Bitcoin Fee Estimation:

A Structural Model Approach

Kristian Praizner
Mentors: Dan Aronoff, Armin Sabouri

digital currency initiative

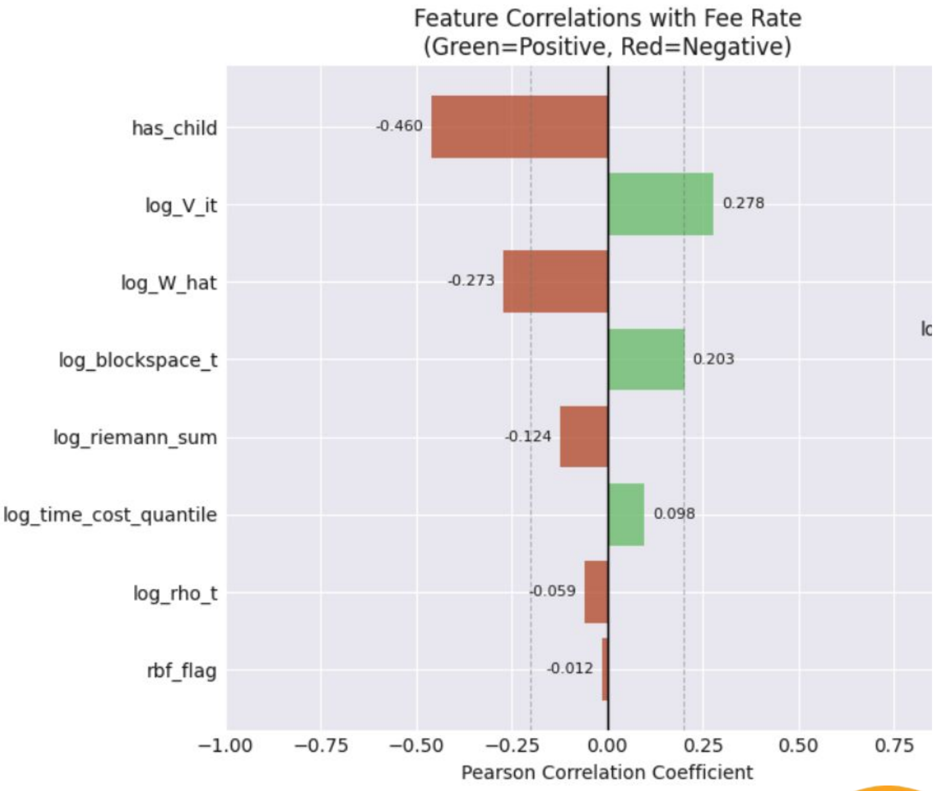
mit media lab

Talk outline

- Motivation
- Data Sources
- Methodology
- Conclusion

Executive Summary:

- Main drivers of fee rate are V (transaction amount) + and CPFP (child pays for parent) -
- Several other variables are correlated with fee rate
- We choose to use a structural model in order to recover the true drivers by eliminating confounders

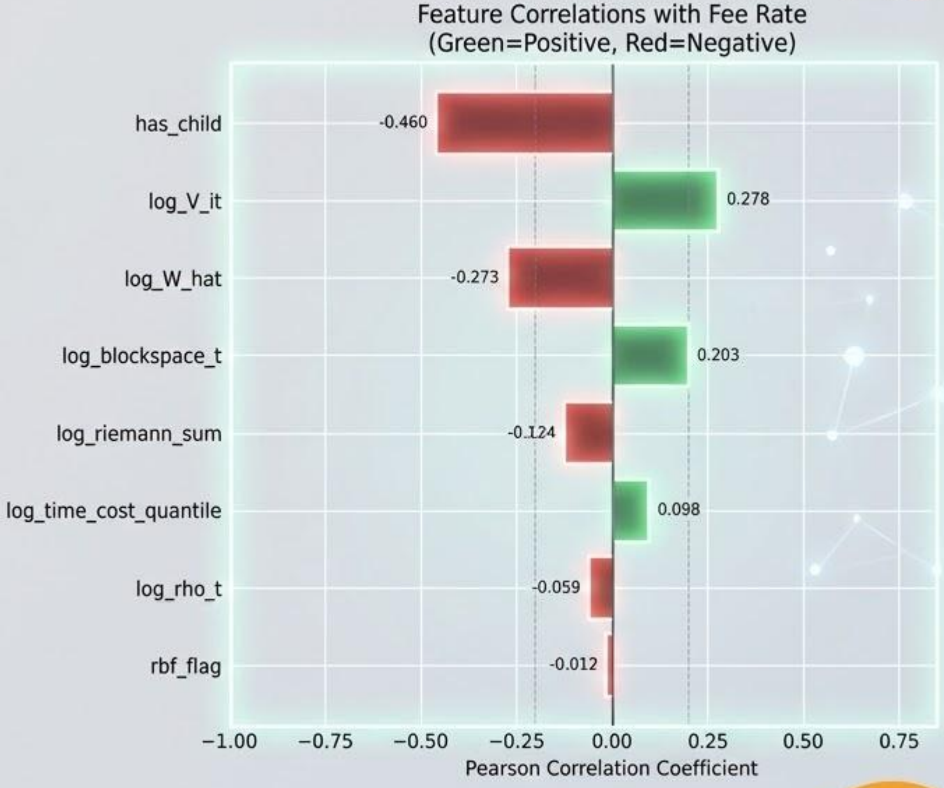


Executive Summary:

- Main drivers of fee rate are V (transaction amount) + and CPFP (child pays for parent) -

- Several other variables are correlated with fee rate

- We choose to use a structural model in order to recover the true drivers by eliminating confounders



Motivation

The driving factors and significance of the research.

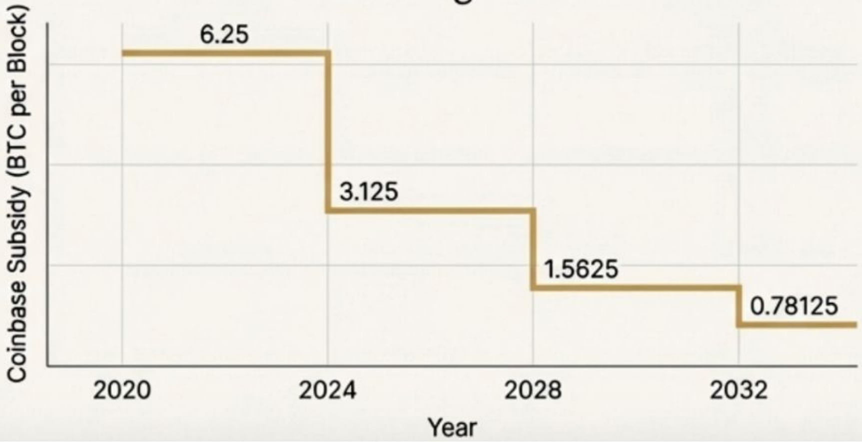
The Block Reward Has Two Components, But One is Programmed to Disappear.

The reward consists of two sources:
1. **Coinbase Subsidy**: A fixed amount of newly minted BTC, set by the protocol.
2. **Transaction Fees**: Voluntary fees paid by users to have their transactions included in a block.

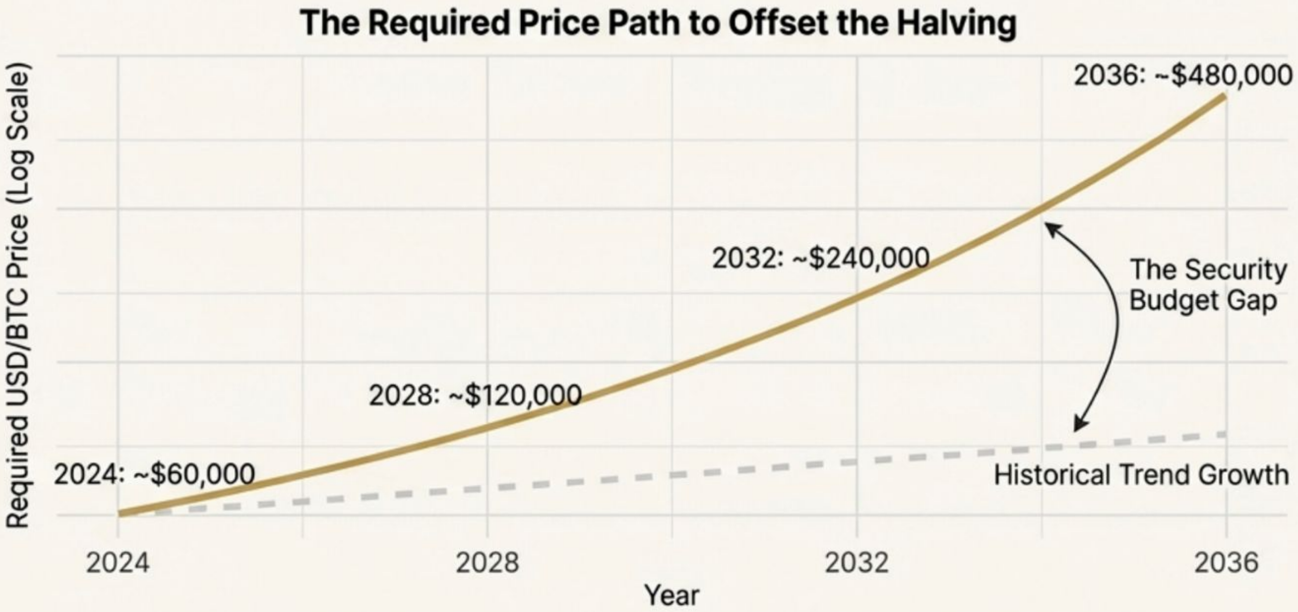
Historically, the coinbase subsidy has comprised the vast majority of the reward. From 2022-2024, it accounted for approximately 95%.

Block Reward Composition (2022-2024)

The Halving Schedule



To Maintain Security, Bitcoin's Price Must Grow Exponentially—An Unsustainable Path



If the BTC-denominated coinbase subsidy halves every four years, the USD price of Bitcoin must double in the same period just to keep the miners' revenue (and thus network security) constant. This implies a required annual growth rate of ~25%, a rate that exceeds historical trends and is dynamically impossible for any asset to sustain in the long term

This raises the critical question for Bitcoin's future viability: **Will transaction fees rise to fill the ever-widening gap?**

Data Sources and First Looks

An overview of the data collection and initial analysis.

The model is Estimated on Granular, High-Frequency data from a Dedicated Bitcoin Node

Data Source & Structure

We operated a custom Bitcoin Node from August to December 2025 to collect high-fidelity mempool and blockchain data

Data Structure

- The timeline was partitioned into epochs of 30 minutes
- For each transaction, we measured fee, mempool density, waittime, UTXO value, re-spend time, etc

Implementation Details

The entire pipeline is available on Github

Data Sourced

tx_id, tx_data, child_txid, conf_block_hash, found_at, mined_at, rbf_fee_total, min_respend_blocks, absolute_fee, fee_rate, version, seen_in_mempool, waittime, weight, size, total_output_amount, mempool_size, mempool_tx_count, output_weights

The model is Estimated on Granular, High-Frequency data from a Dedicated Bitcoin Node

Data Source & Structure

We operated a custom Bitcoin Node from August to December 2025 to collect high-fidelity mempool and blockchain data

Data Structure

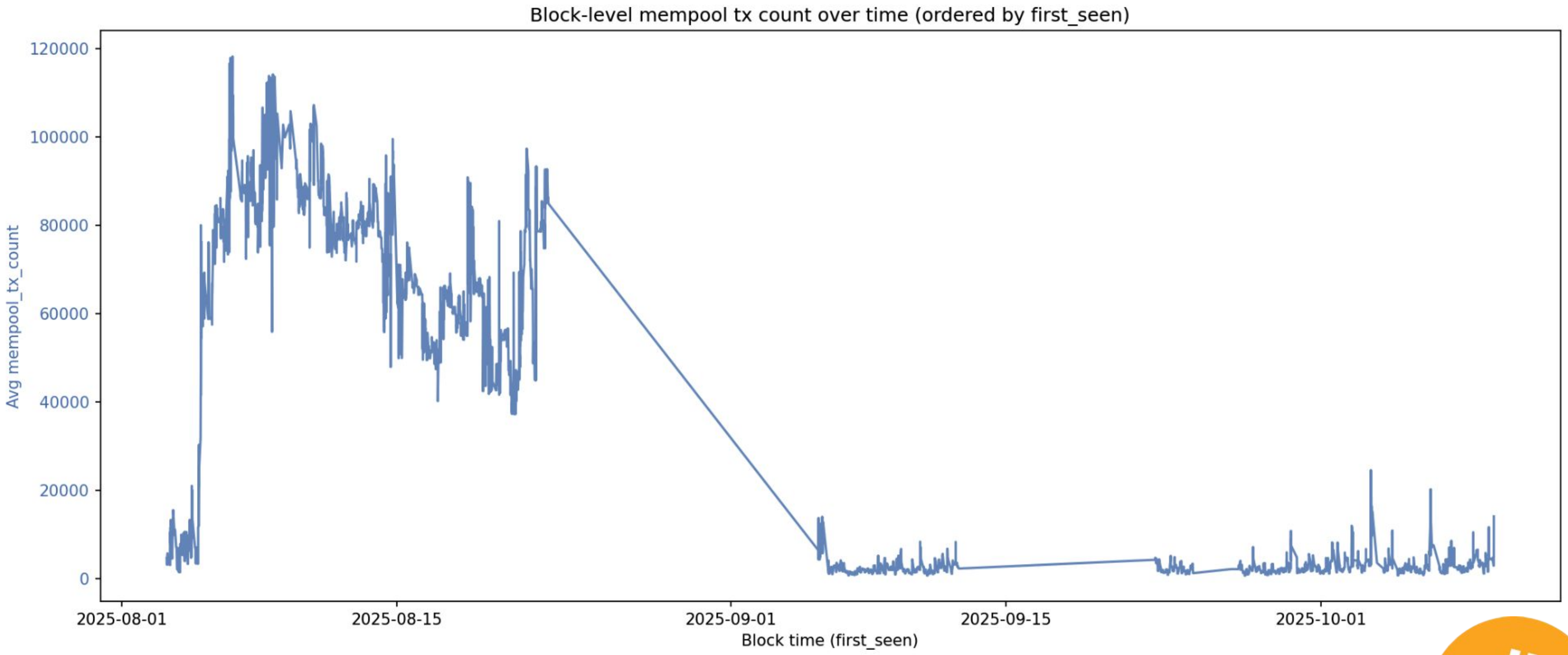
- The timeline was partitioned into epochs of 30 minutes
- For each transaction, we measured fee, mempool density, waittime, UTXO value, re-spend time, etc

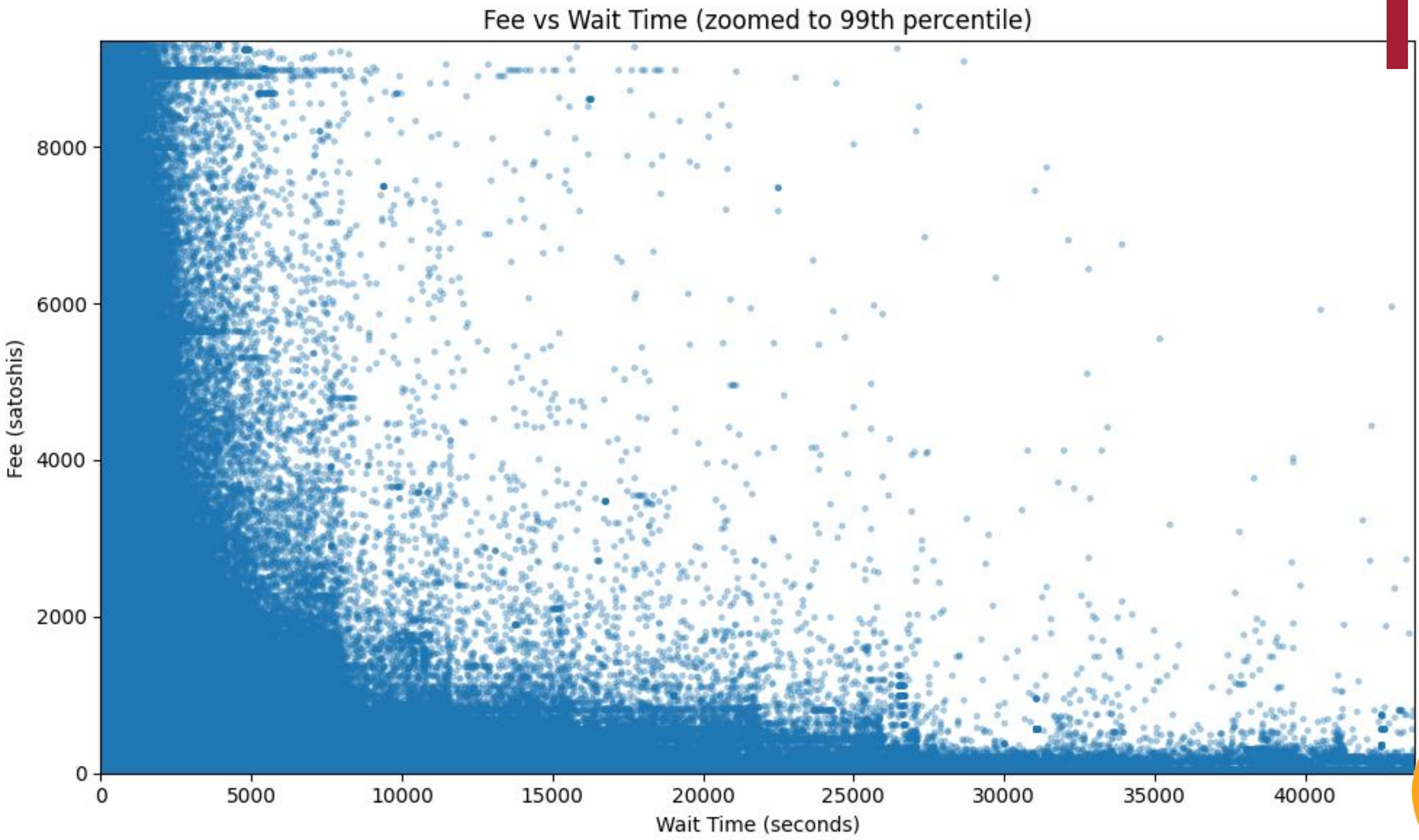
Implementation Details

The entire pipeline is available on Github

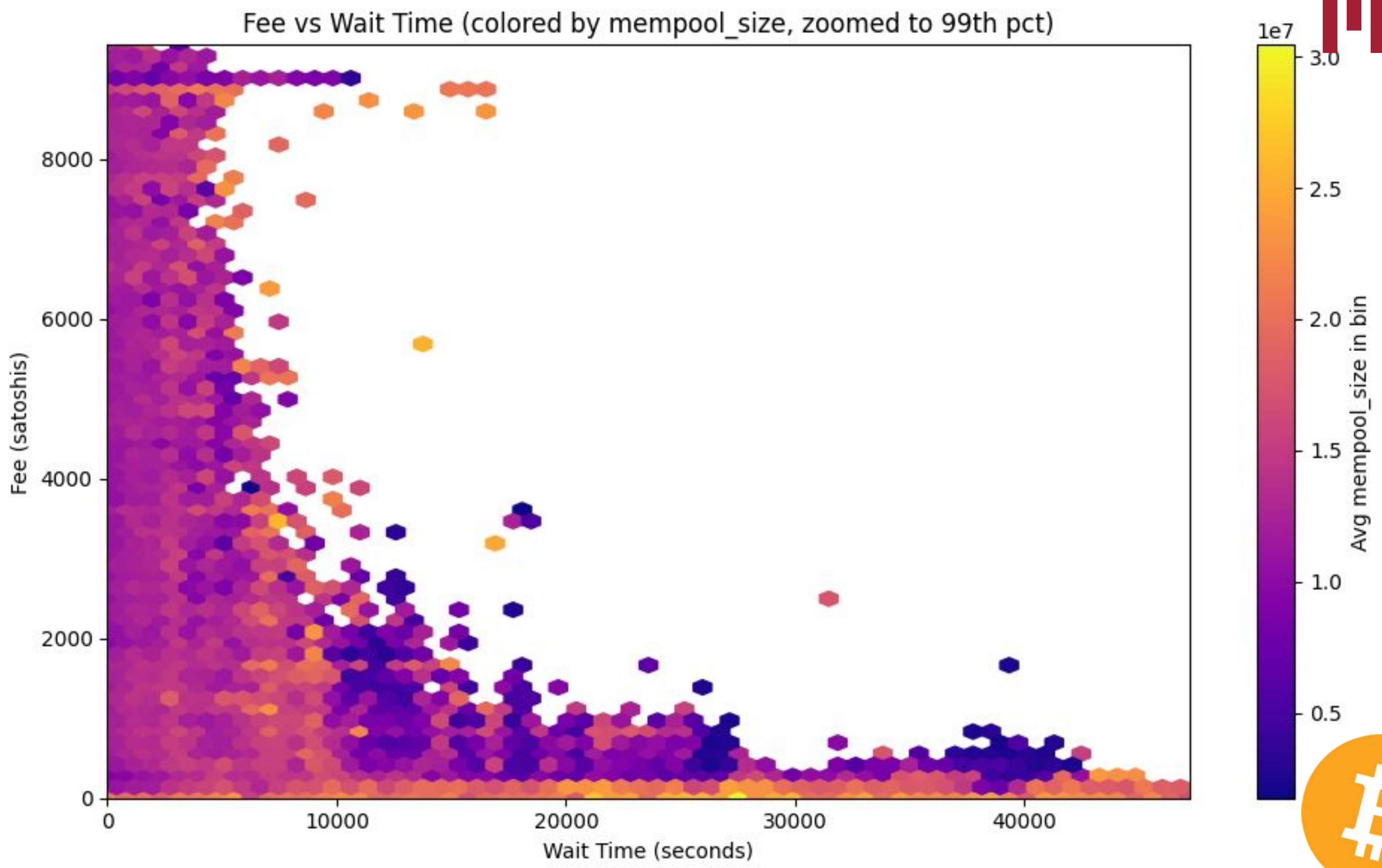
Data Sourced

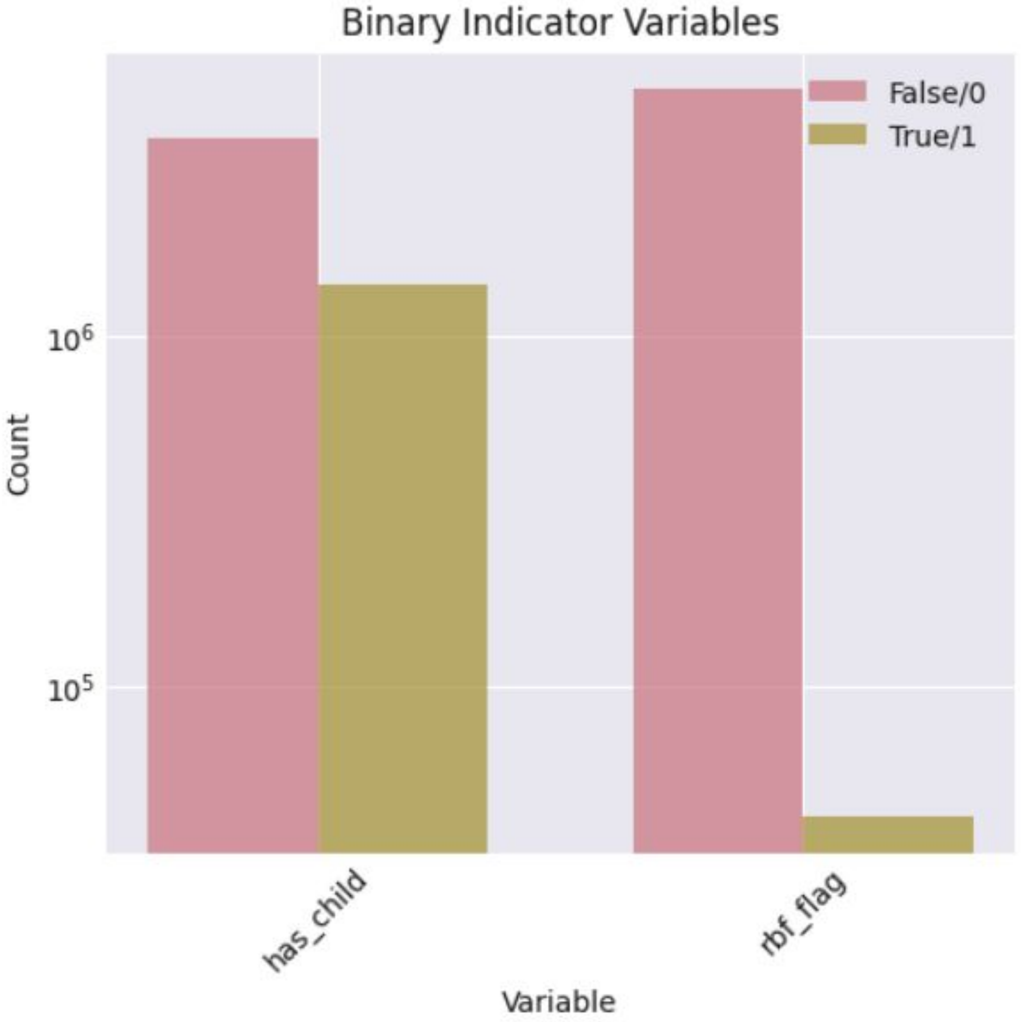
tx_id, tx_data, child_txtid, conf_block_hash, found_at, mined_at, rbf_fee_total, min_respend_blocks, absolute_fee, fee_rate, version, seen_in_mempool, waittime, weight, size, total_output_amount, mempool_size, mempool_tx_count, output_weights

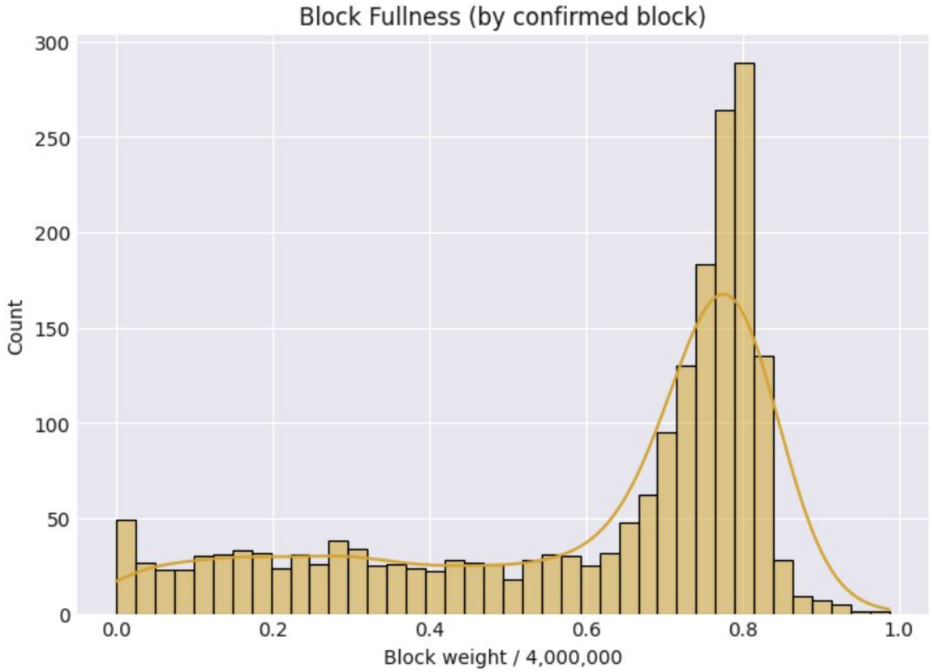

Block-level mempool tx count over time (ordered by first_seen)
Avg mempool_tx_count
0
20000
40000
60000
80000
100000
120000
2025-08-01
2025-08-15
2025-09-01
2025-09-15
2025-10-01
Block time (first_seen)

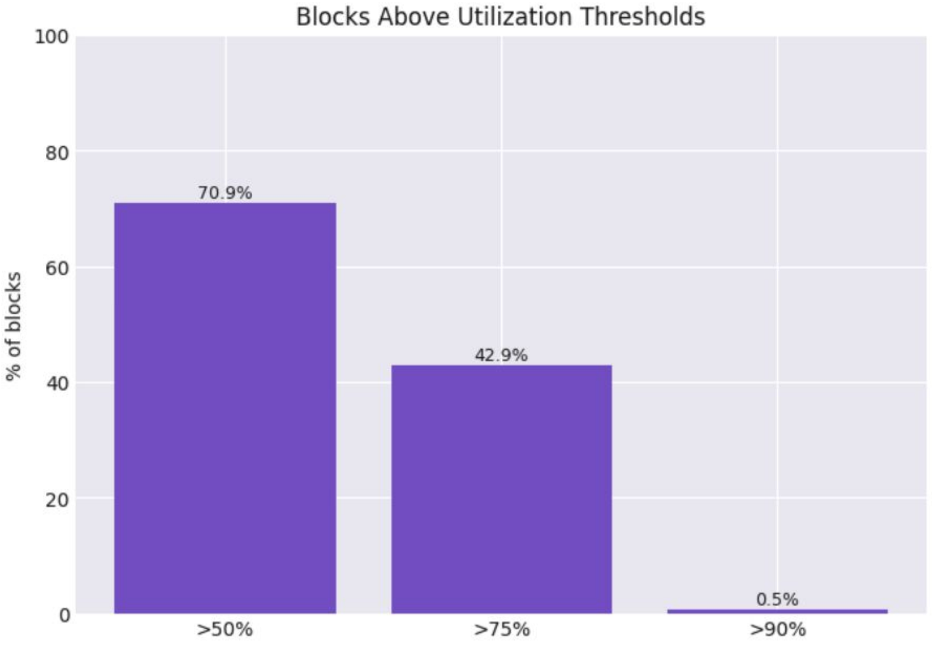


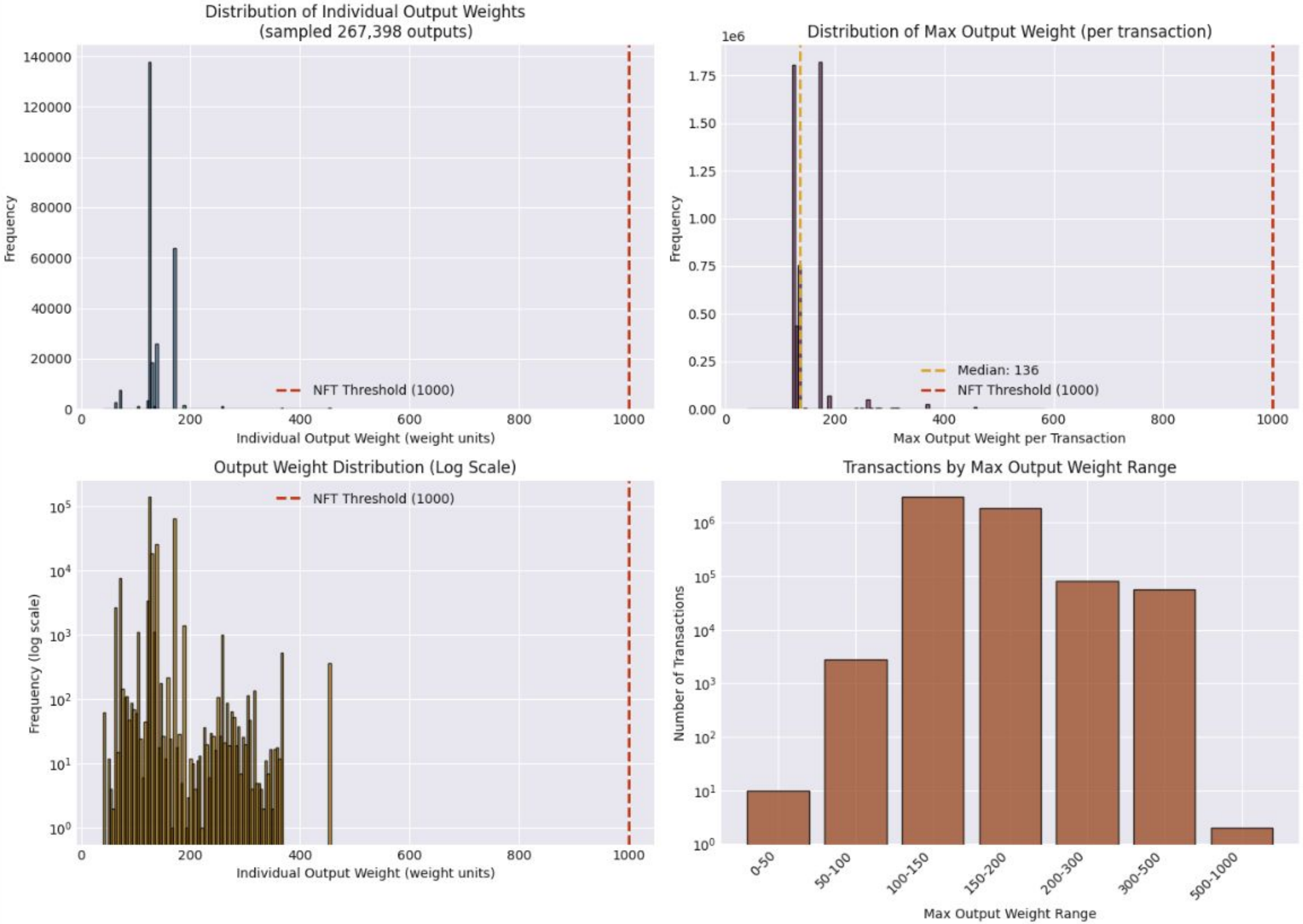
Fee vs Wait Time (zoomed to 99th percentile)
Fee (satoshis)
0
2000
4000
6000
8000
Wait Time (seconds)
0
5000
10000
15000
20000
25000
30000
35000
40000


Fee vs Wait Time (colored by mempool_size, zoomed to 99th pct)
Fee (satoshis)
Wait Time (seconds)
Avg mempool_size in bin
1e7
0
2000
4000
6000
8000
10000
20000
30000
40000
0.5
1.0
1.5
2.0
2.5
3.0


Binary Indicator Variables
False/0
True/1
10^6
10^5
Count
has_child
rbf_flag
Variable


Block Fullness (by confirmed block)
Count
0
50
100
150
200
250
300
0.0
0.2
0.4
0.6
0.8
1.0
Block weight / 4,000,000


Blocks Above Utilization Thresholds
% of blocks
0
20
40
60
80
100
70.9%
42.9%
0.5%
>50%
>75%
>90%



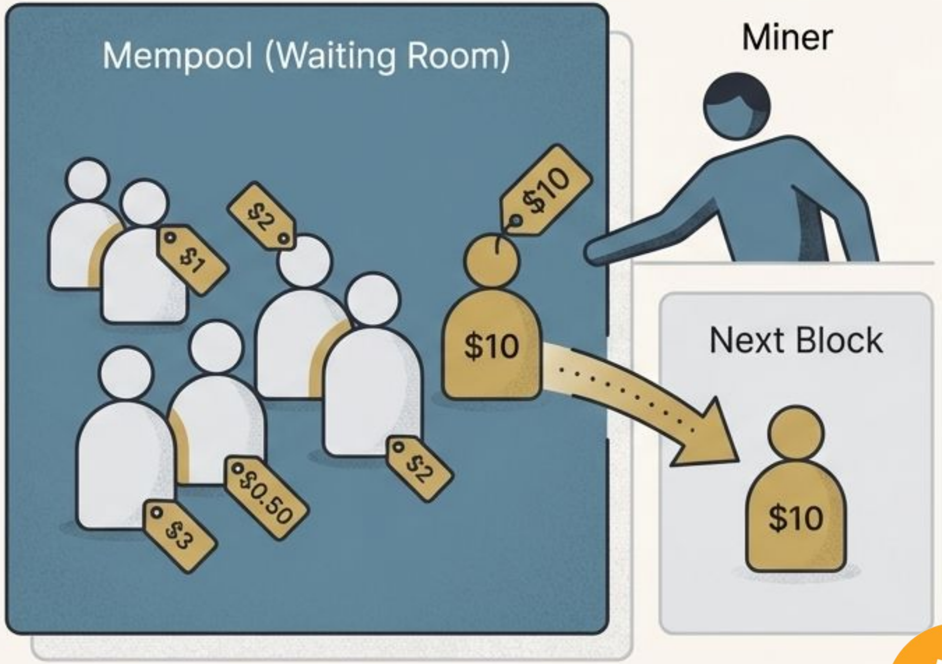
Distribution of Individual Output Weights
(sampled 267,398 outputs)
Frequency
Individual Output Weight (weight units)
NFT Threshold (1000)
Distribution of Max Output Weight (per transaction)
1e6
Median: 136
NFT Threshold (1000)
Max Output Weight per Transaction
Output Weight Distribution (Log Scale)
Frequency (log scale)
Individual Output Weight (weight units)
Transactions by Max Output Weight Range
Number of Transactions
0-50
50-100
100-150
150-200
200-300
300-500
500-1000
Max Output Weight Range

Methodology

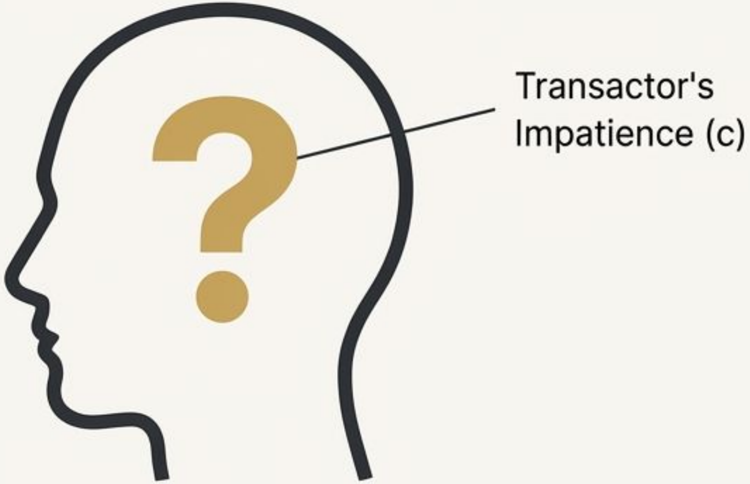
Implementation of a two-stage structural model.

We Model the Fee Market as a Queue Where Impatient Users Pay to Cut the Line.

- Our approach is built on the economic model of Huberman et al. (2021).
- Miners, as profit-maximizers, prioritize transactions with the highest fees.
- Transactors differ in their "impatience" or time-cost. They select a fee to secure a desired spot in the queue of pending transactions (the mempool).
- The resulting fee is therefore a function of network congestion and the distribution of impatience among all users.



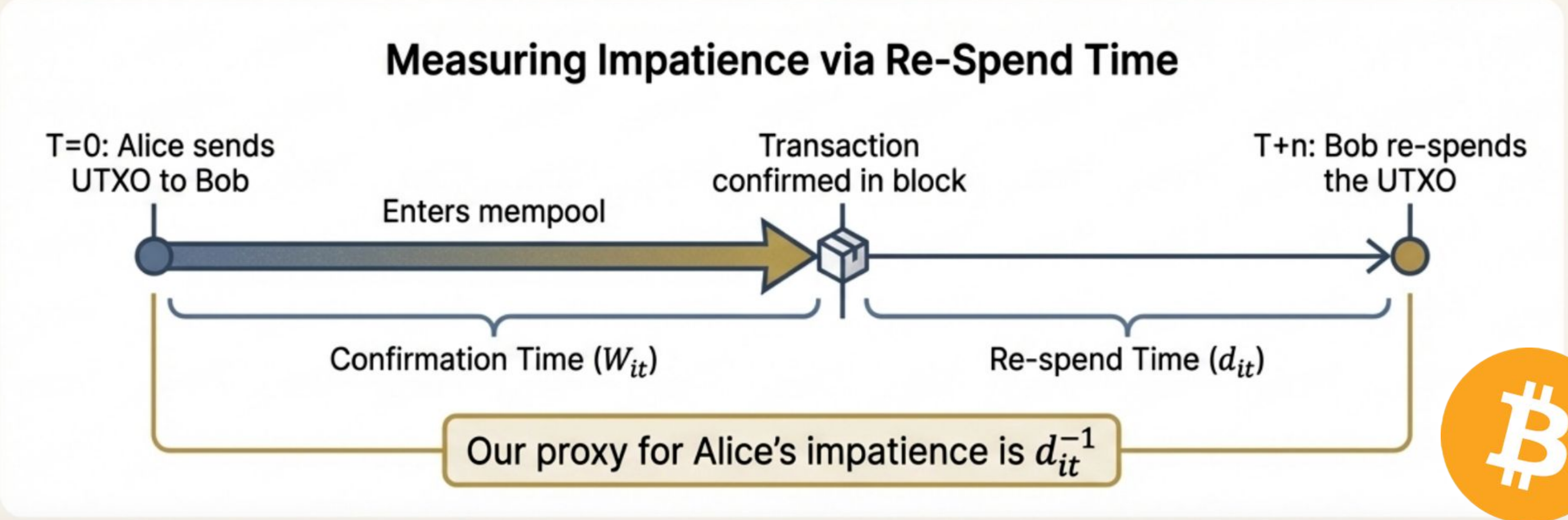
The Challenge: How Can We Empirically Measure a User's 'Impatience'?

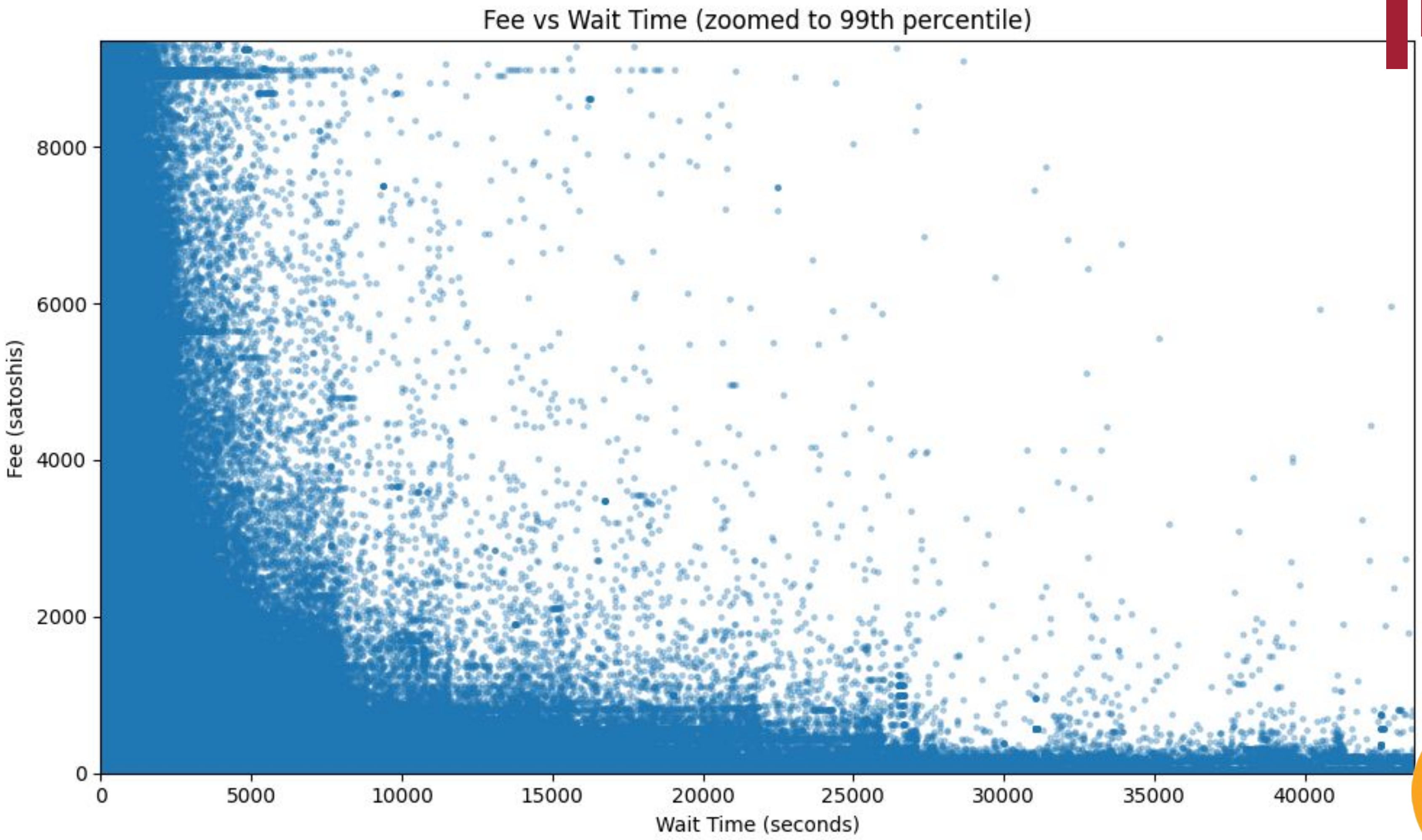


Estimating the model requires a data-driven measure of impatience. This presents a core difficulty, as impatience (a user's "time-cost") is an unobservable internal state of mind. We need a clever, empirical proxy to extract this preference from on-chain data.

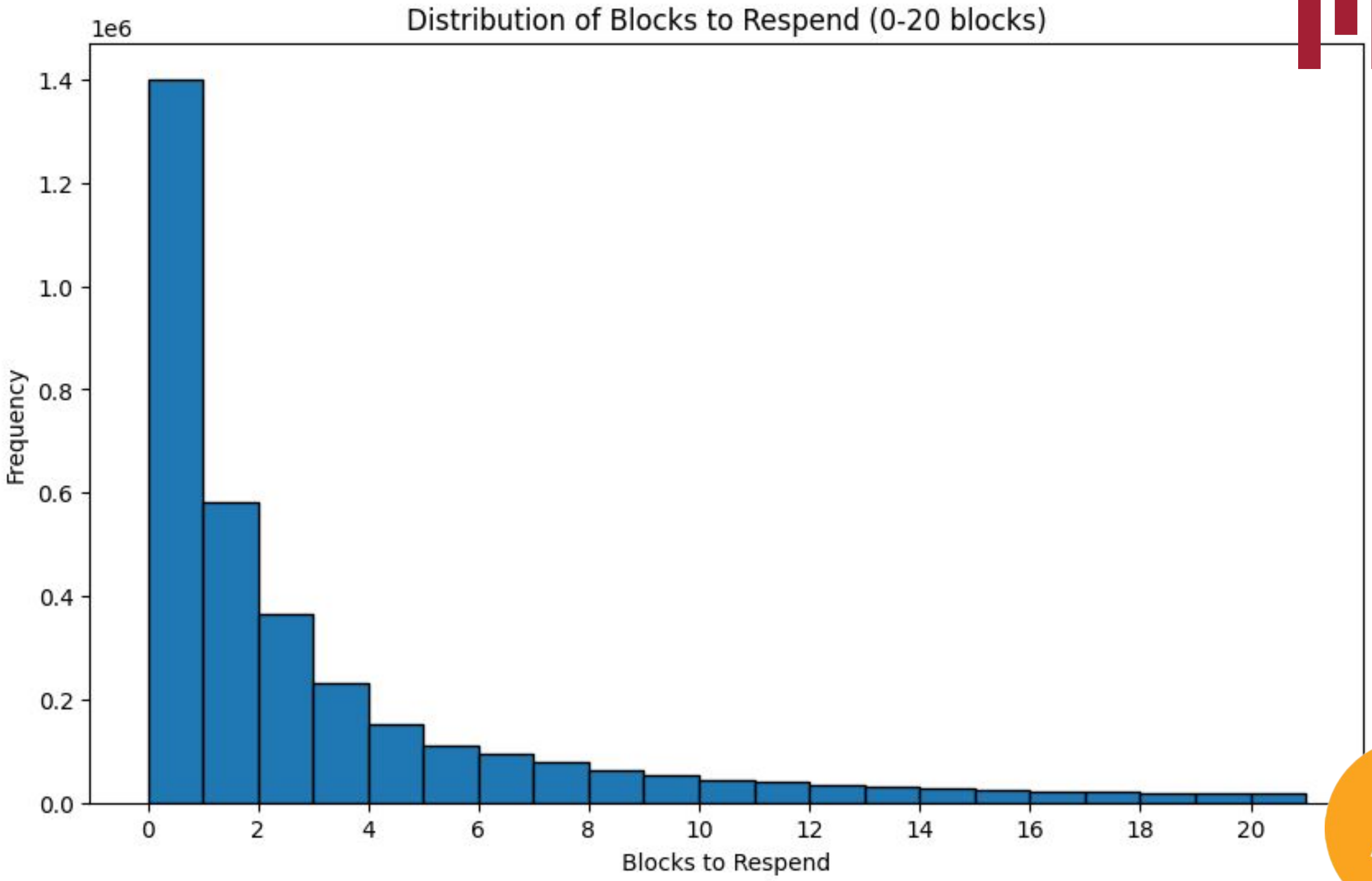
The Breakthrough: Using ‘Re-Spend Time’ as a Proxy for Impatience

- We borrow an ingenious insight from Möser & Böhme (2015).
- We hypothesize that the time it takes for a transaction’s output (UTXO) to be spent again is correlated with the original sender’s urgency.
- A fast re-spend implies high impatience; a slow re-spend implies low impatience. The inverse of the re-spend time becomes our empirical proxy.





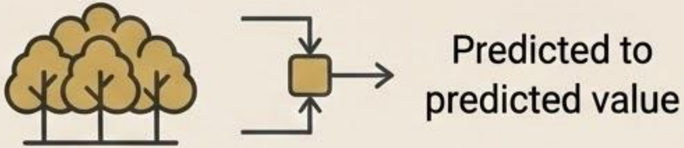
Fee vs Wait Time (zoomed to 99th percentile)
Fee (satoshis)
0
2000
4000
6000
8000
Wait Time (seconds)
0
5000
10000
15000
20000
25000
30000
35000
40000


Distribution of Blocks to Respend (0-20 blocks)
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
0
2
4
6
8
10
12
14
16
18
20
Blocks to Respend

A Random Forest Regressor for Wait Time Prediction

Our model uses a Random Forest Regressor to predict the log-transformed wait time (log_waittime) based on four key log-transformed and binary features.

Model Overview:



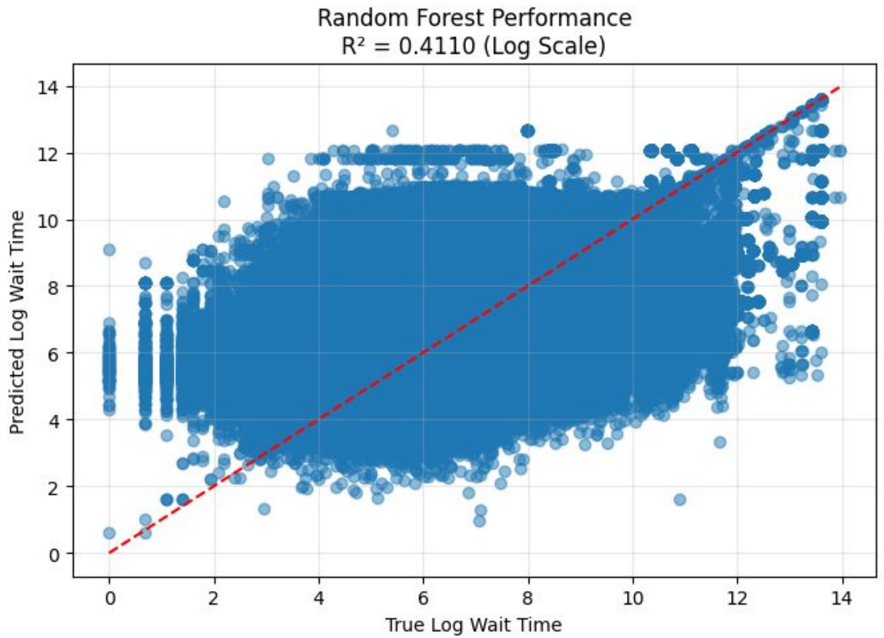
An ensemble learning method that constructs a multitude of decision trees at training time. It outputs the average prediction of the individual trees, reducing overfitting and improving accuracy.

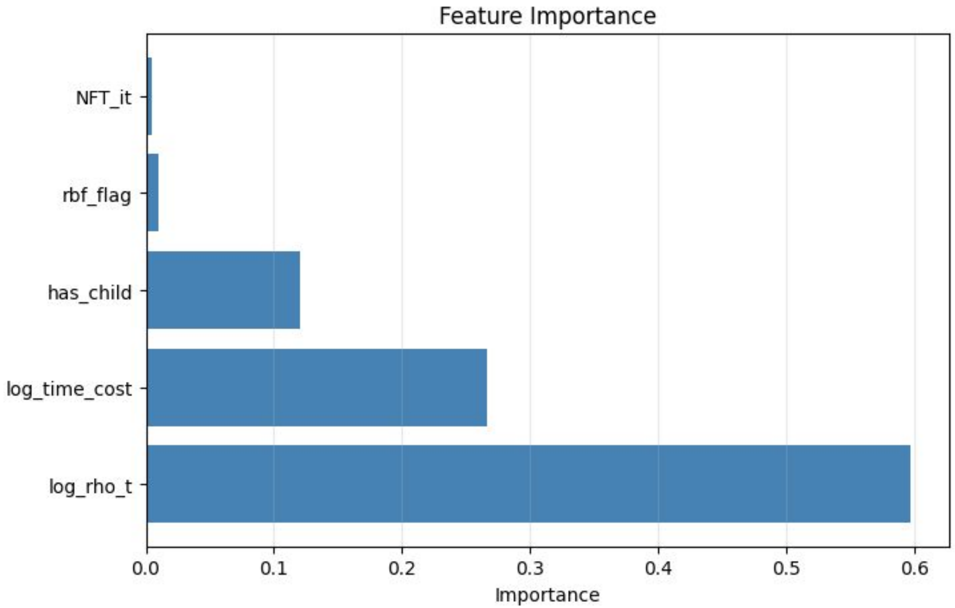
Features & Target (Log-Transformed):

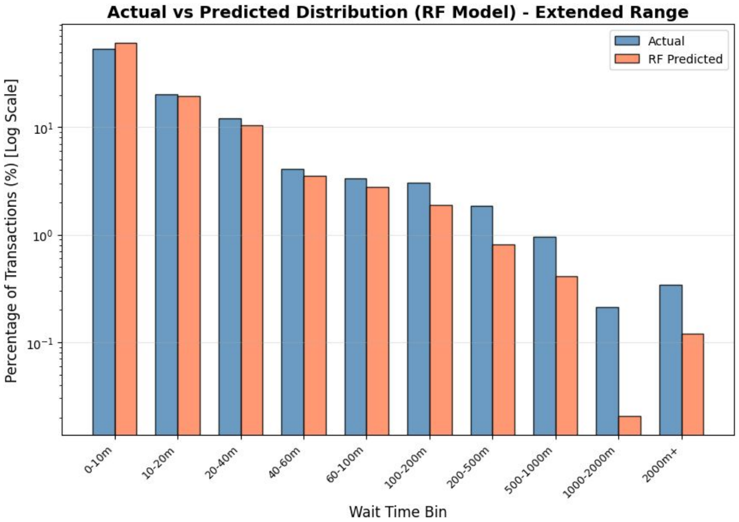
Target (y): log_waittime (Log of Transaction Wait Time)

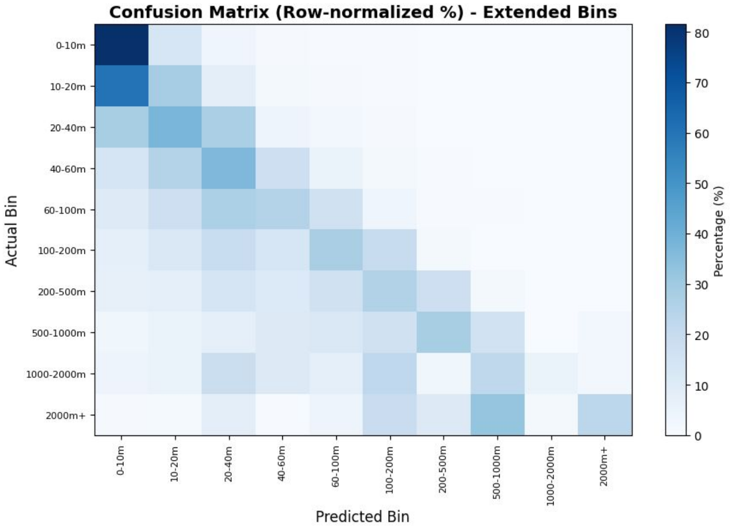
Features (X):

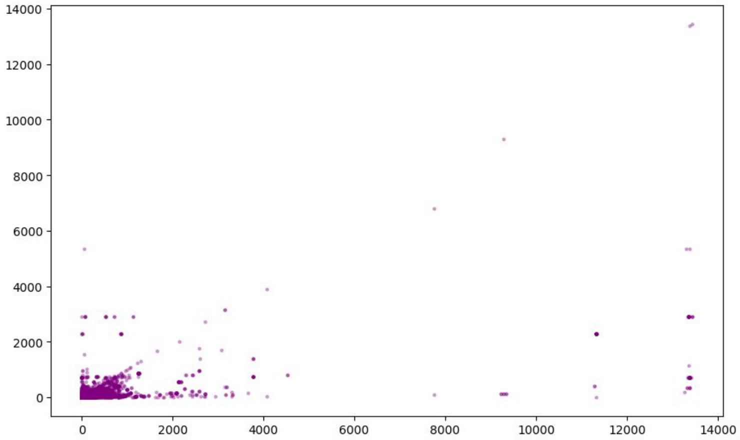
- log_rho_t (Log of Network Congestion)
- log_time_cost (Log of Impatience Proxy)
- has_child (Binary: Has Child Transaction)
- rbf_flag (Binary: Replace-By-Fee Flag)

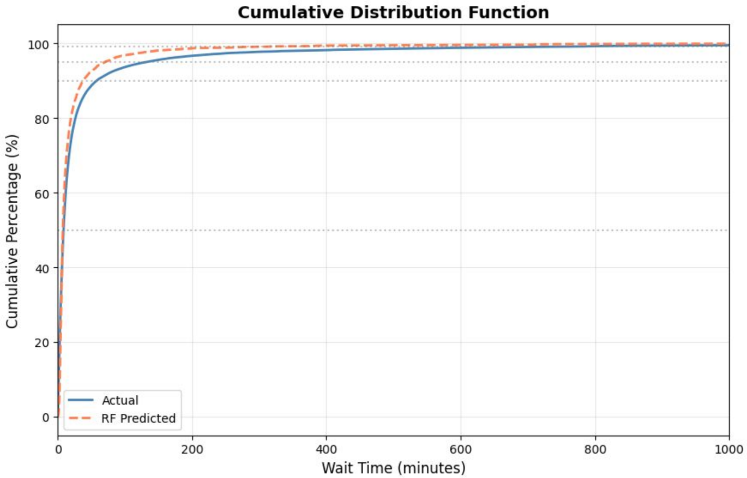

Random Forest Performance
R² = 0.4110 (Log Scale)
Predicted Log Wait Time
True Log Wait Time
0
2
4
6
8
10
12
14


Feature Importance
NFT_it
rbf_flag
has_child
log_time_cost
log_rho_t
Importance
0.0
0.1
0.2
0.3
0.4
0.5
0.6


Actual vs Predicted Distribution (RF Model) - Extended Range
Actual
RF Predicted
Percentage of Transactions (%) [Log Scale]
10^1
10^0
10^{-1}
0-10m
10-20m
20-40m
40-60m
60-100m
100-200m
200-500m
500-1000m
1000-2000m
2000m+
Wait Time Bin


Confusion Matrix (Row-normalized %) - Extended Bins
Actual Bin
0-10m
10-20m
20-40m
40-60m
60-100m
100-200m
200-500m
500-1000m
1000-2000m
2000m+
Predicted Bin
Percentage (%)
0
10
20
30
40
50
60
70
80


0
2000
4000
6000
8000
10000
12000
14000


Cumulative Distribution Function
Cumulative Percentage (%)
0
20
40
60
80
100
Actual
RF Predicted
0
200
400
600
800
1000
Wait Time (minutes)

Rationale

- We use the first stage estimation to eliminate the influence of fees in waittime
- We do this by choosing variables that are not plausibly influenced by fees (congestion, time cost, CPFP)
- This proxy is then used as a variable in our second stage fee model

Rationale

- We use the first stage estimation to eliminate the influence of fees in waittime
- We do this by choosing variables that are not plausibly influenced by fees (congestion, time cost, CPFP)
- This proxy is then used as a variable in our second stage fee model

This Leads to Our Structural Model for Estimating Transaction Fees.

Our model empirically realizes the theory, expressing the transaction fee as a function of congestion, the aggregate impatience of other users, and other key transaction characteristics. The model is log-linear to reflect the “fat tail” distribution of fees.

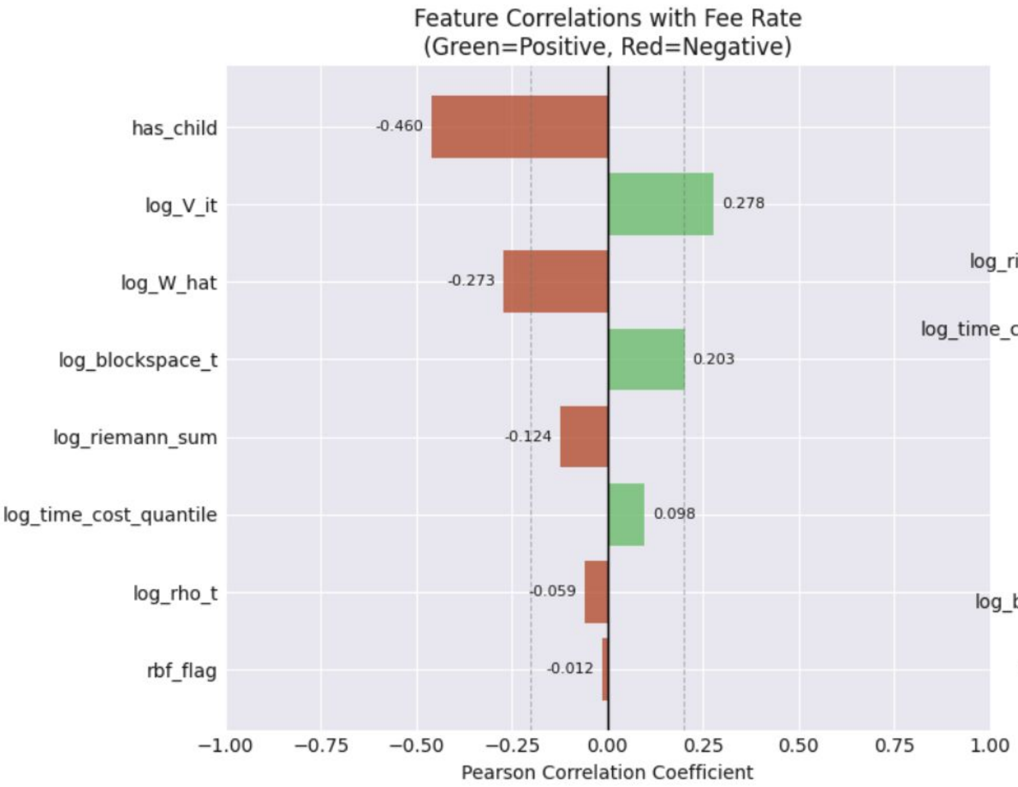
The Fee. The transaction fee (in USD) that we aim to predict.

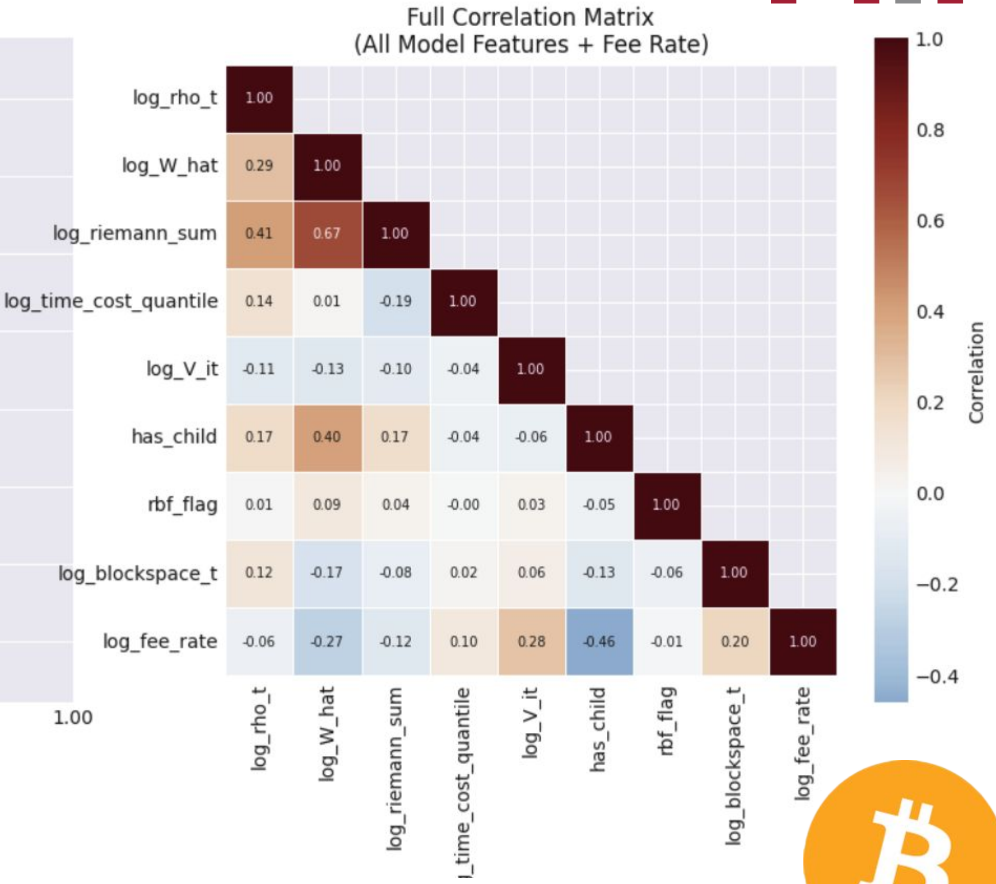
Mempool Congestion. The average number of transactions in the mempool during the epoch. How crowded is the waiting room?

$$b_{it} = \alpha_1 + \alpha_2 \hat{\rho}_t + \alpha_3 * \left[\sum \ldots\right] + \alpha_4 V_{it} + \alpha_5 \mathrm{Weight}_{it} + \ldots + \epsilon_{it}$$

The Impatience Premium. This core term aggregates the effect of all more-impatient users ahead in the queue, based on our re-spend time proxy.

Control Variables. We control for transaction value, weight (size), exchange activity, and other factors.


Feature Correlations with Fee Rate
(Green=Positive, Red=Negative)
has_child
log_V_it
log_W_hat
log_blockspace_t
log_riemann_sum
log_time_cost_quantile
log_rho_t
rbf_flag
-0.460
0.278
-0.273
0.203
-0.124
0.098
-0.059
-0.012
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
Pearson Correlation Coefficient


Full Correlation Matrix
(All Model Features + Fee Rate)
log_rho_t
log_W_hat
log_riemann_sum
log_time_cost_quantile
log_V_it
has_child
rbf_flag
log_blockspace_t
log_fee_rate
1.00
0.29 1.00
0.41 0.67 1.00
0.14 0.01 -0.19 1.00
-0.11 -0.13 -0.10 -0.04 1.00
0.17 0.40 0.17 -0.04 -0.06 1.00
0.01 0.09 0.04 -0.00 0.03 -0.05 1.00
0.12 -0.17 -0.08 0.02 0.06 -0.13 -0.06 1.00
-0.06 -0.27 -0.12 0.10 0.28 -0.46 -0.01 0.20 1.00
Correlation
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Bitcoin Fee Estimation: A Structural Model Approach

Model 1: Huber Robust Regression Analysis

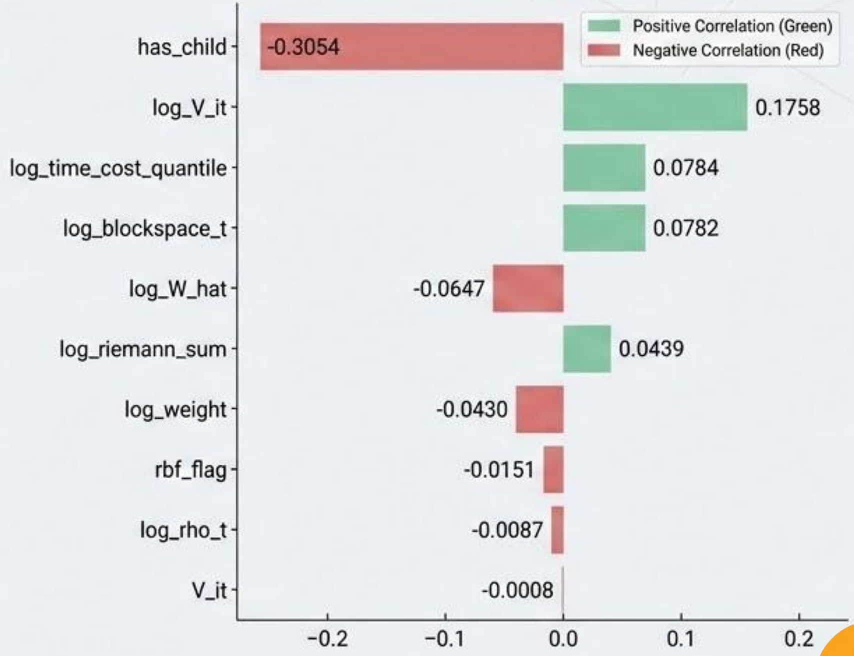
Model Performance Metrics

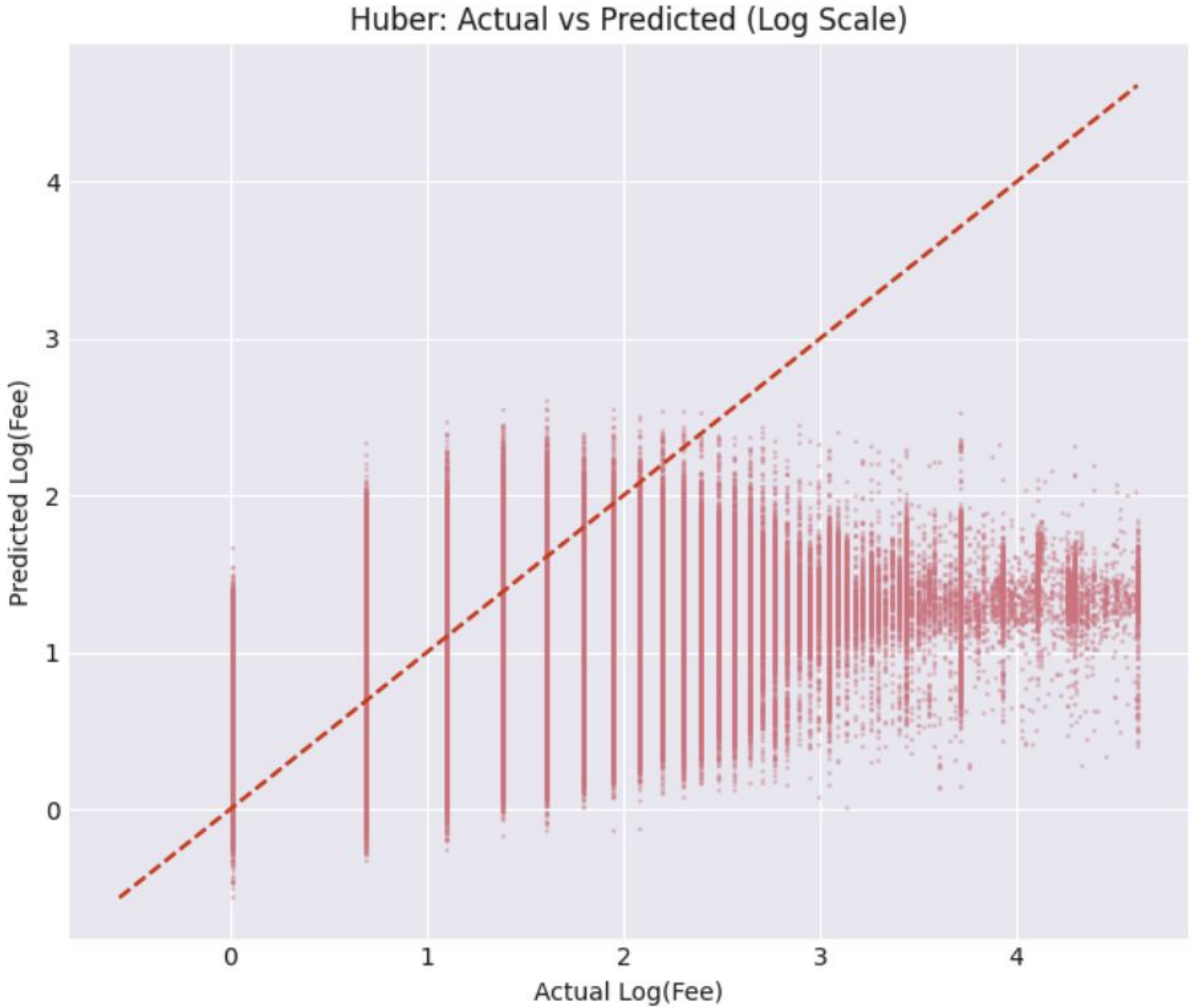
Metric	Training Set	Test Set
R^2 (R-squared)	0.3067	0.3080
MAE (Mean Absolute Error)	0.44	0.44
Median AE (Median Absolute Error)	0.36	0.36
RMSE (Root Mean Squared Error)	0.59	0.59

Outliers Down-weighted

1,291,475 (32.3%)

Top 10 Feature Coefficients (Standardized)





Huber: Actual vs Predicted (Log Scale)
Predicted Log(Fee)
Actual Log(Fee)
0
1
2
3
4

Bitcoin Fee Estimation: A Structural Model Approach

Model 2: Quantile Regression Analysis

Model Performance Metrics

Median Regression (50th percentile)

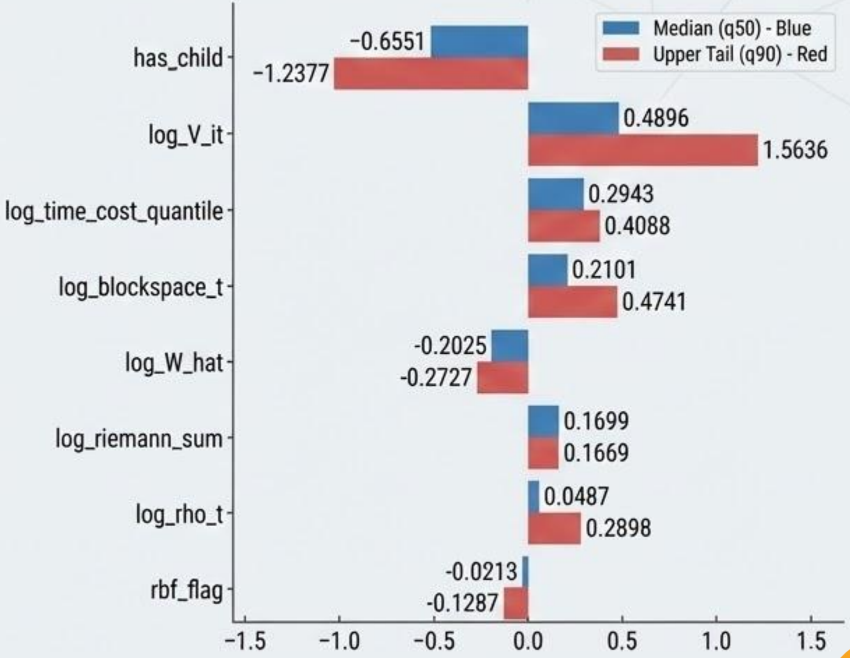
Metric	Training Set	Test Set
Pseudo-R^2	0.1029	0.1038
MAE	1.93	1.96
Median AE	0.91	0.90

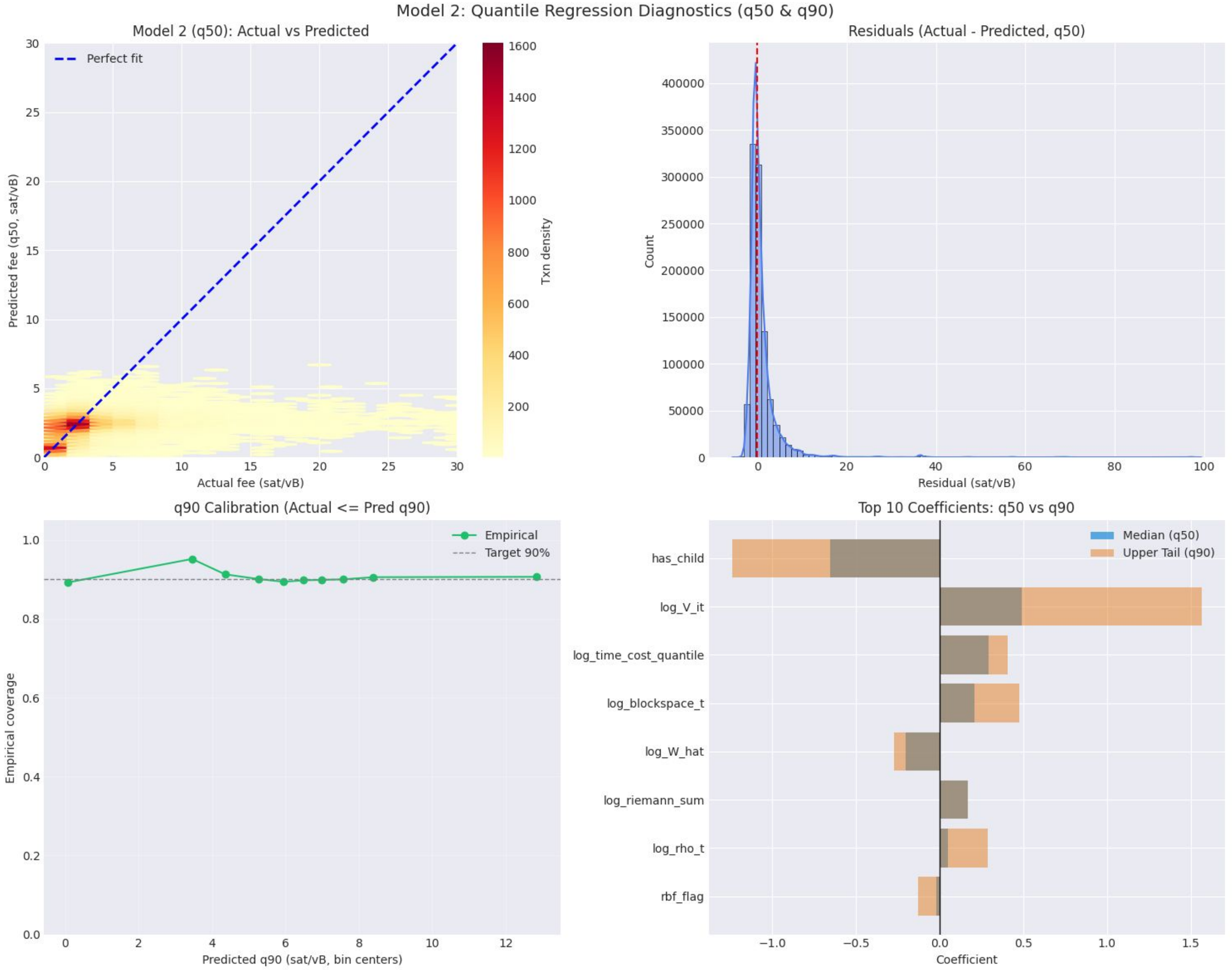
90th Percentile Regression (upper tail)

Metric	Test Set (90th percentile)
Pseudo-R^2	0.0793
MAE	4.14

Using 10,000 samples for quantile regression (computational efficiency)

Coefficient Comparison: Median vs 90th Percentile





Model 2: Quantile Regression Diagnostics (q50 & q90)
Model 2 (q50): Actual vs Predicted
Perfect fit
Predicted fee (q50, sat/vB)
Actual fee (sat/vB)
Txn density
Residuals (Actual - Predicted, q50)
Count
Residual (sat/vB)
q90 Calibration (Actual <= Pred q90)
Empirical
Target 90%
Empirical coverage
Predicted q90 (sat/vB, bin centers)
Top 10 Coefficients: q50 vs q90
Median (q50)
Upper Tail (q90)
has_child
log_V_it
log_time_cost_quantile
log_blockspace_t
log_W_hat
log_riemann_sum
log_rho_t
rbf_flag
Coefficient

Bitcoin Fee Estimation: A Structural Model Approach

Model 3: Spline Regression (Segmented)

Model Configuration & Features

- Using: 50,000 samples for Spline Regression
- Spline features (non-linear): ['log_rho_t', 'log_V_it', 'log_W_hat', 'log_blockspace_t']
- Linear features: ['log_weight', 'log_riemann_sum', 'log_time_cost_quantile', 'has_child', 'rbf_flag', 'V_it']
- Spline configuration:
  - Knots: 30 (creates 29 segments)
  - Degree: 3 (cubic splines)
  - Features per spline variable: 32
- Fitting spline regression model (log1p target)...

Model Performance Metrics

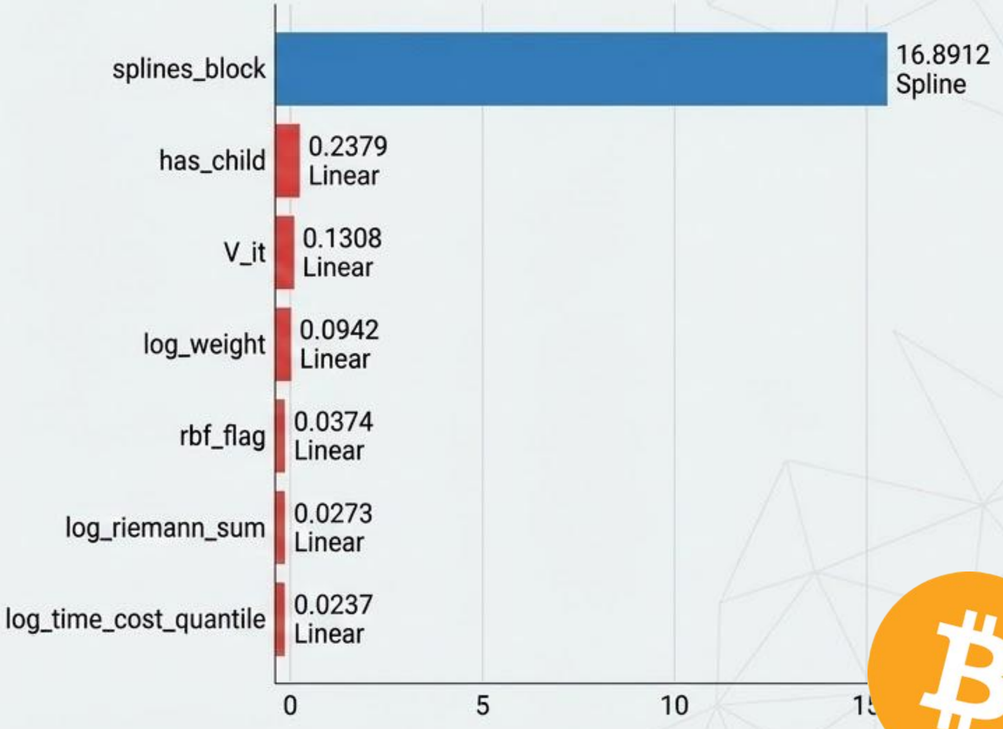
Metric	Training Set	Test Set
R^2	-0.0298	-0.0561
MAE	2.99	2.99
Median AE	1.89	1.89
RMSE	5.52	5.47

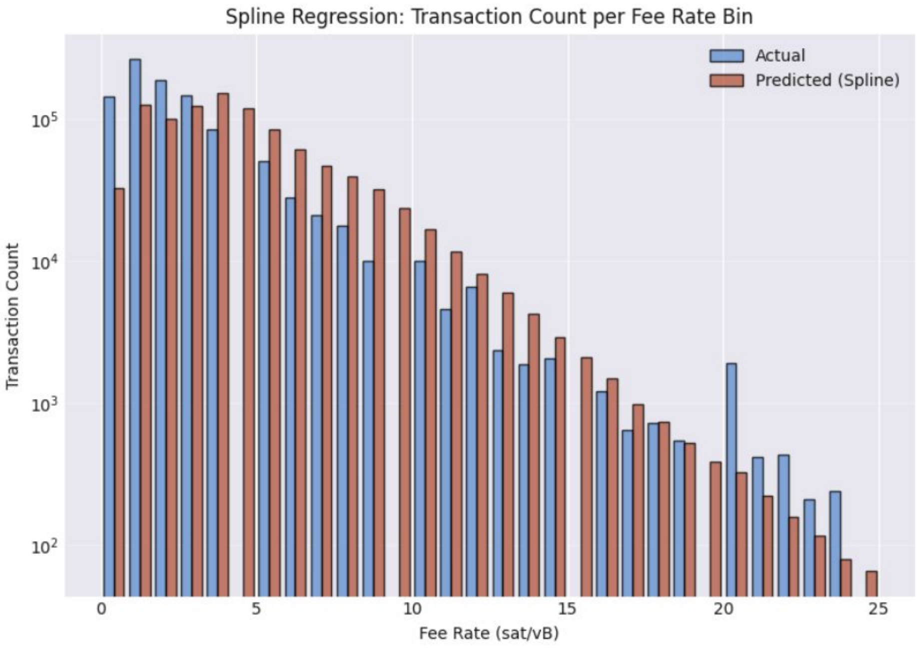
Feature Transformation Summary

Total features after spline transformation: 130

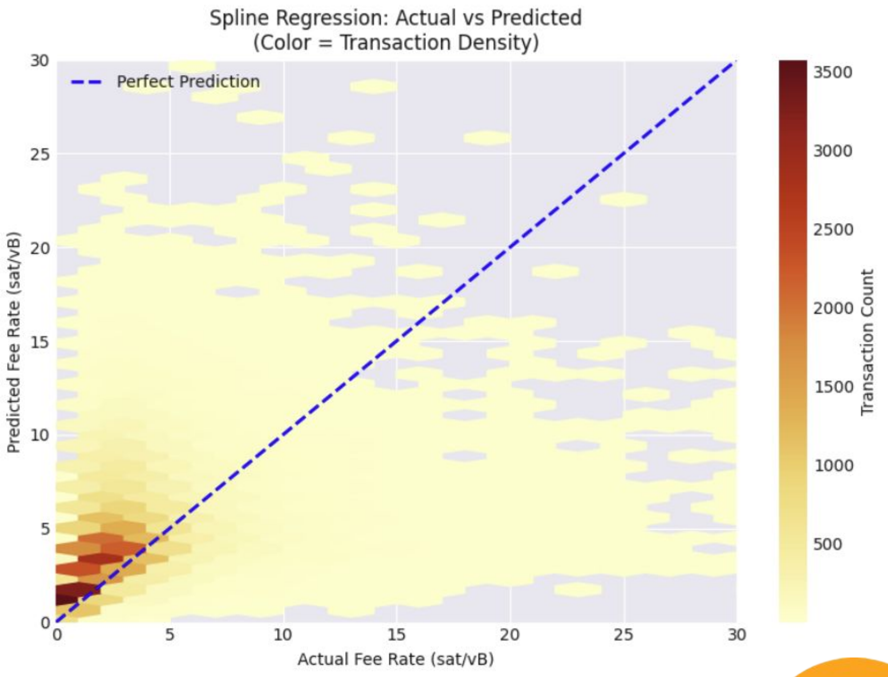
- Spline-derived features: 124
- Linear features: 6

Spline Feature Importance (sum of |coefficients|)





Spline Regression: Transaction Count per Fee Rate Bin
Actual
Predicted (Spline)
Transaction Count
Fee Rate (sat/vB)



Spline Regression: Actual vs Predicted
(Color = Transaction Density)
Perfect Prediction
Predicted Fee Rate (sat/vB)
Actual Fee Rate (sat/vB)
Transaction Count

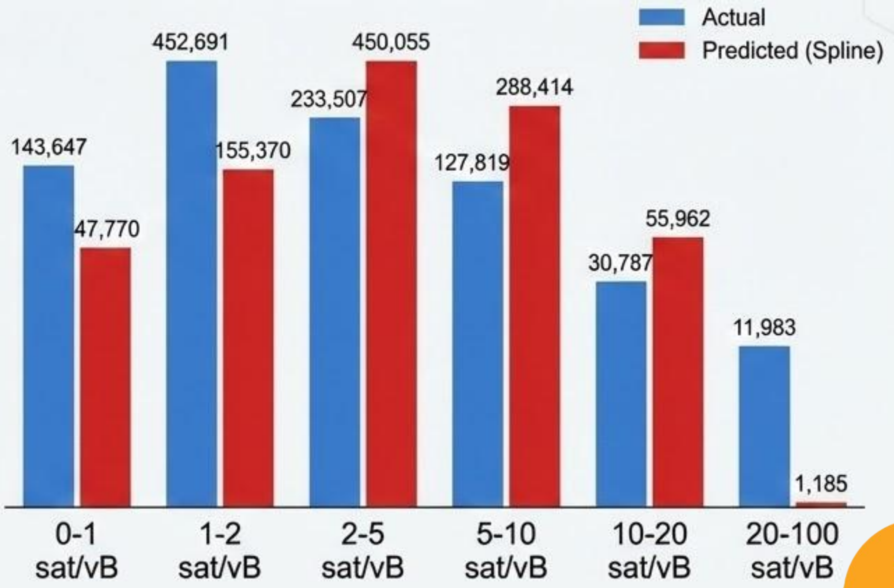
Bitcoin Fee Estimation: A Structural Model Approach

Spline Regression: Fee Rate Distribution Summary

Fee Rate Distribution Statistics

Metric	Actual	Predicted (Spline)
Total Transactions:	1,000,789	-
Mean (sat/vB):	3.09	4.56
Median (sat/vB):	2.00	3.96
Std Dev (sat/vB):	5.32	3.02

Transaction Count by Fee Rate Bracket



Bitcoin Fee Prediction: Scenario Analysis

Based on Input Parameters (Model 3: Spline Regression)

Scenario 1: Lower Mempool Density

INPUTS

rho_t	8,000
blockspace_t	0.3500
V_it	200,000
has_child	No (0) ✗
rbf_flag	Yes (1) ✓

PREDICTED FEE (sat/vB)

2.0992

pred_fee_spline_sat_vB

Scenario 2: Higher Mempool Density

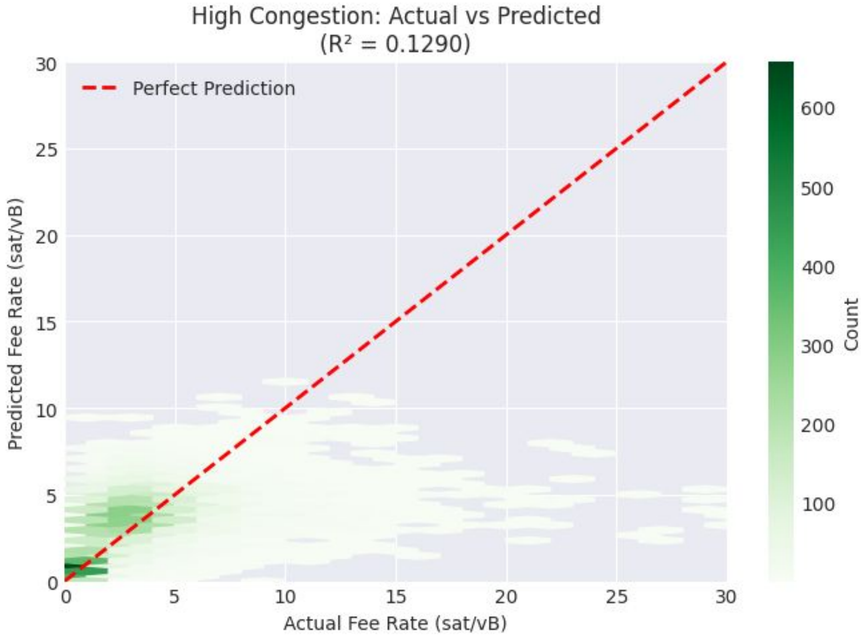
INPUTS

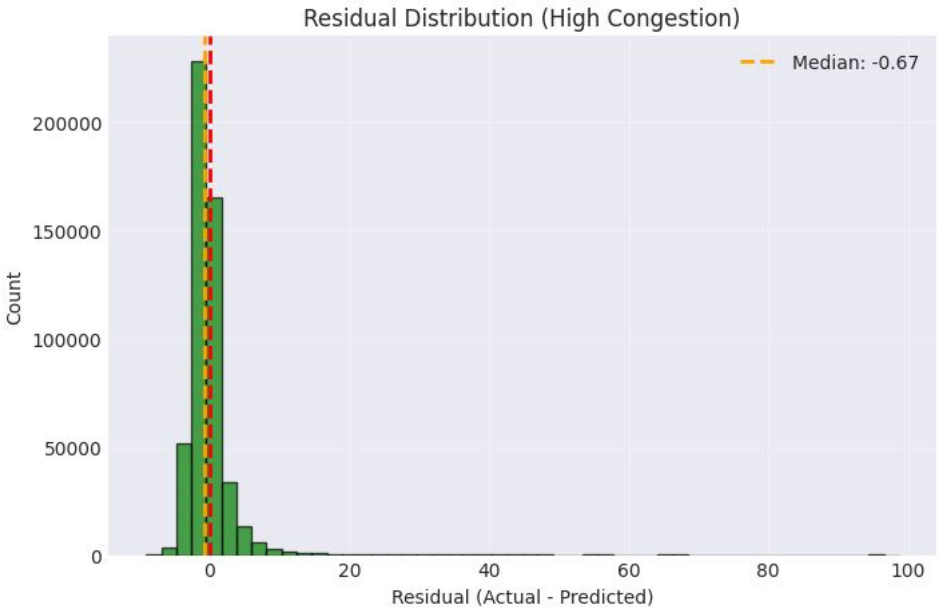
rho_t	60,000
blockspace_t	0.8500
V_it	5,000,000
has_child	Yes (1) ✓
rbf_flag	No (0) ✗

PREDICTED FEE (sat/vB)

1.2015

pred_fee_spline_sat_vB


High Congestion: Actual vs Predicted
(R² = 0.1290)
Perfect Prediction
Predicted Fee Rate (sat/vB)
Actual Fee Rate (sat/vB)
Count
0
5
10
15
20
25
30
100
200
300
400
500
600


Residual Distribution (High Congestion)
Median: -0.67
Count
Residual (Actual - Predicted)
0
50000
100000
150000
200000
20
40
60
80
100

SUMMARY: F-STATISTICS

LINEAR MODEL F-STATISTICS	NONLINEARITY?	SPLINE MODEL F-STATISTICS
log_V_it: 19505.97 (***)	YES (R2 Gain: 0.13%)	log_V_it: 95.70 (***)
has_child: 16780.39 (***)	N/A (binary)	has_child: N/A
log_rho_t: 5625.70 (***)	YES (R2 Gain: 0.66%)	log_rho_t: 475.86 (***)
log_blockspace_t: 1173.33 (***)	YES (R2 Gain: 0.17%)	log_blockspace_t: 123.70 (***)
log_time_cost_quantile: 657.31 (***)	YES (R2 Gain: 0.02%)	log_time_cost_quantile: 12.90 (***)
log_W_hat: 580.18 (***)	YES (R2 Gain: 0.15%)	log_W_hat: 106.83 (***)
rbf_flag: 104.35 (***)	N/A (binary)	rbf_flag: N/A
log_riemann_sum: 89.73 (***)	YES (R2 Gain: 0.05%)	log_riemann_sum: 33.08 (***)

*** $p < 0.001$

Summary and Conclusions

THE GOOD: FOUND THE SIGNAL

F-statistics are **massive for the top features**. This confirms that these **variables are definitely drivers of fee rates.**

log_V_it and **has_child** are the kings. They are 10x more important than almost anything else.

F = 20,850

log_V_it (Transaction Amount)

F = 19,078

has_child (CPFP Chains)

THE TAKEAWAY:
Transaction Amount and CPFP (Child-Pays-For-Parent) chains are the primary drivers of fees.

KEY TAKEAWAYS:

Modeling BTC fees
is hard (power law)

Machine Learning
Models would be better

Mempool congestion
hardly affects fee
rate at all

ACKNOWLEDGMENTS

Dan Aronoff
Economic insights and initial model

Armin Sabouri
Data collection and intuition

DCI
Funding