THRESHOLD-CONSISTENT LEARNING FOR CONTINUOUS FORECASTING AND RARE-EVENT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous forecasting and rare-event detection are often treated as separate objectives in financial time-series forecasting, although they describe two aspects of the same future outcome. A regression model may achieve low prediction error while assigning insufficiently high scores to rare but decision-critical events, such as unusually large positive returns. We propose **Threshold-Consistent Continuous–Discrete Learning (TCDL)**, a unified framework that jointly learns continuous return prediction and threshold-defined event states. The key idea is to couple the continuous prediction with a discrete state representation through a threshold-consistency objective, encouraging the predicted return and inferred event state to remain logically aligned during optimization. In this way, the continuous task preserves fine-grained numerical information, while the discrete task improves sensitivity to rare tail events. We evaluate TCDL across major global market indices against strong time-series forecasting baselines. The proposed method maintains strong regression accuracy while achieving competitive rare-event discrimination and substantially improving tail AUPRC over conventional regression and joint-learning objectives. Statistical significance tests and controlled ablations further isolate the contribution of the threshold-consistency mechanism. These results show that coupling continuous predictions with threshold-defined decision states provides a simple and effective learning paradigm for forecasting problems in which rare events are more consequential than average prediction error.

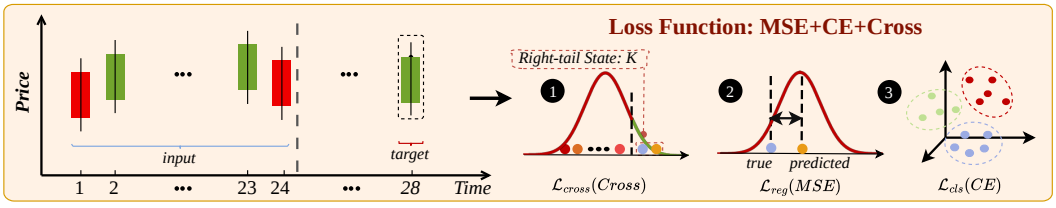

Figure 1: Overview of TCDL with regression, state classification, and threshold consistency.

1 INTRODUCTION

Financial time series forecasting poses a learning challenge: the numerical accuracy does not necessarily determine its practical value. In many forecasting settings, large positive movements are considerably more consequential than small deviations around the conditional mean (Nevasalmi, 2020). However, existing forecasting models are predominantly trained with point-wise regression objectives, such as Mean Squared Error (MSE)(Zhu et al., 2025; Xu et al., 2025), which treat all target values according to their numerical prediction error rather than their decision relevance.

This mismatch is particularly important in rare-event forecasting, where practical decisions depend on whether a continuous target crosses a decision-relevant threshold. A model can achieve a small regression error yet fail to distinguish genuinely extreme events, while a classifier may predict a tail event even when the corresponding continuous forecast remains below the same threshold. We propose **Threshold-Consistent Continuous–Discrete Learning (TCDL)**, a unified framework that jointly optimizes continuous return forecasting and threshold-defined state prediction, as illustrated in Fig. 1.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

TCDL combines three objectives: (1) a regression objective that preserves fine-grained information about future returns; (2) a classification objective that identifies discrete market states, including an explicit right-tail state; and (3) a threshold-consistency objective that penalizes disagreement between the predicted tail state and continuous forecast. Our main contributions are summarized as follows:

- We formulate continuous forecasting and rare-event detection as a coupled learning problem and propose **TCDL**, which jointly optimizes continuous regression, discrete state classification, and cross-task threshold consistency.
- We isolate the effect of threshold-consistent learning through controlled comparisons with conventional regression and alternative tail-aware objectives. The results show that prediction-conditioned consistency improves tail informativeness beyond regression or classification supervision alone.
- Across nine equity indices and two chronological test periods, **TCDL** achieves strong regression accuracy while retaining competitive right-tail discrimination. We further assess robustness through hyperparameter sensitivity, representation-space analysis, and a validation-calibrated backtest.

The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 motivates the problem and identifies key forecasting limitations. Section 4 introduces the proposed TCDL framework and its threshold-consistent learning objective. Section 5 presents the experimental setup and empirical evaluation. Section 6 concludes the paper and discusses future research directions. Additional implementation details, evaluation definitions, and results are provided in the Appendix A.

2 Related Work

Time Series Forecasting. Transformer-based architectures have become a dominant paradigm in time-series forecasting (Wen et al., 2023), with recent studies exploring patch-based representations, linear embeddings, and other efficient architectures to improve long-horizon prediction (Zhou et al., 2021; Wu et al., 2021; Challu et al., 2023). However, recent analyses suggest that time-series representations often exhibit substantially lower effective dimensionality, suggesting that some popular Transformer-based forecasting architectures may be unnecessarily complex for certain tasks (Yu et al., 2026; Zheng et al., 2025; Zeng et al., 2023). More importantly, general-purpose forecasting benchmarks and point-wise accuracy metrics do not necessarily capture the characteristics and practical objectives of financial prediction (Spiliotis et al., 2020; Moraffah et al., 2021). In financial markets, practical utility is often more closely related to short-term directional movements and economically meaningful signals than to improvements in long-horizon numerical accuracy (Tuncer et al., 2022; Sezer & Ozbayoglu, 2018; Alamdari et al., 2024).

Learning Objectives for Forecasting. Point-wise regression losses dominate time-series forecasting, but minimizing average prediction error does not necessarily provide sufficient supervision for infrequent, decision-critical outcomes (Kosma et al., 2022). Alternative objectives have been developed to place greater emphasis on specific regions of the target distribution or difficult samples. Quantile regression targets distributional regions (Koenker & Bassett, 1978), while focal loss prioritizes rare or difficult samples (Lin et al., 2020). However, these objectives do not explicitly model threshold-crossing events. Related approaches introduce discrete supervision or representations into continuous forecasting. For example, Sun et al. (2025) employs hierarchical classification objectives and a hierarchical consistency loss to encourage agreement among predictions at different levels of temporal granularity. Ma et al. (2026), in contrast, learns discrete latent representations via vector quantization, enabling recurring local patterns to be represented by learnable codewords. These developments highlight the value of specialized learning objectives, but the interaction between continuous forecasts and threshold-defined decisions remains underexplored.

Research Gap in Financial Forecasting Objectives. Financial forecasting frequently relies on whether a predicted return crosses an economically meaningful threshold, while rare but consequential movements may receive limited supervision under conventional regression losses. More broadly, recent work has shown that explicitly improving consistency between coupled prediction objectives can be beneficial (Liu et al., 2026), yet such consistency has received limited attention in financial forecasting. TCDL addresses this gap by treating continuous regression and threshold-defined classification as two coupled views of the same future outcome and enforcing their consistency during training. Its threshold-consistency objective uses the predicted discrete state to constrain the continuous forecast, reducing cross-task contradictions within the optimization process.

3 Motivation and Problem Analysis

Financial time-series forecasting has advanced with the adoption of Transformer-based architectures (Coelho e Silva et al., 2024). However, existing forecasting models exhibit several limitations.

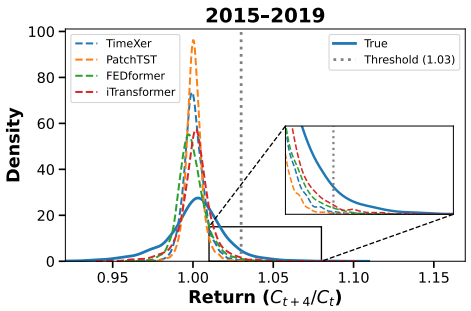

Figure 2: Distribution comparison between realized future returns and models.

Right-tail insensitivity. A prominent issue is the limited sensitivity of point forecasts to the right tail of the return distribution. In practice, models may assign insufficiently differentiated scores to samples associated with large positive returns, which are particularly important in financial decision-making. In our empirical analysis, several representative Transformer-based models implemented via the `neuralforecast` library (Olivares et al., 2022) with default settings, are evaluated on six major indices (S&P 500, Nikkei 225, CAC 40, FTSE 100, DAX 30, and DJIA) using 24-day OHLC inputs to predict the four-day-ahead return $r_t = C_{t+4}/C_t$. These models consistently produce concentrated return forecasts. As illustrated in Fig. 2, the forecasts are substantially narrower than the realized return distributions, with few values extending into the right-tail region (e.g., $r_t > 1.03$). This raises the question of whether point forecasts preserve sufficient ranking information to distinguish genuinely extreme positive outcomes.

Limitations of point-wise evaluation metrics. The right-tail insensitivity discussed above can partly arise from the widespread use of point-wise objectives such as MSE and Mean Absolute Error (MAE) (Thakkar & Chaudhari, 2021). These metrics emphasize average prediction accuracy and penalize deviations uniformly, encouraging models to focus on the central mass of the distribution. Under the random walk assumption, price dynamics follow $C_{t+4} = C_t + \epsilon_t$, where ϵ_t is zero-mean noise, and the optimal predictor satisfies $\mathbb{E}[C_{t+4} \mid C_t] = C_t$. As a result, using the most recent observation C_t as the prediction can already be near-optimal (Meese & Rogoff, 1983). As shown in Fig. 3, Transformer-based models struggle to consistently outperform this naive baseline, suggesting that small gains in point-wise accuracy may not translate into stronger tail discrimination or decision-relevant forecasts.

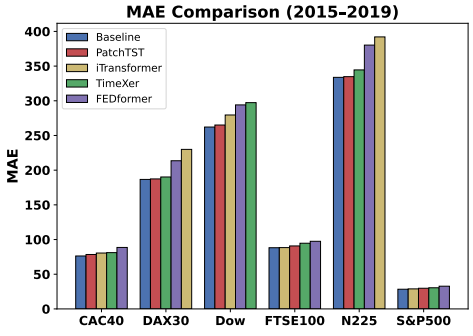

Figure 3: MAE comparison across models and market indices (2015–2019).

Inconsistency between continuous prediction and threshold-based decisions. Beyond numerical accuracy, financial forecasting often requires relating a continuous forecast to a threshold-defined decision state, such as whether a future return exceeds a critical threshold τ. Conventional forecasting models optimize continuous prediction errors without explicitly encoding this threshold relation (Lee & Kim, 2020; Chen et al., 2023). As a result, forecasts with similar numerical errors may carry different decision implications, while the threshold itself induces a precision–coverage trade-off: stricter thresholds yield fewer but more reliable signals, whereas relaxed thresholds increase coverage at the cost of precision (see Fig. 4). This motivates incorporating the relation between continuous forecasts and threshold-defined states directly into the training objective (Park et al., 2022). In TCDL, the predicted discrete state provides training-time supervision for the continuous forecast, encouraging the two prediction spaces to remain semantically aligned during optimization.

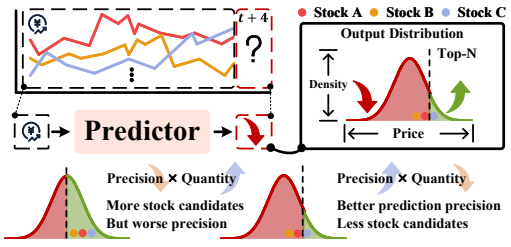

Figure 4: Threshold-based decision making induces a precision–coverage trade-off.

4 METHODOLOGY

Given a historical OHLC sequence, TCDL jointly predicts a continuous future return and a decision-oriented discrete market state. Let r_t denote the future return and $\tau = 1.03$ the right-tail threshold. Rather than replacing the continuous target with a discrete event label, we retain both predictions and couple them through a threshold-consistent objective. An overview of the proposed TCDL framework is shown in Fig. 5.

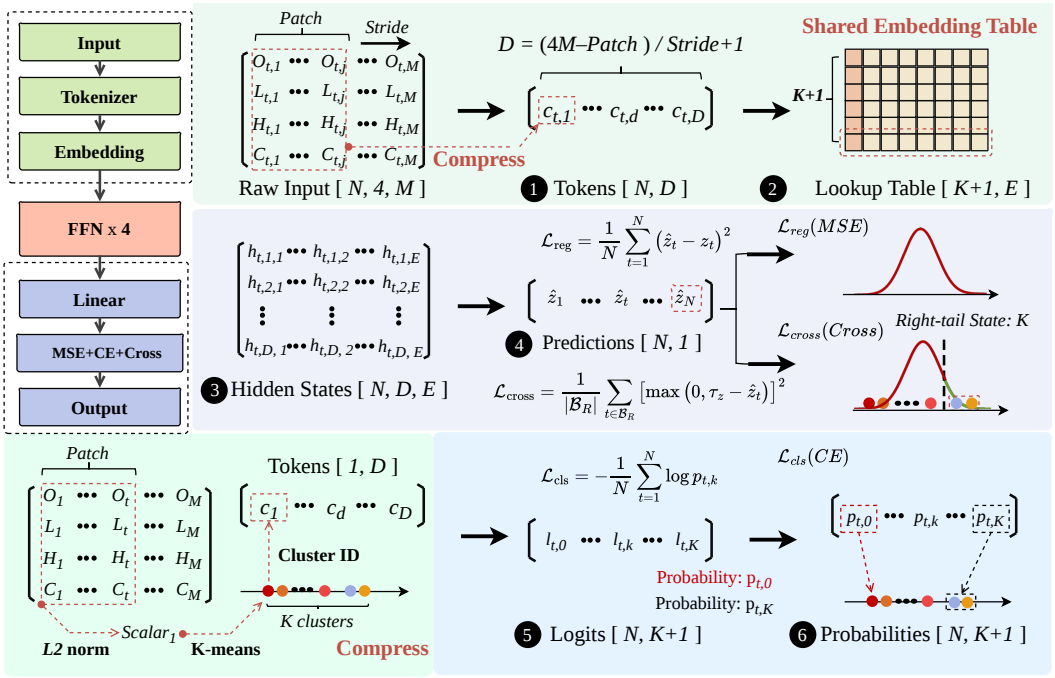

Figure 5: Overview of the proposed TCDL framework. Historical OHLC observations are compressed into discrete tokens through patch-based tokenization and K-means clustering, followed by shared embedding lookup and feature extraction. The resulting hidden representations are jointly optimized with the regression loss $\mathcal{L}_{\text{reg}}$, classification loss $\mathcal{L}_{\text{cls}}$, and threshold-consistency loss $\mathcal{L}_{\text{cross}}$.

Cluster-based Temporal Tokenization. The normalized OHLC observations are flattened and divided into overlapping windows of length W with stride S. Each window $\mathbf{x} \in \mathbb{R}^{d_w}$ is compressed into the normalized magnitude $d(\mathbf{x}) = \|\mathbf{x}\|_2/\sqrt{d_w}$. K-means with K clusters is fitted exclusively on training-set windows, and the resulting cluster indices are used as discrete temporal tokens. We retain the first $L = 5$ cluster tokens, whose windows lie entirely within the 24-day historical input, as context. The ordinary target state c_t is obtained from the K-means assignment of the last sliding window in the complete 28-day sample, which ends at $t + 4$ and therefore includes the four-day prediction horizon. The prediction state is then defined as $s_t = (1 - \mathbb{I}[r_t > \tau])c_t + \mathbb{I}[r_t > \tau]K$, yielding $K + 1$ states in total, where state K represents the explicit right-tail event.

Dual-Head Lightweight Encoder. The L historical tokens are mapped to d-dimensional learnable embeddings, combined with sinusoidal positional encodings, and processed by a lightweight four-block attention-free encoder with pre-LayerNorm residual linear projections and position-wise feed-forward layers. The final token representations are mean-pooled into a shared representation $\mathbf{h}_t$, which is passed to a regression head producing $\hat{z}_t$ and a classification head producing $K + 1$ state probabilities $\mathbf{p}_t$. For regression, the excess return is standardized using training-set statistics as $z_t = ((r_t - 1) - \mu_r)/\sigma_r$, while the corresponding threshold is $\tau_z = ((\tau - 1) - \mu_r)/\sigma_r$. The standardized prediction can be restored to the native return scale by $\hat{r}_t = 1 + \sigma_r \hat{z}_t + \mu_r$.

Threshold-Consistent Joint Learning. The two heads are first optimized with regression loss $\mathcal{L}_{\text{reg}} = \text{MSE}(\hat{z}_t, z_t)$ and classification loss $\mathcal{L}_{\text{cls}} = \text{CE}(\mathbf{p}_t, s_t)$. To explicitly couple their predictions, let $\hat{s}_t = \arg\max_k p_{t,k}$ and let $\mathcal{B}_R = \{t : \hat{s}_t = K\}$ denote samples predicted as belonging to the right-tail state. For these samples, the regression output is required to exceed the standardized

right-tail threshold:

$$\mathcal{L}_{\text{cross}} = \frac{1}{|\mathcal{B}_R|} \sum_{t \in \mathcal{B}_R} \left[\max\left(0, \tau_z - \hat{z}_t\right)\right]^2, \tag{1}$$

where $\mathcal{L}_{\text{cross}} = 0$ when $\mathcal{B}_R$ is empty. From a constrained optimization perspective, TCDL imposes the directional condition $\hat{s}_t = K \Rightarrow \hat{z}_t \geq \tau_z$. The cross loss can therefore be interpreted as a soft squared-hinge penalty for violations of this constraint, with zero cost once the prediction enters the feasible region. This design transfers discrete right-tail supervision to the continuous forecast without forcing threshold-consistent predictions toward the decision boundary. Because continuous return prediction is the primary task and the classification head provides auxiliary supervision, the reverse direction is not enforced. The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{cross}}\mathcal{L}_{\text{cross}}. \tag{2}$$

with $\lambda_{\text{cls}} = 0.5$ and $\lambda_{\text{cross}} = 0.1$ in the default configuration. The crossing term therefore acts only as a training-time penalty for cases in which the discrete head predicts a right-tail state while the continuous prediction remains below the corresponding threshold.

5 Experimental Setup and Results

We evaluate TCDL on nine representative equity indices: the S&P 500, DJIA, DAX 30, FTSE 100, CAC 40, Nikkei 225, Hang Seng Index, SSE 50, and CSI 300, with daily OHLC data collected from AKShare (King & Zhang, 2022). Two chronological train–validation–test configurations are used, as shown in Table 1; within each split, the three periods are strictly ordered and non-overlapping.

Table 1: Temporal splits used for evaluation.

Split	Training	Validation	Test
Split 1	1999–2008	2009–2014	2015–2019
Split 2	2005–2014	2015–2020	2021–2025

Experimental configuration. Each sample contains 24 trading days of OHLC observations, corresponding to 96 scalar inputs, and predicts the four-day-ahead closing-price return $r_t = C_{t+4}/C_t$. A return above 1.03 is treated as a right-tail event, corresponding to an economically meaningful short-horizon market movement.[1] Following Section 4, TCDL uses window length $W = 80$, stride $S = 4$, $K = 60$ clusters, and retains $L = 5$ historical tokens. Token embeddings have dimension $d = 64$, and the encoder contains four residual blocks with dropout 0.1. The model has 178,558 trainable parameters.

Baselines and evaluation. We compare against Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022), DLinear&NLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), TimeMixer (Wang et al., 2024a), TimeXer (Wang et al., 2024b), iTransformer (Liu et al., 2024), and a parameter-free Random Walk reference. TCDL operates on five discrete tokens, whereas the learned baselines receive the original 96-value sequence. Performance is measured using MSE and MAE for point forecasting, and AUPRC and AUROC for right-tail discrimination. For each seed (1,42,1337), predictions from all nine indices are pooled before metric computation, and we report the mean and standard deviation across the three runs. To assess cross-market consistency, each metric is first averaged over seeds for each index, after which TCDL is compared with learned baselines using a two-sided paired Wilcoxon signed-rank test. Holm correction is applied across the eight learned-baseline comparisons for each metric and test period at $\alpha = 0.05$. Random Walk is excluded from this correction family.

5.1 Overall Results and Discussion

Table 2 summarizes the pooled results over the two evaluation periods. Overall, TCDL shows its clearest advantage in point forecasting, while retaining competitive right-tail discrimination with a compact architecture.

Forecasting accuracy. TCDL achieves the lowest MAE in both periods, reaching 0.016053 in 2015–2019 and 0.015831 in 2021–2025. It also ranks second in MSE in both periods, surpassed only by the parameter-free Random Walk while outperforming all learned baselines. Random Walk performs strongly in MSE because short-horizon return ratios are typically concentrated near 1, favoring conservative persistence-based predictions. In contrast, TCDL produces informative return

[1] The 3% threshold reflects market-move levels used in some trading-control mechanisms; see Aju Press.

estimates while simultaneously supporting tail-event prediction. The improvements in MSE and MAE are significant against all learned baselines in the first period and against most baselines in the second period after Holm correction.

Right-tail discrimination. TCDL remains competitive on tail-oriented metrics without uniformly dominating the strongest baseline. In 2015–2019, it achieves the second-highest AUPRC (0.146371), close to Autoformer, while significantly outperforming several competing models. In 2021–2025, its AUPRC remains competitive at 0.132573, although Autoformer and TimeXer perform better. AUROC shows a similar pattern: TCDL remains among the stronger methods in both periods, but no statistically significant advantage is observed over the best learned baselines. By contrast, competing methods often favor one objective at the expense of the other: models with relatively low regression error tend to show weaker right-tail discrimination, whereas stronger tail-oriented performance is often accompanied by substantially larger point-forecasting errors.

Table 2: Pooled four-day-ahead forecasting and right-tail identification results over nine indices.

Model	Params (M) ↓	Reg MSE ↓	Reg MAE ↓	Tail AUPRC ↑	Tail AUROC ↑
2015–2019					
Random Walk	–	0.000517	0.016077	N/A	N/A
Autoformer[a]	0.291	0.001616 ± 0.000147	0.029080 ± 0.001447	0.150325 ± 0.007294	0.622095 ± 0.013226
FEDformer[b]	0.520	0.000704 ± 0.000007	0.018713 ± 0.000078	0.094151 ± 0.000297	0.535651 ± 0.002775
DLinear[c]	0.0031	0.000566 ± 0.000005	0.016825 ± 0.000139	0.135604 ± 0.002218	0.606115 ± 0.005576
NLinear[d]	0.0016	0.000541 ± 0.000002	0.016384 ± 0.000037	0.113511 ± 0.002317	0.538672 ± 0.014353
PatchTST[e]	0.426	0.000572 ± 0.000008	0.016728 ± 0.000050	0.109359 ± 0.004910	0.510893 ± 0.019020
TimeMixer[f]	0.120	0.000589 ± 0.000009	0.017010 ± 0.000074	0.107055 ± 0.006765	0.515947 ± 0.007711
TimeXer[g]	8.525	0.000611 ± 0.000005	0.017382 ± 0.000168	0.111392 ± 0.003302	0.545907 ± 0.016795
iTransformer[h]	6.364	0.000654 ± 0.000007	0.017902 ± 0.000083	0.104199 ± 0.003141	0.516728 ± 0.010803
TCDL	**0.179**	$0.000522 \pm 0.000004^{a,b,c,d,e,f,g,h}$	$0.016053 \pm 0.000046^{a,b,c,d,e,f,g,h}$	$0.146371 \pm 0.004936^{b,e,g,h}$	0.593543 ± 0.033934
2021–2025					
Random Walk	–	0.000485	0.015835	N/A	N/A
Autoformer[a]	0.291	0.001303 ± 0.000049	0.027019 ± 0.000380	0.168372 ± 0.005336	0.595198 ± 0.003399
FEDformer[b]	0.520	0.000599 ± 0.000002	0.017684 ± 0.000048	0.095995 ± 0.004919	0.538307 ± 0.016382
DLinear[c]	0.0031	0.000518 ± 0.000011	0.016356 ± 0.000166	0.128354 ± 0.001919	0.567436 ± 0.006717
NLinear[d]	0.0016	0.000491 ± 0.000001	0.015846 ± 0.000019	0.112338 ± 0.005589	0.577811 ± 0.002494
PatchTST[e]	0.426	0.000495 ± 0.000003	0.016004 ± 0.000034	0.120789 ± 0.003861	0.566822 ± 0.017763
TimeMixer[f]	0.120	0.000511 ± 0.000005	0.016311 ± 0.000052	0.108609 ± 0.005957	0.560089 ± 0.013281
TimeXer[g]	8.525	0.000536 ± 0.000005	0.016649 ± 0.000099	0.140428 ± 0.005478	0.600989 ± 0.012841
iTransformer[h]	6.364	0.000593 ± 0.000000	0.017596 ± 0.000040	0.108978 ± 0.005709	0.556655 ± 0.004132
TCDL	**0.179**	$0.000488 \pm 0.000003^{a,b,c,f,g,h}$	$0.015831 \pm 0.000087^{a,b,c,f,g,h}$	$0.132573 \pm 0.006214^{h}$	0.589602 ± 0.012112

Efficiency. TCDL uses only 0.179M parameters, substantially fewer than Transformer-based competitors such as Autoformer, FEDformer, PatchTST, TimeXer, and iTransformer. Despite its compact size, it achieves lower MSE and MAE than all learned baselines across both periods. The compact parameter count indicates that the gains cannot be attributed to increased parameter count alone.

5.2 Loss-Function Comparison and Ablation Study

To isolate the contribution of the joint objective, Table 3 compares TCDL under different loss configurations. Two settings serve as direct ablations: *MSE only* removes both the classification and crossing terms, while *MSE + CE* retains state supervision but removes $\mathcal{L}_{\text{cross}}$. We further compare against alternatives for asymmetric or imbalanced learning, including quantile loss ($q = 0.95$ and 0.975), tail-weighted CE, class-balanced CE, and focal loss ($\gamma = 2$). Two additional controls apply the threshold hinge directly to ground-truth right-tail samples, either without or with CE supervision. Auxiliary-loss weights follow the full-model configuration.

The full objective provides the best overall balance between regression accuracy and right-tail discrimination. The comparison with MSE+CE shows that the improvement cannot be attributed to auxiliary state classification alone; explicitly coupling the predicted state with the continuous return is necessary to obtain a more tail-informative regression output.

Conventional objectives do not reproduce the gains of threshold-consistent learning. Quantile loss emphasizes asymmetric regression errors, tail-weighted CE assigns greater importance to the right-tail class, balanced CE reweights low-frequency states, and focal loss emphasizes difficult classification examples. None matches the full model in right-tail discrimination across both periods.

Ground-truth threshold supervision remains weaker than prediction-conditioned consistency. Applying the threshold hinge directly to true tail samples improves tail discrimination when combined with CE, reaching $0.1252/0.5532$ AUPRC/AUROC in 2015–2019 and $0.1124/0.5848$ in 2021–2025. However, the full model further improves these results to $0.1464/0.5935$ and $0.1326/0.5896$,

respectively. This indicates that directly supervising realized tail samples is not sufficient; the stronger gains arise from enforcing consistency conditioned on the model's own predicted state.

Table 3: Comparison of alternative loss functions over two evaluation periods.

Period	Loss function	Reg MSE ↓	Reg MAE ↓	Tail AUPRC ↑	Tail AUROC ↑
2015–2019	MSE only	**0.000515 ± 0.000001**	**0.015995 ± 0.000016**	0.092308 ± 0.006025	0.505133 ± 0.051547
	MSE + CE	0.000520 ± 0.000004	0.016135 ± 0.000038	0.110601 ± 0.008838	0.470601 ± 0.003145
	MSE + tail-weighted CE	0.000522 ± 0.000001	0.016157 ± 0.000001	0.110037 ± 0.004850	0.461733 ± 0.004048
	MSE + balanced CE	0.000525 ± 0.000008	0.016179 ± 0.000060	0.104716 ± 0.002138	0.471367 ± 0.006101
	MSE + focal loss	0.000525 ± 0.000004	0.016165 ± 0.000020	0.105637 ± 0.009967	0.476527 ± 0.028748
	MSE + quantile loss ($q = 0.95$)	0.000531 ± 0.000003	0.016140 ± 0.000051	0.079209 ± 0.010160	0.441035 ± 0.009379
	MSE + quantile loss ($q = 0.975$)	0.000532 ± 0.000002	0.016152 ± 0.000022	0.078851 ± 0.017329	0.440326 ± 0.019205
	MSE + GT threshold hinge	0.000528 ± 0.000003	0.016077 ± 0.000055	0.092681 ± 0.016889	0.507934 ± 0.005417
	MSE + CE + GT threshold hinge	0.000528 ± 0.000006	0.016035 ± 0.000041	0.125199 ± 0.012731	0.553176 ± 0.021834
	Proposed loss (Full Model)	0.000522 ± 0.000004	0.016053 ± 0.000046	**0.146371 ± 0.004936**	**0.593543 ± 0.033934**
2021–2025	MSE only	**0.000484 ± 0.000001**	**0.015798 ± 0.000030**	0.086380 ± 0.011463	0.511038 ± 0.030533
	MSE + CE	0.000487 ± 0.000001	0.015853 ± 0.000021	0.084206 ± 0.003728	0.459630 ± 0.018478
	MSE + tail-weighted CE	0.000488 ± 0.000002	0.015887 ± 0.000024	0.078838 ± 0.000859	0.449509 ± 0.013832
	MSE + balanced CE	0.000488 ± 0.000003	0.015869 ± 0.000043	0.085808 ± 0.007093	0.460507 ± 0.009374
	MSE + focal loss	0.000488 ± 0.000001	0.015877 ± 0.000017	0.091376 ± 0.011709	0.475693 ± 0.052867
	MSE + quantile loss ($q = 0.95$)	0.000493 ± 0.000001	0.015895 ± 0.000017	0.090633 ± 0.008634	0.509865 ± 0.019038
	MSE + quantile loss ($q = 0.975$)	0.000491 ± 0.000005	0.015860 ± 0.000092	0.090446 ± 0.005347	0.502855 ± 0.003698
	MSE + GT threshold hinge	0.000489 ± 0.000005	0.015822 ± 0.000065	0.103286 ± 0.015247	0.577006 ± 0.013347
	MSE + CE + GT threshold hinge	0.000487 ± 0.000001	0.015807 ± 0.000021	0.112354 ± 0.009043	0.584834 ± 0.016155
	Proposed loss (Full Model)	0.000488 ± 0.000003	0.015831 ± 0.000087	**0.132573 ± 0.006214**	**0.589602 ± 0.012112**

5.3 Hyperparameter Sensitivity

We examine the sensitivity of TCDL to the classification weight λ_{cls}, crossing weight λ_{cross}, and number of ordinary state clusters K, varying one hyperparameter at a time while keeping the others fixed at their default values $(\lambda_{\text{cls}}, \lambda_{\text{cross}}, K) = (0.5, 0.1, 60)$. Fig. 6 summarizes the results.

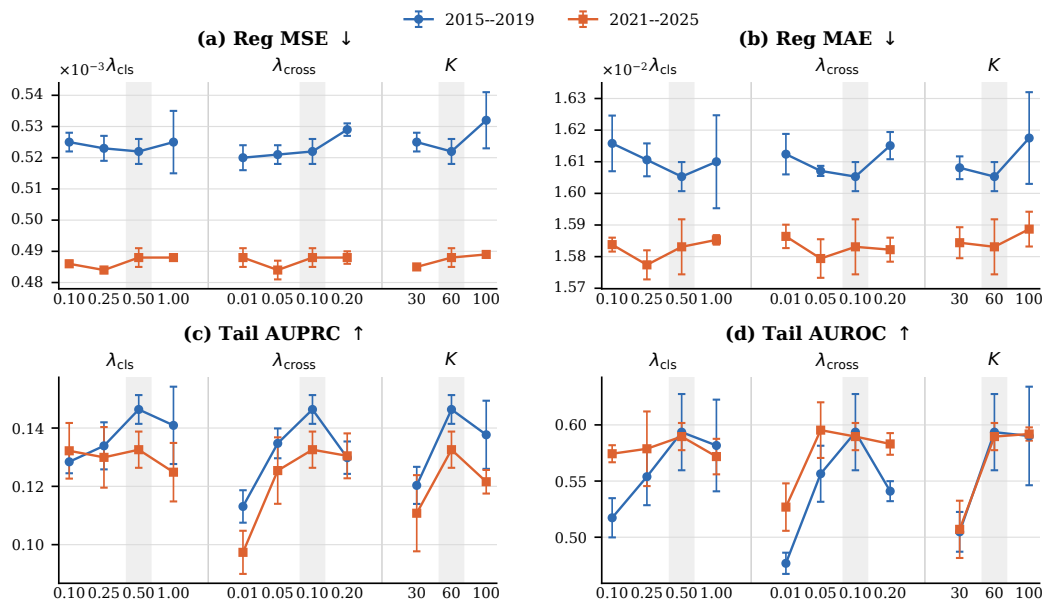

Figure 6: Sensitivity to the loss weights and the number of clusters. Points and error bars denote mean and standard deviation over three seeds. Shaded positions indicate the default setting.

Sensitivity to λ_{cls}. The default configuration provides a stable balance between regression accuracy and right-tail discrimination across both evaluation periods. Increasing λ_{cls} from small values generally improves tail discrimination, with 0.5 yielding the strongest or near-strongest AUPRC and AUROC while maintaining low MSE and MAE. Increasing the weight further to 1.0 provides no additional benefit, suggesting that excessive emphasis on state classification can weaken the shared representation used for regression.

Sensitivity to λ_{cross}. The effect of λ_{cross} is more pronounced. Very small values provide only weak tail discrimination, whereas increasing the weight to 0.1 substantially improves AUPRC in

both periods without materially degrading regression accuracy. Further increasing it to 0.2 does not improve performance and generally reduces the tail metrics. This pattern supports the role of the crossing term as a consistency constraint rather than a dominant optimization objective.

Sensitivity to the number of clusters. The model is moderately robust to the number of clusters. Using $K = 30$ leads to noticeably weaker tail discrimination, while increasing the vocabulary to $K = 100$ does not consistently improve the results. The intermediate setting $K = 60$ achieves the best AUPRC in both periods and remains competitive on the remaining metrics, indicating an effective balance between state granularity and representation stability.

5.4 Representation-Space Analysis

To investigate how the training objective shapes the learned representation, we compare test-set embeddings from MSE-only, MSE+CE, and full TCDL models. We use the 2021–2025 period with seed 1 and the same 2,244 observations for all models, comprising all 748 right-tail samples ($r_t > 1.03$) and 1,496 randomly sampled non-tail observations. For visualization, the 64-dimensional embeddings are first reduced to 30 dimensions by PCA and then projected using t-SNE with identical settings. Because t-SNE is qualitative, separation is additionally measured in the original standardized 64-dimensional space using the Fisher ratio, silhouette coefficient, and 15-NN tail purity.

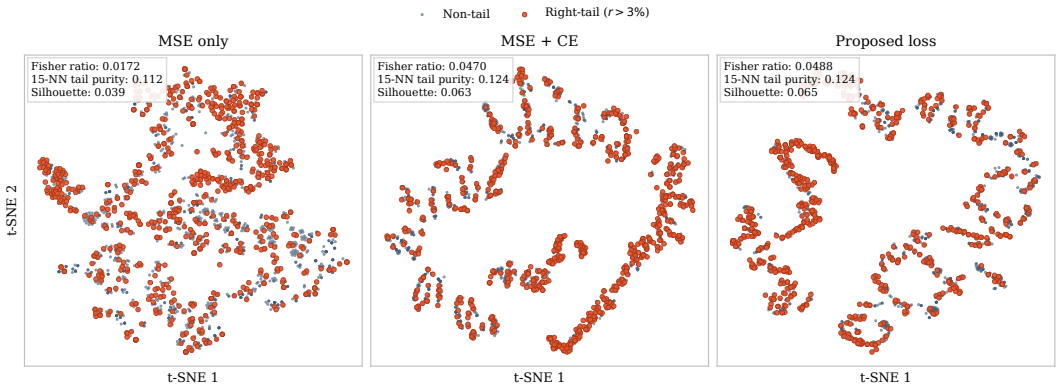

Figure 7: Test-set representation spaces under alternative training objectives.

As shown in Fig. 7, MSE-only training exhibits the weakest structure, with a Fisher ratio of 0.0172 and silhouette coefficient of 0.039. Adding CE substantially increases these values to 0.0470 and 0.063, respectively, while 15-NN tail purity increases from 0.112 to 0.124. This indicates that explicit state supervision is the primary driver of tail-aware representation learning. The full objective further increases the Fisher ratio to 0.0488 and silhouette coefficient to 0.065, while maintaining 15-NN purity of 0.124. Thus, the crossing term does not force right-tail observations into a separate cluster; instead, it provides a modest improvement in global separation while primarily enforcing consistency between the discrete tail state and continuous return prediction. This interpretation is consistent with the stronger tail-discrimination performance observed in the main experiments.

5.5 Trading Strategy Backtest

To evaluate whether the predicted right-tail signals translate into economic value, we construct a validation-calibrated leverage-overlay strategy on top of Buy-and-Hold. For each market and seed, a signal threshold is determined exclusively from the validation predictions as the q-th percentile, with $q \in \{0.90, 0.95, 0.98\}$. During the test period, whenever the predicted return exceeds this threshold, the strategy adds a $0.5\times$ leveraged long overlay for the corresponding non-overlapping four-day holding period. The overlay incorporates a 0.1% round-trip transaction cost and a 5% annual borrowing rate. This design preserves the underlying Buy-and-Hold exposure and uses the model only to selectively increase exposure when a high-confidence positive-return signal is observed.

For a realized four-day return $R_{t,t+4}$ and indicator $I_t = \mathbb{I}(\hat{r}_t > \tau_q)$, the net strategy return is

$$R^{\text{overlay}}_{t,t+4} = R_{t,t+4} + 0.5 I_t \left(R_{t,t+4} - c - \frac{4f}{252} \right), \tag{3}$$

where $c = 0.001$ and $f = 0.05$. Portfolio wealth is updated over non-overlapping four-day blocks. We report cumulative return, annualized Sharpe ratio, and maximum drawdown (MaxDD), with all strategy results averaged over three seeds. Further details of the strategy are provided in Appendix A.

Table 4 shows that the preferred signal threshold varies across periods. Over 2015–2019, Q95 provides the strongest average risk–return trade-off, increasing cumulative return from 0.3029 to 0.3419 and Sharpe ratio from 0.4333 to 0.4603, while leaving MaxDD close to Buy-and-Hold. Q98 yields smaller but more conservative gains, whereas Q90 is less stable, suggesting that overly frequent leverage can dilute high-confidence signals. In 2021–2025, Q90 performs best on average, increasing cumulative return from 0.4555 to 0.5557 and Sharpe ratio from 0.5307 to 0.5783 with comparable drawdown. Overall, the results suggest that the optimal confidence level is market-regime dependent.

Table 4: Market performance of Buy-and-Hold and validation-calibrated leverage-overlay strategies.

Market	Buy-and-Hold			Proposed Q98			Proposed Q95			Proposed Q90		
	Return↑	Sharpe↑	MaxDD↓	Return↑	Sharpe↑	MaxDD↓	Return↑	Sharpe↑	MaxDD↓	Return↑	Sharpe↑	MaxDD↓
2015–2019												
S&P 500	0.5742	0.8012	0.1630	0.5913	0.8128	0.1611	0.6021	**0.8177**	**0.1601**	**0.6177**	0.8114	0.1614
Dow Jones	0.5960	0.8490	0.1617	0.6017	**0.8539**	**0.1587**	**0.6063**	0.8439	0.1680	0.5973	0.8135	0.1633
DAX 30	**0.3621**	**0.4438**	**0.2713**	0.3545	0.4365	0.2768	0.3465	0.4273	0.2828	0.3503	0.4212	0.3010
FTSE 100	0.1519	0.2750	0.1999	0.1549	0.2787	0.1979	**0.1580**	**0.2823**	**0.1970**	0.1458	0.2641	0.2147
CAC 40	0.4198	0.4890	0.2564	0.4357	0.5010	0.2522	0.4984	**0.5389**	**0.2453**	**0.5049**	0.5342	0.2483
Nikkei 225	0.3660	0.4577	0.2540	**0.4304**	**0.5035**	**0.2289**	0.3902	0.4653	0.2537	0.4091	0.4752	0.2727
Hang Seng	0.1870	0.2849	**0.3479**	0.2134	0.3092	**0.3479**	**0.2325**	**0.3256**	0.3505	0.2310	0.3192	0.3763
SSE	-0.0703	0.0631	0.5228	0.0103	0.1408	**0.5058**	**0.0110**	**0.1454**	0.5328	-0.0540	0.1075	0.5693
CSI 300	0.1393	0.2355	0.4625	0.1867	0.2676	**0.4600**	**0.2318**	**0.2964**	0.4627	0.0943	0.2145	0.5325
Average	0.3029	0.4333	0.2933	0.3310	0.4560	**0.2877**	**0.3419**	**0.4603**	0.2948	0.3218	0.4401	0.3155
2021–2025												
S&P 500	0.8503	0.8407	0.2525	0.8671	0.8481	0.2525	0.8886	0.8468	0.2591	**1.0418**	**0.8941**	**0.2307**
Dow Jones	0.5944	0.7152	0.2064	0.5972	0.7175	0.2064	0.6402	0.7415	0.2017	**0.6916**	**0.7678**	**0.1850**
DAX 30	0.7759	0.8099	0.2513	0.8752	0.8512	0.2188	0.9036	0.8654	**0.2187**	**0.9778**	**0.8931**	0.2254
FTSE 100	0.4950	0.7274	**0.1283**	0.4950	0.7274	**0.1283**	0.5299	0.7439	**0.1283**	**0.5622**	**0.7584**	**0.1283**
CAC 40	0.4440	0.5578	0.2169	0.4556	0.5676	0.2137	0.4939	0.5969	**0.2070**	**0.5634**	**0.6330**	0.2148
Nikkei 225	0.8815	0.7483	0.2512	0.9261	**0.7631**	**0.2379**	**0.9516**	0.7571	0.2565	0.9320	0.7234	0.2812
Hang Seng	-0.0351	0.0961	0.5166	-0.0054	0.1255	0.5137	0.0829	0.1955	**0.5087**	**0.1083**	**0.2152**	0.5306
SSE	0.1735	0.2895	0.2663	0.1836	0.3004	0.2663	0.1834	0.2985	**0.2603**	**0.1907**	**0.3008**	0.2705
CSI 300	-0.0800	-0.0085	0.4510	-0.0934	-0.0249	0.4527	-0.0862	-0.0132	**0.4508**	**-0.0663**	**0.0188**	0.4530
Average	0.4555	0.5307	0.2823	0.4779	0.5418	**0.2767**	0.5098	0.5591	0.2768	**0.5557**	**0.5783**	0.2799

Overall, the backtest provides an economic interpretation of the model's tail-aware forecasts. The validation-calibrated overlay improves average return and Sharpe ratio in both evaluation periods without requiring permanent leverage, although performance varies across markets and threshold levels. These results suggest that the predicted right-tail scores may provide useful information for dynamically adjusting exposure beyond unconditional Buy-and-Hold.

6 Conclusion

This work introduced TCDL, a unified framework for jointly modeling continuous return forecasting and threshold-defined rare-event detection. By combining regression, discrete state supervision, and an explicit threshold-consistency objective, TCDL encourages numerical predictions and decision-oriented states to remain logically aligned. Experiments across nine major equity indices and two chronological evaluation periods show that TCDL achieves consistently strong point-forecasting performance while retaining competitive right-tail discrimination with a compact architecture. Ablation and sensitivity analyses further show that the crossing term provides benefits beyond conventional regression, classification, and alternative tail-aware objectives, while representation-space analysis suggests that its primary role is to improve cross-task consistency rather than simply increasing class separability. The downstream backtesting results further indicate that the resulting tail-aware forecasts can provide useful signals for dynamically adjusting market exposure.

Overall, these findings suggest that explicitly coupling continuous predictions with threshold-defined decision states offers a simple and effective learning principle for forecasting settings in which rare outcomes carry disproportionate practical importance. Future work will investigate mixture-of-experts architectures in which individual experts specialize in different decision thresholds and jointly determine the final prediction signal. Such a modular design may avoid overloading a single objective with multiple threshold-specific losses.

AI USE STATEMENT

Generative AI tools were used to assist with figure refinement, plotting code, and language editing, including the visualization of hyperparameter sensitivity results. They were not used to develop the conceptual framework, design the methodology or experiments, implement the proposed method, generate or process datasets, or interpret experimental results. All AI-assisted content was reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide an anonymous repository containing the dataset and a fully executable Jupyter notebook. The notebook includes the complete data-processing pipeline, model implementation, training procedure, evaluation metrics, and detailed code comments. It can be opened and executed directly in Google Colab, allowing the reported experiments to be inspected and reproduced without additional local environment configuration. This repository contains unpublished work provided solely for peer review and reproducibility assessment; please do not redistribute or disseminate its contents.

REFERENCES

Milad Kamali Alamdari, Akbar Esfahanipour, and Hossein Dastkhan. A portfolio trading system using a novel pixel graph network for stock selection and a mean-cdar optimization for portfolio rebalancing. *Applied Soft Computing*, 152:111213, 2024.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler-Canseco, and Artur Dubrawski. Nhits: neural hierarchical interpolation for time series forecasting. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i6.25854. URL https://doi.org/10.1609/aaai.v37i6.25854.

Junde Chen, Yuxin Wen, YA Nanehkaran, MD Suzauddola, Weirong Chen, and Defu Zhang. Machine learning techniques for stock price prediction and graphic signal recognition. *Engineering Applications of Artificial Intelligence*, 121:106038, 2023.

Lucas Coelho e Silva, Gustavo de Freitas Fonseca, and Paulo Andre L. Castro. Transformers and attention-based networks in quantitative trading: a comprehensive survey. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, pp. 822–830, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400710810.

Albert King and Yaojie Zhang. Akshare. https://github.com/akfamily/akshare, 2022.

Roger Koenker and Gilbert Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1913643.

Chrysoula Kosma, Giannis Nikolentzos, Nancy Xu, and Michalis Vazirgiannis. Time series forecasting models copy the past: How to mitigate. In Elias Pimenidis, Plamen Angelov, Chrisina Jayne, Antonios Papaleonidas, and Mehmet Aydin (eds.), *Artificial Neural Networks and Machine Learning – ICANN 2022*, pp. 366–378, Cham, 2022. Springer International Publishing. ISBN 978-3-031-15919-0.

Si Woon Lee and Ha Young Kim. Stock market forecasting with super-high dimensional time-series data using convlstm, trend sampling, and specialized data augmentation. *Expert Systems with Applications*, 161:113704, 2020.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. doi: 10.1109/TPAMI.2018.2858826.

Pengfei Liu, Shuxian Shang, Changguang Song, Fang Li, Yuhan Guo, Yuetong Chang, Han Feng, Guangming Xia, Biwei Wu, Jiubin Tan, and Weibo Wang. Tcd: Towards consistent and unified single stage object detection. *Neurocomputing*, 671:132720, 2026. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2026.132720. URL https://www.sciencedirect.com/science/article/pii/S0925231226001177.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Xiang Ma, Taihua Chen, Pengcheng Wang, Xuemei Li, and Caiming Zhang. Recast: Reliability-aware codebook-assisted lightweight time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24299–24307, 2026.

Richard A. Meese and Kenneth Rogoff. Empirical exchange rate models of the seventies: Do they fit out of sample? *Journal of International Economics*, 14(1):3–24, 1983. ISSN 0022-1996. doi: https://doi.org/10.1016/0022-1996(83)90017-X. URL https://www.sciencedirect.com/science/article/pii/002219968390017X.

Raha Moraffah, Paras Sheth, Mansooreh Karami, Anchit Bhattacharya, Qianru Wang, Anique Tahir, Adrienne Raglin, and Huan Liu. Causal inference for time series analysis: problems, methods and evaluation. *Knowl. Inf. Syst.*, 63(12):3041–3085, December 2021. ISSN 0219-1377. doi: 10.1007/s10115-021-01621-0. URL https://doi.org/10.1007/s10115-021-01621-0.

Lauri Nevasalmi. Forecasting multinomial stock returns using machine learning methods. *The Journal of Finance and Data Science*, 6:86–106, 2020. ISSN 2405-9188. doi: https://doi.org/10.1016/j.jfds.2020.09.001. URL https://www.sciencedirect.com/science/article/pii/S2405918820300143.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The 11th International Conference on Learning Representations*, 2023.

Kin G. Olivares, Cristian Challú, Azul Garza, Max Mergenthaler Canseco, and Artur Dubrawski. NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022. URL https://github.com/Nixtla/neuralforecast.

Hyun Jun Park, Youngjun Kim, and Ha Young Kim. Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. *Applied Soft Computing*, 114:108106, 2022. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2021.108106. URL https://www.sciencedirect.com/science/article/pii/S1568494621009947.

Omer Berat Sezer and Ahmet Murat Ozbayoglu. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70:525–538, 2018.

Evangelos Spiliotis, Andreas Kouloumos, Vassilios Assimakopoulos, and Spyros Makridakis. Are forecasting competitions data representative of the reality? *International Journal of Forecasting*, 36(1):37–53, 2020. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2018.12.007. URL https://www.sciencedirect.com/science/article/pii/S0169207019300159. M4 Competition.

Yanru Sun, Zongxia Xie, Dongyue Chen, Emadeldeen Eldele, and Qinghua Hu. Hierarchical classification auxiliary network for time series forecasting. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i19.34286. URL https://doi.org/10.1609/aaai.v39i19.34286.

Ankit Thakkar and Kinjal Chaudhari. A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. *Expert Systems with Applications*, 177:114800, 2021.

Tuna Tuncer, Uygar Kaya, Emre Sefer, Onur Alacam, and Tugcan Hoser. Asset price and direction prediction via deep 2d transformer and convolutional neural networks. In *Proceedings of the Third ACM International Conference on AI in Finance*, pp. 79–86, 2022.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2024a.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Yunzhong Qiu, Haoran Zhang, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 2024b.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: a survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23, 2023. ISBN 978-1-956792-03-4. doi: 10.24963/ijcai.2023/759. URL https://doi.org/10.24963/ijcai.2023/759.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Yuanjian Xu, Jianing Hao, Anxian Liu, Zhenzhuo Li, Shichang Meng, Shuai Yuan, and Guang Zhang. Lens: Large pre-trained transformer for exploring financial time series regularities. In *Proceedings of the 6th ACM International Conference on AI in Finance*, ICAIF '25, pp. 771–778, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400722202. doi: 10.1145/3768292.3770349. URL https://doi.org/10.1145/3768292.3770349.

Annan Yu, Danielle C. Maddix, Boran Han, Xiyuan Zhang, Abdul Fatir Ansari, Oleksandr Shchur, Christos Faloutsos, Andrew Gordon Wilson, Michael W. Mahoney, and Bernie Wang. Understanding transformers for time series: Rank structure, flow-of-ranks, and compressibility. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=axR2KZwaD3.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0.

Liangwei Nathan Zheng, Chang Dong, Wei Emma Zhang, Lin Yue, Miao Xu, Olaf Maennel, and Weitong Chen. Understanding why large language models can be ineffective in time series analysis: The impact of modality alignment. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 4026–4037, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737169. URL https://doi.org/10.1145/3711896.3737169.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pp. 27268–27286. PMLR, 2022.

Zhuohang Zhu, Haodong Chen, Qiang Qu, and Vera Chung. Fincast: A foundation model for financial time-series forecasting. *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 2025. URL https://api.semanticscholar.org/CorpusID:280918721.

A Appendix

A.1 Data Description and Preprocessing

Table 5 reports the number of valid windows and the corresponding right-tail rate for every market and temporal partition. The different sample counts arise from market-specific trading calendars and data availability.

Table 5: Numbers of valid samples and right-tail event rates for the nine indices. The right tail is defined by $C_{t+4}/C_t > 1.03$. Each cell reports N (tail rate).

Split 1: 1999–2019

Market	Training 1999–2008	Validation 2009–2014	Test 2015–2019
S&P 500	2,520 (6.90%)	1,512 (7.21%)	1,255 (2.63%)
Dow Jones	2,522 (6.50%)	1,510 (5.63%)	1,254 (2.63%)
DAX 30	2,556 (12.60%)	1,530 (12.55%)	1,259 (8.10%)
FTSE 100	2,527 (6.45%)	1,515 (7.52%)	1,261 (4.04%)
CAC 40	2,552 (10.27%)	1,536 (11.65%)	1,275 (6.20%)
Nikkei 225	2,459 (12.53%)	1,473 (13.78%)	1,218 (7.64%)
Hang Seng	2,464 (14.89%)	1,481 (11.34%)	1,226 (7.99%)
SSE	2,410 (16.02%)	1,456 (14.15%)	1,216 (11.35%)
CSI 300	944 (25.00%)	1,456 (17.58%)	1,216 (12.34%)
Pooled	**20,954 (11.37%)**	**13,469 (11.23%)**	**11,180 (6.95%)**

Split 2: 2005–2025

Market	Training 2005–2014	Validation 2015–2020	Test 2021–2025
S&P 500	2,521 (6.07%)	1,512 (4.30%)	1,254 (6.46%)
Dow Jones	2,524 (4.91%)	1,511 (5.16%)	1,254 (5.02%)
DAX 30	2,548 (10.36%)	1,517 (9.49%)	1,273 (6.52%)
FTSE 100	2,526 (6.85%)	1,519 (5.53%)	1,260 (1.98%)
CAC 40	2,559 (9.54%)	1,536 (7.88%)	1,280 (5.23%)
Nikkei 225	2,456 (12.38%)	1,463 (9.16%)	1,222 (11.21%)
Hang Seng	2,466 (11.39%)	1,478 (8.46%)	1,227 (14.10%)
SSE	2,427 (17.31%)	1,462 (11.01%)	1,211 (3.80%)
CSI 300	2,400 (20.50%)	1,462 (13.00%)	1,211 (6.03%)
Pooled	**22,427 (10.95%)**	**13,460 (8.19%)**	**11,192 (6.68%)**

Input and target normalization. The preprocessing pipeline applies separate transformations to the input observations and the continuous forecasting target.

First, the OHLC observations of each sample are normalized feature-wise relative to their values at the forecast-origin date t. For $f \in \{O, H, L, C\}$, we define

$$\widetilde{x}_{u,f} = \frac{x_{u,f}}{x_{t,f}}, \qquad u = t-23, \ldots, t. \tag{4}$$

This transformation sets every input feature to one at the forecast origin and expresses the preceding observations as relative price levels. It reduces differences in absolute price scales across markets while preserving feature-specific temporal variation.

The continuous prediction target is the four-day-ahead closing-price ratio

$$r_t = \frac{C_{t+4}}{C_t}. \tag{5}$$

For model training, the excess return ratio $r_t - 1$ is standardized using statistics computed exclusively from the training partition:

$$z_t = \frac{(r_t - 1) - \mu_{\text{train}}}{\sigma_{\text{train}}}, \tag{6}$$

where

$$\mu_{\text{train}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} (r_i - 1), \qquad \sigma_{\text{train}} = \text{Std}_{i \in \mathcal{D}_{\text{train}}}(r_i - 1). \tag{7}$$

The regression head predicts z_t, which is transformed back to the original return-ratio scale during evaluation:

$$\widehat{r}_t = 1 + \sigma_{\text{train}} \widehat{z}_t + \mu_{\text{train}}. \tag{8}$$

Scalar token compression. After input normalization, each flattened sliding segment $\mathbf{u}_{t,j} \in \mathbb{R}^{d_w}$ is compressed into a scalar using its dimension-normalized Euclidean magnitude:

$$d_{t,j} = \frac{\|\mathbf{u}_{t,j}\|_2}{\sqrt{d_w}}, \tag{9}$$

where d_w is the dimensionality of the sliding segment. Dividing by $\sqrt{d_w}$ makes the scalar magnitude less dependent on the segment dimension. The resulting scalars are assigned to discrete states by K-means. All target-standardization statistics and K-means cluster centers are estimated exclusively from the training partition. They are fixed before being applied to the validation and test partitions.

A.2 Evaluation Metrics

We evaluate the proposed model from three complementary perspectives: point-forecast accuracy, right-tail discrimination, and economic performance. Unless otherwise stated, all predictive metrics are first computed after pooling the observations from the nine common indices for each random seed. The reported mean and standard deviation are then calculated across the three seed-level results.

Point-forecast accuracy. Let $r_i = C_{i+4}/C_i$ denote the realized four-day return ratio and $\widehat{r}_i$ the corresponding continuous prediction from the regression head. We evaluate global forecasting accuracy using MSE and MAE:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(r_i - \widehat{r}_i\right)^2, \tag{10}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|r_i - \widehat{r}_i\right|. \tag{11}$$

Lower values indicate more accurate continuous return forecasts. The Random Walk baseline predicts

$$\widehat{r}_i^{\mathrm{RW}} = 1 \tag{12}$$

for every sample, corresponding to an unchanged closing price over the four-day forecasting horizon.

Right-tail event definition. A right-tail event is defined as a four-day return exceeding 3%:

$$y_i^{\mathrm{tail}} = \mathbb{I}(r_i > 1.03). \tag{13}$$

For the primary results, the continuous regression prediction $\widehat{r}_i$ is used as the right-tail ranking score. A larger predicted return indicates stronger evidence that the sample belongs to the right tail.

Area under the precision–recall curve. Tail AUPRC evaluates the ability of the ranking score to identify rare right-tail observations. For a decision threshold γ, precision and recall are

$$\mathrm{Precision}(\gamma) = \frac{\mathrm{TP}(\gamma)}{\mathrm{TP}(\gamma) + \mathrm{FP}(\gamma)}, \tag{14}$$

$$\mathrm{Recall}(\gamma) = \frac{\mathrm{TP}(\gamma)}{\mathrm{TP}(\gamma) + \mathrm{FN}(\gamma)}. \tag{15}$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively. Tail AUPRC summarizes the precision–recall curve obtained by varying γ. In implementation, we use average precision:

$$\mathrm{AUPRC} = \sum_{k}\left(R_k - R_{k-1}\right)P_k, \tag{16}$$

where P_k and R_k denote precision and recall at the k-th ranking threshold. AUPRC is particularly informative under class imbalance because it directly reflects the precision obtained when retrieving rare positive events. For an uninformative constant score, AUPRC equals the empirical right-tail event rate.

Area under the receiver operating characteristic curve. Tail AUROC measures the probability that a randomly selected right-tail observation receives a higher prediction score than a randomly selected non-tail observation:

$$\mathrm{AUROC} = \Pr\left(\widehat{r}_i > \widehat{r}_j \mid y_i^{\mathrm{tail}} = 1,\, y_j^{\mathrm{tail}} = 0\right), \tag{17}$$

with ties assigned half credit. An AUROC of 0.5 corresponds to random ranking, whereas a value of 1 indicates perfect discrimination. We report them as N/A in the main comparison because Random Walk provides no non-trivial ranking information for right-tail discrimination.

Backtesting metrics. The trading strategy is evaluated using non-overlapping four-day holding periods, consistent with the forecasting horizon. Let $R_j^{(4)}$ denote the realized portfolio return in the j-th four-day period, where $j = 1, \ldots, J$. The cumulative return is computed as

$$R_{\text{cum}} = \prod_{j=1}^{J} \left(1 + R_j^{(4)}\right) - 1. \tag{18}$$

The annualized Sharpe ratio is defined as

$$\text{Sharpe} = \sqrt{\frac{252}{4}} \, \frac{\overline{R}^{(4)}}{s\big(R^{(4)}\big)}, \tag{19}$$

where $\overline{R}^{(4)}$ and $s(R^{(4)})$ are respectively the mean and standard deviation of the non-overlapping four-day portfolio returns. The risk-free rate is assumed to be zero.

Let

$$V_j = \prod_{k=1}^{j} \left(1 + R_k^{(4)}\right) \tag{20}$$

denote the cumulative portfolio value after the j-th holding period. The maximum drawdown is calculated as

$$\text{MaxDD} = \max_{1 \le j \le J} \left(1 - \frac{V_j}{\max_{1 \le k \le j} V_k}\right). \tag{21}$$

A lower maximum drawdown indicates better downside-risk control.

A.3 Implementation Details

Model configuration. TCDL uses an embedding dimension of 64 and four encoder blocks. Each block contains residual feature mixing, layer normalization, and a feed-forward network with hidden dimension $4d_{\text{model}}$. Sinusoidal positional encodings are added to the token embeddings. The encoded token representations are mean-pooled and passed to two readout heads: a regression head for the standardized continuous return and a classification head over $K + 1 = 61$ states. The additional state corresponds to the explicit right-tail event $r_t > 1.03$. Dropout is set to 0.1 throughout the model.

Optimization. The model is optimized using AdamW with a learning rate of 10^{-3}, weight decay of 10^{-4}, and a mini-batch size of 64. The gradient norm is clipped at 1.0. Training is conducted for at most 500 epochs, with early stopping after 30 consecutive epochs without improvement. Model selection is based exclusively on the total validation loss, and the selected checkpoint is evaluated once on the test set. The loss coefficients used in the main experiments are

$$\lambda_{\text{reg}} = 1, \qquad \lambda_{\text{cls}} = 0.5, \qquad \lambda_{\text{cross}} = 0.1. \tag{22}$$

The right-tail threshold is fixed at $\tau = 1.03$. These hyperparameters are held constant across both temporal splits and all nine markets.

Reproducibility. All reported neural-model results are obtained from three independent runs with random seeds 1, 42, and 1337. For each run, the random-number generators of Python, NumPy, and PyTorch are initialized using the same seed. K-means initialization is held fixed so that variation across runs reflects neural-model training rather than changes in the tokenization vocabulary. Training is performed on a GPU when available, while the implementation also supports CPU execution. For complete reproducibility details and access to the accompanying code and data, please refer to the Reproducibility Statement.

A.4 Backtesting Algorithm

The detailed procedure used for the validation-calibrated leverage-overlay backtest is summarized in Algorithm 1. The algorithm determines the trading threshold exclusively from validation-set predictions and applies the resulting signal to non-overlapping four-day test periods, while accounting for transaction costs and borrowing costs.

Algorithm 1 Validation-Calibrated Leverage-Overlay Strategy

Require: Validation predictions $\{\hat{r}_i^{\mathrm{val}}\}_{i=1}^{N_{\mathrm{val}}}$; test predictions $\{\hat{r}_t\}_{t=1}^{T}$; four-day realized returns $R_{t,t+4} = C_{t+4}/C_t - 1$; quantile level $q \in \{0.90, 0.95, 0.98\}$; overlay leverage $\lambda_{\mathrm{lev}} = 0.5$; round-trip transaction cost $c = 0.001$; annual borrowing rate $f = 0.05$; annual trading days $A = 252$; forecast horizon $H = 4$

Ensure: Portfolio wealth sequence $\{P_t^{\mathrm{overlay}}\}$

1: Determine the validation-calibrated signal threshold:

$$\tau_q = \mathrm{Quantile}_q\left(\{\hat{r}_i^{\mathrm{val}}\}_{i=1}^{N_{\mathrm{val}}}\right)$$

2: Initialize $P_0^{\mathrm{overlay}} \leftarrow 1$

3: Compute the horizon financing cost:

$$f_H \leftarrow \frac{fH}{A}$$

4: **for** each non-overlapping four-day test block t **do**

5: Generate the high-confidence signal:

$$I_t = \mathbb{I}\left(\hat{r}_t > \tau_q\right)$$

6: Compute the net four-day portfolio return:

$$R_{t,t+4}^{\mathrm{overlay}} = R_{t,t+4} + \lambda_{\mathrm{lev}} I_t \left(R_{t,t+4} - c - f_H\right)$$

7: Update portfolio wealth:

$$P_{t+4}^{\mathrm{overlay}} = P_t^{\mathrm{overlay}}\left(1 + R_{t,t+4}^{\mathrm{overlay}}\right)$$

8: **end for**

9: **return** $\{P_t^{\mathrm{overlay}}\}$

The Buy-and-Hold benchmark is evaluated over the same non-overlapping four-day intervals as the proposed strategy. Its cumulative return is equivalent to continuously holding the index over the corresponding backtest endpoints, whereas its Sharpe ratio and maximum drawdown are computed from the four-day portfolio-value sequence.

Algorithm 2 Four-Day Buy-and-Hold Benchmark

Require: Test-period close-price sequence $\{C_t\}_{t=0}^{T}$ and forecast horizon $H = 4$

Ensure: Buy-and-Hold wealth sequence $\{P_j^{\mathrm{BH}}\}_{j=0}^{J}$ evaluated at four-day endpoints

1: Set the first eligible forecast origin to t_0

2: Construct non-overlapping holding-period origins

$$t_j = t_0 + jH, \qquad j = 0, \ldots, J-1,$$

where $t_j + H \le T$

3: Initialize $P_0^{\mathrm{BH}} \leftarrow 1$

4: **for** $j = 0, \ldots, J-1$ **do**

5: Compute the realized H-day market return

$$R_j^{\mathrm{BH}} = \frac{C_{t_j+H}}{C_{t_j}} - 1$$

6: Update the Buy-and-Hold wealth

$$P_{j+1}^{\mathrm{BH}} = P_j^{\mathrm{BH}}\left(1 + R_j^{\mathrm{BH}}\right)$$

7: **end for**

8: **return** $\{P_j^{\mathrm{BH}}\}_{j=0}^{J}$

A.5 Complete Hyperparameter Sensitivity Results

Table 6: Sensitivity to the loss weights and the number of ordinary state clusters. Results are mean ± standard deviation over three seeds. Only one hyperparameter is varied at a time, and bold indicates the best result within each period and sensitivity group.

Group	Period	Value	Reg MSE ↓	Reg MAE ↓	Tail AUPRC ↑	Tail AUROC ↑
λ_{cls}	2015–2019	0.10	0.000525 ± 0.000003	0.016158 ± 0.000088	0.128414 ± 0.003923	0.517308 ± 0.017466
		0.25	0.000523 ± 0.000004	0.016106 ± 0.000052	0.133896 ± 0.008073	0.553894 ± 0.025383
		0.50	$\mathbf{0.000522 \pm 0.000004}$	$\mathbf{0.016053 \pm 0.000046}$	$\mathbf{0.146371 \pm 0.004936}$	$\mathbf{0.593543 \pm 0.033934}$
		1.00	0.000525 ± 0.000010	0.016100 ± 0.000147	0.140921 ± 0.013264	0.581680 ± 0.040810
	2021–2025	0.10	0.000486 ± 0.000001	0.015838 ± 0.000022	0.132197 ± 0.009540	0.574322 ± 0.007592
		0.25	$\mathbf{0.000484 \pm 0.000001}$	$\mathbf{0.015774 \pm 0.000046}$	0.129944 ± 0.010399	0.578747 ± 0.033216
		0.50	0.000488 ± 0.000003	0.015831 ± 0.000087	$\mathbf{0.132573 \pm 0.006214}$	$\mathbf{0.589602 \pm 0.012112}$
		1.00	0.000488 ± 0.000001	0.015853 ± 0.000015	0.124855 ± 0.010034	0.571798 ± 0.015802
λ_{cross}	2015–2019	0.01	$\mathbf{0.000520 \pm 0.000004}$	0.016124 ± 0.000064	0.113098 ± 0.005565	0.476779 ± 0.009447
		0.05	0.000521 ± 0.000003	0.016071 ± 0.000016	0.134748 ± 0.005114	0.556499 ± 0.024955
		0.10	0.000522 ± 0.000004	$\mathbf{0.016053 \pm 0.000046}$	$\mathbf{0.146371 \pm 0.004936}$	$\mathbf{0.593543 \pm 0.033934}$
		0.20	0.000529 ± 0.000002	0.016151 ± 0.000043	0.129834 ± 0.005558	0.540997 ± 0.008897
	2021–2025	0.01	0.000488 ± 0.000003	0.015864 ± 0.000037	0.097360 ± 0.007400	0.526826 ± 0.021113
		0.05	$\mathbf{0.000484 \pm 0.000003}$	$\mathbf{0.015794 \pm 0.000061}$	0.125408 ± 0.011399	$\mathbf{0.595343 \pm 0.024777}$
		0.10	0.000488 ± 0.000003	0.015831 ± 0.000087	$\mathbf{0.132573 \pm 0.006214}$	0.589602 ± 0.012112
		0.20	0.000488 ± 0.000002	0.015822 ± 0.000038	0.130462 ± 0.007702	0.583048 ± 0.009643
K	2015–2019	30	0.000525 ± 0.000003	0.016081 ± 0.000036	0.120331 ± 0.006361	0.504804 ± 0.017707
		60	$\mathbf{0.000522 \pm 0.000004}$	$\mathbf{0.016053 \pm 0.000046}$	$\mathbf{0.146371 \pm 0.004936}$	$\mathbf{0.593543 \pm 0.033934}$
		100	0.000532 ± 0.000009	0.016175 ± 0.000145	0.137687 ± 0.011688	0.590147 ± 0.043869
	2021–2025	30	$\mathbf{0.000485 \pm 0.000000}$	0.015844 ± 0.000049	0.110780 ± 0.013065	0.507011 ± 0.025497
		60	0.000488 ± 0.000003	$\mathbf{0.015831 \pm 0.000087}$	$\mathbf{0.132573 \pm 0.006214}$	0.589602 ± 0.012112
		100	0.000489 ± 0.000001	0.015887 ± 0.000055	0.121576 ± 0.004046	$\mathbf{0.591891 \pm 0.005886}$

The default setting is $(\lambda_{\text{cls}}, \lambda_{\text{cross}}, K) = (0.5, 0.1, 60)$. For each seed, observations from the nine common indices are pooled before evaluation. The pooled test-set sizes are 11,180 and 11,192 observations per seed for 2015–2019 and 2021–2025, respectively.

A.6 Details about Baselines

Each baseline receives 24 trading days of OHLC observations, serialized in chronological day order, and predicts the close price on the fourth future trading day. All models are optimized using MAE loss, a learning rate of 10^{-4}, a batch size of 32, and training and inference window-batch sizes of 1024. Validation is performed every 100 optimization steps, and training is terminated after three consecutive validation checks without improvement. Identity scaling is used, and experiments are repeated with random seeds $\{1, 42, 1337\}$.

Table 7: Architecture and training configurations of the baseline models.

Model	Width	Enc.	Dec.	Heads	FFN/Conv.	Patch	Decomposition or normalization	Dropout	Steps	Params (M)
DLinear	–	–	–	–	–	–	MA(25)	0	5000	0.0031
NLinear	–	–	–	–	–	–	Last-value normalization	0	5000	0.0016
Autoformer	128	2	1	4	32	–	MA(25), factor = 3	0.05	1000	0.291
FEDformer	128	2	1	8	32	–	Fourier, 64 random modes, MA(25)	0.05	1000	0.520
PatchTST	128	3	–	16	256	16/8	RevIN, residual attention	0.20	1000	0.426
TimeMixer	32	4	–	–	32	–	MA(25), AvgPool(2), top-$k = 5$	0.10	1000	0.120
TimeXer	512	2	–	8	2048	16/–	Normalization	0.10	1000	8.525
iTransformer	512	2	1	8	2048	–	Normalization	0.10	1000	6.364

Notes. Width denotes the principal hidden or embedding dimension. Enc. and Dec. denote the numbers of encoder and decoder layers, respectively. FFN/Conv. denotes the feed-forward dimension for PatchTST, TimeMixer, TimeXer, and iTransformer, and the convolutional hidden dimension for Autoformer and FEDformer. Patch is reported as patch length/stride. MA(w) denotes moving-average decomposition with window size w. TimeMixer uses one average-pooling down-sampling layer with window size 2. PatchTST additionally uses fully connected dropout 0.2, while its attention and output-head dropout rates are zero.

A.7 Loss Functions Used in the Objective Comparison

This subsection specifies the objectives used in Table 3. All configurations share the same tokenizer, model backbone, regression head, and optimization settings; only the training objective is changed.

Let B denote the mini-batch size, z_i the standardized continuous-return target, and $\hat{z}_i$ the output of the regression head. Let $s_i \in \{0, \dots, K\}$ be the prediction-state label, where class K is the explicit right-tail state defined by $r_i > 1.03$. The classification head produces logits $\boldsymbol{\ell}_i$ and class probabilities

$$p_{i,k} = \frac{\exp(\ell_{i,k})}{\sum_{j=0}^{K} \exp(\ell_{i,j})}. \tag{23}$$

Mean-squared regression loss. All configurations contain the same regression objective,

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{B} \sum_{i=1}^{B} (\hat{z}_i - z_i)^2 . \tag{24}$$

The *MSE only* configuration removes all auxiliary supervision:

$$\mathcal{L}_{\mathrm{MSE\ only}} = \mathcal{L}_{\mathrm{MSE}}. \tag{25}$$

Standard cross-entropy. The ordinary multiclass cross-entropy loss is

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{B} \sum_{i=1}^{B} \log p_{i,s_i}. \tag{26}$$

The *MSE + CE* configuration retains state supervision but removes threshold-crossing consistency:

$$\mathcal{L}_{\mathrm{MSE+CE}} = \mathcal{L}_{\mathrm{MSE}} + 0.5\mathcal{L}_{\mathrm{CE}}. \tag{27}$$

Tail-weighted cross-entropy. To place greater emphasis on the explicit right-tail state, we define class weights

$$w_k^{\mathrm{tail}} = \begin{cases} 2, & k = K, \\ 1, & k \neq K. \end{cases} \tag{28}$$

The corresponding weighted cross-entropy is

$$\mathcal{L}_{\mathrm{TWCE}} = -\frac{\sum_{i=1}^{B} w_{s_i}^{\mathrm{tail}} \log p_{i,s_i}}{\sum_{i=1}^{B} w_{s_i}^{\mathrm{tail}}}. \tag{29}$$

The resulting training objective is

$$\mathcal{L}_{\mathrm{MSE+TWCE}} = \mathcal{L}_{\mathrm{MSE}} + 0.5\mathcal{L}_{\mathrm{TWCE}}. \tag{30}$$

Class-balanced cross-entropy. As a general alternative to weighting only the right-tail state, we assign inverse-frequency weights to all $K+1$ classes. Let n_k denote the number of training samples assigned to class k, and let N_{tr} denote the total number of training samples. The unnormalized class weight is

$$\widetilde{w}_k^{\mathrm{bal}} = \frac{N_{\mathrm{tr}}}{(K+1)\max(n_k, 1)}. \tag{31}$$

The weights are normalized to have unit mean:

$$w_k^{\mathrm{bal}} = \frac{\widetilde{w}_k^{\mathrm{bal}}}{\frac{1}{K+1}\sum_{j=0}^{K} \widetilde{w}_j^{\mathrm{bal}}}. \tag{32}$$

The balanced cross-entropy loss is

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{\sum_{i=1}^{B} w_{s_i}^{\mathrm{bal}} \log p_{i,s_i}}{\sum_{i=1}^{B} w_{s_i}^{\mathrm{bal}}}, \tag{33}$$

and the complete objective is

$$\mathcal{L}_{\mathrm{MSE+BCE}} = \mathcal{L}_{\mathrm{MSE}} + 0.5\mathcal{L}_{\mathrm{BCE}}. \tag{34}$$

Focal loss. The multiclass focal loss reduces the contribution of samples that are already classified with high confidence. For p_{i,s_i}, the probability assigned to the true state, it is defined as

$$\mathcal{L}_{\text{focal}} = -\frac{1}{B}\sum_{i=1}^{B}(1-p_{i,s_i})^{\gamma}\log p_{i,s_i}, \qquad \gamma = 2. \tag{35}$$

No additional class-dependent α weight is used. The complete objective is

$$\mathcal{L}_{\text{MSE+focal}} = \mathcal{L}_{\text{MSE}} + 0.5\mathcal{L}_{\text{focal}}. \tag{36}$$

Quantile loss. For a quantile level $q \in (0,1)$, the pinball loss is

$$\mathcal{L}_q = \frac{1}{B}\sum_{i=1}^{B}\rho_q(z_i - \hat{z}_i), \tag{37}$$

where

$$\rho_q(u) = \begin{cases} qu, & u \geq 0, \\ (q-1)u, & u < 0. \end{cases} \tag{38}$$

Equivalently,

$$\rho_q(u) = \max\{qu, (q-1)u\}. \tag{39}$$

We evaluate $q = 0.95$ and $q = 0.975$. In both cases, the pinball loss and MSE are applied to the same scalar regression output:

$$\mathcal{L}_{\text{MSE+Quantile}(q)} = \mathcal{L}_{\text{MSE}} + 0.5\mathcal{L}_q, \qquad q \in \{0.95, 0.975\}. \tag{40}$$

Thus, the comparison isolates the effect of asymmetric regression supervision without changing the model architecture.

Ground-truth threshold hinge. As an additional technical control, we apply the threshold hinge directly to samples whose ground-truth returns belong to the right tail. Let

$$\mathcal{B}_{\text{GT}} = \{i : r_i > \tau\} = \{i : s_i = K\}. \tag{41}$$

The ground-truth hinge loss is

$$\mathcal{L}_{\text{GT-hinge}} = \begin{cases} \dfrac{1}{|\mathcal{B}_{\text{GT}}|}\displaystyle\sum_{i\in\mathcal{B}_{\text{GT}}}\left[\max(0, \tau_z - \widehat{z}_i)\right]^2, & |\mathcal{B}_{\text{GT}}| > 0, \\ 0, & |\mathcal{B}_{\text{GT}}| = 0. \end{cases} \tag{42}$$

We consider two corresponding objectives:

$$\mathcal{L}_{\text{MSE+GT}} = \mathcal{L}_{\text{MSE}} + 0.1\mathcal{L}_{\text{GT-hinge}}, \tag{43}$$

and

$$\mathcal{L}_{\text{MSE+CE+GT}} = \mathcal{L}_{\text{MSE}} + 0.5\mathcal{L}_{\text{CE}} + 0.1\mathcal{L}_{\text{GT-hinge}}. \tag{44}$$

The first removes classification supervision, while the second retains the same CE supervision as the full model but replaces the prediction-conditioned crossing term with a ground-truth-conditioned hinge.

A.8 Representation-Space Analysis

We additionally quantify class separation in the standardized original 64-dimensional representation. For each model, embedding dimension d is standardized as

$$\widetilde{h}_{i,d} = \frac{h_{i,d} - \mu_d}{\sigma_d}, \tag{45}$$

where μ_d and σ_d are computed over the complete test-set embeddings of that model.

Fisher ratio. Let $\mathcal{T}$ and $\mathcal{N}$ denote the sets of right-tail and non-tail observations, respectively, and let $\boldsymbol{\mu}_{\mathcal{T}}$ and $\boldsymbol{\mu}_{\mathcal{N}}$ be their mean standardized embeddings. We define the between-class scatter as

$$S_{\text{between}} = \|\boldsymbol{\mu}_{\mathcal{T}} - \boldsymbol{\mu}_{\mathcal{N}}\|_2^2 . \tag{46}$$

The within-class scatter is

$$S_{\text{within}} = \sum_{d=1}^{64} \left[\text{Var}_{i\in\mathcal{T}}(\widetilde{h}_{i,d}) + \text{Var}_{i\in\mathcal{N}}(\widetilde{h}_{i,d}) \right], \tag{47}$$

where the variances are calculated using the sample-variance definition. The Fisher ratio is then

$$\text{Fisher} = \frac{S_{\text{between}}}{S_{\text{within}}}. \tag{48}$$

A larger Fisher ratio indicates that the distance between the two class centers is large relative to their within-class dispersion. This metric is calculated using all 11,192 test observations.

Silhouette coefficient. For each visualized observation i, let a_i be its mean Euclidean distance to the other observations with the same tail label, and let b_i be its mean distance to observations with the opposite label. Its silhouette coefficient is

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}. \tag{49}$$

The reported silhouette score is

$$\text{Silhouette} = \frac{1}{|\mathcal{S}|} \sum_{i\in\mathcal{S}} s_i, \tag{50}$$

where $\mathcal{S}$ is the common set of 2,244 observations shown in the visualization. The score lies between -1 and 1: a larger value indicates stronger separation, a value near zero indicates overlapping classes, and a negative value indicates that observations are, on average, closer to the opposite class than to their own class.

15-NN tail purity. For each right-tail observation $i \in \mathcal{T}$, let $\mathcal{N}_{15}(i)$ denote its 15 nearest neighbors among all other test observations, measured by Euclidean distance in the standardized 64-dimensional space. Tail purity is defined as

$$\text{Purity}_{15} = \frac{1}{15|\mathcal{T}|} \sum_{i\in\mathcal{T}} \sum_{j\in\mathcal{N}_{15}(i)} \mathbb{I}(j \in \mathcal{T}). \tag{51}$$

It measures the average proportion of right-tail observations in the local neighborhoods of right-tail samples. A larger value indicates that right-tail observations form more locally concentrated regions. This metric is calculated over the complete set of 11,192 test observations.

All three quantitative measures are computed before PCA and t-SNE. Therefore, they evaluate the learned representation itself rather than the potentially distorted geometry of the two-dimensional visualization.

A.9 Bidirectional Readout Consistency

The primary TCDL objective imposes a one-way consistency constraint from the classification head to the regression head. Specifically, if the classification argmax selects the explicit right-tail state, the regression output is encouraged to exceed the corresponding return threshold. To examine whether consistency should also be imposed in the reverse direction, we additionally evaluate a bidirectional variant.

Let

$$\widehat{s}_i = \arg \max_{k\in\{0,\ldots,K\}} p_{i,k} \tag{52}$$

denote the predicted state, where K is the explicit right-tail class, and let $\hat{z}_i$ and τ_z denote the standardized regression prediction and right-tail threshold, respectively. The original classification-to-regression consistency term is

$$\mathcal{L}_{\text{cls}\to\text{reg}} = \begin{cases} \dfrac{1}{|\mathcal{B}_{\text{fwd}}|} \displaystyle\sum_{i\in\mathcal{B}_{\text{fwd}}} \left[\max(0, \tau_z - \widehat{z}_i)\right]^2, & |\mathcal{B}_{\text{fwd}}| > 0, \\ 0, & |\mathcal{B}_{\text{fwd}}| = 0, \end{cases} \tag{53}$$

where

$$\mathcal{B}_{\text{fwd}} = \{i : \widehat{s}_i = K\}. \tag{54}$$

This term penalizes a *forward conflict*, defined as

$$\widehat{s}_i = K \quad \wedge \quad \widehat{r}_i < \tau. \tag{55}$$

For the reverse direction, we identify samples for which the regression output exceeds the threshold but the classification argmax does not select the right-tail state:

$$\mathcal{B}_{\text{rev}} = \{i : \widehat{z}_i > \tau_z \ \wedge \ \widehat{s}_i \neq K\}. \tag{56}$$

These samples constitute *reverse conflicts*. Instead of pushing their regression predictions below the threshold, we encourage the classification head to assign them to the explicit right-tail state:

$$\mathcal{L}_{\text{reg}\to\text{cls}} = \begin{cases} -\dfrac{1}{|\mathcal{B}_{\text{rev}}|} \displaystyle\sum_{i\in\mathcal{B}_{\text{rev}}} \log p_{i,K}, & |\mathcal{B}_{\text{rev}}| > 0, \\ 0, & |\mathcal{B}_{\text{rev}}| = 0. \end{cases} \tag{57}$$

The bidirectional objective is therefore

$$\mathcal{L}_{\text{bi}} = \mathcal{L}_{\text{MSE}} + 0.5\mathcal{L}_{\text{CE}} + 0.1\mathcal{L}_{\text{cls}\to\text{reg}} + 0.1\mathcal{L}_{\text{reg}\to\text{cls}}. \tag{58}$$

We train the bidirectional variant from scratch for both temporal splits using the same three random seeds, model architecture, tokenizer, optimization settings, and early-stopping procedure as the one-way TCDL model. Thus, the comparison changes only the addition of $\mathcal{L}_{\text{reg}\to\text{cls}}$. They quantify disagreement between the two prediction heads and do not use the ground-truth tail label.

Table 8: Comparison of one-way and bidirectional threshold-consistency objectives.

Period	Configuration	Reg MSE ↓	Reg MAE ↓	Tail AUPRC ↑	Tail AUROC ↑
2015–2019	One-way TCDL	0.000522 ± 0.000004	0.016053 ± 0.000046	0.146371 ± 0.004936	0.593543 ± 0.033934
	Bidirectional TCDL	0.000525 ± 0.000005	0.016102 ± 0.000022	0.140607 ± 0.002866	0.579958 ± 0.026709
2021–2025	One-way TCDL	0.000488 ± 0.000003	0.015831 ± 0.000087	0.132573 ± 0.006214	0.589602 ± 0.012112
	Bidirectional TCDL	0.000487 ± 0.000003	0.015803 ± 0.000019	0.128207 ± 0.003888	0.595132 ± 0.007318

As shown in Table 8, the bidirectional variant does not consistently improve predictive performance. It performs worse on all four metrics in 2015–2019. Although it slightly improves MSE, MAE, and AUROC in 2021–2025, its AUPRC decreases from 0.1326 to 0.1282. Therefore, the additional reverse constraint does not provide a robust improvement across market periods.

These results suggest that occasional regression-threshold crossings are not always reliable enough to supervise the classification head. Forcing the classifier to follow such crossings may propagate regression noise into the discrete-state prediction. We therefore retain the more conservative one-way classification-to-regression constraint in the final TCDL model.