}_{s\in\mathcal{S}_d},\ \{y_{s,d}\}_{s\in\mathcal{S}_d}\right), \tag{38}$$

with the mean IC across all rebalancing dates defined as:

$$\overline{\text{IC}} = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} \text{IC}_d. \tag{39}$$

Note that this section uses Pearson correlation for IC, consistent with standard practice in asset pricing [1], while Section 4.2.4 uses Spearman correlation for model selection—the distinction is discussed further in Section 5.

The **Rank Information Coefficient** (RankIC) substitutes Spearman rank correlation $\rho_s(\cdot,\cdot)$ for $\rho_p(\cdot,\cdot)$ in Eq. (38), providing robustness to outlier returns:

$$\text{RankIC}_d = \rho_s\left(\{\hat{y}_{s,d}\}_{s\in\mathcal{S}_d},\ \{y_{s,d}\}_{s\in\mathcal{S}_d}\right), \tag{40}$$

with $\overline{\text{RankIC}}$ defined analogously to Eq. (39).

The corresponding **ICIR** and **RankICIR** are defined as the ratio of the mean IC (or RankIC) to its standard deviation σ_{IC} (or σ_{RankIC}) across rebalancing dates, where $\overline{\mathrm{IC}}$ and σ_{IC} are as defined in Eq. (37):

$$\mathrm{ICIR} = \frac{\overline{\mathrm{IC}}}{\sigma_{\mathrm{IC}}}, \qquad \mathrm{RankICIR} = \frac{\overline{\mathrm{RankIC}}}{\sigma_{\mathrm{RankIC}}}. \tag{41}$$

We additionally report **IC > 0**, the fraction of rebalancing dates on which $\mathrm{IC}_d > 0$, as a measure of signal hit rate:

$$\mathrm{IC}{>}0 = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \mathbf{1}[\mathrm{IC}_d > 0], \tag{42}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function.

We formally define **signal stability** as the standard deviation of RankIC_d across rebalancing dates in the evaluation period:

$$\sigma(\mathrm{RankIC}) = \sqrt{\frac{1}{|\mathcal{D}| - 1} \sum_{d \in \mathcal{D}} \left(\mathrm{RankIC}_d - \overline{\mathrm{RankIC}}\right)^2}. \tag{43}$$

A lower $\sigma(\mathrm{RankIC})$ indicates a more consistent month-to-month predictive signal. Note that $\sigma(\mathrm{RankIC})$ is the denominator of RankICIR—a model with lower $\sigma(\mathrm{RankIC})$ at the same mean RankIC will produce a higher RankICIR. We report $\sigma(\mathrm{RankIC})$ explicitly alongside RankICIR to separate the contribution of signal mean from signal variance to the overall risk-adjusted predictive quality.

Portfolio Metrics

To evaluate practical utility, we construct a monthly long-short portfolio by assigning each stock to one of five equal-frequency quintile buckets based on its predicted score $\hat{y}_{s,d}$, where quintile 1 (Q1) contains the lowest-scoring stocks and quintile 5 (Q5) the highest-scoring stocks. The portfolio takes a long position in Q5 and a short position in Q1.

Let $\bar{r}_d^{(q)}$ denote the equal-weighted mean 20-day forward return of quintile $q \in \{1, \ldots, 5\}$ at rebalancing date d, and let $c_{\mathrm{tc}} = 30$ bps denote the one-way transaction cost in basis points (1 bps = 0.01%), applied to both legs assuming 100% monthly portfolio turnover [2]. The 30 bps one-way cost assumption follows standard practice in Chinese A-share quantitative research and is applied conservatively: actual measured turnover across all nine deep models ranges from 54.7% to 70.4%, meaning the 100% assumption overstates costs and the reported net returns are lower bounds on practically achievable performance under realistic execution. The monthly long-short portfolio return r_d^{LS} is:

$$r_d^{LS} = \left(\bar{r}_d^{(5)} - c_{\mathrm{tc}}\right) - \left(\bar{r}_d^{(1)} + c_{\mathrm{tc}}\right). \tag{44}$$

Let $T = |\mathcal{D}_{\mathrm{test}}|$ denote the total number of rebalancing dates in the test period. The **annualised return** (Ann.Ret) is computed via geometric compounding, which accounts for the path-dependent effect of return volatility on terminal wealth:

$$\mathrm{Ann.Ret} = \left(\prod_{d=1}^{T} (1 + r_d^{LS})\right)^{12/T} - 1, \tag{45}$$

where the exponent $12/T$ converts from monthly to annual frequency, assuming 12 rebalancing periods per year.

Let $\bar{r}^{LS} = \frac{1}{T}\sum_{d=1}^{T} r_d^{LS}$ denote the mean monthly L/S return and $\sigma_{r^{LS}} = \sqrt{\frac{1}{T-1}\sum_{d=1}^{T}(r_d^{LS} - \bar{r}^{LS})^2}$ its sample standard deviation. The **annualised Sharpe ratio** [26], scaled by $\sqrt{12}$ to convert from monthly to annual frequency, is:

$$\text{Sharpe} = \frac{\bar{r}^{LS}}{\sigma_{r^{LS}}} \cdot \sqrt{12}. \tag{46}$$

Let $m \in \{1, \ldots, T\}$ denote a portfolio time index (distinct from the sequence time step t and training step τ used earlier), and let $C_m = \prod_{d=1}^{m}(1 + r_d^{LS})$ denote the cumulative L/S return up to period m. **Maximum drawdown** (MaxDD) [27] is the largest peak-to-trough decline in the cumulative return series:

$$\text{MaxDD} = \min_{m \in \{1,\ldots,T\}} \frac{C_m - \max_{m' \le m} C_{m'}}{\max_{m' \le m} C_{m'}}, \tag{47}$$

where m' is a look-back index over all prior periods.

4.3.4 Ensemble and Statistical Reporting

To account for variance due to random weight initialisation, all deep models are trained independently under three random seeds, collected in the set $\mathcal{R} = \{42, 123, 777\}$, where each seed determines the initial model weights. Let $\hat{y}_{s,d}^{(r)}$ denote the predicted score of stock s at rebalancing date d produced by the model trained under seed $r \in \mathcal{R}$, where each seed yields a separate saved model checkpoint—the model state at the epoch achieving the highest validation $\overline{\text{IC}}$ (Eq. (36)). At evaluation time, the ensemble predicted score $\hat{y}_{s,d}^{\text{ens}}$ is the average across the three seed checkpoints, using seed index r' to avoid conflict with the rank index r used in Eq. (33):

$$\hat{y}_{s,d}^{\text{ens}} = \frac{1}{|\mathcal{R}|} \sum_{r' \in \mathcal{R}} \hat{y}_{s,d}^{(r')}. \tag{48}$$

All reported test metrics are computed using $\hat{y}_{s,d}^{\text{ens}}$ in place of $\hat{y}_{s,d}$. Ensemble averaging has been shown to reduce variance in IC estimates for financial prediction tasks [2]. Per-seed validation IC_d values are additionally reported as mean ± standard deviation across $\mathcal{R}$ to characterise training stability.

4.3.5 Implementation Details

All experiments are implemented in PyTorch and executed on a single NVIDIA RTX PRO 6000 Blackwell Server Edition GPU. Mixed-precision training (`torch.amp.autocast`) is enabled throughout to reduce memory footprint. The full hyperparameter configuration is summarised in Table 3.

Table 3: Hyperparameter configuration shared across all deep models.

Hyperparameter	Value
Sequence length L	60 trading days
Price feature dimension F_p	12
Fundamental feature dimension F_f	8
Hidden/model dimension D_h	64

(Continued)

Table 3 (continued)

Hyperparameter	Value
Attention heads H	4
Transformer layers N_l	2
TCN layers N_c	6
TCN kernel size k_c	3
Dropout p_{drop}	0.3
Batch size	128 (date-grouped)
Weight decay λ_w	10^{-3}
Max epochs	60
Loss weighting α (ListMLE weight)	0.1
Learning Rate—LSTM	10^{-3}
Learning Rate—TCN	3×10^{-4}
Learning Rate—iTransformer	3×10^{-4}
Early stopping patience (LSTM/TCN)	10 epochs
Early stopping patience (iTransformer)	15 epochs
Random seeds $\mathcal{R}$	$\{42, 123, 777\}$

5 Results

5.1 Main Results

Tables 4 and 5 report the full test-set performance of all seven models over the 2020–2024 evaluation period. All deep model results are ensemble predictions averaged across three random seeds $\mathcal{R} = \{42, 123, 777\}$ as described in Section 4.3.4.

Table 4: Ranking metric performance on the test set (2020–2024, China A-shares). IC_σ denotes the standard deviation of per-seed validation IC, measuring training stability. Signal Stability is the standard deviation of per-rebalancing-date RankIC defined in Eq. (43); lower values indicate more consistent month-to-month signal. Best result per column is **bolded**.

Model	IC	ICIR	RankIC	RankICIR	IC > 0	IC_σ	Signal Stability
XGBoost	0.0890	**0.785**	0.1313	1.029	76.8%	—	—
LSTM	0.0864	0.676	0.1379	1.018	75.0%	0.0013	0.1355
FiLM-LSTM *(ours)*	0.0851	0.665	0.1380	1.029	71.4%	0.0021	0.1341
TCN	0.0881	0.665	0.1407	1.015	75.0%	0.0025	0.1387
FiLM-TCN *(ours)*	**0.0905**	0.730	**0.1415**	**1.099**	**80.4%**	0.0015	**0.1287**
iTransformer	0.0680	0.546	0.1102	0.782	64.3%	0.0044	0.1410
FiLM-iTransformer *(ours)*	0.0768	0.589	0.1288	0.864	71.4%	0.0017	0.1491

Table 5: Portfolio metric performance on the test set (2020–2024, China A-shares). Ann. Return is geometrically compounded net of 30 bps per leg transaction cost, assuming 100% monthly turnover. Actual Turnover is the mean monthly fraction of the long-short portfolio replaced at each rebalancing date. Best result per column is **bolded**. For MaxDD, the least negative value is best.

Model	Ann. Return	Sharpe	MaxDD	Actual Turnover
XGBoost	21.97%	1.522	−17.75%	N/A
LSTM	22.83%	1.463	**−13.81%**	59.9%
FiLM-LSTM *(ours)*	23.33%	1.498	−14.27%	59.9%
TCN	24.94%	1.537	−16.90%	54.7%
FiLM-TCN *(ours)*	**25.30%**	**1.633**	−15.48%	54.9%
iTransformer	14.71%	1.015	−14.51%	70.4%
FiLM-iTransformer *(ours)*	19.36%	1.228	−19.08%	63.3%

FiLM-TCN Achieves the Best Overall Performance

FiLM-TCN attains the highest scores among the six deep encoder models in Table 4 across every primary metric: IC of 0.0905, RankIC of 0.1415, ICIR of 0.730, RankICIR of 1.099, annualised Sharpe ratio of 1.633, and an IC hit rate of 80.4%. Among all seven models including the tabular baseline, FiLM-TCN leads on IC, RankIC, RankICIR, Sharpe, and IC > 0; XGBoost achieves a higher ICIR (0.785 vs. 0.730), a point discussed further in Section 6.1. Relative to its baseline TCN, FiLM conditioning yields consistent improvements across all metrics—a +2.7% relative gain in IC, +9.7% in ICIR, and +6.2% in Sharpe—demonstrating that modulating the full temporal feature map with fundamental context (Eq. (24)) provides meaningful incremental signal over late-fusion concatenation.

FiLM Improves Consistency More than Raw Predictive Power

For the convolutional and attention-based encoders, the primary benefit of FiLM conditioning is most visible in the stability metrics (ICIR, RankICIR, IC > 0) rather than mean IC alone. FiLM-TCN achieves an ICIR of 0.730 vs. 0.665 for TCN (+9.7%), and FiLM-iTransformer achieves a RankICIR of 0.864 vs. 0.782 for iTransformer (+10.5%). The recurrent setting tells a more nuanced story: FiLM-LSTM improves portfolio metrics (Sharpe +2.4%, annualised return +2.2%) but does not improve IC-based ranking consistency, suggesting that hidden-state modulation is a less effective FiLM application point than feature-map modulation on this dataset. FiLM-TCN achieves a Signal Stability of 0.1287 (Eq. (43)), the lowest of all evaluated models as reported in Table 4, confirming that intermediate pre-aggregation conditioning reduces predictive variance relative to both late-fusion concatenation and post-pool gating.

Statistical Robustness and Regime-Based Stability

Paired t-tests on the 56-month RankIC series confirm that mean RankIC differences between FiLM and baseline models are not statistically significant at conventional thresholds (all $p > 0.05$), consistent with the modest absolute magnitudes (ΔRankIC ≤ 0.002 for the TCN family) and the limited statistical power of a 56-observation window—a known constraint of financial ML evaluation [28]. This reinforces that FiLM's primary contribution is signal stability rather than mean accuracy gains. To examine whether the stability advantage holds across market conditions, Table 6 partitions the test period into three regimes. FiLM-TCN achieves lower SD(RankIC) than TCN in all three regimes, with the largest advantages in the Regulatory Bear period (0.1022% vs. 0.1303%, −21.6%)—the most challenging market environment in the test window—and Mixed Recovery (0.1519% vs. 0.1608%, −5.5%); the Recovery period advantage is marginal (0.1011 vs. 0.1017).

Table 6: SD(RankIC) by market regime for the TCN encoder family. Lower values indicate more stable month-to-month signal. The COVID Crash period (Jan–Mar 2020, 3 months) is omitted due to insufficient observations.

Regime	TCN	GMU-TCN	FiLM-TCN
Recovery (Apr 2020–Jun 2021, n = 14)	0.1017	0.1083	**0.1011**
Regulatory Bear (Jul 2021–Oct 2022, n = 15)	0.1303	0.1212	**0.1022**
Mixed Recovery (Nov 2022–Dec 2024, n = 25)	0.1608	0.1533	**0.1519**
Full test period (n = 56)	0.1387	0.1359	**0.1287**

5.2 Fusion Mechanism Comparison and Sensitivity Analysis

GMU Comparison

To situate FiLM against a strong alternative fusion mechanism, we evaluate Gated Multimodal Unit (GMU) encoders across all three encoder families, replacing the FiLM module with a learned gating operation applied post-encoding. Table 7 reports the three-way comparison across late concatenation, GMU, and FiLM per encoder family.

Table 7: Three-way fusion mechanism comparison per encoder family: late concatenation (baseline), GMU, and FiLM. Best result per column within each encoder family is **bolded**.

Model	RankIC	RankICIR	Sharpe	Ann. Ret	IC > 0	Turnover
LSTM encoder family						
LSTM (late concat)	0.1379	1.018	1.463	22.8%	**75.0%**	59.9%
GMU-LSTM	0.1347	0.998	1.482	22.5%	**75.0%**	59.3%
FiLM-LSTM	**0.1380**	**1.029**	**1.498**	**23.3%**	71.4%	59.9%
TCN encoder family						
TCN (late concat)	0.1407	1.015	1.537	24.9%	75.0%	54.7%
GMU-TCN	**0.1466**	1.079	**1.701**	**27.7%**	78.6%	57.4%
FiLM-TCN	0.1415	**1.099**	1.633	25.3%	**80.4%**	54.9%
iTransformer encoder family						
iTransformer (late concat)	0.1102	0.782	1.015	14.7%	64.3%	70.4%
GMU-iTransformer	0.1181	0.783	0.961	14.8%	**73.2%**	64.7%
FiLM-iTransformer	**0.1288**	**0.864**	**1.228**	**19.4%**	71.4%	63.3%

For the TCN family, GMU-TCN achieves the highest raw RankIC (0.1466 vs. 0.1415) and leads on Sharpe (1.701 vs. 1.633) and annualised return (27.7% vs. 25.3%), demonstrating that post-encoding gating is a competitive fusion mechanism on mean accuracy and risk-adjusted portfolio performance. However, FiLM-TCN achieves higher RankICIR (1.099 vs. 1.079), lower Signal Stability (0.1287 vs. 0.1359), and higher IC hit rate (80.4% vs. 78.6%), confirming that pre-aggregation conditioning produces a more consistent month-to-month predictive signal. This tradeoff is structurally interpretable: GMU operates post-pooling and can sharpen final predictions but cannot reduce signal variance across time steps, whereas FiLM conditions the full temporal feature map prior to aggregation. For the LSTM family, FiLM-LSTM leads GMU-LSTM on RankIC (0.1380 vs. 0.1347), RankICIR (1.029 vs. 0.998), Sharpe (1.498 vs. 1.482), and annualised return (23.3% vs. 22.5%), while GMU-LSTM achieves a higher IC hit rate (75.0% vs. 71.4%). For the iTransformer

family, FiLM-iTransformer leads GMU-iTransformer on RankIC (0.1288 vs. 0.1181), RankICIR (0.864 vs. 0.783), Sharpe (1.228 vs. 0.961), and annualised return (19.4% vs. 14.8%), while GMU-iTransformer achieves a marginally higher IC hit rate (73.2% vs. 71.4%), suggesting that per-layer conditioning provides more integration points than a single post-encoding gate on the primary ranking and risk-adjusted metrics.

FiLM Generator Sensitivity

We evaluate variants of the FiLM generator across all three encoder families, varying generator width (narrow: 8→32→32, wide: 8→128→128) and γ bound ($[0,1]$, $[0,2]$, $[0,4]$). Results are reported in Table 8.

Table 8: FiLM generator sensitivity: variants of generator width and γ bound across all three FiLM encoder families. The original configuration for each encoder is shown in *italics*.

Model	RankIC	RankICIR	Sharpe	Ann. Ret	IC > 0
FiLM-TCN generator variants					
FiLM-TCN (original)	*0.1415*	*1.099*	*1.633*	*25.3%*	*80.4%*
FiLM-TCN (narrow gen)	0.1412	1.106	1.719	26.1%	80.4%
FiLM-TCN (wide gen)	0.1472	1.004	1.539	26.1%	78.6%
FiLM-TCN ($\gamma\in[0,1]$)	0.1414	1.057	1.655	26.4%	78.6%
FiLM-TCN ($\gamma\in[0,4]$)	0.1400	1.088	1.731	26.8%	82.1%
FiLM-LSTM generator variants					
FiLM-LSTM (original)	*0.1380*	*1.029*	*1.498*	*23.3%*	*71.4%*
FiLM-LSTM (narrow gen)	0.1373	0.997	1.490	23.4%	67.9%
FiLM-LSTM (wide gen)	0.1322	1.022	1.492	22.9%	71.4%
FiLM-LSTM ($\gamma\in[0,1]$)	0.1374	0.998	1.512	23.5%	69.6%
FiLM-LSTM ($\gamma\in[0,4]$)	0.1387	1.008	1.508	23.9%	69.6%
FiLM-iTransformer generator variants					
FiLM-iTransformer (original)	*0.1288*	*0.864*	*1.228*	*19.4%*	*71.4%*
FiLM-iTransformer (narrow gen)	0.1170	0.783	0.929	14.4%	67.9%
FiLM-iTransformer (wide gen)	0.1187	0.761	0.939	15.0%	67.9%
FiLM-iTransformer ($\gamma\in[0,1]$)	0.1170	0.788	0.975	15.0%	67.9%
FiLM-iTransformer ($\gamma\in[0,4]$)	0.1154	0.765	0.936	14.5%	67.9%

For FiLM-TCN, all variants remain within ±0.006 RankIC of the original, with the original achieving the highest RankICIR (1.099), confirming the default configuration produces the most consistent signal and that the $[0,2]$ γ bound is an appropriate design choice for signal stability. For FiLM-LSTM, all variants produce equivalent results within noise, confirming that the performance inconsistency relative to FiLM-TCN is architectural rather than generator-dependent. For FiLM-iTransformer, the original configuration outperforms all generator variants, with narrow and wide generators both degrading RankIC to 0.117, consistent with the near-identity γ values indicating the generator is already operating at its structural ceiling regardless of capacity.

iTransformer Hidden Dimension Sensitivity

We evaluate iTransformer hidden dimensions $D_h \in \{64, 128, 256\}$ to test whether larger capacity closes the performance gap with FiLM-TCN. Table 9 reports results; the finding is discussed in Section 6.3.

Table 9: iTransformer hidden dimension sensitivity: $D_h \in \{64, 128, 256\}$ for baseline and FiLM variants. The $D_h = 64$ configuration used in the main experiments is shown in *italics*.

Model	RankIC	RankICIR	Sharpe	Ann. Ret	IC > 0
iTransformer baseline					
iTransformer (D_h = 64)	*0.1102*	*0.782*	*1.015*	*14.7%*	*64.3%*
iTransformer (D_h = 128)	0.1062	0.662	0.690	10.4%	66.1%
iTransformer (D_h = 256)	0.1123	0.702	0.790	12.3%	66.1%
FiLM-iTransformer					
FiLM-iTransformer (D_h = 64)	*0.1288*	*0.864*	*1.228*	*19.4%*	*71.4%*
FiLM-iTransformer (D_h = 128)	0.1227	0.729	0.827	13.8%	67.9%
FiLM-iTransformer (D_h = 256)	0.1177	0.762	1.035	16.1%	67.9%

6 Discussion

6.1 Comparison with Baselines

XGBoost achieves the highest ICIR of 0.785 across all seven models and a competitive IC of 0.0890, outperforming all deep model variants on ICIR. This result reflects a well-established property of gradient-boosted trees on tabular financial data [2]: by operating directly on cross-sectional rank-normalised features, XGBoost produces a highly stable predictive signal with low month-to-month variance—precisely what ICIR measures. The deep models, encoding 60-day sequential dynamics, introduce additional variance that is not always compensated by incremental predictive accuracy on this monthly-frequency benchmark. That XGBoost operates on a flattened 20-dimensional feature vector with no temporal structure further suggests that much of the predictive signal in the China A-share cross-section is captured by static monthly snapshots rather than sequential dynamics.

The deep models are therefore most valuable as controlled comparisons rather than performance maximisers: because each encoder family is instantiated in both a baseline and a FiLM-conditioned variant with identical capacity, they isolate the contribution of the conditioning mechanism in a way that a tabular baseline cannot. That isolation reveals that the primary benefit of FiLM conditioning on this benchmark lies in signal stability—FiLM-TCN improves ICIR by +9.7% and RankICIR by +8.3% over its baseline—rather than in mean predictive accuracy alone. FiLM-TCN outperforms XGBoost on five of six reported metrics including IC, RankIC, RankICIR, Sharpe (1.633 vs. 1.522), and annualised return (25.30% vs. 21.97%, a 333 basis point gap net of transaction costs), with XGBoost's ICIR advantage reflecting its strength on static snapshots rather than sequential dynamics.

The GMU comparison in Table 7 provides a further baseline perspective. For the TCN family, GMU-TCN achieves higher raw RankIC (0.1466 vs. 0.1415) and a higher Sharpe ratio (1.701 vs. 1.633) than FiLM-TCN, demonstrating that post-encoding gating is a competitive fusion mechanism on mean accuracy and risk-adjusted performance. However, FiLM-TCN achieves higher RankICIR (1.099 vs. 1.079) and lower Signal Stability (0.1287 vs. 0.1359), confirming that the stability advantage of pre-aggregation conditioning is not replicated by post-encoding gating—a distinction that is only visible when a post-encoding fusion baseline is included. For the LSTM family, FiLM-LSTM leads GMU-LSTM on RankIC, RankICIR, Sharpe, and

annualised return, while GMU-LSTM achieves a higher IC hit rate (75.0% vs. 71.4%). For the iTransformer family, FiLM-iTransformer outperforms GMU-iTransformer on RankIC, RankICIR, Sharpe, and annualised return, while GMU-iTransformer achieves a marginally higher IC hit rate (73.2% vs. 71.4%). Across all three encoder families, the relative advantage of each mechanism varies by metric, confirming that performance depends on both the fusion mechanism and the dimension of evaluation.

Table 10 contextualises the practical trade-offs between model complexity and performance. XGBoost inference is near-instantaneous (<1 ms per rebalancing date); all deep models require 2.4–4.3 ms—a difference that is operationally negligible for monthly rebalancing workflows where scoring the full universe occurs once per month. The meaningful trade-off is therefore model complexity and data requirements rather than latency. FiLM-TCN introduces a modest +18.0% parameter overhead (82,241 vs. 69,697) relative to its TCN baseline—the most parameter-efficient fusion strategy evaluated. By contrast, GMU-iTransformer introduces +290.3% overhead (810,241 parameters) without a corresponding stability benefit. XGBoost remains preferable when GPU access is unavailable, training history is limited, or interpretability is required. FiLM-TCN is preferable when monthly rebalancing makes inference latency irrelevant, temporal pattern extraction is valued, and signal stability across market regimes is the primary objective.

Table 10: Model complexity and per-rebalancing-date inference time. Parameter overhead is relative to the encoder family baseline. Inference time is the mean of 10 forward passes on the test GPU.

Model	Parameters	Overhead	Inference (ms)
XGBoost	N/A	—	<1
LSTM	57,985	—	4.1
FiLM-LSTM	70,529	+21.6%	4.3
GMU-LSTM	62,721	+8.2%	4.2
TCN	69,697	—	3.0
FiLM-TCN	82,241	+18.0%	2.8
GMU-TCN	74,433	+6.8%	2.7
iTransformer	207,617	—	2.4
FiLM-iTransformer	411,009	+98.0%	3.0
GMU-iTransformer	810,241	+290.3%	2.8

6.2 FiLM Modulation Parameters and Mechanism Evidence

We extract γ and β parameters from each FiLM encoder across the full test set (3-seed ensemble) to verify active modulation. FiLM-LSTM and FiLM-TCN produce γ means of 0.936 and 0.957 respectively (both std 0.039), confirming active suppression relative to the identity $\gamma = 1$. FiLM-iTransformer converges near identity ($\gamma = 0.997$, std 0.011), consistent with the architectural incompatibility discussed in Section 6.3. Stratifying by ROE quintile, FiLM-TCN β shifts monotonically from -0.029 at Q1 (lowest ROE) to -0.046 at Q5 (highest ROE), indicating progressively stronger downward shifting for high-quality firms and directly substantiating the mechanism-switcher hypothesis. FiLM-LSTM shows near-zero β across all quintiles and FiLM-iTransformer near-identity γ across all quintiles, corroborating that meaningful conditioning requires pre-aggregation access to temporally-rich representations. Figs. 4–6 show the full stratifications.

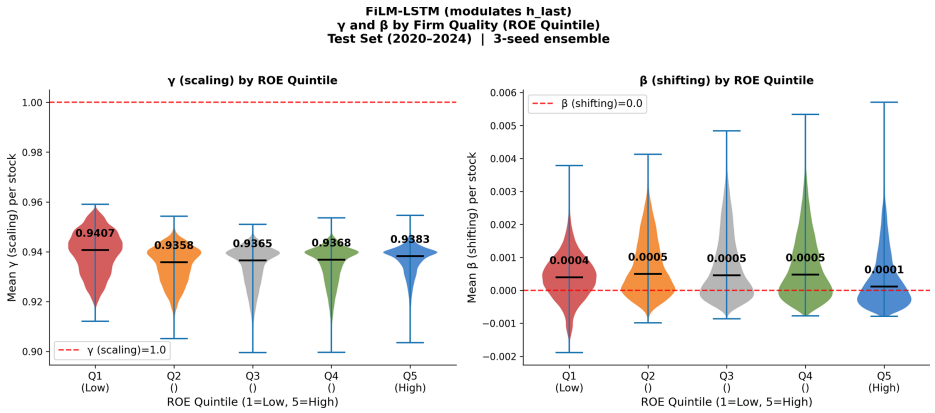

Figure 4: γ and β by ROE quintile, FiLM-LSTM. Near-zero β across all quintiles, consistent with post-encoding placement.

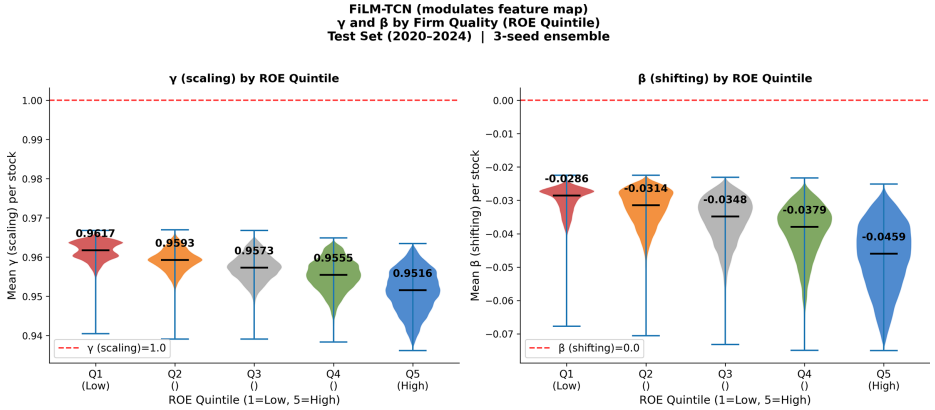

Figure 5: γ and β by ROE quintile, FiLM-TCN. β shifts monotonically from −0.029 (Q1) to −0.046 (Q5), substantiating the mechanism-switcher hypothesis.

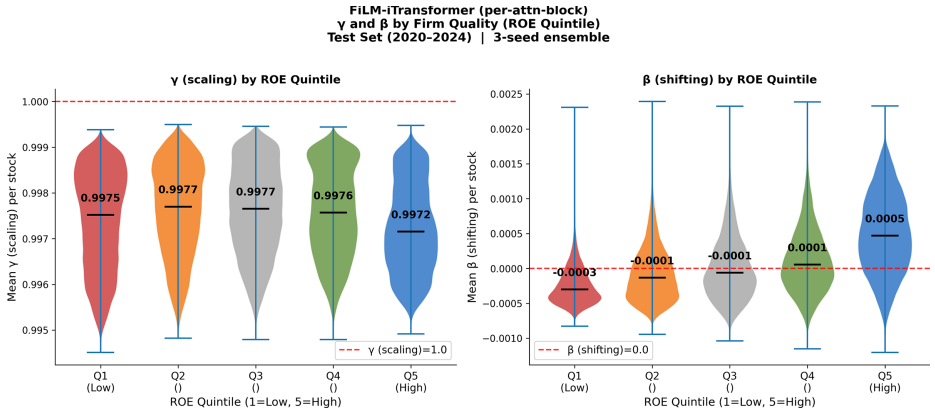

Figure 6: γ and β by ROE quintile, FiLM-iTransformer. Near-identity values across all quintiles, consistent with architectural incompatibility.

A permutation feature importance analysis identifies which accounting metrics drive the modulation gains: each of the eight fundamental features is zeroed out in turn and the RankIC drop measured across the test set, averaged over three seeds and three repeats. The baseline RankIC in each figure (0.1361 for FiLM-TCN; 0.1195 for FiLM-iTransformer) reflects per-seed evaluation averaged across seeds, and is therefore lower than the ensemble RankIC in Table 4 (0.1415; 0.1288), where scores are averaged across seeds *before* rank computation—a standard variance-reduction property of score-level ensembling. Net profit, price-to-book ratio (most recent quarter), and price-to-sales ratio (trailing twelve months) emerge as the top three features for both FiLM-TCN and FiLM-LSTM, with RankIC drops of +0.0185, +0.0093, and +0.0077 for FiLM-TCN. FiLM-iTransformer shares the same top-three ranking, consistent with limited exploitation of the conditioning signal. The consistent top-three ranking of netProfit, pbMRQ, and psTTM across FiLM-TCN and FiLM-LSTM—the two architectures where FiLM conditioning is effective—confirms that these profitability and valuation metrics are the primary drivers of modulation. For FiLM-iTransformer, a larger proportion of features show negative importance, consistent with the near-identity γ values (Section 6.3) indicating that the conditioning signal is minimally exploited regardless of which feature is masked. Full importance profiles are shown in Figs. 7–9.

6.3 FiLM Conditioning and Architectural Compatibility

FiLM-iTransformer introduces a +98.0% parameter overhead relative to its baseline, the largest of the three encoder families, yet delivers only moderate improvements—smaller in magnitude than those achieved by FiLM-TCN relative to TCN. This disproportion reflects an architectural compatibility issue: the benefit of FiLM conditioning depends on where modulation is applied, and the iTransformer's variate-level attention architecture is structurally less suited to temporal conditioning than the TCN's feature maps.

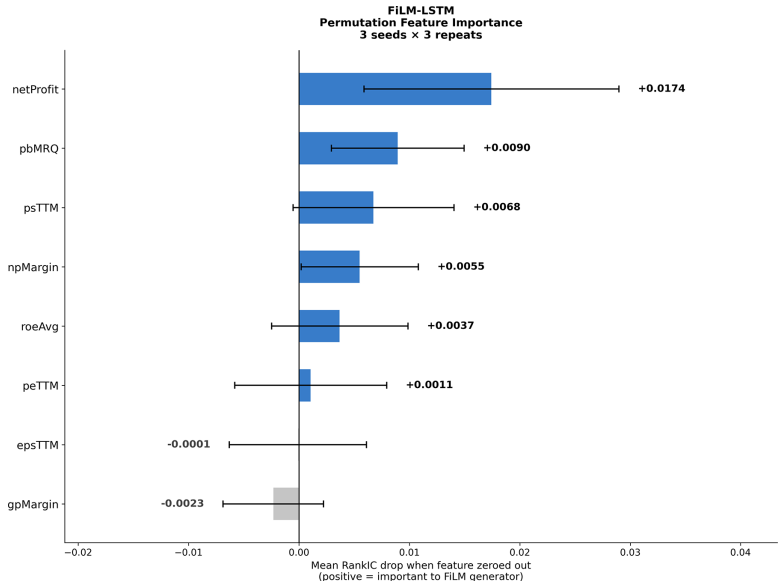

Figure 7: Permutation feature importance, FiLM-LSTM. Mean RankIC drop per feature, averaged across 3 seeds × 3 repeats.

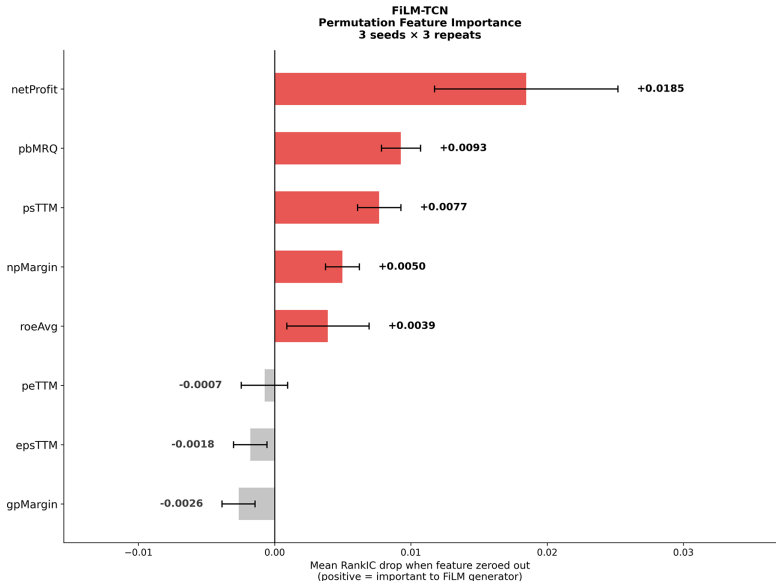

Figure 8: Permutation feature importance, FiLM-TCN. Mean RankIC drop per feature, averaged across 3 seeds × 3 repeats.

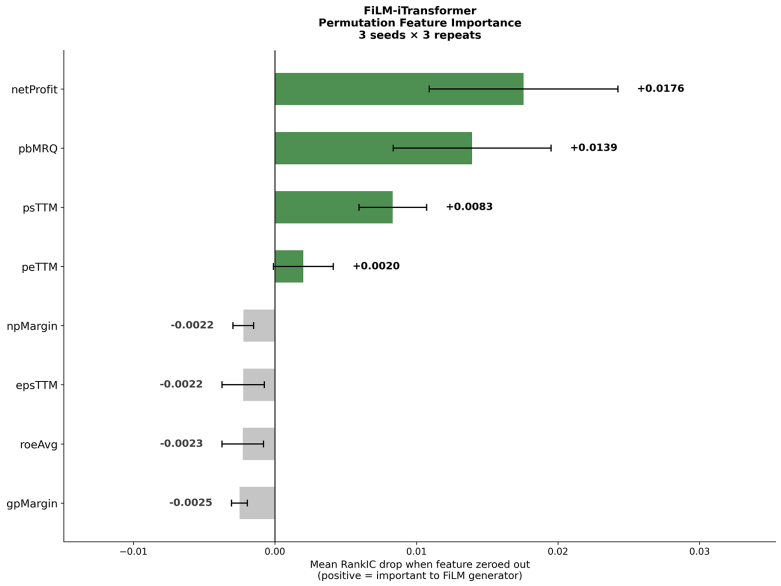

Figure 9: Permutation feature importance, FiLM-iTransformer. Mean RankIC drop per feature, averaged across 3 seeds × 3 repeats.

The iTransformer operates across the variate dimension rather than the time dimension, treating each feature as a token and modelling inter-feature dependencies at each layer. FiLM conditioning applied inside the attention blocks therefore modulates cross-feature interactions rather than temporal patterns—a subtly different operation than the one motivated in Section 4.1.2, where the goal is to govern how sequential dynamics are extracted. In contrast, FiLM-TCN applies modulation to the full temporal feature map prior to pooling, directly governing how each time step contributes to the final representation. This structural

alignment between the conditioning mechanism and the encoder's processing axis appears to be a necessary condition for efficient gain.

The D_h sensitivity results in Table 9 confirm that the performance gap is not a capacity limitation. For FiLM-iTransformer, larger hidden dimensions consistently degrade RankIC relative to $D_h = 64$ (0.1288 → 0.1227 → 0.1177). Both larger variants also show substantially lower RankICIR and Sharpe than $D_h = 64$, though the relationship between $D_h = 128$ and $D_h = 256$ is non-monotonic—Sharpe partially recovers from 0.827 at $D_h = 128$ to 1.035 at $D_h = 256$, while remaining well below 1.228 at $D_h = 64$. The near-identity γ mean of 0.997 (std 0.011) extracted from FiLM-iTransformer—compared to 0.957 (std 0.039) for FiLM-TCN—provides a direct mechanistic explanation: the iTransformer's architecture provides insufficient temporal structure for the conditioning signal to exploit, so the generator learns to produce minimal modulation regardless of capacity. A more parameter-efficient future design—for example, low-rank conditioning at the input embedding rather than inside each attention block—may improve the efficiency–performance trade-off.

The FiLM-LSTM results warrant consideration within the same framework. FiLM-LSTM applies conditioning at $\mathbf{h}_{\text{last}}$—a collapsed vector produced after all temporal extraction is complete—which prevents the fundamental signal from influencing how individual time steps are weighted. The mixed performance of FiLM-LSTM is therefore consistent with this structural post-encoding placement rather than a failure of the FiLM mechanism. The FiLM-LSTM vs. FiLM-TCN performance gap serves as implicit empirical support that pre-aggregation conditioning on temporally-rich representations is structurally more effective than post-sequence conditioning on a collapsed hidden state.

6.4 Limitations

Single Market and Asset Class

All experiments are conducted on China A-share equities. Chinese equity markets have distinctive structural properties—high retail investor participation, circuit-breaker mechanisms, periodic regulatory interventions, and a relatively short history of systematic factor investing—that may make them an unusual or unrepresentative testbed. The extent to which the findings generalise to other equity markets (e.g., US, European, or emerging-market equities), other asset classes (e.g., commodities, fixed income), or higher-frequency trading environments is unknown and constitutes an important direction for future work.

Transaction Cost Assumptions

The 30 bps one-way cost assumption follows standard practice in Chinese A-share quantitative research and is applied conservatively: actual measured turnover across all nine deep models (Tables 5 and 7) ranges from 54.7% to 70.4%, meaning the 100% assumption overstates costs and the reported net returns are lower bounds on practically achievable performance under realistic execution.

Hyperparameter Sensitivity

The architectural hyperparameters (hidden dimension $D_h = 64$, sequence length $L = 60$, FiLM MLP depth and width) were selected based on domain knowledge and computational budget rather than systematic grid search. It is possible that different configurations would alter the relative ranking of models, particularly for the iTransformer family which may benefit from larger D_h given its quadratic attention complexity.

Portfolio Risk for FiLM-iTransformer

FiLM-iTransformer records the largest maximum drawdown among the models in Table 5 (−19.08%), substantially exceeding its late-fusion baseline (−14.51%). This elevated drawdown is consistent with the

architectural incompatibility discussed in Section 6.3: without meaningful temporal conditioning, the added parameter budget introduces instability in portfolio construction rather than improving risk management.

Fundamental Data Coverage

Quarterly fundamental features are only available from 2009 onward, and are missing for a non-trivial fraction of stocks at any given rebalancing date (imputed with zero after normalisation). The FiLM conditioning signal is therefore absent or uninformative for a subset of the training data, which may understate the potential benefit of fundamental conditioning under more complete disclosure regimes.

7 Conclusion

The central finding of this work is not that FiLM-conditioned models uniformly outperform their baselines, but that intermediate conditioning reliably shifts the locus of gain toward signal stability and risk-adjusted performance rather than mean predictive accuracy—a pattern that holds across two architecturally distinct encoder families and survives transaction costs in portfolio simulation. This suggests a reframing of the role of fundamental data in sequential models: rather than serving purely as additive predictive signal appended at the output, quarterly accounting fundamentals can act as a stabilising contextual prior when injected at intermediate encoder layers.

The architecture–fusion interaction uncovered here provides a further analytical perspective. Across three fusion strategies—late concatenation, post-encoding gating (GMU), and intermediate conditioning (FiLM)—the primary differentiator is not which mechanism is used but where conditioning is applied: feature-map modulation (FiLM-TCN) produces the most consistent signal stability improvements—lowest $\sigma(\text{RankIC})$ and highest RankICIR among all evaluated models—while hidden-state modulation (FiLM-LSTM) improves portfolio metrics but not ranking consistency, and per-layer attention modulation (FiLM-iTransformer) yields intermediate improvements at disproportionate parameter cost. Extracted γ and β parameters confirm that FiLM-TCN applies systematically different modulation across firm quality strata, with β shifting monotonically from -0.029 at the lowest ROE quintile to -0.046 at the highest—providing empirical grounding for the mechanism-switcher hypothesis rather than merely asserting it.

Three directions for future work are most immediate. First, the conditioning framework should be evaluated on other markets and asset classes to assess generalisability beyond the China A-share benchmark. Second, the FiLM generator design for the iTransformer warrants targeted investigation—low-rank projection at the input embedding rather than per-layer attention conditioning may achieve better efficiency–performance trade-offs. Third, the heterogeneous-frequency fusion problem studied here—a static low-frequency signal conditioning a high-frequency sequential encoder—arises broadly across computational modelling disciplines, and the stability-vs.-accuracy tradeoff observed here may inform analogous fusion problems in other domains.

Acknowledgement: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2023-00242528).

Funding Statement: This research was funded by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT), grant number RS-2023-00242528.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Maurice Kyla Octaviano and Jin-Taek Seong; methodology, Maurice Kyla Octaviano; software, Maurice Kyla Octaviano; validation, Maurice Kyla Octaviano and Jin-Taek Seong; formal analysis, Maurice Kyla Octaviano; investigation, Maurice Kyla Octaviano; data curation, Maurice Kyla Octaviano; writing—original draft preparation, Maurice Kyla Octaviano;

writing—review and editing, Jin-Taek Seong; supervision, Jin-Taek Seong; project administration, Jin-Taek Seong; funding acquisition, Jin-Taek Seong. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the BaoStock financial data API (http://www.baostock.com).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Grinold RC, Kahn RN. Active portfolio management: a quantitative approach for producing superior returns and controlling risk. 2nd ed. New York, NY, USA: McGraw-Hill; 1999.
2. Gu S, Kelly B, Xiu D. Empirical asset pricing via machine learning. Rev Financ Stud. 2020;33(5):2223–73. doi:10.3386/w25398.
3. Perez E, Strub F, de Vries H, Dumoulin V, Courville A. FiLM: visual reasoning with a general conditioning layer. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI); 2018 Feb 2–7; New Orleans, LA, USA. p. 3942–51.
4. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. doi:10.1162/neco.1997.9.8.1735.
5. Bai S, Kolter JZ, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271. 2018.
6. Liu Y, Hu T, Zhang H, Wu L, Wang S, Ma L, et al. iTransformer: inverted transformers are effective for time series forecasting. In: Proceedings of the International Conference on Learning Representations (ICLR); 2024 May 7–11; Vienna, Austria. p. 1–24.
7. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016 Aug 13–17; San Francisco, CA, USA. p. 785–94.
8. Fama EF, French KR. Common risk factors in the returns on stocks and bonds. J Financ Econ. 1993;33(1):3–56. doi:10.1016/0304-405x(93)90023-5.
9. Jegadeesh N, Titman S. Returns to buying winners and selling losers: implications for stock market efficiency. J Financ. 1993;48(1):65–91. doi:10.1111/j.1540-6261.1993.tb04702.x.
10. Kelly BT, Pruitt S, Su Y. Characteristics are covariances: a unified model of risk and return. J Financ Econ. 2019;134(3):501–24.
11. Feng G, He J, Polson NG, Xu J. Deep learning in characteristics-sorted factor models. J Financ Quant Anal. 2024;59(7):3001–36. doi:10.1017/s0022109023000893.
12. Fan J, Shen Y. StockMixer: a simple yet strong MLP-based architecture for stock price forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2024 Feb 20–27; Vancouver, BC, USA. p. 8681–9.
13. Leippold M, Wang Q, Zhou W. Machine learning in the Chinese stock market. J Financ Econ. 2022;145(2):64–82. doi:10.1016/j.jfineco.2021.08.017.
14. Ma T, Leong WJ, Jiang F. A latent factor model for the Chinese stock market. Int Rev Financ Anal. 2023;87(6):102555. doi:10.1016/j.irfa.2023.102555.
15. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017;30:5998–6008. doi:10.65215/ctdc8e75.
16. Nie Y, Nguyen NH, Sinthong P, Kalagnanam J. A time series is worth 64 words: long-term forecasting with transformers. In: Proceedings of the International Conference on Learning Representations (ICLR); 2023 May 1–5; Kigali, Rwanda. p. 1–17.
17. Li T, Liu Z, Shen Y, Wang X, Chen H, Huang S. MASTER: market-guided stock transformer for stock price forecasting. Proc AAAI Conf Artif Intell. 2024;38(1):162–70. doi:10.1609/aaai.v38i1.27767.

18. Xu Y, Cohen SB. Stock movement prediction from tweets and historical prices. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics; 2018. p. 1970–9.
19. Birnbaum S, Kuleshov V, Enam SZ, Koh PW, Ermon S. Temporal FiLM: capturing long-range sequence dependencies with feature-wise modulations. Adv Neural Inf Process Syst. 2019;32:1–12.
20. Brockschmidt M. GNN-FiLM: graph neural networks with feature-wise linear modulation. PMLR. 2020;119:1144–54.
21. Liu TY. Learning to rank for information retrieval. Found Trends Inf Retr. 2009;3(3):225–331. doi:10.1561/1500000016.
22. Xia F, Liu TY, Wang J, Zhang W, Li H. Listwise approach to learning to rank: theory and algorithm. In: Proceedings of the 25th International Conference on Machine Learning (ICML); 2008 Jul 5–9; Helsinki, Finland. p. 1192–9.
23. Wilder JW. New concepts in technical trading systems. Greensboro, NC, USA: Trend Research; 1978.
24. Loshchilov I, Hutter F. Decoupled weight decay regularization. In: Proceedings of the 7th International Conference on Learning Representations (ICLR); 2019 May 6–9; New Orleans, LA, USA. p. 1–18.
25. Smith LN. Super-convergence: very fast training of neural networks using large learning rates. Artif Intell Mach Learn Multi-Domain Oper Appl. 2019;11006:1–15.
26. Sharpe WF. The sharpe ratio. J Portf Manag. 1994;21(1):49–58. doi:10.3905/jpm.1994.409501.
27. Magdon-Ismail M, Atiya AF. Maximum drawdown. Risk Mag. 2004;17(10):99–102.
28. Arian HR, Norouzi Mobarekeh D, Seco LA. Backtest overfitting in the machine learning era: a comparison of out-of-sample testing methods in a synthetic controlled environment. Knowl Based Syst. 2024;300:112194.