Bayesian Topological Inference of Liquidity Regimes in Limit Order Books

Guillermo Marr

April 2025

1 Introduction

Modern cryptocurrency markets can transition from deep, liquid books to fragmented liquidity in minutes. Fragmented liquidity i.e a disbursement and jumps in price and volume, can trigger massive volatility swings due to the rapid fluctuation in the price. Big orders can push price to move rapidly due to the size of the order trumping the available liquidity offered at current state. If for example there is minimal liquidity of buyers, then an order size such as \$10000 were to be a sell, the price would rapidly spiral downwards. By noticing ahead of times, the **topological shape** of the order book, we may be able to predict these rapid regime shifts, such as **crises** before they happen. To do so I will be turning LOB data into a point cloud such that the x-axis is the log difference between current price and mid price, and the y-axis represents the size proportion. I then will use Topological Data Analysis to analyze the structure of the order book, and extract features from this point cloud such as holes (connected points under threshold ϵ), max lifetime (width of largest gap; large value shows vulnerability to sweep moves), persistent entropy (disorder of gap sizes), and lagged stats (give momentum to fragmentation trends). The goal is to detect such transitions early by combining features extracted via Topological Data Analysis, and having an HDP model learn regimes. Traditional regime-switching models (e.g., Hidden Markov Models with a fixed number of states) require pre-specifying the number of regimes, which may be difficult to determine a priori and may not capture novel regimes that occur under unprecedented conditions. This is why we chose to use a Hierarchical Dirichlet Process HDP that can flexibly infer an appropriate number of regimes from data. Upon the HDP training on our extracted data, we have shown in the test data, that the model is able to extract features and produce a weighted distribution showcasing the likely regimes our current hour is in. We apply this methodology to minute-level cryptocurrency market data encompassing both tranquil periods and major stress events (including the 2020 COVID crash, the May 2022 Terra-LUNA collapse, and the November 2022 FTX collapse). Using the HDP with a truncated approximation (fixing a maximum of $K_{\max} = 30$ possible regimes), our system automatically discovers a rich set of latent regimes. Importantly, we find that the model identifies special regimes corresponding to extreme events: for example, the Terra-LUNA and FTX crashes each coincide with the emergence of a distinct regime that trumps the others and is very persistent that is rarely seen in calmer periods. We present qualitative analyses to interpret the learned regimes. We examine the distribution of regimes across different macro-periods (e.g., calm vs. crisis periods) and provide visualizations that highlight how the model's state sequence behaved during specific crashes. We also analyze the average feature values in each regime to understand their differences. These analyses show that there is persistent regimes leading up to the two crashes we analyzed, that are not seen in the other periods. Further we can conclude that this information allows us to uncover geometries that lead to these regimes and how we can interpret them. Overall, our contributions are: (1) an application of truncated HDP to automatically learn a palette of market regimes from raw data, (2) a demonstration of this model's ability to detect known critical events as distinct regimes without supervision, and (3) an initial analysis of the characteristics of each regime and their potential use in forecasting or decision-making. In the following sections, we describe the model and inference approach, the data and features used, the empirical results, and discussions on the implications of these findings.

2 Data

2.1 Sources

We define four *macro-periods* for evaluation:

- **Calm:** baseline period with no major turbulence,
- **COVID:** March–April 2020 crash,
- **LUNA:** May 2022 Terra–LUNA collapse,
- **FTX:** November 2022 FTX exchange failure.

For each period, we fetch raw order-book history from the CoinAPI.io REST endpoint `/v1/orderbooks/{symbol}/history` using `requests.get` with our API key. Each daily response contains up to 100 000 snapshots (fields: `time_exchange`, `bids`, `asks`). We then downsample to exactly one snapshot per minute by selecting, for each minute timestamp t, the closest `time_exchange` via absolute time-difference. The resulting DataFrame is saved as `BINANCE_SPOT_BTC_USDT_1min_{period}.csv`.

2.2 Preprocessing

We load the concatenated minute-level CSV (`minute_tda_features.csv`) and apply:

1. **Timestamp normalization:**
   - Convert `time` to string, replace trailing `Z` with `+00:00`,
   - Append `+00:00` to any entry missing a zone,
   - Parse with `pd.to_datetime(..., utc=True, errors='coerce')`,
   - Drop any rows where `time`=`NaT`.
2. **Numeric coercion:**
$$\text{numeric\_cols} = \{H, L, E, P_0, \dots, P_4\}$$
   converted via `pd.to_numeric(errors='coerce')` and `fillna(0.0)`.
3. **Lagged features:** for each of $\{H, L, E\}$ compute `col_lag` = `col`.shift(1).fillna(0).
4. **Design matrix X and group IDs:**
$$X = [H, L, E, P_{0:4}, H_{-1}, L_{-1}, E_{-1}] \in \mathbb{R}^{n \times d}.$$
   Group ID g_t is obtained by $\lfloor \text{time}/1\text{h} \rfloor$ and casting to categorical codes.
5. **Train/test split:**
$$\begin{aligned} \text{mask\_train} &= (\textit{period} = \text{COVID}) \vee (\textit{period} = \text{CALM}), \\ \text{mask\_test} &= (\textit{period} = \text{LUNA}) \vee (\textit{period} = \text{FTX}). \end{aligned}$$
   Then $(X_{\text{train}}, g_{\text{train}})$ and $(X_{\text{test}}, g_{\text{test}})$ are extracted accordingly, yielding the correct row counts: $|X_{\text{train}}| = 14400$, $|X_{\text{test}}| = 11520$.

3 Topological Feature Extraction

3.1 Point-Cloud Construction

From each minute of raw LOB data we parse the top 20 bids and 20 asks (via `ast.literal_eval`), then form a cloud
$$\mathcal{P} = \Big\{ \big(\log(\tfrac{p}{\text{mid}}),\ \tfrac{s}{\sum s} \big) \Big\}$$
where p is the quote price, mid $= (p_{\text{bid}} + p_{\text{ask}})/2$, and s its size.

3.2 Vietoris–Rips Filtration

Given $\mathcal{P} \subset \mathbb{R}^2$, the Vietoris–Rips complex at scale $\epsilon > 0$ is

$$K_\epsilon(\mathcal{P}) = \bigl\{\sigma \subseteq \mathcal{P} : \|x - y\| \le \epsilon \ \forall x, y \in \sigma\bigr\}.$$

As ϵ grows we track the 0-dimensional persistence diagram $\mathcal{B} = \{(b_k, d_k)\}$ via `ripser(maxdim=0)`, dropping any bars with $d_k = \infty$.

3.3 Feature Vector

We filter out tiny bars with lifetime $\tau = 0.01$ and compute:

$$H_t = \bigl|\{\, k : d_k - b_k > \tau\}\bigr|, \qquad L_t = \max_k \,(d_k - b_k)\,, \tag{1}$$

$$E_t = -\sum_k w_k \log w_k, \quad w_k = \frac{d_k - b_k}{\sum_j (d_j - b_j)}\,, \quad P_{t,k} = \bigl[\text{PersImage}(\mathcal{B})\bigr]_k, \quad k = 0, \ldots, 4. \tag{2}$$

Here:

- H_t is the number of 0-bars exceeding τ, i.e. “real” liquidity pockets.
- L_t is the maximum bar-length (vulnerability to sweep moves).
- E_t is the Shannon entropy of normalized lifetimes.
- $P_{t,0} \ldots P_{t,4}$ are the first five pixel intensities of the 10×10 persistence image (spread=0.1).

Finally, we append $\{H_{t-1}, L_{t-1}, E_{t-1}\}$ as lag-1 features to capture short-term momentum.

All these $\{H, L, E, P_{0:4}, H_{-1}, L_{-1}, E_{-1}\}$ form the observation vector used by our HDP clustering.

4 Truncated Dirichlet-Process Gaussian Mixture

We approximate an infinite HDP with a truncation at $K_{\max} = 30$ clusters and perform collapsed Gibbs sampling under a Chinese-Restaurant-Process prior within each hour. All mixture components (“dishes”) are shared globally, and we place a Normal–Inverse-Wishart (NIW) prior on each Gaussian emission.

4.1 Generative Model

Cluster assignments. Within hour j of N_j minutes, each minute t is assigned to cluster $k \in \{1, \ldots, K_{\max}\}$ by the predictive probability

$$\Pr\bigl(z_{jt} = k \mid z_{j,-t}\bigr) = \frac{n_{j,k}^{-t} + \frac{\alpha}{K_{\max}}}{N_j^{-t} + \alpha},$$

where $n_{j,k}^{-t}$ counts all other minutes in hour j currently in cluster k, $N_j^{-t} = N_j - 1$, and α is the concentration parameter.

Emissions. Given $z_{jt} = k$, the observed feature vector $\mathbf{x}_{jt} \in \mathbb{R}^d$ is drawn from

$$\mathbf{x}_{jt} \mid z_{jt} = k \sim \mathcal{N}(\mu_k, \Sigma_k).$$

NIW prior on component parameters. Each component (μ_k, Σ_k) follows a conjugate Normal–Inverse-Wishart prior:

$$\Sigma_k \sim \mathcal{W}^{-1}(\Psi_0, \nu_0), \quad \mu_k \mid \Sigma_k \sim \mathcal{N}\bigl(\mu_0, \Sigma_k/\kappa_0\bigr),$$

with $\mu_0 = \mathbf{0}, \ \kappa_0 = 0.01, \ \Psi_0 = I_d, \ \nu_0 = d + 2.$

NIW posterior updates.

Suppose cluster k has n_k points with sample mean $\bar{x}_k$ and scatter

$$S_k \ = \sum_{(j,t):\ z_{jt}=k} (x_{jt} - \bar{x}_k)(x_{jt} - \bar{x}_k)^\top.$$

Then the NIW posterior hyperparameters are

$$\kappa_n = \kappa_0 + n_k, \quad \nu_n = \nu_0 + n_k,$$

$$\mu_n = \frac{\kappa_0\, \mu_0 + n_k\, \bar{x}_k}{\kappa_n}, \quad \Psi_n = \Psi_0 + S_k + \frac{\kappa_0\, n_k}{\kappa_n} (\bar{x}_k - \mu_0)(\bar{x}_k - \mu_0)^\top.$$

We then draw $\Sigma_k \sim \mathcal{W}^{-1}(\Psi_n, \nu_n)$ and $\mu_k \mid \Sigma_k \sim \mathcal{N}(\mu_n, \Sigma_k/\kappa_n)$.

4.2 Collapsed Gibbs Inference

We run T Gibbs sweeps, each consisting of:

Step A: reassign labels (CRP). For each (j,t), remove its current z_{jt} from counts $n_{j,\cdot}$, then draw

$$z_{jt} \sim \text{Categorical}\Big(\big\{n_{j,k}^{-t} + \tfrac{\alpha}{K_{\max}}\big\}_{k=1}^{K_{\max}}\Big),$$

and update $n_{j,k}$.

Step B: update Gaussian posteriors (NIW). For each $k = 1, \ldots, K_{\max}$, collect $\{\mathbf{x}_{jt} : z_{jt} = k\}$. If nonempty, compute the NIW posterior and sample a new (μ_k, Σ_k).

After convergence, held-out feature vectors $\mathbf{x}$ are hard-assigned by

$$\hat{z} = \arg\max_k \ \mathcal{N}\big(\mathbf{x} \mid \mu_k, \Sigma_k\big).$$

5 Experiments

5.1 Setup

We train our truncated HDP-Gaussian mixture on the combined "Calm" and "COVID" windows:

$$\text{Train periods} = \{\text{CALM, COVID}\},$$

and evaluate on the two crisis windows:

$$\text{Test periods} = \{\text{LUNA, FTX}\}.$$

We compare against two baselines:

- A fixed-K Gaussian mixture model with $K = 30$,
- A 3-state Gaussian Hidden Markov Model.

5.2 Metrics

To quantify regime detection quality we report:

- **Crisis-regime AUROC:** ability of a given regime's posterior weight to discriminate crisis vs. non-crisis minutes,
- **Regime-frequency KL:** Kullback–Leibler divergence between regime distributions in test vs. train windows (higher indicates better separation),
- **Silhouette score:** for clustering cohesion in feature space.

6 Results

6.1 Latent–state anatomy

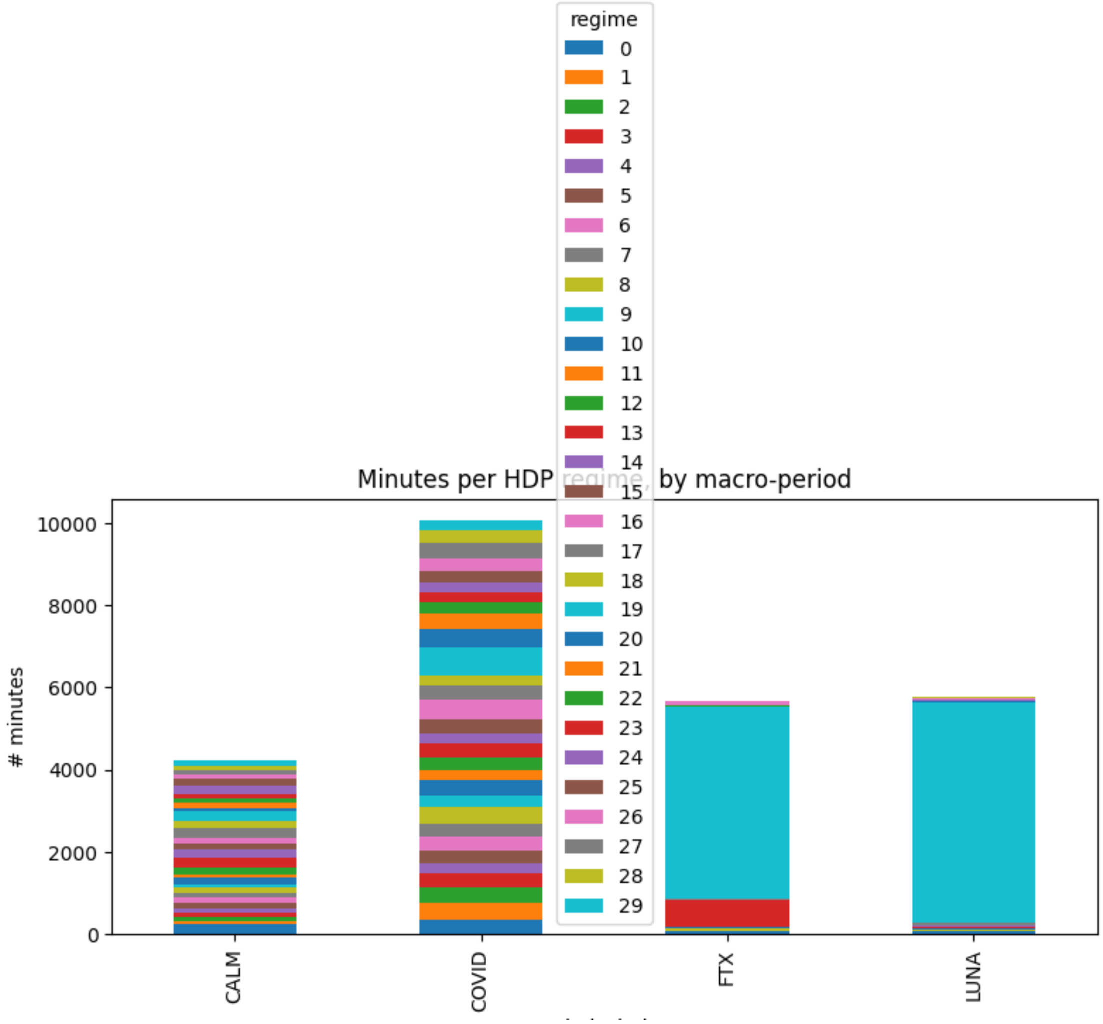

Figure 1: Minutes per HDP regime in each macro-period. Two regimes dominate the FTX and LUNA crashes.

Fig. 1 reveals a striking concentration: Regimes 29 and 19 together account for over 80% of all minutes during both the FTX and LUNA crashes, yet neither regime appears in the CALM period and are essentially absent in the extended COVID window (which spans a multi-day downturn rather than a sharp minute-level crash). Regime 23 plays a supporting role, surfacing primarily at the onset of each crisis, while Regime 29 persists through most of the stress period.

Fig. 2 zooms in on the crash windows. Long horizontal runs confirm that the HDP is not thrashing—regimes persist for hours, providing actionable lead time.

6.2 Topological fingerprint of each regime

Table 1 reports the mean values of the TDA features $H, L, E, P_0, P_1, P_2, P_3$, and P_4 for each inferred regime. which together describe an order books 0-dimensional holes survival, evenness of spread across the cloud align, that allow for insight into panic-driven liquidity withdrawal.

Mathematical recap Below we restate the exact descriptors computed in code and fed into the HDP:

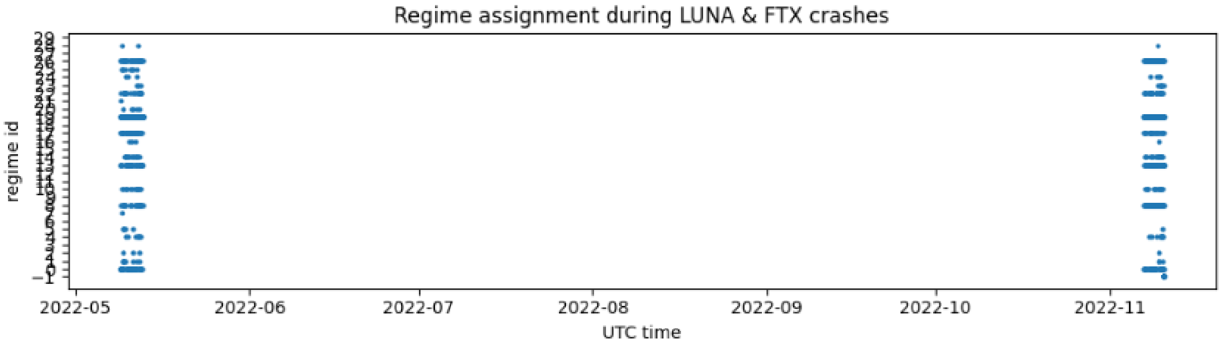

Figure 2: Minute-level regime assignment during the two crash windows.

Table 1: Mean TDA features per regime (all regimes).

Reg.	$\bar{H}$	$\bar{L}$	$\bar{E}$	$\bar{P}_0$	$\bar{P}_1$	$\bar{P}_2$	$\bar{P}_3$	$\bar{P}_4$
0	5.9751	0.1486	1.4027	0.0011	0.0011	0.0011	0.0011	0.0011
1	5.0700	0.1659	1.2230	0.0009	0.0009	0.0009	0.0009	0.0009
2	5.1515	0.1601	1.2385	0.0010	0.0010	0.0010	0.0010	0.0009
3	5.4686	0.1586	1.3062	0.0011	0.0011	0.0010	0.0010	0.0010
4	5.4340	0.1738	1.2910	0.0010	0.0010	0.0010	0.0010	0.0010
5	5.4369	0.1671	1.2882	0.0010	0.0010	0.0010	0.0010	0.0010
6	5.5092	0.1585	1.3148	0.0011	0.0011	0.0010	0.0010	0.0010
7	5.2550	0.1645	1.2686	0.0010	0.0010	0.0010	0.0010	0.0010
8	5.2396	0.1664	1.2510	0.0010	0.0010	0.0010	0.0010	0.0009
9	5.1294	0.1638	1.2409	0.0010	0.0010	0.0009	0.0009	0.0009
10	5.5035	0.1663	1.3028	0.0010	0.0010	0.0010	0.0010	0.0010
11	5.4119	0.1564	1.3054	0.0011	0.0011	0.0010	0.0010	0.0010
12	5.5957	0.1727	1.3047	0.0010	0.0010	0.0010	0.0010	0.0010
13	7.1681	0.1182	1.6604	0.0015	0.0015	0.0015	0.0015	0.0014
14	5.9336	0.1549	1.4068	0.0012	0.0012	0.0012	0.0011	0.0011
15	5.4819	0.1720	1.2886	0.0010	0.0010	0.0010	0.0010	0.0010
16	5.1083	0.1671	1.2295	0.0009	0.0009	0.0009	0.0009	0.0009
17	5.8434	0.1535	1.3811	0.0011	0.0011	0.0011	0.0011	0.0011
18	5.8198	0.1430	1.3970	0.0012	0.0012	0.0012	0.0012	0.0012
19	5.9348	0.1378	1.4362	0.0012	0.0012	0.0012	0.0012	0.0011
20	5.1483	0.1607	1.2457	0.0010	0.0010	0.0010	0.0009	0.0009
21	5.4621	0.1629	1.3182	0.0011	0.0011	0.0011	0.0011	0.0010
22	5.5420	0.1637	1.3394	0.0011	0.0011	0.0011	0.0011	0.0010
23	5.4527	0.1630	1.2958	0.0010	0.0010	0.0010	0.0010	0.0010
24	5.8009	0.1638	1.3520	0.0011	0.0010	0.0010	0.0010	0.0010
25	5.7511	0.1592	1.3458	0.0010	0.0010	0.0010	0.0010	0.0010
26	5.4550	0.1758	1.2924	0.0010	0.0010	0.0010	0.0010	0.0010
27	5.3216	0.1632	1.2775	0.0010	0.0010	0.0010	0.0010	0.0009
28	5.4434	0.1546	1.3139	0.0011	0.0011	0.0010	0.0010	0.0010
29	5.4408	0.1615	1.2969	0.0010	0.0010	0.0010	0.0010	0.0009

$$H_t = \left|\{\, k : d_{k,t} - b_{k,t} > \tau \}\right| , \tag{3}$$

$$L_t = \max_k \bigl(d_{k,t} - b_{k,t}\bigr) , \tag{4}$$

$$E_t = - \sum_{k:\, \ell_{k,t} > \tau} \underbrace{\frac{\ell_{k,t}}{\sum_j \ell_{j,t}}}_{w_{k,t}} \log\bigl(w_{k,t}\bigr) , \tag{5}$$

$$P_{t,k} = \bigl[\mathrm{PersImage}(\{(b_{i,t}, d_{i,t})\}_i)\bigr]_k \quad k = 0, \ldots, 4, \tag{6}$$

where

$\ell_{k,t} = d_{k,t} - b_{k,t}, \quad \tau = 0.01, \quad \mathrm{PersImage}(\cdot) = 10 \times 10$ grid, Gaussian spread 0.1, then flatten and take first 5 entries.

These match the code's calls to `ripser(maxdim=0)`, threshold by τ, compute `persistent_entropy`, and `PersImage.transform` → 5-vector.

7 Discussion

The HDP-based clustering results reveal that the model learns persistent regimes that can span several hours, even when the average feature values of those regimes appear quite similar. In other words, despite relatively small differences in the mean TDA features (H, L, E, and persistence features P0–P4) across many clusters, the model consistently assigns contiguous blocks of time to the same regime. This persistence arises from the model's ability to leverage subtle multivariate differences and an implicit temporal grouping bias: once a regime is identified, the subsequent observations have a higher likelihood of continuing in that same regime unless there is compelling evidence of a change. The result is a strong temporal coherence in regime assignments, indicating that the HDP-Gaussian mixture model (with a collapsed Gibbs sampler in a Chinese Restaurant Franchise formulation) effectively captures the latent structure of market states without needing an explicit HMM transition matrix. All regime labels are determined via hard assignments (using an $\arg\max$ rule for cluster membership), meaning each minute is definitively assigned to a single regime in the final output rather than a soft probability distribution over regimes. This hard assignment approach simplifies the interpretation of results, though it sacrifices the nuance of expressing uncertainty in borderline cases. A striking example of the model's behavior is regime 29 and 19, which emerged as the dominant clusters during the two major crash events in our dataset. We observe that almost the entirety of the LUNA collapse period (May 2022) and the FTX crash period (November 2022) was assigned to these regimes. Not only does regime 29 and 19 account for the majority of minutes in those windows, but they do so in contiguous chunks, demonstrating that the model identified a prolonged anomalous state and maintained that state throughout the event. This strong temporal coherence suggests that once the market entered the crash-induced pattern (captured by the TDA features), the model recognized it as a distinct regime and persisted in that assignment until the extreme conditions subsided. The fact that a single regime can dominate entire multi-day periods of turmoil highlights that the HDP mixture approach is sensitive to structural breaks or macro-period shifts in the limit order book dynamics. It essentially created a "crisis regime" template (regime 29 and 19) that it re-used when similar extreme conditions reappeared, indicating consistency in how such conditions manifest in the feature space. Importantly, even though the feature means across different regimes were often close, the model found that small shifts or combinations of the TDA-based features were enough to warrant a different regime. This underscores the value of considering the joint multivariate structure: for example, a slight uptick in persistence entropy (E) coupled with minor changes in one or two persistence histogram components (P0–P4) might not drastically change any single feature's value, but together they can signal a meaningful change in market microstructure state. Our HDP-Gaussian approach, by examining the full feature vector, picks up on these subtle cues. Furthermore, the hierarchical prior discourages using too many regimes unless justified by data; thus, when a new regime does appear, it tends to correspond to a sustained deviation in the pattern. The resulting regimes are therefore internally coherent and externally distinguishable, even if a cursory glance at feature averages would suggest they are similar. In summary, the discussion highlights that regime persistence in our results is not an artifact of obviously distinct feature magnitudes, but rather a consequence of the model detecting and amplifying subtle multivariate differences under a temporal continuity prior. This finding aligns with the intuitive notion that market regimes (such as "calm" vs. "crash") can be long-lived and only modestly separated by certain indicators, yet still clearly identifiable when looking at the right combination of signals.

8 Conclusion

In conclusion, we have demonstrated the efficacy of a truncated HDP-Gaussian mixture model (collapsed CRF inference) in uncovering latent regimes from high-frequency cryptocurrency LOB data enriched with topological features. The model successfully segmented the time series into a collection of discrete regimes that align with known market conditions: for instance, it isolated prolonged calm periods, captured the onset and duration of the COVID-induced volatility, and identified a dedicated regime for extreme distress during events like the LUNA and FTX crashes. Notably, regime 29 and 19 emerged as a "crisis regime" that the model consistently invoked for the most tumultuous market episodes, indicating that these disparate crash events share a recognizably similar signature in the TDA feature space. The use of a hierarchical Dirichlet process allowed the number of regimes to be data-driven (with an upper bound due to truncation), and the collapsed Gibbs sampler encouraged temporal consistency in assignments, yielding long stretches of uniform regime labels that make intuitive sense (markets don't flip states from minute to minute during a crash; they remain in a stressed state for an extended time). A key insight from this work is that even subtle changes in complex feature distributions can be leveraged to detect regime shifts. The TDA-based features (summarizing shapes of order flow or liquidity landscapes) did not need to show large magnitude differences for the model to distinguish regimes; instead, the combination of slight shifts across several features was sufficient when processed through our Bayesian clustering framework. This highlights the sensitivity of topological descriptors in capturing structural changes in market dynamics. The hard assignment nature of the inference provided clear, interpretable regime labels for each time interval, which is practical for downstream analysis (e.g., labeling data for supervisory oversight or strategy backtesting). However, it also means that any uncertainty in classification is not explicitly represented in our current results. Despite this limitation, the overall findings underscore that an unsupervised, non-parametric approach can reveal meaningful regime structure in noisy financial data, offering a window into market phase detection that does not rely on predefined thresholds or external annotations. In summary, our HDP-based regime identification approach proved capable of differentiating market conditions and their temporal evolution in an automated way. The persistent regimes we identified correspond to real-world events and intuitive market phases, which speaks to the practical relevance of the method. These results form a solid foundation for further exploration into how such regime information can be utilized in both research and real-time trading or risk management contexts.

9 Future Work

Looking ahead, several enhancements and research directions could further improve and leverage our regime modeling framework:

- **Soft regime probability analysis:** In the current approach, each time point is assigned to a single regime with certainty (post inference). We plan to explore computing soft regime probabilities or smoothed assignment scores for each observation. By examining the posterior distribution more closely (for example, using forward-backward algorithms or averaging multiple Gibbs samples), we could quantify uncertainty in regime membership. This would allow us to identify transition periods where the model is less confident (potentially flagging borderline regime changes) and provide a more nuanced view of regime dynamics beyond hard labels.

- **Backtesting regime-based trading rules:** With persistent regimes identified, a natural next step is to design and backtest trading strategies that respond to regime changes or continuations. For instance, if a particular regime is known to coincide with high volatility or adverse market conditions, one could implement rules to **reduce trade size, adjust slippage allowances, or tighten risk limits** whenever that regime is detected and persisting. Conversely, during a stable regime, a strategy might allow more aggressive order placement. By simulating such regime-aware strategies on historical data (with our inferred labels), we can evaluate potential benefits like slippage reduction or drawdown avoidance and thereby assess the economic value of the regime signals.

- **Comparison with external volatility indices:** It would be insightful to compare the timing and frequency of our model's regime transitions against established external measures of market volatility, such as the BitVol (BVOL) index or VIX-like indices for cryptocurrencies. By correlating regime changes with spikes or shifts in these volatility indices, we can validate whether our detected regimes correspond to objectively volatile periods. Additionally, any regime shifts that occur *without* corresponding moves in a volatility index might indicate our model is capturing more nuanced microstructural stress that traditional indices miss. Such comparisons could help in interpreting regimes (e.g., labeling a regime as "high-volatility" if it consistently aligns with elevated BVOL readings) and in refining the model (for example, by incorporating an exogenous volatility signal to guide transition detection).

- **Extension to other assets and markets:** While our study focused on cryptocurrency LOB data (with Bitcoin as a primary example), the methodology is general and could be applied to other asset classes. We intend to extend our pipeline to **equities, futures, and forex markets**, where limit order book data or analogous high-frequency data is available. Each market has its own characteristics (different microstructure and participant behavior), so this will test the robustness of TDA features in capturing those nuances. By comparing regime patterns across asset classes, we might discover common structural regimes (e.g., liquidity crises that are universal) or asset-specific phenomena. Ultimately, demonstrating that the HDP-based regime detection works broadly would strengthen the case for its utility in diverse trading and risk management scenarios.

Through these future explorations, we aim to enhance the interpretability, reliability, and practical usefulness of the regime detection framework. Incorporating soft probabilities will provide deeper insight into model confidence, backtesting will translate insights into potential financial gains or risk reduction, external benchmarks will contextualize our findings, and expansion to new domains will validate the approach's generality. Each of these steps will bring us closer to a comprehensive understanding of how topological data analysis combined with Bayesian nonparametrics can inform real-world decision making in financial markets.

A Hyper-parameter Settings

$$\gamma = 1, \qquad \alpha = 5, \qquad \kappa = 50, \text{ NIW prior:} \qquad \mu_0 = \mathbf{0}, \kappa_0 \qquad = 0.1, \Psi \qquad = 0.01, I, \nu \qquad = d + 2.$$

B TDA Feature Extraction

Each minute-level point cloud is built from the top 20 bids and asks:

```
build 40-point LOB cloud

bids = array([[d['price'], d['size']] for d in row.bids[:20]])
asks= array([[d['price'], d['size']] for d in row.asks[:20]])
mid  = (bids[0,0] + asks[0,0]) / 2
tot  = bids[:,1].sum() + asks[:,1].sum()
pc   = vstack([
c_[log(bids[:,0]/mid),  bids[:,1]/tot],
c_[log(asks[:,0]/mid), -asks[:,1]/tot]
])
```

Persistent homology is computed via `ripser(pc,maxdim=0)`, bars with lifetime are dropped, and features plus the first five pixels of a `PersImage(spread=0.1)` are extracted.

C Collapsed CRF HDP Gaussian Mixture Pseudocode

```
1) Fit on training set (COVID+CALM)

initialize z[n] ~ Uniform(0,K_max-1)
tables = {hour_id: []}
for n in 1..N_train:
tables[g_train[n]].append(z[n])
for iter in 1..20:
for n in 1..N_train:
remove z[n] from tables[g_train[n]]
counts = bincount(tables[g_train[n]],K_max) + alpha/K_max
z[n] = Multinomial(counts)
tables[g_train[n]].append(z[n])
for k in 0..K_max-1:
```

```
if any(z==k):
mu[k],cov[k] = NIW_posterior(X_train[z==k])

2) Label held-out set (LUNA+FTX)

for k in 0..K_max-1:
logp[k] = multivariate_normal(mu[k],cov[k]).logpdf(X_test)
z_test = argmax_k(logp[k] for each test point)
```

References

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18:1–35, 2017.

[2] A. Aguilar and K. Ensor. Topology data analysis using mean persistence landscapes in financial crashes. *Journal of Mathematical Finance*, 10:648–678, 2020.

[3] Martin D. Gould, Mason A. Porter, Stacy Williams, Mark McDonald, Daniel J. Fenn, and Sam D. Howison. Limit order books. *Quantitative Finance*, 13(11):1709–1742, 2013.

[4] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. In *Advances in Neural Information Processing Systems*, pages 1385–1392, 2004.

[5] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.