MEV

by Pad Deligiannis, Rohan Sanjay, Joshua Garbuio, Garret Nourse, and Lia Garrett

OUR TEAM

GARRET

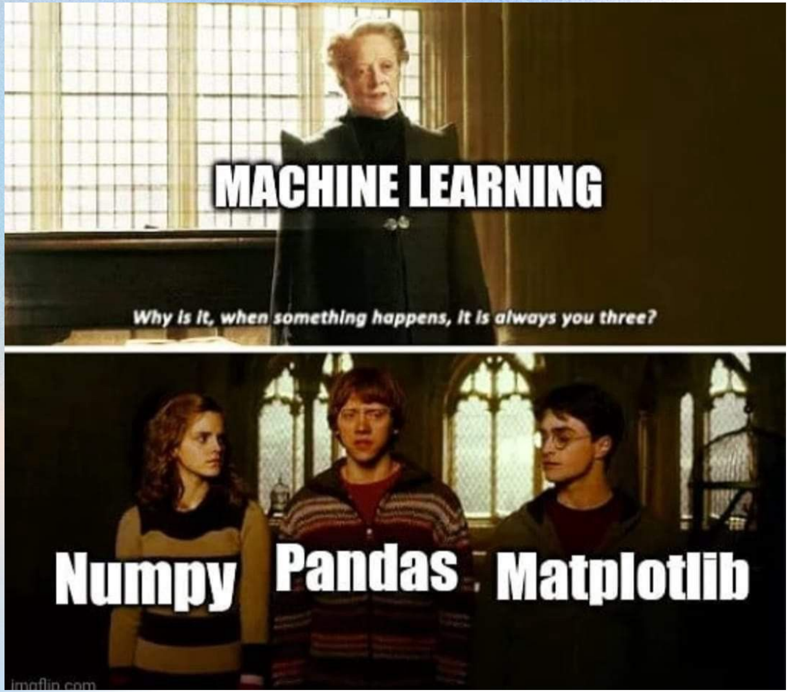
LIA

ROHAN

JOSH

PAD

Our Project



MACHINE LEARNING
Why is it, when something happens, it is always you three?
Numpy
Pandas
Matplotlib
imgflip.com

01

MEV Background

What is MEV

- MEV - “Miner Extractable Value”
- Based on Ethereum papers, miners had total control over transaction process
- MEV is widespread across all smart contract-enabled blockchains
- Key players are miners, searchers, users, and exchanges
- MEV totals more than 689 million in the network to date (Flashbots)
- Invisible “tax” on users

Common Tactics

- Front-running
- Sandwich attacks
- Back-running
- Liquidations
- Time-bandit

Implications for NFTs

- NFT transactions and Ethereum share blockchain
    - Searchers can use similar techniques as MEV
- MEV techniques in a NFT context:
    - Thrackle releases an NFT, or a set of NFTs, and a searcher wants to get it/them
    - They can program a transaction such that they are the first in line to buy it/them
    - They can buy an NFT/NFTs at a mistakenly low price

Examples

- CryptoPunks is a vendor of NFTs
  - They have a 10,000 collection, featuring humans, apes, zombies, and aliens
- One NFT, Punk 3860, was a male Punk with a cigarette, big shades, a shadow beard, and a mohawk
- It was sold on July 29th for $69,369
- The owner mistakenly made it public for under a cent
- Someone used MEV to beat everyone to the purchase
  - Spent 22 ETH to bribe Ethereum miner to prioritize their inclusion in the block
  - Did so with Flashbots communication
- There were other bids before this error, the max one at 35 ETH ($91,000)
- The MEV buyer recently put the NFT for sale in August
  - This could render a $74,000 profit

02

Flashbots Data

mev-blocks API

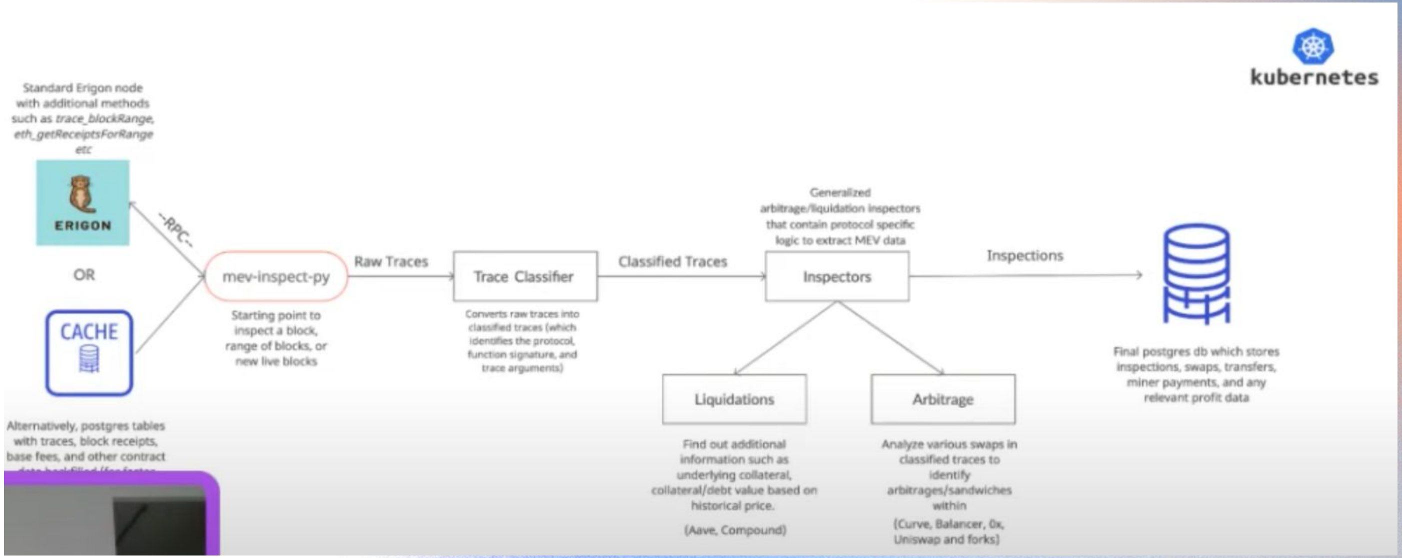
blocks.csv

- block_number
- miner_reward
- miner
- coinbase_transfers
- gas_used
- gas_price
- is_megabundle

transactions.csv

- transaction_hash
- tx_index
- bundle_type
- bundle_index
- block_number
- eoa_address
- to_address
- gas_used
- gas_price
- coinbase_transfer
- total_miner_reward

mev-inspect tool



mev-inspect data

arbitrages.csv

- 'id'
- 'created_at',
- 'account_address'
- 'profit_token_address'
- 'block_number'
- 'transaction_hash'
- 'start_amount'
- 'end_amount’
- 'profit_amount'

swaps.csv

- 'created_at'
- 'abi_name'
- 'transaction_hash'
- 'block_number'
- 'protocol'
- 'contract_address'
- 'from_address'
- 'to_address',
- 'token_in_address'
- 'token_in_amount'
- 'token_out_address'
- 'token_out_amount'
- 'trace_address'
- 'error'

transfers.csv

- 'created_at'
- 'block_number'
- 'transaction_hash'
- 'trace_address'
- 'protocol'
- 'from_address'
- 'to_address'
- 'token_address'
- 'amount'
- 'error'

miner_payments.csv

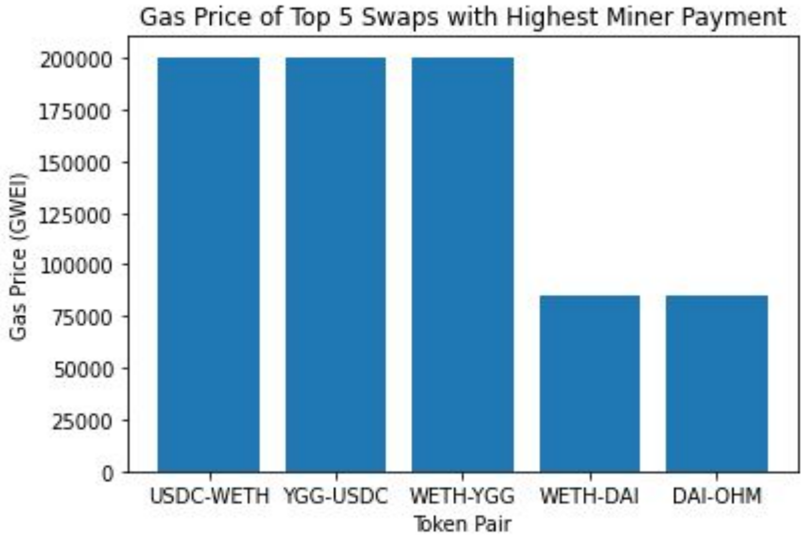
- 'created_at'
- 'liquidated_user'
- 'liquidator_user'
- 'debt_token_address'
- 'debt_purchase_amount'
- 'received_amount'
- 'protocol'
- 'transaction_hash'
- 'trace_address'
- 'block_number'
- 'received_token_address'

and more…

03

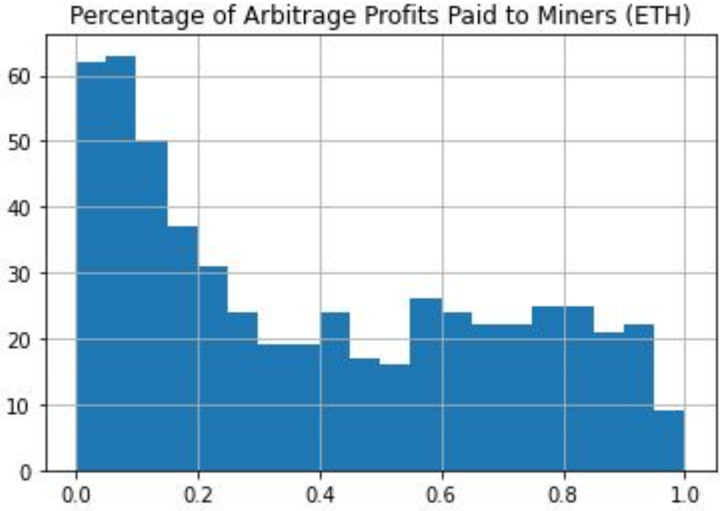
Analysis

High Miner Payment Swaps



token_pair	token_in_amount	token_out_amount
USDC-WETH	2.835889e+11	1.238484e+20
YGG-USDC	1.973861e+23	2.835889e+11
WETH-YGG	7.028764e+19	1.973861e+23
WETH-DAI	4.172835e+20	1.649611e+24
DAI-OHM	1.649611e+24	3.196099e+12

Arbitrage Profits Paid to Miners

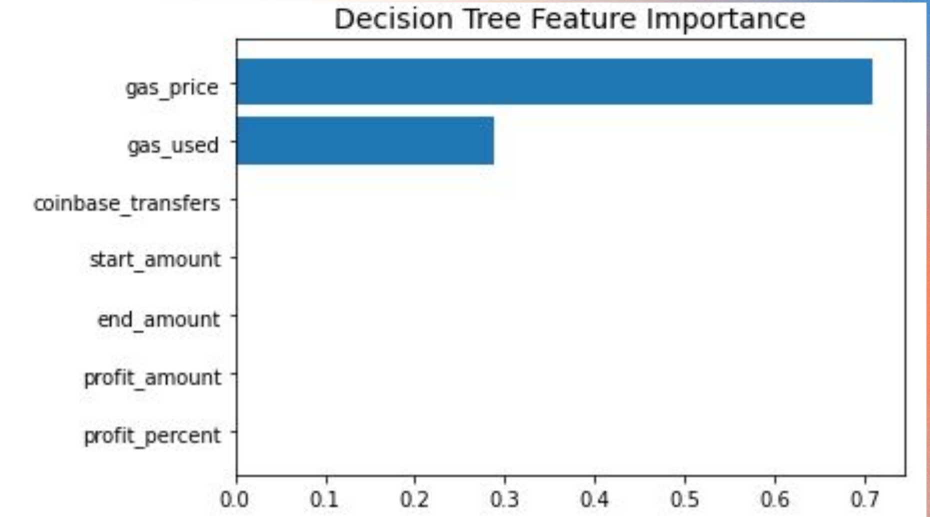


miner_payment/profit	profit_amount_eth	miner_payment_eth
1.000000	0.049068	0.049068
0.983859	1.636424	1.610011
0.969907	0.540054	0.523803
0.966530	0.677676	0.654995
0.966238	0.269022	0.259940

miner_payment/profit	profit_amount_eth	miner_payment_eth
0.000064	692.367899	0.044039
0.000121	91.271071	0.011037
0.900000	53.560708	48.204637
0.047613	19.520147	0.929423
0.959000	18.695878	17.929347

Predicting MEV

Using the combined datasets (the advent of MEV_insepct), a basic decision tree shows us that MEV can be reasonably predicted with gas price info, and that in terms of feature importance it ranks the highest



Random Forest Model

- K-folds validation, optimization for hyperparameter tuning of random forest model
- Model yielded successful results with an average accuracy level over k folds of 80.2%

```python
# Hyperparameter Tuning
prev_accuracy = .5
optimal_tree_num = 0
optimal_depth = 0
optimal_criteria = ''
optimal_split = 0
optimal_features = 0
for num_trees in [25, 40, 60, 90, 120, 300]:
  for depth in range(4,8):
    for max_features in range(3,6):
      for criteria in ['gini', 'entropy']:
        for split in [10, 15, 20, 50, 80, 120]:
          forest = RandomForestClassifier(n_estimators=num_trees, criterion=criteria, \
            max_features=max_features, min_samples_split=optimal_split, max_depth=depth)
          k_folds = cross_val_score(forest, X_train, y_train, cv=8)
          if k_folds.mean() > prev_accuracy:
            optimal_tree_num = num_trees
            optimal_depth = depth
            optimal_criteria = criteria
            optimal_features = max_features
            optimal_split = split
```

```
Random Forest Accuracy: 80.2%
```

04

ETH2

Research Question

How will the expected transition from ‘Proof of Work (PoW)’ to ‘Proof of Stake (PoS)’ consensus protocol with Eth2 on Ethereum affect MEV opportunities?

What is Eth2?

Proof of Stake (PoS)

Ethereum's consensus is secured by validators rather than by miners (PoW/eth1).

Validators

Stake security deposits of 32 ETH and vote to come to consensus on the state of the beacon chain. Economically incentivised via rewards and penalties for downtime or malicious behavior.

Beacon Chain

Eth2's new underlying PoS consensus mechanism.

Eth2

Post-merge Ethereum blockchain which contains the eth1 execution engine and the beacon chain for its consensus mechanism.

Eth2 Consensus Method

Epoch

Reaches a consensus in 6.4 minute increments called epochs. Each epoch contains 32 slots.

Slot

Each slot is 12 seconds long and is a chance for a block to be added to the beacon chain.

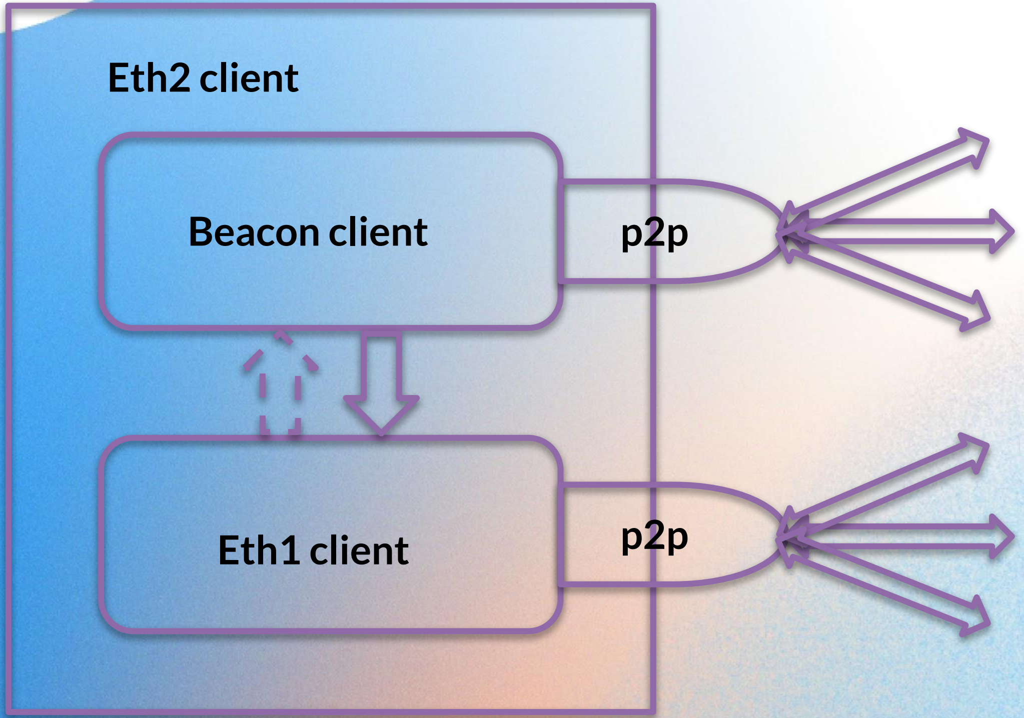
Assignment

For each epoch, validators are pseudo-randomly assigned to propose or attest blocks proposed by other validators.

Reward

One base reward is paid to each validator for accurately voting on the head of the chain, the justified checkpoint, and the finalized checkpoint. A 4th (inclusion reward) is given if their attestation is included into the beacon chain block.

Eth2 Client



Composed of two sub-clients (one for execution engine (eth1) and one for consensus (beacon)

They run concurrently, each maintaining their own p2p networking stack (libp2p for beacon and devp2p for eth1)

Eth1 Block Proposal

Eth1

Eth1 client in eth2 maintains a local mempool of transactions from its p2p network.

Eth1/ Eth2

The beacon client queries the eth1 client for an eth1 block and checks it meets validity conditions.

Eth2

It is then included by the proposer in the current beacon block and becomes part of the data attesters vote on.

Eth1/ Eth2

Beacon client then asks the eth1 client to update the head of its chain to the latest eth1 block included.

Eth1/ Eth2

The epoch is ‘finalized’ and the beacon client notifies the eth1 client that the eth2 block has been finalized at the consensus layer.

Does MEV exist in eth2?

The transaction ordering process remains the same in eth1 and eth2, therefore MEV opportunities will still exist. The difference lies in who has ultimate control over the ordering, namely the validators rather than the miners, who have been selected to propose a beacon block which will contain a eth1 block queried from the eth1 client.

05

Analysis of MEV rewards in ETH2

Adapted from Pintail and Flashbots code.

Assumptions

Ideal case:

Where all validators participate perfectly and get the maximum protocol reward possible (no slashing), and all rewards are distributed evenly since all validators stand to propose the same amount of blocks on an infinite scale.

Caveat

Transaction fees are not included in the calculation of yield. Therefore the relative impact of MEV on staking yields is overstated.

Defining Base Rewards

$$base\ reward = \frac{effective\ reward * base\ reward\ factor}{base\ reward\ per\ epoch * \sqrt{total\ balance}}$$

Use ideal case values and convert to ETH

$$base\ reward = \frac{32 * 64}{4 * \sqrt{n*32e9}} = \frac{512}{\sqrt{n*32e9}}$$

Annualize formula

$$annualized\ base\ reward = epochs\ per\ year * base\ reward = \frac{82180 * 512}{\sqrt{n*32e9}}$$

Term	Meaning	Value in Ideal Case
Effective balance	An integer number of ETH according to each validator's balance (Gwei)	32e9
Base reward factor	Factor to tune the overall issuance of ETH	Constant value of 64 in eth2 spec
Total balance	Sum of effective balance for all active validators	N * 32e9
Base rewards per epoch	Corresponds to the 4 components of validator reward, each worth one base reward, which can be earned by each validator, each epoch.	Constant value of 4

Defining Ideal Rewards With and Without MEV

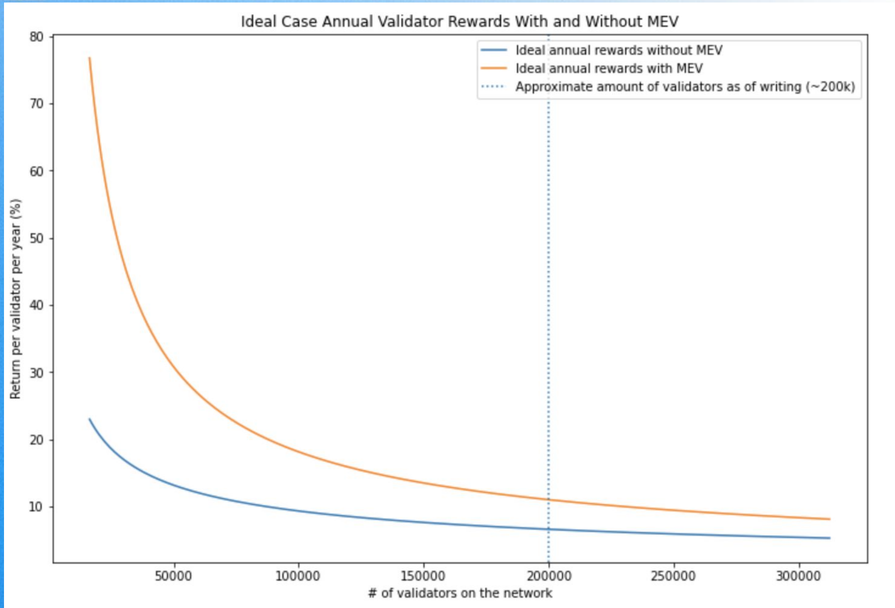
Term	Meaning	Value in Ideal Case
Ideal reward	Maximum reward per validator	4 * base reward
Avg mev reward per block	Obtain by subtracting the Flashbots miner tip from the tail gas price multiplied by the gas by the mined Flashbots bundle.	Constant value of 64 in eth2 spec
Avg blocks proposed per year	Seconds per year / (seconds per slot * n)	Seconds per slot = 12 Seconds per year = 31556952
Block selection frequency	% of blocks seen by Flashbots-enabled miners contains Flashbots bundles	58

$$ideal\ reward \;=\; annualized\ base\ reward \;*\; 4$$

$$ideal\ reward\ with\ MEV \;=\; ideal\ reward \;+\; (avg\ mev\ per\ year \;*\; \frac{block\ selection\ frequency}{100}),$$

$$avg\ mev\ per\ year \;=\; avg\ mev\ reward\ per\ block \;*\; avg\ blocks\ proposed\ per\ year$$

Annual Validator Rewards



Current # of active validators

200,000 as of July 2021

Difference between orange and blue curve

Increase of validator rewards from MEV

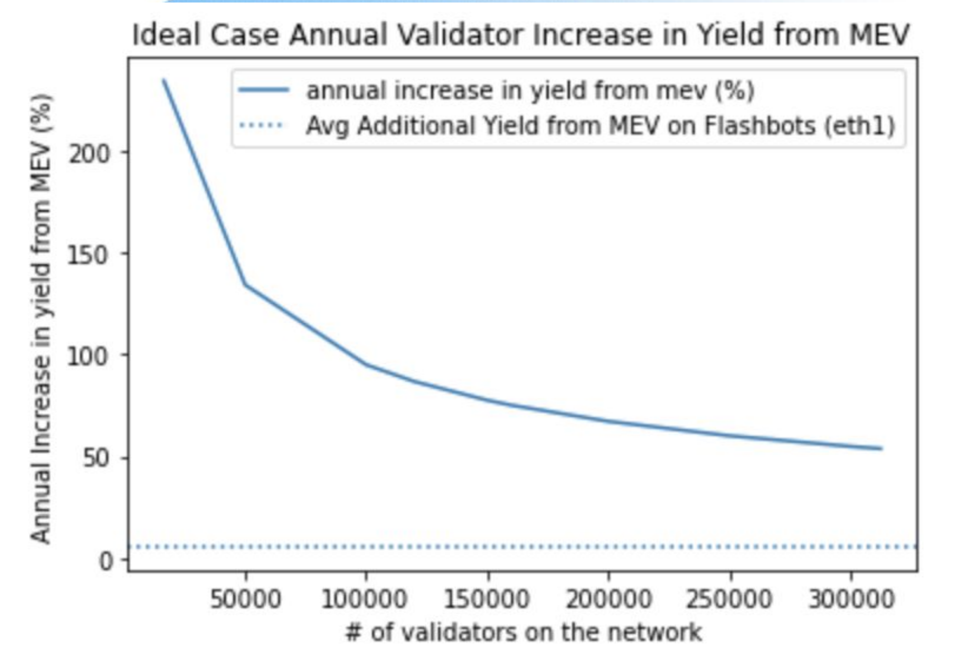
Annual Validator Increase in Yield

@ 200k~ increase in yield by 67.08%

Higher rewards indicate incentives for eth holders to be a validator, in turn making Ethereum more secure.

# of validators	annual reward (ETH)	annual reward with mev (ETH)	annual validator yield (%)	annual validator yield with mev (%)	annual increase in reward from mev (ETH)	annual increase in yield from mev (%)
16384	7.35	24.57	22.97	76.79	17.22	234.31
50000	4.21	9.85	13.15	30.78	5.64	134.12
100000	2.98	5.80	9.30	18.12	2.82	94.84
120000	2.72	5.07	8.49	15.84	2.35	86.58
150000	2.43	4.31	7.59	13.47	1.88	77.44
160000	2.35	4.12	7.35	12.86	1.76	74.98
200000	2.10	3.51	6.57	10.98	1.41	67.06
250000	1.88	3.01	5.88	9.41	1.13	59.98
300000	1.72	2.66	5.37	8.31	0.94	54.76
312500	1.68	2.59	5.26	8.08	0.90	53.65

Annual Validator Increase in Yield



5.6%

Average additional yield from MEV on Flashbots (eth1).

Yield vs # of validators

As the number of validators increase, yield increases become less significant.

Variability in Rewards

Find the distribution of luck per block proposal frequency in a year using the binomial distribution formula.

Every validator has an equal chance of being selected to propose each slot, if n = 100,000, chance of being selected = 10^-5

2629746 slots per year

$$P(X) = \binom{2629746}{k} \cdot (10^{-5})^k (1 - 10^{-5})^{2629746-k}$$

Method

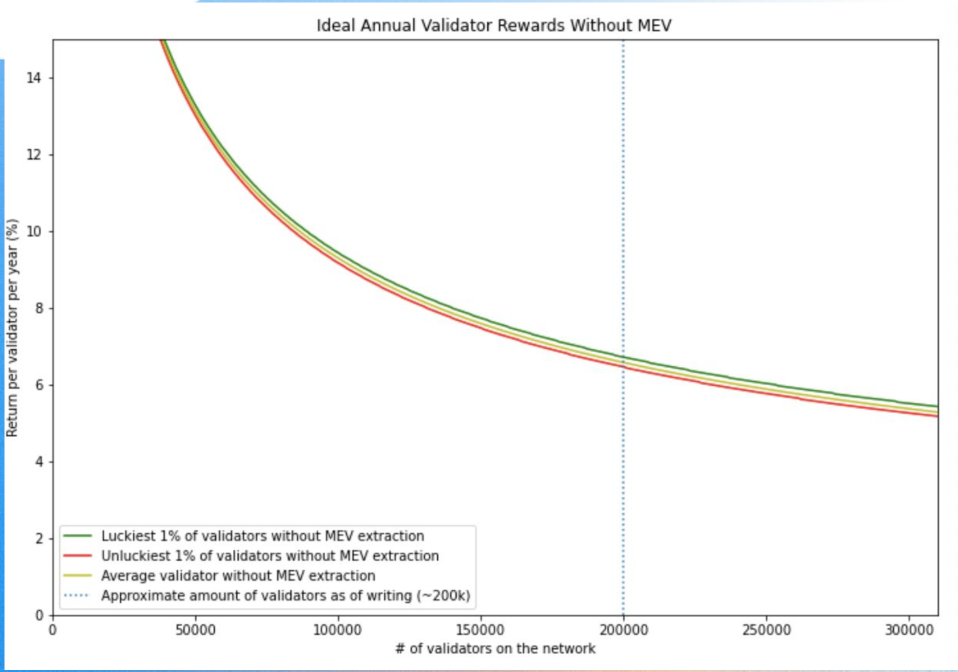
Calculate the 1% unluckiest, 1% luckiest, and median

binom.ppf(0.01, 0.5, 0.99, 31556952/12, 1e-5)

Repeat for varying values of validators

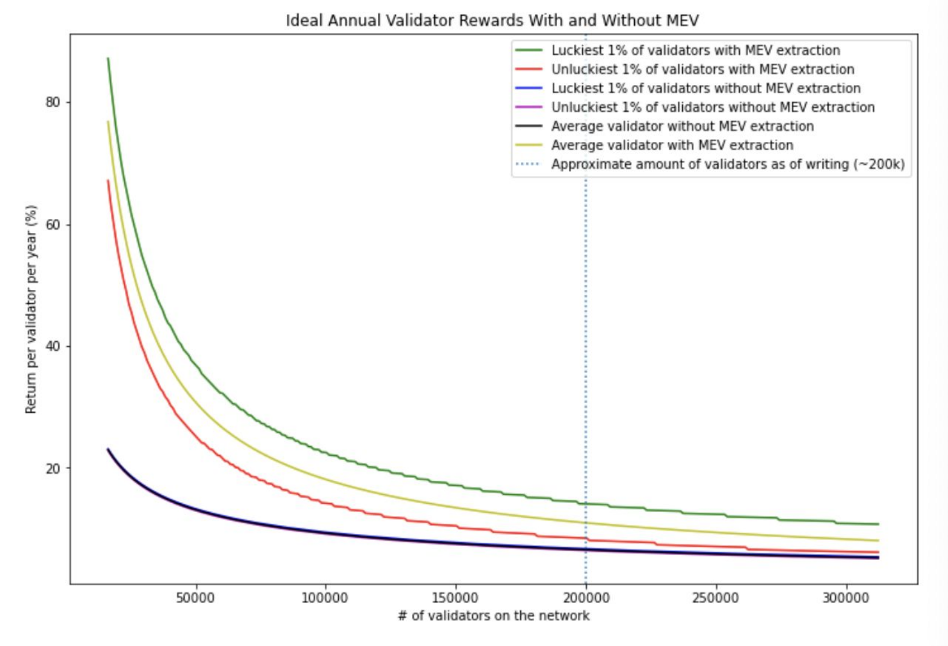
binom.ppf(0.01, 0.5, 0.99, 31556952/12, 1/n)

Plot of Maximum Validator Rewards over Finite Timescale



Block-proposal luck influence without MEV

Plot of Maximum Validator Rewards over Finite Timescale



Block-proposal luck influence with and without MEV

MEV rewards and its uneven distribution amplified the inequality between luckiest and unluckiest validators with respect to block proposal luck.

Conclusion

While Eth2 will make reordering and attacks more difficult by requiring 50% of all validators to agree on a vote and the random assignment and 12 second windows makes it harder for miners to capitalize on MEV, it will open opportunities for validators to make significantly more in MEV. The shift in control for reordering essentially shifts the downfalls and advantages of MEV onto the validators. The higher rewards could discourage individual validators as the variability in block proposal luck makes it advantageous to join a pool for predictable and earlier liquidity. This could possibly lead to enriching the entities with the most 32-eth stakes faster (rich get richer dynamics) which would destroy the decentralization of consensus voting power.

06

Deliverable

github.com/rohansanjay/mev

Thank you!