Deep Learning for VWAP Execution

Rémi Genet   Fabrice Riva

DRM, Université Paris Dauphine

Deep Learning for VWAP Execution in Crypto Markets: Beyond the Volume Curve

VWAP is one of the most common strategic orders for institutional investors, as it presents several advantages:

- As it splits the execution over a given period, it reduces the instantaneous market impact
- With its pre-defined duration, investors know exactly when the order will complete
- VWAP serves as an unbiased benchmark that investors can use to evaluate their brokers' performance

The Volume Weighted Average Price (VWAP) is calculated by weighting each transaction price by its corresponding trading volume. For a given time period, it is computed as:

$$\text{VWAP} = \frac{\sum_{i=1}^{n} P_i V_i}{\sum_{i=1}^{n} V_i} \tag{1}$$

where P_i represents the price of the i-th transaction and V_i its corresponding volume. This formulation naturally gives more weight to prices at which larger volumes were traded. VWAP serves as an ideal benchmark for trading performance because it represents a zero-sum game: the total value gained by traders who outperformed VWAP exactly equals the total value lost by those who underperformed it.

Current literature primarily focuses on a key assumption: perfectly predicting the volume curve over the execution period would achieve optimal VWAP execution.

However, this approach largely overlooks the critical cross-interaction between volume and volatility.

While calibrating a model to predict the volume curve (whether static or dynamic) is relatively straightforward, developing one that leverages both interactions to optimize execution strategy proves significantly more challenging with conventional tools.

This is where deep learning offers a compelling solution:

- Deep learning's foundation rests on **automatic differentiation through back-propagation**
- At its core, deep learning consists of optimizing functions with large parameter spaces using sophisticated optimizers
- Crucially, both absolute and quadratic deviations of VWAP are **differentiable functions**

We formulate VWAP execution as an optimization problem by minimizing one of those three loss functions:

1. Quadratic VWAP Loss:

$$L_Q = \mathbb{E}\left[\left(\frac{\text{VWAP}_{\text{achieved}}}{\text{VWAP}_{\text{market}}} - 1\right)^2\right] \tag{2}$$

2. Absolute VWAP Loss:

$$L_A = \mathbb{E}\left[\left|\frac{\text{VWAP}_{\text{achieved}}}{\text{VWAP}_{\text{market}}} - 1\right|\right] \tag{3}$$

3. Volume Curve Loss:

$$L_V = \mathbb{E}\left[\sum_{i=1}^{n}\left(\frac{v_i}{\sum_{j=1}^{n} v_j} - \frac{V_i}{\sum_{j=1}^{n} V_j}\right)^2\right] \tag{4}$$

We implemented a static linear model within a neural network framework, utilizing the architecture described in *A Temporal Linear Network for Time Series Forecasting, Genet and Inzirillo*. The softmax activation function determines optimal quantity allocation across timesteps. The implementation proceeds as follows:

1. TLN Output Generation: Let $v_{[t,t+h]} = \text{TLN}(x_t)$ represent the raw output of the Temporal Linear Network for the horizon $[t, t+h]$, where x_t denotes the input features at time t.
2. Softmax Transformation: Trading quantities $q_{[t,t+h]}$ are derived by applying the softmax function to normalize the TLN output:

$$q_i = \frac{e^{v_i}}{\sum_{j=t}^{t+h} e^{v_j}} \quad \text{for } i \in [t, t+h]$$

3. Optimization Strategy: The model minimizes one of the previously defined loss functions using the Adam optimizer.

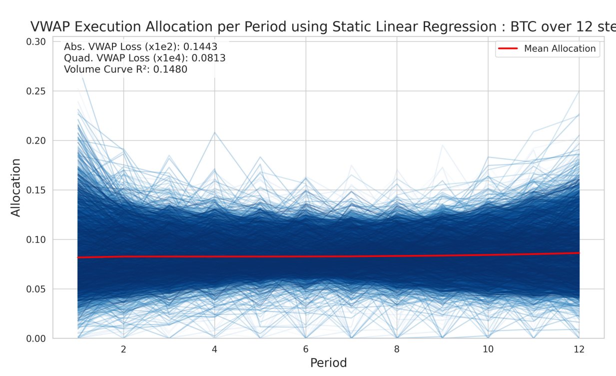
We evaluated our model against the following benchmarks:

- Naive: Equal-weighted allocation across all timesteps
- Fixed Volume Curve: Pre-optimized curve based on training data, operating without real-time market information
- Static Linear: Standard linear regression model for volume curve prediction
- Dynamic Linear: Adaptive linear regression following the methodology of *J. Bialkowski, S. Darolles, and G. Le Fol, "Improving VWAP strategies: A dynamic volume approach"*

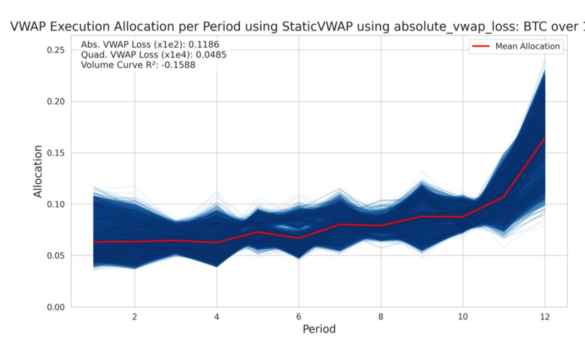
Table 1. VWAP Optimization Results

Model Type	Asset	Optimization	Abs. VWAP Loss (10^{-2})	Quad. VWAP Loss (10^{-4})	R² Vol. Curve
Naive	BTC	N/A	0.158743	0.087808	0.000000
StaticVWAP	BTC	Absolute	**0.119742**	0.050175	-0.133930
StaticVWAP	BTC	Quadratic	0.120810	**0.047403**	-0.361577
StaticVWAP	BTC	Volume	0.149938	0.084989	0.134904
Dynamic Linear	BTC	N/A	0.134621	0.076604	**0.174309**
Static Linear	BTC	N/A	0.144357	0.081390	0.146436
Fixed Volume Curve	BTC	Absolute	0.129670	0.054213	-0.273680
Fixed Volume Curve	BTC	Quadratic	0.127530	0.049655	-0.467000
Fixed Volume Curve	BTC	Volume	0.160575	0.089770	-0.003349
Naive	ETH	N/A	0.177758	0.116196	0.000000
StaticVWAP	ETH	Absolute	**0.138627**	0.076506	-0.146691
StaticVWAP	ETH	Quadratic	0.139999	0.073385	-0.297154
StaticVWAP	ETH	Volume	0.170102	0.122055	0.109135
Dynamic Linear	ETH	N/A	0.154125	0.140239	-6.611908
Static Linear	ETH	N/A	0.165134	0.121097	-0.414701
Fixed Volume Curve	ETH	Absolute	0.148407	0.076206	-0.269217
Fixed Volume Curve	ETH	Quadratic	0.147058	**0.072852**	-0.348835
Fixed Volume Curve	ETH	Volume	0.174744	0.112633	-0.006033
Naive	XRP	N/A	0.223855	0.275925	0.000000
StaticVWAP	XRP	Absolute	**0.182436**	0.187474	-0.268477
StaticVWAP	XRP	Quadratic	0.188880	0.158377	-0.800538
StaticVWAP	XRP	Volume	0.218816	0.291965	0.053552
Dynamic Linear	XRP	N/A	0.200789	0.274814	**0.095260**
Static Linear	XRP	N/A	0.211910	0.293099	0.056375
Fixed Volume Curve	XRP	Absolute	0.186341	0.178947	-0.347910
Fixed Volume Curve	XRP	Quadratic	0.189142	**0.152440**	-0.794792
Fixed Volume Curve	XRP	Volume	0.224941	0.276570	-0.001512

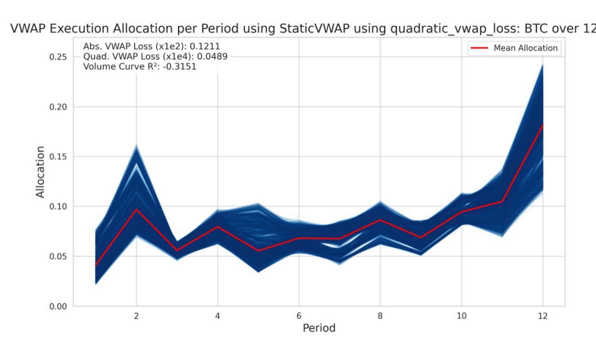
To illuminate the mechanisms behind this enhanced efficiency, we compare the allocation curves produced by each model:



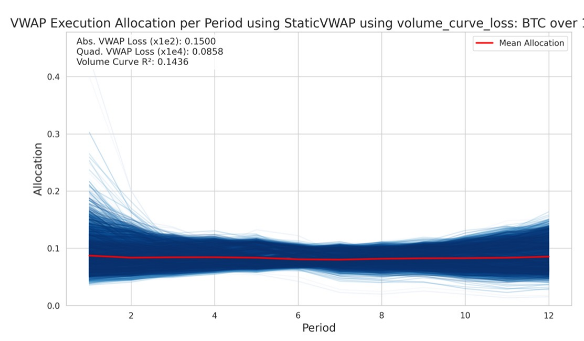
(a) Optimized using Linear Regression



(b) Optimized for Absolute deviation



(c) Optimized for Quadratic deviation



(d) Optimized for Volume Curve

The allocation curves generated by our deep learning approach exhibit notable deviations from traditional volume curves, particularly showing increased allocation weights towards the order's end. This strategic overweighting serves as an effective hedge against potential late-period spikes in both volatility and volume during order execution.

Recurrent Networks for Dynamic VWAP Execution

Having demonstrated the effectiveness of deep learning for VWAP execution, we next sought to enhance our approach by incorporating dynamic capabilities.

While our initial TLN model generates allocation curves at the order's inception, it lacks the ability to adapt during execution.

Moreover, traditional methods for introducing dynamic behavior prove incompatible with our approach due to fundamental architectural differences.

To achieve true dynamic adaptation, we leverage recurrent neural networks (RNNs):

- RNNs are deep learning architectures designed to process sequential data iteratively
- They employ sophisticated memory mechanisms to incorporate historical information
- This enables predictions that simultaneously account for both historical context and current market conditions

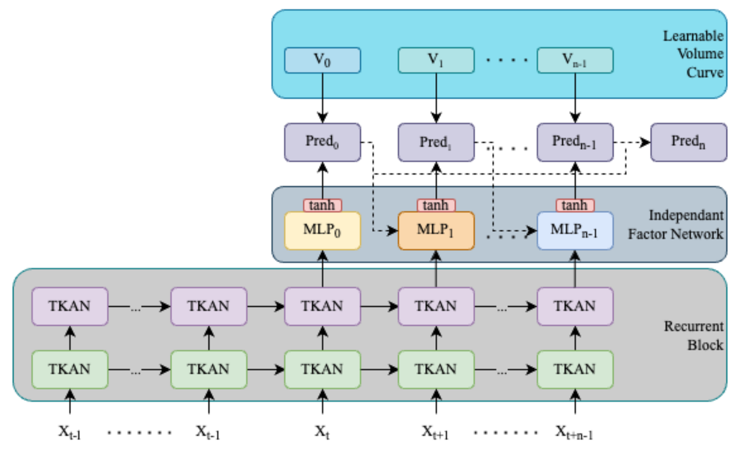


Figure 2. Proposed dynamic VWAP architecture

The architecture consists of four key components:

- A learnable base volume curve with trainable parameters that establish the default execution profile
- A recurrent neural network block implementing the *Temporal Kolmogorov-Arnold Networks* architecture (Genet and Inzirillo)
- Time-specific multilayer perceptrons that integrate both recurrent block outputs and previous predictions at each timestep
- A transformation layer that applies hyperbolic tangent activation and adds unity, mapping outputs to [0,2] to serve as multiplicative adjustments to the base curve

Table 2. VWAP Optimization Results for Selected Models and Cryptocurrencies

Model Type	Asset	Optimization	Abs. VWAP Loss (10^{-2})	Quad. VWAP Loss (10^{-4})	R² Vol. Curve
Naive	BTC	N/A	0.15874311	0.08780818	0.00000000
StaticVWAP	BTC	Absolute	0.11927138	0.04950001	-0.15073566
StaticVWAP	BTC	Quadratic	0.12114690	0.04740549	-0.36392904
StaticVWAP	BTC	Volume Curve	0.14936921	0.08490292	0.13865099
DynamicVWAP	BTC	Absolute	**0.10525936**	**0.04129920**	-0.25283452
DynamicVWAP	BTC	Quadratic	0.11716457	0.04235737	-0.43806944
DynamicVWAP	BTC	Volume Curve	0.14388469	0.10851773	**0.53748146**
Naive	ETH	N/A	0.17775816	0.11619556	0.00000000
StaticVWAP	ETH	Absolute	0.13896361	0.07704772	-0.13596994
StaticVWAP	ETH	Quadratic	0.13992102	0.07361006	-0.28744147
StaticVWAP	ETH	Volume Curve	0.16925061	0.12173233	0.11599889
DynamicVWAP	ETH	Absolute	**0.12090351**	**0.05820394**	-0.36164006
DynamicVWAP	ETH	Quadratic	0.13420704	0.06155421	-0.34871240
DynamicVWAP	ETH	Volume Curve	0.15213849	0.14477868	**0.55199230**
Naive	XRP	N/A	0.22385468	0.27592453	0.00000000
StaticVWAP	XRP	Absolute	0.18250506	0.18804847	-0.26329018
StaticVWAP	XRP	Quadratic	0.18838942	0.16050903	-0.70089282
StaticVWAP	XRP	Volume Curve	0.21978152	0.29401255	0.05323325
DynamicVWAP	XRP	Absolute	**0.17419565**	0.17708713	-0.39368663
DynamicVWAP	XRP	Quadratic	0.18598679	**0.14618580**	-0.69780867
DynamicVWAP	XRP	Volume Curve	0.20582119	0.36081353	**0.47710991**

VWAP Execution with Signature-Enhanced Transformers: A Multi-Asset Learning Approach

In our previous papers, models were calibrated and deployed on individual assets. While effective, this approach necessitates increasingly complex production pipelines, particularly in cryptocurrency markets where new assets are frequently introduced.

To address this limitation, we enhance our model with a more sophisticated architecture inspired by *A Temporal Kolmogorov-Arnold Transformer for Time Series Forecasting* (Genet and Inzirillo). This design leverages attention mechanisms while maintaining causality through appropriate masking to prevent forward-looking bias.

Furthermore, we incorporate a learnable path signature layer to generate additional covariates over extended sequences, offering superior computational efficiency compared to traditional RNN approaches for long-sequence processing.

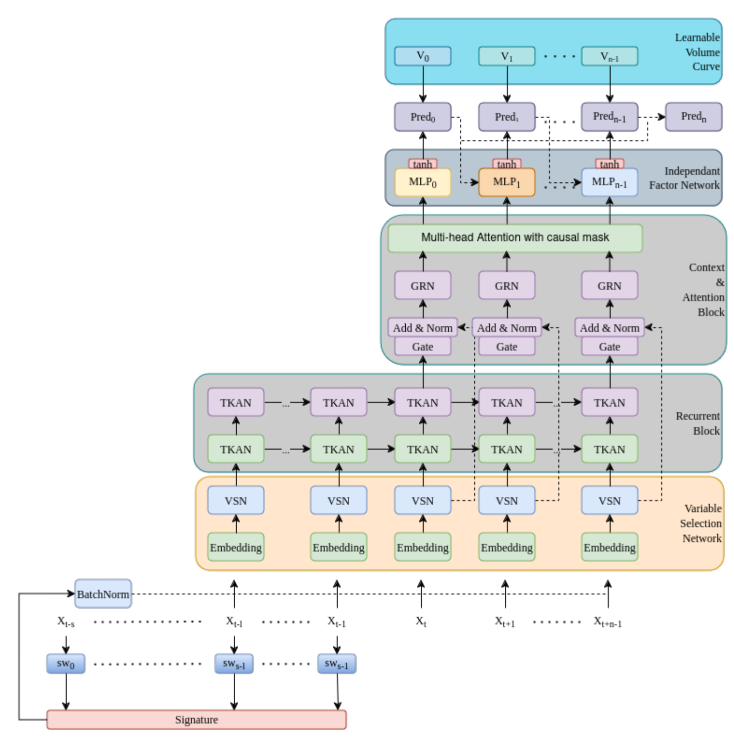


Figure 3. Signature-Enhanced Transformer Architecture for Multi-Asset VWAP Execution

We evaluate performance through comparative analysis, measuring average improvements relative to the equally-weighted baseline. The models compared are:

- AFD: Asset-Fitted Dynamic model from our previous work (note: test asset distinction not applicable)
- GFD: Globally-Fitted Dynamic model from our previous work
- GFT-Sig: Globally-Fitted Dynamic Transformer with signature components, representing our complete proposed model
- GFT: Globally-Fitted Dynamic Transformer without path signature components, serving as an ablation study

All results presented below are out-of-sample evaluations, where 'train' and 'test' designate whether the assets were present in the model's training dataset. Both asset-fitted and globally-fitted models are evaluated using identical out-of-sample test periods across all assets to ensure fair comparison.

Table 3. Comparative Performance Analysis: Average Improvement over Naive Baseline by Model and Asset Category

Model	Train/Test	Absolute VWAP Loss Improvement	Quadratic VWAP Loss Improvement
AFD	Test	16.46%	32.68%
AFD	Train	15.79%	31.09%
GFD	Test	19.22%	35.18%
GFD	Train	19.78%	36.91%
GFT	Test	20.22%	26.18%
GFT	Train	20.01%	25.56%
GFT-Sig	Test	21.87%	35.96%
GFT-Sig	Train	21.91%	36.98%