Time-Series Volatility Clustering and Adaptive Trading Signals with Graph Neural Networks

Context:

The global foreign exchange (FX) market involves non-linear, interdependent dynamics of 28 major currency pairs. Volatility clustering, a well-documented phenomenon in financial time series, poses a unique opportunity and challenge. Volatility tends to persist or "cluster," meaning periods of high volatility are followed by high volatility and periods of low volatility by low volatility. This behavior can be influenced by macroeconomic news, liquidity changes, and cross-asset flows.

The task is to design an **adaptive trading strategy** that:

1. Identifies and models the clustering behavior of volatility in the FX market using **time-series features**.
2. Dynamically incorporates interdependencies between currency pairs through **Graph Neural Networks (GNNs)**.
3. Combines these insights with **Principal Component Analysis (PCA)** to reduce dimensionality and focus on key market-driving features.
4. Outputs robust, market-adaptive trading signals in real-time.

The problem requires solving **stochastic differential equations (SDEs)** for time-series volatility, modeling clustering behavior mathematically, and adapting GNN weights dynamically in response to evolving market conditions.

Mathematical Formulation

1. **Volatility as a Stochastic Process**

The volatility $\sigma_i(t)$ of a currency pair C_i is modeled as a stochastic process with clustering behavior:

$$d\sigma_i(t) = \alpha_i(\mu_i - \sigma_i(t))dt + \beta_i\sqrt{\sigma_i(t)}\,dW_i(t),$$

where:

- α_i: Mean-reversion speed for $\sigma_i(t)$.
- μ_i: Long-term average volatility.
- β_i: Volatility of volatility (vol-of-vol).
- $dW_i(t)$: Wiener process representing random shocks.

2. **Volatility Clustering Metric**

To measure clustering, use the ratio of rolling mean to rolling standard deviation of volatility:

$$\text{Cluster}_i(t) = \frac{\text{SMA}(\sigma_i(t), L)}{\text{StdDev}(\sigma_i(t), L)},$$

where L is the window size.

3. **Time-Series Features**

Extract additional time-series features, such as:

- **Autocorrelation** at lag k:

$$\text{Auto}_i(k) = \frac{\sum_{t=1}^{T-k}(\sigma_i(t) - \bar{\sigma}_i)(\sigma_i(t+k) - \bar{\sigma}_i)}{\sum_{t=1}^{T}(\sigma_i(t) - \bar{\sigma}_i)^2}.$$

- **Lagged volatility**:

$$\text{Lag}_i(k) = \sigma_i(t-k).$$

4. **PCA Dimensionality Reduction**

The interdependencies between volatility features are modeled using a kernel-based PCA. The kernel matrix K is computed as:

$$K_{ij} = \exp\left(-\frac{\|\mathbf{f}_i - \mathbf{f}_j\|^2}{2\sigma_k^2}\right),$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ are feature vectors for currency pairs C_i and C_j, and σ_k controls sensitivity.

Perform eigen decomposition:

$$K = \sum_{k=1}^{N} \lambda_k \mathbf{v}_k \mathbf{v}_k^\top,$$

where:

- λ_k: Eigenvalues representing the variance explained by each component.
- $\mathbf{v}_k$: Corresponding eigenvectors.

Retain the top m components such that:

$$\frac{\sum_{k=1}^{m} \lambda_k}{\sum_{k=1}^{N} \lambda_k} \geq 90\%.$$

5. **Graph Neural Networks**

The GNN models interdependencies between PCA-reduced features. The adjacency matrix A is constructed as:

$$A_{ij} = \exp\left(-\frac{\|\mathbf{R}_i - \mathbf{R}_j\|^2}{2\sigma_A^2}\right),$$

where $\mathbf{R}_i$ is the PCA-reduced feature vector for C_i.

The GNN feature propagation follows:

$$\mathbf{H}_i^{(l+1)} = \mathrm{ReLU}\left(\sum_{j=1}^{N} A_{ij} \mathbf{H}_j^{(l)} W^{(l)}\right),$$

where:

- $\mathbf{H}_i^{(l)}$: Feature vector for C_i at layer l.
- $W^{(l)}$: Weight matrix at layer l.
- ReLU: Rectified Linear Unit activation function.

6. **Signal Generation**

The GNN outputs probabilities for actions $\mathrm{Buy, Sell, Hold}$:

$$p_i^{(k)} = \frac{\exp(\mathbf{H}_i^{(L)}[k])}{\sum_{k=1}^{3} \exp(\mathbf{H}_i^{(L)}[k])}, \quad k \in \{\mathrm{Buy, Sell, Hold}\}.$$

The trading signal for C_i is:

$$\mathrm{Signal}_i = p_i^{(\mathrm{Buy})} - p_i^{(\mathrm{Sell})}.$$

7. **Dynamic Thresholds**

Dynamic thresholds for trading are defined based on:

- Mean and standard deviation of signals across all pairs:

$$\Theta(t) = \mu(\mathrm{Signal}) + \delta \cdot \sigma(\mathrm{Signal}),$$

where δ is a risk-adjustment parameter.

8. **Optimization Objective**

Maximize the portfolio's Sharpe ratio:

$$\mathrm{Sharpe\ Ratio} = \frac{\mathrm{E}[R]}{\mathrm{StdDev}[R]},$$

where R is the portfolio return.

Minimize drawdown DD:

$$\mathrm{DD} = \frac{\max(E) - \min(E)}{\max(E)},$$

where E is the equity curve.

Challenges and Complexity

1. **Multi-Dimensional Dependencies**:
   - Modeling cross-pair volatility interactions with PCA and GNN requires accurate kernel definitions and adjacency matrices.
2. **Stochastic Dynamics**:
   - Solving the SDEs for volatility clustering involves numerical approximations and stability analysis.
3. **Real-Time Adaptation**:
   - Thresholds and GNN weights must be updated dynamically to reflect changing market conditions.
4. **Computational Constraints**:
   - Eigen decomposition for PCA and GNN propagation are computationally intensive, especially with 28 pairs and multi-layer networks.