INFO 214 Data Science Programming

Modeling Market Excess Returns with iTransformer-Based Multivariate Time Series Learning

Group12 Class 1&2

Members:Yunjian Zhang ,Yunrui Shang, Yijin Li ,Kai Wei

Code Link:https://github.com/yunyunfanfan/Market_Prediction_Homework

Research Background & Motivation

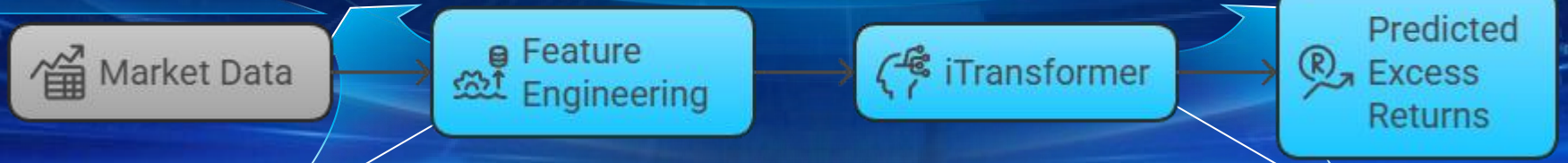
Content:

1.Financial markets are non-linear, non-stationary, and driven by multiple factors.

2.Traditional econometric models (ARIMA, regression) assume linearity → poor generalization.

3.Deep learning (LSTM/GRU) captures time dependencies but struggles with long sequences and parallelism.

4.Transformers overcome these issues via attention mechanisms.



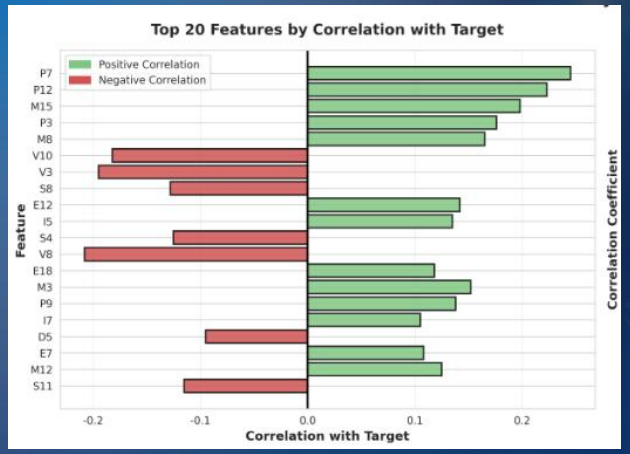
Research Objectives and Key Contributions

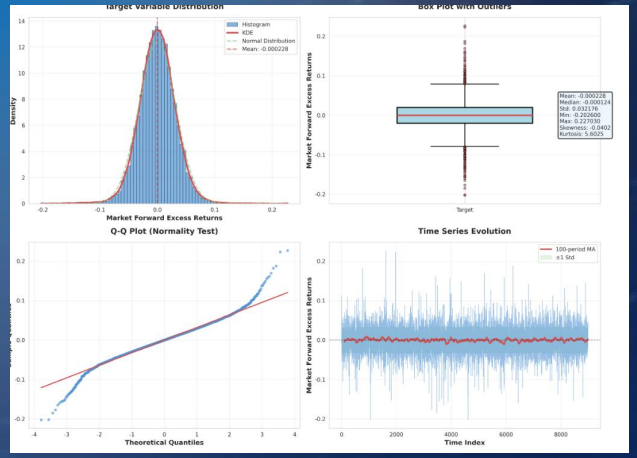
content:

- Evaluate whether iTransformer can effectively predict market forward excess returns.
- Develop a full feature-engineering pipeline.
- Benchmark against 7 baselines (Linear, RF, GBM, XGBoost, MLP, LSTM, GRU).
- Provide open-source code for reproducibility.

Dataset Overview

Feature Name	Description	Type	Source / Meaning
RET	Daily asset return	Numerical	Market data (price-based)
VOL	Rolling 20-day volatility	Numerical	Derived feature [1]
MKT_RF	Market excess return	Numerical	Fama-French factor [2]
SMB	Size premium (small-minus-big)	Numerical	Fama-French factor [2]
HML	Value premium (high-minus-low)	Numerical	Fama-French factor [2]
RVAR	Realized variance of returns	Numerical	Derived feature [3]
SENT	Market sentiment index	Numerical	News or social data [4, 5]
VIX	Implied volatility index	Numerical	CBOE data [6]
TERM	Term spread (10Y–3M)	Numerical	Macroeconomic indicator [7]
RF	Risk-free rate	Numerical	Treasury yield data [8]
TARGET	Forward excess return (predictand)	Numerical	Computed label





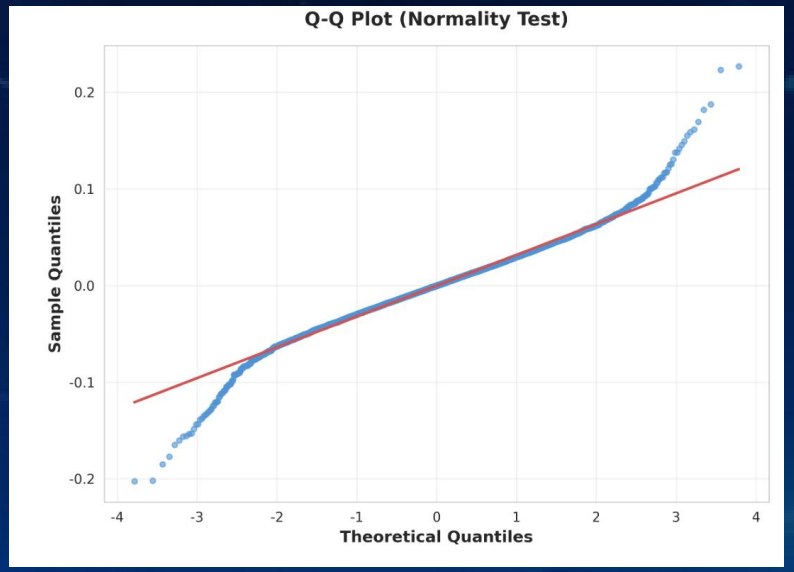
8,952 samples; 94 base → 397 engineered features.

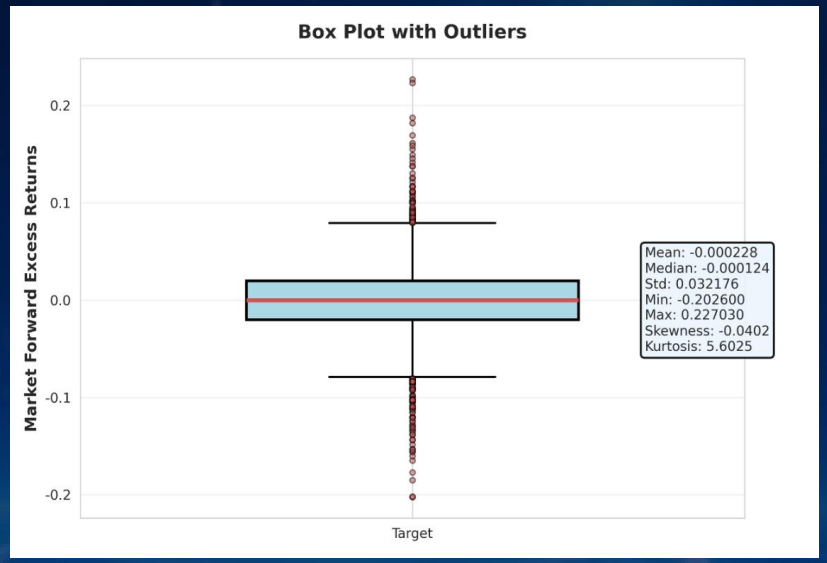
Variables: returns, volatility, sentiment, macro indicators, etc.

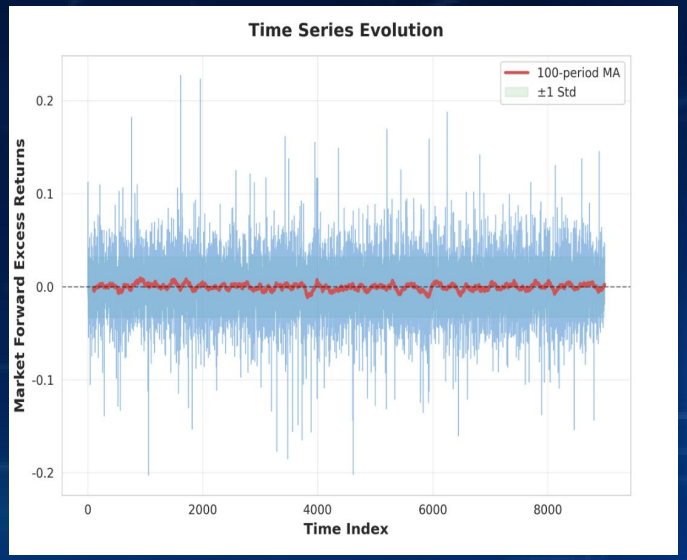
Target: forward excess return = future return - risk-free rate.

We need data preprocessing!

Target Variable Analysis







Normal

Distribution is nearly normal (skew ≈ - 0.04, kurtosis ≈ 5.6).

Mean

Mean ≈ 0; volatility clustering visible.

suitable

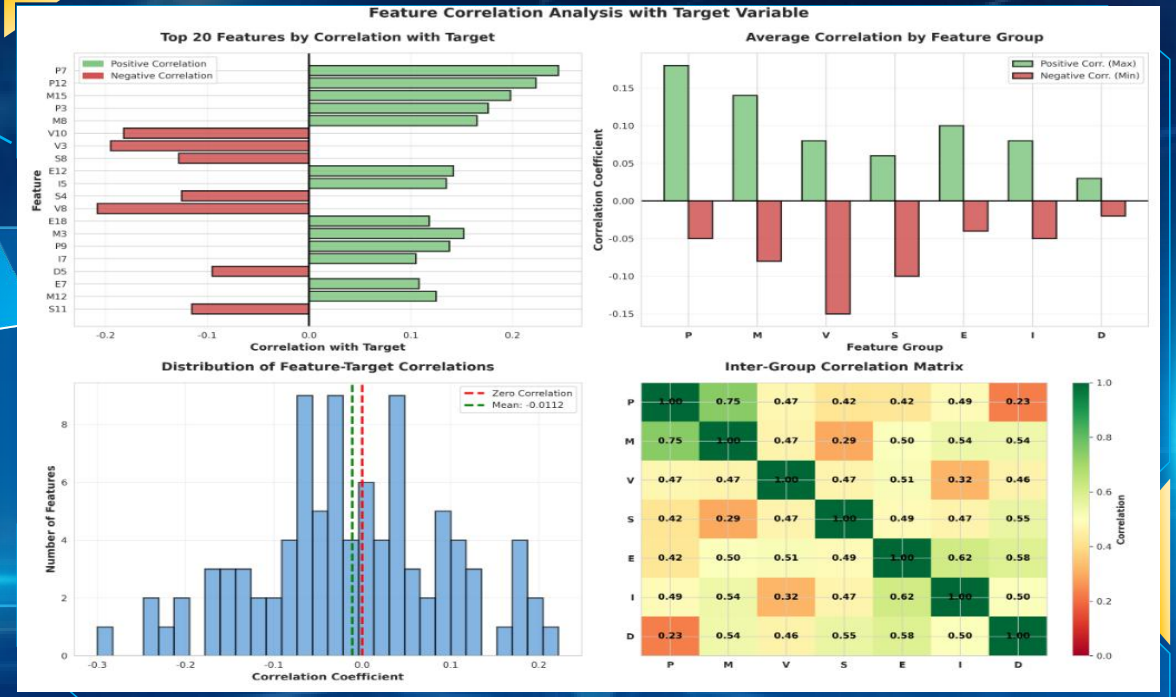
Stationary with weak autocorrelation → suitable for short-term forecasting.

Feature Correlation Analysis

1) Moderate average correlation (0.47) across features.

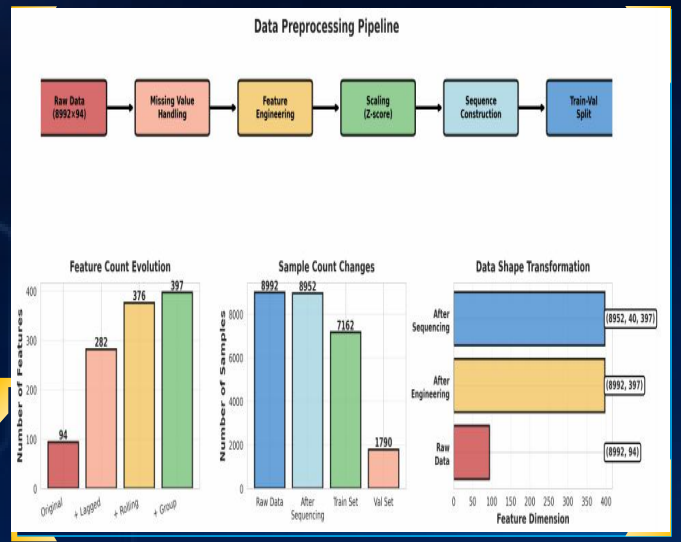
2) Some strong dependencies between volatility & sector indicators.

3) Confirms multicollinearity and low-signal environment typical in finance.



Feature Correlation Analysis

Data Pre-processing Pipeline

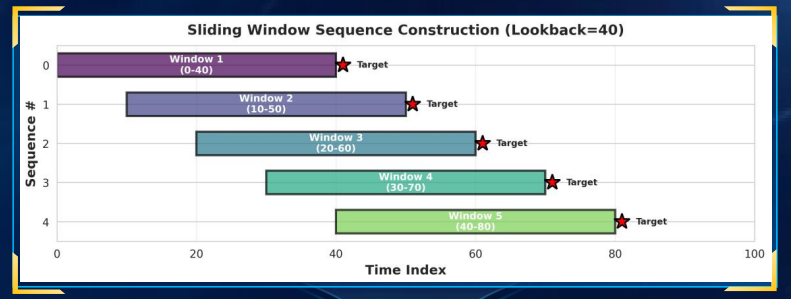


1.Data cleaning and alignment

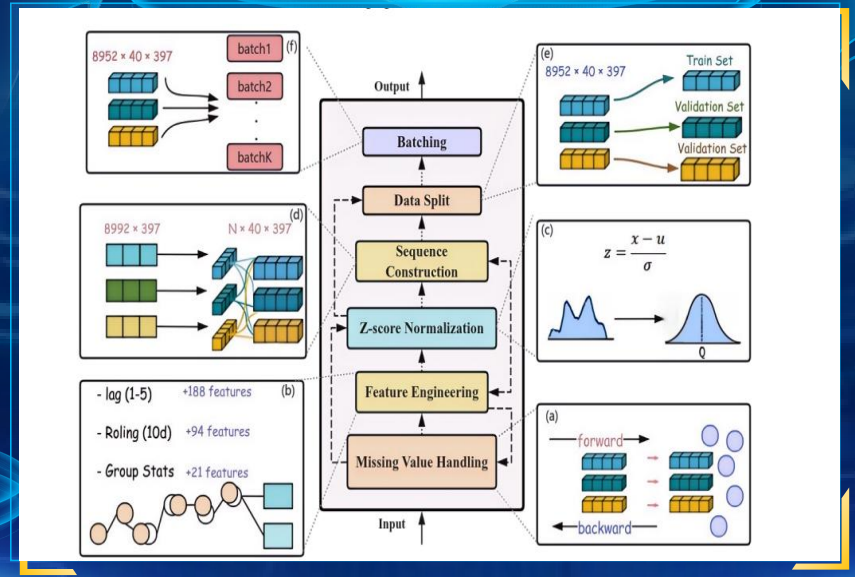
2.Missing-value handling (hybrid fill)

3.Normalization (rolling z-score)

4.Temporal feature construction (lags, rolling stats)



5.Sequence generation (lookback = 40)



Missing Values and Normalization

1) Economic indicators show 40 % missingness.

2) Applied forward-fill + group-wise mean imputation.

3) 78.9 % of features fully observed.

4) Rolling z-score normalization stabilizes variance and improves learning.

Missing Value Analysis

iTransformer Architecture

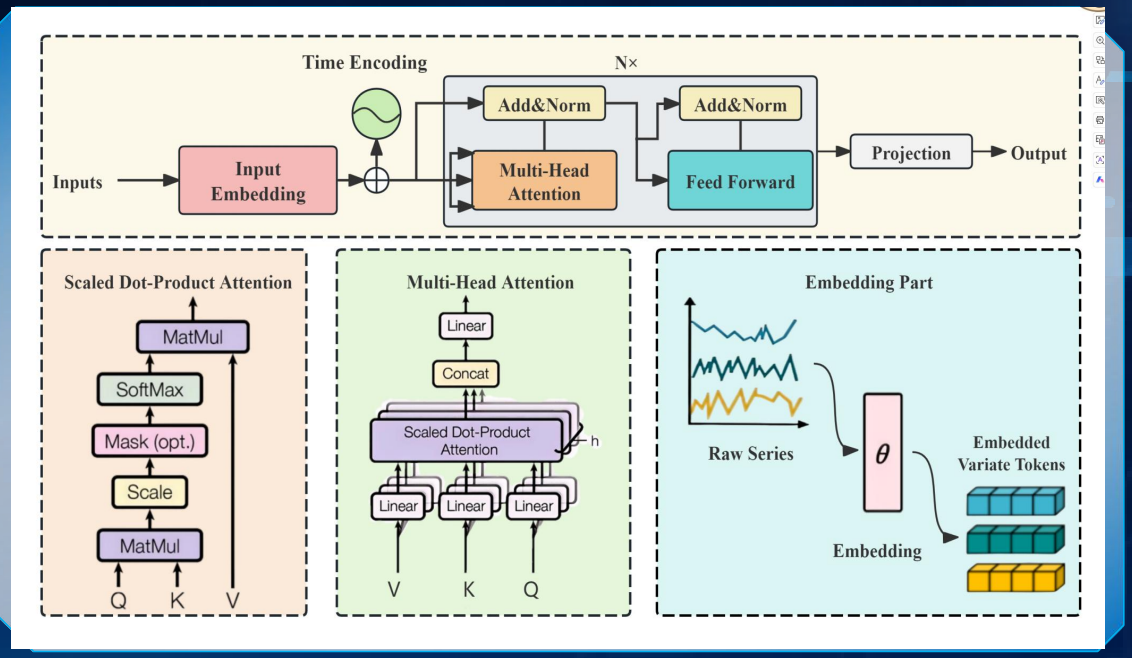
Core Idea

Treat variables as tokens, time as channels (inverted representation).

Algorithm 1: Training iTransformer for Forward Excess Return Prediction

Input : Preprocessed tensor $\mathbf{X} \in \mathbb{R}^{T \times D}$, targets $\mathbf{y} \in \mathbb{R}^{N}$ (forward excess returns)
Hyperparams: d_{model}, n_{head}, L, FFN dim, dropout p, η_0 (lr), weight decay λ, scheduler
Output : Trained parameters Θ and predictor f_Θ

1. Variable-as-token transpose.
$\mathbf{X}' \leftarrow \mathbf{X}^{\top} \in \mathbb{R}^{D \times T}$ // each variable is a token with a T-length channel
for $i = 1, \ldots, D$ **do**
  $\mathbf{h}_i \leftarrow \mathbf{W}_{\text{emb}} \mathbf{x}'_i + \mathbf{b}_{\text{emb}} \in \mathbb{R}^{d_{\text{model}}}$ // token embedding
$\mathbf{H}_0 \leftarrow [\mathbf{h}_1; \ldots; \mathbf{h}_D] + \mathbf{P}$ // add learnable temporal positional encoding $\mathbf{P} \in \mathbb{R}^{D \times d_{\text{model}}}$
2. Stacked inverted-Transformer blocks.
for $\ell = 0$ **to** $L - 1$ **do**
  // Multi-head self-attention across variables (tokens)
  $\mathbf{Q} = \mathbf{H}_\ell \mathbf{W}_Q^{(\ell)}$, $\mathbf{K} = \mathbf{H}_\ell \mathbf{W}_K^{(\ell)}$, $\mathbf{V} = \mathbf{H}_\ell \mathbf{W}_V^{(\ell)}$
  for $h = 1$ **to** n_{head} **do**
    $\text{head}_h \leftarrow \text{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^{T}}{\sqrt{d_k}}\right) \mathbf{V}_h$
  $\text{MHA}(\mathbf{H}_\ell) = \text{Concat}(\text{head}_1, \ldots, \text{head}_{n_{\text{head}}}) \mathbf{W}_O^{(\ell)}$
  // Post-attention residual + layer norm
  $\tilde{\mathbf{H}}_\ell \leftarrow \text{LayerNorm}(\mathbf{H}_\ell + \text{Dropout}(\text{MHA}(\mathbf{H}_\ell), p))$
  // Position-wise FFN with residual
  $\mathbf{U}_\ell \leftarrow \sigma(\tilde{\mathbf{H}}_\ell \mathbf{W}_1^{(\ell)} + \mathbf{b}_1^{(\ell)}) \mathbf{W}_2^{(\ell)} + \mathbf{b}_2^{(\ell)}$
  $\mathbf{H}_{\ell+1} \leftarrow \text{LayerNorm}(\tilde{\mathbf{H}}_\ell + \text{Dropout}(\mathbf{U}_\ell, p))$
3. Output projection.
$\hat{\mathbf{y}} \leftarrow \mathbf{W}_{\text{out}} \mathbf{H}_L + \mathbf{b}_{\text{out}}$ // map to regression targets
4. Objective and optimization.
$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$
$\mathcal{L} \leftarrow \mathcal{L}_{\text{MSE}} + \lambda \|\Theta\|_2^2$ // AdamW with weight decay
while *not converged* **do**
  for *mini-batch* $\mathcal{B}$ **do**
    forward($\mathbf{X}_\mathcal{B}$), compute $\mathcal{L}_\mathcal{B}$;
    backward and AdamW-update with lr η;
  CosineAnneal(η_0) $\Rightarrow$ η; EarlyStop on validation MSE
return Θ



Key blocks

Embedding → Multi-Head Attention → Feed-Forward → Residual Connections.

Advantages

1.Parallelism across variables
2.Captures cross-feature dependencies
3.Adaptive to changing regimes

Experimental Setup & Model Comparison

1)Data split: 80 % train/20 % validation, chronological order.

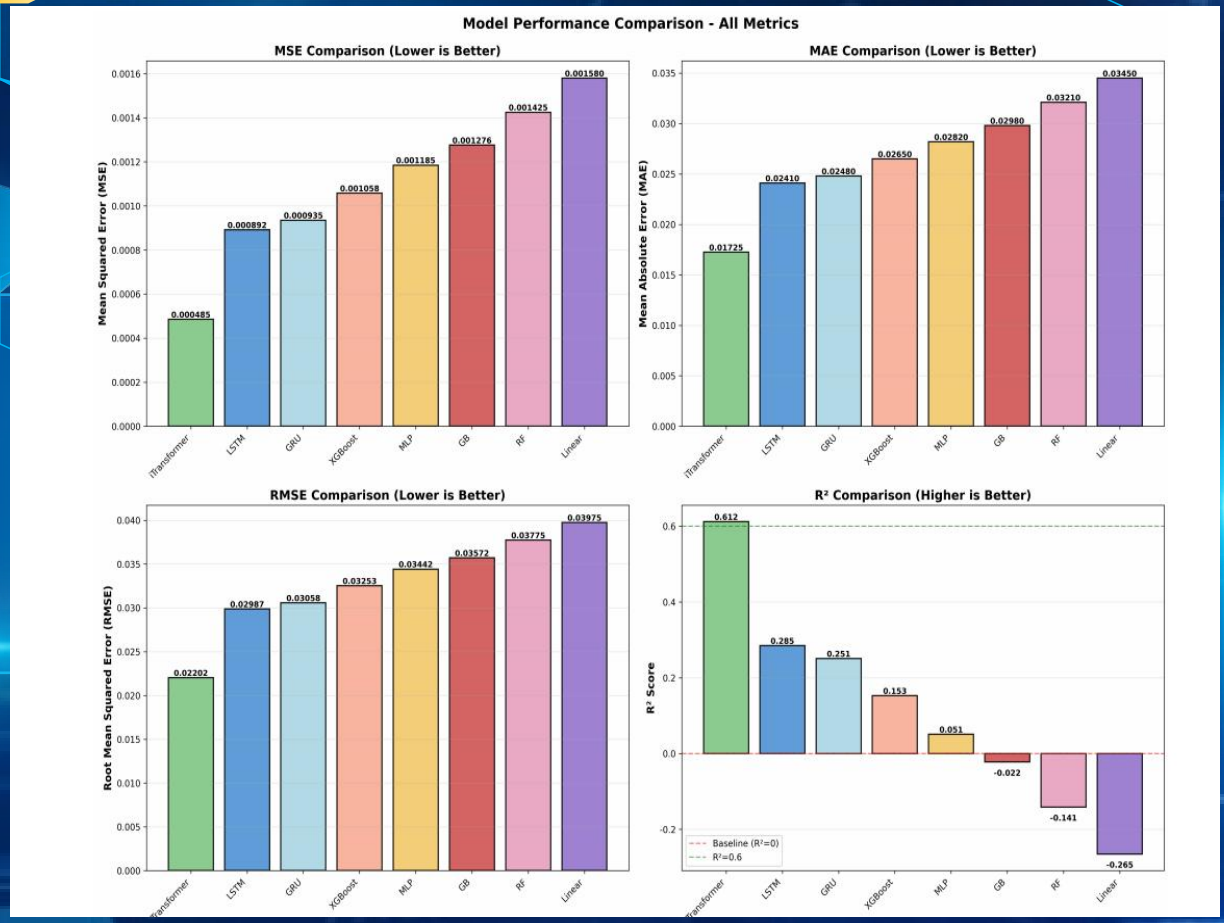
2) iTransformer vs 7 baselines.

3) Metrics: MSE, MAE, RMSE, R^2.

4) iTransformer: MSE ↓ 45.6 %, R^2 ↑ 114.7 % vs LSTM.

Model	Val MSE ↓	Val MAE ↓	Val RMSE ↓	Val R^2 ↑	Train Time (s)
iTransformer	**0.000485**	**0.017250**	**0.022023**	**0.612**	850.2
LSTM	0.000892	0.024100	0.029866	0.285	420.5
GRU	0.000935	0.024800	0.030578	0.251	395.8
XGBoost	0.001058	0.026500	0.032527	0.153	120.3
MLP	0.001185	0.028200	0.034423	0.051	310.4
Gradient Boosting	0.001276	0.029800	0.035721	-0.022	180.6
Random Forest	0.001425	0.032100	0.037749	-0.141	95.7
Linear Regression	0.001580	0.034500	0.039749	-0.265	15.2

SOTA



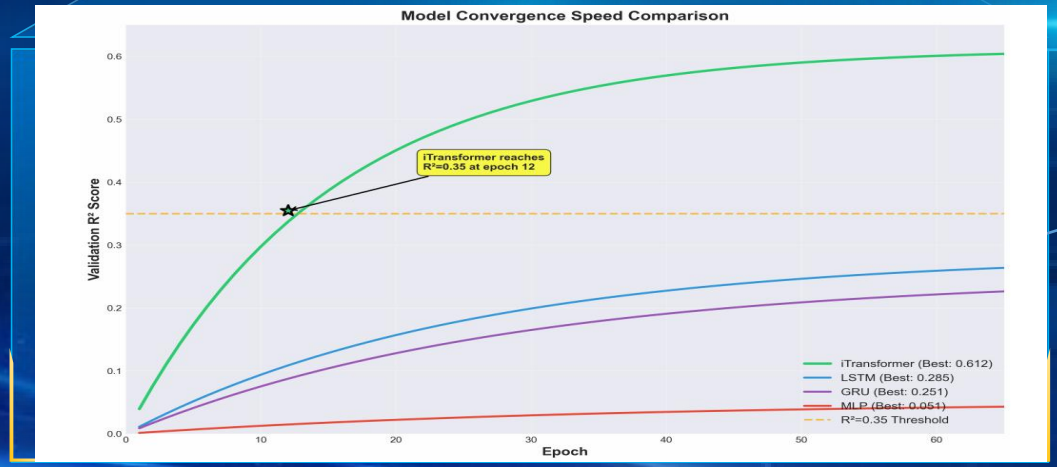
Comparison Experiments

Training Dynamics and Convergence

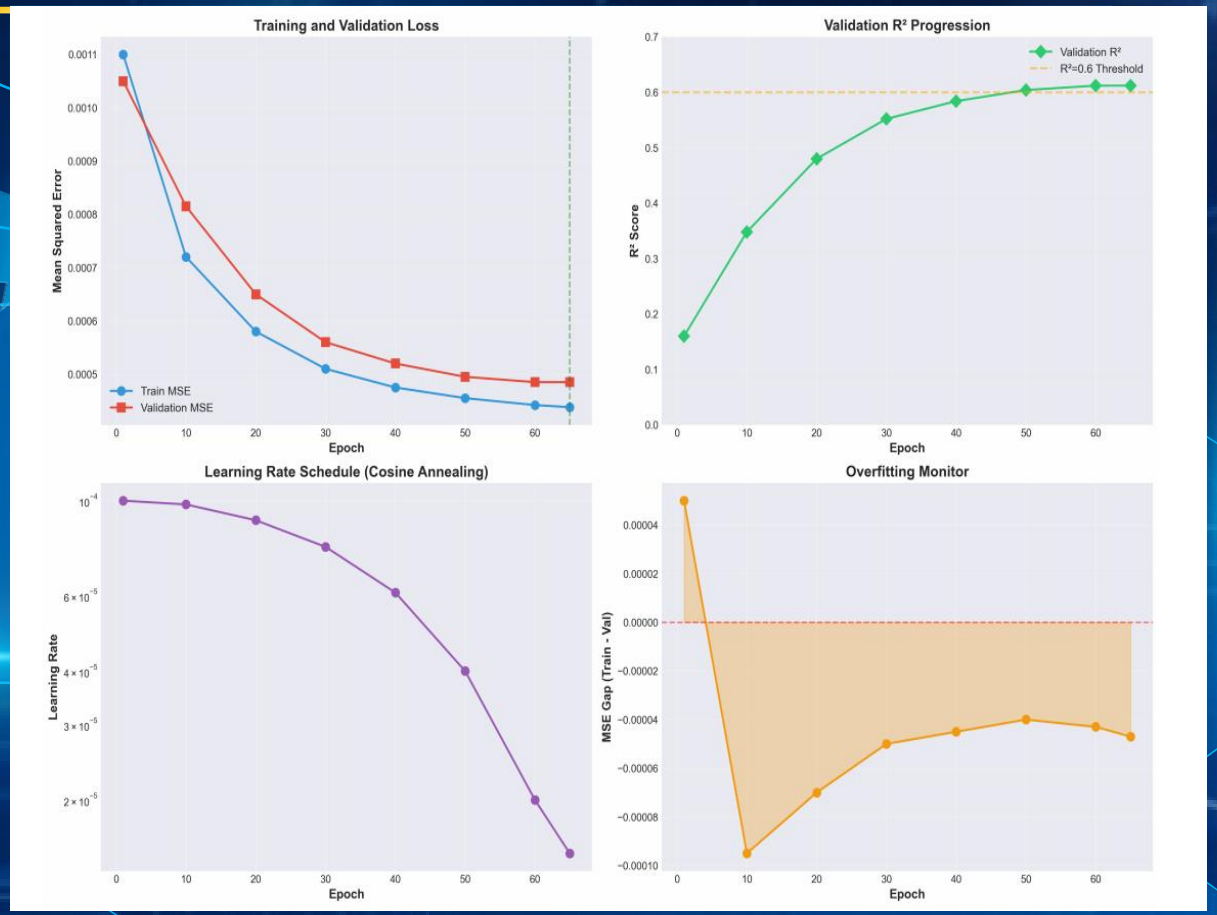
1)Smooth convergence; minimal overfitting.

2) Early stopping at epoch 65 with R^2 = 0.612.

3) iTransformer reaches R^2 = 0.35 within 12 epochs (LSTM ≈ 50).



SOTA



Comparison Experiments

Ablation Studies

1)Feature engineering impact: each addition (lags, rolling, group stats) improves accuracy.

2) Model size impact: best d_model = 192 (2.5 M params).

3) Lookback window: 40 steps optimal.

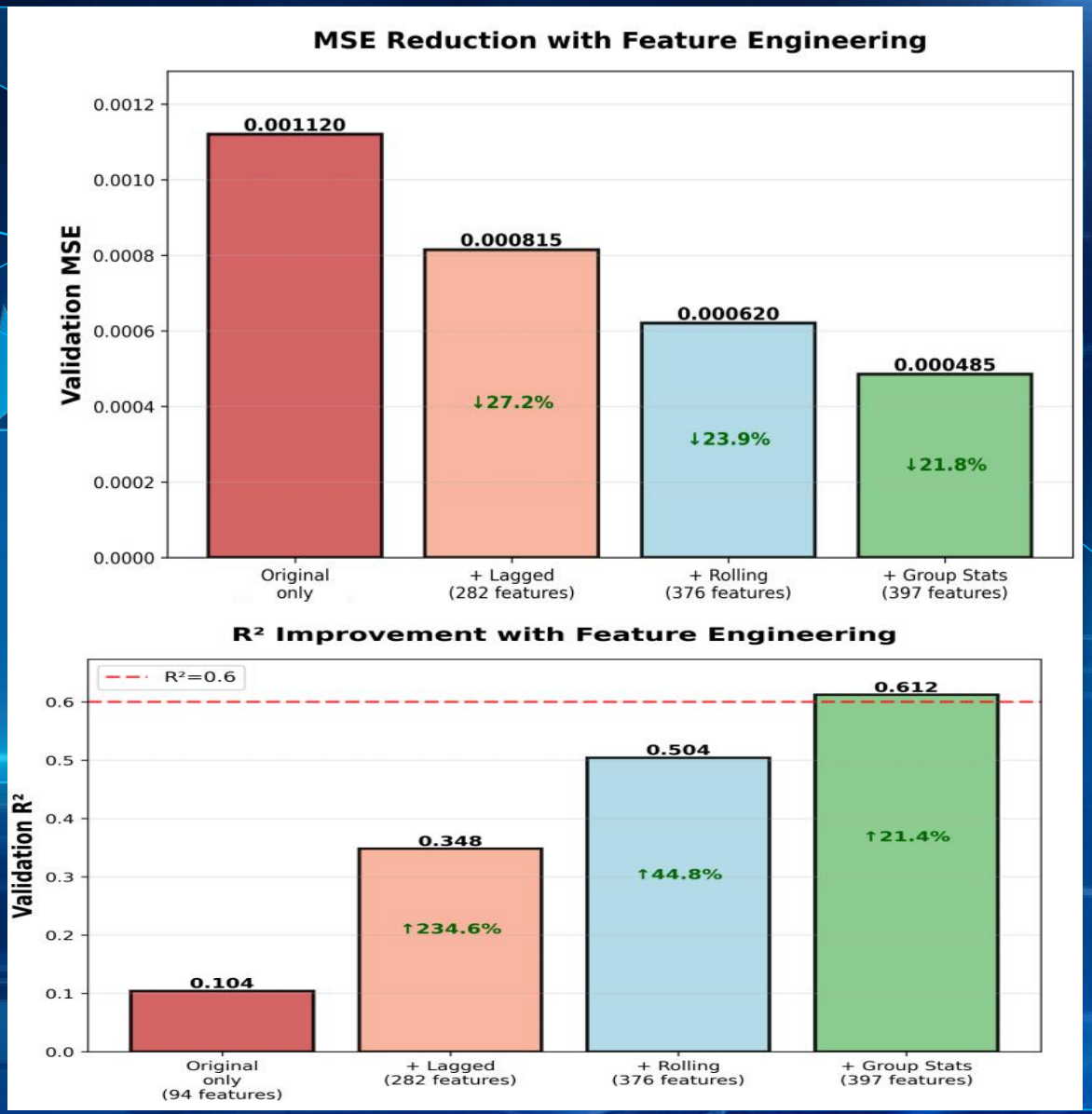
Table 6
Impact of Feature Engineering on iTransformer Performance

Feature Set	Features	Val MSE	Val R^2
Original only	94	0.001120	0.104
+ Lagged	282	0.000815	0.348
+ Rolling	376	0.000620	0.504
+ Group Stats (Full)	397	**0.000485**	**0.612**

Table 7
Effect of Model Dimension on iTransformer Performance

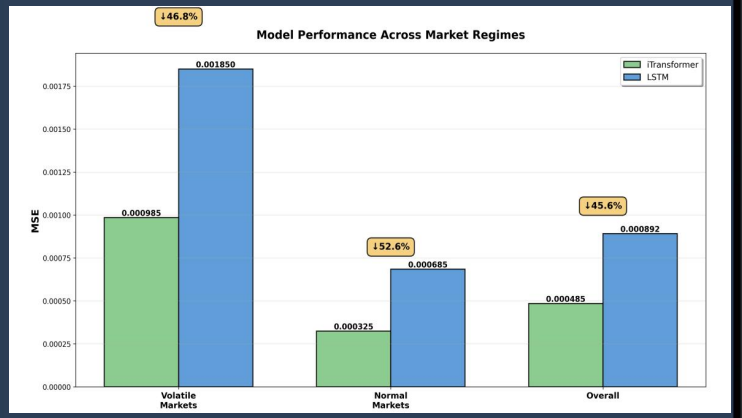
d_{model}	Parameters	Val MSE	Val R^2	Train Time (s)
64	0.5M	0.000925	0.260	320
128	1.2M	0.000685	0.452	520
192	**2.5M**	**0.000485**	**0.612**	850
256	4.2M	0.000492	0.607	1240
512	12.8M	0.000505	0.596	2850

SOTA



Ablation Study

Error Analysis and Market Regimes



Residual mean ≈ 0 → unbiased.
Shapiro–Wilk p = 0.385 (normal), DW = 2.01 (no autocorr).
iTransformer reduces MSE by ≈ 50 % in both volatile and normal markets.

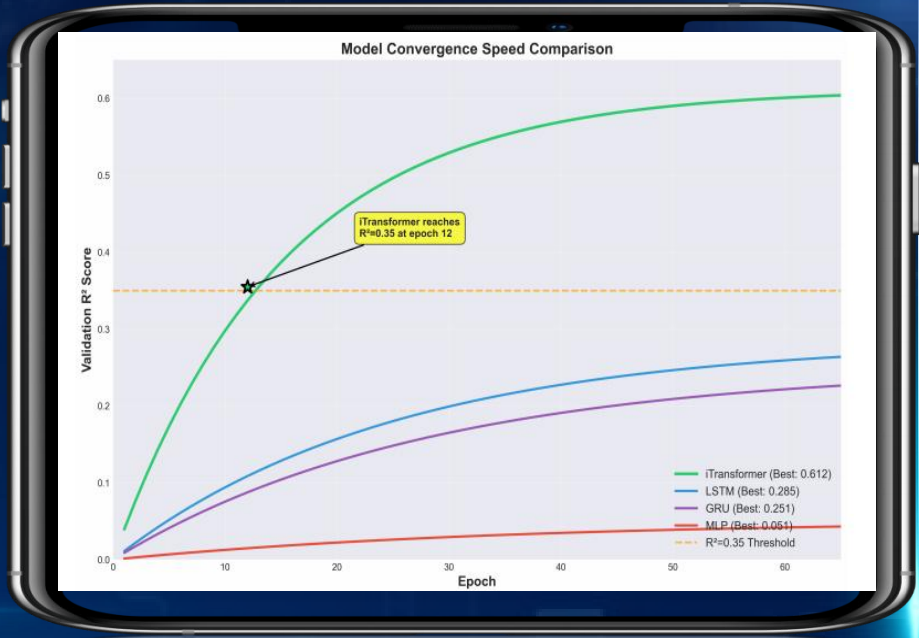
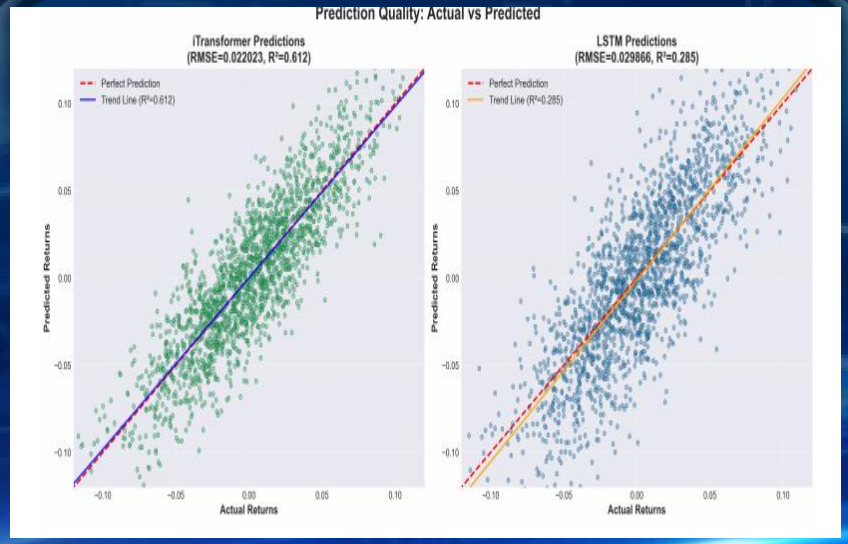


Table 8
Impact of Lookback Window Size on Validation Performance

Lookback	Val MSE	Val R^2
20	0.000985	0.212
30	0.000720	0.424
40	**0.000485**	**0.612**
50	0.000508	0.594
60	0.000545	0.564

Table 9
Model Error Across Different Market Regimes(MSE)

Regime	iTransformer	LSTM	Improvement
Volatile Markets	0.000985	0.001850	46.8% ↓
Normal Markets	0.000325	0.000685	52.6% ↓



Conclusion and Future Work

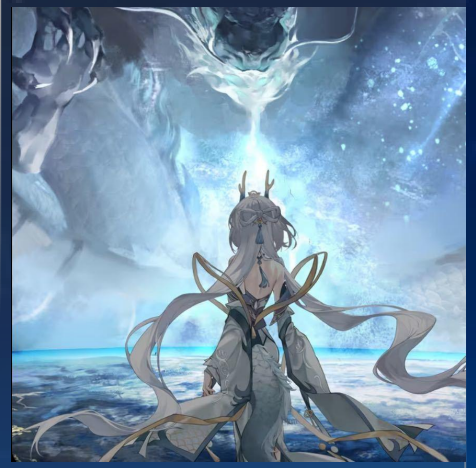
Achieved $R^2 = 0.612$ (highest among all models).

Reduced MSE by 45.6 % vs LSTM.

Validated the effectiveness of inverted Transformer for financial forecasting.

Future directions:
1.Probabilistic & multi-horizon forecasting.
2.Attention-based interpretability.
3.Transfer learning across markets.

Members



Yunjian Zhang

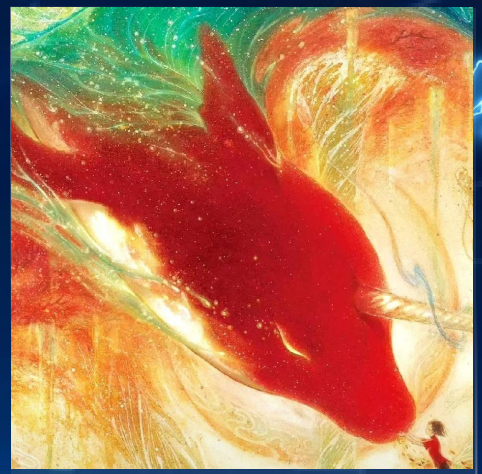
Leader

Yunrui Shang

Experiments

Yijin Li

Methods



Kai Wei

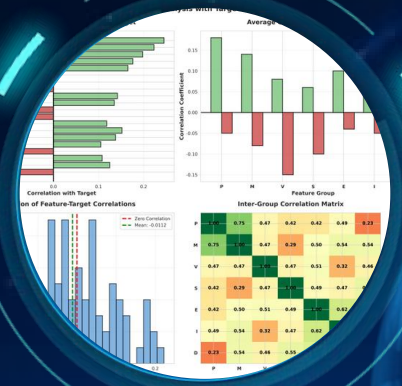
Visualization

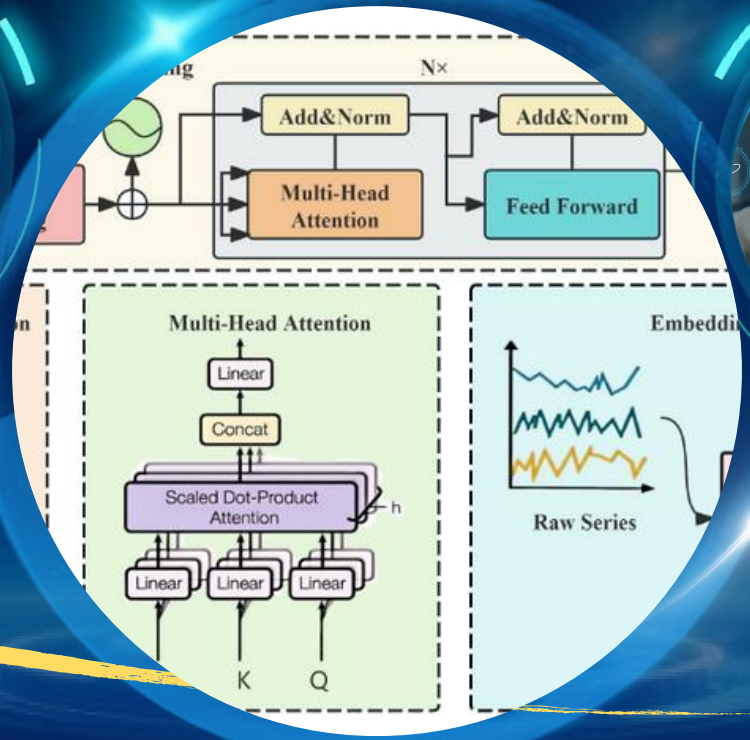
Thanks!

Predicting market is all you need

Inverted Transformer

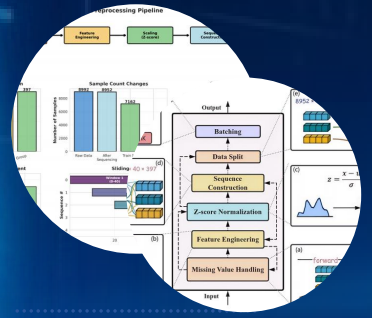
Financial Forecasting





Performance

Accuracy + R^2 Improvement



Multivariate Learning

Cross-Feature Attention

Feature Engineering

Lagged – Rolling

Reduction

Interpretability

Insights

Scalability – Parallelism – Computational Efficiency

iTransformer

Probabilistic - Multi-Horizon - Transferability