Mapping Public and Private JIT Liquidity Dynamics in Ethereum's Builder Ecosystem

Koshi Ota[1] and Davide Rezzoli[2]

[1] The University of Tokyo
cassiopeia01620@gmail.com
[2] PBS foundation
davide.rezzoli@gmail.com

Abstract. Just-In-Time (JIT) liquidity, defined as ephemeral liquidity provision centered around a trade within a single block, occurs both in the public mempool and via private relays to specialized builders. We present the first comprehensive empirical analysis of JIT liquidity under Proposer-Builder Separation (PBS), quantifying how realized JIT profits are reflected in fees paid to builders at the builder-searcher-pair level. To detect JIT activity, we use Uniswap v3 event logs, the principal concentrated-liquidity automated market maker (AMM) on Ethereum, which allow precise reconstruction of Mint $\rightarrow$ Swap $\rightarrow$ Burn sequences within a block. Using these logs, we identify JIT LPing on Ethereum mainnet from January 2024 to September 2025, attribute bundles to builders and searchers, and classify execution channels. Our dataset comprises approximately 442,000 JIT bundles, including about 52,700 private bundles. Among the top builder-searcher pairs, public JIT bundles exhibit higher average total fees, while private bundles are more concentrated and have lower fees per bundle. These patterns motivate us to examine why fees paid to builders differ across searchers and whether heterogeneity in markouts is associated with cross-searcher variation. We decompose markouts into a fee component and a price impact component and estimate OLS regressions with total fees as the dependent variable. On public routes, both components are positively and significantly associated with higher total fees. Fees on private channels are less sensitive to realized profits, consistent with weaker bidding pressure or partially exclusive order flow in private channels.

Keywords: Automated market maker · Uniswap v3 · JIT liquidity · Public mempool · Private order flow

1 Introduction

Automated Market Makers (AMMs) are core primitives of decentralized finance, enabling asset exchange without order books. Liquidity is supplied by third parties so called liquidity providers (LPs) who deposit tokens into pools and earn fees from trader activity. The transparency of AMMs and the order-dependence of transaction execution on blockchains have also made them a natural locus for Maximal Extractable Value (MEV) strategies.

Among these strategies, Just-In-Time (JIT) liquidity denotes the temporary provision of liquidity around a trade within a single block: a position is added immediately before a swap to capture fees and price impact, and removed shortly thereafter. While early discussion touched on Uniswap v2, the practice was popularized with Uniswap v3s concentrated-liquidity design, which allows targeted, short-lived liquidity provision.

Since ethereum's move to proposer-builder separation (PBS) with MEV-Boost, searchers increasingly submit bundles directly to specialized builders via private relays, alongside continued use of the public mempool. As a result, JIT execution today occurs across two channels: (i) *public* broadcasts observable in the mempool and (ii) *private* submissions to builders via relay or API pathways that reduce information leakage.

This shift motivates several empirical questions: Which builders and searchers account for most JIT activity in each channel? How concentrated are these interactions, and how does concentration differ between public and private execution? How do fee outcomes compare across channels and buildersearcher pairs?

Contributions

This paper makes three main contributions:

(i) **Comprehensive JIT detection and buildersearcher identification across public and private routes** We develop a scalable framework to detect JIT windows on Uniswap v3 (Mint $\rightarrow$ Swap(s) $\rightarrow$ Burn within a block), attribute events to builders and searchers, and classify execution channels (public mempool vs. private relays).

(ii) **Public–private execution landscape.** We jointly characterize *public* and *private* JIT execution at scale ($\sim$442,000 bundles on Mainnet, of which $\sim$52,700 private), including concentration across major builder–searcher pairs. Among top pairs (Ratio $> 1\%$), public accounts for 90.07% of occurrences and private for 9.93%; within private, SCP originates 100% of bundles and 78.04% of those route to Beaver.

(iii) **Empirical explanation of priority-fee heterogeneity:** We decompose realized markouts into a *fee income* component and a *price impact* component, and estimate OLS regressions with total fees (priority fees plus coinbase transfers) as the dependent variable. The specification includes a private-route indicator and interactions that allow profit-to-fee pass-through to differ by channel, as well as a dummy for known searcher–builder integration, controlling for trade size, JIT-provided liquidity, and token-pair and fee-tier fixed effects. On public routes, both components are strongly and significantly associated with higher total fees (pass-through ≈ 0.75). On private routes, pass-through is significantly weaker for price impact(implied slope ≈ 0.32), while attenuation for fee income is not statistically significant; the private-channel level effect is small and statistically insignificant. Integrated pairs remit higher observed total fees conditional on realized profits and fixed effects.

By quantifying participation, concentration, and pass-through across builders and channels, we provide the first comprehensive empirical characterization of JIT liquidity across public and private builder routes under PBS, without taking a stance on desirability or mechanism design.

Organization. The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the data and identification strategy. Section 4 presents combined public/private findings. Section 5 details private-channel results. Section 6 *decomposes markouts and estimates profit-to-fee pass-through*, comparing public and private routes and the role of integration. Section 7 concludes.

2 Related Work

JIT liquidity. Uniswap Labs researchers formalized *just-in-time* (JIT) liquidity on Uniswap, documenting within-block Mint → Swap(s) → Burn patterns and their microeconomic drivers [11, 12]. Subsequent analyses emphasize that JIT is targeted and short-lived rather than a general LP strategy.

Public JIT and MEV context. Foundational MEV work shows how ordering dependencies create extractable opportunities in DEXs, situating public JIT alongside sandwiching and backrunning [3]. Open tooling later enabled measurement of such behaviors across pools.

PBS and the builder ecosystem. With MEV-Boost, Ethereum adopted an opt-in form of proposer-builder separation (PBS), mediating block construction via relays and specialized builders [5]. Empirical studies document concentration among builders/relays and characterize distributional effects under PBS [1,6,14].

Private and exclusive order flow. Recent theory and measurement analyze *private* (multiplexed) and *exclusive* (single-destination) order-flow arrangements, showing how private flow alters bidding, surplus splits, and scale advantages in builder markets [2, 9, 13].

Positioning. Prior research either characterizes JIT at the pool/transaction level or studies PBS market structure and private flow. We connect the two by mapping *public and private* JIT across builders and searchers, quantifying participation shares and fee outcomes at the pair level.

3 Data Collection

Scope. We study JIT liquidity on Ethereum from *January 2024* to *September 2025*, focusing on Uniswap v3. A *JIT event* is an LP round trip that occurs within a single block and coincides with at least one swap in the same pool between the entry and exit.

Definition (JIT window). A JIT window is the ordered sequence

$$\text{Mint (add)} \;\rightarrow\; \text{Swap(s)} \;\rightarrow\; \text{Burn (remove)}$$

such that all transactions occur in the same block and pool and are attributable to the same LP *owner*.

3.1 Data Source

Platform: Dune Analytics
Tables:

```
uniswap_v3_ethereum.uniswapv3pool_evt_swap
uniswap_v3_ethereum.uniswapv3pool_evt_mint
uniswap_v3_ethereum.uniswapv3pool_evt_burn
```

These tables store all logs (Mint/Burn/Swap) emitted by each Uniswap v3 pool.

Required fields.

All events:
`event_type`, `block_number`, `block_timestamp`, `transaction_hash`, `transaction_index`, `log_index`, `liquidity_pool_address`, `owner`.
Mint/Burn:
`tick_lower`, `tick_upper`, `liquidity_delta`.
Swap:
`amount0`, `amount1`, `sqrt_price_x96`, `tick`.

3.2 Identification Rules

Candidate Mint–Burn pairs. We first form candidate pairs (m, b) satisfying:

(a) **Same block:** $m.\texttt{block\_number} = b.\texttt{block\_number}$.
(b) **Same pool:** $m.\texttt{liquidity\_pool\_address} = b.\texttt{liquidity\_pool\_address}$.
(c) **Same actor:** $m.\texttt{owner} = b.\texttt{owner}$.
(d) **Temporal order:** $m.\texttt{transaction\_index} < b.\texttt{transaction\_index}$.

Swap-between condition. A candidate (m, b) is classified as JIT if there exists at least one swap s s.t.:

(a) $s.\texttt{block\_number} = m.\texttt{block\_number}$,
(b) $s.\texttt{liquidity\_pool\_address} = m.\texttt{liquidity\_pool\_address}$,
(c) $(m.\texttt{tx\_idx}, m.\texttt{log\_idx}) < (s.\texttt{tx\_idx}, s.\texttt{log\_idx}) < (b.\texttt{tx\_idx}, b.\texttt{log\_idx})$ in lexicographic order.

If multiple swaps occur between m and b, we treat all as part of the JIT window.

3.3 Attribution and Channel Classification

Searcher attribution. We map `owner` addresses (Mint/Burn signer) to searcher labels using known ENS names, public attributions, and address dictionaries. We then normalize common aliases to short canonical names. For example:

- `jaredfromsubway.eth` → **jared**
- `Wintermute Trading` → **Wintermute**
- `SCP` → **SCP**
- `Senna` → **Senna**
- *unlabeled addresses* → `0x...` **(hex)**

Builder attribution. For each JIT window we attribute the builder of the containing block using block-level metadata (e.g., coinbase/extraData tags and relay-reported names). We then normalize common aliases. For example:

- `beaverbuild.org`→ **Beaver**
- `Titan (titanbuilder.xyz)` → **Titan**
- `rsync-builder.xyz` → **Rsync**
- `BuilderNet (Beaver)` → **Beaver *via BuilderNet***
- `BuilderNet (Flashbots)` → **Flashbots**
- `BuilderNet (Nethermind)` → **Nethermind**

Ambiguous or inconsistent tags are excluded from channel-level statistics.

Public vs. private. We classify the execution channel using the Flashbots *Mempool Dumpster* [4]. Mempool Dumpster is a dataset archiving transactions from the public mempool. It connects to multiple public mempools from multiple nodes simultaneously, collecting observable pending transactions as comprehensively as possible.[3] For each JIT window, we collect the hashes of the Mint, Swap(s), and Burn transactions and check *presence* in the Dumpster:

- **Public**: the swap hash is present in the Dumpster (seen at least once).
- **Private**: none of the window transaction hashes are present in the Dumpster (never seen).

The Mempool Dumpster is not exhaustive; there are periods when it is not operating normally due to system failures or other issues. During these periods, transactions that are actually observable in the public mempool are not recorded by the Dumpster. This creates a problem where such transactions are erroneously classified as private transactions. Since there are no official records regarding Mempool Dumpster downtime, this study statistically identified and excluded these periods based on the total transaction data using the following procedure. First, to prevent contamination of the baseline statistics, we excluded obvious anomalous periods where the Public Tx ratio was less than 1% from the

[3] Some publicly broadcast transactions may be missing, and thus both misclassification and underestimation of public order flow remain possible.

initial calculation. Next, we calculated the 30-day moving average and standard deviation (σ) of the Public Tx ratio for each point in time. Periods where the ratio deviated by more than -3σ were defined as "Mempool Dumpster down periods" and were excluded. Furthermore, regarding the period from February 17, 2025, 16:00 to February 26, 2025, 08:00, although it did not formally violate the statistical criterion (3σ), a visual inspection of the time series revealed a significant decline in the Public Tx ratio. We determined that data quality was degraded during this time and, to maintain the robustness of our analysis, we conservatively excluded this period from our sample.

3.4 Fee Variables

In this study, for each JIT bundle, we define the gas priority fee (tip) and the direct payment to the builder's `coinbase` address as $JIT\ bundle_{\text{tip}}$ and $JIT\ bundle_{\text{coinbase}}$, respectively. Here, the gas priority fee $JIT\ bundle_{\text{tip}}$ is calculated as the sum of the priority fees specified in the mint and burn transactions that constitute the JIT bundle. In contrast, $JIT\ bundle_{\text{coinbase}}$ represents the on-chain payments to the builder's `coinbase` address, which in most cases are made via the burn transaction within the JIT bundle. We define the sum of these two components as the total fee and use it as an indicator for estimating the value embedded in a JIT bundle. It should be noted, however, that the total fee represents only the best available estimate based on on-chain data. A more robust assessment would require taking into account searchers' and builders' private valuations, as well as additional off-chain settlement arrangements.

$$JIT\ bundle_{\text{total}} = JIT\ bundle_{\text{tip}} + JIT\ bundle_{\text{coinbase}}.$$

3.5 Markouts

Markouts capture the informativeness of order flow and the hypothetical profitability for liquidity providers by comparing its execution price to a benchmark price observed after a fixed horizon. We adopt the five-minute time horizon commonly used. Let t_s denote the execution time of swap s, and $\Delta_m > 0$ the markout horizon. For each swap s, we denote by $q^I(s)$ and $q^O(s)$ the token-in and token-out amounts:

$$\begin{aligned} q^I(s) &:= \text{amount of token that swap } s \text{ sends into the pool,} \\ q^O(s) &:= \text{amount of token that swap } s \text{ receives from the pool,} \\ \tilde{q}^O(s) &:= \frac{q^O(s)}{1-\text{fee}}\,[4] \end{aligned}$$

Write $p^{\text{binance}}_{t_s+\Delta_m}$ for the binance mid-price at time $t_s + \Delta_m$. To make buy/sell directions commensurate, introduce the sign variable $D_s \in \{+1, -1\}$ so that

[4] In Uniswap V3, trading fees are applied to the input token, so the output token amount that is typically observed is already net of fees. To explicitly compute the fee term, we therefore define a hypothetical output token amount that excludes fees.

$D_s = +1$ for swaps that *sell* the base asset (e.g., WETH) and $D_s = -1$ for swaps that *buy* the base asset.[5]

Define the (per-unit-price) markout for swap s by

$$\text{Markout}_s = D_s \left|q^I(s)\right| \left(\left| \frac{q^O(s)}{q^I(s)} \right| - p^{\text{binance}}_{t_s+\Delta_m} \right)$$

To isolate the components of JIT LP revenues, we decompose the markouts into a *price impact* term and a *fee* term, following [10].

$$\begin{aligned}
&\text{Markout}_s = \text{Fee}_s + \text{PriceImpact}_s, \\
&\text{Fee}_s = D_s \left|q^{\text{I}}(s)\right| \left(\left| \frac{q^{\text{O}}(s)}{q^{\text{I}}(s)} \right| - \left| \frac{\tilde{q}^{\text{O}}(s)}{q^{\text{I}}(s)} \right| \right), \\
&\text{PriceImpact}_s = D_s \left|q^{\text{I}}(s)\right| \left(\left| \frac{\tilde{q}^{\text{O}}(s)}{q^{\text{I}}(s)} \right| - p^{\text{binance}}_{t_s+\Delta_m} \right).
\end{aligned}$$

Due to listing constraints on Binance, it is not possible to obtain sufficient price information for all tokens. We therefore restrict attention to major listed token pairs[6] and compute markouts only for those. As a result, some long-tail token pairs are excluded from the sample. It should be noted that certain searchers may be particularly skilled at constructing JIT bundles in long-tail assets, and our analysis does not fully capture that comparative advantage.

4 JIT Bundles Findings

We analyze over 442,000 JIT liquidity bundles overall. Table 1 reports the 14 (builder, searcher) pairs with Ratio $> 1\%$, totaling 353,883 bundles within this filtered set. Of these, 318,753 (90.07%) were executed publicly and 35,130 (9.93%) privately, spanning 2 builders and 1 searcher. Public activity is led by Beaver-jared (153,042 occurrences; 34.55% of all bundles), while private activity is concentrated in Beaver-SCP (26,807; 6.06%). Within the filtered set, jared accounts for 244,265 bundles (69.02%), with no private JIT bundles, whereas SCP makes up 100% of private bundles. The combined dataset in Tables 1 and 2 highlights clear structural and economic differences between privately and publicly executed JIT liquidity bundles. The tables represent all JIT bundle pairs between searchers and builders executed either privately or publicly.

From a structural standpoint, public JIT activity dominates in absolute frequency. The top searcher, *jared*, leads the dataset with the highest public pair, *Beaver-jared*, which accounts for over 153,000 occurrences nearly six times more than the largest private pair. In aggregate, public JIT interactions represent

[5] Throughout, prices are expressed with a fixed orientation (e.g., token0 per token1).

[6] USDC-WETH, USDT-WETH, USDC-WBTC, USDT-WBTC, WBTC-WETH, UNI-WETH, AAVE-WETH, LINK-USDC, LINK-USDT, AAVE-USDC, AAVE-USDT, UNI-USDT, UNI-USDC

Type	Builder	Searcher	Occurrences #	Ratio%
Public	Beaver	jared	153,042	34.55
Public	Titan	jared	62,165	14.03
Public	Beaver	SCP	29,573	6.68
Private	Beaver	SCP	26,807	6.06
Public	Rsync	Wintermute	20,746	4.68
Public	Rsync	jared	11,200	2.53
Public	Beaver	Senna	8,447	1.91
Private	Titan	SCP	8,323	1.88
Public	Flashbots	jared	6,506	1.47
Public	Beaver via BuilderNet	jared	6,090	1.37
Public	Titan	SCP	5,440	1.23
Public	Nethermind	jared	5,262	1.19
Public	Beaver	0x00...51d3	5,186	1.17
Public	Titan	0xe8...a2e5	5,096	1.15

Table 1. Top buildersearcher pairs by share of private and public JIT bundles.

Builder	Searcher	Average Total Fee (USD)	Median Total Fee (USD)
Beaver	jared	238.37	58.02
Titan	jared	61.04	19.43
Beaver	SCP	64.16	14.40
Beaver	SCP (Private)	25.02	6.21
Rsync	Wintermute	62.53	26.56
Rsync	jared	112.28	37.85
Beaver	Senna	36.77	8.51
Titan	SCP (Private)	1.26	1.12
Flashbots	jared	118.84	12.98
Beaver via BuilderNet	jared	82.67	12.00
Titan	SCP	1.02	1.05
Nethermind	jared	105.34	13.34
Beaver	0x00...51d3	11.18	0.34
Titan	0xe8...a2e5	93.63	24.39

Table 2. Average and median total fees for both private and public JIT bundles.

the majority of observed events, indicating that the public mempool hosts substantial opportunistic liquidity provision. Private JIT bundles, by contrast, are far fewer in number. The leading private pair *Beaver-SCP* accounts for 26,807 bundles. A pattern emerges across domains: in public data, *jared* appears more prominent in both frequency and typical fees, whereas in private execution *SCP* has a larger share of bundles. This difference may reflect access to private relay flow or coordination.

Fee distributions suggest differences in typical per-bundle fees. Public bundles tend to show higher averages and medians in several prominent pairsfor example, *Beaver-jared* reports an average total fee of $\sim$ 238.37 USD while several private pairs are lower. This indicates that public JIT opportunities, while competitive, can correspond to higher per-transaction fees when successful a better fee analysis is made in section 6

SCP appears in both public and private contexts, suggesting hybrid strategies. SCP's public interactions with *Beaver* (average fee 64.16 USD) generate higher returns than its private ones (average fee 25.02 USD).

At the block level, JIT activity also differs markedly across mempool types. The average JIT bundles per block is 0.0853 (i.e., 8.53 bundles per 100 blocks) for public transactions and only 0.0115 for private ones[7]. This discrepancy shows that public JIT provision remains a frequent phenomenon in open blockspace, whereas private JIT execution though less common constitutes a smaller but strategically significant share of total block composition.

Looking across builders, the distribution of total fees further highlights economic asymmetry within the JIT ecosystem. *Beaver* consistently captures the largest share of fee volume, both publicly and privately, driven primarily by its connections with *jared* and *SCP*. When weighting by each pairs ratio, Beaver's aggregated mean total fee contribution exceeds that of *Titan* by roughly a factor of four. This difference, however, does not necessarily indicate lower builder efficiency. Instead, it reflects strategic behavior by searchers, who tend to offer smaller tips to Titan possibly due to deal structures or integration pathways involving other builders.

Overall, the data reveal a distinct segmentation of the JIT landscape. Public bundles dominate in both count and fee magnitude, driven by open competition where the top-performing searchers are *jared* followed by *SCP*. Private bundles, on the other hand, form a smaller and more controlled domain in which select searchers leverage privileged access or coordination. While the same top actors persist across both domains, their relative positions invert: *SCP* leads in private coordination, whereas *jared* maintains dominance in the public mempool.

5 JIT Private Bundles

Builder	**Searcher**	**Occurrences #**	**Ratio%**
Beaver	SCP	26,807	50.79
Titan	SCP	8323	15.77
Flashbots	SCP	3686	6.98
Beaver via BuilderNet	SCP	3637	6.89
Nethermind	SCP	3046	5.77
Flashbots (Not BuilderNet)	0x4a...0a78	1431	2.76

Table 3. Counts and Ratio for major builder–searcher pairs.

[7] We measure how often JIT liquidity appears by counting the number of JIT bundles in our dataset and dividing by the number of L1 blocks over the study window (Jan 1, 2024Sep 30, 2025) [8]

Table 3 includes private JIT only builder–searcher pairs with a ratio greater than 1%. Interactions below this threshold were excluded to focus on the most statistically significant and recurrent private JIT relationships. The full dataset can be explored on our public dashboard [7]. Between January 2024 and September 2025, we identified a total of approximately 52,700 privately executed JIT liquidity bundles across a small number of recurring builder-searcher pairs. All bundles in this sample are classified as private, in the sense that none of their transactions appear in the Mempool Dumpster dataset; they are therefore consistent with direct submission to builders, bypassing the public mempool.

Searcher-to-Builder Distribution

The majority of JIT activity originated from SCP, which accounted for five of the top ten builder–searcher combinations by volume. In total, SCP sent over 45,400 private JIT bundles during this timeframe. Its largest destination, *Beaver*, alone received 50.8% of all observed JIT bundles-three times more than any other builder. The following clusters *Titan*, and *BuilderNet (Flashbots, Beaver, Nethermind)* each accounted for between 5–15% of total JIT activity. This consistent multi-builder routing pattern suggests that SCP maintains parallel private channels across several major builder infrastructures, with a strong preference toward Beaver.

Contrary to public JIT, we find that jared does not perform JIT on private transactions. Instead, the remaining JIT bundles are distributed across a long tail of smaller actors.

6 Priority Fee Heterogeneity

In this section, we study why the level of total fees, defined as priority fees plus direct transfers to builders, differs across searchers in JIT liquidity provision. We proceed in two steps. First, we provide descriptive evidence based on the combinations in Table 4. Second, we estimate OLS with the total fee as the dependent variable, regressing it on swap fees, the price impact component, an indicator for private mempool routing, and searcherbuilder integration dummies.

We examine how cross-sectional variation in total fees is related to (i) fee driven component, (ii) price impact-driven component, (iii) the use of private routes, and (iv) known searcher–builder integration. Given that only executed bundles are observable, we do not claim to identify structural bidding strategies or causal effects.

6.1 Descriptive Patterns by Mempool, Builder, and Searcher

Table 4 reports summary statistics for selected (mempool, builder, searcher) combinations. For each combination, we show the mean and median of: (i) total fee in USD, (ii) fee income component, (iii) price-impact component, and (iv) overall markout.

Three descriptive implications emerge from Table 4:

Mempool	Builder	Searcher	Total fee (USD)		Swap Fee (USD)		Price impact		Markout	
			avg	med	avg	med	avg	med	avg	med
Public	Beaver	jared	238.37	58.02	80.70	27.24	391.59	42.90	449.73	60.36
Public	Titan	jared	61.04	19.43	40.05	12.51	36.05	5.48	48.40	10.65
Public	Beaver	SCP	64.16	14.40	82.22	26.89	8.28	0.35	88.92	17.66
Private	Beaver	SCP	25.02	6.21	41.65	10.16	−4.84	−0.41	36.76	5.87
Public	Rsync	Wintermute	62.53	26.56	72.01	33.43	22.30	9.11	84.80	36.03
Public	Rsync	jared	112.28	37.85	75.52	33.44	356.71	45.19	418.92	61.45
Public	Beaver	Senna	36.77	8.51	20.06	5.49	5.98	0.64	18.07	3.42
Private	Titan	SCP	1.26	1.12	20.59	7.61	−4.22	0.04	15.00	6.38
Public	Flashbots (BuilderNet)	jared	118.84	12.98	18.88	4.00	161.31	16.17	185.91	26.01
Public	Beaver (BuilderNet)	jared	82.67	12.00	15.96	3.87	128.93	19.83	143.31	27.13

Table 4. Key metrics by mempool, builder, and searcher. The table reports mean and median total fees (USD), fee income, price-impact component, and markout for selected triplets.

1. **Heterogeneity across searchers within the same builder.** For instance, on Public–Beaver–jared exhibits much higher average and median total fees than SCP. This aligns with larger price-impact and markout components, suggesting that searchers with stronger extraction capabilities (especially on the price-impact side) also tend to pay higher bribes.
2. **Private routes are associated with lower total fees relative to realized profits.** Comparing Public–Beaver–SCP and Private–Beaver–SCP, the private configuration shows lower total fees despite sizable fee income. Similar patterns appear in other private configurations, implying that private routing may reduce competitive intensity in bribe setting.
3. **Fee vs. price-impact contributions differ systematically.** Across many cells, fee income accounts for a large and relatively stable share of JIT profits, while the price-impact component is more volatile and distinguishes "aggressive" searchers (e.g., Public–Beaver–jared). This suggests that heterogeneity in total fees is closely tied to heterogeneity in the price-impact component, while fee income behaves more like a predictable baseline.

These patterns motivate a more empirical analysis. The cross-cell comparisons in Table 4 are informative but rely on selected combinations and do not control for overlapping effects of searchers, builders, pool characteristics, or mempool types. We therefore turn to a regression framework that (i) treats the total fee as the dependent variable and includes the swap fee and the price-impact component of markout as explanatory variables, (ii) compares public and private routes, and (iii) includes searcherbuilder integration dummies.

6.2 Econometric Specification

We estimate an OLS model to quantify how payments to builders vary with realized JIT profitability and execution channels. Each observation is indexed by searcher i, builder b, and event (bundle) t. The key variables are:

- b_{ibt}: total fee paid to the builder in USD (priority fees plus direct transfers to coinbase),
- Price Impact$_{ibt}$: price impact component of realized markout (USD),

– Fee_{ibt}: fee income component of realized markout (USD),
– $\text{Private}_{ibt} \in \{0, 1\}$: indicator for private-route execution,
– Integrated Pair_{ib}: indicator for known searcher–builder integration,
– controls: $\log(\text{amount0 usd}_{ibt})$, the logarithm of the swap's token0 value in USD and $\log(\text{liquidity amount usd}_{ibt})$, the logarithm of the USD value of JIT liquidity supplied, and token-pair and/or fee-tier fixed effects.

Our baseline specification is:

$$\begin{aligned} b_{ibt} = {} & \gamma\, \text{Private}_{ibt} + \beta^P_{\text{pub}}\, \text{Price Impact}_{ibt} + \beta^F_{\text{pub}}\, \text{Fee}_{ibt} \\ & + \left(\beta^P_{\text{priv}} - \beta^P_{\text{pub}}\right) \text{Private}_{ibt} \times \text{Price Impact}_{ibt} + \left(\beta^F_{\text{priv}} - \beta^F_{\text{pub}}\right) \text{Private}_{ibt} \times \text{Fee}_{ibt} \\ & + \theta\, \text{Integrated Pair}_{ib} + \delta_1 \log(\text{amount0 usd}_{ibt}) \\ & + \delta_2 \log(\text{liquidity amount usd}_{ibt}) + \text{FEs} + \varepsilon_{ibt}. \end{aligned} \tag{1}$$

The coefficients β^P_{pub} and β^F_{pub} capture how total fees respond to realized markout components on public routes ($\text{Private}_{ibt} = 0$). The corresponding marginal responses on private routes are

$$\beta^P_{\text{priv}} = \beta^P_{\text{pub}} + \left(\beta^P_{\text{priv}} - \beta^P_{\text{pub}}\right), \qquad \beta^F_{\text{priv}} = \beta^F_{\text{pub}} + \left(\beta^F_{\text{priv}} - \beta^F_{\text{pub}}\right).$$

The parameter γ captures any residual level shift for private bundles after conditioning on realized profits, integration, controls, and fixed effects. We report cluster-robust standard errors.

6.3 Regression Results: Profit Sharing and Private Routes

Table 5 reports estimates of (1) under alternative fixed-effect choices. We focus on column (4), which includes both token-pair and fee-tier fixed effects.

On public routes ($\text{Private}_{ibt} = 0$), both components of realized markout are positively and strongly associated with total fees. In column (4), the coefficient on PriceImpact_{ibt} is 0.7511 (significant at the 1% level) and the coefficient on Fee_{ibt} is 0.7464 (significant at the 1% level). Interpreted literally, a \$1 increase in realized price impact is associated with approximately \$0.75 higher total payments to builders on public routes, and a \$1 increase in fee income is associated with approximately \$0.75 higher builder payments. This pattern is consistent with profit-sharing: higher realized surplus is accompanied by higher payments to builders.

Attenuated pass-through on private routes, concentrated in price impact. The private-route interaction with the price impact component is negative and statistically significant. In column (4), $\text{Private}_{ibt} \times \text{Price Impact}_{ibt}$ has coefficient -0.4303 (1% level), implying a materially flatter profit-to-fee slope on private routes:

$$\widehat{\beta}^P_{\text{priv}} = 0.7511 - 0.4303 \approx 0.3208.$$

Thus, relative to public routes, private routes exhibit substantially lower sensitivity of total fees to incremental price impact.

Table 5. Determinants of Total Fees

	(1) Total fee (USD)	(2) Total fee (USD)	(3) Total fee (USD)	(4) Total fee (USD)
Intercept	276.8903	239.4131	229.8082*	104.2175***
	(175.334)	(158.550)	(133.126)	(30.651)
private	-3.7474	-9.6759***	-0.7768	-1.3334
	(8.923)	(3.392)	(9.866)	(7.235)
price impact	0.7480***	0.7504***	0.7482***	0.7511***
	(0.047)	(0.049)	(0.048)	(0.049)
fee	0.8045***	0.7563***	0.8052***	0.7464***
	(0.246)	(0.211)	(0.248)	(0.209)
private × price impact	-0.4798***	-0.4578***	-0.4686***	-0.4303***
	(0.045)	(0.028)	(0.039)	(0.024)
private × fee	-0.1793	-0.2437	-0.2011	-0.2680
	(0.174)	(0.242)	(0.195)	(0.259)
integrated pair	49.3258**	42.2026**	52.3848**	46.6561**
	(24.220)	(20.769)	(24.203)	(20.932)
log amount0 usd	-17.2816	-13.8627	-17.2087	-13.3684
	(14.966)	(12.983)	(14.552)	(12.073)
log liquidity amount usd	-9.0438**	-9.4663**	-9.9680***	-13.7432***
	(3.860)	(4.396)	(3.501)	(5.251)
Token-pair FEs	No	No	Yes	Yes
Fee-tier FEs	No	Yes	No	Yes
Adj. R^2	0.515	0.516	0.515	0.517
Observations	106,677	106,677	106,677	106,677

Notes: Dependent variable is total_fee_usd. Standard errors are clustered at the searcher level. $^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$.

For the fee income component, the interaction $\text{Private}_{ibt} \times \text{Fee}_{ibt}$ is also negative in column (4) (-0.2680), implying a private-route slope of

$$\widehat{\beta}^{F}_{\text{priv}} = 0.7464 - 0.2680 \approx 0.4784,$$

but the interaction term is not statistically significant. Accordingly, while the point estimate suggests weaker fee-income pass-through on private routes, the data do not allow us to reject equality of fee income slopes across channels in this specification.

Level effect of private routing. The coefficient on Private_{ibt} itself is small and statistically insignificant in column (4) (-1.3334). Notably, the private indicator is negative and significant in column (2) (which includes fee-tier fixed effects but not token-pair fixed effects), but becomes close to zero and insignificant once token-pair fixed effects are included (columns (3)–(4)). This pattern suggests that apparent level differences in fees between public and private routes partly reflect differences in token-pair composition rather than a uniform private-route discount.

Integration and controls. The coefficient on Integrated Pair_{ib} is positive and statistically significant in column (4) (46.6561, 5% level), indicating that (known) integrated pairs remit higher observed total fees, conditional on realized markouts, controls, and fixed effects.

Among controls, log(liquidity amount usd_{ibt}) enters negatively and is statistically significant in column (4) (-13.7432, 1% level), whereas log(amount0 usd_{ibt}) is not statistically significant.

6.4 Discussion

Combining the descriptive patterns in Table 4 with the regression evidence in Table 5 yields two main takeaways.

Total fees are highly correlated with realized JIT profits on public routes. On public routes, both the fee income and price impact components are positively and significantly reflected in total payments to builders. The estimates in Table 5 imply roughly three-quarters pass-through of realized surplus into builder payments. This is consistent with competitive fee setting in public routing, where successful bundles that realize higher surplus tend to pay higher total fees.

Private routes are less sensitive to incremental profits, especially price impact. Private routing is associated with a materially smaller pass-through from the price impact component into builder payments: the implied slope falls from about 0.75 on public routes to about 0.32 on private routes, and the difference is statistically significant. For the fee income component, the point estimate also suggests attenuation (from about 0.75 to about 0.48), but the difference is not statistically significant in the baseline specification. Overall, the channel distinction appears primarily as a change in the profit-to-fee slope, rather than as a uniform level shift in total fees once fixed effects are included.

These conclusions are subject to important limitations. We observe only executed (winning) bundles; losing bids and participation decisions are unobserved, so the coefficients are conditional on selection. Profit measures are ex-post realizations and may be noisy proxies for the ex-ante expected profits that drive bidding. Integration and routing indicators are constructed from observable on-chain relationships and cannot capture off-chain rebates or informal settlement arrangements. Accordingly, we interpret the results as a structured descriptive account of how realized profits, routing choices, and integration correlate with builder payments, rather than as a fully identified structural model of bidding behavior or market power.

7 Conclusion

Our analysis shows that JIT liquidity is highly concentrated among a small set of searchers and builders. This structure can improve instantaneous execution

and reduce slippage for trades routed through private channels, but it also has clear distributional consequences across market participants.

From a microeconomic perspective, JIT liquidity operates as an optimized, event-driven form of market making: liquidity is injected precisely for the duration of profitable trades, minimizing adverse selection risk and maximizing fee capture per unit of capital. The flip side is that passive LPs are left with the residual risk. Because JIT LPs arrive immediately before large swaps and exit right after, they capture a disproportionate share of fees while passive LPs remain exposed to adverse selection with diminished compensation. In this sense, JIT behaves like fee sniping against honest LPs who sustain baseline depth.

At a systemic level, the privatization of JIT order flow entrenches asymmetries in the builder market. Reliable execution requires privileged builder access; as a result, value capture concentrates in a tight buildersearcher network.

Crucially, the priority fee heterogeneity that clarifies how surplus is split across channels and pairs. Descriptively (Table 4), total fees vary sharply across searchers using the same builder, with differences tightly aligned to the price impact component of markout. Private configurations systematically exhibit lower total fees relative to realized profits than their public counterparts, consistent with reduced bidding pressure on private routes. The fee income component is comparatively stable, while the price impact component is the main driver of cross-searcher dispersion.

The econometric results quantify this surplus sharing. On public routes, total fees increase strongly with both the price impact and fee income components. In our preferred specification with token-pair and fee-tier fixed effects (Table 5, col. (4)), the marginal response of total fees is about 0.75 per dollar of price-impact(0.7511) and about 0.75 per dollar of fee income (0.7464). On private routes, profit-to-fee pass-through is attenuated, most clearly for price impact: the marginal response falls from about 0.75 on public routes to about 0.32 on private routes ($0.7511 - 0.4303 \approx 0.3208$), and the difference is statistically significant. For fee-income markout, the point estimate also implies a lower slope on private routes (about 0.48 versus 0.75 on public routes, $0.7464 - 0.2680 \approx 0.4784$), but the difference is not statistically significant. The level effect of the private indicator is small and statistically insignificant, indicating that the key distinction is in slopes rather than a uniform surcharge or discount. Finally, the indicator for known searcher–builder integration is positive and statistically significant, suggesting that integrated pairs tend to remit higher observed total fees after conditioning on realized profits, controls, and fixed effects.

References

1. Bahrani, M., Garimidi, P., Roughgarden, T.: Centralization in block building and proposer-builder separation (2024), https://arxiv.org/abs/2401.12120
2. Capponi, A., et al.: Proposer-builder separation, payment for order flows, and centralization in blockchain. SSRN (2024), https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4723674

3. Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., Juels, A.: Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges (2019), https://arxiv.org/abs/1904.05234
4. Flashbots: Mempool dumpster (2025), https://mempool-dumpster.flashbots.net/index.html
5. Flashbots: Mev-boost specifications (2025), https://docs.flashbots.net/flashbots-mev-boost/architecture-overview/specifications, documentation
6. Heimbach, L., et al.: Ethereum's proposer-builder separation (2023), https://arxiv.org/abs/2305.19037
7. Ota, K., Rezzoli, D.: Jit classification only private (dune query 59605299617682) (2025), https://dune.com/queries/6161226, interactive dashboard and dataset of private JIT liquidity bundles. Accessed: 29 October 2025
8. Ota, K., Rezzoli, D.: Jit ratio in block (dune query 60662769744171) (2025), https://dune.com/queries/6066276, dashboard and dataset of JIT liquidity bundles in blocks. Accessed: 29 October 2025
9. Pahari, V., Canidio, A.: How exclusive are ethereum transactions? evidence from non-winning blocks (2025), https://arxiv.org/abs/2509.16052
10. Trotti, B.L., Tang, W., El-Azouzi, R., Fanti, G., Menasche, D.S.: Strategic analysis of just-in-time liquidity provision in concentrated liquidity market makers (2025), https://arxiv.org/abs/2509.16157
11. Wan, X., Adams, A.: Just-in-time liquidity on the uniswap protocol. Uniswap Labs Research Blog (September 2022), https://blog.uniswap.org/jit-liquidity
12. Wan, X., Adams, A.: Automated market maker and liquidity provision: Evidence from uniswap. https://ssrn.com/abstract=4382303 (2023), elsevier BV, SSRN Working Paper No. 4382303. DOI: https://doi.org/10.2139/ssrn.4382303
13. Wang, S., Huang, Y., Zhang, W., Huang, Y., Wang, X., Tang, J.: Private order flows and builder bidding dynamics: The road to monopoly in ethereum's block building market (2024), https://arxiv.org/abs/2410.12352
14. Öz, B., Sui, D., Thiery, T., Matthes, F.: Who wins ethereum block building auctions and why? (2024), https://arxiv.org/abs/2407.13931