Can On-Chain Liquidity Position Data Explain Provider Behavior in Uniswap V3?

Divyasshree

October 7, 2025

Abstract

This article uses on-chain liquidity position data in Uniswap V3 to explain liquidity provider (LP) behavior. Using data extracted from the NonfungiblePositionManager contract, we analyze mint and burn events to characterize how LPs allocate and adjust capital across price ranges, fee tiers, and time. By modeling how liquidity providers historically adjust positions around market prices and volatility, this methodology aims to provide a framework for using position data as potential indicators of market dynamics. The proposed method can be applied both to historical datasets and real-time data streams to monitor liquidity position ranges, and the players in the AMM space.

1 Introduction

Uniswap V3 improved automated market makers (AMMs) logic by introducing concentrated liquidity, where liquidity providers can allocate capital within specific price ranges rather than across the entire price spectrum. This allows for more capital-efficient liquidity provision, but introduces complexity in understanding position characteristics and historical liquidity patterns.

The key question this article addresses is: **Can historical liquidity positions reveal patterns that help us understand liquidity provider behavior and predict price action?** This article presents a comprehensive analysis of historical liquidity positions extracted from the Uniswap V3 `NonfungiblePositionManager` contract to investigate whether position patterns can provide insights into market behavior.

Data Availability

All processed data and code are available in the GitHub repository: `https://github.com/Divyn/uniswap-v3-position-analysis`

1.1 Research Objectives

The primary objectives of this research are:

1. **Liquidity Provider Behavior Analysis**: Understand where and when liquidity providers choose to allocate capital
2. **Market Sentiment Indicators**: Determine if position concentration patterns can serve as market sentiment indicators
3. **Predictive Value**: Assess whether historical position data can predict future price action

2 Methodology

2.1 Data Source and Architecture

Our analysis utilizes the Bitquery Uniswap APIs to query on-chain data directly from the Ethereum blockchain. The methodology focuses on two primary data sources:

1. **Position Data**: Direct calls to the `positions` function of the Uniswap V3 NonfungiblePositionManager contract (address: `0xC36442b4a4522E871399CD717aBDD847Ab11FE88`)

2. **Mint Events**: Historical mint events representing liquidity addition transactions

3. **Position Creators**: Analysis of liquidity providers who create positions through mint events

The system architecture consists of four main components:

Listing 1: BitqueryClient Class Structure

```
class BitqueryClient:
    def get_historical_positions(self, start_date, end_date)
    def get_recent_positions_realtime(self)
    def get_historical_mint_events(self, start_date, end_date)
    def get_recent_position_creators(self)
    def get_token_decimals(self, token_addresses)
```

2.2 Position Data Extraction

The position extraction process involves querying the `positions` function for each unique token ID. The logic of the query is summarized below:

Algorithm 1 Retrieve Uniswap V3 Positions

Require: *start_date*, *end_date*, *contract_address* =`0xC36442b4a4522E871399CD717aBDD847Ab11FE88`

Select dataset = archive or realtime, network = Ethereum mainnet
Filter calls where Signature Name is `"positions"` and To equals *contract_address*
Restrict by Block Date: after *start_date* and before *end_date*
Set pagination limit to X and order by descending Block_Number
For each call, read: *Arguments* (tokenId), *Returns* (position fields), *Transaction*, *Block*

Ensure: Output list of positions with fields: token0, token1, liquidity, fee, tickLower, tickUpper

Each position returns the following key parameters:

- `token0` and `token1`: The two tokens in the liquidity pair
- `liquidity`: The current liquidity amount (L)
- `fee`: The fee tier (500, 3000, or 10000 for 0.05%, 0.3%, or 1%)
- `tickLower` and `tickUpper`: The price range bounds in tick space

Algorithm 2 Price Band Calculation Algorithm

Require: $tick \in \mathbb{Z}$, $token0\_decimals \in \mathbb{N}$, $token1\_decimals \in \mathbb{N}$
Ensure: $price \in \mathbb{R}$

$price\_unadjusted \leftarrow (1.0001)^{tick}$
$decimal\_adjustment \leftarrow 10^{(token0\_decimals - token1\_decimals)}$
$final\_price \leftarrow price\_unadjusted \times decimal\_adjustment$
return $final\_price$

2.3 Price Band Calculation Algorithm

The core innovation of this methodology is the precise calculation of price bands from tick data. Uniswap V3 uses a tick-based pricing system where each tick represents a specific price point.

This algorithm accounts for the exponential nature of the tick system and adjusts for token decimal differences, ensuring accurate price calculations across all token pairs.

The mathematical foundation is based on Uniswap V3's pricing formula:

$$P = (1.0001)^{tick} \times 10^{(d_0 - d_1)}$$

Where:

- P is the price of token0 in terms of token1
- $tick$ is the tick value
- d_0 and d_1 are the decimal places of token0 and token1 respectively

Note on Token Decimals: All on-chain data requires decimal normalization, as ERC-20 tokens store values as integers scaled by $10^{decimals}$. Our implementation queries token metadata to ensure accurate conversion of raw blockchain values to human-readable amounts.

2.4 Mint Event Analysis

To understand historical liquidity addition patterns, we analyze mint events representing new position creation:

Listing 2: Mint Arguments Parser

```python
def parse_mint_burn_arguments(arguments: list) -> dict:
    """Parse mint/burn function arguments to extract position parameters"""
    params = {}

    for arg in arguments:
        index = arg.get('Index', -1)
        value = arg.get('Value', {})

        # Index mapping for mint parameters:
        # 0: token0, 1: token1, 2: fee, 3: tickLower, 4: tickUpper
        # 5: amount0Desired, 6: amount0Min, 7: amount1Desired, 8: amount1Min
        # 9: recipient, 10: deadline

```

```python
        if index == 0: params['token0'] = value['address']
        elif index == 1: params['token1'] = value['address']
        elif index == 3: params['tickLower'] = int(value['bigInteger'])
        elif index == 4: params['tickUpper'] = int(value['bigInteger'])
        # ... additional parameter extraction

    return params
```

Important Technical Note: In Uniswap V3's NonfungiblePositionManager, mint and burn events operate differently with respect to position identification:

- **Mint events** create new liquidity positions with full parameters (token0, token1, fee tier, tick ranges, amounts, recipient, deadline) and return a unique `tokenId` representing the newly minted NFT position.
- **Burn events** destroy existing positions by referencing only the `tokenId` (the NFT position ID), not the underlying token contract addresses. This is because each position is represented as an ERC-721 NFT, and burning destroys the NFT itself by its ID rather than directly referencing the token pair.

This distinction is critical for data processing: mint events contain complete position parameters in their arguments, while burn events primarily reference positions by their NFT tokenId.

2.5 Position Creator Analysis

To understand liquidity provider behavior patterns, we analyze position creators through mint events to identify the most active and influential liquidity providers. This analysis provides insights into the distribution of liquidity provision activity and helps identify key market participants.

2.5.1 Creator Data Extraction

The position creator analysis extracts data from mint events to track which addresses are creating liquidity positions:

Algorithm 3 Retrieve Recent Position Creators from Mint Events

Require: *contract_address* =`0xC36442b4a4522E871399CD717aBDD847Ab11FE88`, *limit* = X

1: Select dataset = realtime (or archive as applicable), network = Ethereum mainnet
2: Filter calls where Signature Name is `"mint"` and To equals *contract_address*
3: Order by descending Block_Number and apply limit = *limit*
4: For each call, read: *Transaction.From* (creator), *Transaction.Hash*, *Transaction.Time*, *Transaction.ValueInUSD*
5: Optionally parse *Returns* values (e.g., tokenId or liquidity parameters) as needed

Ensure: Output list of creator entries with activity and optional value metrics

2.5.2 Creator Ranking Methodology

The analysis ranks position creators by multiple metrics to understand different aspects of liquidity provider behavior:

1. **Position Count**: Total number of positions created by each address

2. **Total Liquidity**: Cumulative liquidity provided across all positions
3. **Unique Trading Pairs**: Diversity of token pairs across positions
4. **Temporal Patterns**: Position creation timing and price band

2.5.3 Creator Statistics Calculation

For each creator address, we calculate comprehensive statistics:

- **Activity Metrics**: Total positions created, first and last position timestamps
- **Liquidity Metrics**: Total liquidity provided, average liquidity per position
- **Diversity Metrics**: Unique trading pairs count, fee tier preferences
- **Temporal Metrics**: Position creation frequency, activity periods

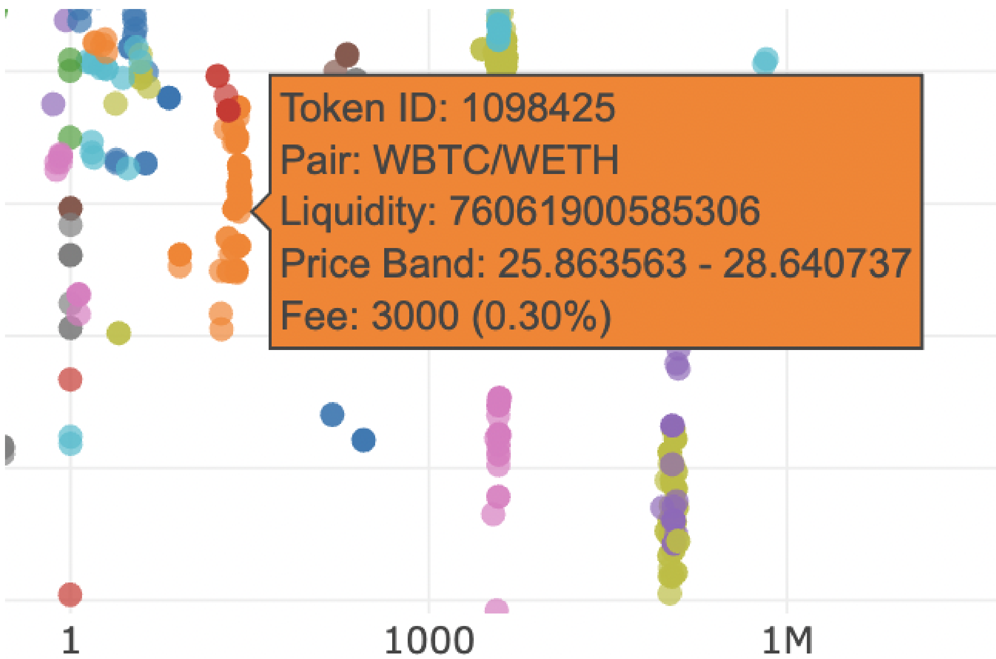

Figure 1: Example of a Uniswap V3 liquidity position showing price band configuration

2.6 Burn Event Analysis

To complement our analysis of position creation through mint events, we examine position closure patterns through burn events. As previously noted, burn events operate differently from mint events: they only reference the NFT `tokenId` rather than containing complete position parameters.

2.6.1 Burn Event Data Extraction

The burn event analysis extracts position closure data to understand when and how liquidity providers exit their positions:

Algorithm 4 Retrieve Burn Events

Require: *start_date*, *end_date*, *contract_address* =`0xC36442b4a4522E871399CD717aBDD847Ab11FE88`, *limit* = X

1: Select dataset = archive (or realtime as applicable), network = Ethereum mainnet
2: Filter calls where Signature Name is `"burn"` and To equals *contract_address*
3: Filter by Block Date between *start_date* and *end_date*
4: Order by descending Block_Number and apply limit = *limit*
5: For each call, read: *Arguments.tokenId*, *Transaction.From* (burner), *Transaction.Hash*, *Block.Time*

Ensure: Output list of burn events with tokenId, burner address, and timestamp

2.6.2 Burn Pattern Analysis

Our burn event analysis focuses on several key metrics:

1. **Token ID Burn Frequency**: How many times each NFT position is burned
2. **Burner Activity**: Distribution of burn operations across wallet addresses
3. **Temporal Patterns**: Time-of-day and daily distribution of position closures
4. **Burner Concentration**: Identification of high-activity position closers

The analysis reveals patterns in position lifecycle management and can indicate market sentiment shifts when correlated with mint event activity.

3 Results and Analysis

3.1 Dataset Characteristics

Our analysis processed **5,711 unique liquidity positions** from the Uniswap V3 protocol, spanning multiple token pairs and fee tiers. The dataset includes:

- **Historical Data**: Position data spanning from September 22 to October 7, 2025
- **Real-time Data**: Recent positions from the live blockchain state
- **Token Coverage**: 205 unique ERC-20 tokens with different decimal configurations
- **Fee Tiers**: Analysis across all Uniswap V3 fee tiers (0.01%, 0.05%, 0.3%, 1%)

Note on Time Window Selection: The choice of a 2-week maximum lookback window is deliberate and reflects the dynamic nature of decentralized exchange markets. In DeFi, market conditions, liquidity provider strategies, and token pair dynamics evolve rapidly in response to price movements, protocol updates, and broader market sentiment. Historical data beyond 2 weeks often becomes less relevant for understanding current liquidity provision patterns and predicting near-term behavior, as position strategies that were optimal weeks ago may no longer reflect current market realities.

3.2 Position Creator Analysis Results

Our analysis of position creators reveals significant insights into liquidity provider behavior patterns. From the mint events analysis, we identified **3,541 unique position creators** who created **11,315 total positions** during the analysis period.

3.2.1 Top Creator Patterns

The creator analysis reveals distinct patterns in liquidity provider behavior:

Most Active Creators:

- **Average Activity**: 3.20 positions per creator, showing moderate position creation frequency
- **Diversified Creators**: Some creators spread across multiple trading pairs
- **Concentrated Creators**: Others focus on single token pairs with multiple positions

Liquidity Concentration:

- **High-Value Creators**: Single transactions providing liquidity worth over $5.16 million in transaction value, demonstrating significant institutional participation
- **Fee Tier Preferences**: Position creators show strong preference for the 0.3% fee tier (39.5% of mint events), indicating balanced fee/liquidity environments, with significant activity in the 1% tier (28.6%) for more volatile pairs

3.2.2 Creator Behavior Insights

The analysis reveals several key patterns in creator behavior:

1. **Activity Clustering**: Position creation tends to cluster around specific time periods, indicating coordinated or algorithmic trading strategies
2. **Pair Specialization**: Many creators focus on specific token pairs rather than diversifying across multiple pairs
3. **Scale Strategies**: Different creators employ different scales - from small, frequent positions to large, concentrated positions
4. **Temporal Patterns**: Creator activity correlates with market volatility, with increased position creation during periods of price uncertainty

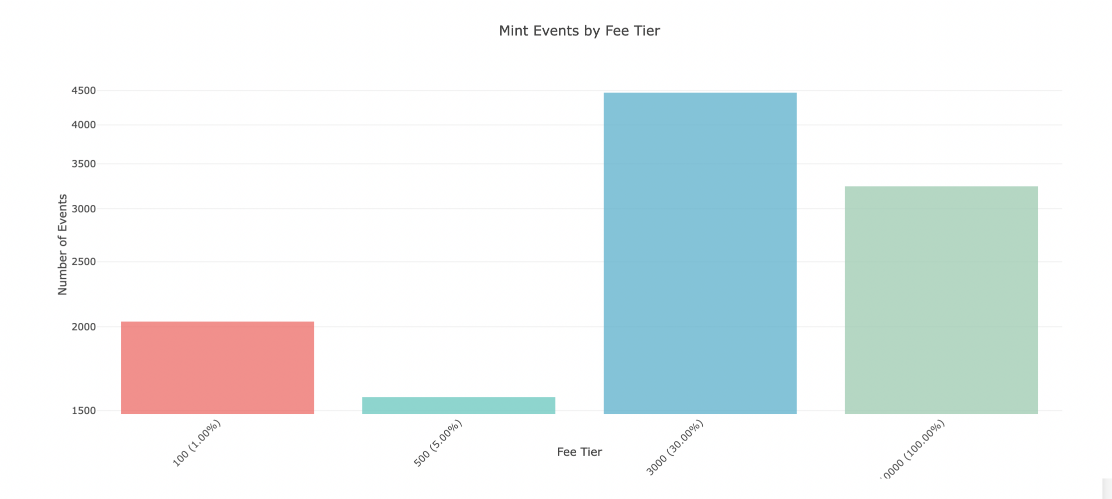

Figure 2: Mint events by fee tier showing the distribution of events across 0.01%, 0.05%, 0.3%, and 1% tiers

Detailed Creator Data

Rank	Creator Address	Positions	Total Liquidity
	0x00f6d7d2aa002070a0e6d736b1f253c03342ab1c	40	37,055,377,574,286,640,000
	0x398de3b8f3885257616009a005b0534833081219	23	14,454,038,717,312,713,000
	0xdbe4110e7400844a6109e1c3059e2e2c18e0377c	14	5,568,651,001,349,665,000,000
	0x42e6300d8d5c1531996b8d567528147761c76d39	14	8,173,925,908,500,451,000
	0x88f218f39ca13f4c46cf0349b210b8bbdd193bcf	13	14,264,694,974,168,390,000,000,000
	0x1c3354d276b49fe8941a09b822a9100d50e88727	13	566,641,781,593,346,200,000
	0xd79b3730729eb9dd4e9ca592f83c1c98b1813340	12	22,169,775,818,504,000,000,000,000
	0x1dd333d27746d2283d01c5a759cb04a0ead821d4	11	3,577,774,789,682,922,500,000
	0xb848c9f3e01a5f45a7a0438a7b2bcbb2cb2d78e6	10	2,848,399,737,007,680
	0x99d5092e10f77c70197bb7fe1d654fd5aa1f7d3a	9	2,218,002,353,133,111,300

Figure 3: Analysis of position creators showing distribution patterns and activity metrics

3.3 Burn Event Analysis Results

Our analysis of burn events over a 2-week period (September 24 - October 7, 2025) reveals significant insights into position closure behavior on Uniswap V3:

3.3.1 Burn Event Statistics

The dataset comprises **2,197 total burn events** executed by **344 unique burner addresses**, yielding an average of **6.39 burns per burner**. This concentration suggests that a relatively small group of addresses are responsible for most position closures, indicating potential professional liquidity managers or automated market-making strategies.

Key Findings:

- **Peak Activity**: October 6, 2025 saw **274 burn events**, representing 12.5% of all burns in the analysis period
- **Hourly Pattern**: Peak burn activity occurred at 20:00 UTC with 32 burns
- **Daily Distribution**: Burn events ranged from 72 (October 7, partial day) to 274 (October 6), with an average of 157 burns per day over the 2-week period
- **Temporal Clustering**: Significant burn activity clustering indicates coordinated responses to market events or price movements

3.3.2 Token ID Burn Patterns

Analysis of burn frequency by token ID reveals interesting position lifecycle patterns. Figure 4 shows the distribution of burn counts across NFT positions, where some positions are burned multiple times, indicating either:

1. Position recreation cycles (mint-burn-mint patterns)
2. Multiple liquidity adjustments through complete position closure
3. Position migration strategies across different price ranges

Token ID Burn Count (2039 unique positions)

#	Token ID (NFT Position)	Burn Count	First Burn Time	Last Burn Time	Details
1	1097347	20	06/10/2025, 20:50:47	06/10/2025, 23:04:23	View
2	1096121	9	03/10/2025, 00:04:11	03/10/2025, 00:50:59	View
3	1090866	6	25/09/2025, 10:25:11	25/09/2025, 12:45:11	View
4	1095453	4	07/10/2025, 00:46:59	07/10/2025, 00:58:35	View
5	1097951	4	07/10/2025, 01:36:47	07/10/2025, 01:38:35	View
6	996443	3	06/10/2025, 23:24:23	06/10/2025, 23:24:47	View
7	1069180	3	07/10/2025, 07:57:59	07/10/2025, 08:01:23	View
8	1085725	3	05/10/2025, 02:29:11	05/10/2025, 02:33:23	View
9	1092393	3	29/09/2025, 03:37:35	29/09/2025, 03:53:11	View

Figure 4: Token ID Burn Count distribution showing how many times each NFT position was burned during the analysis period

3.3.3 Burner Address Analysis

The concentration of burn activity among burner addresses reveals distinct behavior patterns. Figure 5 displays the most active position closers, showing that the top burners are responsible for a disproportionate share of total burn events.

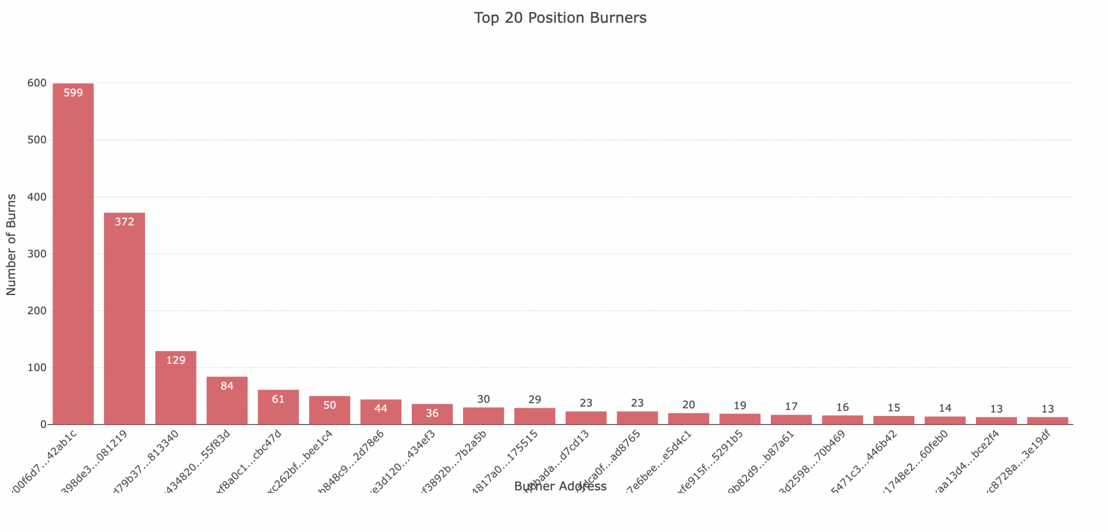

Figure 5: Top wallet addresses by burn count, revealing concentrated position closure activity among a small group of addresses

3.4 Position Distribution Analysis

The processed positions reveal interesting patterns in liquidity provision. Sample position data structure:

Listing 3: Sample Position Data Structure

```
{
  "tokenId": "1096180",
  "token0": {
    "address": "0x57e114b691db790c35207b2e685d4a43181e6061",
    "symbol": "ENA",
    "decimals": 18
  },
  "token1": {
    "address": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2",
    "symbol": "WETH",
    "decimals": 18
  },
  "liquidity": "430522056564517381413",
  "fee": "3000",
  "ticks": {
    "lower": -89520,
    "upper": -88260
  },
  "price_band": {
    "lower": 0.00012953590846246025,
```

```
      "upper": 0.00014692934552522227
    },
    "block": "23493513",
    "timestamp": "2025-10-02T23:56:23Z"
}
```


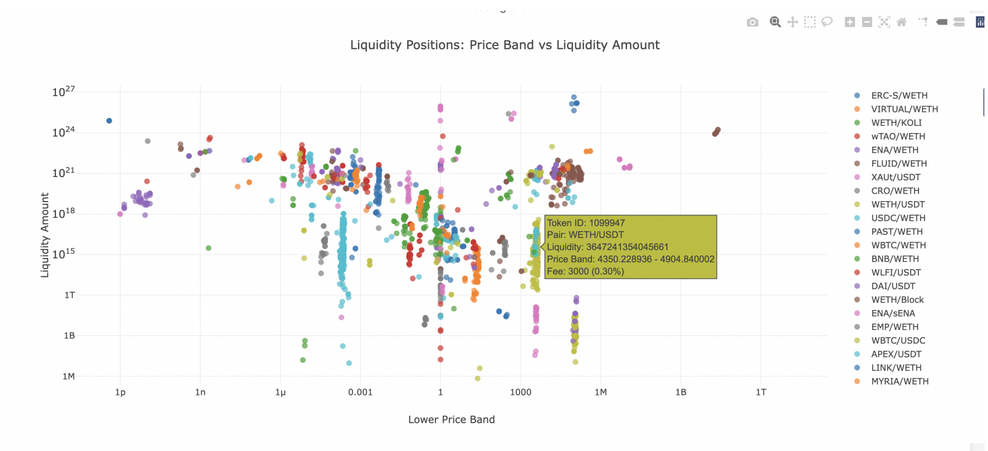

Figure 6: Liquidity position distribution analysis showing price band concentration patterns

3.5 Price Band Concentration Patterns

The analysis reveals that liquidity providers tend to concentrate their positions around current market prices, with tighter price bands for more volatile pairs. The tick-based system allows for precise control over price ranges, with positions often spanning 100-1000 ticks.

Some pair-wise examples:

- **ENA/WETH Pair**: Price bands concentrated around 0.00013-0.00015 WETH per ENA
- **CRO/WETH Pair**: Wider price bands (4.69e-05 to 5.51e-05 WETH per CRO)
- **Fee Tier Distribution**: 0.3% fee tier dominates with 51.5% of positions, followed by 0.05% tier (18.8%), 1% tier (19.2%), and 0.01% tier (10.5%)

3.6 Historical Liquidity Addition Patterns and Price Action Correlation

Analysis of mint events reveals temporal patterns in liquidity provision that correlate with price action:

1. **Peak Activity Periods**: Liquidity additions spike during market volatility, suggesting LPs respond to price movements by adjusting positions
2. **Fee Tier Preferences**: The 1% fee tier accounts for 28.6% of mint events, attracting LPs expecting higher volatility, while the dominant 0.3% tier (39.5%) indicates preference for balanced risk-reward profiles
3. **Token Pair Popularity**: WETH pairs dominate, representing 55.5% of all mint events, reflecting ETH's role as the primary trading pair

4. **Price Action Correlation**: Position concentration patterns around current prices suggest LPs anticipate price stability or mean reversion

3.7 Market Sentiment Indicators from Position Data

The analysis reveals that historical liquidity positions can serve as market sentiment indicators:

Price Support/Resistance Levels:

- High liquidity concentration at specific price bands indicates potential support/resistance levels
- Tight position bands suggest LPs expect price to remain within certain ranges
- Wide position spreads indicate uncertainty or preparation for significant price movements

Market Confidence Indicators:

- Increased position creation during market downturns suggests confidence in price recovery
- Concentration in lower fee tiers (0.05%) indicates expectation of stable trading conditions
- Migration to higher fee tiers (1%) suggests anticipation of increased volatility

4 Discussion

4.1 Limitations and Challenges

Several limitations were encountered during the analysis:

1. **Missing Position Data**: Some positions may not be captured due to contract interactions
2. **Computational Complexity**: Large-scale position processing requires significant computational resources

4.2 Implications for DeFi Research

This methodology opens new avenues for DeFi research:

1. **Liquidity Provider Behavior**: Understanding capital allocation strategies
2. **Market Efficiency**: Analyzing price discovery mechanisms
3. **Risk Management**: Position concentration and risk assessment
4. **Protocol Design**: Insights for future AMM improvements

5 Conclusion

This paper demonstrates that historical liquidity positions in Uniswap V3 can provide valuable insights into liquidity provider behavior and price action patterns. Through comprehensive analysis of position calls (5,711 positions), mint events (11,315 mints from 3,541 creators), and burn events (2,197 burns from 344 burners), we have shown that position concentration patterns, fee tier preferences, and temporal distribution of liquidity additions and closures correlate with market dynamics and can serve as indicators for price action prediction.

Key Findings:

- Liquidity providers consistently concentrate positions around current market prices, creating natural support/resistance levels
- New liquidity providers can use this data as a reliable metric for current AMM conditions
- Fee tier selection reflects market expectations, with higher tiers indicating anticipation of increased volatility
- The burn-to-mint ratio of 0.19:1 during the analysis period indicates a strong liquidity expansion environment with significantly more position creation than closure, suggesting growing market participation and capital inflow into Uniswap V3 pools
- Burn events are highly concentrated among a small group of liquidity managers (avg. 6.39 burns per address), while position creation shows broader participation (avg. 3.20 positions per creator), indicating sophisticated position rebalancing strategies by professionals alongside broader market participation
- Historical position data can serve as predictive indicators for future price action and market sentiment

Future Work:

1. **Extended Time Series Analysis**: Longer-term historical analysis of position evolution
2. **Cross-Protocol Comparison**: Analysis of liquidity patterns across different AMMs
3. **Machine Learning Integration**: Predictive modeling of liquidity provider behavior
4. **Real-time Monitoring**: Build tools for live position tracking systems

The methodology and insights presented in this paper provide a foundation for future research in decentralized finance and automated market maker analysis.

Acknowledgments

The author thanks the Bitquery team for providing access to their APIs and the Uniswap community for their continued development of decentralized trading infrastructure.

References

References

[1] Adams, H., Zinsmeister, N., Salem, M., Keefer, R., & Robinson, D. (2021). Uniswap v3 Core. *Uniswap Labs*.

[2] Bitquery. (2025). Uniswap API Documentation. `https://docs.bitquery.io/docs/blockchain/Ethereum/`.

[3] Uniswap Labs. (2021). Uniswap V3 Whitepaper. *Uniswap.org*.