Hexagon-Net: Heterogeneous Cross-View Aligned Graph Attention Networks for Implied Volatility Surface Prediction

Kaiwei Liang*
690874999@qq.com
South China University of Technology
Guangzhou, China

Ruirui Liu*
ruirui.liu@kcl.ac.uk
King's College London
London, United Kingdom

Huichou Huang
huichou.huang@exeter.oxon.org
City University of Hong Kong
Bayescien Technologies
Hong Kong, China

Johannes Ruf
j.ruf@lse.ac.uk
London School of Economics and Political Science
London, United Kingdom

Peilin Zhao
peilinzhao@hotmail.com
Tencent AI Lab
Shenzhen, China

Qingyao Wu†
qyw@scut.edu.cn
South China University of Technology
Guangzhou, China

Abstract

Implied Volatility Surface (IVS) prediction is critical for options hedging, portfolio management, and risk control. These applications encounter significant challenges, including imbalanced data distributions and inherent uncertainties in forecasting. Recent advances relying on deep learning have led to significant progress in addressing these issues. However, several key problems have not yet been solved: (i) Moneyness and maturities of traded options change over time. Therefore, proper spatio-temporal alignment is an essential prerequisite for the downstream forecasting exercise. (ii) Different regions of the IVS are unevenly informed because of liquidity constraints and therefore should not be modeled uniformly. (iii) The complex interconnections among data points in the IVS from various perspectives—such as the well-known 'smirk' patterns across dimensions—are neither explicitly addressed nor effectively captured by existing models. To address these issues, we propose a novel end-to-end **he**terogeneous cross(**x**)-view **a**ligned **g**raph attenti**on net**work (**Hexagon-Net**), which aligns historical IVS data, learns distinctive IVS patterns, propagates predictive information, and forecasts future IVS movements simultaneously. Extensive experiments on stock index options datasets demonstrate that **Hexagon-Net** significantly and consistently outperforms the previous approaches in IVS modeling and deep learning. Additionally, we present further experiments—such as ablation studies, sensitivity analyses, and alternative configurations—to explore the reasons behind its superior performance.

CCS Concepts

• **Applied computing** → **Economics**; • **Information systems** → **Data mining**.

Keywords

Arbitrage-Free Conditions; Cross-View Alignment; Heterogeneous Graph Convolution; Implied Volatility Surface; Multi-Head Attention; Spatio-Temporal Interconnectedness

ACM Reference Format:
Kaiwei Liang, Ruirui Liu, Huichou Huang, Johannes Ruf, Peilin Zhao, and Qingyao Wu. 2025. Hexagon-Net: Heterogeneous Cross-View Aligned Graph Attention Networks for Implied Volatility Surface Prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3736996

1 Introduction

Volatility, a key factor influencing option prices, has attracted significant research interest from both academia and the financial industry. Accurate modeling and forecasting of volatility are essential for informed decision-making across a variety of investment activities, including option pricing and hedging, as well as portfolio and risk management. In particular, implied volatility—derived from the Black-Scholes option pricing model [5]—provides valuable forward-looking insights into the underlying markets. The implied volatility surface (IVS) for a given underlying asset consists of a three-dimensional set of implied volatility values that vary with time-to-maturity (τ) and log-moneyness (m), which depends on the spot-to-strike price ratio S/K. Predicting the IVS (see Figure 1) is therefore critically important for downstream investment applications.

Over the past decades, numerous IVS models have been developed, including stochastic volatility models [12], parametric approaches [9], and nonparametric methods [7], among others. Although these models are mathematically rigorous and economically meaningful, their simplified frameworks often struggle to fully capture the complex and evolving dynamics of the IVS. With the rapid advancements in deep learning (DL) techniques, recent

*Authors contributed equally to this research.
†Corresponding Author

KDD '25, Toronto, ON, Canada.

ACM ISBN 979-8-4007-1454-2/25/08
https://doi.org/10.1145/3711896.3736996

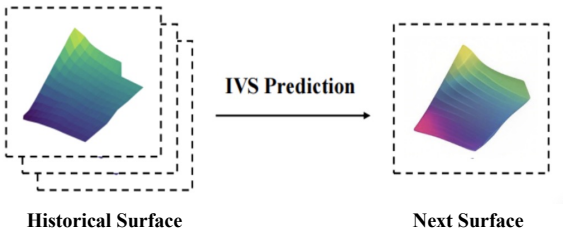

Figure 1: IVS prediction aims to predict the next IVS given historical data.

applied studies have shown promising potential for using these methods to improve IVS prediction [1, 22, 26, 32, 17, 31, 6].

Despite of the significant progress made, several challenges remain for IVS prediction. First, only a limited number of options are actively traded on a daily basis, implying that the set of time-to-maturity (τ) and log-moneyness (m) values are neither continuous nor consistently available over time. As a consequence, the IVS data are not aligned across both the cross-sectional and time-series dimensions. Traditional approaches for addressing this misalignment either lack generalizability and stochasticity—such as the deterministic volatility function [9]—or fail to accommodate the heterogeneity inherent in the data, as is the case with the Nadaraya–Watson estimator [14]. Second, thin option markets at certain strike prices and maturities result in varying degrees of informativeness and distinct dynamics across different regions of the IVS, underscoring the need for models to learn and distinguish meaningful features from the data. Third, existing models often fail to explicitly consider or effectively capture the interconnections between data points from different perspectives of the IVS—such as term structure patterns. In summary, the effective propagation of predictive information across the spatio-temporal structure of the IVS is essential for accurate modeling and forecasting. Yet, this remains insufficiently addressed in the existing literature, leading to unstable or suboptimal predictive performance.

In this paper, we propose an end-to-end **he**terogeneous cross(**x**)-view **a**ligned **g**raph attenti**on net**work, **Hexagon-Net**, to tackle the above mentioned challenges simultaneously in IVS reconstruction and prediction tasks. Our contributions are summarized as follows:

(1) We propose a heterogeneous graph grid alignment module that constructs a unified grid to aggregate information from individual options and subsequently propagates the processed information back to them. This mechanism is crucial for enhancing model performance, as it (i) addresses inconsistencies in data availability, (ii) enables the transfer of richer feature representations from highly liquid options to thinly traded ones, and (iii) facilitates the learning of distinct feature representations across different regions of the IVS.
(2) Given the large number of data points on each surface, it is challenging to extract valuable signals from noisy or ambiguous option features while preserving the well-known smirk patterns [30]. To tackle this, we introduce a Cross-View Transformer (**CVT**) module that captures and integrates economically meaningful multi-scale spatio-temporal relationships—both within the IVS across log-moneyness and maturity, and across IVSs over time. In addition, we enforce no-arbitrage constraints on the IVS.
(3) Through extensive experiments and ablation studies on three of the most prominent stock index options datasets—S&P500, NASDAQ100, and STOXX50—we demonstrate that the proposed **Hexagon-Net** significantly outperforms a wide range of baselines. These include traditional IVS benchmark models from the mathematical finance literature, models incorporating standard DL techniques, and state-of-the-art DL approaches for time-series forecasting. Moreover, the results are robust across different forecasting horizons and under various sensitivity analyses.
(4) To better understand the superior performance of the proposed **Hexagon-Net**, we compare it against its variants augmented with additional contrastive and stochastic learning components, designed to handle the heterogeneous and evolving nature of market liquidity. This analysis shows that these enhancements offer little to no statistically significant improvement in accuracy or stability over the standard **Hexagon-Net**.

2 Related Work

Existing literature in the finance domain primarily relies on linear techniques to model IVS dynamics. For example, [7, 23] apply Principal Component Analysis (PCA) to IVS data and show that the first three eigenmodes are both economically meaningful—interpretable as loadings on common latent factors—and statistically sufficient to capture most of the variation in the IVS. These eigenmodes approximate the IVS through linear combinations, with their temporal evolution modeled using autoregressive processes. Acknowledging the degenerate nature of IVS data, [10] propose a semiparametric factor model that captures local IVS dynamics by exploiting expiry effects. Stochastic volatility models [4] are also widely used in mathematical finance. The Stochastic Volatility Inspired (SVI) model introduced by [11] imposes no-arbitrage conditions and performs well on real-world data. [13] further refine the SVI framework by simplifying the arbitrage constraints, resulting in the SSVI model. More recently, [3] improve the performance of traditional parametric models by using a neural network to model the residuals.

In recent years, both academics and industry practitioners have increasingly turned to machine and DL methods for IVS prediction, yielding promising results. For instance, [32] incorporate prior domain knowledge by proposing a novel activation function that accounts for the volatility smile. However, their approach does not explicitly capture IVS dynamics when mapping τ and m to implied volatility. [1] integrate a standard arbitrage-free IVS model with a neural network architecture, demonstrating robust performance even when the IVS data are sparse, noisy, or erroneous. [31] apply a variational autoencoder (VAE) to learn IVS feature representations, followed by an LSTM model for IVS prediction.

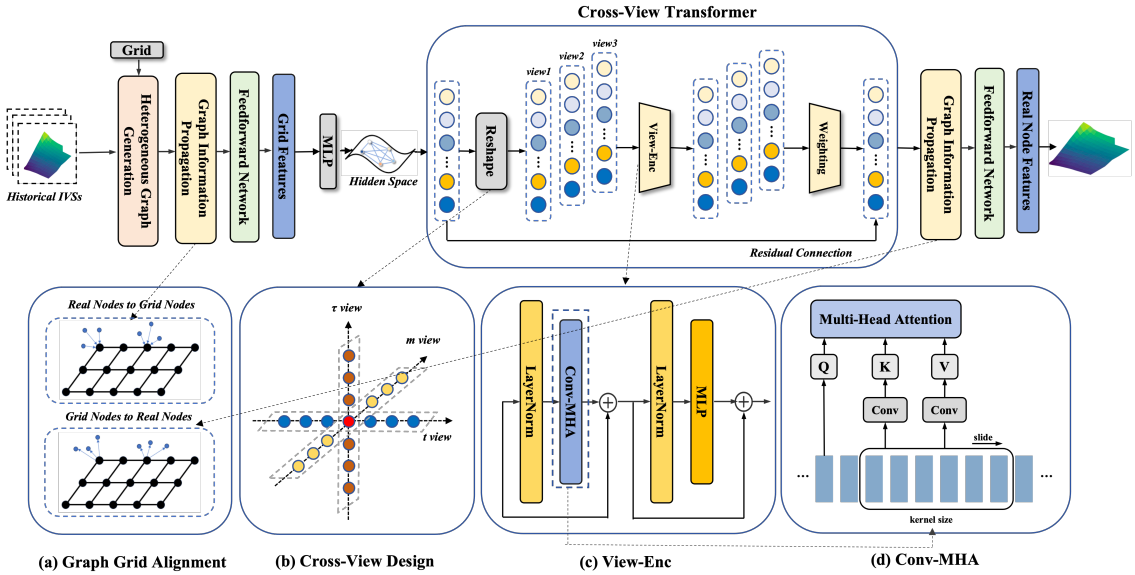

Figure 2: Architecture of Hexagon-Net. We use heterogeneous graph neural networks to align historical data onto a unified grid. After aligning all historical IVS using the same grid, we feed the aligned IVS data into CVT, which considers the Inner-IVS relations and Inter-IVS dynamics from different views. Finally, the information is propagated to each option using the obtained grid features in order to predict the IVS.

3 Implied Volatility Surfaces

Let $P_{\text{mkt}}(K,\tau)$ denote the market price of a European option with time-to-maturity $\tau > 0$ and strike price $K > 0$. Let S, K, τ, σ, r, and δ represent the underlying asset price, strike price, time-to-maturity, volatility, risk-free rate, and dividend yield, respectively. Under the classical Black-Scholes framework, the price of a European call option is given by

$$P_{\text{BS}}(S,K,\tau,\sigma,r,\delta) = Se^{-\delta\tau}\Phi(d_1) - Ke^{-r\tau}\Phi(d_2),$$

where

$$d_1 = \frac{\ln(\frac{S}{K}) + (r - \delta + 0.5\sigma^2)\tau}{\sigma\sqrt{\tau}}; \quad d_2 = d_1 - \sigma\sqrt{\tau},$$

and $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution. A similar formula applies for the price of European put options.

The implied volatility of an option is then defined as the value of σ that solves $P_{\text{BS}}(S,K,\tau,\sigma,r,\delta) = P_{\text{mkt}}(K,\tau)$ given observed market prices. The collection of implied volatilities across different strikes and maturities at a given time defines the IVS.

3.1 IVS Modeling and Prediction

The IVS at time t is represented by $I_t \in \mathbb{R}^{N_t \times 3}$, where N_t denotes the number of available option observations at time t. Each row in I_t corresponds to an option characterized by three features: time-to-maturity (τ), log-moneyness (m), and implied volatility (σ). We define $\mathcal{G}_t \in \mathbb{R}^{N_t \times 2}$ as the grid of inputs in the (τ, m) space, and $V_t \in \mathbb{R}^{N_t}$ as the corresponding implied volatilities.

An IVS is modeled as a function of τ and m. We fit the model using a lead-lag sequential setup: given historical data $I_{1:T} = [I_1, \ldots, I_T]$, grid $\mathcal{G}_{T+1} = (\tau_i, m_i)_{i=1}^{N_{T+1}}$ for the next period, and time embedding $\mathcal{T}_{1:T+1}$, we aim to predict V_{T+1} using a predictive model f; i.e.,

$$\hat{V}_{T+1} = f(I_{1:T}, \mathcal{G}_{T+1}, \mathcal{T}_{1:T+1}; \theta),$$

where θ is a set of model parameters.

Moreover, besides future IVS prediction we train the model simultaneously on a second task, namely IVS mask reconstruction. For the future prediction task, the model ingests the complete sequence $I_{1:T}$. For the mask reconstruction task, we input IVS data with certain points masked: at each time t, we randomly select $N_{t,\text{mask}} = \lfloor pN_t \rfloor$ points to mask, while the remaining $N_t - N_{t,\text{mask}}$ points remain visible. Here, $p \in [0,1]$ is the mask probability (set to 0.1 in our experiments) and $\lfloor \cdot \rfloor$ denotes the floor function.

3.2 Static Arbitrage-Free Conditions for IVS

Absence of static arbitrage is a fundamental assumption in asset pricing, implying that investors should not be able to extract risk-free, costless profits from inconsistencies among the market prices of an asset. The literature has identified necessary and sufficient conditions to ensure that the IVS is free of static arbitrage. Following the approach of DeepSmooth [1], we incorporate these conditions (see Appendix A) by applying soft constraints as penalty terms in the loss function, see also [8]. Additional implementation details are provided in Subsection 4.5.

4 Methodology

We present the proposed **Hexagon-Net** in Figure 2. Given a sequence $I_{1:T}$ of historical IVS data, we introduce a heterogeneous graph grid aligner that first constructs a unified grid to aggregate IVS information from individual options and then propagates the

processed information back to each option. In addition, a cross-view feature extraction strategy is employed to capture and fuse multi-facet spatio-temporal relations—both inter-IVS dynamics over time and intra-IVS structure across log-moneyness and maturity—while respecting the characteristic "smirk" patterns via learnable, dynamic weights.

4.1 Heterogeneous Graph Grid Aligner

4.1.1 Motivation. Considering the changing grids required for IVS at different times, we introduce an aligner based on heterogeneous graph neural networks (Figure 2). We incorporate a virtual alignment grid that transfers IVS information to adjacent grid points for alignment. After learning the aligned grid features, these are propagated back to the actual data nodes, where a neural network is used to predict the implied volatilities.

4.1.2 Grid design. To construct a unified grid over the entire historical IVS data, we partition the (τ, m) space into a fixed, aligned grid of size $A \times B$, where A and B denote the number of discretization levels along the time-to-maturity (τ) and log-moneyness (m) dimensions, respectively., i.e.,

$$\mathcal{G}^{\text{al}} = \left\{(\tau, m) \mid \tau \in \mathcal{G}^{\tau},\ m \in \mathcal{G}^{m}\right\}.$$

Here, $\mathcal{G}^{\tau}$ consists of A reference points selected according to the empirical percentiles of the τ observations, such that each interval between consecutive points contains approximately the same number of τ values. Similarly, $\mathcal{G}^{m}$ consists of B points according to the empirical percentile of the m observations. In our implementation, we set $A = 25$ and $B = 40$, resulting in $A \times B$ grid nodes per IVS graph.

4.1.3 Construction of Heterogeneous Graphs. The IVS graph consists of two types of nodes: real (data) nodes $\mathbb{V}_{\text{r}}$ and grid nodes $\mathbb{V}_{\text{g}}$ (constructed in Subsection 4.1.2). In the next subsections, we shall define three types of edges to capture interactions between these nodes: bidirectional edges $R_{\text{g}\leftrightarrow\text{g}}$ between grid nodes, unidirectional edges from real nodes to grid nodes $R_{\text{r}\to\text{g}}$, and unidirectional edges from grid nodes to real nodes $R_{\text{g}\to\text{r}}$.

Each node v is associated with two key attributes, time-to-maturity τ and log-moneyness m, as well as a feature vector. Each edge similarly carries its own feature vector. The node attributes capture relevant properties of the IVS, while the edge attributes are obtained by calculating the absolute differences of the corresponding key attributes (τ and m) between the connected node pair.

4.1.4 Grid Node-Grid Node Connection. We denote the connectivity among grid nodes by $R_{g\leftrightarrow g}$, with the corresponding adjacency matrix defined as $\mathbb{I}(\tau_i = \tau_j \vee m_i = m_j)$. It is worth noting that the precise form of grid connectivity is not consequential in what follows, as grid nodes do not directly communicate with each other.

4.1.5 Real Node-Grid Node Connection. To enable information flow from real nodes to grid nodes, we define directional connections between nodes $v_i \in \mathbb{V}_{\text{r}}$ and $v_j \in \mathbb{V}_{\text{g}}$ when they are close in the (τ, m) space. Note that these connections are directional and in particular $R_{\text{r}\to\text{g}}$ and $R_{\text{g}\to\text{r}}$, do not necessarily represent the same edges.

We first compute the distance matrix D_{rg} between all real and grid nodes, using the Euclidean distance in the (τ, m)-space. The adjacency matrix $A_{\text{r}\to\text{g}}$, which defines connections from real to grid nodes, is then given by $A^{i,j}_{\text{r}\to\text{g}} = \mathbb{I}(j = \arg\min_k D^{i,k}_{\text{rg}})$, i.e., each real node is connected to its nearest grid node. Similarly, we consider the adjacency matrix $A_{\text{g}\to\text{r}}$ for the reverse directional connections.

4.2 Propagation from Real Nodes to Grid Nodes

Using a heterogeneous graph network, we propagate information from the real nodes in $\mathbb{V}_{\text{r}}$ to the grid nodes in $\mathbb{V}_{\text{g}}$ via edges in the relation set $R_{\text{r}\to\text{g}}$.

For a given grid node, we consider all real nodes that are connected to it. For each such real node, a message function computes two feature vectors, h and $\tilde{h}$, based on the input attribute x of the real node and the edge attribute x_{e} associated with the connecting edge:

$$\text{Message}_{\text{r}\to\text{g}}(x, x_{\text{e}}) = \left(h, \tilde{h}\right).$$

The message function is implemented as a multi-layer perceptron (MLP).

Each connected real node contributes a pair of intermediate features, $(h^i, \tilde{h}^i)$, where h^i is the candidate node feature and $\tilde{h}^i$ determines its aggregation weight. These features are aggregated using a weighted sum:

$$\overline{h} = \text{Aggregate}_{\text{g}}\left(\{(h^i, \tilde{h}^i)\}\right) = \sum_i a_i h^i,$$

where the sum ranges over all real nodes connected to the given grid node. The weights a_i are computed using another MLP $f_{\text{r}\to\text{g},w}$ applied to the edge features $\tilde{h}^i$ and normalized via a softmax:

$$a_i = \frac{\exp\left(f_{\text{r}\to\text{g},w}(\tilde{h}^i)\right)}{\sum_j \exp\left(f_{\text{r}\to\text{g},w}(\tilde{h}^j)\right)}.$$

Finally, the feature of the grid node is updated by adding the aggregated message to its existing feature:

$$h'_{\text{g}} = \text{Update}_{\text{g}}(h_{\text{g}}, \overline{h}) = h_{\text{g}} + \overline{h},$$

where h_{g} is the current feature of the grid node, and $\overline{h}$ is the aggregated message from neighboring real nodes.

Each MLP employed in this study is a three-layer fully connected network with a default hidden layer size of 64. The activation function used between consecutive linear layers is GELU.

4.3 Cross-View Transformer (CVT)

4.3.1 Motivation. The **CVT** architecture is tailored to the distinctive characteristics of the IVS. Implied volatilities typically display a smile-shaped profile as a function of log-moneyness m for fixed time-to-maturity τ, and exhibit similar structure across varying τ for fixed m. This empirical behavior motivates the design of **CVT** to effectively capture such "smirk" patterns and to learn economically meaningful feature representations from multiple perspectives of the IVS.

To address the computational challenges posed by the thousands of data points in each IVS, we employ a convolutional multi-head attention mechanism (**Conv-MHA**). It reduces computational complexity by a factor of $\kappa \times \kappa$, where κ denotes the kernel size of the convolutional layers. In our implementation, we set $\kappa = 4$.

4.3.2 Cross-View Design. The structure of **CVT** is illustrated in Figures 2(b) and (c). We begin by projecting the aligned IVS graph into a hidden feature space $H^0 \in \mathbb{R}^{T\times P\times C}$ using an MLP, where $P = A \times B$ is the number of grid points and C denotes the hidden layer dimensionality.

Subsequently, L (in our implementation $L = 2$) cross-view layers are applied to hierarchically extract cross-view features, allowing us to incorporate information from multiple views in each layer.

As depicted in Figure 2(b), the historical IVSs $I_{1:T}$ are defined over the triplet (τ, m, t). We define a *view* of the historical IVS as a dependent variable of a subset of (τ, m, t). In the l-th layer, the hidden representation from the previous layer, $H^{l-1} \in \mathbb{R}^{T\times P\times C}$, is first reshaped to $H^{l-1,v} \in \mathbb{R}^{\mathcal{N}\times\mathcal{S}\times C}$ and passed into the **View-Enc** module. Here, $\mathcal{N}$ is the view-specific sample size, and $\mathcal{S}$ denotes the view length, which varies across views.

For instance, in the τ-view, the tensor H^{l-1} is reshaped such that $N = T \times B$ and $\mathcal{S} = A$, corresponding to T time points and B log-moneyness levels. The set of views utilized in CVT is $\{\tau, m, t\}$. After encoding, the resulting hidden feature H^l_{view} for each view is reshaped back to the original dimensionality $\mathbb{R}^{T\times P\times C}$ for further processing.

4.3.3 View-Enc. The architecture of the **View-Enc** module is presented in Figure 2(c). At the l-th layer, **View-Enc** receives as input the hidden feature tensor $H^{l-1} \in \mathbb{R}^{\mathcal{N}\times\mathcal{S}\times C}$ from the previous layer. These features are first normalized and then passed to **Conv-MHA**, illustrated in Figure 2(d).

Within **Conv-MHA**, the input is first linearly projected to form a query matrix $\mathcal{Q}^l \in \mathbb{R}^{\mathcal{N}\times\mathcal{S}\times C}$. To compute the key matrix $\mathcal{K}^l \in \mathbb{R}^{\mathcal{N}\times\lfloor\mathcal{S}/\kappa\rfloor\times C}$ and the value matrix $\mathcal{V}^l \in \mathbb{R}^{\mathcal{N}\times\lfloor\mathcal{S}/\kappa\rfloor\times C}$, we apply a one-dimensional convolution over the sequence dimension:

$$\mathcal{K}^l_{i,:,j} = b^{\mathcal{K}}_j + \sum_c H^{l-1}_{i,:,c} W^{\mathcal{K}}_{j,:,c},$$
$$\mathcal{V}^l_{i,:,j} = b^{\mathcal{V}}_j + \sum_c H^{l-1}_{i,:,c} W^{\mathcal{V}}_{j,:,c},$$

where $b^{\mathcal{K}}, b^{\mathcal{V}} \in \mathbb{R}^C$ are bias vectors and $W^{\mathcal{K}}, W^{\mathcal{V}} \in \mathbb{R}^{C\times\kappa\times C}$ are convolution kernels with both kernel size and stride equal to κ.

This convolutional mechanism enables the model to extract local structure from the feature sequence while simultaneously reducing its length, thereby improving computational efficiency.

Following the convolutional encoding, the multi-head attention (MHA) mechanism captures both intra-IVS dependencies and inter-IVS dynamics. The attention output is computed as:

$$h_{\text{MHA}} = \text{softmax}\left(\frac{\mathcal{Q}^l(\mathcal{K}^l)^\top}{\sqrt{d_f}}\right)\mathcal{V}^l,$$

where d_f denotes a scaling factor that stabilizes gradients by adjusting for the dimensionality of the feature space [24]. After layer normalization and transformation by an MLP, the resulting hidden features are denoted by H^l_{view}.

4.3.4 Cross-View Attention Weighting. Simple aggregation strategies such as summation or averaging fail to account for variations in the informativeness of features across different views when modeling and predicting IVS. To address this, we design an attention-based weighting mechanism that dynamically allocates weights across views.

Given N_v view-specific feature representations (here, $N_v = 3$),

$$H^l_{\text{view}_1}, \ldots, H^l_{\text{view}_{N_v}} \in \mathbb{R}^{T\times P\times C},$$

we compute attention weights as

$$w^{l,t,p}_{\text{view}_i} = \frac{\exp(d^{l,t,p}_{\text{view}_i})}{\sum_{j=1}^{N_v}\exp(d^{l,t,p}_{\text{view}_j})},$$

where

$$d^{l,t,p}_{\text{view}_i} = \Psi^\top \text{GELU}\left(H^{l,t,p}_{\text{view}_i} W + \Gamma\right),$$

with learnable parameters $W \in \mathbb{R}^{C\times C}$, $\Gamma \in \mathbb{R}^C$, and $\Psi \in \mathbb{R}^C$. Here, $H^{l,t,p}_{\text{view}_i} \in \mathbb{R}^C$ denotes the feature vector for view i at layer l, time step t, and grid index p. The final fused feature is then obtained by a weighted sum:

$$H^{l,t,p} = \sum_{j=1}^{N_v} w^{l,t,p}_{\text{view}_j} H^{l,t,p}_{\text{view}_j}.$$

4.4 Propagation from Grid Nodes to Real Nodes

After feature learning has been performed on the grid nodes—either through reconstruction of masked values or prediction of future values—the resulting grid node features are propagated back to the real nodes. This step enables the reconstruction or prediction of the IVS at real locations. We propagate information from the grid nodes in $\mathbb{V}_{\text{g}}$ to the real nodes in $\mathbb{V}_{\text{r}}$ via edges in the relation set $R_{\text{g}\to\text{r}}$.

For a given real node, we consider all grid nodes connected to it. (Usually this is only one single grid point.) For each such grid node, a message function computes a pair of feature vectors, h and $\tilde{h}$, based on the grid node feature and the associated edge attribute:

$$\text{Message}_{\text{g}\to\text{r}}(h, h_{\text{e}}) = (x, \tilde{x}),$$

where the message function is again implemented via an MLP.

Each connected grid node contributes a message $(x^i, \tilde{x}^i)$ to the real node. These are aggregated using a weighted sum:

$$\overline{x} = \text{Aggregate}_{\text{r}}\left(\{(x^i, \tilde{x}^i)\}\right) = \sum_i a_i x^i,$$

where the sum is over all grid nodes connected to the real node. The attention weights a_i are determined by applying another MLP $f_{\text{g}\to\text{r},w}$ as follows:

$$a_i = \frac{\exp\left(f_{\text{g}\to\text{r},w}(\tilde{h}^i)\right)}{\sum_j \exp\left(f_{\text{g}\to\text{r},w}(\tilde{h}^j)\right)}.$$

Finally, the updated feature x'_{r} for the real node is computed using a residual update:

$$x'_{\text{r}} = \text{Update}_{\text{r}}(x_{\text{r}}, \overline{x}) = x_{\text{r}} + \overline{x},$$

where x_{r} is the real node feature before the update.

4.5 Static Arbitrage-Free Loss

In addition to the alignment grid $\mathcal{G}^{\text{al}}$ used for historical IVS matching, we design two auxiliary grids, $\mathcal{G}^{C34}$ and $\mathcal{G}^{C5}$, to enforce static arbitrage constraints. Specifically, $\mathcal{G}^{C34}$ targets Conditions 3 and 4 of Proposition 1 in Appendix A, while $\mathcal{G}^{C5}$ targets Condition 5. Full details of the grid construction are provided in Appendix B.

Conditions 1 and 2 are satisfied due to the use of SoftPlus activations in our neural architecture, which guarantees non-negativity and twice differentiability. Condition 6 is not applicable as we model the implied-volatility function only for strictly positive time-to-maturity $\tau > 0$. To enforce the remaining static no-arbitrage constraints, we define the following loss terms:

$$\mathcal{L}_{C3} = \frac{1}{|\mathcal{G}^{C34}|} \sum_{(\tau,m)\in\mathcal{G}^{C34}} \max\left(0,\ -l_{\text{cal}}(m,\tau)\right),$$

$$\mathcal{L}_{C4} = \frac{1}{|\mathcal{G}^{C34}|} \sum_{(\tau,m)\in\mathcal{G}^{C34}} \max\left(0,\ -l_{\text{but}}(m,\tau)\right),$$

$$\mathcal{L}_{C5} = \frac{1}{|\mathcal{G}^{C5}|} \sum_{(\tau,m)\in\mathcal{G}^{C5}} \left|\partial^2_{mm}\sigma^2_t(m,\tau)\right|,$$

where $|\mathcal{G}^{C34}|$ and $|\mathcal{G}^{C5}|$ denote the respective numbers of grid points. The total static arbitrage-free loss is defined as the sum of these components:

$$\mathcal{L}_{\text{SAF}} = \mathcal{L}_{C3} + \mathcal{L}_{C4} + \mathcal{L}_{C5}.$$

4.6 Prediction and Loss Function

Let X_{T+1} denote the real node features, obtained through the message propagation mechanism described in Subsection 4.4, which provides information from grid nodes to their corresponding future real nodes. The prediction of the IVS is then given by

$$\hat{V}_{T+1} = f_o\left(X_{T+1}, \mathcal{G}_{T+1}, \mathcal{T}_{1:T+1}\right),$$

where f_o is implemented as an MLP. An analogous formulation is used for the masked IVS reconstruction task.

The training objective for **Hexagon-Net** is a combination of the IVS prediction loss and the static arbitrage-free loss:

$$\mathcal{L}_{\text{pred}} = \text{MSE}(\hat{V}_{T+1}, V_{T+1}) + \beta \cdot \mathcal{L}_{\text{SAF}}.$$

Here MSE denotes the mean squared error between the predicted or reconstructed IVS and the ground truth, and β is a hyperparameter controlling the strength of the arbitrage penalty. For the masking task, we have an analogous expression with the same hyperparameter β.

5 Experiments

5.1 Datasets

Following [2, 32], we conduct empirical experiments on stock index options using data for the S&P 500, NASDAQ100, and STOXX50 indices. These markets are well-suited for benchmarking due to their deep liquidity, as evidenced by consistently high trading volumes, although certain regions of the IVS may still suffer from sparse observations. We obtain daily option data from OptionMetrics, spanning the period from January 2010 to June 2019, and clean and pre-process the dataset following the approach in [21].

While we focus here on European-style equity options, the proposed methodology is applicable to options on other asset classes—such as foreign exchange and commodities—provided the markets are sufficiently liquid for implied volatilities to reflect true market conditions. This requires that trade quotations permit the extraction of implied volatilities; for example, in foreign exchange markets, major investment banks provide reliable implied-volatility data.

5.2 Experimental Setup

We train, validate, and test all models on the three datasets to assess the qualitative robustness of the experimental results. Each dataset is partitioned into non-overlapping sub-samples of 150 trading days. Within each sub-sample, we further divide the data into training, validation, and testing sets using a 4:3:3 ratio. Model optimization is performed using the Adam optimizer with a fixed learning rate of 5×10^{-4}. All models are trained for 30,000 epochs. At each epoch, the model parameters yielding the best performance on the validation set are selected and subsequently evaluated on the test set. Each model is jointly trained and evaluated on two tasks: future prediction and mask reconstruction of the IVS.

We evaluate model performance using the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE), defined respectively as

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

$$\text{MAPE}(\hat{y}, y) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|,$$

where n denotes the number of data points to be reconstructed or predicted. Lower values of RMSE and MAPE indicate better model performance.

5.3 Baselines

We compare **Hexagon-Net** against a suite of baselines spanning three main categories:

(1) Mathematical Finance (MF):

- **SSVI** [13]: An enhanced version of the SVI model [11], which incorporates simplified no-arbitrage constraints for modeling IVS.

(2) DL Methods with a Time-Series Focus:

- **Transformer** [24]: A foundational architecture leveraging self-attention.
- **GAT** [25]: Utilizes Graph Attention Networks to learn relationships.
- **DA-RNN** [18]: Incorporates a temporal attentive aggregation layer utilizing recurrent recurrent neural networks
- **DLinear** [29]: Employs a decomposition approach combined with linear layers.
- **NLinear** [29]: A simple linear baseline using additive updates to shift model prediction towards ground truth.
- **Autoformer** [27]: Leverages an autocorrelation mechanism to learn inter-series dependencies, together with seasonal-trend decomposition.
- **FEDformer** [34]: Proposes a frequency-enhanced Transformer with reduced computational complexity.

Table 1: IVS Prediction Performance (%). Bold indicates the best result.

		S&P500				NASDAQ100				STOXX50			
		Future Prediction		Mask Reconstruction		Future Prediction		Mask Reconstruction		Future Prediction		Mask Reconstruction	
	Model	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
MF	**SSVI**	3.14±0.32	9.27±0.99	3.13±0.32	9.22±0.99	3.23±0.27	9.17±1.05	3.20±0.27	9.23±1.06	3.78±0.44	12.46±2.22	3.84±0.48	12.58±2.23
TS	**Transformer**	3.84±0.14	10.99±0.63	3.13±0.27	8.67±0.52	3.86±0.48	11.27±1.52	3.18±0.55	8.89±1.36	3.70±0.33	11.68±1.19	3.25±0.33	9.85±1.04
	GAT	3.05±0.84	8.75±0.65	2.21±0.08	6.93±0.16	3.27±0.10	9.52±0.43	2.50±0.10	7.87±0.38	3.45±0.37	10.59±0.95	3.11±0.36	9.65±0.90
	DA-RNN	2.82±0.11	8.60±0.48	2.25±0.08	6.87±0.22	3.29±0.15	9.59±0.36	2.51±0.10	7.86±0.30	3.43±0.12	10.71±0.39	3.06±0.16	9.55±0.51
	DLinear	3.20±0.41	10.10±0.87	2.79±0.60	8.21±1.48	3.12±0.06	9.55±0.20	2.42±0.10	7.39±0.26	3.33±0.07	10.33±0.36	3.05±0.19	9.14±0.44
	NLinear	3.16±0.18	10.15±0.64	2.71±0.29	8.22±0.90	3.72±0.20	10.93±0.54	3.03±0.29	8.51±0.45	3.73±0.30	11.46±1.00	3.53±0.44	10.57±1.65
	Autoformer	5.01±0.32	19.36±1.64	2.47±0.16	7.64±0.41	5.04±0.16	17.30±0.60	8.30±0.36	7.74±0.17	4.76±0.13	16.86±0.51	3.09±0.29	9.64±0.67
	FEDformer	2.83±0.08	8.94±0.31	2.27±0.07	7.13±0.27	3.27±0.13	9.94±0.40	2.57±0.10	7.93±0.42	3.42±0.12	10.62±0.45	3.00±0.15	9.33±0.37
	Informer	2.82±0.09	9.17±0.43	2.24±0.10	7.10±0.33	3.31±0.09	9.95±0.36	2.57±0.12	7.91±0.42	3.24±0.08	10.31±0.34	2.83±0.10	8.83±0.27
	AST	2.47±0.35	7.64±1.86	2.12±0.32	6.30±1.48	3.13±0.06	9.35±0.19	2.46±0.06	7.83±0.26	3.32±0.21	10.25±0.45	3.07±0.25	9.49±0.67
DL-IVS	**DeepSmooth**	2.71±0.68	6.68±2.01	2.70±0.68	6.66±1.99	3.33±1.05	9.30±4.38	3.30±1.06	9.45±4.50	3.73±0.71	10.79±3.11	3.80±0.76	10.87±3.21
	Multi	2.80±0.06	8.99±0.28	5.74±0.12	18.09±0.68	3.12±0.09	9.28±0.35	5.78±0.12	19.70±0.41	3.81±0.31	13.46±1.36	6.56±0.23	23.10±0.84
	VAE-DNN	2.79±0.08	8.25±0.27	2.43±0.12	7.52±0.35	3.11±0.10	9.32±0.23	2.44±0.08	7.38±0.15	3.30±0.12	10.29±0.39	3.38±0.12	10.38±0.39
	Hexagon-Net	**1.74±0.03**	**4.96±0.14**	**0.85±0.10**	**1.97±0.33**	**2.42±0.03**	**5.57±0.08**	**0.92±0.06**	**2.00±0.44**	**2.01±0.04**	**5.15±0.14**	**0.86±0.03**	**1.96±0.27**

Table 2: Ablation Study on Two Main Modules. Bold indicates the best result.

		S&P500				NASDAQ100				STOXX50			
		Future Prediction		Mask Reconstruction		Future Prediction		Mask Reconstruction		Future Prediction		Mask Reconstruction	
	Model	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
HGA	**DFW**	2.39±0.04	6.58±0.15	1.43±0.06	4.26±0.22	2.48±0.04	7.07±0.15	1.62±0.04	4.61±0.23	2.59±0.04	7.34±0.28	1.93±0.03	4.86±0.67
	NW	2.30±0.04	6.43±0.15	1.34±0.05	3.94±0.29	2.54±0.04	7.23±0.16	1.57±0.06	4.41±0.25	2.63±0.05	7.60±0.28	1.74±0.05	4.85±0.73
	w/o HGA	2.45±0.03	6.77±0.15	1.51±0.06	5.15±0.26	2.56±0.04	7.39±0.15	1.61±0.04	4.50±0.25	2.69±0.06	7.91±0.27	1.99±0.05	4.92±0.75
CVT	***t*-view**	1.82±0.05	5.14±0.28	0.98±0.13	2.09±0.17	2.48±0.03	5.78±0.11	1.01±0.05	2.11±0.21	2.06±0.04	5.35±0.23	1.01±0.11	2.25±0.35
	τ-view	1.84±0.03	5.17±0.30	0.97±0.07	2.24±0.35	2.49±0.05	5.81±0.12	1.04±0.06	2.32±0.32	2.07±0.04	5.38±0.23	1.01±0.11	2.25±0.35
	***m*-view**	1.80±0.02	4.99±0.13	0.98±0.07	2.16±0.25	2.48±0.04	5.80±0.10	1.01±0.06	2.19±0.21	2.08±0.04	5.37±0.24	1.02±0.08	2.43±0.53
	w/o CVT	2.21±0.04	5.45±0.15	1.35±0.03	2.42±0.46	2.78±0.04	6.24±0.17	1.23±0.05	2.29±0.15	2.57±0.06	5.95±0.25	1.23±0.05	2.31±0.20
	Hexagon-Net	**1.74±0.03**	**4.96±0.14**	**0.85±0.10**	**1.97±0.33**	**2.42±0.03**	**5.57±0.08**	**0.92±0.06**	**2.00±0.44**	**2.01±0.04**	**5.15±0.14**	**0.86±0.03**	**1.96±0.27**

- **Informer** [33]: Introduces a sparse self-attention mechanism with good time complexity.
- **AST** [28]: Replaces the standard softmax with α-entmax for learning sparse attention maps and employs adversarial training via a discriminator to enhance prediction performance.

(3) DL Methods for IVS (DL-IVS):

- **DeepSmooth** [1]: Integrates neural networks into a standard arbitrage-free IVS modeling framework.
- **Multi** [32]: Introduces financial domain knowledge into a multi-agent architecture for learning a weighting mechanism tailored to IVS prediction.
- **VAE-DNN** [31]: Combines a VAE for feature extraction from IVS with an LSTM for feature prediction.

5.4 Experimental Results

We run each model 10 times and report the mean and standard deviation of the evaluation metrics for both tasks. As shown in Table 1, in the future prediction task, **Hexagon-Net** significantly outperforms all baseline models across both RMSE and MAPE. Compared to the second-best model, **Hexagon-Net** achieves RMSE reductions of 29.55%, 22.18%, and 37.96% on the S&P500, NASDAQ100, and STOXX50 datasets, respectively. The corresponding reductions in MAPE are 25.75%, 39.26%, and 49.76%. Furthermore, **Hexagon-Net** exhibits consistently lower standard deviations across runs, highlighting its robustness and improved generalization capability.

It is notable that **VAE-DNN** performs competitively, highlighting the benefit of explicitly modeling uncertainty in IVS prediction. This motivates us to examine (see Subsection 5.7) whether **Hexagon-Net** can also address model uncertainty, even though it was not specifically designed as a probabilistic model. However, **VAE-DNN** operates solely on the (τ, m) slice of the data and thus lacks the ability to exploit the full spatio-temporal structure of the IVS. Similarly, advanced generic time-series models, which rely purely on the temporal view (t) fail to capture dependencies across multiple dimensions. These empirical results support our design rationale for enabling cross-view information propagation and fusion in **Hexagon-Net**.

In the mask reconstruction task, **Hexagon-Net** performs even better, outperforming all baseline models by substantial margins across both evaluation metrics. Specifically, it reduces RMSE by 59.91%, 62.30%, and 69.61% on the S&P500, NASDAQ100, and STOXX50 datasets, respectively, compared to the second-best model. The improvements in MAPE are even more pronounced, with reductions of 68.73%, 72.90%, and 77.80% on the same datasets. These results highlight the ability of **Hexagon-Net** to effectively capture inter-IVS temporal dependencies while also leveraging the spatial structure across multiple well-aligned IVS views.

Table 3: Study on Two Alternative Setups. Bold indicates the best result.

	S&P500				NASDAQ100				STOXX50			
	Future Prediction		Mask Reconstruction		Future Prediction		Mask Reconstruction		Future Prediction		Mask Reconstruction	
Model	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Hexagon-Net (with MoCo)	1.75±0.02	**4.92±0.10**	0.86±0.07	2.00±0.20	2.44±0.02	5.66±0.12	0.93±0.03	2.03±0.32	2.00±0.04	5.25±0.38	**0.85±0.05**	1.99±0.24
Hexagon-Net (with StoL)	1.76±0.03	4.96±0.19	1.00±0.04	2.02±0.23	2.47±0.06	5.58±0.07	0.94±0.04	**1.99±0.44**	**1.99±0.03**	**5.12±0.17**	0.97±0.05	2.10±0.73
Hexagon-Net	**1.74±0.03**	4.96±0.14	**0.85±0.10**	**1.97±0.33**	**2.42±0.03**	**5.57±0.08**	**0.92±0.06**	2.00±0.44	2.01±0.04	5.15±0.14	0.86±0.03	**1.96±0.27**

5.5 Ablation Studies

Table 2 presents the results of the ablation studies, in which the **Heterogeneous Graph Grid Aligner** is abbreviated as **HGA**.

*5.5.1 Effectiveness of **HGA**.* We assess the contribution of **HGA** by comparing **Hexagon-Net** to three alternative configurations: (1) **DFW**, which replaces **HGA** with the DFW method [9]; (2) **NW**, which uses the Nadaraya–Watson estimator in place of **HGA**; (3) **w/o HGA**, which directly uses raw historical IVS data without any alignment. The results clearly indicate that **HGA** significantly outperforms the baselines in both tasks, confirming its effectiveness. Notably, the performance gains are more pronounced in the mask reconstruction task, suggesting that **HGA** is particularly effective when information is partially missing.

*5.5.2 Effectiveness of **CVT**.* We evaluate (CVT) by comparing it with four ablated variants: (1) ***t*-view**, which uses only the temporal view; (2) ***τ*-view**, which uses only the time-to-maturity view; (3) ***m*-view**, which uses only the log-moneyness view; (4) **w/o CVT**, which removes all structured views and uses only an MLP for feature extraction. Across all comparisons, **CVT** consistently achieves superior performance, underscoring the effectiveness of leveraging cross-view interactions.

5.6 Sensitive Analysis

As illustrated in Figures 3, 4, and 5, we conduct a sensitivity analysis of the RMSE with respect to three key hyperparameters: the number of hidden dimensions C, the number of layers L, and the static arbitrage-free loss weight β, across both the prediction and reconstruction tasks. The results reveal the following trends.

(1) When C is small, performance deteriorates due to insufficient model capacity. RMSE stabilizes at $C = 64$, beyond which further increases yield diminishing returns.

(2) Performance is significantly worse at $L = 1$, indicating the necessity of a deeper architecture. Models with $L = 2$ and $L = 3$ achieve similar results, and we cap the depth at $L = 3$ to prevent potential out-of-memory (OOM) issues.

(3) A small SAF loss weight (e.g., $\beta = 0.1$) improves performance by encouraging the absence of static arbitrage opportunities, but larger values of β degrade performance.

5.7 Alternative Setups

We compare **Hexagon-Net** against several alternative setups to assess its ability to address two core challenges. (1) The heterogeneous informativeness of different regions of the IVS, driven by the illiquidity of certain options, implies that a uniform modeling approach is inadequate. (2) The inherent stochasticity of financial markets introduces ambiguity and noise, leading to model uncertainty that must be effectively managed.

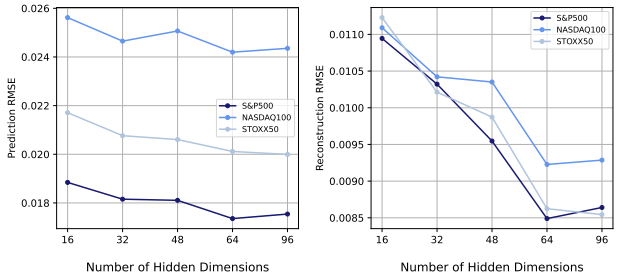

Figure 3: Impact of number of hidden dimensions C on prediction and reconstruction RMSE.

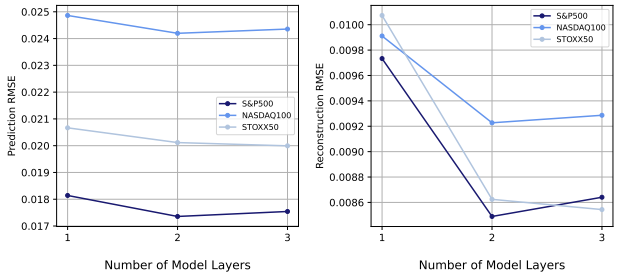

Figure 4: Impact of number of model layers L on prediction and reconstruction RMSE.

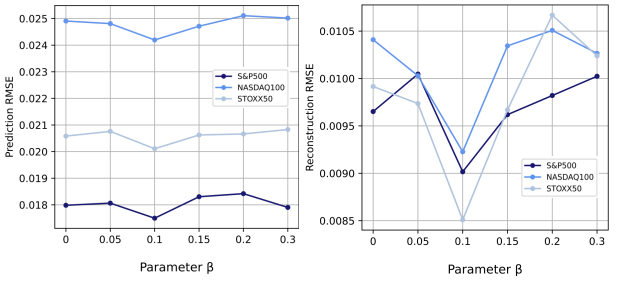

Figure 5: Impact of penalty factor β for static arbitrary free conditions on prediction and reconstruction RMSE.

5.7.1 Discriminative Feature Representations. To assess the ability of **Hexagon-Net** to learn discriminative feature representations, we integrate the Contrastive Learner **MoCo** [15] into the network

Table 4: Portfolio Performance (with Long and Short Positions). Bold indicates the best result.

		S&P500				NASDAQ100)				STOXX50			
		Long/Short 10%		Long/Short 20%		Long/Short 10%		Long/Short 20%		Long/Short 10%		Long/Short 20%	
	Model	AR	SR	AR	SR	AR	SR	AR	SR	AR	SR	AR	SR
MF	**SSVI**	7.98±0.51	1.61±0.08	3.84±0.35	0.92±0.07	7.66±1.33	1.48±0.25	2.66±0.21	0.69±0.04	4.41±1.07	0.71±0.20	1.44±1.56	0.49±0.37
TS	**Transformer**	8.24±0.86	1.64±0.18	1.88±0.76	0.56±0.16	8.39±0.64	1.57±0.12	2.81±0.55	0.71±0.11	3.94±0.95	0.64±0.20	1.70±0.79	0.48±0.16
	GAT	7.32±1.14	1.47±0.23	2.14±0.47	0.61±0.09	8.85±0.78	1.63±0.15	3.33±0.51	0.81±0.11	3.89±1.14	0.68±0.23	1.61±0.64	0.49±0.15
	DA-RNN	7.83±1.25	1.55±0.23	2.15±0.63	0.62±0.13	7.73±0.92	1.45±0.18	2.98±0.54	0.75±0.11	4.09±1.47	0.72±0.25	1.44±1.28	0.47±0.34
	DLinear	7.75±1.13	1.57±0.28	2.42±0.56	0.68±0.12	7.90±0.80	1.48±0.16	3.14±0.60	0.78±0.12	4.77±1.06	0.86±0.22	1.83±0.95	0.53±0.23
	NLinear	6.96±0.61	1.39±0.12	1.88±0.75	0.57±0.15	7.85±1.06	1.45±0.17	3.00±0.39	0.74±0.08	4.50±1.71	0.79±0.33	1.67±0.88	0.53±0.22
	Autoformer	6.33±0.91	1.34±0.17	1.47±0.37	0.48±0.08	7.64±1.09	1.46±0.19	2.33±0.49	0.62±0.09	3.58±1.19	0.69±0.25	1.12±0.42	0.42±0.11
	FEDformer	8.07±1.73	1.61±0.30	2.22±0.53	0.63±0.11	8.00±1.02	1.48±0.20	3.00±0.42	0.75±0.09	5.21±1.24	0.87±0.24	1.64±0.79	0.49±0.18
	Informer	6.92±1.37	1.36±0.33	2.07±0.55	0.60±0.12	8.50±0.76	1.57±0.14	2.99±0.38	0.75±0.08	5.26±1.31	0.95±0.26	1.35±0.82	0.44±0.20
	AST	8.26±0.86	1.64±0.19	1.90±0.74	0.56±0.17	8.42±0.63	1.58±0.12	2.82±0.53	0.71±0.10	3.95±0.94	0.64±0.19	1.71±0.78	0.48±0.17
DL-IVS	**DeepSmooth**	8.06±0.85	1.57±0.19	3.23±0.62	0.84±0.13	9.14±1.16	1.66±0.24	3.91±0.57	0.93±0.11	4.17±1.47	0.66±0.24	1.67±0.90	0.46±0.20
	Multi	8.56±1.10	1.68±0.21	2.53±0.54	0.69±0.12	8.95±0.64	1.62±0.14	3.42±0.51	0.83±0.11	4.66±1.63	0.78±0.32	2.38±0.67	0.62±0.19
	VAE-DNN	8.79±1.07	1.76±0.21	3.77±0.96	0.94±0.19	8.14±0.93	1.50±0.18	3.04±0.54	0.76±0.10	4.62±1.13	0.81±0.28	1.87±0.90	0.56±0.24
	Hexagon-Net	**9.07±0.78**	**1.85±0.25**	**4.02±0.72**	**0.96±0.16**	**9.77±0.50**	**1.69±0.09**	**4.27±0.27**	**0.95±0.10**	**5.32±0.96**	**0.96±0.23**	**2.47±0.34**	**0.63±0.11**

architecture and evaluate its performance relative to the original **Hexagon-Net**. The detailed implementation of **MoCo** is described in Appendix C.1.

As reported in Table 3, the performance of **Hexagon-Net** augmented with the **Contrastive Learner** closely matches that of the baseline **Hexagon-Net** without contrastive learning. This outcome indicates that **Hexagon-Net** inherently learns effective discriminative feature representations reflecting the illiquidity-driven heterogeneous informativeness present in options data.

5.7.2 Stochasticity. To evaluate the capability of **Hexagon-Net** in modeling uncertainty within IVS feature representations, we integrate a VAE-based Stochastic Learner (**StoL**) specifically designed to address such uncertainty. We then compare its performance against the baseline **Hexagon-Net**. The detailed architecture of **StoL** is provided in Appendix C.2.

As presented in Table 3, the performance of **Hexagon-Net** augmented with **StoL** closely aligns with that of the original **Hexagon-Net**. This indicates that **Hexagon-Net** inherently captures stochastic feature representations, a prevalent challenge in financial markets, thereby achieving robust predictive accuracy.

5.8 Robustness across Market Regimes

To further evaluate the robustness of **Hexagon-Net**, we partition the data based on high- and low-volatility regimes of the VIX index, using a rule-of-thumb threshold of 30%. We then assess the model's performance under these two market conditions. The mean and standard deviation of both RMSE and MAPE are found to be marginally higher in the high-volatility regime, but the differences are negligible. Detailed results are reported in Appendix D.

5.9 Options Portfolio Selection

To demonstrate the economic value of the proposed **Hexagon-Net**, we conduct a straightforward portfolio experiment. Specifically, we derive the expected options prices from the predicted implied volatilities according to the classical Black-Scholes model, and calculate the expected returns for each option. We rank all available options according to the predicted returns, and long (short) the top (bottom) 10% or 20% basket of options with equal weights. We rebalance the portfolio according to the updated rank of expected options returns. Given the relatively high liquidity of the stock index options used in this study, we assume a transaction cost of 50 basis points (i.e., 0.5%) per trade.

We calculate the annualized average return AR and Sharpe ratio SR of the options portfolios for each model in the testing period, where $r = [r_1, \ldots, r_T]$ is the portfolio return series. Each portfolio return r_t is calculated as $r_t = w_t^\top y_t$, where w_t is the portfolio weight vector for N_t options with $\sum_{i=1}^{N_t} w_{t,i} = 1$ and y_t is the realized options return vector.

It is intuitive that better performance on implied volatility prediction leads to better portfolio selection performance. As shown in Table 4, **Hexagon-Net** beats all competing models in both performance metrics AR and SR, suggesting that **Hexagon-Net** can effectively convert predictive precision into actual economic value.

6 Conclusion

The proposed **Hexagon-Net** is designed to effectively learn feature representations from misaligned imbalanced IVS historical data for both mask reconstruction and future prediction tasks. It not only offers a grid aligner based on a heterogeneous graph to aggregate and propagate information across different parts of the IVS that are characterized by liquidity-driven informativeness, but also explicitly captures and fuses cross-view spatio-temporal features of inner-IVS patterns and inter-IVS dynamics. To the best of our knowledge, this is the first work that addresses the IVS data misalignment issue and explicitly models IVS across views in a unified end-to-end framework to perform reconstruction and prediction tasks. **Hexagon-Net** significantly and robustly outperforms existing models and methods in terms of RMSE and MAPE across three major datasets of stock index options, and it also survives extensive ablation studies, sensitivity analyses, and alternative setups. Moreover, it is able to successfully convert its predictive precision into substantial economic value in options portfolio investment.

Acknowledgments

We thank the referees for insightful comments and helpful suggestions. This work was supported by National Natural Science Foundation of China (NSFC) 62272172.

References

[1] Damien Ackerer, Natasa Tagasovska, and Thibault Vatter. 2020. Deep smoothing of the implied volatility surface. In *Advances in Neural Information Processing Systems 33*, 11552–11563.
[2] Emre Aksan and Otmar Hilliges. 2019. STCN: stochastic temporal convolutional networks. *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.
[3] Caio Almeida, Jianqing Fan, Gustavo Freire, and Francesca Tang. 2023. Can a machine correct option pricing models? *Journal of Business and Economic Statistics*, 41, 3, 995–1009.
[4] Lorenzo Bergomi. 2015. *Stochastic Volatility Modeling*. CRC press.
[5] Fischer Black and Myron Scholes. 1973. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81, 3, 637–654.
[6] Vedant Choudhary, Sebastian Jaimungal, and Maxime Bergeron. 2024. Funvol: multi-asset implied volatility market simulator using functional principal components and neural SDEs. *Quantitative Finance*.
[7] Rama Cont and Jose da Fonseca. 2002. Dynamics of implied volatility surfaces. *Quantitative Finance*, 2, 1, 45–60.
[8] Rama Cont and Milena Vuletić. 2023. Simulation of arbitrage-free implied volatility surfaces. *Applied Mathematical Finance*.
[9] Bernard Dumas, Jeff Fleming, and Robert E Whaley. 1998. Implied volatility functions: empirical tests. *The Journal of Finance*, 53, 6, 2059–2106.
[10] Matthias R. Fengler, Wolfgang K. Härdle, and Enno Mammen. 2007. A semiparametric factor model for implied volatility surface dynamics. *Journal of Financial Econometrics*, 5, 2, 189–218.
[11] Jim Gatheral. 2004. *A parsimonious arbitrage-free implied volatility parameterisation with application to the valuation of volatility derivatives*. Presentation at Global Derivatives (2004). Global Derivatives and Risk Management, Madrid, May 2004.
[12] Jim Gatheral. 2011. *The Volatility Surface: A Practitioner's Guide*. John Wiley & Sons.
[13] Jim Gatheral and Antoine Jacquier. 2014. Arbitrage-free SVI volatility surfaces. *Quantitative Finance*, 14, 1, 59–71.
[14] Wolfgang Härdle. 1990. *Applied Nonparametric Regression*. Cambridge University Press.
[15] Kaiming He, Haoqi Fan, Yuxin Wu, et al. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9729–9738.
[16] D.P. Kingma and M. Welling. 2014. Auto-encoding variational Bayes. In *Proceedings of the 1st International Conference on Learning Representations (ICLR)*.
[17] Brian Ning, Sebastian Jaimungal, Xiaorong Zhang, and Maxime Bergeron. 2023. Arbitrage-free implied volatility surface generation with variational autoencoders. *SIAM Journal on Financial Mathematics*, 14, 4, 1004–1027.
[18] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.
[19] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 2nd International Conference on Machine Learning (ICLR)*. PMLR, 1278–1286.
[20] Michael Roper. 2010. Arbitrage free implied volatility surfaces. *Preprint, The University of Sydney*.
[21] Johannes Ruf and Weiguan Wang. 2022. Hedging with linear regressions and neural networks. *Journal of Business and Economic Statistics*, 40, 4, 1442–1454.
[22] Johannes Ruf and Weiguan Wang. 2020. Neural networks for option pricing and hedging: a literature review. *Journal of Computational Finance*, 24, 1–46, 1.
[23] Han Lin Shang and Fearghal Kearney. 2022. Dynamic functional time-series forecasts of foreign exchange implied volatility surfaces. *International Journal of Forecasting*, 38, 3, 1025–1049.
[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.
[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.
[26] Spyridon D Vrontos, John Galakis, and Ioannis D Vrontos. 2021. Implied volatility directional forecasting: a machine learning approach. *Quantitative Finance*, 21, 10, 1687–1706.
[27] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS)*, 22419–22430.
[28] Sifan Wu, Xi Xiao, Qianggang Ding, Peilin Zhao, Ying Wei, and Junzhou Huang. 2020. Adversarial sparse transformer for time series forecasting. In *Proceedings of 34th Conference on Neural Information Processing Systems (NeurIPS)*, 17105–17115.
[29] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, 11121–11128.
[30] Jin E Zhang and Yi Xiang. 2008. The implied volatility smirk. *Quantitative Finance*, 8, 3, 263–284.
[31] Wenyong Zhang, Lingfei Li, and Gongqiu Zhang. 2023. A two-step framework for arbitrage-free prediction of the implied volatility surface. *Quantitative Finance*, 23, 1, 21–34.
[32] Yu Zheng, Yongxin Yang, and Bowei Chen. 2021. Incorporating prior financial domain knowledge into neural networks for implied volatility surface prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3968–3975.
[33] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence.
[34] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 10th International Conference on Machine Learning (ICLR)*. PMLR, 27268–27286.

A Static Arbitrage-Free Conditions for IVS

We recall a theorem from [20], which provides conditions for the absence of static arbitrage.

Proposition 1. *Consider an implied volatility function $\sigma(m,\tau)$ and suppose the following conditions are satisfied.*

(1) ***(Positivity)*** *For every (m,τ) with $\tau > 0$, one has $\sigma(m,\tau) > 0$.*
(2) ***(Smoothness)*** *For every τ, the function $m \longrightarrow \sigma(m,\tau)$ is twice differentiable.*
(3) ***(Monotonicity in τ)*** *For every m, the function $\tau \longrightarrow \sigma(m,\tau)^2\tau$ is nondecreasing. Equivalently,*

$$l_{cal}(m,\tau) = \sigma(m,\tau) + 2\tau\partial_\tau\sigma(m,\tau) \geq 0.$$

(4) ***(Durrleman's Condition)*** *For every (m,τ), one has*

$$l_{but}(m,\tau) = \tau\sigma(m,\tau)\partial_{mm}\sigma(m,\tau) - \frac{1}{4}(\tau\sigma(m,\tau)\partial_m\sigma(m,\tau))^2 + \left(1 - \frac{m\partial_m\sigma(m,\tau)}{\sigma(m,\tau)}\right)^2 \geq 0.$$

(5) ***(Large Moneyness Behavior)*** *For every τ, $\sigma(m,\tau)^2$ is linear as $m \longrightarrow \pm\infty$.*
(6) ***(Value at Maturity)*** *For every m, one has $\sigma(m,0) = 0$.*

Then the resulting implied volatility function is free of static arbitrage.

Note that in the setup of the previous proposition, Condition (5) is equivalent to the second-order derivative of $\sigma(m,\tau)^2$ going to zero when $m \longrightarrow \pm\infty$. Here we have

$$\partial_{mm}\sigma^2(m,\tau) = 2\sigma(m,\tau)\partial_{mm}\sigma(m,\tau) + 2(\partial_m\sigma(m,\tau))^2.$$

B Grids Design For Static Arbitrage-Free

To align the IVS and compute the static arbitrage-free loss, we design two custom grids following [31]. In particular, we construct denser grids for smaller time-to-maturity values τ to better capture near-term structure.

The set of τ values is given by the following $A = 25$ points:

$$\overline{\mathcal{G}}^{\tau} = [0, 0.05)_5 \oplus [0.08, 0.13)_5 \oplus [0.13, 0.20)_5 \oplus [0.20, 0.40)_5 \oplus [0.40, 1.36]_5,$$

where $\oplus$ denotes array concatenation, and $[a, b)_k$ indicates an arithmetic sequence of k evenly spaced points from a up to (but not including) b.

Similarly, the set of m values (log-moneyness) consists of the following $B = 40$ points:

$$\overline{\mathcal{G}}^{m} = [-0.23, 0)_{13} \oplus [0, 0.08)_{13} \oplus [0.08, 0.4]_{14},$$

with denser resolution around the at-the-money (ATM) region to better capture local IVS behavior. To construct the two grids, we now extend the m values to increase coverage in the wings:

$$\overline{\mathcal{G}}^{m}_{34} = \overline{\mathcal{G}}^{m} \cup \left\{ m^3 : m \in [-(-2m_{\min})^{1/3}, (2m_{\max})^{1/3}]_{40} \right\},$$
$$\overline{\mathcal{G}}^{m}_{5} = \overline{\mathcal{G}}^{m} \cup \left([6m_{\min}, 4m_{\min}]_{20} \oplus [4m_{\max}, 6m_{\max}]_{20} \right),$$

where $m_{\min}$ and $m_{\max}$ are the minimum and maximum values of $\overline{\mathcal{G}}^{m}$, respectively.

The final aligned grids are now constructed as the Cartesian product of these sets:

$$\mathcal{G}^{C34} = \overline{\mathcal{G}}^{\tau} \times \overline{\mathcal{G}}^{m}_{34}, \qquad \mathcal{G}^{C5} = \overline{\mathcal{G}}^{\tau} \times \overline{\mathcal{G}}^{m}_{5}.$$

C Alternative Setups

C.1 Contrastive Learner

To improve the framework's ability to learn discriminative features and capture intrinsic relationships between grid points we incorporate two cross-view attention mechanisms. One model's parameters are updated via standard back-propagation, while the other employs momentum-based updates (e.g., **MoCo**), progressively inheriting parameters from the former. To facilitate discriminative feature learning across grid points, representations corresponding to identical locations are treated as positive pairs, whereas those from distinct locations serve as negative pairs.

Formally, let the grid representations learned by the two models be denoted as $q, k \in \mathbb{R}^{T \times P \times C}$ (Recall that $P = A \times B$ is the number of grid points and C denotes the hidden layer dimensionality.). The grid contrastive learning loss is then defined as

$$\mathcal{L}_C = -\frac{1}{T \times P} \sum_{t=1}^{T} \sum_{i=1}^{P} \ln \frac{\exp(\mathrm{sim}(q_{t,i}, k_{t,i})/\tau)}{\sum_{j \neq i} \exp(\mathrm{sim}(q_{t,i}, k_{t,j})/\tau)}.$$

More details can be found in [15].

Additionally, we explore an alternative contrastive learning scheme involving only near-expiry and far-into-future options, as well as at-the-money and deep out-of-money options. The results are qualitatively similar.

C.2 Stochastic Learner

Given the high stochasticity inherent in financial markets, rather than modeling IVS distributions directly, we assume that these distributions are governed by a set of latent random variables following conditional Gaussian priors. We learn the distributions of these latent variables from observed IVS data via an inference model, while concurrently training a generative model that reconstructs variational IVS from the latent space. The **VAE** [16, 19] provides a standard framework for modeling variation in non-sequential data through latent random variables. However, its conventional distributional assumptions are not readily applicable to IVS prediction, due to the presence of temporal interdependence in historical IVS sequences.

To address this, we adopt the approach of [2], modeling temporal dependencies in IVS using a dependency graph (see Figure 6) embedded within both the generative and inference models. Specifically, we reshape the hidden representations $H^1, \ldots, H^l \in \mathbb{R}^{T \times P \times C}$ from all layers into tensors of shape $\mathbb{R}^{T \times \mathcal{D}}$, where $\mathcal{D} = P \times C$. Because higher-layer representations exhibit broader receptive fields, the hierarchical generative and inference architectures implicitly capture temporal dependencies across time steps.

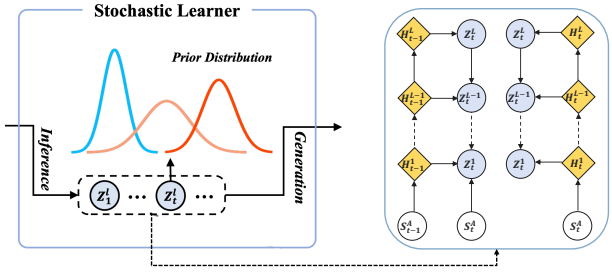

Figure 6: Structure of Stochastic Learner

C.2.1 Generation Model. As depicted in the left panel of Figure 6, we employ a generative model to produce IVS from latent variables, formulated as $p_\theta(I^{\mathrm{al}}_t \mid \mathcal{Z}_t) = f_{\mathrm{gen}}(\mathcal{Z}_t)$, where f_{gen} is parameterized by an MLP. The temporal interdependence among hierarchical latent variables $\mathcal{Z}_t = \{Z^1_t, \ldots, Z^L_t\}$ and historical IVS $I^{\mathrm{al}}_{1:t-1}$ is modeled by the following structured prior:

$$p_\theta(\mathcal{Z}_t \mid I^{\mathrm{al}}_{1:t-1}) = p_\theta(Z^L_t \mid H^L_{t-1}) \prod_{l=1}^{L-1} p_\theta(Z^l_t \mid Z^{l+1}_t, H^l_{t-1}),$$

where each conditional distribution is modeled as a Gaussian with diagonal covariance, following [2]:

$$p_\theta(Z^L_t \mid H^L_{t-1}) = \phi(Z^L_t \mid \mu^L_{t,p}, \sigma^L_{t,p}), \quad p_\theta(Z^l_t \mid Z^{l+1}_t, H^l_{t-1}) = \phi(Z^l_t \mid \mu^l_{t,p}, \sigma^l_{t,p}),$$

where ϕ denotes the density of the standard normal distribution. The parameters μ and σ are generated by neural networks:

$$[\mu^L_{t,p}, \sigma^L_{t,p}] = f^L_p(H^L_{t-1}), \quad [\mu^l_{t,p}, \sigma^l_{t,p}] = f^l_p(Z^{l+1}_t, H^l_{t-1}).$$

C.2.2 Inference Model. Analogously, the inference model approximates the posterior distribution of the latent variables $\mathcal{Z}_t$, as illustrated in the right panel of Figure 6:

$$q_\phi(\mathcal{Z}_t \mid I^{\mathrm{al}}_{1:t}) = q_\phi(Z^L_t \mid H^L_t) \prod_{l=1}^{L-1} q_\phi(Z^l_t \mid Z^{l+1}_t, H^l_t),$$

where each conditional distribution is Gaussian with diagonal covariance:

$$q_\phi(Z^L_t \mid H^L_t) = \phi(Z^L_t \mid \mu^L_{t,q}, \sigma^L_{t,q}), \quad q_\phi(Z^l_t \mid Z^{l+1}_t, H^l_t) = \phi(Z^l_t \mid \mu^l_{t,q}, \sigma^l_{t,q}).$$

Again, the mean and standard deviation are parameterized by neural networks:

$$[\mu^L_{t,q}, \sigma^L_{t,q}] = f^L_q(H^L_t), \quad [\mu^l_{t,q}, \sigma^l_{t,q}] = f^l_q(Z^{l+1}_t, H^l_t).$$

Table 5: IVS Prediction Performance (%) in Low-Volatility Regime. Bold indicates the best result.

		S&P500				NASDAQ100				STOXX50			
		Future-Prediction		Mask-Reconstruction		Future-Prediction		Mask-Reconstruction		Future-Prediction		Mask-Reconstruction	
	Model	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
MF	**SSVI**	3.14±0.27	9.26±0.83	3.12±0.22	9.20±0.76	3.22±0.16	9.16±0.92	3.20±0.25	9.22±0.87	3.78±0.36	12.42±1.87	3.82±0.45	12.55±2.10
TS	**Transformer**	3.84±0.11	10.97±0.55	3.12±0.23	8.66±0.45	3.85±0.42	11.26±1.35	3.18±0.49	8.87±1.20	3.70±0.28	11.67±1.05	3.25±0.29	9.85±0.92
	GAT	3.04±0.75	8.74±0.58	2.21±0.06	6.93±0.13	3.26±0.08	9.52±0.37	2.50±0.08	7.87±0.32	3.45±0.32	10.59±0.83	3.11±0.31	9.65±0.79
	DA-RNN	2.82±0.08	8.60±0.41	2.25±0.06	6.87±0.18	3.28±0.12	9.59±0.30	2.51±0.08	7.86±0.25	3.43±0.10	10.70±0.33	3.06±0.13	9.55±0.44
	DLinear	3.18±0.35	10.09±0.76	2.79±0.52	8.21±1.31	3.12±0.04	9.55±0.16	2.42±0.08	7.39±0.22	3.33±0.05	10.33±0.30	3.05±0.16	9.14±0.38
	NLinear	3.15±0.15	10.14±0.55	2.71±0.25	8.22±0.80	3.72±0.17	10.92±0.46	3.03±0.25	8.51±0.40	3.73±0.26	11.46±0.88	3.53±0.39	10.57±1.47
	Autoformer	5.00±0.27	19.35±1.45	2.47±0.13	7.64±0.35	5.03±0.13	17.30±0.51	8.30±0.30	7.74±0.14	4.76±0.10	16.86±0.43	3.09±0.24	9.64±0.59
	FEDformer	2.83±0.05	8.94±0.26	2.27±0.05	7.13±0.22	3.26±0.10	9.94±0.34	2.57±0.08	7.93±0.36	3.42±0.09	10.62±0.38	3.00±0.12	9.33±0.31
	Informer	2.81±0.06	9.16±0.36	2.24±0.08	7.10±0.28	3.30±0.07	9.95±0.30	2.57±0.09	7.91±0.36	3.24±0.06	10.31±0.28	2.83±0.08	8.83±0.23
	AST	2.46±0.29	7.64±1.65	2.12±0.27	6.30±1.31	3.12±0.04	9.35±0.15	2.46±0.04	7.83±0.22	3.32±0.18	10.25±0.38	3.07±0.21	9.49±0.59
DL-IVS	**DeepSmooth**	2.71±0.58	6.68±1.78	2.70±0.59	6.66±1.76	3.32±0.92	9.30±3.89	3.30±0.93	9.45±4.00	3.73±0.62	10.79±2.76	3.80±0.66	10.87±2.85
	Multi	2.80±0.04	8.98±0.23	5.74±0.09	18.09±0.59	3.11±0.07	9.28±0.29	5.78±0.09	19.70±0.35	3.81±0.26	13.46±1.19	6.56±0.19	23.10±0.73
	VAEDNN	2.79±0.05	8.25±0.22	2.43±0.09	7.52±0.30	3.10±0.07	9.32±0.19	2.44±0.06	7.38±0.12	3.30±0.09	10.29±0.33	3.38±0.09	10.38±0.33
	Hexagon-Net	**1.74±0.03**	**4.96±0.15**	**0.84±0.08**	**1.95±0.29**	**2.41±0.04**	**5.56±0.06**	**0.92±0.05**	**2.00±0.38**	**2.00±0.03**	**5.13±0.12**	**0.86±0.03**	**1.95±0.25**

Table 6: IVS Prediction Performance (%) in High-Volatility Regime. Bold indicates the best result.

		S&P500				NASDAQ100				STOXX50			
		Future-Prediction		Mask-Reconstruction		Future-Prediction		Mask-Reconstruction		Future-Prediction		Mask-Reconstruction	
	Model	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
MF	**SSVI**	3.16±0.28	9.29±0.93	3.17±0.29	9.23±0.95	3.27±0.24	9.19±1.01	3.25±0.24	9.24±1.02	3.82±0.41	12.47±2.17	3.88±0.45	12.62±1.89
TS	**Transformer**	3.88±0.11	11.02±0.59	3.18±0.24	8.69±0.49	3.90±0.45	11.28±1.47	3.23±0.52	8.91±1.32	3.75±0.30	11.72±1.15	3.30±0.30	9.86±1.00
	GAT	3.06±0.81	8.77±0.61	2.25±0.05	6.94±0.14	3.31±0.07	9.54±0.40	2.54±0.07	7.89±0.35	3.49±0.34	10.62±0.91	3.16±0.33	9.67±0.86
	DA-RNN	2.87±0.08	8.62±0.45	2.29±0.05	6.88±0.19	3.34±0.12	9.62±0.33	2.56±0.07	7.88±0.27	3.48±0.09	10.74±0.36	3.11±0.13	9.59±0.48
	DLinear	3.25±0.38	10.13±0.83	2.84±0.57	8.25±1.43	3.17±0.03	9.58±0.17	2.47±0.07	7.42±0.23	3.38±0.04	10.35±0.33	3.10±0.16	9.14±0.41
	NLinear	3.18±0.15	10.17±0.60	2.74±0.26	8.23±0.86	3.76±0.17	10.95±0.51	3.05±0.26	8.53±0.42	3.76±0.27	11.47±0.96	3.55±0.41	10.58±1.60
	Autoformer	5.04±0.29	19.37±1.59	2.50±0.13	7.65±0.38	5.06±0.13	17.32±0.57	8.32±0.33	7.75±0.14	4.80±0.10	16.88±0.48	3.10±0.26	9.66±0.63
	FEDformer	2.85±0.05	8.95±0.28	2.30±0.04	7.14±0.24	3.32±0.10	9.96±0.37	2.60±0.07	7.94±0.39	3.45±0.09	10.64±0.42	3.03±0.12	9.36±0.34
	Informer	2.86±0.06	9.19±0.40	2.26±0.07	7.15±0.30	3.34±0.06	9.97±0.33	2.60±0.09	7.94±0.39	3.27±0.05	10.35±0.31	2.86±0.07	8.87±0.24
	AST	2.50±0.32	7.66±1.81	2.15±0.29	6.32±1.43	3.16±0.03	9.39±0.16	2.50±0.03	7.86±0.23	3.33±0.18	10.28±0.42	3.10±0.26	9.56±0.64
DL-IVS	**DeepSmooth**	2.74±0.65	6.69±1.97	2.71±0.65	6.68±1.95	3.35±1.02	9.34±4.33	3.33±1.03	9.49±4.45	3.74±0.68	10.84±3.06	3.82±0.73	10.89±3.16
	Multi	2.83±0.03	9.01±0.25	5.77±0.09	18.12±0.64	3.14±0.06	9.30±0.32	5.79±0.09	19.73±0.38	3.84±0.28	13.48±1.32	6.57±0.20	23.12±0.80
	VAEDNN	2.80±0.05	8.29±0.24	2.46±0.09	7.54±0.32	3.14±0.07	9.34±0.20	2.46±0.05	7.40±0.12	3.32±0.09	10.30±0.36	3.40±0.09	10.42±0.36
	Hexagon-Net	**1.76±0.02**	**4.98±0.23**	**0.90±0.13**	**2.03±0.41**	**2.43±0.04**	**5.59±0.05**	**0.94±0.08**	**2.03±0.46**	**2.04±0.03**	**5.17±0.12**	**0.89±0.03**	**1.99±0.25**

C.2.3 Probability Learning. Following [2], the variational lower bound of the log-likelihood at time t equals

$$\mathcal{L}_{\text{rec}}(\theta,\phi;t) + \mathcal{L}_{\text{KL}}(\theta,\phi;t) = -\mathbb{E}_{q_\phi(\mathcal{Z}_t \mid I_t^{\text{al}})}\left[\log p_\theta(I_t^{\text{al}} \mid \mathcal{Z}_t)\right] + \text{KL}\big(q_\phi(\mathcal{Z}_t \mid I_{1:t}^{\text{al}}) \,\|\, p_\theta(\mathcal{Z}_t \mid I_{1:t-1}^{\text{al}})\big),$$

where the first term is the reconstruction loss, and the second is the Kullback–Leibler divergence between the posterior and the prior.

D Robustness across Market Regimes

Tables 5 and 6 compare the model performance in high-volatility and low-volatility regimes of the VIX index using a rule-of-thumb threshold of 30%.