Volatility Prediction of Shanghai Gold Futures Based on Transfer Learning

Jingyun Zhao*
School of Mathematics and Statistics, North China University of Water Resources and Electric Power, Zhengzhou, Henan, China
**Corresponding Author*

Abstract: Gold futures volatility forecasting is of great significance for asset allocation, risk management, and the interpretation of macroeconomic signals. However, its nonlinearity, long memory, and cross- market heterogeneity pose challenges to both traditional econometric models and deep learning methods. This paper proposes a transfer learning-based volatility forecasting model for futures, named CrossFormer-GRU, which transfers knowledge from the Brent crude oil market to the Shanghai gold futures market. By incorporating an uncertainty-weighted adversarial domain adaptation mechanism, the model dynamically optimizes multi-task losses. Experimental results show that CrossFormer-GRU significantly outperforms benchmark models such as GARCH, SVR, LSTM, and Transformer in terms of MAE, RMSE, MAPE, and R^2. Ablation studies further validate the critical roles of transfer learning and uncertainty weighting in enhancing model performance.

Keywords: Gold Futures; Transfer Learning; Uncertainty Weighting; CrossFormer-GRU

1. Introduction

As a financial instrument featuring leverage trading, hedging, and short-selling mechanisms, gold futures have become a core vehicle for investors in asset allocation and risk management. Its price fluctuations are increasingly correlated with macroeconomic policies, geopolitical events, and market sentiment [1,2]. The Shanghai gold futures, a Renminbi-denominated gold futures contract launched by the Shanghai Futures Exchange, are more closely aligned with the domestic market. They provide domestic gold enterprises and investors with hedging tools, effectively reducing exchange rate risks [3].

Gold volatility forecasting is crucial for optimizing asset allocation and risk management, offering investors a basis for trading decisions and hedging, and assisting financial institutions in risk pricing and capital management. It also serves as an important window for observing macroeconomic policy transmission and market risk aversion. Its complexity continuously drives the evolution of time series analysis methods. However, financial volatility, as a measure of asset return uncertainty, inherently exhibits typical characteristics such as clustering, asymmetry, and long memory [4], which form the foundation and difficulty of volatility forecasting modeling.

In the early stages of volatility forecasting model evolution, Autoregressive Conditional Heteroskedasticity (ARCH) family models and their extensions dominated. Their core idea is to let conditional variance itself become a time-varying sequence. For instance, Bentes [5] employed three volatility models from the GARCH family to examine the volatility behavior of gold returns. Results indicated that FIGARCH(1,d,1) was the best model for capturing the linear dependence in the conditional variance of gold returns as indicated by information criteria. Yu [6] used the GARCH-MIDAS model, incorporating cryptocurrency policy and price uncertainty, along with several other commonly used uncertainty measures, to compare their in-sample impact and out-of-sample predictive power on the volatility of COMEX gold and silver futures markets. Ampountolas [4] found that the standard GARCH model performed better for gold futures forecasting; whereas for the S&P 500 index, the EGARCH model, which captures asymmetry, was superior. However, these models are inherently linear or parametric frameworks, struggling to fully capture the complex nonlinear dynamics in financial time series and the intricate patterns from

multi-source data [7].

To overcome the limitations of traditional parametric models, data-driven machine learning methods have gradually become the mainstream paradigm in volatility forecasting research. The core advantage of these methods lies in not presupposing a strict data generation process; instead, they learn complex nonlinear patterns directly from historical data through algorithms, offering significantly enhanced flexibility [8]. Traditional machine learning, such as Support Vector Regression (SVR), has been shown to potentially outperform traditional GARCH models in short-term volatility forecasting, although its feature engineering relies on manual expertise.

With the advancement of computational power, deep learning models, leveraging their powerful automatic feature extraction capabilities, have fundamentally transformed the approach to prediction model construction. Through multi-layer neural network structures, deep learning can automatically learn high-level abstract representations from raw or minimally processed data, thus more effectively characterizing the complex dynamic patterns within volatility series. As a natural choice for sequence modeling, RNNs and their improved variant, LSTM, have become benchmark models for deep learning-based volatility forecasting. LSTM alleviates the long-term dependency problem through gating mechanisms and is widely applied to capture the clustering and persistence features of volatility. Empirical studies show it outperforms traditional machine learning methods in various asset volatility forecasting tasks [9]. In recent years, the Transformer architecture, based entirely on the self-attention mechanism, has revolutionized sequence prediction. Its core multi-head self-attention mechanism allows the model to process information from all positions in the sequence in parallel, showing significant potential in capturing long-term dependencies and multi-scale periodic features of volatility [10]. In finance, Transformers and their variants (e.g., Informer, Autoformer) have been successfully applied to high-frequency volatility forecasting, demonstrating advantages in handling long sequences and identifying complex cross-period correlations [11]. Despite the powerful predictive capabilities of deep learning models, their practical application in futures markets still faces challenges such as cold start, market non-stationarity, and complex correlations.

Transfer learning is an important branch of machine learning dedicated to applying acquired knowledge to new domains, thereby improving the efficiency and capability of solving new problems. This precisely addresses the data scarcity and heterogeneity challenges in the Shanghai gold futures market. In time series forecasting tasks, transfer learning has demonstrated good generalization performance [12]. Ye [13] proposed the RATL algorithm for exchange rate forecasting, divided into two stages: representation relation alignment and regression relation alignment. Nguyen [14] introduced the DTRSI framework, optimizing stock price prediction performance by pre-training and fine-tuning LSTMs. He [15] proposed a transfer learning training strategy based on two source datasets, using Dynamic Time Warping (DTW) to measure time series similarity for source domain selection. The Meta-LSTR framework proposed by Chen [8] during the training phase, performs meta-training on various futures varieties to extract "meta-knowledge" across varieties. When applied to a new variety with scarce data, the model can be quickly fine-tuned using this meta-knowledge, significantly improving small-sample prediction accuracy. However, traditional transfer learning faces issues like negative transfer caused by inconsistent distributions between source and target domains, and the weight allocation in multi-task optimization often relies on manual parameter tuning. Kendall [16] pioneered the introduction of homoscedastic uncertainty into multi-task deep learning. By maximizing the Gaussian likelihood estimation of each task's homoscedastic uncertainty, they derived learnable uncertainty parameters, enabling the model to automatically learn the relative weights of different tasks. Wang [17] further combined uncertainty estimation with Wasserstein gradient flow, quantifying the transferability of source and target domain samples through domain prediction uncertainty, and down-weighting domain- specific samples, effectively mitigating negative transfer in adversarial domain matching.

2. Model Construction

This paper proposes a transfer learning-based framework for futures price volatility prediction,

aiming to transfer knowledge from the Brent crude oil market to the gold futures market. First, input data undergoes feature extraction via the CrossFormer model. The extracted features are then passed to a GRU-based volatility predictor for volatility forecasting. Simultaneously, a domain classifier identifies the specific domain type of the data. Combined with uncertainty weighting technology, it dynamically adjusts the weights between the volatility prediction and domain classification tasks, thereby enhancing the model's volatility prediction capability. Finally, model optimization and parameter updates are achieved through the backpropagation algorithm. Model architecture diagram is shown in Figure 1.

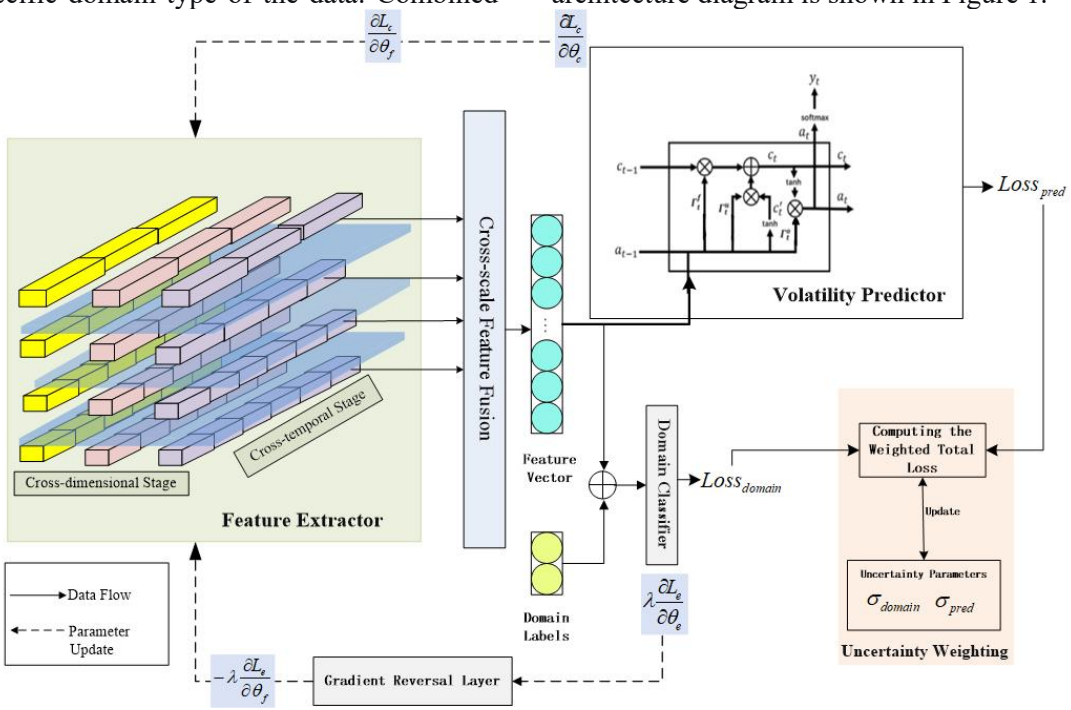

Figure 1. Architecture of the CrossFormer-GRU Model

2.1 CrossFormer-based Feature Extractor

The CrossFormer model is a long-sequence forecasting method based on the Transformer architecture, designed to effectively capture cross-dimensional dependencies in multivariate time series data. It has demonstrated excellent performance in many tasks such as traffic flow prediction, electricity load forecasting, and financial time series forecasting. In this study, CrossFormer is applied to extract common features from both the Brent crude oil market and the gold futures market. Its process mainly includes the following two parts:

2.1.1 Dimension-Specific weighting (DSW) embedding

The DSW embedding is a key feature of the CrossFormer model, aimed at capturing cross-dimensional dependencies in multivariate time series data. The time series data for each dimension is segmented into segments of length L, denoted as $x_{i,d}$,where T represents the past time steps, and D represents the number of dimensions. Each segment is embedded into a vector $h_{i,d}$. After embedding, the resulting two-dimensional vector matrix is H :

$$H = \left\{ h_{i,d} \middle| 1 \le i \le \frac{T}{l}, 1 \le d \le D \right\} \quad (1)$$

2.1.2 Two-Stage attention mechanism

The purpose of the TSA layer is to capture cross-time and cross-dimensional dependencies within the two-dimensional vector matrix. It processes information in two stages. In the cross-time stage, a multi-head self-attention (MSA) mechanism is applied within each dimension. The calculation formula is as follows:

$$Z_{:,d}^{time} = LayerNorm\left(Z_{:,d} + MSA^{time}\left(Z_{:,d}, Z_{:,d}, Z_{:,d} \right) \right) \quad (2)$$

$$Z^{time} = LayerNorm\left(\hat{Z}^{time} + MLP\left(\hat{Z}^{time} \right) \right) \quad (3)$$

Where $LayerNorm$ denotes layer normalization, MLP denotes a multi-layer feed-forward network, and $\hat{Z}$ and Z represent the outputs of the MSA and FFN sub-layers, respectively.

In the cross-dimension stage, to reduce

computational complexity, a router mechanism is employed to aggregate information from different dimensions. The calculation formulas are as follows:

$$B_i = MSA_1^{\dim}\left(R_{i,:}, Z_{i,:}^{time}, Z_{i,:}^{time}\right), 1 \le i \le L \quad (4)$$

$$\overline{Z}_{i,:}^{\dim} = MSA_2^{\dim}\left(R_{i,:}, B_{i,:}, B_{i,:}\right), 1 \le i \le L \quad (5)$$

$$\hat{Z}^{\dim} = LayerNorm\left(Z^{time} + \overline{Z}^{time}\right) \quad (6)$$

$$Z^{\dim} = LayerNorm\left(\hat{Z}^{\dim} + MLP\left(\hat{Z}^{dim}\right)\right) \quad (7)$$

Where B denotes information from all dimensions. The final output of the entire two-stage attention can be expressed as:

$$Y = Z^{\dim} = TSA(Z) \quad (8)$$

where Z represents the input vector array for the TSA layer, and Y represents the output vector array for the TSA layer.

2.2 GRU-based Volatility Predictor

The Gated Recurrent Unit (GRU) neural network is optimized based on the Long Short-Term Memory (LSTM) neural network. It aims to improve sequence data processing efficiency through structural simplification, optimizing information processing efficiency and memory capability. It achieves a more refined design by simplifying the LSTM's gating mechanism. Compared to classic Recurrent Neural Networks (RNNs), it effectively alleviates the vanishing gradient problem in deep network training by introducing gating units. This architectural improvement significantly enhances the model's ability to model long-term dependencies in sequence data.

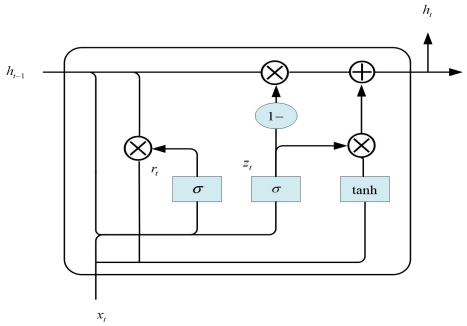

Figure 2. Hidden Layer Architecture of Gated Recurrent Neural Network (GRU)

The GRU neural network restructures the core mechanisms of the LSTM network through structural simplification. Compared to the triple control structure of LSTM (input gate, forget gate, and output gate), GRU innovatively compresses the gating system into a dual-gate architecture with reset and update gates. This design not only simplifies the model structure and significantly reduces computational complexity but also maintains the network's effective memory capacity when processing sequential data. Through this refined gating control strategy, GRU achieves the goal of optimizing resource utilization efficiency while maintaining performance. Simultaneously, GRU significantly enhances the model's training efficiency and performance by optimizing its hidden state update mechanism.

In Figure 2, z_t and r_t represent the update gate and reset gate, respectively. The update gate controls the extent to which the previous moment's state information influences the current state; a larger value indicates more retention of past information. The reset gate controls how much of the previous moment's state information is considered when forming the current candidate state; a smaller value means more of the past information is ignored. This gating mechanism enables the model to dynamically adjust the flow of information, thereby more effectively capturing dependencies within time series data. Its mathematical expressions are:

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \quad (9)$$

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right) \quad (10)$$

$$\tilde{h}_t = \tanh\left(W \cdot \left[r_t \odot h_{t-1}, x_t\right]\right) \quad (11)$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (12)$$

In the above formulas, h_t represents the hidden state at the current time step, h_{t-1} represents the hidden state from the previous time step, x_t is the output at the current time step, $\tilde{h}_t$ represents the candidate hidden state, W is the weight for the current output, σ denotes the sigmoid activation function, $\tanh$ denotes the hyperbolic tangent activation function, and $\odot$ represents element-wise multiplication.

2.3 Domain Classifier

The domain classifier is a neural network containing two fully connected layers and corresponding activation functions. Its objective is to correctly classify data into two domains (Brent crude oil, Shanghai gold) based on the feature representation. We denote the domain classifier as $G_e\left(R_F;\theta_e\right)$, where θ_e are the classifier parameters and R_F is the input feature

representation. For the $i-th$ data point, the classifier outputs a two-dimensional probability vector, corresponding to the predicted probabilities for the two domains:

$$p_i = G_e\left(G_f\left(\mathrm{x}_i;\theta_f\right);\theta_e\right) \tag{13}$$

The loss function for the domain classifier aims to perform binary classification of the domain. This paper uses the domain label set Y_e and optimizes the model using cross-entropy loss:

$$L_e\left(\theta_f,\theta_e\right) = -\mathrm{E}_{(x,y)\sim(\chi,Y_e)}\sum_{k=1}^{2}\mathrm{I}\left(y=k\right)\log\left(p_k\right) \tag{14}$$

Where p_k represents the model's predicted probability for the domain. $\mathrm{I}\left(y=k\right)$ represents the model's predicted probability for the label. When the classification is correct, the value is 1; when the classification is incorrect, the value is 0. Minimizing the discriminative loss $L_e\left(\theta_f,\theta_e\right)$ optimizes the feature extractor parameters θ_f and the classifier parameters θ_e as (15) shown:

$$\hat{\theta}_e = \arg\min_{\theta_e} L_e\left(\theta_f,\theta_e\right) \tag{15}$$

2.4 Adversarial Learning Optimization based on Uncertainty Weighting

During the training phase, the feature extractor $G_f\left(\cdot;\theta_f\right)$ cooperates with the volatility predictor $L_C\left(\theta_f,\theta_c\right)$ to minimize the volatility prediction loss, thereby enhancing the performance of the volatility prediction task. Simultaneously, the feature extractor $G_f\left(\cdot;\theta_f\right)$ aims to fool the domain discriminator $G_e\left(\cdot;\theta_e\right)$ by maximizing the domain discrimination loss $L_e\left(\theta_f,\theta_e\right)$ to learn domain-invariant representations. Therefore, the final loss defining the game between these two components is shown as:

$$L_{total} = L_{pred} - \lambda L_{domain} \tag{16}$$

The design of the loss function above essentially involves the collaborative optimization of objectives within a multi-task learning framework. In multi-task learning, a model needs to optimize multiple objectives simultaneously (e.g., regression and classification). Traditional methods typically use a weighted linear sum of losses for each task as the total loss function:

$$L_{total} = \sum_i \omega_i L_i \tag{17}$$

Here, the weights ω_i are usually set manually or determined through grid search. However, this method has significant drawbacks: model performance is highly dependent on the choice of weights, and manually or grid searching for the optimal weight combination is costly and lacks theoretical basis.

To address these issues, this paper draws on the ideas of Kendall et al., introducing homoscedastic uncertainty as task-dependent weights. It proposes a probabilistic model-based multi-task loss function that enables the model to automatically learn and optimize the relative weights of each task's loss from the data.

Uncertainty is an important criterion for measuring the robustness of a deep model. Given a labeled sample (x,y) and a model with parameters θ trained on a domain D, its uncertainty can be decomposed into:

$$P\left(y|x,D\right) = \iint \underbrace{P\left(y|\mu\right)}_{\text{Data}}\underbrace{P\left(\mu|x,\theta\right)}_{\text{Distributional}}\underbrace{P\left(\theta|D\right)}_{\text{Model}}\mathrm{d}\theta\,\mathrm{d}\mu \tag{18}$$

Here, $\mu=\theta(x)$ represents the predicted label distribution, and the three probability density functions correspond to data uncertainty, model uncertainty, and distributional uncertainty, respectively. Due to the inherent complexity of the data, its uncertainty is nearly irreducible. Model uncertainty measures the degree to which the model fits the training distribution. Distributional uncertainty quantifies the probability that an input sample is drawn from regions unfamiliar to the model. Homoscedastic uncertainty, on the other hand, depends solely on the task itself, remaining constant across all input data while varying across different tasks.

In multi-task learning, homoscedastic uncertainty can naturally be interpreted as the relative confidence or noise level among different tasks. Tasks with high uncertainty (high noise) should have their weights automatically reduced in the total loss. Therefore, we use the homoscedastic uncertainty of each task as the basis for its loss weight, implementing an adaptive adjustment mechanism where higher uncertainty leads to lower weight. Consequently, the final loss is defined as:

$$L_{total} = \frac{1}{2\sigma_{pred}^2}L_{pred} - \frac{1}{2\sigma_{domain}^2}L_{domain} + \log\left(\sigma_{pred} * \sigma_{domain}\right) \tag{19}$$

3. Experimental Design

3.1 Experimental Data

In this paper, the source domain selects Brent

crude oil futures from January 1, 2009, to December 31, 2025, as the research object, while the target domain targets Shanghai gold futures. Both domains use the 20-day historical volatility of futures prices as the dependent variable. Specific variables details in Table 1.

3.2 Environment Settings

The experiments in this chapter are implemented based on the PyTorch deep learning framework. The model adopts an uncertainty-weighted adversarial domain adaptation architecture. Core components include: a CrossFormer feature extractor (input projection layer, segment embedding layer, 4-layer Transformer encoder, multi-head attention pooling), a GRU volatility predictor, and a domain classifier incorporating a gradient reversal layer. Regarding data strategy, all crude oil data (domain label 0) is used as the source domain training set. Gold data (domain label 1) is chronologically divided: data from 2009-2021 serves as the target domain training set, 2022-2023 data as the validation set, and 2024-2025 data as the test set. The main batch size is set to 32, with validation and test batches maintaining the same scale to ensure evaluation stability. The model is trained using the AdamW optimizer with an initial learning rate of 1×10^{-3}, a weight decay coefficient of 1×10^{-4}. Gradient clipping (max gradient norm of 1.0) is applied during training to prevent gradient explosion, and the gradient reversal strength is dynamically adjusted (initial α=0.1, increasing with training epochs). To ensure experimental reproducibility, a global random seed of 42 is set.

Table 1. Comparison of Variables Between Source and Target Domains

Indicator Name	Number	Source Domain	Target Domain
Dependent Variable		Historical Volatility	Historical Volatility
Price Indicators	1	Brent Crude Oil Futures Closing Price	Gold Futures Closing Price
	2	WTI Crude Oil Futures Closing Price	COMEX Gold Futures Closing Price
	3	Oil Price Ratios to Other Fuels (Crude Oil/Natural Gas)	Gold Price Ratios to Other Metals (Gold/Platinum)
	4	Oil Price Ratios to Other Fuels (Crude Oil/Diesel)	Gold Price Ratios to Other Metals (Gold/Silver)
Trading Volume Indicators	5	WTI Crude Oil Futures Volume	COMEX Gold Futures Volume
	6	WTI Crude Oil Futures Open Interest	COMEX Gold Futures Open Interest
	7	Brent Crude Oil Futures Volume	Gold Futures Volume
	8	Brent Crude Oil Futures Open Interest	Gold Futures Open Interest
Market Structure Indicators	9	Crude Oil Spot-Futures Spread	Gold Spot-Futures Spread
Substitute Indicators	10	Fuel Oil Settlement Price	Gold ETF Settlement Price
Related Commodity Indicators	11	IPE Rotterdam Coal Futures Closing Price	WTI Crude Oil Futures Closing Price

3.3 Evaluation Metrics

To comprehensively evaluate the performance of various models in financial volatility forecasting tasks, this study selects the following key regression and prediction performance indicators: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Mean Absolute Percentage Error (MAPE). Through a comprehensive consideration of this set of indicators, we can not only accurately measure the absolute error and goodness-of-fit of model predictions but also deeply evaluate their stability and practicality in capturing data trends. This comprehensive evaluation index system lays a solid data foundation for subsequent model comparison and ablation analysis, while also providing investors with multi-dimensional insights into the model's predictive capabilities under complex market conditions.

(1) Root Mean Squared Error (RMSE)

RMSE is the square root of the Mean Squared Error. It shares the same unit as the target variable, providing a more intuitive reflection of the average error level of predictions.

$$RMSE=\sqrt{MSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \quad (20)$$

(2) Mean Absolute Error (MAE)

MAE measures the average magnitude of the absolute differences between predicted values and actual values. Unlike MSE, MAE handles errors linearly and is therefore less sensitive to outliers. Its formula is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{21}$$

(3) Coefficient of Determination (R^2)

R^2 measures the proportion of variance in the target variable that is predictable from the model. It reflects the improvement of the model's predictions over simply using the mean of the target variable. R^2 ranges from 0 to 1, with values closer to 1 indicating a better fit. Its formula is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{22}$$

Where $\bar{y}$ is the mean of the actual values.

(4) Mean Absolute Percentage Error (MAPE)

MAPE expresses prediction accuracy as a percentage of the average relative error. This allows for comparison across data with different scales. Its formula is:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{23}$$

4. Experimental Results and Analysis

4.1 Feature Analysis

Figure 3 shows a comparison of the distribution of influencing factors between the source and target domains. According to the figure, only the market structure indicator "term spread" shows a similar distribution; all other indicators exhibit significant differences. The reason might be that Brent crude oil prices and demand are primarily driven by geopolitics, OPEC+ policies, and global industrial demand. In contrast, Shanghai gold prices, while inheriting international gold prices, are significantly influenced by domestic Renminbi exchange rate expectations and domestic risk aversion. This characteristic of "distributions differing but overlapping" perfectly reflects the "commonality within individuality" of commodity markets. The overlapping regions are products of macroeconomic consensus, while the differing regions map onto the fundamental logics of each respective market. This "similar yet different, different yet similar" characteristic is precisely where the value of cross-variety transfer analysis lies.

Analyzing the differences and similarities in trading volume indicators, for example: the distribution of Brent crude oil open interest is quite concentrated, whereas the distribution of Shanghai gold open interest is more dispersed, exhibiting a bimodal pattern. The concentration in Brent crude oil open interest primarily stems from its highly institutionalized market structure and a monolithic geopolitical driving logic. This results in fund behavior showing a clear "herd effect," with open interest concentrated in the hands of a few institutions capable of reacting quickly to macro risks. In contrast, the dispersion in Shanghai gold futures open interest arises from the diversified participant structure and complex holding motives in the gold market. Gold possesses commodity, monetary, and financial attributes simultaneously, leading to a much more complex demand structure than crude oil. As the only gold futures contract in China, Shanghai gold attracts a wide range of participants, from large institutions to ordinary retail investors. This naturally results in dispersed open interest. Furthermore, the open interest of Brent crude oil and COMEX gold, as well as the volume and open interest of Brent crude oil and Shanghai gold futures, both exhibit distributional differences and overlap in certain ranges. The core reason lies in the common driving logic of global liquidity, risk aversion, and inflation shared by both asset classes, yet fundamental differences exist in terms of volume/position sensitivity to event-driven shocks and market participant structure.

Figure 4 displays heatmaps of correlation coefficients between features in the source and target domains. The numbers in Figure 4 correspond to the numbers in Table 1. The figure reveals significant differences in the correlation coefficients of influencing factors between the source and target domains, mainly reflected in: First, differences in correlation strength. Correlations among variables in Brent crude oil are generally weak and often negative. Conversely, variables related to gold generally exhibit high positive correlations. This indicates a high degree of internal consistency among indicators within the gold market. Second, several factors in the crude oil domain show negative correlations, while almost all factors in the Shanghai gold domain are positively correlated, suggesting that indicators in the gold

market tend to move in the same direction. These differences stem from the distinct asset attributes of crude oil and gold. Crude oil is an industrial raw material, with prices dominated by supply disruptions and global demand; gold is a monetary metal, with its pricing centered on real interest rates and risk aversion.

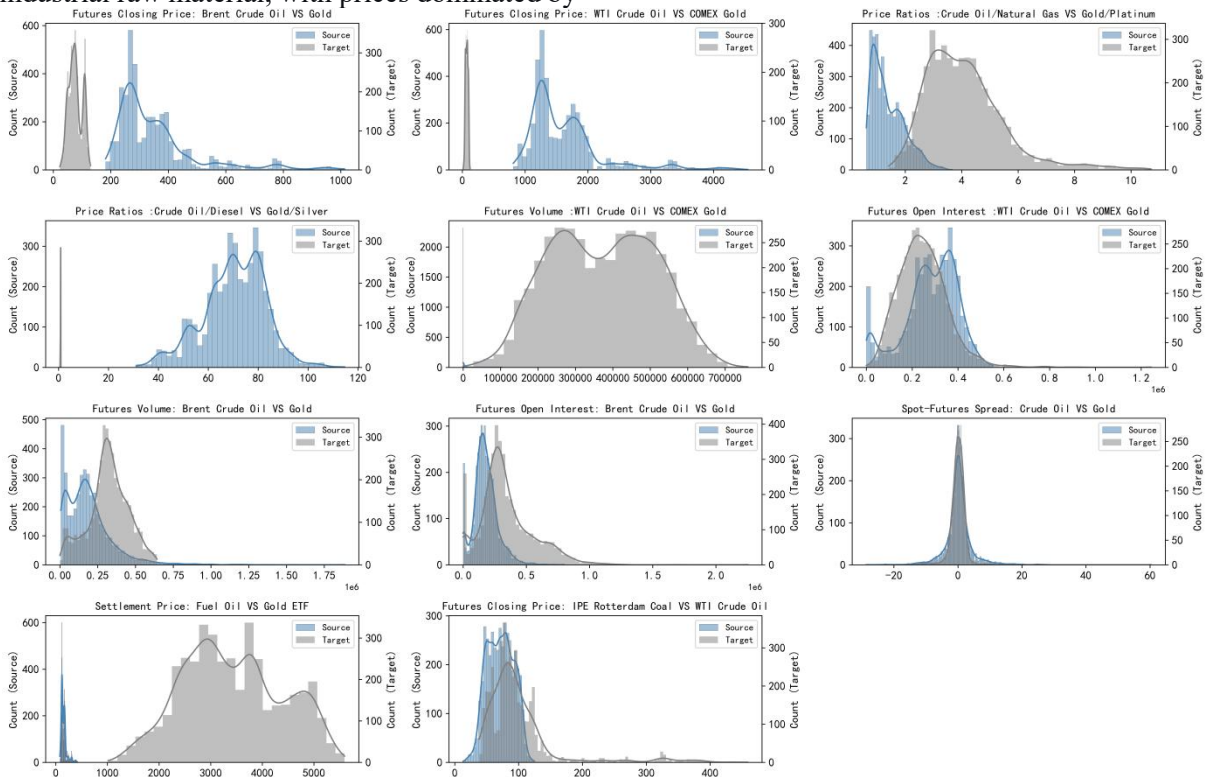

Figure 3. Comparison of Feature Distributions Between Source and Target Domain

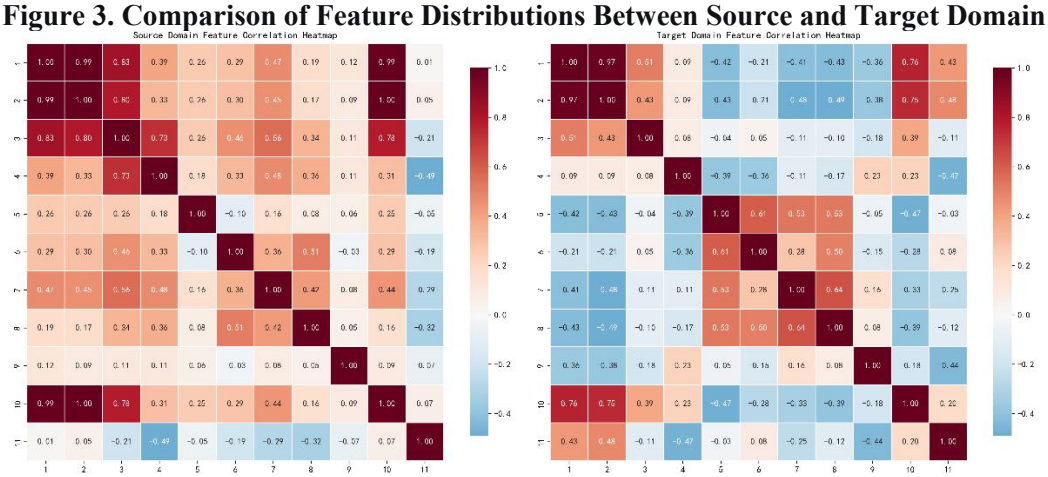

Figure 4. Heatmap of Correlation Coefficients between Source and Target Domain Features

4.2 Performance Evaluation

Based on the experimental results presented in Table 2, there are significant differences in the performance of various models on the volatility forecasting task. From the perspective of evaluation metrics, the Crossformer-GRU model performs the best, with its MAE, RMSE, and MAPE reaching as low as 0.2995, 0.4133, and 7.6652%, respectively, and an R^2 as high as 0.9920, fitting the true volatility series almost perfectly. In contrast, the Autoformer and Transformer models rank next; although they are significantly better than traditional models, they still lag considerably behind the Crossformer-GRU. The LSTM and SVR models show moderate performance, whereas the EWMN and GARCH models perform the worst, exhibiting substantial prediction errors. Furthermore, as can be intuitively observed in Figure 5, the predicted values of the Crossformer-GRU model are highly consistent with the true values at all time points. In comparison, other models generally display

systematic biases. For instance, the predictions of the GARCH and EWMN models often deviate from the true trend, and while the LSTM and Transformer models can capture the general outline, their accuracy is insufficient.

These differences indicate that the models vary in their ability to capture temporal dependencies. The GARCH and EWMN models are classical statistical models based on linear autoregression and homoscedasticity assumptions, making it difficult for them to characterize the non-linearity, long memory, and heteroscedasticity commonly found in volatility series. Although the SVR model can handle non-linearity, its kernel method is inherently limited by a finite historical window and cannot effectively utilize long-range dependency information. The LSTM model alleviates the vanishing gradient problem through its gating mechanism and can capture medium-length temporal dependencies, but its representational power is still constrained by its recurrent structure. The Transformer model introduces a self-attention mechanism, enabling parallel processing of global dependencies and demonstrating a stronger capacity for modeling long sequences, thus outperforming the LSTM model. The Autoformer model further enhances the extraction of trends and seasonality through its decomposition mechanism and auto-correlation module, leading to slightly better performance than the Transformer. The Crossformer-GRU model combines the cross-dimensional dependency modeling capability of Crossformer with the efficient sequence learning of GRU, allowing it to more fully exploit the complex patterns implicit in volatility series. Its extremely low error indicates that the model not only accurately fits the training data but also possesses strong generalization ability, effectively avoiding overfitting. Additionally, as shown in Figure 5, other models exhibit noticeable lags or deviations at certain time points (e.g., February 2024, May 2025), whereas the Crossformer-GRU is nearly unbiased, suggesting it is more sensitive and responsive to abrupt change points or extreme values.

Table 2. Experimental Results of Different Models

Model	MAE	RMSE	MAPE	R^2
GARCH	2.0394	2.9150	46.6947%	0.5967
EWMN	1.9607	2.8131	44.1151%	0.6240
SVR	1.6497	2.3643	37.4394%	0.7387
LSTM	1.5341	2.1968	35.6459%	0.7745
Transformer	1.2798	1.8152	29.2978%	0.8460
Autoformer	1.2806	1.7952	29.2253%	0.8511
Crossformer-GRU	0.2995	0.4133	7.6652%	0.9920

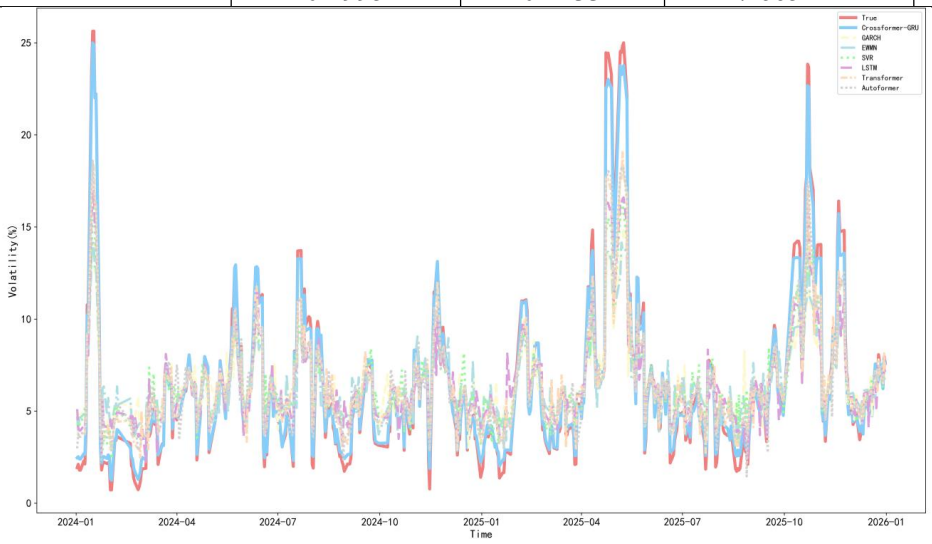

Figure 5. Comparison of Prediction Results on Test Sets Across Different Models

4.3 Comparative Experiments

Table 3 and Figure 6 show the different performances of models combining three different feature extractors with GRU in time series forecasting tasks. The experimental results indicate that models incorporating feature extractors significantly outperform the baseline

GRU model in volatility forecasting. Specifically, the model using GRU directly for prediction performs the worst across all metrics, while models combined with feature extractors show substantial performance improvements. Among them, the CrossFormer-GRU model achieves the best results on all indicators; Transformer-GRU performs slightly worse; and although the PatchTST-GRU model shows average performance, its results are still better than the GRU model alone. This suggests that effective feature extraction is crucial for volatility forecasting.

This performance difference may stem from the representational capacity of feature extractors concerning the intrinsic structure of time series and their compatibility with the temporal patterns of volatility data. Compared to the baseline GRU model, which relies solely on raw data and struggles to extract discriminative patterns from high-noise, non-stationary sequences, each feature extractor enhances GRU's temporal modeling capability through specific architectures: PatchTST captures local dependencies via patch partitioning, Transformer models global long-term correlations using self-attention, and CrossFormer achieves multi-resolution fusion of local detailed features and global long-term dependencies through its cross-scale attention mechanism, thus more comprehensively characterizing local abrupt change and overall trends in volatility. Furthermore, the performance differences among the three are closely related to their structural characteristics -- CrossFormer, due to its inherent advantage in simultaneously capturing multi-scale dynamic changes, synergizes with the uncertainty weighting mechanism to further optimize loss allocation, achieving optimal prediction accuracy and robustness; while Transformer, although strong in global dependency modeling, is relatively less sensitive to local temporal patterns; PatchTST's local partitioning strategy may, to some extent, compromise the continuity and overall trend information of the sequence, resulting in slightly inferior overall performance compared to the others.

Table 3. Comparison Results of Models with Different Feature Extractors

Model	MAE	RMSE	MAPE	R^2
GRU	1.8826	2.6803	42.5447%	0.6643
PatchTST-GRU	1.5473	2.2088	34.8578%	0.7720
Transformer-GRU	0.9684	1.3800	21.8354	0.9110
CrossFormer-GRU	0.2995	0.4133	7.6652%	0.9920

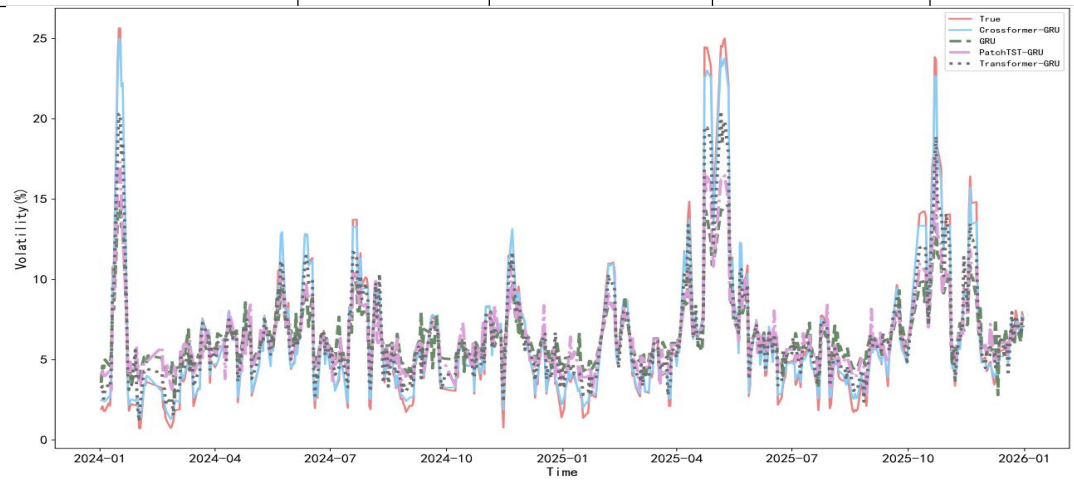

Figure 6. Comparison of Prediction Results on Test Sets with Different Feature Extractors

Table 4 and Figure 7 display the different performances of models using different volatility predictors in time series forecasting tasks. Based on feature extraction using the CrossFormer encoder, the CrossFormer-GRU model performs optimally across all evaluation metrics, with MAE, RMSE, and MAPE of 0.2995, 0.4133, and 7.6652%, respectively. In comparison, the CrossFormer-CNN and CrossFormer-MLP models show moderate prediction accuracy, while the baseline model using only CrossFormer itself for prediction performs the weakest. This trend indicates that after extracting temporal features with CrossFormer, combining it with an appropriate predictor significantly enhances volatility forecasting accuracy.

This performance difference likely arises from the varying capabilities of different predictors in

modeling temporal features. CrossFormer-GRU performs best because GRU, as a recurrent neural network, possesses gating mechanisms that effectively capture long-term dependencies and dynamic changes within time series, thereby fully utilizing the multi-scale temporal features extracted by CrossFormer. In contrast, although CNN can extract local patterns through convolution operations, its ability to model global temporal dependencies is limited; MLP, as a static network, struggles to explicitly handle the temporal structure of sequence data. The original CrossFormer model, while capable of feature extraction, is not combined with modules specifically designed for sequence prediction, resulting in underperformance when directly applied to volatility forecasting. In summary, CrossFormer-GRU achieves more accurate volatility predictions by combining CrossFormer's multi-scale feature extraction with GRU's temporal modeling strengths.

Table 4. Comparison Results of Models with Different Volatility Predictors

Model	MAE	RMSE	MAPE	R2
CrossFormer	1.4889	2.1013	33.6850%	0.7936
CrossFormer-MLP	1.3324	1.8776	30.4353%	0.8352
CrossFormer-CNN	0.8988	1.2514	20.4374%	0.9268
CrossFormer-GRU	0.2995	0.4133	7.6652%	0.9920

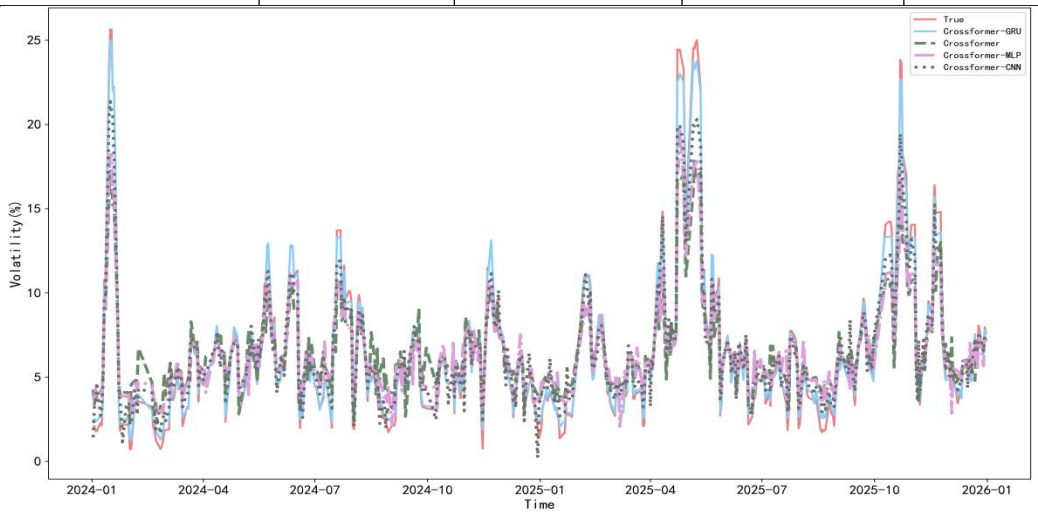

Figure 7. Comparison of Prediction Results on Test Sets with Different Volatility Predictors

Ablation experiment results are shown in Table 5. The fully implemented model achieves the lowest MAE value, indicating that both the transfer learning and uncertainty weighting mechanisms play positive roles in improving volatility prediction performance. Specifically, removing the transfer learning component causes a significant increase in the CrossFormer-GRU model's Mean Absolute Error (MAE), from 0.2995 to 1.5754. This reflects the crucial role of transfer learning in cross-domain knowledge transfer and feature generalization. Removing uncertainty weighting also leads to a performance decrease, but the magnitude is relatively smaller, suggesting that this mechanism primarily enhances model stability by optimizing loss weight allocation.

This difference mainly stems from the distinct mechanisms by which these two technologies influence the model training process. Transfer learning, by learning common temporal patterns from source domain data during the pre-training phase, provides a better initial feature representation for target domain volatility prediction. This effectively mitigates overfitting problems caused by scarce target domain data or distributional shifts. Therefore, its absence leads to a significant decline in the model's foundational representational capability. Uncertainty weighting, on the other hand, dynamically adjusts the contribution of the prediction task and the domain adaptation task to the overall loss, balancing the trade-off between feature invariance and prediction accuracy. However, its effectiveness depends on the quality of the already learned features, so the impact of its absence is relatively weaker. In conclusion, transfer learning provides the foundational feature extraction capability for the model, while uncertainty weighting further optimizes the multi-task training process. Together, they synergistically enhance the

model's generalization performance and prediction accuracy.

Table 5. Results of CrossFormer-GRU Ablation Experiments

Model Component	MAE	RMSE	MAPE	R^2
- Transfer Learning	1.5754	2.256	35.7347%	0.7621
- Uncertainty Weighting	0.3919	0.4764	10.6333%	0.9894
Fully Implemented	0.2995	0.4133	7.6652%	0.9920

According to Table 6 and Figure 8, which present the experimental results of the CrossFormer-GRU model under different prediction step lengths (forecast horizons), the model performance exhibits a pattern of first improving and then stabilizing as the step length increases. When the prediction step is 1, the model's MAE is 0.4971 and RMSE is 1.1043. Although already quite accurate, this is relatively weaker among all step lengths. As the step increases to 2, various indicators improve significantly, with MAE dropping to 0.3709. When the step reaches 5, the model performance becomes optimal. However, when the step further extends to 10, performance does not continue to improve but slightly declines, although it remains better than the results for steps 1 and 2.

The aforementioned differences may be related to the inherent laws of time series prediction and model adaptability. First, the match between prediction step length and information content. For a step of 1, although the prediction task is simple, the model might overly focus on noise in micro-fluctuations rather than effectively capturing macro trends. At step 2, the model begins to encounter longer-period dependencies, leading to performance improvement. At step 5, the prediction window coincidentally aligns with common periodic fluctuations in volatility series (such as weekly effects or macroeconomic data release cycles), enabling the model to fully extract effective information from multi-scale features, thus achieving the best fit. Second, there exists a boundary to the model's ability to model long-term dependencies. While CrossFormer-GRU combines CrossFormer's multi-scale feature extraction with GRU's temporal gating mechanism, theoretically capable of handling long sequences, the performance decline at step 10 shows that even advanced models cannot entirely avoid error accumulation and information decay for excessively long prediction windows. In long-term forecasting, future uncertainty increases, and minor early deviations may be amplified, leading to a slight decrease in overall accuracy.

Table 6. Experimental Results of CrossFormer-GRU with Different Prediction Steps

Prediction Steps	MAE	RMSE	MAPE	R^2
1	0.4971	1.1043	7.5293%	0.9430
2	0.3709	0.6461	7.3005%	0.9805
5	0.2995	0.4133	7.6652%	0.9920
10	0.3600	0.4820	7.1947%	0.9891

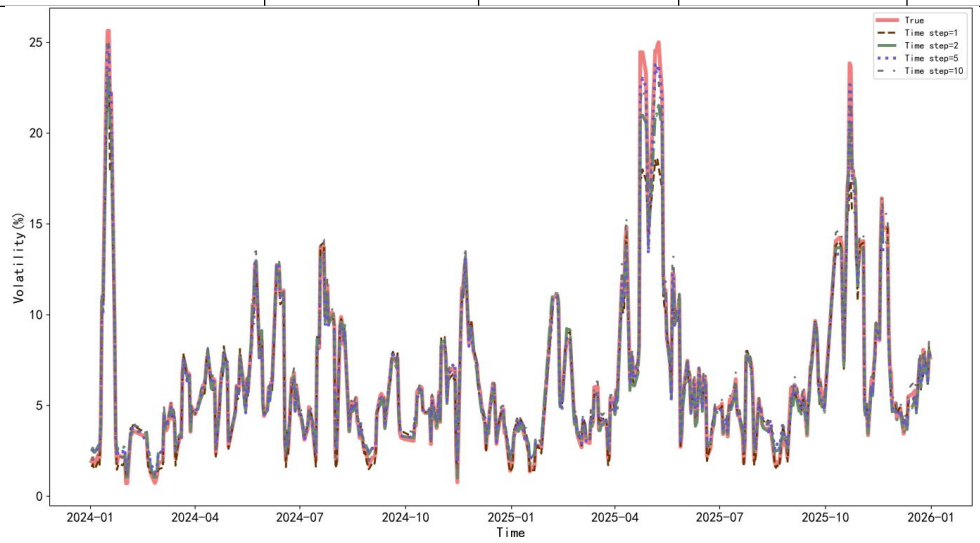

Figure 8. Comparison of Prediction Results of CrossFormer-GRU Model with Different Time Steps

5. Conclusion

Addressing the challenges of data scarcity and difficult volatility modeling in the Shanghai gold futures market, this paper proposes a CrossFormer-GRU volatility prediction model integrating transfer learning and uncertainty weighting. By using Brent crude oil futures as the source domain, leveraging an adversarial domain adaptation mechanism to extract cross-market invariant features, and introducing homoscedastic uncertainty to automatically balance the loss weights between volatility prediction and domain classification tasks, the model achieves accurate and robust predictions on the target domain. The main conclusions are as follows:

(1) Feature analysis reveals significant differences between crude oil and gold markets in terms of price drivers and participant structure. However, the common regions driven by macro factors provide a feasibility basis for transfer learning.

(2) Performance evaluation shows that CrossFormer-GRU outperforms all comparison models including GARCH, EWMN, SVR, LSTM, Transformer, and Autoformer across all evaluation metrics, particularly excelling in capturing abrupt volatility change and long-term trends.

(3) Comparative experiments confirm that Crossformer, as a feature extractor, effectively fuses multi-scale temporal information, outperforming PatchTST and standard Transformer; GRU, as a predictor, is more suitable for the dynamic evolution of volatility series than CNN or MLP.

(4) Ablation studies reveal that the contribution of transfer learning to model performance is greater than that of uncertainty weighting, and their synergistic effect further enhances generalization capability.

(5) Step length analysis indicates that the model achieves the best fit for a 5-day prediction window. Excessively long steps lead to a slight decrease in accuracy due to error accumulation.

This study provides a new approach for cross-domain knowledge transfer in financial time series forecasting. Future research could explore more source domain combinations (e.g., multiple commodities, stock index futures) and introduce graph neural networks to capture inter-variety relationships, further enhancing the model's adaptability and interpretability.

References

[1] LI Bin, WANG Wen, ZHANG Jingze. US Stock Market Shocks and the Link Between Gold Futures Prices in China and the US. Financial Market Research, 2024, (12): 120-129.

[2] ZHONG M R, SHI Y, YIN L B, et al. Research on the Impact of Investor Attention on Gold Futures Market from the Perspective of High Frequency. Journal of Systems Science and Mathematical Sciences, 2020, 40(11): 1935- 1949.

[3] ZHONG Meirui, SHI Yue, YIN Libo, et al. Research on the Impact of Investor Attention on Gold Futures Market from the Perspective of High Frequency. Journal of Systems Science and Mathematical Sciences, 2020, 40(11): 1935- 1949.

[4] Wang L, Wang Z, Qu H, et al. Optimal forecast combination based on neural networks for time series forecasting. Applied Soft Computing, 2018, 66: 1-17.

[5] Ampountolas A. Enhancing Forecasting Accuracy in Commodity and Financial Markets: Insights from GARCH and SVR Models. International Journal of Financial Studies, 2024, 12(3): 59-59.

[6] Bentes RS. Forecasting volatility in gold returns under the GARCH, IGARCH and FIGARCH frameworks: New evidence. Physical A: Statistical Mechanics and its Applications, 2015, 438355-364.

[7] Yu W, Yi zhi W, M.BL, et al. Cryptocurrency uncertainty and volatility forecasting of precious metal futures markets. Journal of Commodity Markets, 2023, 29

[8] Chen Y, Ye N, Zhang W, et al. Meta-LSTR: Meta-Learning with Long Short-Term Transformer for futures volatility prediction. Expert Systems With Applications, 2025, 265125926- 125926.

[9] Kim Christensen, Mathias Siggaard, Bezirgen Veliyev. A machine learning approach to volatility forecasting. Financial econometrics, 2025, 23, 032, https://doi.org/10.1093/jjfinec/nbac032

[10] Fischer T, Krauss C. Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions. European Journal of Operational Research, 2018 (270): 654-669.

[11] A. Vaswani, N. Shazeer, N. Parmar, et al.

Attention is all you need, in Advances in Neural Information Processing Systems 30 (NeurIPS 2017), Long Beach, CA, USA, Dec. 2017, 5998-6008.

[12] H. Zhou, S. Zhang, J. Peng, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting, in Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), vol. 35, no. 12, 2021, pp. 11106-11115.

[13] LAPTEV N. Reconstruction and Regression Loss for Time-Series Transfer Learning//24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. London: ACM, 2018: 1-8.

[14] YE R, DAI Q.A Relationship-Aligned Transfer Learning Algorithm for Time Series Forecasting. Information Sciences, 2022, 593:17-34.

[15] NGUYEN T T, YOON S. A Novel Approach to Short-Term Stock Price Movement Prediction Using Transfer Learning. Applied Sciences, 2023, 9(22):4745 (2019-11-07).

[16] Kendall A, Gal Y, Cipolla R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 7482-7491

[17] Wang R, Zhang R, Henao R. Wasserstein uncertainty estimation for adversarial domain matching. Frontiers in Big Data, 2022, 5: 878716.

