

## 1. 学界和业界的目標差異

- 學術關注解釋異常(abnormal): 什麼因子解釋了橫截面的收益率?

$r_{i,t+1} = g(f_{i,t}) + \varepsilon_{i,t+1}$ ,  $f_{i,t}$  為橫截面因子,  $r_{i,t+1}$  為下一期收益率, 最小化  $MSE = \frac{1}{N} \sum_i (r_{i,t+1} - \hat{r}_{i,t+1})^2$  是“自然”的

- 投資者關注超额收益(alpha): 什麼因子能預測超额收益, 因子选股是否能跑贏市場? 模型是否需要準確貼合絕對收益(minimize MSE/MAE) ?

【從純多頭的角度, 學習Label的排名分布更重要!】

MSE 偏大, 但對於前20%股票預測更準確的模型 >> MSE偏小, 而僅對後60%股票預測更準確的模型

## 2. 突破MSE/MAE (預測值精度 → 排序分布), Loss Function優化為: - 預測值與Label的加權相關係數

Weighted Pearson Correlation Coefficient

對真實收益 $r_{i,t+1}$ 按降序排名, 第  $j$  名的權重設為  $w_{(j)} = 0.5 \frac{j-1}{N-1}$

```
_, argsort = torch.sort(label, descending=True, dim=0)
weight = torch.zeros_like(pred)
weight_new = torch.tensor([0.5 ** ((j - 1) / (pred.shape[0] - 1)) for j in range(1, pred.shape[0] + 1)],
                           device=pred.device).unsqueeze(dim=1 if pred.ndim > 1 else 0)
weight[argsort] = weight_new.squeeze()
wcov = (pred * label * weight).sum() / weight.sum() - \
        (pred * weight).sum() / weight.sum() * (label * weight).sum() / weight.sum()
pred_std = torch.sqrt(((pred - pred.mean()) ** 2 * weight).sum() / weight.sum() + 1e-8)
label_std = torch.sqrt(((label - label.mean()) ** 2 * weight).sum() / weight.sum() + 1e-8)
loss = -(wcov / (pred_std * label_std))
```

## 1. Validation Set评估指标

交易者真正关心的不是预测误差，而是可兑现的  $\alpha$ 。

- 如果一只股票当前平均每天只能成交几百万元，我们却假设买入几千万元，那么纸面  $\alpha$  无法落地，甚至可能因为冲击成本变成负收益。
- 因此，用一笔真实资金规模（1.5 亿元）+ 流动性限额来“实盘化”模型分数
- 将这些资金按照股票流动性（总交易量）买入模型打分的前 $n$ 名，得到的流动性加权收益作为Validation Set的评估分数

## 2. 在实际交易的情境下，考虑流动性的影响，优化加权相关系数Loss Function

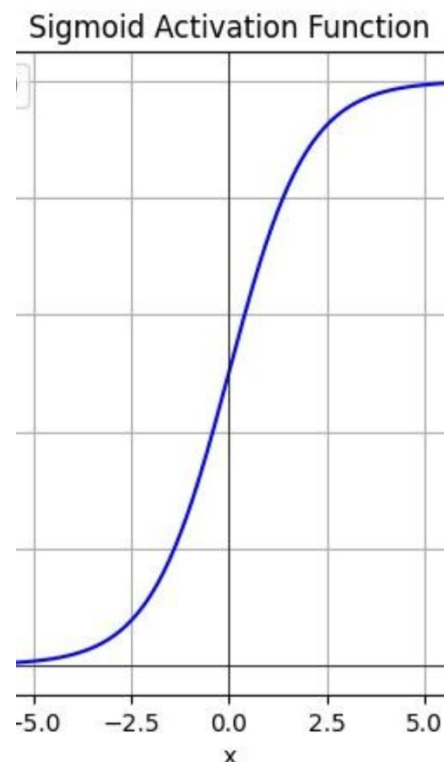
### LIQ-WPCC

- 首先对每只股票的流动性做对数化+Min-Max归一化得到 $l_i$
- $w_i = 0.5 + 0.5\sigma(\alpha l_i r_{i,t+1})$ ,  $\sigma(\cdot)$ 为Sigmoid,  $\alpha$ 取40(将数据范围缩放至 $[-4, 4]$ 之间)
- 高收益且流动性好  $\rightarrow w_i$  接近1；收益/流动性任一较差  $\rightarrow w_i$  接近0.5，减少学习权重

## 3. input\_size的升维: (batch\_size, feature) $\rightarrow$ (batch\_size, stock, feature)

一个样本为：某一天全市场所有股票的特征矩阵

- 损失函数要在同一天、不同股票之间计算相关性和排序；
- 只有把一天所有股票同时送进网络，梯度才能正确反映“谁该排在谁前面”
- 后续加入时序attention，样本dimension  $\rightarrow$  (batch\_size, time\_step, stock, feature)



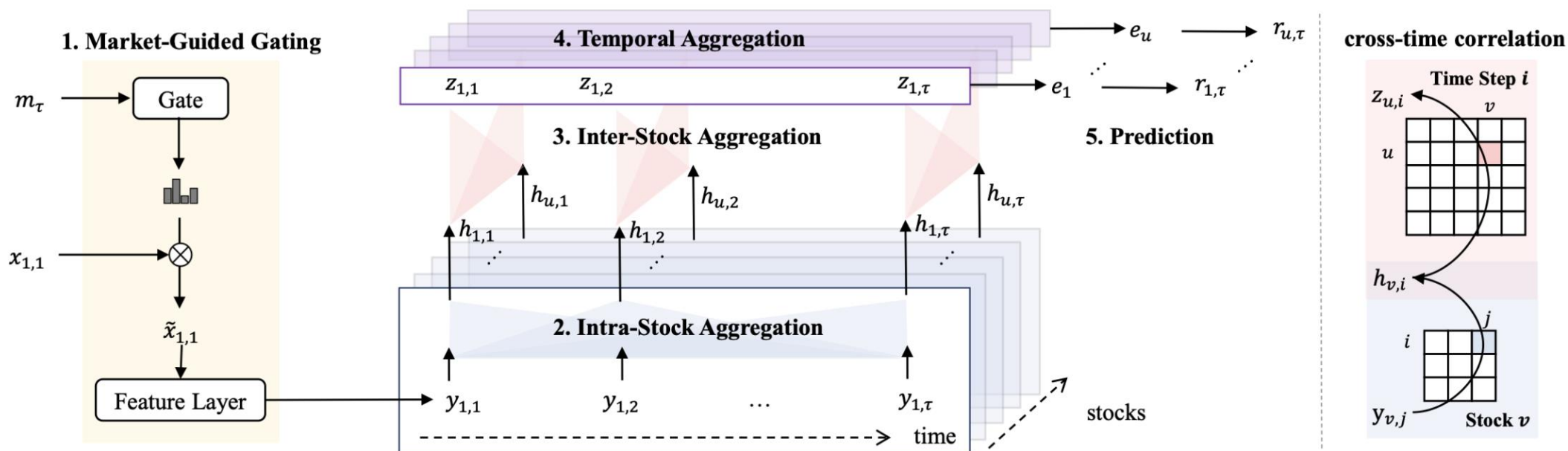
# 超越截面Baseline [3层MLP]

截面 → 时序-截面混合

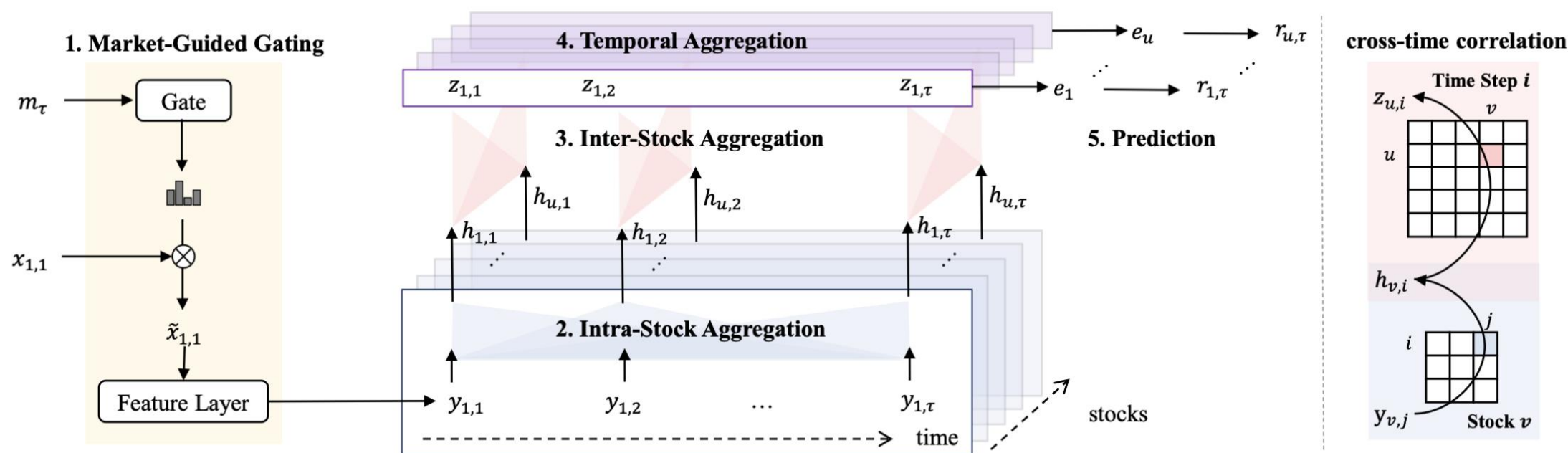
- 原有截面模型的局限: 单时点、单股票视角, 只关注  $r_{i,t+1} = g(f_{i,t}) + \varepsilon_{i,t+1}$ , 在真正多变的 market 里, 这种模型只能学到 “一张静态照片” 的规律
- 传统的 RNN/LSTM等时序模型将一条序列最后的隐藏层作为股票的 representation, 再做图或注意力聚合, 结果把真正发生在[某股 × 某时刻]的局部共振模糊掉

## AI会议论文模型复现(去年的还算热乎)

Li, T., Liu, Z., Shen, Y., Wang, X., Chen, H., & Huang, S. (2024, March). Master: Market-guided stock transformer for stock price forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 1, pp. 162-170).



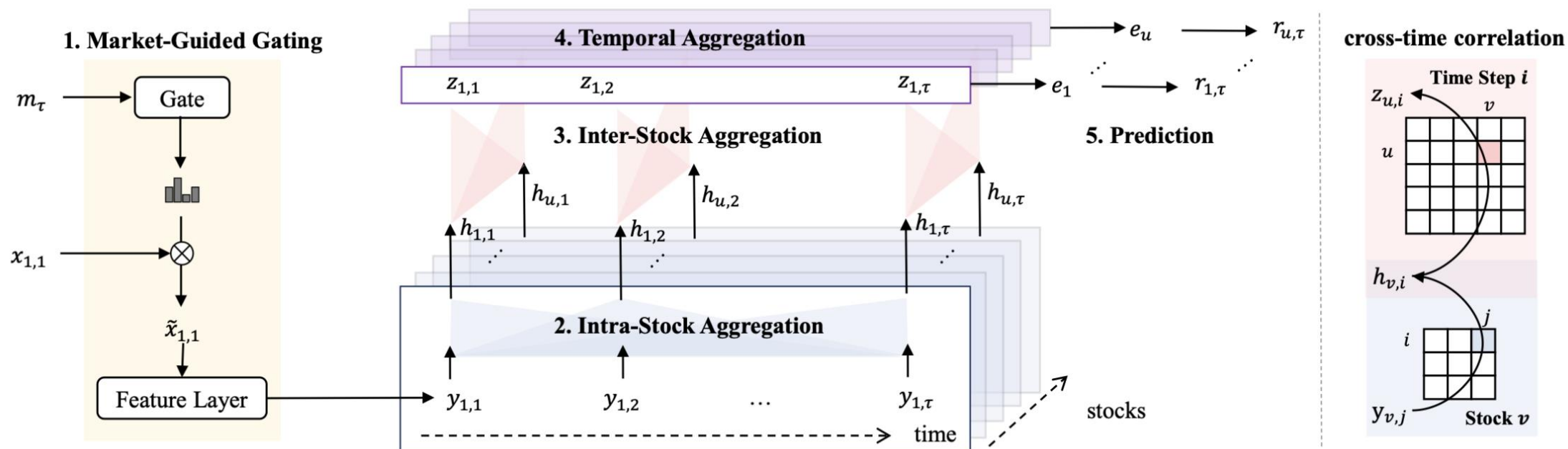
# Master(Stock Transformer)



- 1. Intra-Stock Aggregation:** 在单只股票的时间序列中，每个时间步聚合其他时刻信息生成  $h_{u,t}$ 。该嵌入既保留个股时序局部特征，又整合时间轴上的关键信号，将作为中继节点在后续模块向其他股票传递信息
- 2. Inter-Stock Aggregation:** 在每个时间步，attention 计算股票间的相关性，每只股票进一步聚合其他股票的局部嵌入表征。最终生成的聚合信息  $z_{u,t}$  不仅包含当前时刻  $t$  的瞬时关联股票信息，同时完整保留了股票  $u$  的自身时序信息
- 3. Temporal Aggregation:** 对于每只股票，以最新时序嵌入作为 query，time window 中所有时序嵌入作为 key，value 计算 attention score，并加权求和生成综合股票嵌入  $e_u$

最难的难点： 每日可交易的股票数量不同，股票/时序维度对不齐  $\rightarrow$  大量 Padding, Masking 及相关的后续操作来保证每个 batch 维度必须对齐， 但原文使用中证 300/800 的指数成分股，股票数量固定，好做很多

# Master(Stock Transformer)



对于Gating的尝试——市场状态向量通过门控单元进行Feature Selection

市场状态向量的构建: Index price过去d天的均值和标准差, Index Trading Volume/过去六日的均值/标准差等等  
 给定市场状态表示向量  $m_t$ , 其维度为  $|m_t| = F'$ , 我们首先使用一个单一的线性层将  $m_t$  转换到特征维度  $F = |x_{u,t}|$ 。然后, 沿着特征维度执行 Softmax 操作, 得到  $\alpha(m_t)$ 之后与feature做点乘, 实现特征缩放(选择)

$$\alpha(m_\tau) = F \cdot \text{softmax}_\beta(W_\alpha m_\tau + b_\alpha),$$

但是效果不好, 遂放弃, 原因应该是本任务样本与原文样本的差异, CSI300指数对其成分股的影响相较于全A股更加显著, 后续考虑纳入更多的宏观变量作为Gating的input:

模型训练只有两年, 优先考虑日频/月频的宏观变量 (相对“高频”的变量有足够的variation能和因子进行交互, 而低频变量比如季频变量, 数据集只能覆盖到4-5个值, 提供不了太多信息增量)。目前思路是:

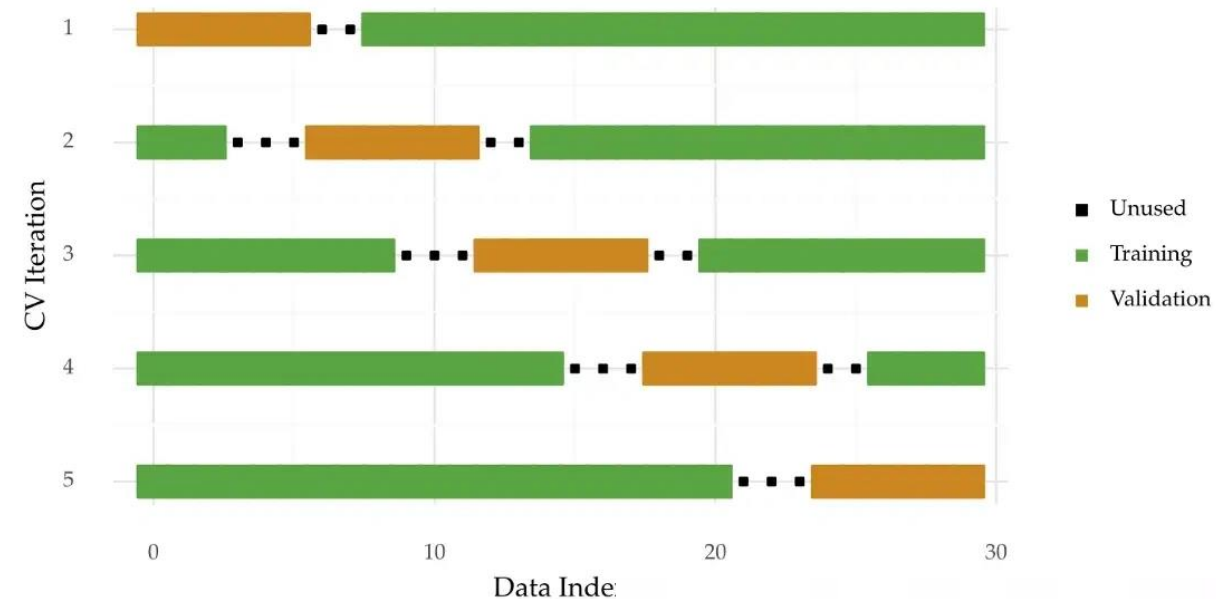
日频: 汇率, 国债收益率, 股指收益率, 利率, 大宗商品价格 (国际原油/金属等)



# Training Result

Data Leakage? 为什么不是Rolling训练

Data Partitions with hvBlockedKFold



Mean IC: 0.1028  
Mean Return: 22.51%  
Annual Return: 71.65%  
Annual Volatility: 8.95%  
Sharpe Ratio: 8.00  
Max Drawdown: -5.90%

