

Do Rollups Self-Organize? Multi-Agent Reinforcement Learning in the EIP-4844 Blob Fee Market

Alessio Brini
Pratt School of Engineering
Duke University
Durham, NC, USA
alessio.brini@duke.edu

Abstract—EIP-4844 gave Ethereum rollups a separate fee dimension for data, governed by a multiplicative update rule that targets three “blobs” per block. Whether this target is reached depends on how many independent, self-interested rollups choose to post data, a question that closed-form models address only for a single agent under exogenous prices or in a static equilibrium. We propose a calibrated multi-agent reinforcement learning environment for the blob fee market in which the blob base fee is endogenous, set by the EIP-4844 rule on the aggregated decisions of heterogeneous rollups fit to post-Dencun Ethereum data. After confirming that the environment reproduces the known single-agent analytical optimum, we show that independent learners, with no convergence guarantee, still settle near the target without coordination (3.32 ± 0.01 blobs per block over five seeds). Over a fiftyfold demand range ($10\times$ to $500\times$), supply stays at the target from $10\times$ to $200\times$ and rises above it at higher demand, where the fee can no longer hold it. Posting frequency increases with rollup size, an emergent analog of the size threshold predicted by prior theory. A costless layer-1 calldata outside option pushes the fee down by roughly an order of magnitude while aggregate supply stays at the target, so the fee rule acts as a negative-feedback controller on decentralized supply.

Index Terms—blockchain, EIP-4844, fee markets, multi-agent reinforcement learning, Ethereum, rollups, mechanism design

I. INTRODUCTION

The Dencun upgrade (March 2024) activated EIP-4844 (Ethereum Improvement Proposal 4844), which gave Ethereum rollups a dedicated data channel, *blobs*, priced on a fee dimension separate from ordinary gas. The blob base fee follows a multiplicative update rule designed to hold realized usage at a target of three blobs per block. From a resource-market standpoint this is a control system in which the price steers aggregate usage back to the set point. Whether the set point is reached, however, is not a property of the rule alone. It depends on how many independent rollups, each minimizing its own posting and delay cost, respond to the price they collectively create.

Existing theory characterizes a single rollup’s optimal posting policy under an *exogenous* price process [1]–[3], and field studies document blob-market behavior empirically [4]–[7]. Neither line answers the equilibrium question directly. Once the fee is endogenous in the agents’ joint actions, the single-agent optimality arguments no longer apply. Relatedly, [8]

characterize the choice between blobs and calldata, together with blob cost-sharing, in a static market equilibrium, but closed-form equilibrium strategies for the dynamic multi-agent regime with an endogenous fee are lacking. We take a computational route and ask whether the protocol’s intended operating point *emerges* when self-interested rollups learn how to post.

We make three contributions.

- We propose a calibrated multi-agent reinforcement learning (RL) environment for the EIP-4844 blob market in which the blob base fee is endogenous, set by the update rule on the aggregated blob decisions of N heterogeneous rollups whose arrival rates and price processes are fit to post-Dencun Ethereum data (Sections III and IV).
- We show that independent proximal policy optimization (PPO) [9] learners self-organize to a steady state near the protocol target without any coordination mechanism. Posting frequency rises monotonically with rollup size, an endogenous analog of the arrival-rate threshold of [8], and the empirical steady state stays at the target across a twentyfold range of aggregate demand (phase diagram, Section V).
- We show that adding a costless layer-1 (L1) calldata¹ outside option collapses the fee level by roughly an order of magnitude while leaving aggregate supply pinned at the target, so the rule reaches its set point independently of the price level.

The environment also reproduces, as a sanity check, the single-agent analytical threshold of [1]. All code and run configurations are at <https://github.com/AlessioBrini/eip4844-blob-marl>, including the calibrated parameters and the BigQuery queries that regenerate the raw on-chain data.

II. RELATED WORK

Posting strategies under exogenous prices. In the single-agent setting, [1] derive threshold publishing policies under

¹Calldata is the pre-EIP-4844 channel for rollup data, posted in the payload of an L1 transaction. It remains available and is cheaper than blobs only when the blob fee is well above its floor.

a broad class of price dynamics that includes martingales, and [2] extend this analysis to fee-elastic Poisson arrivals under a mean-reverting L1 gas price, where the optimal queue threshold depends on the gas price. Earlier, [3] design rollup batch-posting algorithms with operations-research tools. All assume the rollup is a price taker. Our environment makes the blob fee endogenous in the agents' joint actions.

Blob-market economics and empirics. [8] analyze EIP-4844 economics and predict an arrival-rate threshold below which a (smaller) rollup should defect to calldata. Field studies [4], [5] characterize the early blob fee market, while [6] document builder suboptimality and [7] documents the unrealized potential of blob sharing. We complement these by treating the market as a learning system and asking whether its steady state is the one the protocol intends.

Learning in fee markets. Multi-agent RL with independent learners [10] and agent-based market simulators [11] are established tools for studying emergent behavior in markets. The blob fee adopts the EIP-1559 congestion-pricing design (a base fee that adjusts toward a target, here in fully exponential form rather than the linear approximation used for gas), whose economics [12] and convergence dynamics [13] are well studied. We apply those tools to a resource market with a protocol-defined target, asking whether the set point is reached.

III. MODEL AND ENVIRONMENT

One time step is one L1 block (≈ 12 s). The market hosts N heterogeneous rollups. Rollup i maintains a queue $Q_{i,t}$ that grows each block by a Poisson draw with mean λ_i and observes the local state

$$s_{i,t} = (Q_{i,t}/\tau_i, B_t, P_t^{\text{gas}}), \quad (1)$$

where B_t is the blob base fee (wei), P_t^{gas} the L1 gas base fee, and τ_i the rollup's transactions-per-blob capacity ($\tau_i = 2^{17}/d_i \approx 262$ for the homogeneous $d_i = 500$ bytes per transaction we use). Dividing the queue by τ_i expresses it in blob units, keeping the observation order-one under arrival-rate scaling. The two price components enter in log form. We otherwise leave the observation unstandardized, with advantages normalized per update. Each rollup selects an action $a_{i,t} \in \{0, 1, \dots, 6\}$, the number of blobs to post in block t (0 = wait). A post of $a_{i,t}$ blobs removes $\min(a_{i,t} \tau_i, Q_{i,t})$ transactions from the queue. An underfilled blob still occupies, and pays for, a full blob.

Endogenous blob base fee. After every block the fee updates through the EIP-4844 rule² applied to the *aggregate* blobs posted by all rollups,

$$B_{t+1} = B_t \cdot \exp\left(\frac{b_t - b^*}{8b^*}\right), \quad b_t = \sum_{i=1}^N b_{i,t}, \quad (2)$$

²The spec accumulates excess blob gas across blocks and sets the fee to $\text{MIN} \cdot \exp(\text{excess}/F)$. Equation (2) is the per-block reduction of that recursion, and its denominator $8b^* = 24$ differs by $\sim 6\%$ from the spec's effective $F/2^{17} \approx 25.5$, without affecting the qualitative results.

with $b_{i,t}$ the blobs rollup i posts ($a_{i,t}$ capped by queue availability), target $b^* = 3$, and cap $b^{\text{max}} = 6$.³ Equation (2) couples the rollups, so no agent sees the fee as exogenous. The L1 gas base fee follows a mean-reverting first-order autoregressive (AR(1)) process fit to data (Section IV).

Cost and reward. A rollup pays a delay cost on its standing queue and a posting cost for the blob gas and execution gas it consumes,

$$r_{i,t} = -\left(\underbrace{\alpha_i Q_{i,t}}_{\text{delay}} + \underbrace{B_t b_{i,t} \cdot 2^{17} + P_t^{\text{gas}} g^{\text{fix}}}_{\text{posting}}\right), \quad (3)$$

where $2^{17} = 131,072$ is the blob gas per blob and g^{fix} the fixed execution gas of a posting transaction. Delay (dimensionless queue) and posting (wei) costs live in different units. We reconcile them by a revealed-preference calibration of α_i that normalizes both by a reference posting cost $C_i^{\text{ref}} = \bar{P}^{\text{gas}} g^{\text{fix}}$, giving a dimensionless effective delay weight $\tilde{\alpha}_i = \alpha_i / C_i^{\text{ref}} = 2\lambda_i$.

Outside option. In the calldata variant (Section V) a rollup may instead drain its full queue through a single L1 calldata transaction at cost $P_t^{\text{gas}}(g^{\text{fix}} + 16 d_i Q_{i,t})$. Calldata uses no blob gas, so it does not enter (2). In our runs both execution-gas terms, here and in (3), are negligible, so calldata is effectively costless.

Learners. Each rollup runs an independent PPO policy, a multilayer perceptron actor-critic with two hidden layers of 64 units and tanh activations, learning rate 3×10^{-4} , clip coefficient 0.2, generalized advantage estimation with $\lambda_{\text{GAE}} = 0.95$, discount $\gamma = 0.99$, rollouts of 2,000 blocks per update with four epochs of minibatches of 256, value and entropy loss coefficients 0.5 and 0.01, and gradient-norm clip 0.5. The implementation is PyTorch, and each 10^5 -step run trains in about a minute on a laptop CPU. There is no shared critic, communication, or reward sharing.

IV. CALIBRATION

We calibrate on block-level Ethereum data from Google BigQuery's `crypto_ethereum` dataset over the post-Dencun window (March 13 to August 31, 2024), covering 1,218,246 blocks. We fit the L1 gas base fee and the log blob base fee each as an AR(1) process. The log parameterization for the blob fee matches the multiplicative structure of (2). Table I reports the fits. Both processes are highly persistent, the blob fee far more so.

We estimate per-rollup arrival rates λ_i from type-3 transactions grouped by batcher address, counted over March 2024 to March 2025 and divided by the window's block count. These are L1 posting events, not the underlying layer-2 transaction arrivals, which for large rollups run to tens or hundreds per block. The calibrated λ_i is thus a posting

³There is no builder-side inclusion rule. When aggregate requests exceed b^{max} , the fee update counts $\min(b_t, b^{\text{max}})$ while queues drain and costs accrue in full. In rollouts of the trained policies, the cap binds on one block in 10^5 at $100\times$ demand, on 0.03% of blocks at $200\times$, and on 8.4% (mean excess 1.7 blobs) at $500\times$, so this simplification affects only the saturation numbers.

TABLE I
CALIBRATED AR(1) PRICE PROCESSES (MARCH TO AUGUST 2024).
HALF-LIFE IS IN BLOCKS.

Process	μ	θ	σ	R^2	Half-life
Gas fee (ETH/gas)	1.09×10^{-8}	2.140×10^{-3}	9.85×10^{-10}	0.996	324
Blob fee (log-wei)	1.533	1.242×10^{-4}	0.081	0.9998	5,582

intensity. It enters the simulation as the mean of the Poisson arrivals feeding $Q_{i,t}$ and pins down relative sizes, while the demand level is set by the multiplier below, so results are read per demand regime. We label 18 rollups covering 86.3% of post-Dencun blob-posting transactions. Their rates are well described by $\lambda_i \sim \text{LogNormal}(-2.78, 1.18)$ (Kolmogorov–Smirnov $p = 0.98$). The five largest, namely Taiko, Base, Arbitrum One, Scroll, and World Chain (rates in Table III), anchor the $N = 5$ experiments.⁴ We use these five for the per-rollup analysis and confirm generality on the 18-rollup roster below.

Demand scaling. At empirical rates the top rollups in our roster post below the three-blob target (about 1.3 blobs per block in simulation), so the fee sits at its floor and posting every block is trivially optimal, consistent with the sub-target demand and near-floor fees documented post-Dencun [4], [5]. The economically active regime is the forward-looking one, reached as rollup adoption grows, the growth that data-availability upgrades (Pectra, PeerDAS [14]) are designed to accommodate. We therefore scale every λ_i by a common multiplier, holding α_i at its empirical calibration, and treat the high-multiplier regime as a stress test of the fee rule, reporting the steady-state supply across a wide range of multipliers.

V. RESULTS

We train independent PPO learners to a steady state ($\sim 10^5$ environment steps, 5×10^4 for the phase diagram and the target variation) over five random seeds and report each steady-state statistic as a mean $\pm$ s.d. across seeds, computed over the final 20% of rollouts.⁵

Single-agent validation. Before the multi-agent results we confirm the environment reproduces known single-agent behavior (Table II). A deep Q-network (DQN) [15] agent⁶ matches the threshold policy of [1] under a log-normal random-walk price, posts every block at the fee floor, and times the market under extreme congestion.

Reaching the target. Fig. 1 shows the calibrated and the congested regime for $N = 5$. Under $100\times$ demand the dynamics become economically non-trivial. The five independent

⁴Taiko is a based rollup posting one blob per transaction by design. Taiko, World Chain, and Zircuit (18-rollup roster) launched mid-window, so their rates also average over blocks before launch. The target/cap are the Dencun-era values. Section VI scopes the results to this historical configuration.

⁵We judge convergence by the ordinary-least-squares slope of the aggregate blobs-per-block series over this window, which is negligible relative to its mean.

⁶The DQN is a Q-network with two hidden layers of 64 units and rectified-linear-unit activations, learning rate 10^{-4} , replay buffer 10^5 , batch 64, target network refreshed every 1,000 steps, ϵ -greedy decayed over the first 30% of training, $\gamma = 0.995$ (0.99 for the last two regimes of Table II).

TABLE II
SINGLE-AGENT VALIDATION (EXOGENOUS-PRICE SPECIAL CASE). THE ENVIRONMENT MATCHES THE ANALYTICAL THRESHOLD POLICY AND TIMES THE MARKET ONLY WHEN IT SHOULD.

Regime	DQN posting freq.	Reference
Random-walk price [1]	0.46	0.48 (threshold policy)
Calibrated fee floor	1.00	1.00 (always-post opt.)
Extreme congestion	0.64	$8.8\times$ cheaper (always-post)

learners settle to 3.32 ± 0.01 blobs per block, slightly above the protocol target $b^* = 3$, and the fee holds on the order of 10^{10} to 10^{11} wei, far above the 1-wei floor but below the simulation’s numerical fee ceiling. The endogenous feedback of (2) alone drives the population to the set point, and the tight band across seeds rules out a lucky seed.

Non-learning baselines. Rule-based policies in the same congested environment (five seeds, 2×10^4 blocks) separate the fee rule’s contribution from learning. Price-blind rules miss the target from both sides. Draining the queue every block yields 5.00 blobs per block with the fee pinned at the numerical ceiling, and posting only full blobs yields 0.63 with the fee at its floor. A myopic threshold that posts when the delay cost exceeds the posting cost at current prices, a rule that knows the cost model, also reaches the target (3.29, s.d. < 0.01). The controller thus pins the aggregate whenever supply responds to price. Learning recovers the price response from rewards alone, at higher realized cost than the informed rule (236 to 715 versus 221 in normalized units).

Strategy ordering in λ_i . In the congested regime, posting frequency increases with the rollup’s arrival rate (Table III). Averaged across seeds, the ordering is strictly monotone (and holds in four of five seeds individually). The largest rollup (Taiko) posts in 97% of blocks, the smallest (World Chain) in 36%. Low-volume rollups thus wait out most blocks, shrinking their realized blob footprint. This is an endogenous, posting-frequency form of the calldata arrival-rate threshold of [8], here without an explicit outside option. Since the delay weight $\tilde{\alpha}_i = 2\lambda_i$ also scales with size, we retrain with a common $\tilde{\alpha}_i$ set from the roster-mean rate. Aggregate supply is nearly unchanged (3.39 ± 0.02 versus 3.32 ± 0.01) and Taiko still posts most (0.80 ± 0.05), but the gradient flattens (0.80 to 0.62) and its ordering is not stable across seeds, so the steep gradient of Table III reflects arrival volume and the calibrated delay pricing jointly.

Phase diagram. Fig. 2 varies the arrival-rate multiplier from $10\times$ to $500\times$ (fiftyfold) at $N = 5$ (five seeds per point). The steady-state supply stays within one blob of the target for multipliers from 10 to 200 (3.29 to 3.43 blobs per block), then rises to 4.29 at $500\times$, where the fee approaches the simulation’s numerical ceiling and can no longer price-ration supply to the target, the regime in which a strategic builder or an outside option becomes economically necessary. Scaling the roster to all 18 calibrated rollups (at $300\times$) yields the same picture (Fig. 3). Aggregate supply settles at 3.66 ± 0.01 blobs per block, and posting frequency broadly increases with

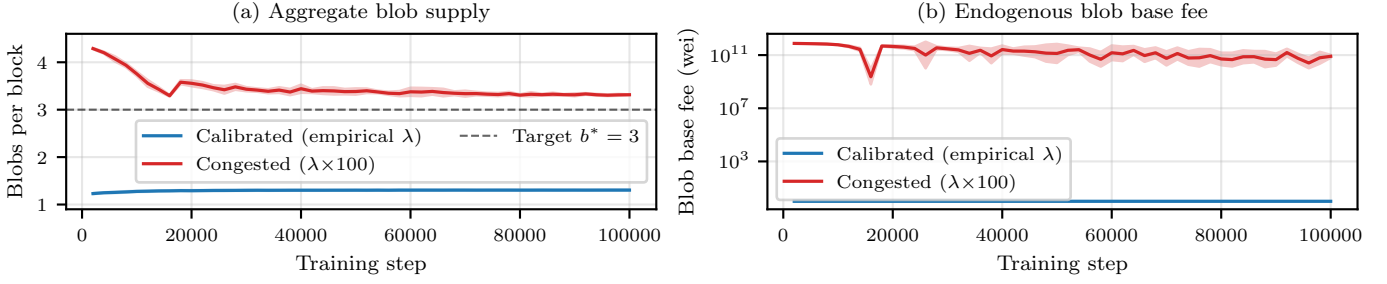


Fig. 1. Multi-agent training dynamics, $N = 5$ (each curve is the mean across five seeds, shaded $\pm$ s.d.). (a) Aggregate blob supply against the EIP-4844 target $b^* = 3$, at empirical arrival rates (blue) and under $100\times$ demand (red). (b) Endogenous blob base fee (log scale).

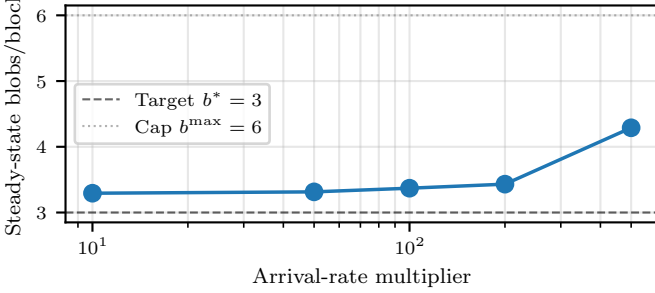


Fig. 2. Steady-state blob supply versus the arrival-rate multiplier, $N = 5$ (mean $\pm$ s.d. over five seeds), against the target $b^* = 3$ and the cap $b^{\max} = 6$.

TABLE III

LEARNED POSTING FREQUENCY, $N = 5$ CONGESTED REGIME ($100\times$ DEMAND), MEAN $\pm$ S.D. OVER FIVE SEEDS. POSTING FREQUENCY IS THE FRACTION OF BLOCKS IN WHICH A ROLLUP POSTS AT LEAST ONE BLOB. THE AGGREGATE (LAST ROW) IS THE MEAN TOTAL BLOBS PER BLOCK MEASURED DIRECTLY FROM THE RUNS (A POSTING ROLLUP MAY PLACE SEVERAL BLOBS).

Rollup	λ_i (tx/block)	Posting freq.
Taiko	0.803	0.97 ± 0.02
Base	0.361	0.84 ± 0.06
Arbitrum One	0.218	0.67 ± 0.07
Scroll	0.153	0.47 ± 0.05
World Chain	0.125	0.36 ± 0.03
Aggregate blobs/block		3.32 ± 0.01

λ_i , strictly so among the 13 largest rollups, while the four smallest post more often than the next few.

Calldata outside option. Re-running the $N = 5$ congested regime with the calldata action enabled (Fig. 4) separates an aggregate effect stable across seeds from a distributional one that is not. The stable effect is twofold. The fee level collapses by roughly an order of magnitude (geometric-mean fee $\sim 10^{10}$ wei with the option versus $\sim 10^{11}$ without), while the aggregate blob supply stays pinned near the target at 3.20 ± 0.08 per block (Equation (2) drives the aggregate to target as long as some rollups use blobs). Which rollups defect varies with the seed. Across five seeds every rollup uses calldata 31 to 39% of the time, with no monotone size ordering, so we draw the size-threshold conclusion only from the posting-frequency gradient, not from calldata defection.

VI. DISCUSSION AND LIMITATIONS

Feedback on the aggregate. The target is an attractor of the fee dynamics. The exponent in (2) depends only on the total $b_t = \sum_i b_{i,t}$, so the fee rises when aggregate posting exceeds b^* and falls when posting is below it, whatever the posters' identity. The rule has a stable fixed point at b^* . That uncoordinated learners actually settle there, however the blobs are split, is the empirical content. The steady state sits slightly above b^* (3.32 versus 3) because the fee restarts at 1 wei every 2,000-block episode, and by (2) climbing back takes excess supply. The arrival-rate threshold surfaces here on the *intensive* margin (Table III) rather than on the *extensive* margin of a binary calldata defection (Fig. 4).

Varying the target. Holding demand fixed at $200\times$ and varying $b^* \in \{2, 3, 4\}$, the steady-state supply moves one-for-one with the target (2.38 ± 0.01 , 3.43 ± 0.03 , and 4.47 ± 0.02), each 0.38 to 0.47 above target. So b^* sets the operating point, while $8b^*$ in (2) governs only adjustment speed.

Protocol scope. All experiments use the Dencun-era configuration ($b^* = 3$, $b^{\max} = 6$) in force over the calibration window. Pectra (May 2025) raised the pair to 6/9 [16], blob-parameter-only forks [17] after the Fusaka upgrade reached 14/21 in January 2026, EIP-7918 now bounds the blob base fee from below by execution cost [18], and EIP-7623 raised calldata costs [19]. The one-for-one response of supply to the target above suggests the picture transfers to larger targets, but the calldata fee drop is specific to a market without a reserve price, and our calldata option is effectively costless.

Several limitations bound these claims. Convergence is empirical, not by theorem (no general guarantee for independent-learner multi-agent RL [20]). The tight bands across seeds (± 0.01) are our evidence. The L1 gas process is exogenous, and the builder is a rule-based pass-through that abstracts away builder-side strategy, empirically often suboptimal [6]. The calldata option is effectively costless. Natural next steps are a learning builder and a cooperative blob-sharing mechanism.

VII. CONCLUSION

We presented a calibrated multi-agent RL environment for the EIP-4844 blob fee market with an endogenous fee. Across a wide demand range, independent PPO learners self-organize to the protocol target without coordination, with posting frequency ordered by rollup size. A costless calldata outside

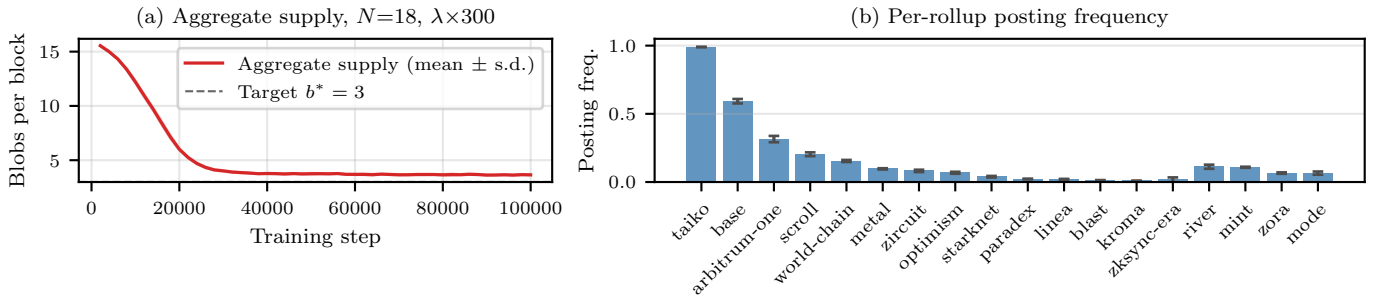


Fig. 3. Generality across roster size, with all 18 calibrated rollups at $300\times$ demand (mean over five seeds, shaded/error bars $\pm$ s.d.). (a) Aggregate supply against the target $b^* = 3$. (b) Per-rollup posting frequency, ordered by decreasing λ_i .

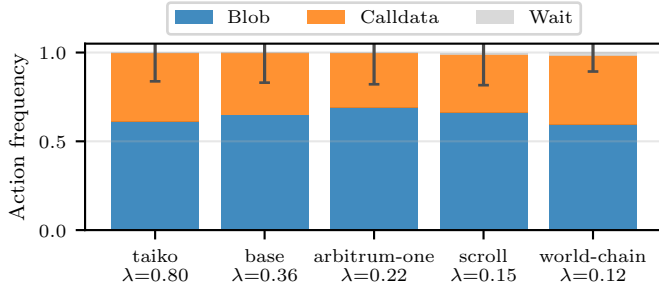


Fig. 4. Per-rollup action mix in the congested regime with a calldata outside option (mean over five seeds, error bars $\pm$ s.d. on the calldata fraction), ordered by decreasing λ_i .

option drops the fee level while supply holds at target. The fee rule thus acts as a self-stabilizing congestion signal for learning rollups, a property invisible to single-agent, exogenous-price analysis, and the calibrated environment is a reusable testbed for vetting fee-rule changes before deployment, with b^* a clean design knob.

REFERENCES

- [1] Y. Bar-On and Y. Mansour, “Optimal publishing strategies on a base layer,” in *Financial Cryptography and Data Security (FC 2024)*, Willemstad, Curaçao, 2024, pp. 314–330, doi: 10.1007/978-3-031-78676-1_18.
- [2] S. Wang, D. Crapis, and C. C. Moallemi, “A framework for combined transaction posting and pricing for Layer 2 blockchains,” in *Financial Cryptography and Data Security (FC 2025)*, Miyakojima, Japan, 2025, pp. 56–72, doi: 10.1007/978-3-032-07024-1_4.
- [3] A. Mameishvili and E. W. Felten, “Efficient rollup batch posting strategy on base layer,” in *Financial Cryptography and Data Security: FC 2023 International Workshops*, Bol, Brač, Croatia, 2023, pp. 355–366, doi: 10.1007/978-3-031-48806-1_23.
- [4] S. Park, B. Mun, S. Lee, W. Jeong, J. Lee, H. Eom, and H. Jang, “Impact of EIP-4844 on Ethereum: Consensus security, Ethereum usage, rollup transaction dynamics, and blob gas fee markets,” *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 1294–1310, 2026, doi: 10.1109/TNSE.2025.3594281.
- [5] L. Heimbach and J. Milonis, “The early days of the Ethereum blob fee market and lessons learnt,” in *Financial Cryptography and Data Security (FC 2025)*, Miyakojima, Japan, 2025, pp. 53–71, doi: 10.1007/978-3-032-07035-7_4.
- [6] Y. Huang, S. Wang, Y. Huang, G. Tyson, H. Duan, and J. Tang, “A first look at Ethereum blob revolution: Market, strategies, and optimality,” arXiv:2411.03892, 2024.
- [7] S. Lee, “180 days after EIP-4844: Will blob sharing solve dilemma for small rollups?” in *IEEE 45th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Glasgow, United Kingdom, 2025, pp. 487–492, doi: 10.1109/ICDCSW63273.2025.00096.
- [8] D. Crapis, E. W. Felten, and A. Mameishvili, “EIP-4844 economics and rollup strategies,” in *Financial Cryptography and Data Security: FC 2024 International Workshops*, Willemstad, Curaçao, 2024, pp. 135–149, doi: 10.1007/978-3-031-69231-4_10.
- [9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” arXiv:1707.06347, 2017.
- [10] M. Tan, “Multi-agent reinforcement learning: Independent vs. cooperative agents,” in *Proceedings of the Tenth International Conference on Machine Learning (ICML)*, Amherst, MA, USA, 1993, pp. 330–337, doi: 10.1016/B978-1-55860-307-3.50049-6.
- [11] D. Byrd, M. Hybinette, and T. H. Balch, “ABIDES: Towards high-fidelity multi-agent market simulation,” in *Proceedings of the 2020 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS)*, Miami, FL, USA, 2020, pp. 11–22, doi: 10.1145/3384441.3395986.
- [12] T. Roughgarden, “Transaction fee mechanism design,” *ACM SIGecom Exchanges*, vol. 19, no. 1, pp. 52–55, 2021, doi: 10.1145/3476436.3476445.
- [13] S. Leonardos, B. Monnot, D. Reijnders, E. Skoulakis, and G. Piliouras, “Dynamical analysis of the EIP-1559 Ethereum fee market,” in *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies (AFT)*, Arlington, VA, USA, 2021, pp. 114–126, doi: 10.1145/3479722.3480993.
- [14] D. Ryan, D. Feist, F. D’Amato, H.-W. Wang, and A. Stokes, “EIP-7594: PeerDAS – Peer data availability sampling,” Ethereum Improvement Proposals, no. 7594, 2024, Accessed: Oct. 5, 2026. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-7594>
- [15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi: 10.1038/nature14236.
- [16] P. Jayanthi, T. Wahrstätter, S. Calder-Mason, A. Davis, and A. Dietrichs, “EIP-7691: Blob throughput increase,” Ethereum Improvement Proposals, no. 7691, 2024, Accessed: Oct. 5, 2026. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-7691>
- [17] M. Mackey and R. Kripalani, “EIP-7892: Blob parameter only hardforks,” Ethereum Improvement Proposals, no. 7892, 2025, Accessed: Oct. 5, 2026. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-7892>
- [18] A. Elowsson, B. Adams, and F. D’Amato, “EIP-7918: Blob base fee bounded by execution cost,” Ethereum Improvement Proposals, no. 7918, 2025, Accessed: Oct. 5, 2026. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-7918>
- [19] T. Wahrstätter and V. Buterin, “EIP-7623: Increase calldata cost,” Ethereum Improvement Proposals, no. 7623, 2024, Accessed: Oct. 5, 2026. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-7623>
- [20] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, “A survey and critique of multiagent deep reinforcement learning,” *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 6, pp. 750–797, 2019, doi: 10.1007/s10458-019-09421-1.