RAVEN: A REGIME-AWARE VARIABLE-CONTEXT EXPERT NETWORK FOR FINANCIAL TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

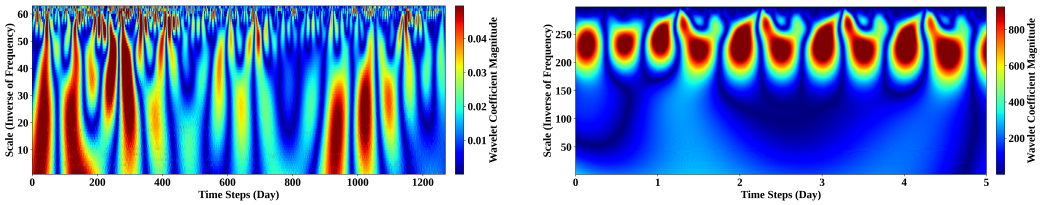
Financial time series forecasting presents structural challenges absent from standard benchmarks. Log-returns are non-stationary, exhibit exceptionally low signal-to-noise ratios, and follow regime-dependent temporal dependencies. Recent forecasting models typically rely on a fixed context window, which is mismatched with the time-varying optimal look-back of log-return processes. To address this limitation, we propose the Regime-Aware Variable-context Expert Network (RAVEN), a Mixture-of-Experts framework that adaptively determines the temporal context for each input sample. Specifically, RAVEN constructs a hierarchy of nested contiguous windows determined by the data itself. The model scores historical patches in reverse chronological order and applies Cumulative Importance Thresholding (CIT) to route each prefix to a scale-specialized expert. To overcome representation redundancy caused by overlapping inputs, it penalizes pairwise expert cosine similarity prior to aggregation. Meanwhile, a parallel global compression branch is incorporated to preserve macro-level temporal coherence. Experiments on cumulative log-return prediction (HS300, S&P500) and retail fund sales forecasting demonstrate that RAVEN achieves state-of-the-art performance, which translates into consistent risk-adjusted alpha gains in market backtests. Beyond financial applications, evaluations on four PEMS benchmarks confirm that its adaptive look-back generalizes effectively to broader physical dynamic systems.

1 INTRODUCTION

Financial time series forecasting is a cornerstone of quantitative investment, supporting tasks from risk management to automated trading. Unlike well-known benchmark datasets in the general time series domain, such as ETTh, ETTm, Weather, Electricity, or Traffic (Wu et al., 2021), which often exhibit clear periodic patterns and deterministic trends, financial data is noisy and non-stationary (Lo, 2004). In this adversarial environment, predicting raw closing price levels C_t at time t is impractical: prices are not scale-invariant or cross-asset comparable, typically follow a random walk, and exhibit extremely high autocorrelation, leading to spurious regression and inflated in-sample performance. To capture meaningful predictive signals, existing approaches reformulate the problem as the regression of log-returns: $r_t = \ln(C_t/C_{t-1})$ (Hasbrouck, 1991; Gu et al., 2020). This transformation shifts the modeling objective from tracking absolute values to capturing price innovations (Hasbrouck, 1991), the unpredictable stochastic components driven by the arrival of new market information. While log-returns are statistically more stable than raw prices, they still exhibit exceptionally low signal-to-noise ratios (SNR) and heavy-tailed distributions, making accurate regression a challenging task (Gu et al., 2020).

Historically, financial time-series analysis has been dominated by tree-based ensembles like XGBoost (Chen & Guestrin, 2016) and LightGBM (Ke et al., 2017) (Friedman, 2001). They model non-linear interactions over handcrafted features, but treat forecasts as static tabular problems, ignoring the temporal topology of market regimes. Subsequent deep architectures, from Multilayer Perceptrons and RNNs (Elman, 1990) to LSTMs (Hochreiter & Schmidhuber, 1997) and GRUs (Cho et al., 2014), restore temporal memory through recurrence and gating, yet remain biased toward recent observations and struggle to separate transient noise from structural regime shifts. Re-

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.



(a) HS300 constituent 600176.SS (daily log-returns) **(b)** PEMS03 traffic flow (5-min)

Figure 1: CWT scalograms for multi-scale analysis. Financial data (a) exhibits non-stationary energy distribution with no fixed periodicity, while traffic data (b) shows stable, periodic patterns.

cently, Transformer-based and frequency-domain architectures (Zhou et al., 2021; Nie et al., 2023; Wu et al., 2023; Liu et al., 2024; Chen et al., 2024; Wang et al., 2024) have advanced forecasting by capturing long-range dependencies. However, these models inherit an unexamined bottleneck through their reliance on a fixed-length context window L. In non-stationary environments, this static context window creates an irreconcilable conflict, where short windows lack the memory to capture structural regime shifts, while long windows unavoidably inject stale dynamics from prior regimes as additive noise.

To empirically substantiate this mismatch, we perform Continuous Wavelet Transform (CWT) diagnostics (Torrence & Compo, 1998). Figure 1 visualizes the scalograms of two representative series. For the HS300 constituent 600176.SS (Figure 1a), spectral energy drifts unpredictably across scales over time—alternating between high-frequency bursts and low-frequency regimes without stable periodicity. In contrast, the PEMS03 traffic series (Figure 1b) displays stationary, horizontal bands reflecting fixed daily periods. This divergence confirms that the optimal look-back horizon in non-stationary data is intrinsically sample dependent, rendering any fixed window structurally mismatched with the underlying dynamics.

Classical econometric models already hinted at the utility of multi-horizon reasoning. The HAR-RV framework (Corsi, 2009) linearly aggregates daily, weekly, and monthly horizons to capture complementary temporal scales. Recent research, like MLF (Zhang et al., 2025), extends this multi-period intuition via multi-branch attention over predefined periods and uniform patches. However, both paradigms rely on static, heuristic horizon splits rather than learning them dynamically from the data. Meanwhile, emerging Mixture-of-Experts (MoE) architectures (Shi et al., 2025; Sun et al., 2025) route isolated tokens via localized similarity, which disrupts the contiguous chronological order essential for modeling causal temporal processes.

To address these limitations, we propose the Regime-Aware Variable-context Expert Network (RAVEN), a MoE framework that dynamically determines the temporal context for each input sample. Unlike token-level routing, RAVEN evaluates historical patches in reverse chronological order from the present timestep and uses Cumulative Importance Thresholding (CIT) to construct nested, variable-length windows for scale-specialized experts. Because overlapping window inputs induce representation redundancy under low SNR ratios, the framework incorporates Correlation-Aware Weighting (CAW) to suppress pairwise expert collinearity prior to aggregation. Meanwhile, a parallel global compression branch preserves macro-level temporal coherence across the entire sequence.

Our main contributions are summarized as follows:

- We identify the fundamental vulnerability of a fixed context window in non-stationary forecasting via wavelet spectral diagnostics, and propose RAVEN, a MoE framework that dynamically infers the optimal look-back horizon for each sample while preserving temporal causal order.
- We introduce **CIT** to dynamically partition history into nested contiguous windows for scale specialized experts, combined with **CAW** and diversity regularization to suppress severe representation collinearity induced by overlapping inputs.
- Across multiple equity indices (HS300, S&P500) and retail fund forecasting, RAVEN consistently outperforms competitive baselines, translating into substantial risk adjusted alpha gains in market backtests. Furthermore, evaluations on four PEMS benchmarks demonstrate its robust generalization to broader physical dynamic systems.

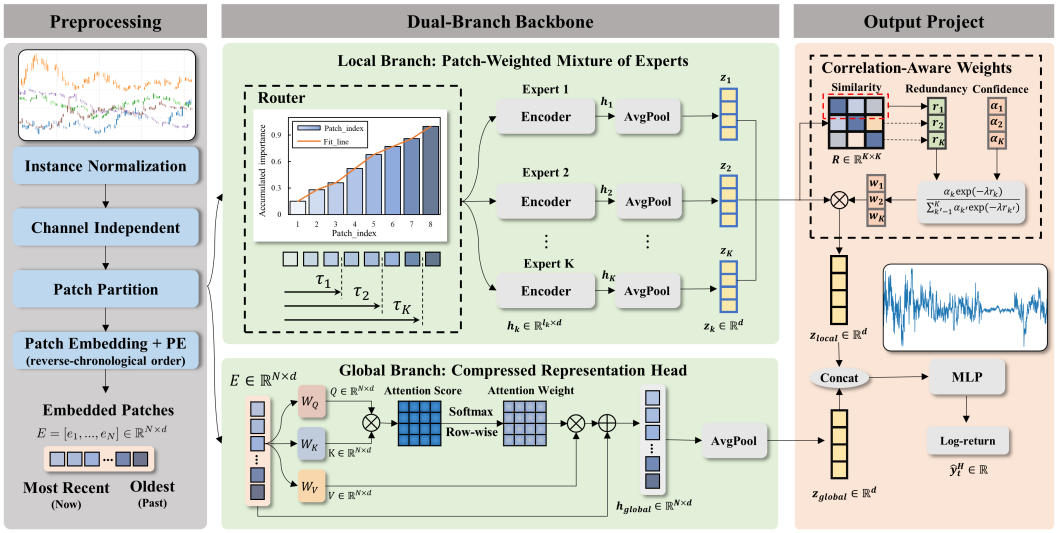


Figure 2: Overview of RAVEN. **Preprocessing** converts the input into patch embeddings. The **Dual-Branch Backbone** combines a local adaptive branch, which uses **CIT** to construct nested contexts for scale-specialized experts, with a Global Compressed Representation (**GCR**) branch over the full context. **Output Projection** aggregates the local experts with **CAW**, fuses the local and global representations, and produces the final forecast.

2 RAVEN Design

2.1 Design Overview

We consider multi-horizon log-return forecasting from multivariate financial time series. Let $\mathbf{x}_t \in \mathbb{R}^D$ denote the D-dimensional market state observed at time t, and let C_t denote the closing price. The one-step log-return is $r_t = \ln(C_t/C_{t-1})$. Given a maximum look-back length $L_{\max}$, the input instance is $\mathbf{X}_t = [\mathbf{x}_{t-L_{\max}+1}, \ldots, \mathbf{x}_t]^\top \in \mathbb{R}^{L_{\max} \times D}$. For forecast horizon H, the prediction target is the cumulative log-return:

$$y_t^{(H)} = \sum_{h=1}^{H} r_{t+h} = \ln(C_{t+H}/C_t) \in \mathbb{R}. \tag{1}$$

As illustrated in Figure 2, RAVEN consists of three stages. **Preprocessing** transforms $\mathbf{X}_t$ into patch embeddings. The **Dual-Branch Backbone** combines CIT-based adaptive local context modeling with a GCR branch over the full historical context. Finally, **Output Projection** uses CAW to aggregate the local experts, fuses the local and global representations, and maps them to $\hat{y}_t^{(H)}$.

2.2 Preprocessing

RAVEN processes each channel independently using instance normalization, non-overlapping patching, and a shared linear embedding (Kim et al., 2022; Nie et al., 2023):

$$\mathbf{E} = \text{LinearEmbed}(\text{Patching}(\text{RevIN}(\mathbf{X}_t))) \in \mathbb{R}^{D \times N \times d}, \tag{2}$$

where $N = \lceil L_{\max}/p_{\text{len}} \rceil$, p_{len} is the patch length, and d is the embedding dimension. Patches are arranged in reverse chronological order, and $\mathbf{E}$ serves as the shared input to both backbone branches.

2.3 Dual-Branch Backbone

The backbone contains two parallel branches. The local branch constructs adaptive historical contexts and models them with experts at different scales. The GCR branch captures information from the full historical context.

Local branch. RAVEN first assigns an importance score $s_i = \phi(\mathbf{e}_i)$ to each embedded patch $\mathbf{e}_i$ using a learnable scoring function $\phi(\cdot)$. The scores are normalized over all patches:

$$\tilde{s}_i = \frac{\exp(s_i)}{\sum_{j=1}^{N} \exp(s_j)}, \qquad i = 1, \ldots, N. \tag{3}$$

CIT then accumulates the normalized scores in reverse chronological order and applies K ordered thresholds to the cumulative importance, $0 < \tau_1 < \tau_2 < \cdots < \tau_K \leq 1$:

$$\mathcal{G}_k = \{i \in \{1, \ldots, N\} : \Psi_i \leq \tau_k\}, \qquad \Psi_i = \sum_{j=1}^{i} \tilde{s}_j. \tag{4}$$

The resulting groups satisfy $\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \cdots \subseteq \mathcal{G}_K$, forming nested contiguous contexts anchored at the most recent patch. Their lengths $\ell_k = |\mathcal{G}_k|$ adapt to each input. We set $K = 3$ and $(\tau_1, \tau_2, \tau_3) = (0.3, 0.6, 0.9)$ by default. RAVEN next processes each group $\mathcal{G}_k$ with an independent encoder:

$$\mathbf{h}_k = \text{Encoder}_k\left(\{\mathbf{e}_i\}_{i \in \mathcal{G}_k}\right) \in \mathbb{R}^{\ell_k \times d}. \tag{5}$$

Each $\mathbf{h}_k$ therefore represents a temporal context at a different scale. $Expert_1$ receives the shortest context, while $Expert_K$ receives the longest. Average pooling then maps each expert output to a representation of the same dimension:

$$\mathbf{z}_k = \text{AvgPool}(\mathbf{h}_k) \in \mathbb{R}^d. \tag{6}$$

Since the expert inputs overlap, their representations can be correlated. To address this redundancy, CAW measures pairwise similarity using the cosine similarity matrix $\mathbf{R} \in \mathbb{R}^{K \times K}$, where $R_{jk} = \mathbf{z}_j^\top \mathbf{z}_k / (\|\mathbf{z}_j\| \|\mathbf{z}_k\|)$. The positive redundancy score for expert k is $r_k = \sum_{j \neq k} \max(R_{jk}, 0)$. CAW uses this score to adjust the raw routing confidence α_k:

$$w_k = \frac{\alpha_k \exp(-\lambda r_k)}{\sum_{k'=1}^{K} \alpha_{k'} \exp(-\lambda r_{k'})}, \qquad \lambda \geq 0, \tag{7}$$

where λ is learnable. The final local representation is:

$$\mathbf{z}_{\text{local}} = \sum_{k=1}^{K} w_k \mathbf{z}_k \in \mathbb{R}^d. \tag{8}$$

Global branch. GCR applies self-attention to the full patch sequence $\mathbf{E}$ and compresses the resulting representations through average pooling:

$$\mathbf{z}_{\text{global}} = \text{AvgPool}(\text{SelfAttn}(\mathbf{E})) \in \mathbb{R}^d. \tag{9}$$

2.4 Output Projection

The prediction head concatenates the local and global representations and projects them to the final forecast:

$$\hat{y}_t^{(H)} = \text{MLP}\big(\text{Concat}(\mathbf{z}_{\text{local}}, \mathbf{z}_{\text{global}})\big) \in \mathbb{R}. \tag{10}$$

2.5 Training Objective

RAVEN is trained with a composite objective that combines the MSE forecasting loss with two regularization terms:

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{ent}} \mathcal{L}_{\text{ent}} + \lambda_{\text{div}} \mathcal{L}_{\text{div}}, \tag{11}$$

where $\mathcal{L}_{\text{ent}}$ and $\mathcal{L}_{\text{div}}$ are the routing entropy and expert diversity regularization terms, respectively, with weights λ_{ent} and λ_{div}. They are defined as:

$$\mathcal{L}_{\text{ent}} = \sum_{i=1}^{N} \tilde{s}_i \log \tilde{s}_i, \qquad \mathcal{L}_{\text{div}} = \|\mathbf{R} - \mathbf{I}_K\|_F^2, \tag{12}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix. The routing entropy regularizer $\mathcal{L}_{\text{ent}}$ acts on the normalized patch importance distribution $\{\tilde{s}_i\}_{i=1}^{N}$. Minimizing it discourages excessive concentration on a few

patches, preventing the cumulative importance from reaching different CIT thresholds within nearly identical prefixes. This helps maintain distinct context lengths for experts operating at different temporal scales. The expert diversity regularizer $\mathcal{L}_{\text{div}}$ acts on the expert similarity matrix $\mathbf{R}$ defined in the local branch. It penalizes similarity between different expert representations, reducing redundancy caused by their overlapping contexts. This complements CAW by encouraging representation diversity during training, while CAW adjusts correlated expert contributions during aggregation. Further details are provided in Appendix A.5.

3 EXPERIMENTS

We evaluate RAVEN across financial and general time series forecasting tasks. We further conduct ablation and efficiency analyses to assess the key design choices and computational cost. Finally, we analyze the learned routing behavior to reveal how RAVEN adapts its temporal context.

3.1 DATASETS AND BASELINES

Datasets. We evaluate RAVEN on three categories of datasets. For financial return prediction, we use HS300 and S&P500 stocks from FinMultiTime (Xu et al., 2025). We further consider fund sales prediction on the Fund dataset (Zhang et al., 2025) and general time series forecasting on four PEMS traffic benchmarks (PEMS03, PEMS04, PEMS07, and PEMS08) (Guo et al., 2019). Dataset statistics and preprocessing details are provided in Appendix A.2.

Baselines. We compare RAVEN with representative forecasting models across four categories. Fixed context and predefined multi-period baselines include PatchTST (Nie et al., 2023), Patch-Concat, Patch-Ensemble, and MLF (Zhang et al., 2025). Multiscale models include Path-Former (Chen et al., 2024), Scaleformer (Shabani et al., 2023), and NHits (Challu et al., 2023). We further include cross-variate models iTransformer (Liu et al., 2024) and Crossformer (Zhang & Yan, 2023), as well as frequency-domain models TimesNet (Wu et al., 2023), FEDformer (Zhou et al., 2022b), and FiLM (Zhou et al., 2022a). Configurations, evaluation protocols, and implementation details are provided in Appendix A.4.

3.2 STOCK LOG-RETURN PREDICTION

For HS300 and S&P500, we evaluate 10-day cumulative log-return prediction from 2020 to 2024 under a rolling out-of-sample setting. We report Pearson correlation (Corr), MSE, and ICIR. Corr measures the temporal association between predicted and realized returns for individual stocks, while ICIR summarizes the strength and consistency of these correlations across the stock universe. Detailed definitions and evaluation protocols are provided in Appendix A.3.

Prediction performance. Table 1 reports the year-by-year results on HS300 and S&P500. On HS300, RAVEN ranks first or second across all 15 year-metric comparisons, with 7 best and 8 second-best results. On S&P500, RAVEN achieves the best result in 14 of 15 comparisons. These results show consistent performance across different markets and evaluation years, supporting the effectiveness of adaptive context modeling under changing market conditions. Table 2 further summarizes performance over the full 2020 to 2024 test period. Compared with MLF, the strongest competing baseline, RAVEN improves Corr by 9.2% on HS300 and 20.2% on S&P500.

Backtesting. To examine whether these predictive gains translate into investment performance, we conduct a portfolio backtest on HS300 from 2020 to 2024 using Qlib (Yang et al., 2020). All models use the same portfolio construction and execution settings, with their predicted returns serving as trading signals. Figure 3 shows the cumulative return advantage of RAVEN over each baseline from 2020 to 2024, where $\Delta(\cdot)$ denotes the cumulative return of RAVEN minus that of the corresponding baseline. The return gaps increase over time, with RAVEN achieving a final cumulative return advantage of 12.79% over MLF. Detailed portfolio construction and execution settings are provided in Appendix A.7.

Table 1: Performance comparison on HS300 and S&P500 (2020–2024). Corr and ICIR: higher is better (↑); MSE: lower is better (↓). Best in **bold**, second underlined.

Dataset: HS300

Model	2020			2021			2022			2023			2024		
	Corr	MSE	ICIR	Corr	MSE	ICIR	Corr	MSE	ICIR	Corr	MSE	ICIR	Corr	MSE	ICIR
RAVEN	**.0567**	.9706	**.5943**	**.0292**	**.9897**	**.1478**	.0422	.9967	.5224	**.0300**	**.9978**	.3321	.0500	.9971	.3765
MLF	.0525	**.9647**	.5457	.0173	1.002	.1056	**.0583**	**.9786**	**.5467**	.0152	1.002	.2771	.0388	1.004	.3325
NHits	.0338	.9885	.3222	.0143	.9943	.1153	.0238	.9998	.4677	.0125	1.012	.3147	**.0525**	**.9938**	**.3772**
PathFormer	.0462	.9759	.5438	.0216	.9917	.1363	.0253	.9996	.4063	.0176	1.003	**.3332**	.0315	1.012	.3080
PatchTST	.0322	.9893	.3277	.0043	1.012	.0593	.0193	1.006	.4207	.0089	1.012	.2664	.0429	1.001	.3151
Patch-Concat	.0373	.9803	.5082	.0140	1.003	.1190	.0286	.9974	.4233	.0138	1.005	.2202	.0428	1.002	.3550
Patch-Ensemble	.0320	.9908	.3477	.0185	1.001	.0756	.0175	1.004	.4469	.0093	1.014	.2657	.0376	1.003	.3092
FiLM	.0372	.9818	.5282	.0164	1.003	.1123	.0195	1.004	.4709	.0173	1.006	.3280	.0326	1.010	.3200

Dataset: S&P500

Model	2020			2021			2022			2023			2024		
	Corr	MSE	ICIR	Corr	MSE	ICIR	Corr	MSE	ICIR	Corr	MSE	ICIR	Corr	MSE	ICIR
RAVEN	**.0339**	**1.003**	**.7256**	**.0365**	**.9989**	**.5892**	**.0426**	.9985	**.6078**	**.0405**	**1.002**	**.5481**	**.0283**	**1.001**	**.5795**
MLF	.0268	1.027	.6515	.0283	1.023	.5420	.0350	1.011	.5689	.0339	1.008	.5149	.0207	1.013	.5128
NHits	.0268	1.015	.6423	.0301	1.001	.5475	.0379	**.9957**	.5827	.0310	1.007	.5025	.0221	1.007	.5143
PathFormer	.0258	1.027	.6337	.0212	1.039	.4737	.0297	1.028	.5204	.0397	1.006	.5394	.0220	1.009	.5070
PatchTST	.0195	1.042	.5616	.0219	1.032	.5016	.0225	1.037	.4837	.0305	1.014	.4850	.0126	1.026	.4431
Patch-Concat	.0242	1.027	.6254	.0262	1.023	.5154	.0234	1.032	.5132	.0324	1.011	.5041	.0186	1.013	.4713
Patch-Ensemble	.0302	1.017	.6742	.0275	1.021	.5263	.0314	1.021	.5298	.0342	1.007	.5114	.0243	1.009	.5290
FiLM	.0275	1.022	.6524	.0193	1.036	.4724	.0322	1.018	.5511	.0302	1.014	.5091	.0230	1.013	.5127

Table 2: Overall performance on HS300 and S&P500.

Model	HS300			S&P500		
	Corr	MSE	ICIR	Corr	MSE	ICIR
RAVEN	**.0390**	**.9905**	**.3932**	**.0363**	**1.001**	**.5980**
MLF	.0357	.9932	.3579	.0302	1.016	.5577
NHits	.0282	.9987	.3210	.0294	1.002	.5518
PathFormer	.0278	.9990	.3260	.0272	1.020	.5300
PatchTST	.0217	1.003	.2758	.0225	1.030	.4836
Patch-Concat	.0272	.9987	.3177	.0254	1.023	.5203
Patch-Ensemble	.0218	1.003	.2811	.0290	1.018	.5523
FiLM	.0247	1.003	.3463	.0264	1.023	.5360

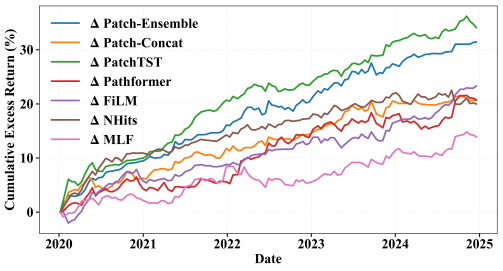


Figure 3: Cumulative return advantage on HS300.

Table 3: Performance comparison on Fund dataset. MSE (↓) and WMA (↓). Best in **bold**; second-best underlined.

	RAVEN		MLF		PatchTST		Patch-E		Patch-C		NHits		FiLM		Scaleformer		PathFormer	
	MSE	WMA	MSE	WMA	MSE	WMA	MSE	WMA	MSE	WMA	MSE	WMA	MSE	WMA	MSE	WMA	MSE	WMA
1	**27.69**	**74.63**	33.85	75.84	36.68	81.05	40.3	86.68	39.68	85.87	36.53	80.29	46.12	96.1	-	-	35.54	80.75
5	**30.22**	**80.09**	38.28	80.37	40.28	83.09	39.95	83.24	41.6	87.14	40.91	83.93	42.86	85.35	40.43	83.74	39.9	82.58
8	**33.62**	**85.32**	41.94	86.06	44.71	88.81	44.06	88.77	43.49	88.08	44.81	89.23	44.57	89.12	43.97	87.62	44.17	88.67
10	**38.13**	**86.46**	44.42	88.66	46.18	90.63	46.09	91.36	45.59	90.46	56.73	91.12	46.62	92.67	45.75	89.63	45.56	89.82
Avg.	**32.42**	**81.63**	39.62	82.73	41.96	85.90	42.60	87.51	42.59	87.89	44.75	86.14	45.04	90.81	43.38	87.00	41.29	85.46

3.3 Fund Sales Prediction

Table 3 reports fund sales prediction across four forecast horizons. RAVEN achieves the best MSE and WMA at every horizon. Compared with MLF, the strongest baseline, RAVEN reduces the average MSE by 18.2% (32.42 vs. 39.62), with improvements ranging from 14% to 21% across horizons. These results show that adaptive context modeling also benefits financial time series beyond stock return prediction. Other multiscale and predefined-context baselines remain less competitive, suggesting that broader temporal coverage alone does not explain the gains.

Table 4: Short-term forecasting results on traffic datasets (MSE / MAE ↓). Best in **bold**; second-best underlined.

Model	PEMS03				PEMS04				PEMS07				PEMS08			
	$H=12$		$H=24$		$H=12$		$H=24$		$H=12$		$H=24$		$H=12$		$H=24$	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
RAVEN	**.067**	**.169**	**.091**	.204	**.074**	**.172**	**.093**	**.195**	.072	**.153**	**.086**	**.181**	**.077**	**.168**	**.110**	**.207**
MLF	.072	.184	.098	.227	.084	.194	.103	.233	.088	.165	.096	.207	.107	.192	.142	.246
PatchTST	.099	.216	.142	.259	.105	.224	.153	.275	.095	.207	.150	.262	.168	.232	.224	.281
iTransformer	.071	.174	.093	**.201**	.078	.183	.095	.205	**.067**	.165	.088	.190	.079	.182	.115	.219
NHits	.083	.197	.112	.229	.081	.185	.098	.221	.083	.162	.094	.193	.114	.214	.169	.257
TimesNet	.085	.192	.118	.223	.087	.195	.103	.215	.082	.181	.101	.204	.112	.212	.141	.238
FiLM	.089	.208	.120	.233	.079	.188	.106	.227	.092	.182	.117	.226	.124	.225	.195	.273
Patch-Ensemble	.107	.239	.159	.285	.118	.247	.176	.290	.104	.221	.153	.267	.182	.246	.231	.298
Patch-Concat	.103	.224	.131	.247	.096	.218	.157	.279	.094	.213	.143	.255	.149	.228	.192	.260
Crossformer	.090	.203	.121	.240	.098	.218	.131	.256	.094	.200	.139	.247	.165	.214	.215	.260
Scaleformer	.086	.204	.109	.216	.088	.197	.120	.233	.096	.175	.105	.207	.131	.226	.174	.253
PathFormer	.091	.209	.126	.237	.084	.190	.133	.231	.085	.173	.116	.219	.138	.215	.182	.258
FEDformer	.126	.251	.149	.275	.138	.262	.177	.293	.109	.225	.125	.244	.173	.273	.210	.301

Table 5: Ablation study on HS300 (Pearson Corr ↑).

Year	**RAVEN**	w/o AR	w/o CAW	w/o GCR	w/o $\mathcal{L}_{\text{ent}}$	w/o $\mathcal{L}_{\text{div}}$
2020	0.0567	0.0550	0.0569	0.0573	0.0583	**0.0584**
2021	**0.0292**	0.0271	0.0260	0.0263	0.0290	0.0288
2022	**0.0422**	0.0309	0.0313	0.0320	0.0321	0.0318
2023	**0.0300**	0.0193	0.0184	0.0204	0.0202	0.0199
2024	**0.0500**	0.0432	0.0441	0.0434	0.0449	0.0452
All	**0.0390**	0.0352	0.0358	0.0360	0.0373	0.0372

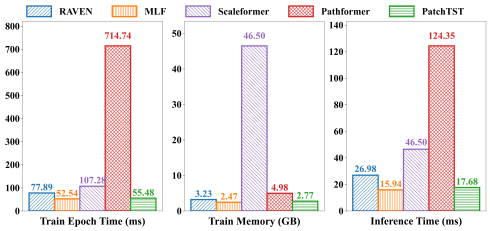


Figure 4: Computational efficiency on HS300.

3.4 General Time Series Forecasting

To examine whether adaptive context modeling generalizes beyond financial time series, we further evaluate RAVEN on four PEMS traffic forecasting benchmarks. We consider horizons $H \in \{12, 24\}$ and report MSE and MAE. Table 4 shows that RAVEN achieves the best result in 14 of the 16 test cases across all four datasets and both forecast horizons. Among the baselines, iTransformer is the most competitive, achieving the best results in the remaining two cases. These results suggest that the benefits of adaptive context modeling extend to non-financial time series with different temporal dynamics.

3.5 Ablation of Core Components

To isolate the contributions of RAVEN's core components, we evaluate five ablated variants on HS300. w/o AR removes adaptive routing and replaces the learned context selection with predefined windows. w/o CAW removes correlation-aware expert weighting, while w/o GCR removes the global context branch. Finally, w/o $\mathcal{L}_{\text{ent}}$ and w/o $\mathcal{L}_{\text{div}}$ remove the routing entropy and expert diversity regularizers, respectively. Table 5 reports the year-by-year and overall Corr for all variants. Among the architectural components, removing adaptive routing causes the largest degradation, reducing the overall Corr by 9.7%. Removing CAW and GCR leads to drops of 8.2% and 7.7%, respectively. These results identify adaptive context selection as the primary architectural contribution, while correlation-aware aggregation and global context modeling provide substantial complementary gains. The two regularization terms each contribute approximately 4.5%, further supporting their roles in stabilizing routing and encouraging diversity among expert representations.

3.6 CIT Configuration Analysis

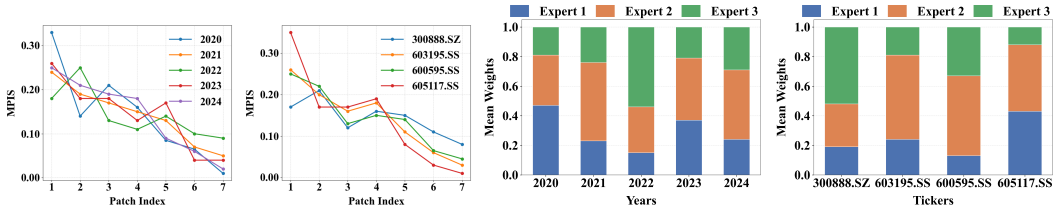
We examine how the number of experts K and the cumulative importance thresholds affect CIT on HS300. We vary $K \in \{2, 3, 4\}$ with thresholds evenly spaced between 0.3 and 0.9. Table 6 shows that $K = 3$ achieves the highest overall Corr of 0.0390, compared with 0.0339 for $K = 2$ and 0.0361 for $K = 4$. Using fewer experts limits the number of temporal scales. Increasing K, however, introduces more overlap between adjacent contexts, which can increase representation redundancy without yielding additional predictive gains.

Table 6: Effect of number of experts K on HS300.

Year	K=2 (0.3, 0.9)	K=3 (0.3, 0.6, 0.9)	K=4 (0.3, 0.5, 0.7, 0.9)
2020	0.0533	0.0567	**0.0582**
2021	0.0273	**0.0292**	0.0274
2022	0.0300	**0.0422**	0.0321
2023	0.0173	**0.0300**	0.0184
2024	0.0421	**0.0500**	0.0442
All	0.0339	**0.0390**	0.0361

Table 7: Effect of threshold values on HS300.

Year	K=3 (0.3, 0.6, 0.9)	K=3 (0.2, 0.4, 0.8)	K=3 (0.1, 0.5, 0.9)
2020	0.0567	**0.0602**	0.0583
2021	0.0292	**0.0304**	0.0303
2022	**0.0422**	0.0313	0.0321
2023	**0.0300**	0.0207	0.0199
2024	**0.0500**	0.0442	0.0449
All	**0.0390**	0.0380	0.0370



(a) Across years. **(b)** Across stocks. **(c)** Across years. **(d)** Across stocks.

Figure 5: Adaptive routing behavior on HS300. (a–b) Learned patch importance and (c–d) expert aggregation weights across years and stocks. Patch index 1 denotes the most recent patch, while $Expert_1$ and $Expert_3$ denote the short- and long-context experts, respectively.

We further evaluate three threshold configurations under $K = 3$. Table 7 shows that the overall Corr varies only from 0.0370 to 0.0390 across these settings. Paired tests on per-sample MSE show no significant difference from the default $(0.3, 0.6, 0.9)$ configuration, with $p = 0.66$ and $p = 0.19$ for the two alternatives. This suggests that CIT is not highly sensitive to the exact threshold values and does not rely on a narrowly tuned threshold configuration. Additional sensitivity analyses for the maximum look-back window and patch length are provided in Appendix A.6.

3.7 Adaptive Routing Analysis

We further analyze the learned routing behavior to understand how RAVEN adjusts its temporal context for different inputs. Figure 5 visualizes the upstream patch importance used by CIT and the downstream weights used for expert aggregation. Figures 5a–b show substantial variation in the learned patch importance profiles. For the same stock, the distribution changes across years, assigning different importance to recent and distant patches. The profiles also vary across stocks within the same year. Since CIT constructs nested contexts from cumulative patch importance, these changes allow the effective look-back range to vary with the input instead of using a fixed historical window. Figures 5c–d show corresponding variation in expert aggregation. The weights assigned to experts with different temporal scales change across years for the same stock and also differ across stocks within the same year. This shows that RAVEN adjusts both the historical context selected by CIT and the contribution of different temporal scales to the final representation. Together, these observations provide direct evidence that RAVEN adapts its temporal context across both temporal and cross-sectional variations, rather than relying on a fixed or predefined context.

3.8 Efficiency Analysis

We evaluate the computational efficiency of RAVEN on HS300 against representative lightweight and multiscale baselines. As shown in Figure 4, RAVEN incurs moderate overhead over lightweight models such as MLF and PatchTST due to its adaptive routing and multiple experts. In contrast, it is substantially more efficient than heavier multiscale models. Compared with PathFormer, RAVEN reduces training time by 89.1%, peak memory by 35.1%, and inference latency by 78.3%. Overall, the moderate overhead over lightweight baselines is accompanied by consistent predictive gains, while RAVEN remains substantially more efficient than heavier multiscale models, yielding a favorable trade-off between predictive performance and computational cost.

4 Related Work

Deep learning for financial forecasting. Financial forecasting has evolved from tree-based models such as XGBoost (Chen & Guestrin, 2016) and LightGBM (Ke et al., 2017) to sequential neural architectures including RNNs (Elman, 1990) and LSTMs (Hochreiter & Schmidhuber, 1997). More recently, Transformer-based models have improved long-range dependency modeling through attention mechanisms (Wu et al., 2021; Zhou et al., 2021). Subsequent time series architectures further explore patch-based representations (Nie et al., 2023), cross-variate modeling (Zhang & Yan, 2023; Liu et al., 2024), and frequency-domain representations (Zhou et al., 2022b; Wu et al., 2023; Piao et al., 2024; Murad et al., 2025). Financial forecasting models have also introduced domain-specific mechanisms for cross-asset relations (Xu et al., 2021), market-guided attention (Li et al., 2024), adaptation to distribution shifts (Zhao et al., 2023), and conditional routing (Lin et al., 2021). Despite these advances, the historical context is typically fixed across inputs, leaving adaptation of the effective look-back range largely unexplored.

Adaptive context modeling. Several approaches incorporate multiple temporal scales to relax the reliance on a single context. MLF (Zhang et al., 2025) fuses predefined look-back periods with inter-period redundancy filtering, while PathFormer (Chen et al., 2024) dynamically aggregates features extracted at predefined patch resolutions. TimeSqueeze (Ankireddy et al., 2026) further adapts patch boundaries according to local signal complexity. These methods improve temporal flexibility, but the global historical context remains fixed or is selected from predefined scales. In contrast, RAVEN uses learned patch importance and CIT to construct nested contexts whose lengths vary with the input, allowing the effective look-back range itself to adapt to changing temporal dynamics.

Mixture of experts for time series. Sparse MoE architectures increase model capacity through conditional expert activation (Fedus et al., 2022). Recent time series models such as Time-MoE (Shi et al., 2025), Moirai-MoE (Liu et al., 2025), and TFPS (Sun et al., 2025) route tokens or patches to experts according to their content, promoting specialization in local patterns. Such routing does not explicitly assign experts to contiguous historical contexts at different temporal scales. In contrast, RAVEN routes nested contiguous contexts to experts operating at different temporal scales, preserving chronological structure within each context. CAW further accounts for redundancy introduced by overlapping contexts, while the GCR branch retains information from the full historical sequence.

5 Conclusion

We presented RAVEN, a MoE framework for adaptive context modeling in financial time series forecasting. Instead of using a fixed or predefined historical context, RAVEN learns patch importance and uses CIT to construct nested contexts for experts operating at different temporal scales. CAW mitigates redundancy among overlapping expert representations, while the GCR branch preserves information from the full historical context. Across stock log-return and fund sales forecasting, RAVEN consistently outperforms competitive baselines, improving Corr by 9.2% on HS300 and 20.2% on S&P500 and reducing average MSE by 18.2% on fund sales. Its strong performance on four PEMS benchmarks further shows that the benefits of adaptive context modeling extend beyond financial time series. Ablation studies identify adaptive routing as the primary architectural contribution, with CAW and GCR providing complementary gains. Analysis of the learned routing behavior further shows that both patch importance and expert aggregation vary across years and stocks, supporting the intended adaptation of temporal context to different inputs. While this adaptation introduces moderate overhead over lightweight baselines, RAVEN remains substantially more efficient than heavier multiscale models, providing a practical trade-off between predictive performance and computational cost. The current design adapts the look-back window while using a fixed patch length, leaving a deeper understanding of how historical context should change with latent market regimes as an important direction for future work. More broadly, adaptive context modeling could be incorporated into financial time series pretraining to develop foundation models that better accommodate the low signal-to-noise ratios and strong non-stationarity of financial data across diverse applications.

AI USE STATEMENT

In this work, we did not use generative AI tools for any of the tasks requiring disclosure, including generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating or proving mathematical claims, proposing hypotheses, designing methodologies or experiments, implementing methods, cleaning datasets, assisting with translation, or interpreting results.

We occasionally used generative AI tools solely for language editing of the manuscript, such as suggesting alternative wording and correcting grammar and spelling, with the aim of improving readability and clarity.

All AI-assisted edits were carefully reviewed by the authors. All technical ideas, experimental designs, analyses, conclusions, and substantive content were developed and carried out entirely by the authors. Any AI-suggested language edits were manually reviewed to ensure that they accurately reflected the authors' original intent and did not alter the underlying scientific claims. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REFERENCES

Sravan Kumar Ankireddy, Nikita Seleznev, Nam H. Nguyen, Yulun Wu, Senthil Kumar, Furong Huang, and C. Bayan Bruss. Timesqueeze: Dynamic patching for efficient time series forecasting. CoRR, abs/2603.11352, 2026. doi: 10.48550/ARXIV.2603.11352. URL https://doi.org/10.48550/arXiv.2603.11352.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramírez, Max Mergenthaler Canseco, and Artur Dubrawski. NHITS: neural hierarchical interpolation for time series forecasting. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023, pp. 6989–6997. AAAI Press, 2023. doi: 10.1609/AAAI.V37I6.25854. URL https://doi.org/10.1609/aaai.v37i6.25854.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net, 2024. URL https://openreview.net/forum?id=lJkOCMP2aW.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (eds.), Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pp. 785–794. ACM, 2016. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1724–1734. ACL, 2014. doi: 10.3115/V1/D14-1179. URL https://doi.org/10.3115/v1/d14-1179.

Fulvio Corsi. A simple approximate long-memory model of realized volatility. Journal of financial econometrics, 7(2):174–196, 2009.

Jeffrey L. Elman. Finding structure in time. Cogn. Sci., 14(2):179–211, 1990. doi: 10.1207/S15516709COG1402_1. URL https://doi.org/10.1207/s15516709cog1402_1.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. J. Mach. Learn. Res., 23:120:1–120:39, 2022. URL https://jmlr.org/papers/v23/21-0998.html.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. Annals of statistics, pp. 1189–1232, 2001.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5):2223–2273, 2020.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pp. 922–929. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.3301922. URL https://doi.org/10.1609/aaai.v33i01.3301922.

Joel Hasbrouck. Measuring the information content of stock trades. The Journal of Finance, 46(1): 179–207, 1991.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Comput., 9(8): 1735–1780, 1997. doi: 10.1162/NECO.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 3146–3154, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. Master: Market-guided stock transformer for stock price forecasting. In Proceedings of the AAAI conference on artificial intelligence, volume 38, pp. 162–170, 2024.

Hengxu Lin, Dong Zhou, Weiqing Liu, and Jiang Bian. Learning multiple stock trading patterns with temporal routing adaptor and optimal transport. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao (eds.), KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021, pp. 1017–1026. ACM, 2021. doi: 10.1145/3447548.3467358. URL https://doi.org/10.1145/3447548.3467358.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Junnan Li, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/liu25an.html.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Andrew W Lo. The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. Journal of Portfolio Management, Forthcoming, 2004.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025, pp. 19581–19588. AAAI Press, 2025. doi: 10.1609/AAAI.V39I18.34156. URL https://doi.org/10.1609/aaai.v39i18.34156.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased transformer for time series forecasting. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024, pp. 2400–2410. ACM, 2024. doi: 10.1145/3637528.3671928. URL https://doi.org/10.1145/3637528.3671928.

Mohammad Amin Shabani, Amir H. Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL https://openreview.net/forum?id=sCrnllCtjoE.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025. OpenReview.net, 2025. URL https://openreview.net/forum?id=e1wDDFmlVu.

Yanru Sun, Zongxia Xie, Emadeldeen Eldele, Dongyue Chen, Qinghua Hu, and Min Wu. Learning pattern-specific experts for time series forecasting under patch-level distribution shift. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), Advances in Neural Information Processing Systems, volume 38, pp. 91810–91844. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/8491a7fcc218946b471b600a915c8b02-Paper-Conference.pdf.

Christopher Torrence and Gilbert P Compo. A practical guide to wavelet analysis. Bulletin of the American Meteorological society, 79(1):61–78, 1998.

Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. CARD: channel aligned robust blend transformer for time series forecasting. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net, 2024. URL https://openreview.net/forum?id=MJksrOhurE.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp. 22419–22430, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Wentao Xu, Weiqing Liu, Lewen Wang, Yingce Xia, Jiang Bian, Jian Yin, and Tie-Yan Liu. HIST: A graph-based framework for stock trend forecasting via mining concept-oriented shared information. CoRR, abs/2110.13716, 2021. URL https://arxiv.org/abs/2110.13716.

Wenyan Xu, Dawei Xiang, Yue Liu, Xiyu Wang, Yanxiang Ma, Liang Zhang, Chang Xu, and Jiaheng Zhang. Finmultitime: A four-modal bilingual dataset for financial time-series analysis. CoRR, abs/2506.05019, 2025. doi: 10.48550/ARXIV.2506.05019. URL https://doi.org/10.48550/arXiv.2506.05019.

Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. Qlib: An ai-oriented quantitative investment platform. CoRR, abs/2009.11189, 2020. URL https://arxiv.org/abs/2009.11189.

Xu Zhang, Zhengang Huang, Yunzhi Wu, Xun Lu, Erpeng Qi, Yunkai Chen, Zhongya Xue, Qitong Wang, Peng Wang, and Wei Wang. Multi-period learning for financial time series forecasting. In Yizhou Sun, Flavio Chierichetti, Hady W. Lauw, Claudia Perlich, Wee Hyong Tok, and Andrew Tomkins (eds.), Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025, pp. 2848–2859. ACM, 2025. doi: 10.1145/3690624.3709422. URL https://doi.org/10.1145/3690624.3709422.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Lifan Zhao, Shuming Kong, and Yanyan Shen. Doubleadapt: A meta-learning approach to incremental learning for stock trend forecasting. In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (eds.), Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023, pp. 3492–3503. ACM, 2023. doi: 10.1145/3580305.3599315. URL https://doi.org/10.1145/3580305.3599315.

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 Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pp. 11106–11115. AAAI Press, 2021. doi: 10.1609/AAAI.V35I12.17325. URL https://doi.org/10.1609/aaai.v35i12.17325.

Tian Zhou, Ziqing Ma, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. Film: Frequency improved legendre memory model for long-term time series forecasting. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022a. URL http://papers.nips.cc/paper_files/paper/2022/hash/524ef58c2bd075775861234266e5e020-Abstract-Conference.html.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, Proceedings of Machine Learning Research, pp. 27268–27286. PMLR, 2022b. URL https://proceedings.mlr.press/v162/zhou22g.html.

Table A.1: List of notations.

Symbol	Description
$\mathbf{x}_t, \mathbf{X}_t, C_t, r_t$	Market state, input window, closing price, and one-step log-return
$y_t^{(H)}, \hat{y}_t^{(H)}, H$	Cumulative log-return target, prediction, and forecast horizon
$L_{\max}, D, p_{\text{len}}, N, d$	Maximum look-back length, input dimension, patch length, patch count, and embedding dimension
$\mathbf{E}, \mathbf{e}_i$	Patch embedding tensor and the embedding of patch i
$s_i, \tilde{s}_i, \Psi_i$	Raw patch importance, normalized importance, and cumulative importance
$K, \tau_k, \mathcal{G}_k, \ell_k$	Number of experts, CIT threshold, selected patch group, and its context length
$\mathbf{h}_k, \mathbf{z}_k$	Sequence output and pooled representation of expert k
$\mathbf{R}, R_{jk}, r_k$	Expert cosine similarity matrix, pairwise similarity, and redundancy score
α_k, w_k, λ	Raw routing confidence, CAW aggregation weight, and learnable redundancy coefficient
$\mathbf{z}_{\text{local}}, \mathbf{z}_{\text{global}}$	Aggregated local representation and global compressed representation
$\mathcal{L}, \mathcal{L}_{\text{MSE}}$	Composite training objective and MSE forecasting loss
$\mathcal{L}_{\text{ent}}, \lambda_{\text{ent}}$	Routing entropy regularization and its weight
$\mathcal{L}_{\text{div}}, \lambda_{\text{div}}, \mathbf{I}_K$	Expert diversity regularization, its weight, and the $K \times K$ identity matrix

A Appendix

A.1 Notations

Table A.1 summarizes the notations used throughout the paper.

A.2 Dataset Statistics

We conduct experiments on three categories of datasets covering stock log-return prediction, fund sales forecasting, and general time series forecasting. Table A.2 summarizes the basic statistics of all datasets used in our experiments.

(1) FinMultiTime (Xu et al., 2025). For stock log-return prediction, we use the HS300 and S&P500 subsets of FinMultiTime, a cross-market financial dataset containing daily price and volume data for 892 HS300 constituents and 4,694 S&P500 constituents from 2009 to 2024. The two subsets cover the Chinese and U.S. equity markets over a long historical period with substantially changing market conditions, providing a challenging setting for evaluating adaptive temporal context in financial return forecasting.

(2) Fund (Zhang et al., 2025). The Fund dataset contains sales records of different fund products from Ant Fortune, an online wealth management platform on the Alipay app. It consists of daily user transactions for fund subscriptions and redemptions from January 2015 to January 2023. The dataset contains 306 funds and provides a complementary financial setting for evaluating RAVEN on fund sales forecasting beyond equity return prediction.

(3) PEMS (Guo et al., 2019). We use four widely adopted traffic forecasting benchmarks, PEMS03, PEMS04, PEMS07, and PEMS08, collected from the California Department of Transportation Performance Measurement System (PeMS). These datasets contain aggregated traffic flow measurements from highway sensor networks at 5-minute intervals. Compared with financial log-returns, traffic series exhibit more regular periodic patterns, providing a complementary non-financial setting for evaluating the generalizability of adaptive context modeling.

A.3 Evaluation Metrics and Protocols

We use task-specific evaluation metrics for different forecasting settings. For stock log-return prediction, we report Mean Squared Error (MSE), Pearson correlation (Corr), and Information Coefficient

Table A.2: Dataset statistics. Ticker/Sensors denotes the number of stocks or funds in financial datasets or sensors in traffic datasets. Timepoints reports the total available data samples across all entities and time steps.

Domain	Dataset	Ticker/Sensors	Timepoints	Granularity
Financial	HS300	892	18,836,466	1-Day
	S&P500	4694	104,692,416	1-Day
	Fund	306	647,892	1-Day
Traffic	PEMS03	358	9,382,464	5-Min
	PEMS04	307	15,649,632	5-Min
	PEMS07	883	24,921,792	5-Min
	PEMS08	170	9,106,560	5-Min

Information Ratio (ICIR). For fund sales forecasting, we follow MLF (Zhang et al., 2025) and report MSE and Weighted Mean Absolute Percentage Error (WMAPE), abbreviated as WMA following the notation in the original work. For the PEMS benchmarks, we report the standard MSE and MAE metrics.

Pearson correlation (Corr). For each stock i, we compute the temporal Pearson correlation, denoted IC_i, between its predicted and realized cumulative log-returns over the evaluation period. Let $\mathcal{T}$ denote the set of evaluation dates within the period. The stock-level correlation is defined as:

$$\mathrm{IC}_i = \frac{\sum_{t\in\mathcal{T}}(\hat{y}_{t,i}-\bar{\hat{y}}_i)(y_{t,i}-\bar{y}_i)}{\sqrt{\sum_{t\in\mathcal{T}}(\hat{y}_{t,i}-\bar{\hat{y}}_i)^2}\sqrt{\sum_{t\in\mathcal{T}}(y_{t,i}-\bar{y}_i)^2}}, \tag{A.1}$$

where $\hat{y}_{t,i}$ and $y_{t,i}$ denote the predicted and realized cumulative log-returns of stock i at date t, respectively, and $\bar{\hat{y}}_i$ and $\bar{y}_i$ are their temporal means over $\mathcal{T}$. Higher correlation indicates stronger temporal agreement between predicted and realized returns for an individual stock.

Information Coefficient Information Ratio (ICIR). Given the stock-level correlations $\{IC_i\}_{i=1}^{N_s}$, ICIR is computed across the stock universe as

$$\mathrm{ICIR} = \frac{\mathrm{Mean}_{i=1,\ldots,N_s}(IC_i)}{\mathrm{Std}_{i=1,\ldots,N_s}(IC_i)}. \tag{A.2}$$

ICIR therefore summarizes both the average predictive correlation and its dispersion across stocks. A higher ICIR indicates that predictive correlations are stronger and more consistent across the stock universe.

Weighted Mean Absolute Percentage Error (WMA). For fund sales forecasting, we follow MLF (Zhang et al., 2025) and use Weighted Mean Absolute Percentage Error (WMAPE), which is abbreviated as WMA in the original work and throughout our experiments. For predictions $\hat{y}_i$ and ground-truth sales y_i, WMA is computed as:

$$\mathrm{WMA} = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{\sum_{i=1}^{n}|y_i|}. \tag{A.3}$$

Equivalently, WMA can be interpreted as the absolute percentage error of each sample weighted by its share of the total ground-truth magnitude. It measures the aggregate absolute forecasting error relative to the overall scale of observed fund sales. Lower WMA indicates better forecasting performance.

Evaluation protocols and forecast horizons. For HS300 and S&P500, we use a rolling chronological evaluation from 2020 to 2024. For each test year $Y \in \{2020, 2021, 2022, 2023, 2024\}$, data from 2010 to $Y-2$ are used for training, year $Y-1$ for validation, and year Y for testing. We predict 10-day cumulative log-returns, corresponding approximately to two trading weeks. For the year-by-year results, stock-level IC values are computed from the prediction and realization sequences within each calendar year, and ICIR is then computed across stocks within that year. For the overall 2020–2024 results, the evaluation window is extended to the full five-year test period before computing the corresponding Corr and ICIR.

For fund sales forecasting, we evaluate horizons of $H \in \{1,5,8,10\}$ days following MLF (Zhang et al., 2025). For the 5-minute PEMS datasets, we evaluate $H \in \{12,24\}$, corresponding to 1 and 2 hours ahead, respectively.

A.4 Implementation Details

We implement RAVEN in PyTorch and train all components end-to-end using the composite objective in Eq. equation 11. Unless otherwise specified, we use $K = 3$ local experts with CIT thresholds $(\tau_1, \tau_2, \tau_3) = (0.3, 0.6, 0.9)$. The redundancy coefficient λ in CAW is learnable and optimized jointly with the remaining model parameters. The routing entropy and expert diversity regularizers are weighted by λ_{ent} and λ_{div}, respectively. Algorithm 1 provides pseudocode for the overall procedure of RAVEN using the notation summarized in Appendix A.1.

Summary of key hyperparameters. The main architectural and optimization configurations are summarized below.

- **Maximum look-back window:** $L_{\max} \in \{90, 120\}$.
- **Patch length:** $p_{\text{len}} = 16$.
- **Embedding dimension:** $d = 128$.
- **Number of experts:** $K = 3$.
- **CIT thresholds:** $(\tau_1, \tau_2, \tau_3) = (0.3, 0.6, 0.9)$.
- **CAW redundancy coefficient:** λ is learnable.
- **Regularization weights:** $\lambda_{\text{ent}} = 0.1$ and $\lambda_{\text{div}} = 0.01$.
- **Batch size:** $\{512, 1024\}$.
- **Optimizer and learning rate:** $\{10^{-5}, 3 \times 10^{-5}, 10^{-4}\}$.
- **Early stopping:** 8 epochs.
- **Hardware:** single NVIDIA A800, 80GB.

A.5 Regularization Details

The adaptive routing and nested expert structure may introduce two issues during training. First, the patch importance may become highly concentrated, causing different CIT thresholds to select similar historical contexts. We refer to this behavior as routing collapse. Second, the nested contexts contain overlapping historical information, which may cause different experts to learn highly similar representations. We refer to this behavior as representation collapse. We use routing entropy regularization and expert diversity regularization to address these two issues.

Routing entropy regularization. CIT operates on the normalized patch importance distribution $\tilde{\mathbf{s}} = \{\tilde{s}_i\}_{i=1}^{N}$. Without regularization, the softmax distribution may become concentrated on a small number of patches. The cumulative importance Ψ_i then increases rapidly, causing different thresholds $\{\tau_k\}_{k=1}^{K}$ to be reached at similar positions. As a result, experts intended to capture different temporal ranges may receive contexts of similar lengths, leading to routing collapse. To reduce excessive concentration, we use the negative entropy regularizer:

$$\mathcal{L}_{\text{ent}} = \sum_{i=1}^{N} \tilde{s}_i \log \tilde{s}_i. \tag{A.4}$$

Minimizing $\mathcal{L}_{\text{ent}}$ encourages a higher entropy distribution of patch importance. This makes the cumulative importance increase more gradually and helps different CIT thresholds retain distinct context lengths. The regularizer does not impose fixed context lengths or require uniform patch importance. The importance scores remain input dependent and are learned jointly with the forecasting objective. Therefore, CIT can still adapt the historical context to each input while avoiding routing collapse.

Algorithm 1 Overall procedure of RAVEN.

Require: Input window $\mathbf{X}_t \in \mathbb{R}^{L_{\max} \times D}$; patch length p_{len}; number of experts K; ordered CIT thresholds $0 < \tau_1 < \cdots < \tau_K \leq 1$; model parameters

Ensure: Forecast $\hat{y}_t^{(H)} \in \mathbb{R}$

Input transformation

$\widetilde{\mathbf{X}}_t \leftarrow \text{RevIN}(\mathbf{X}_t)$

$\mathbf{X}_P \leftarrow \text{Patching}(\widetilde{\mathbf{X}}_t, p_{\text{len}})$ ▷ $\mathbf{X}_P \in \mathbb{R}^{D \times N \times p_{\text{len}}}$

$\mathbf{E} \leftarrow \text{LinearEmbed}(\mathbf{X}_P)$ ▷ $\mathbf{E} \in \mathbb{R}^{D \times N \times d}$

Arrange $\{\mathbf{e}_i\}_{i=1}^N$ in reverse chronological order

Cumulative Importance Thresholding

for $i = 1, \ldots, N$ **do**

  $s_i \leftarrow \phi(\mathbf{e}_i)$

end for

$\tilde{s}_i \leftarrow \dfrac{\exp(s_i)}{\sum_{j=1}^N \exp(s_j)}, \quad i = 1, \ldots, N$ ▷ $\tilde{\mathbf{s}} \in \mathbb{R}^N$

$\Psi_i \leftarrow \sum_{j=1}^i \tilde{s}_j, \quad i = 1, \ldots, N$

for $k = 1, \ldots, K$ **do**

  $\mathcal{G}_k \leftarrow \{i : \Psi_i \leq \tau_k\}$

  $\mathbf{h}_k \leftarrow \text{Encoder}_k\left(\{\mathbf{e}_i\}_{i \in \mathcal{G}_k}\right)$

  $\mathbf{z}_k \leftarrow \text{AvgPool}(\mathbf{h}_k)$ ▷ $\mathbf{z}_k \in \mathbb{R}^d$

end for

Correlation-Aware Weighting

for $j = 1, \ldots, K$ **do**

  for $k = 1, \ldots, K$ **do**

    $R_{jk} \leftarrow \dfrac{\mathbf{z}_j^\top \mathbf{z}_k}{\|\mathbf{z}_j\|_2 \|\mathbf{z}_k\|_2}$

  end for

end for

for $k = 1, \ldots, K$ **do**

  $r_k \leftarrow \sum_{j \neq k} \max(R_{jk}, 0)$

end for

Obtain raw routing confidences $\{\alpha_k\}_{k=1}^K$

for $k = 1, \ldots, K$ **do**

  $w_k \leftarrow \dfrac{\alpha_k \exp(-\lambda r_k)}{\sum_{k'=1}^K \alpha_{k'} \exp(-\lambda r_{k'})}$

end for

$\mathbf{z}_{\text{local}} \leftarrow \sum_{k=1}^K w_k \mathbf{z}_k$ ▷ $\mathbf{z}_{\text{local}} \in \mathbb{R}^d$

Global Context Representation

$\mathbf{z}_{\text{global}} \leftarrow \text{AvgPool}\left(\text{SelfAttn}(\mathbf{E})\right)$ ▷ $\mathbf{z}_{\text{global}} \in \mathbb{R}^d$

Prediction

$\hat{y}_t^{(H)} \leftarrow \text{MLP}\left(\text{Concat}[\mathbf{z}_{\text{local}}; \mathbf{z}_{\text{global}}]\right)$

return $\hat{y}_t^{(H)}$

Expert diversity regularization. The contexts constructed by CIT are nested:

$$\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \cdots \subseteq \mathcal{G}_K. \tag{A.5}$$

A longer context therefore contains the historical patches used by shorter contexts. This preserves chronological continuity, but also creates substantial overlap among expert inputs. Without an explicit diversity constraint, different experts may learn highly similar representations from these overlapping contexts. Such representation collapse reduces the complementary information provided by

Table A.3: Effect of maximum look-back window on HS300 (Corr ↑).

Year	60	90	120	150
2020	**0.0644**	0.0585	0.0567	0.0584
2021	0.0283	0.0290	**0.0292**	0.0289
2022	0.0294	0.0322	**0.0422**	0.0330
2023	0.0197	0.0198	**0.0300**	0.0204
2024	0.0430	**0.0458**	0.0379	0.0452
All	0.0361	0.0378	**0.0388**	0.0376

Table A.4: Effect of patch length p_{len} on HS300 (Corr ↑). N denotes the number of patches.

Year / N	p_{len}=8 / 15	p_{len}=12 / 10	p_{len}=16 / 7	p_{len}=20 / 6
2020	0.0543	0.0557	**0.0567**	0.0520
2021	**0.0303**	0.0287	0.0292	0.0271
2022	0.0415	0.0404	**0.0422**	0.0353
2023	**0.0315**	0.0314	0.0300	0.0289
2024	0.0490	0.0473	**0.0500**	0.0482
All	0.0384	0.0384	**0.0390**	0.0372

experts operating over different temporal ranges. Let $\mathbf{R} \in \mathbb{R}^{K\times K}$ denote the cosine similarity matrix of the expert representations:

$$R_{jk} = \frac{\mathbf{z}_j^\top \mathbf{z}_k}{\|\mathbf{z}_j\|_2 \|\mathbf{z}_k\|_2}. \tag{A.6}$$

We define the expert diversity regularizer as:

$$\mathcal{L}_{\text{div}} = \|\mathbf{R} - \mathbf{I}_K\|_F^2, \tag{A.7}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix. Since the diagonal entries of $\mathbf{R}$ represent self similarity, minimizing $\mathcal{L}_{\text{div}}$ mainly reduces the similarity between different expert representations. This encourages experts to preserve complementary information despite their overlapping historical contexts.

Relation to Correlation-Aware Weighting. The diversity regularizer and CAW both address redundancy among experts, but they operate at different stages. $\mathcal{L}_{\text{div}}$ acts during training and encourages different experts to learn less correlated representations. CAW acts when the expert representations are aggregated. For each input, CAW measures the similarity among the resulting representations and reduces the contribution of experts that contain more redundant information. Thus, $\mathcal{L}_{\text{div}}$ reduces representation collapse during training, while CAW further adjusts representation redundancy during aggregation.

Overall, $\mathcal{L}_{\text{ent}}$ and $\mathcal{L}_{\text{div}}$ address two complementary issues in the MoE architecture. The former prevents routing collapse and helps maintain distinct historical contexts across experts. The latter reduces representation collapse caused by overlapping contexts. Their individual contributions are evaluated in Table 5.

A.6 Additional Hyperparameter Analysis

In addition to the number of experts and CIT thresholds analyzed in Section 3.6, we study the sensitivity of RAVEN to the maximum look-back window $L_{\max}$ and patch length p_{len} on HS300. These parameters determine the maximum amount of historical information available to the model and the temporal granularity used to construct patches.

Maximum look-back window. We vary $L_{\max} \in \{60, 90, 120, 150\}$ days while keeping the remaining settings fixed. As shown in Table A.3, the overall Corr increases from 0.0361 at 60 days to 0.0378 at 90 days and reaches 0.0388 at 120 days. Extending the window to 150 days does not provide further improvement, with the overall Corr decreasing slightly to 0.0376. The results suggest that a sufficiently long history is useful for capturing temporal dependencies, while simply extending the available history does not necessarily improve forecasting performance. We therefore use $L_{\max} = 120$ as the default setting.

Patch length. We further vary $p_{\text{len}} \in \{8, 12, 16, 20\}$. For $L_{\max} = 120$, these settings produce 15, 10, 8, and 6 patches, respectively. Table A.4 shows that the overall Corr ranges from 0.0372 to 0.0390 across these settings, with $p_{\text{len}} = 16$ achieving the highest overall result. Using $p_{\text{len}} = 16$ as the reference, paired t-tests on per-sample MSE show no significant difference for $p_{\text{len}} = 8$, 12, or 20, with p-values of 0.26, 0.21, and 0.18, respectively. These results indicate that RAVEN is relatively robust to the patch length within the evaluated range. We use $p_{\text{len}} = 16$ as the default setting.

A.7 Backtesting Details

To evaluate whether improvements in return prediction translate into investment performance, we conduct backtesting on HS300 from January 2020 to December 2024 using Qlib (Yang et al., 2020). All forecasting models are evaluated under the same portfolio construction and execution settings so that differences in backtesting performance are driven by their prediction signals.

Portfolio construction. We construct a cross-sectional portfolio using the predicted 10-day cumulative log-returns as ranking signals. At each portfolio update, HS300 constituents are ranked in descending order according to their predicted returns. We use Qlib's TopkDropoutStrategy with $K = 30$ and $N_{\text{drop}} = 30$ to maintain a fixed-size portfolio while controlling portfolio turnover. The portfolio is updated every 10 trading days, consistent with the forecast horizon used for stock log-return prediction.

Execution settings. Portfolio execution is simulated using Qlib's SimulatorExecutor at daily frequency. The initial capital is 100 million RMB. We use the default transaction cost settings in Qlib, including commission and stamp duty. The same execution settings are applied to RAVEN and all competing methods throughout the full evaluation period.

Cumulative return comparison. Let $R_m(t)$ denote the cumulative portfolio return obtained from the prediction signal of model m up to time t. For each baseline b, Figure 3 reports the cumulative return difference

$$\Delta_b(t) = R_{\text{RAVEN}}(t) - R_b(t). \tag{A.8}$$

A positive value indicates that the portfolio constructed from RAVEN has accumulated a higher return than the corresponding baseline portfolio under the same strategy and execution settings. The persistent positive growth of these differences over the test period shows that the forecasting improvements are also reflected in the corresponding portfolio returns.

A.8 Limitations and Future Directions

RAVEN establishes adaptive historical context as an effective modeling approach for financial time series forecasting, but several challenges remain open. These limitations also point to broader directions for extending adaptive context modeling beyond the current formulation.

Joint adaptation of historical context. RAVEN adapts the amount of historical information used for each input, but the patch length remains fixed. The model therefore learns how far to look back while the temporal organization of the historical sequence is predefined. Our sensitivity analysis shows that RAVEN remains relatively robust across the evaluated patch lengths, but this does not determine which temporal representation is most appropriate for a particular input. A natural extension is to learn the historical range, together with the way that history is organized into temporal units. Such a formulation would allow the model to determine not only how much past information is relevant, but also how that information should be represented for forecasting.

Context adaptation and market regimes. CIT learns the relevant historical context directly from predictive signals without requiring explicit market regime labels. Our analysis shows that the learned patch importance and expert aggregation vary across years and stocks. However, relating these changes to the underlying market state remains challenging. Market regimes are latent, evolve over time, and may reflect multiple interacting factors rather than a single observable variable. Moreover, changes in the relevant historical context need not correspond to discrete or predefined regimes. An important direction is therefore to understand when and why the useful historical context changes as market conditions evolve, rather than simply assigning learned routing patterns to predefined regime labels. Future work could study how context selection relates to observable market characteristics such as volatility, liquidity, and cross-sectional dependence, while also investigating whether the learned context itself provides information about latent changes in market dynamics. Establishing this connection could provide a deeper understanding of time-varying temporal dependence in financial data and improve the interpretability of adaptive context modeling.

Financial foundation models. A broader direction is to extend adaptive context modeling from supervised forecasting to financial time series pretraining. Financial data combines low signal-to-noise ratios with strong non-stationarity and changing temporal dependencies. Under these conditions, a fixed historical context may not provide a consistent basis for representation learning across assets, markets, and time periods. Future financial foundation models could learn context selection jointly with the pretraining objective, allowing the historical information used for representation learning to change with the input.

This direction also raises a broader question of how pretraining should be designed for financial time series. Rather than directly adopting pretraining strategies developed for more regular time series, future work could develop financial pretraining objectives that explicitly account for changing temporal dependencies. Combined with self-supervised learning and cross-asset pretraining, this may support representations that transfer across markets, assets, and downstream financial tasks. In this view, adaptive context modeling becomes part of the representation learning principle of a financial foundation model rather than a component attached to a particular forecasting backbone.