Hybrid Neural-Classical Correction for Frozen Time Series Foundation Models: A Comprehensive Ablation Study on High-Frequency Stock Prediction

1st Kasun Dewage
Dept. of Mathematics
University of Central Florida
Orlando, FL, USA
KasunTharuka.Dewage@ucf.edu

2nd Suranadi De Silva
Dept. of Computer Science
University of Central Florida
Orlando, FL, USA
su966204@ucf.edu

3rd Shankhadeep Mondal
Dept. of Mathematics
University of Central Florida
Orlando, FL, USA
shankhadeep.mondal@ucf.edu

***Abstract*—Foundation models for time series forecasting demonstrate impressive zero-shot generalization but often underperform on specialized domains such as high-frequency finance. We present a comprehensive study of *hybrid neural-classical correction* for adapting frozen TimesFM (200M parameters) to stock return prediction during the volatile opening trading hour. We compare two neural correction architectures—AttnCorrect (multi-head self-attention, ∼471K parameters) and GatedLinear (low-rank bilinear projection with gating, ∼49K parameters)—each augmented with Random Forest residual learning. Through systematic ablation across 10 major technology stocks (NVDA, MSFT, AAPL, GOOG, GOOGL, AMZN, META, AVGO, TSLA, NFLX) spanning 2 million data points, we reveal critical insights: (1) The hybrid neural-classical approach achieves 0.597 pooled correlation and 6.4× mean per-day correlation improvement over frozen TimesFM; (2) Classical residual learning (Random Forest) provides the largest single-component contribution , matching or exceeding the neural correction component; (3) Simpler neural architectures surprisingly outperform complex ones when classical residual learning is removed; (4) Self-attention provides the largest neural-only contribution. GatedLinear+RF achieves best overall performance with 9× fewer neural parameters than AttnCorrect+RF. We report three complementary correlation metrics—mean per-day, cross-day cumulative, and pooled—to provide a complete picture of predictive quality. Our results provide practical guidance: effective foundation model adaptation requires careful integration of neural and classical components, with classical methods playing a crucial complementary role.**

***Index Terms*—Foundation Models, Time Series Forecasting, Hybrid Methods, Neural-Classical Integration, Random Forest, Attention Mechanisms, Stock Prediction, Model Adaptation**

I. Introduction

Foundation models have revolutionized machine learning across domains. Recent extensions to time series—including TimesFM [5], Chronos [1], and Lag-Llama [20]—demonstrate that models pretrained on billions of time points can achieve strong zero-shot generalization across diverse forecasting tasks.

However, zero-shot performance does not guarantee domain-specific accuracy. When applying TimesFM to high-frequency stock prediction, we observe **near-zero mean per-day correlation** (0.059) with actual returns—essentially uninformative predictions. This gap between general capability and domain-specific performance motivates investigation into effective adaptation strategies.

We focus on a challenging prediction task: forecasting stock returns during the **opening trading hour** (9:30–10:30 AM), when markets exhibit high volatility as they digest overnight information [9]. Premarket trading data (4:30–9:29 AM) provides potentially predictive signals, but extracting useful information requires sophisticated processing.

Rather than fine-tuning the foundation model—computationally expensive and prone to overfitting with limited domain data—we investigate **correction-based adaptation**: keeping TimesFM completely frozen while training lightweight modules to correct its outputs. Critically, we employ a **hybrid neural-classical design** combining neural correction with Random Forest [3] residual learning.

We compare two neural correction architectures representing different design philosophies:

- **AttnCorrect**: Multi-head self-attention [22] over premarket sequences, enabling flexible temporal pattern learning (∼471K trainable parameters)
- **GatedLinear**: Low-rank bilinear projection [21] with learned gating for extreme parameter efficiency (∼49K trainable parameters)

Through comprehensive ablation across **10 major technology stocks** with over 2 million data points, we address the critical question:

What components matter most when adapting frozen foundation models, and how should neural and classical methods be integrated?

Our key contributions and findings:

1) A **hybrid neural-classical framework** achieving 6.4× mean per-day correlation improvement and 0.597 pooled correlation over frozen TimesFM
2) Comprehensive **ablation across 12 model variants** revealing component contributions
3) Evidence that **classical residual learning matches or exceeding neural correction**

4) The surprising finding that **simpler neural architectures outperform complex ones** when classical components are removed
5) **Per-stock analysis** across 10 major technology stocks with detailed performance breakdowns
6) Practical guidance: GatedLinear+RF achieves best performance with 9× fewer neural parameters

Open Science Statement: Our code and results are publicly available at https://github.com/Kasun-Dewage/Hybrid_Neural2026.git to facilitate reproducibility and further research in this area upon publication of this manuscript. All experiments use a fixed random seed of 42 for reproducibility.

II. Related Work

A. Foundation Models for Time Series

TimesFM [5] is a decoder-only transformer with 200M parameters, pretrained on over 100 billion time points from Google Trends, Wikipedia pageviews, and synthetic data. The model uses input patching with patch length 32 and achieves strong zero-shot performance on standard benchmarks including ETT, Weather, and Electricity datasets.

Chronos [1] takes a different approach, tokenizing time series values into discrete bins and training T5-style encoder-decoder models [19]. This enables the use of language modeling techniques for time series. Lag-Llama [20] adapts the LLaMA architecture with lag-based tokenization for probabilistic forecasting, demonstrating that decoder-only architectures can effectively model temporal dependencies.

While these models generalize impressively to standard benchmarks, domain adaptation to specialized applications like high-frequency finance remains challenging due to the unique statistical properties of financial time series.

B. Efficient Model Adaptation

Parameter-efficient fine-tuning has been extensively studied for large language models. LoRA [13] introduces low-rank decomposition of weight updates, training only $\mathbf{W} + \mathbf{BA}$ where $\mathbf{B} \in \mathbb{R}^{d\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times k}$ with $r \ll \min(d, k)$. Adapter layers [12] insert small bottleneck modules between transformer layers. Prefix tuning [15] prepends learnable tokens to the input sequence.

These methods modify model internals, requiring access to architecture details and intermediate representations. Our approach operates *externally* through output correction, treating the foundation model as a black box. This enables adaptation without architectural knowledge and avoids potential instabilities from modifying pretrained weights.

C. Hybrid Neural-Classical Methods

The integration of neural networks with classical machine learning methods has shown success across domains. In forecasting, hybrid approaches combining neural networks with statistical methods often outperform pure neural approaches [18]. Recent work demonstrates that gradient boosting and Random Forests remain highly competitive with deep learning for tabular data [8], particularly when feature engineering captures domain knowledge.

Random Forests [3] offer several advantages for residual learning: robustness to outliers, natural handling of feature interactions, and strong performance with limited training data. Our work provides empirical evidence for the value of neural-classical integration in foundation model adaptation.

D. Attention Mechanisms and Bilinear Models

The Transformer architecture [22] introduced scaled dot-product attention, enabling flexible modeling of long-range dependencies. For time series, attention has been adapted in various ways: the Temporal Fusion Transformer [16] combines LSTM encoders with multi-head attention for interpretable forecasting, while Informer [24] introduces ProbSparse attention for computational efficiency.

Bilinear models capture multiplicative interactions between features [21]. Low-rank bilinear pooling [14] reduces computation via matrix factorization. We apply bilinear projection to compress high-dimensional premarket sequences with minimal parameters.

E. Financial Time Series Prediction

Despite efficient market hypothesis arguments [6], empirical evidence supports short-term predictability during information asymmetry periods. Deep learning approaches using LSTM [7] and attention mechanisms have shown promise for financial forecasting. The opening trading hour exhibits particularly high volatility and potential predictability as markets process overnight information [9].

III. Methodology

A. Problem Setting

Let $\mathbf{X}^{(pm)} \in \mathbb{R}^{T\times F}$ denote the premarket sequence with $T = 300$ one-minute bars and $F = 7$ features:

- Normalized OHLC prices (4 features)
- Log-transformed volume (1 feature)
- Bar-to-bar momentum (1 feature)
- Intrabar volatility (1 feature)

Given frozen foundation model prediction $\hat{\mathbf{y}}^{(tfm)} \in \mathbb{R}^H$ for horizon $H = 60$ minutes and multiscale summary features $\mathbf{m} \in \mathbb{R}^{21}$, we learn a hybrid correction:

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}^{(tfm)} + \underbrace{\Delta\mathbf{y}_{neural}(\mathbf{X}^{(pm)}, \mathbf{m}, \hat{\mathbf{y}}^{(tfm)})}_{\text{Neural correction}} + \underbrace{\Delta\mathbf{y}_{RF}(f_{summary})}_{\text{Classical residual}} \tag{1}$$

The foundation model remains completely frozen throughout; only the correction modules are trained.

B. Hybrid Correction Framework

Figure 1 illustrates the complete hybrid neural-classical correction framework shared by both architectures.

Both architectures share these components:

- **TimesFM backbone**: Frozen 200M-parameter foundation model

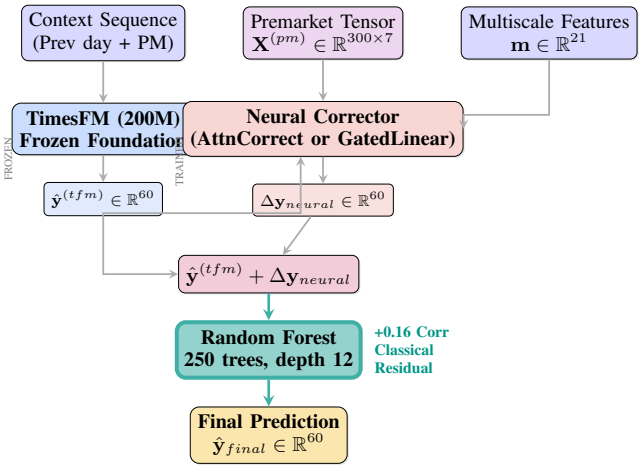

Fig. 1: **Hybrid Neural-Classical Correction Framework.** Both architectures share this pipeline: frozen TimesFM provides base predictions, a trainable neural corrector generates $\Delta\mathbf{y}_{neural}$, and Random Forest learns residual patterns. Our ablation reveals the classical component (RF, highlighted) provides nearly equally or exceeding to the neural component.

- **Multiscale features** $\mathbf{m}$: Returns, volatility, and volume computed over 5/15/30/60-minute windows plus overnight gap (21 dimensions total)
- **Random Forest residual**: 250 trees, max depth 12, min samples leaf 3, trained on neural correction residuals
- **Directional loss**: $\mathcal{L} = \mathcal{L}_{MSE} + 0.3\mathcal{L}_{dir} + 0.2\mathcal{L}_{cum}$ to encourage correct direction prediction

C. AttnCorrect: Attention-Based Correction

Figure 2 presents the complete AttnCorrect architecture with ∼471K trainable parameters.

1) Premarket Encoding: The sequence is projected to hidden dimension $d = 128$:

$$\mathbf{H}^{(0)} = \text{Linear}_{128}(\text{Drop}_{0.3}(\text{GELU}(\text{LN}(\mathbf{X}^{(pm)}\mathbf{W}_e + \mathbf{b}_e)))) \tag{2}$$

where $\mathbf{W}_e \in \mathbb{R}^{7\times128}$. GELU [10] provides smooth nonlinearity and LayerNorm [2] stabilizes training.

2) Self-Attention Layers: We apply $L = 2$ transformer layers with $h = 4$ heads and head dimension $d_k = 32$:

$$\mathbf{H}' = \text{LN}(\mathbf{H}^{(\ell-1)} + \text{MHA}(\mathbf{H}^{(\ell-1)})) \tag{3}$$

$$\mathbf{H}^{(\ell)} = \text{LN}(\mathbf{H}' + \text{FFN}(\mathbf{H}')) \tag{4}$$

The feed-forward network uses expansion factor 2:

$$\text{FFN}(\mathbf{x}) = \text{Drop}_{0.3}(\text{GELU}(\mathbf{x}\mathbf{W}_1))\mathbf{W}_2 \tag{5}$$

with $\mathbf{W}_1 \in \mathbb{R}^{128\times256}$ and $\mathbf{W}_2 \in \mathbb{R}^{256\times128}$.

3) Cross-Modal Fusion: The pooled context $\mathbf{c}_{pm} = \frac{1}{T}\sum_t \mathbf{H}_t^{(L)}$ is concatenated with separately encoded TimesFM predictions and multiscale features:

$$\mathbf{h} = \text{MLP}_{fusion}([\mathbf{c}_{pm}; \mathbf{c}_{tfm}; \mathbf{c}_{ms}]) \tag{6}$$

where the fusion MLP maps $\mathbb{R}^{384} \to \mathbb{R}^{128}$.

D. GatedLinear: Bilinear-Gated Correction

Figure 3 presents the GatedLinear architecture with only ∼49K trainable parameters—9× fewer than AttnCorrect.

1) Low-Rank Bilinear Projection: Instead of attention over the full sequence, we compress the premarket tensor through learned projection matrices:

$$\mathbf{Z} = \mathbf{V}^\top \mathbf{X}^{(pm)} \mathbf{U} \in \mathbb{R}^{8\times4} \tag{7}$$

where $\mathbf{V} \in \mathbb{R}^{300\times8}$ projects the temporal dimension and $\mathbf{U} \in \mathbb{R}^{7\times4}$ projects features. This bilinear form [21] captures interactions between temporal positions and feature channels with only $300 \times 8 + 7 \times 4 = 2,428$ parameters.

The temporal projection $\mathbf{V}$ is initialized with a slight linear trend to encourage learning of temporal patterns. The result $\mathbf{Z}$ is flattened to $\mathbf{z} \in \mathbb{R}^{32}$.

2) Gated Correction: Features are concatenated: $[\mathbf{z}; \mathbf{m}; \hat{\mathbf{y}}^{(tfm)}] \in \mathbb{R}^{113}$ and encoded through a two-layer MLP. The gating mechanism, inspired by LSTM [11] and GRU [4] gates, computes:

$$\mathbf{g} = \sigma(\mathbf{W}_g\mathbf{h} + \mathbf{b}_g) \in [0,1]^{60} \tag{8}$$

$$\mathbf{r} = \mathbf{W}_r\mathbf{h} + \mathbf{b}_r \in \mathbb{R}^{60} \tag{9}$$

$$\Delta\mathbf{y}_{neural} = \mathbf{g} \odot \mathbf{r} \tag{10}$$

The gate $\mathbf{g}$ controls correction magnitude per time step, allowing the model to selectively correct when confident.

E. Training Details

All experiments use a fixed random seed of 42 across PyTorch, NumPy, and Python's random module to ensure full reproducibility. Both models are trained with AdamW optimizer [17]:

- Learning rate: 5×10^{-4} (AttnCorrect), 8×10^{-4} (GatedLinear)
- Weight decay: 10^{-4}
- Early stopping patience: 20 (AttnCorrect), 25 (GatedLinear)
- Gradient clipping: max norm 1.0
- Dropout: 0.3 (AttnCorrect), 0.25 (GatedLinear)
- Random seed: 42 (fixed for all random number generators)

F. Ablation Study Design

Table I summarizes our comprehensive ablation with 12 model variants across 5 baselines, 2 full models, and 5 ablation variants.

IV. Experimental Setup

A. Dataset

We evaluate on 1-minute bar data for 10 major technology stocks representing diverse market capitalizations and volatility profiles. Table II provides dataset statistics. The timeframe spans December 2024 to January 2026.

The dataset spans approximately 266 trading days per stock (except NFLX with 149 days due to data availability). Stock-specific volatility varies significantly, from 0.001000 (MSFT, most stable) to 0.001988 (TSLA, most volatile).

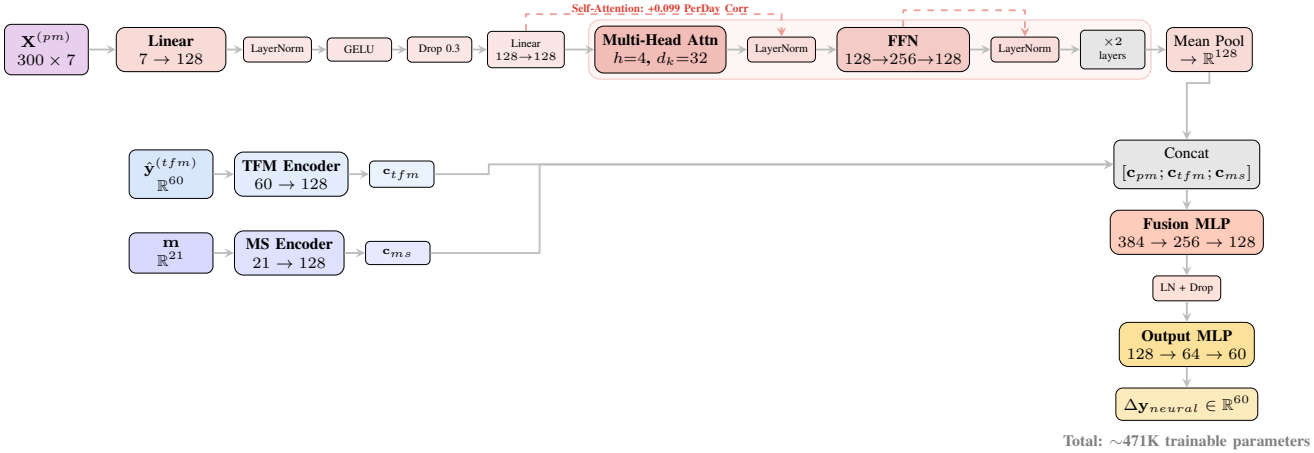

Fig. 2: **AttnCorrect Architecture.** Premarket sequences (300 × 7) pass through embedding layers, then L=2 transformer blocks with 4-head self-attention (highlighted box). Pooled features fuse with encoded TimesFM predictions and multiscale features. Output weights initialized to zero [23] for stable training.

TABLE I: Ablation Study Design: 12 Model Variants

ID	Model	Ablated Component	Research Question
Baselines (5 models)			
01	HistMean	–	Naive baseline
02	MLP	–	Standard neural baseline
03	LSTM	–	Recurrent sequence modeling
04	BiLSTM	–	Bidirectional modeling
05	TimesFM Base	–	Frozen foundation model
Full Hybrid Models (2 models)			
06	AttnCorrect+RF	None (Full)	Attention + classical
07	GatedLinear+RF	None (Full)	Bilinear + classical
Ablation Variants (5 models)			
08	AttnCorrect-NoRF	Random Forest	How much does RF add?
09	GatedLinear-NoRF	Random Forest	How much does RF add?
10	GatedLinear-NoGate	Gating mechanism	Is gating necessary?
11	GatedLinear-NoBilinear	Bilinear projection	Is bilinear helpful?
12	AttnCorrect-NoAttn	Self-attention	Is attention necessary?

TABLE II: Dataset Statistics: 10 Technology Stocks

Stock	Train Days	Val Days	Test Days	Training Volatility
NVDA	186	40	40	0.001627
MSFT	186	40	40	0.001000
AAPL	186	40	40	0.001098
GOOG	186	40	40	0.001050
GOOGL	186	40	40	0.001070
AMZN	186	40	40	0.001190
META	186	40	40	0.001298
AVGO	186	40	40	0.001764
TSLA	186	40	40	0.001988
NFLX	104	22	23	0.001150
Total	**2,011,399 rows across 10 tickers**			

B. Data Leakage Prevention

To ensure temporal validity and avoid look-ahead bias:

1) Strict input/label time separation: For each trading date, all model inputs use only pre-9:30 AM data (previous day's regular session + current day's premarket). Targets are computed exclusively from 9:30–10:30 AM.

2) Chronological splits: Data is split by trading day in strict chronological order—no shuffling. Test days occur strictly after all training and validation days.

3) Training-only normalization: Stock-specific volatility for feature normalization is computed using **training dates only**.

4) Model fitting restricted to training set: TimesFM remains frozen. Neural correctors and Random Forest are fitted only on training data; validation is used only for early stopping.

C. Evaluation Metrics

We report three complementary correlation metrics to provide a complete picture of predictive quality, alongside error metrics:

- **MAE (%)**: Mean Absolute Error of return predictions
- **RMSE (%)**: Root Mean Squared Error
- **Mean Per-Day Correlation** ($\overline{\rho}_{day}$): For each test day d, we compute the Pearson correlation between the 60-bar predicted return vector $\hat{\mathbf{y}}_d$ and the 60-bar actual return vector $\mathbf{y}_d$, then average across all D test days: $\overline{\rho}_{day} = \frac{1}{D}\sum_{d=1}^{D} \rho(\hat{\mathbf{y}}_d, \mathbf{y}_d)$. This measures *within-day temporal alignment*—whether the model correctly predicts when returns are larger or smaller within each trading session.
- **Cross-Day Correlation** (ρ_{cross}): Pearson correlation between the cumulative predicted return per day and the cumulative actual return per day, computed across all D test days. This measures whether the model correctly predicts which days have positive vs. negative net returns—i.e., *directional forecasting across days*.
- **Pooled Correlation** (ρ_{pool}): Pearson correlation computed by concatenating all predictions and all actuals across all days and bars into single vectors.

V. Results

A. Aggregate Results

Table III presents aggregate results across all 10 stocks, sorted by RMSE. All three correlation metrics are reported.

Key Observations:

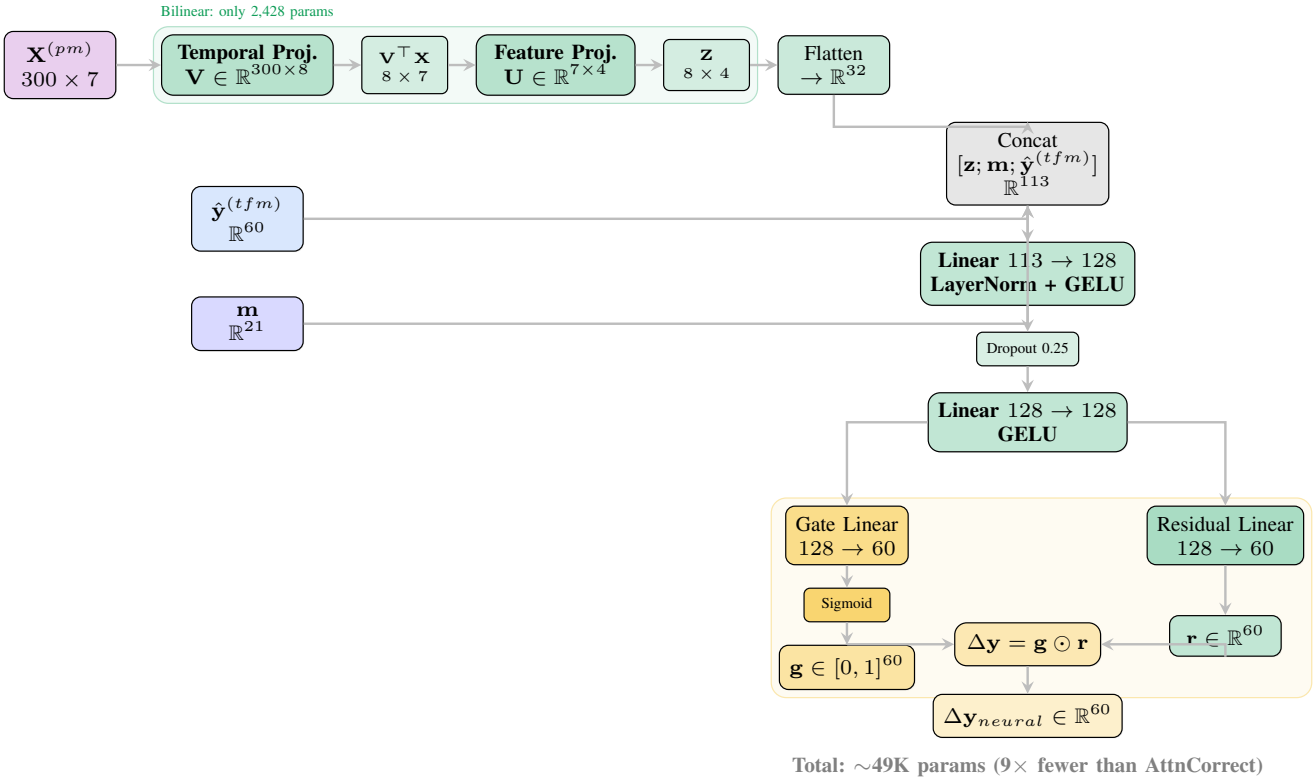

Fig. 3: **GatedLinear Architecture.** The premarket tensor is compressed via low-rank bilinear projection $\mathbf{Z} = \mathbf{V}^\top\mathbf{X}\mathbf{U}$ (only 2,428 parameters). A gating mechanism $\mathbf{g} \in [0,1]^{60}$ modulates the correction magnitude per time step. Despite 9× fewer parameters than AttnCorrect, this architecture achieves the best overall performance when combined with Random Forest.

TABLE III: Aggregate Results Across 10 Stocks (Sorted by RMSE). Three correlation metrics: mean per-day ($\overline{\rho}_{day}$), cross-day (ρ_{cross}), and pooled (ρ_{pool}).

Model	MAE(%)↓	RMSE(%)↓	$\overline{\rho}_{day}$↑	ρ_{cross}↑	ρ_{pool}↑
07_GatedLinear+RF	**0.1079**	**0.1535**	**0.3730**	0.5631	**0.5972**
06_AttnCorrect+RF	0.1081	0.1547	0.3678	**0.5819**	0.5890
11_GatedLinear-NoBilinear	0.1071	0.1600	0.3422	0.5055	0.5040
10_GatedLinear-NoGate	0.1084	0.1666	0.2864	0.3620	0.4317
02_MLP	0.1105	0.1679	0.2407	0.4957	0.4584
09_GatedLinear-NoRF	0.1087	0.1710	0.2147	0.3058	0.3219
08_AttnCorrect-NoRF	0.1091	0.1740	0.2335	0.3829	0.3698
03_LSTM	0.1082	0.1744	0.3519	0.4628	0.4943
04_BiLSTM	0.1090	0.1779	0.3420	0.4653	0.4737
12_AttnCorrect-NoAttn	0.1109	0.1922	0.1347	0.4179	0.1997
05_TimesFM Base	0.1113	0.1946	0.0586	−0.0357	0.0614
01_HistMean	0.1115	0.1946	0.0442	0.0000	0.0610

1) **Hybrid methods dramatically outperform all baselines**: Mean per-day correlation improves from 0.059 (TimesFM) to 0.373 (GatedLinear+RF)—a **6.4× improvement**. Pooled correlation reaches 0.597.
2) **GatedLinear+RF achieves best overall performance**: Best RMSE (0.1535%), best MAE (0.1079%), and best mean per-day (0.373) and pooled (0.597) correlation with 9× fewer parameters
3) **Cross-day vs. per-day correlations reveal different model strengths**: AttnCorrect+RF achieves the highest cross-day correlation (0.582), indicating slightly stronger daily directional forecasting, while GatedLinear+RF leads on per-day and pooled metrics.

B. Per-Stock Performance Analysis

Table IV provides detailed per-stock results comparing the two full hybrid models against frozen TimesFM. We report RMSE, mean per-day correlation, and pooled correlation.

Per-Stock Insights:

- **GatedLinear+RF wins on 7/10 stocks for RMSE**, including the most volatile stocks (AVGO, TSLA)
- **Both methods transform near-zero per-day correlations into moderate-to-strong ones**
- **Highest improvements on volatile stocks**: NVDA and AVGO show largest correlation gains
- **Per-day vs. pooled correlations differ**: For AAPL, per-day correlation is only 0.171 (GatedLinear+RF) while pooled is 0.347, suggesting the model captures cross-day variance structure better than within-day temporal patterns for less volatile stocks.

C. Ablation Analysis: Component Contributions

1) Finding 1: Classical Residual Learning Provides the Largest Single-Component Contribution: **Random Forest provides the largest improvement of any individual component, whether neural or classical**:

TABLE IV: Per-Stock Performance: TimesFM vs. Hybrid Corrections. Mean per-day correlation ($\overline{\rho}_{day}$) and pooled correlation (ρ_{pool}) shown separately.

	RMSE (%)			Mean Per-Day Corr ($\overline{\rho}_{day}$)			Pooled Corr (ρ_{pool})		
Stock	**TFM**	**Attn+RF**	**Gated+RF**	**TFM**	**Attn+RF**	**Gated+RF**	**TFM**	**Attn+RF**	**Gated+RF**
NVDA	0.2466	0.1689	**0.1683**	0.018	**0.485**	0.473	0.165	0.734	**0.739**
MSFT	0.1228	**0.0971**	0.1010	0.099	**0.417**	0.424	0.075	**0.633**	0.624
AAPL	0.1095	0.1073	**0.1068**	0.033	0.135	**0.171**	0.040	0.311	**0.347**
GOOG	0.2140	0.1659	**0.1651**	0.054	**0.396**	0.383	−0.014	0.644	**0.646**
GOOGL	0.2151	0.1690	**0.1616**	0.120	0.376	**0.408**	0.070	0.638	**0.669**
AMZN	0.1463	0.1272	**0.1258**	0.087	0.344	**0.351**	0.085	0.505	**0.527**
META	0.1636	0.1370	**0.1363**	0.083	0.398	**0.399**	0.158	**0.665**	0.659
AVGO	0.3183	0.2372	**0.2320**	0.010	0.392	**0.395**	−0.075	**0.688**	0.685
TSLA	0.2541	**0.1948**	0.1954	0.109	**0.461**	0.458	0.136	**0.647**	0.644
NFLX	0.1558	**0.1428**	0.1431	−0.027	**0.274**	0.268	−0.025	0.426	**0.434**
Average	0.1946	0.1547	**0.1535**	0.059	0.368	**0.373**	0.061	0.589	**0.597**
Best on	0/10	3/10	**7/10**	0/10	4/10	**6/10**	0/10	4/10	**6/10**

TABLE V: Detailed Ablation Analysis: With vs. Without Each Component. We report mean per-day ($\overline{\rho}_{day}$) and cross-day (ρ_{cross}) correlation contributions.

Component	RMSE_{with}	$\text{RMSE}_{w/o}$	$\overline{\rho}_{day,with}$	$\overline{\rho}_{day,w/o}$	$\Delta\overline{\rho}_{day}$	$\Delta\rho_{cross}$
RF (GatedLinear)	0.1535	0.1710	0.3730	0.2147	**+0.1583**	**+0.2573**
RF (AttnCorrect)	0.1547	0.1740	0.3678	0.2335	**+0.1343**	**+0.1990**
Self-Attention	0.1740	0.1922	0.2335	0.1347	**+0.0988**	−0.0350
Gating Mechanism	0.1710	0.1666	0.2147	0.2864	−0.0717	−0.0562
Bilinear Projection	0.1710	0.1600	0.2147	0.3422	**−0.1276**	−0.1997

$\Delta\overline{\rho}_{day}$: positive means component improves mean per-day correlation; negative means removing component *improves* performance. $\Delta\rho_{cross}$: same for cross-day correlation.

TABLE VI: Parameter Count Breakdown

Component	AttnCorrect	GatedLinear
Premarket processing	∼66,000	2,428
Attention/Bilinear	∼200,000	–
Encoder/Fusion MLP	∼190,000	∼31,000
Output layers	∼15,000	∼15,000
Total trainable	**∼471,000**	**∼49,000**
Ratio	9.6×	1×
Performance	0.368/ 0.589 Corr	**0.373/ 0.597 Corr**

- **GatedLinear**: RF adds +0.158 mean per-day correlation (0.215 → 0.373) and +0.257 cross-day correlation, reduces RMSE by 10.2%
- **AttnCorrect**: RF adds +0.134 mean per-day correlation (0.234 → 0.368) and +0.199 cross-day correlation, reduces RMSE by 11.1%

This finding demonstrates that classical machine learning methods remain highly valuable even when combined with neural approaches for foundation model adaptation. RF's contribution is particularly large for cross-day correlation, indicating it is effective at capturing features that predict daily directional returns.

2) Finding 2: Simpler Architectures Outperform Without Classical Components: **The most surprising result**: When Random Forest is removed, simpler neural architectures outperform more complex ones.

Comparing GatedLinear-NoRF (0.215 per-day corr) vs. GatedLinear-NoBilinear (0.342 per-day corr):

- Removing bilinear projection **improves** mean per-day correlation by +0.128
- Removing bilinear projection **improves** RMSE by 6.4%

The bilinear projection compresses $300 \times 7 = 2,100$ dimensions to just 32—a 65× compression that loses fine-grained temporal information. The NoBilinear variant uses the last 60 premarket bars flattened directly (420 dimensions), preserving recent dynamics better.

3) Finding 3: Self-Attention Provides the Largest Positive Neural Contribution: Self-attention contributes +0.099 mean per-day correlation—the largest positive contribution from any neural-only component:

- **With attention**: RMSE=0.1740%, $\overline{\rho}_{day}$=0.2335
- **Without (mean pooling)**: RMSE=0.1922%, $\overline{\rho}_{day}$=0.1347

Self-attention enables flexible temporal pattern learning across the 300-bar premarket sequence, allowing the model to attend to relevant time periods dynamically. The cross-day correlation contribution is minimal (−0.035), confirming that self-attention's main value is in within-day temporal modeling rather than daily directional prediction.

4) Finding 4: Gating Mechanism Provides Marginal Benefit: Removing the gating mechanism slightly **improves** performance:

- $\Delta\overline{\rho}_{day}$ = −0.072 (removing gate improves by 0.072)
- The learned gates may introduce unnecessary complexity without adding predictive value

D. Parameter Efficiency Analysis

Table VI details the parameter breakdown for both architectures.

GatedLinear achieves **better performance with 9.6× fewer parameters**, demonstrating that parameter efficiency does not require sacrificing accuracy in this domain.

VI. Discussion

A. On Correlation Metrics for Multi-Step Forecasting

A critical methodological point: for multi-step time series predictions (60 bars $\times$ D days), how correlation is computed significantly affects reported results. Pooled correlation (flattening all predictions into one vector) can be inflated by cross-day variance structure—if the model merely captures that some days are more volatile than others, pooled correlation will be nonzero even without genuine within-day predictive ability. Mean per-day correlation is more conservative and more relevant for intraday trading, as it measures whether the model correctly predicts the *temporal pattern* of returns within each session. Cross-day correlation captures a different skill: predicting which days will have positive vs. negative cumulative returns. We recommend reporting all three metrics in future work on multi-step financial forecasting.

B. Why Does Classical Residual Learning Complement Neural Correction?

Several factors explain Random Forest's strong performance:

1) **Complementary feature spaces**: RF operates on handcrafted multiscale statistics (21 dimensions) capturing domain knowledge, while neural networks process raw sequences
2) **Non-linear feature interactions**: Decision trees naturally capture complex interactions between features without explicit specification
3) **Robustness to outliers**: Tree-based methods handle the heavy-tailed distributions common in financial data
4) **Residual learning**: RF learns patterns that neural networks systematically miss, providing orthogonal improvements

C. Why Do Simpler Architectures Excel Without RF?

The bilinear projection's 65× compression is too aggressive:

- Loses fine-grained temporal dynamics important for prediction
- Forces the model to learn optimal temporal patterns that may not generalize
- Recent premarket information (last 60 bars) is more predictive than temporally-aggregated patterns

When RF is present, it compensates by accessing premarket summary statistics directly, explaining why the full GatedLinear+RF system achieves best performance despite the bilinear bottleneck.

D. Practical Recommendations

Based on our comprehensive ablation:

1) **Always include classical residual learning**: RF provides contributions nearly matching at minimal computational cost
2) **Start with simpler neural architectures**: Complex neural components may not provide benefits proportional to their parameter cost
3) **Self-attention is valuable for sequence modeling**: Provides largest positive neural contribution (+0.099 per-day correlation)
4) **Avoid aggressive dimensionality reduction**: Preserve recent temporal information unless classical components can compensate
5) **Prioritize parameter efficiency**: GatedLinear+RF achieves best results with 9× fewer parameters
6) **Report multiple correlation metrics**:

E. Limitations and Future Work

Limitations:

- Evaluation limited to 10 large-cap technology stocks
- Test period spans 40 days; longer evaluation needed
- Transaction costs and market impact not modeled
- Only TimesFM tested; other foundation models may behave differently
- The limited baseline correlation of frozen TimesFM means that even modest absolute improvements yield large relative gains
- Premarket signals may be inherently more predictive of early-session returns than of later trading hours, so the reported correlations may not generalize to full-day forecasting
- This work evaluates the hybrid correction methodology rather than claiming that TimesFM itself is suited for financial prediction tasks

Future directions:

- Extend to other asset classes and market conditions
- Investigate adaptive weighting of neural vs. classical components
- Apply to other foundation models (Chronos, Lag-Llama)
- Develop theoretically-grounded guidelines for neural-classical integration

VII. Conclusion

We presented a comprehensive study of hybrid neural-classical correction for adapting frozen time series foundation models to high-frequency stock prediction. Through systematic ablation across 10 technology stocks and 12 model variants, we reveal that:

1) **Hybrid approaches achieve dramatic improvements**: 6.4× mean per-day correlation improvement over frozen TimesFM (0.059 $\rightarrow$ 0.373), with pooled correlation reaching 0.597
2) **Classical and neural components contribute nearly equally**
3) **Simpler neural architectures outperform complex ones** when classical components are removed
4) **Parameter efficiency is achievable**: GatedLinear+RF achieves best performance with 9× fewer neural parameters

Our key message: **effective foundation model adaptation requires thoughtful integration of neural and classical methods**. Classical machine learning remains highly valuable even in the era of foundation models, providing complementary capabilities that neural networks alone cannot match.

ACKNOWLEDGMENT

This paper was prepared with the assistance of AI tools. Specifically, AI tools were used for text editing, citation formatting and optimization assistance, and GitHub Copilot and other tools were used for coding assistance. The authors take full responsibility for the content and have verified all AI-assisted contributions.

The authors thank the anonymous reviewers for their valuable feedback.

REFERENCES

[1] A. F. Ansari *et al.*, "Chronos: Learning the language of time series," *arXiv:2403.07815*, 2024.
[2] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv:1607.06450*, 2016.
[3] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[4] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. EMNLP*, 2014.
[5] A. Das *et al.*, "A decoder-only foundation model for time-series forecasting," in *Proc. ICML*, 2024.
[6] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *J. Finance*, vol. 25, no. 2, pp. 383–417, 1970.
[7] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European J. Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
[8] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" in *Proc. NeurIPS*, 2022.
[9] J. Hasbrouck and G. Saar, "Low-latency trading," *J. Financial Markets*, vol. 16, no. 4, pp. 646–679, 2013.
[10] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv:1606.08415*, 2016.
[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[12] N. Houlsby *et al.*, "Parameter-efficient transfer learning for NLP," in *Proc. ICML*, 2019.
[13] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2022.
[14] J.-H. Kim *et al.*, "Hadamard product for low-rank bilinear pooling," in *Proc. ICLR*, 2017.
[15] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. ACL-IJCNLP*, 2021.
[16] B. Lim *et al.*, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.
[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.
[18] S. Makridakis *et al.*, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PloS One*, vol. 13, no. 3, 2018.
[19] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *JMLR*, vol. 21, no. 140, pp. 1–67, 2020.
[20] K. Rasul *et al.*, "Lag-Llama: Towards foundation models for probabilistic time series forecasting," *arXiv:2310.08278*, 2023.
[21] J. B. Tenenbaum and W. T. Freeman, "Separating style and content with bilinear models," *Neural Computation*, vol. 12, no. 6, pp. 1247–1283, 2000.
[22] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017.
[23] H. Zhang, Y. N. Dauphin, and T. Ma, "Fixup initialization: Residual learning without normalization," in *Proc. ICLR*, 2019.
[24] H. Zhou *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI*, 2021.