When Does Being Selfish Pay Off? Temporal Composability and Profitability in Selfish Mining

Colin Finkbeiner, Connor Shaw, and Ghada Almashaqbeh

University of Connecticut
{colin.finkbeiner,connor.shaw,ghada.almashaqbeh}@uconn.edu

Abstract. Selfish mining undermines incentive compatibility of proof-of-work blockchains, letting a miner earn disproportionate rewards at a hashrate lower than the majority threshold. A decade of work has asked *whether* a strategy is profitable, however, far less is understood about *when* it becomes profitable. Timing is critical since selfish mining operates at a loss before it turns a profit, typically requiring tens of weeks to break even in the classic case.

In this paper, we present a holistic study of the time-to-profitability (TTP) of existing selfish mining strategies structured around four contributions. First, in the single-attacker setting, we characterize TTP across the full strategy space and find that TTP-minimizing and profit-maximizing strategies frequently diverge, making attack horizon a critical metric. In particular, under realistic fee dynamics, the use of incentive transactions to recruit honest-but-rational miners enable incentivized strategies to reach profitability up to 15× faster than classic selfish mining at the same hash rate. Second, we explore TTP for the first time in the multi-attacker setting, showing that the difference in strategies between opposing attackers has a dramatic impact on joint-profitability lag. Third, we generalize intermittent selfish mining by exploring temporal composition over the full strategy space and show that its purported benefits are largely overstated. That is, alternating strategies rarely outperform the best static strategy in terms of either TTP or long-term profits.

Finally, and building off our earlier findings, we explore adaptive, state-conditioned strategy selection at the difficulty adjustment period (DAP) level. We compare a general-purpose LLM agent against a fixed decision-tree selector, both implementing the same selection criteria. We find that both selectors reliably identify profit-maximizing strategies from observed network conditions, at a low operating cost, lowering the expertise barrier to exploiting adaptive selfish mining.

1 Introduction

Selfish mining poses a fundamental threat to incentive compatibility of proof-of-work blockchains such as Bitcoin. By selectively withholding mined blocks, and exploiting Bitcoin's fork resolution process and difficulty adjustment mechanisms, a miner can earn disproportionate rewards at a hashrate well below the majority threshold. In the decade since Eyal and Sirer's seminal result on classic selfish

mining [10], a substantial body of work has emerged and expanded our understanding of this threat [2, 3, 5, 6, 12, 13, 15–17, 19–21, 23, 24, 26, 27]. These showed new strategy variants that outperform classic selfish mining, explored optimal policies via analytical models and deep reinforcement learning, and studied richer model settings that cover propagation delay, multiple competing attackers, etc.

Most of these works focused on profitability in the long term. However, a related basic question that has received little attention is *when* selfish mining becomes profitable. This is a critical question since a selfish miner first has to waste its own blocks, to lower the difficulty target, thus operating at a loss before starting profit reaping. In fact, it has been estimated that classic selfish mining takes on the order of weeks of sustained attack before turning a profit [12]. For a rational attacker, this time-to-profitability (TTP) could be a vital factor in practice. When having a finite attack window, such an attacker may care less about asymptotic gains and more about how long it must absorb losses to collect profits.

Despite the large amount of work on selfish mining, existing examinations around TTP are narrow. It has been studied almost exclusively for classic selfish mining and a handful of purpose-built strategies aiming to minimize this time. These come in the form of intermittent strategies alternating between epochs of honest and selfish mining [8, 18, 20], which like most of the selfish mining literature [11], have been studied in the single attacker and block-reward only model. To date, no work has studied TTP systematically across the strategy space, various reward models, or the setting of multiple competing attackers. Furthermore, beyond intermittent strategies, there are no studies on the more general case of *temporal composition of strategies*. That is, how alternating among distinct selfish mining strategies over time may affect profitability and TTP.

Moreover, and as highlighted by [5, 13], optimal selfish mining depends on a set of dynamically changing factors—reward state, network conditions, and behavior of competing miners—that greatly complicate closed-form analysis. While these works have shown that deep reinforcement learning (DRL) can identify near-optimal strategies, doing so requires per-environment training and DRL expertise, and operates at a block-by-block granularity. To our knowledge, no work has examined whether strategies can instead be adapted at a coarser macro-strategy level, i.e., per difficulty adjustment period (DAP), and whether a strategy selector (a general-purpose LLM agent or a simple rule-based procedure) can successfully respond to evolving blockchain state.

Open questions. Thus, we ask the following questions: *How does TTP vary across the strategy space, reward models, and the number of competing attackers? Also, does temporal strategy composition improve upon static strategy choice, and can a per-DAP selector—agentic or rule-based—adapt effectively to evolving network conditions?*

1.1 Contributions

In this paper, we answer these questions by presenting a holistic study of these issues. It covers TTP for existing selfish mining strategies in both the single- and

multi-attacker setting, as well as temporal composability of these strategies that generalizes intermittent selfish mining. We also examine how AI-agents may assist in temporal composition; adaptively selecting strategies to maximize profit under changing blockchain state. In particular, we make the following contributions.

TTP across the static single-attacker strategy space. We present the first systematic TTP analysis that, beyond classic selfish mining, covers lead-, trail-, fork-stubborn families [17], publish-N [15], and incentive-aware trailing [11]. We introduce a family of *value-aware* strategy variants generalizing improved selfish mining [6] across the strategies discussed above, i.e., strategy variants that conditionally trigger publishing blocks based on their total transaction fee value. The central finding is that TTP-minimizing and profit-maximizing strategies frequently diverge, making TTP a leading metric for an attacker with a finite attack window. Also, under realistic fee dynamics, incentivized trailing strategies (where an attacker recruits honest-but-rational miners to build upon its chain) outperform both value-aware and value-agnostic strategies in terms of profitability.

Expanding on this, we characterize how TTP varies across reward models of increasing complexity: block-reward only, block-reward with transaction fees, and block reward with transaction fees and occasional high-value "whale" transactions. This is done while having honest-but-rational miners who can be swayed by monetary incentives. Our findings are class conditional. While, naturally, non-incentivized strategies are largely insensitive to the fee model, incentivized strategies (relying on honest-but-rational miners) accelerate substantially with whale transaction frequency and magnitude.

TTP in the multi-attacker setting. We provide the first TTP analysis of two competing selfish miners, finding TTP to be less than 1 DAP in the case of value-aware incentive-trail strategies. Moreover, we introduce the *joint TTP* (JTTP) metric, which is the TTP analog of joint profitability. We show that the combination of strategies chosen by attackers has a considerable impact on JTTP. In particular, when attackers adopt strategies of similar types (i.e., incentive-driven, value-aware) JTTP is less than that of adopting different strategy types (for example, in some instances we find the opponent may never reach profitability). We also study TTP in the multi-attacker setting for different reward models and in the presence of honest-but-rational-miners, observing that the class-conditional behavior persists.

Temporal strategy composition. We generalize prior intermittent selfish mining to per-DAP composition over the full examined strategy space, rather than just alternating between honest and classic selfish mining. We evaluate this generalized temporal composability in the multi-attacker setting under the presence of whale transactions and rational-but-honest miners. The result is a negative one for the prevailing narrative around temporal strategies: across the space tested, and static network conditions, alternating rarely outperforms the best static strategy for TTP or asymptotic profit gains.

Adaptive strategy selection at the DAP level. We explore whether adaptive temporal composition of selfish mining strategies can be automated at a macro-strategy, per-DAP granularity. We compare two selector implementations: an LLM agent and a fixed rule-based decision tree, both following the same encoded strategy knowledge. We show that a per-DAP selector, with access to recent public blockchain state, can locate the dominant strategy from observable signals alone and reacts to network condition shifts. Different from what one would expect, this trend holds regardless of whether the selector is agentic or rule-based, and holds for the multi-attacker setting without knowledge of the opposing attacker's strategy. We believe that this is in part due to a single dominant strategy identified over much of the parameter space, as well as the encoded strategy knowledge mapping the state space to a clean set of strategy categories with distinct applicability conditions. Thus, once the selector locates the right state space, the strategy choice follows regardless of implementation. Overall, the security implication is that adaptive selfish mining is operationally feasible in practice, accessible to miners without DRL expertise, and made even easier by using AI agents (compared to manually configuring a rule-based selector).

1.2 Related Work

TTP in selfish mining. The question of when selfish mining becomes profitable has received little attention in the literature. Grunspan et al. [12] showed that classic selfish mining requires on the order of weeks to become profitable. Subsequent TTP-aware works [8, 18, 20] have been confined to the single-attacker and block-reward-only settings, while considering only binary alternation between selfish and honest mining. On the reward-model side, Carlsten et al. [6] introduced improved selfish mining, exploiting per-block fee variation. Bar et al. [5] introduced WeRLman that jointly models fees and whale transactions but only for the single-attacker setting and focusing on long-term profitability. In terms of model setting, multi-attacker selfish mining has been studied for asymptotic profitability (within a range of simulation-based and mathematical modeling approaches) [3, 4, 15, 16, 23, 26]. None of these works characterize TTP across the broader strategy space, richer reward models that account for transaction fees, or the multi-attacker setting where competitive interaction reshapes the realization of individual and joint attacker profitability in ways that single-setting asymptotic analyses do not capture. Our work aims to address this gap.

Temporal composability in selfish mining. A few works have studied strategies that vary over time. Table 1 summarizes these along three axes: the strategy set considered, decision granularity, and model setting. Negy et al. [18] introduced intermittent selfish mining, alternating between honest and selfish behavior for every DAP to exploit difficulty-target reductions. It achieves single-DAP profitability at the cost of a higher security threshold (0.37 vs. 0.33 of classic selfish mining based on the attacker connectivity setting). Sarenche et al. [20] introduced smart intermittent selfish mining, with the same strategy alternation but within DAPs, identifying a tradeoff between TTP and the security threshold.

Table 1: Comparison of temporally composable selfish mining works. † = Classic selfish mining, lead-/trail-/fork-stubborn families and their compositions, incentive-aware trailing strategies [11], value-aware strategies (generalizing Carlsten [6]), and Publish-N [15].

	Intermittent Selfish [18]	**Smart Int. Selfish** [20]	**L-Stubborn Int.** [8]	**Undetectable Selfish** [2]	**Our Work**
Composition structure					
Strategy set composed over	{honest, classic selfish}	{honest, classic selfish}	{honest, L-stubborn}	{honest, selfish}	Full family†
Composition granularity	per-DAP	per-DAP & sub-DAP	per-DAP	per-block	per-DAP
Objective					
Profit-maximizing	✗	✓	✓	✓	✓
Time-to-profitability minimizing	✓	✓	✓	✗	✓
Model setting					
Single attacker	✓	✓	✓	✓	✓
Multiple attackers	✗	✗	✗	✗	✓
Block-reward only	✓	✓	✓	✓	✓
Transaction-fee model	✗	✗	✗	✗	✓
Whale transactions	✗	✗	✗	✗	✓
Honest-but-rational miners modeled	✗	✗	✗	✗	✓
Automated / learned selection	✗	✗	✗	✗	✓

Doger et al. [8] extended this analysis with an L-stubborn intermittent variant that offers short-term gains, and interestingly showed that classic selfish mining still dominates in the long run. The undetectable selfish mining family [2] can also be viewed as temporal, though attacker's decision around block publication are driven by detection-resistance rather than profitability.

All these temporal strategies compose over a binary set—honest paired with a single selfish variant—leaving the richer strategy space unexplored. Also, similar to above, they are evaluated only under the single-attacker, block-reward-only model and without considering honest-but-rational miners. We address these limitations in this paper.

Optimality in selfish mining. Optimality in selfish mining strategies has been studied under various conditions [2, 7, 14, 15, 17, 19, 24]. Table 2 summarizes these prior efforts organized by the decision-making mechanism they employ. Sapirshtein et al. [19] formulated selfish mining as a Markov Decision Process (MDP) and obtained an ϵ-optimal policy in the single-attacker, block-reward-only setting. Bai et al. [3] extended that to a partially-observable MDP for the multi-attacker case. To circumvent the intractability of large state spaces, i.e., being hard to be captured by a closed form analytical model, DRL has been employed in making optimal decisions. In particular, SquiRL [13] trained a DRL model that replicated [19] within 1% and conjectured that no profitable Nash equilibrium exists in the multi-attacker block-reward setting. While WeRLman [5] extends the model to incorporate occasional high-value whale transactions. The obtained results show that fee variability directly lowers the security threshold, falling to $\sim$20% under frequent whale transactions. The work [9] took a different route with L^*-stubborn mining, operating at strategy granularity rather than per-block—a distinction shared by our work (although their goal is maximizing profitability of double spending).

Table 2: Comparison of optimal and near-optimal selfish mining works (✔ indicates the property is present; ✖ indicates it is not. POMDP = partially-observable MDP).

	Optimal [19]	**Optimal MA** [3]	L^***-Stubborn** [9]	**SquiRL** [13]	**WeRLman** [5]	**Our Study**
Decision model						
Method	MDP	POMDP	analytic	DRL	DRL	sim. study
Decision granularity	per-block	per-block	strategy-level	per-block	per-block	per-DAP
Generalizes across reward models without retraining	✖	✖	✔	✖	✖	✔
Objective						
Profit-maximizing	✔	✔	✔	✔	✔	✔
Time-to-profitability minimizing	✖	✖	✖	✖	✖	✔
Attacker setting						
Single attacker	✔	✔	✔	✔	✔	✔
Multiple attackers	✖	✔	✖	✔	✖	✔
Reward model						
Block-reward only	✔	✔	✔	✔	✔	✔
Transaction-fee model	✖	✖	✖	✖	✔	✔
Whale transactions	✖	✖	✖	✖	✔	✔
Honest-but-rational miners	✖	✖	✖	✖	✖	✔

All of these approaches share two properties: they target asymptotic profitability rather than TTP, and they require an offline optimization procedure—MDP solution, DRL training, or analytical derivation—tailored to a specific model setting. Re-specifying the model (e.g., inclusion of whale transactions, honest-but-rational miners, or rival attackers) requires re-deriving or re-training the policy. Our LLM-based approach explores an alternative paradigm: rather than learning or solving for a policy offline, we use a general-purpose AI agent as a state-conditioned strategy selector at DAP boundaries informed with mining context. We preface this is not a drop-in replacement to single-setting optimizations, but as a complementary direction that would be beneficial for navigating model variations.

2 Preliminaries

We offer an overview of Nakamoto consensus and selfish mining, then we present our threat and system models, followed by the strategies and the temporal composition we examine.

2.1 Background

Nakamoto consensus. We consider Nakamoto-style proof-of-work blockchains. Miners package transactions into blocks, solve a computational puzzle whose difficulty is calibrated by a difficulty adjustment mechanism, and extend the longest chain. A miner whose block is adopted by the longest chain earns a block reward and collects the fees of the transactions included in that block. In Bitcoin, the difficulty target is recomputed every $N = 2016$ blocks—a DAP, roughly two weeks—to hold the expected block time near 10 minutes.

Selfish mining. Eyal and Sirer [10] showed that honest mining is not always rational. A selfish miner who withholds newly mined blocks and selectively publishes them will be able to *orphan* honest miners' blocks, thus capturing a disproportionate revenue share. Difficulty adjustment is vital to such behavior: orphaned blocks reduce the rate of appending blocks to the longest-chain, which in turn reduces the next epoch's difficulty target and inflates future rewards. Thus, selfish mining is profitable when the attacker's share of total longest-chain rewards is greater than the expected one based on its hashrate.

2.2 System and Threat Model

Network and miners. We model a blockchain network as a set of miners each controlling a fraction of the total network hashrate. Following the conventional model in the literature, block discovery can be viewed as a Poisson process, and the probability that a miner produces blocks depends on its hashrate. Miners in the system can be classified into three classes:

- *Honest miners* follow the protocol as described: they mine on the longest chain known to them and publish each block immediately upon discovery. In the case of a fork, they choose the longest branch, and when receiving two or more blocks at the same height, they choose the block that was broadcast first. The collective hashrate of honest miners in the network is denoted by $\beta \in [0, 1]$.
- *Selfish miners*, or simply attackers, deviate from the protocol to maximize their utility. An attacker i is characterized by a hashrate $\alpha_i \in [0, 1]$, (sometimes represented as a percentage) and a strategy s_i from the selfish mining strategy space. Selfish miners also share a connectivity parameter γ, defined below. In the single attacker setting, we denote an attacker's configuration as α, γ, s; in the multi-attacker setting, as γ and a tuple α_i, δ_i, s_i, where δ_i (also defined below) is attacker i's share of γ.
- *Honest-but-rational miners* are those that do not selfishly mine on their own, but their mining choices can be swayed by what maximizes their utility. That is, they can be incentivized to adopt a selfish miner's branch/block if this results in additional profits. Honest-but-rational miners are denoted by a hashrate fraction $\kappa \in [0, 1]$.

The hashrate fractions satisfy $\beta + \alpha + \kappa = 1$ in the single attacker setting, and $\beta + \sum_i \alpha_i + \kappa = 1$ in the multi-attacker case.

An attacker has complete knowledge of the public chain and can withhold blocks, maintain a private chain, and release blocks strategically. In multi-attacker settings, attackers have partial information in the sense that they can observe each other's published blocks but not private chains or strategies. An attacker's objective could be: (1) maximizing profits, in which accumulated profits in the long term are being maximized; or (2) minimizing TTP, in which the time needed for a strategy to become profitable is being minimized. These objectives are not equivalent; a strategy that maximizes long-run revenue may absorb substantial

losses at the beginning, while a strategy that minimizes TTP may forfeit some asymptotic gains.

Fork resolution. When an attacker releases a block that competes with an honest block(s) at the same height, the network is temporarily split. The connectivity parameter $\gamma \in [0, 1]$ determines the fraction of honest miners that mine on top of the attacker's block while the remainder mine on top of the honest block. In effect, $\gamma = 0$ models a poorly connected attacker whose blocks are always broadcast after honest miner's blocks, while $\gamma = 1$ models a well-connected attacker whose blocks always reach honest miners first. Similarly, in the multi-attacker case, when a fork contains blocks from multiple attackers, δ_i denotes the fraction of γ that mines on attacker i's block, where $\sum_i \delta_i = 1$. For simplicity, we assume δ_i is equally split across all attackers (e.g., for $i = 2$, we have $\delta_1 = \delta_2 = 0.5$).

Difficulty adjustment. At the end of each DAP (orphaned blocks do not count toward the N-block total), the target threshold is rescaled by the ratio of the expected time needed to produce N blocks to the actual time observed. The goal is to maintain the expected block production time constant. This rescaling is the mechanism selfish mining exploits, so we use the DAP as our unit of time throughout the paper.

Reward model. We consider three reward models. First, block-reward only, where each block carries a fixed reward R. Second, block reward + transaction fees, where similar to [6], fees accrue continuously between blocks at a fixed rate. Specifically, fees accumulate proportionally to the time elapsed since the last block, with fee_T denoting expected fees in a block found after interval T. And third, existence of whale transactions, which is a sub-case of the second model in which blocks may contain rare, high-fee transactions. Such events occur with per-block probability p and yield a reward of $R + m \cdot \mathsf{fee_T}$, where $m > 1$ is the magnitude multiplier over the baseline fee $\mathsf{fee_T}$. Non-whale blocks have a reward of $R + \mathsf{fee_T}$.

Measuring profitability and TTP. For profitability, we measure an attacker's *cumulative relative advantage*, cumRA, which is the percentage difference between the rewards captured by running a selfish strategy and those captured by mining honestly. An attacker is profitable at a given point if its cumulative relative revenue exceeds its hashrate, i.e., $\mathsf{cumRA} > 0$ (or $\mathsf{cumRA_i} > 0$ for attacker i in the multi-attacker setting). Additionally, in the multi-attacker setting we define joint profitability to be the case when all attackers are mutually profitable, i.e., all $\mathsf{cumRA_i} > 0$.

We define TTP as the earliest DAP after which $\mathsf{cumRA} > 0$ and remains so for the rest of the attack duration (similarly $\mathsf{TTP_i}$ in the multi-attacker case). We use $\mathsf{TTP} = \mathsf{undefined}$ when an attacker's cumRA does not exceed 0, i.e., does not become profitable. Analogous to joint profitability, in the multi-attacker setting we additionally define joint TTP (JTTP) as the earliest DAP after which all attackers are simultaneously profitable and remain so.

2.3 Examined Strategies

Base strategies. We have honest mining denoted as H, and classic selfish mining [10] denoted as S. The stubborn family [17] generalizes S by being more aggressive in three distinct directions: L (stubborn-lead) continues withholding even when ahead of the public chain, thus gambling on extending the private lead further; F (stubborn-fork) keeps mining on its own branch when tied with the public chain rather than conceding; and T_n (stubborn-trailing) continues mining on the private branch even when lagging behind the public chain by up to n blocks. We also have publish-N [25], denoted as P_N, which withholds blocks until the private branch has a lead of N blocks before releasing. We refer readers to [11] for a detailed overview of these strategy families.

Stubborn mining composition. We use $\circ$ to denote (static) composition of base strategies. For example, $L{\circ}T_1{\circ}F$ applies stubborn-lead, stubborn-trail-1, and stubborn-fork behaviors jointly. Composition is governed by the rules in [17]—base stubborn strategies govern mutually exclusive mining scenarios, lead (attacker ahead), trail (attacker behind), and fork (tie), thus at most one applies at any decision point making composition well-defined.

Value-aware selfish mining. Carlsten et al. [6] introduced *improved selfish mining* in which an attacker is willing to publish a high-fee block early (instead of withholding and extending their private chain) that would otherwise be at risk of being claimed by the honest network's chain. This captures the Bitcoin state when mining rewards decline and transaction fees become the dominant source of miners' income.

A *value-aware* strategy is a generalization of this idea applied to any selfish mining strategy (including composed and incentivized ones). Immediate publication of blocks can be triggered if the most recent block carries sufficiently large fees. Formally, let f_{in} be the actual total transaction fees of the most recently mined block on the private chain, and let $\mathbb{E}[f]$ be the expected total fees of a block. Value-aware override check fires when $f_{\text{in}} \geq \tau \cdot \mathbb{E}[f]$. In this case, the attacker immediately publishes its entire private chain, regardless of the strategy. The factor $\tau \geq 1$ quantifies the relative minimum fee magnitude to deviate from the base strategy's publication policy. Delaying publication risks losing these fees to the honest chain, so τ trades off the expected value of continued withholding against the guaranteed reward of immediate publication. Because inter-block times are exponentially distributed around a 10-minute mean, a fixed fee threshold would be miscalibrated; representing τ as a ratio instead lets the trigger scale with fee accrual.

Incentivized trailing strategies. Recently, Finkbeiner et al. [11] demonstrated how the presence of honest-but-rational miners enables selfish miners to create a mutually beneficial scenario. That is, selfish miners inject additional fees in the network to attract rational miners, such that collecting these fees is conditional on adopting the selfish miner's chain. This was found specifically beneficial for stubborn-trail selfish mining; a selfish miner coming from a trailing position improves the chances of having honest miners mine on its block.

As in [11], we use $I(\eta, n)$ to denote incentivized trailing strategies, with fee-incentive magnitude η (expressed as a multiplier of the expected transaction fees fee_T) and stubborn-trail depth n. From a trailing position, the attacker posts $\eta \cdot \mathbb{E}[f]$ in fees conditional on its branch being extended. $I(\eta, n)$ is only meaningful when honest-but-rational miners are present ($\kappa > 0$) and the reward model includes transaction fees.

Notational examples. VA(1.5)$\circ L$ is value-aware stubborn-lead strategy with $\tau = 1.5$. While VA(1.5)$\circ I(2,1)\circ L$ is a value-aware ($\tau = 1.5$) incentivized ($\eta = 2$, trail depth 1) stubborn-lead strategy.

2.4 Temporal Strategy Composition

Here, an attacker need not commit to a single strategy for the duration of an attack. Prior work on intermittent selfish mining [8, 18, 20] alternates honest mining with a single selfish strategy—classic [18, 20] or L-stubborn [8]—at DAP boundaries, exploiting the difficulty-target drop from elevated orphan rate. We refer to this more generally as *temporal composition*: DAP-level selection from the strategies introduced in 2.3, of which honest-selfish alternation is one special case. We write $X \leftrightarrow Y$ for strict alternation between strategies X and Y on DAP boundaries, and $X \rightarrow Y$ for a one-shot transition from X to Y.

Destabilization and capture. Prior intermittent strategies can be decomposed into two phases: a *destabilization* phase that maximizes orphan production to lower the next epoch's difficulty target, and a *capture* phase that maximizes relative revenue at the reduced difficulty. Past work pairs a selfish variant (destabilization) with honest mining (capture), but selfish strategies themselves may capture more effectively than honest mining at a reduced target. We use this decomposition in Section 6 as a heuristic for generating candidate temporal strategy compositions to evaluate empirically. It is an open question whether this decomposition identifies the best pairings, rather we utilize it to provide a principled way to narrow down the search space of compositions to be examined.

3 Evaluation Setup

The simulator. Our evaluation builds on the discrete-event mining simulator of [11], itself an extension of [6]. We extend this simulator in two ways. First, we add per-epoch logging; at every DAP boundary, we record each attacker's cumulative and per-DAP relative revenue, orphan rate, and realized difficulty. This allows recovering the full TTP curve rather than a single endpoint, and supplies the per-epoch state summaries we need for the AI agent-based approach (Section 7). Second, we add per-epoch strategy selection; an attacker may switch strategies at any DAP boundary based on observed fees and network conditions. This enables evaluating our temporal strategy composition framework, and is needed for the adaptive strategy modeling.

Block production and DAP. Following the Bitcoin protocol, we assume an expected block production time of 10 minutes and a DAP of $N = 2016$ blocks.

Reward models. We examine the three reward models discussed earlier. Under the block-reward-only model, each block carries a fixed reward normalized to 1. While for the other two models (that add transaction fees), transaction fees accrue continuously over time between blocks (captured by fee_T defined earlier). The baseline transaction fees contribute approximately 17.5% of total block rewards for a block produced at the expected 10-minute production rate. We adopt this elevated baseline, consistent with prior work [5,6], to reflect fees' growing share of future miner revenue [22]. To model noise in realized fees, a block's fee is drawn uniformly from $[0.75 \cdot \mathsf{fee}_\mathsf{T}, \mathsf{fee}_\mathsf{T}]$.

Whale transactions. When enabled, each block independently contains a whale transaction with probability $p \in \{0.05, 0.1\}$ (with $p = 0.0$ being the no-whale baseline), and a high-value fee represented as a magnitude multiplier $m \in \{4\times, 8\times\}$ applied to fee_T for the corresponding inter-block period. We vary p and m to examine their impact on TTP.

Game runs and statistics. Each (simulated) game run continues until the public chain reaches block height $24,192$—representing 12 DAPs. Reported quantities are averaged, with 95% confidence intervals, over 100 independent runs per parameter configuration.

4 TTP Across the Single-attacker Strategy Space

Baseline strategy dominance. We examine each strategy across the full (α, γ) parameter grid with 0.05 increments. We measure profitability, which is taken as the minimum of the computed 95% CI collected across 100 game runs each of $12,096$ blocks (6 DAPs, $\approx$ 12 weeks based on the 10-minute block-production rate). Figure 1 shows the dominant strategies, i.e., those with the highest profitability, among the 10 strategies in the block-reward only setting. As noted, the dominant strategy varies systematically with α and γ, forming distinct regions rather than a single strategy dominating; while this result is established in the block-reward setting [17], we include it as a baseline comparison for our TTP-analysis that follows.

Min-TTP strategy rationality. In comparison, the left panel of Figure 2 shows the minimum TTP-strategy of the 10 strategies. The landscape is fragmented; no single strategy minimizes TTP across the entire profitable region, mirroring asymptotic profitability findings, though TTP-minimizing regions map to different dominant strategies. For $\alpha \in [0.25, 0.35]$ and $\gamma < 0.3$, the composed strategy $L \circ T_1$ (i.e., stubborn lead-trailing-1-fork) achieves the minimum TTP, while classic selfish mining S dominates for $\gamma > 0.5$. The publish-N families P_3 and P_4 dominate only near the right edge, when $\alpha \geq 0.40$, where their publication threshold takes on a key advantage. The grey unprofitable region (within 6 DAPs) is for $\alpha \leq 0.33$, which aligns with asymptotic profitability results.

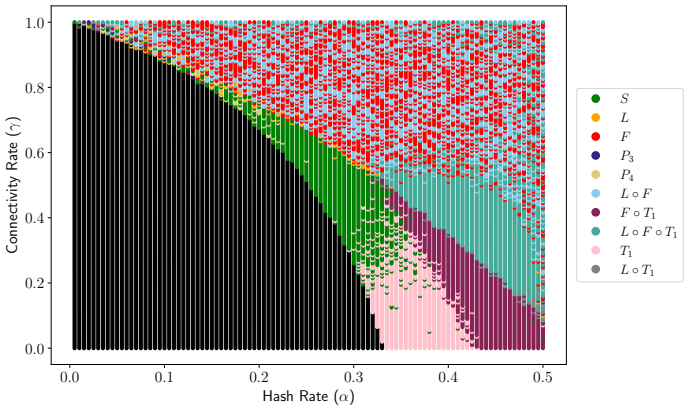


Fig. 1: Profit dominant strategy across parameter space (single attacker, block-reward only).

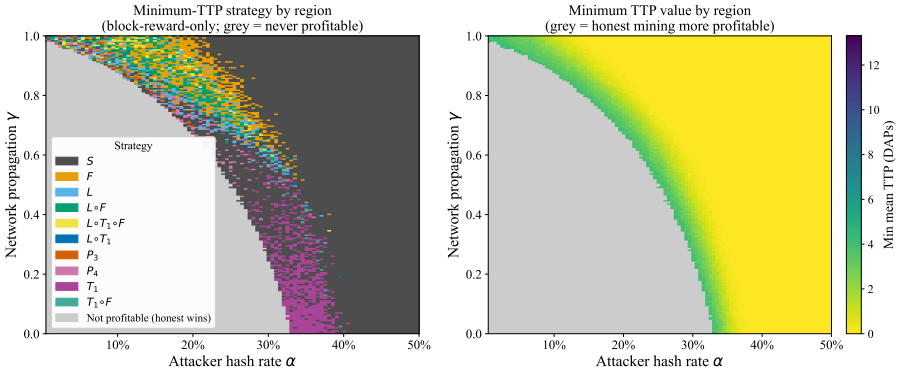


Fig. 2: TTP-minimizing strategy across parameter region, and minimal TTP per parameter choice (single attacker, block-reward only).

TTP magnitude. The right panel of Figure 2 shows that even within the profitable zone, TTP ranges from 0 DAPs (immediate profitability at high α) to $5-6$ DAPs near the boundary. TTP drops sharply when α moves from 0.38 to 0.40 at $\gamma = 0.0$, falling from ≈ 4 to ≈ 1. At $\alpha = 0.35$ and $\gamma = 0.25$, the min-TTP strategy stubborn-lead-fork mining $L \circ F$ reaches TTP ≈ 2 DAPs, while classic selfish mining S (despite higher asymptotic profit fraction at this α value) has TTP ≈ 4.5 DAPs. Profit-maximizing and TTP-minimizing strategies therefore diverge; an attacker with a 3-DAP attack window should choose $L \circ F$ over S despite S's greater long-run advantage.

Value-aware (VA) strategies: Profit and TTP under fees and whale transactions. Figure 3 plots each strategy family at $\alpha = 0.25$ and $\tau = 1.5$ according to its final cumulative revenue advantage cumRA (expressed as a

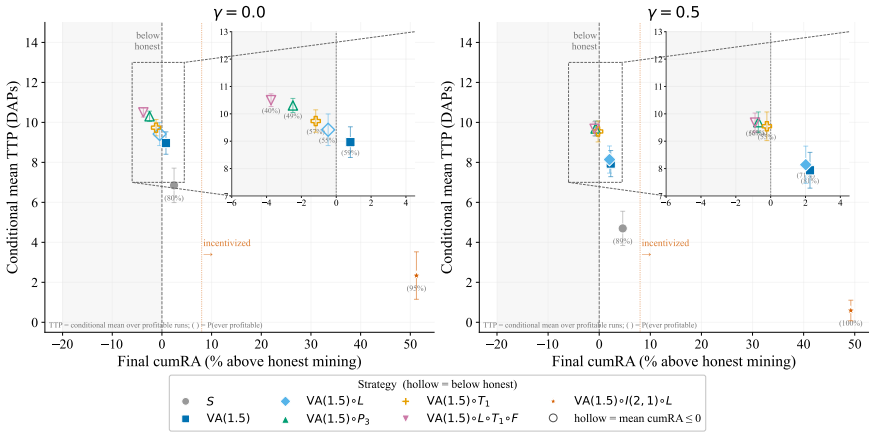


Fig. 3: Value-aware strategies; profitability versus TTP at $\alpha = 25\%$, $\tau = 1.5$, $\kappa = 0.25$ (single attacker setting).

percentage above what the miner would earn when mining honestly) and its conditional mean TTP. Filled markers indicate strategies whose mean revenue is positive across all 100 runs, while hollow markers indicate strategies that are unprofitable in expectation. The conditional mean TTP refers to the DAP of the first sustained revenue profitability—the first DAP from which cumulative cumRA remains positive—over the subset of runs where the selfish miner achieves positive revenue advantage by game end. Cases where a selfish miner never crosses into continual profit are excluded from the mean but are reflected in the finite extent of each curve.

Among the non-incentivized VA strategies, every family except the value-aware classic selfish mining VA$\circ S$ earns less than honest mining in expectation at $\gamma = 0$. VA-selfish is marginally above honest at +0.8%, while VA-stubborn-lead VA$\circ L$ sits near zero (-0.5% to $+0.1\%$), and VA-Publish-3 VA$\circ P_3$, VA-trail-1 VA$\circ T_1$), and VA-Stubborn-lead-trailing-1-fork VA$\circ L \circ T_1 \circ F$ fall between -1.2% and -3.7%. For these hollow-marker strategies, the reported TTP ($\approx$ 9–10 DAPs) reflects only the 40–57% of runs that happen to turn profitable; the remaining runs never recover. Plain selfish mining (S), by contrast, earns +2.4% with 76–80% probability of profitability.

The incentivized strategy value-aware incented-stubborn-lead-trailing-1 VA$(1.5) \circ I(2,1) \circ L$ is notably different. It earns approximately +47% above honest mining, and reaches profitability in a conditional mean of 0.25 DAPs, and is profitable in 96% of runs. In general, incentivized strategies deliver improved gains compared to other strategy variants on both dimensions, where they increase profitability and decrease TTP. Despite this, it is critical to note that this is under the presence of honest-but-rational miners (here, we have $\kappa = 0.25$).

At $\gamma = 0.5$, the same strategies appear to be profitable and unprofitable, as well as following the same order of profitability, with only a slight rightward shift



Fig. 4: Cumulative revenue advantage (% relative to honest mining) for seven selfish-mining strategies in the single attacker setting across three reward-models and two γ values in profitable cells indicate mean TTP in DAPs. $\alpha = 0.25, \kappa = 0.25$, $\tau = 1.5$.

to the final cumRA. Similarly, $\mathsf{VA}(1.5) \circ I(2,1) \circ L$ remains dominant at +46% profitability with 98% profitability having the same TTP.

Reward model sensitivity: Transaction fees and whale transactions. Figure 4 shows the cumulative advantage cumRA and TTP for value-aware and incentive-driven strategies under three reward models: block rewards and transactions fees only, light whale amplification (i.e., $p = 0.05$, $m = 4\times$), and heavy whale amplification (i.e., $p = 0.10$, $m = 8\times$). Here, $\alpha = 0.25$, $\gamma \in \{0.0, 0.5\}$ and $\kappa = 0.25$.[1]

Our findings highlight two distinct classes. Firstly, non-incentivized strategies S, VA$\circ S$, VA$\circ P_3$, VA$\circ T_1$, and VA$\circ L \circ T_1 \circ F$ remain in the range of -13% to $+5\%$ across all reward models and are largely insensitive to fee variations where whale events add at most a few percentage points. Secondly, incentivized strategies VA$\circ I(2,1)$ and VA$\circ I(2,1) \circ L$ have consistent profit across all reward models, ranging from +19% (fees only, $\gamma = 0.5$) to +81% (heavy whale, $\gamma = 0.5$), with profitability increasing and TTP decreasing consistent with increases of whale transaction frequency and fee magnitude. When $\gamma = 0.5$, we observe that incentivized strategies reach profitability remarkably quickly with VA $\circ\, I(2,1)$ achieving TTP < 0.6 DAP (≈ 8 days) across all fee regimes, and VA $\circ\, I(2,1) \circ L$ as low as 0.2 DAP in the fees-only baseline. When $\gamma = 0$, we see that the same strategies remain profitable but TTP rises to 8–10 DAP ($\approx$ 16–20 weeks).

Finding: Incentivization, not value-awareness, drives fast profitability. Across the static strategy results above, a sharp separation emerges between strategies that include the incentive mechanism $I(\eta, n)$ and those that do not.

[1] In Appendix A, we assess the robustness of the parameter choices underlying our results.

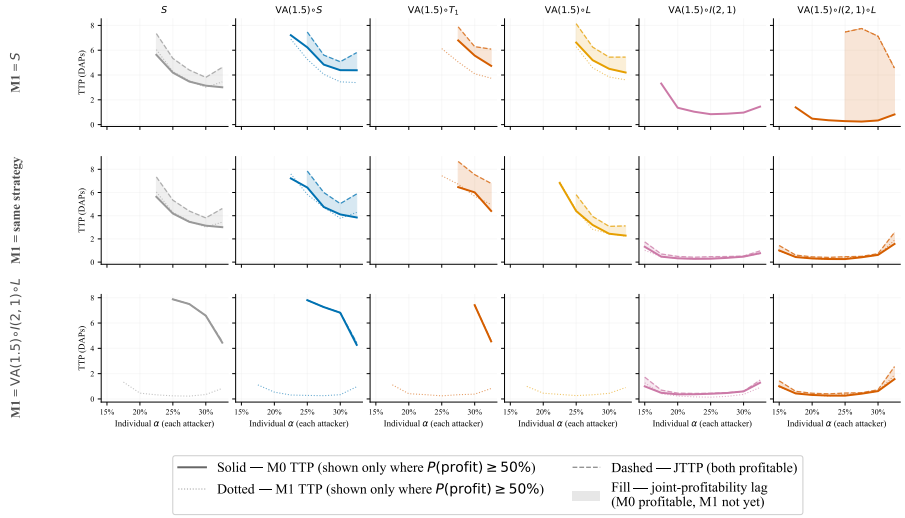


Fig. 5: TTP and JTTP in the multi-attacker setting ($\gamma = 0.5$). Lines appear only where $P(\mathsf{profit}) \geq 50\%$.

The separation holds along every axis we test; profitability, TTP, probability of profitability, and sensitivity to whale dynamics.

As seen in Figure 4, for $\alpha = 0.25$, $\gamma = 0.5$, $\kappa = 0.25$, the best non-incentivized value-aware strategy VA$\circ S$ earns +5% above honest mining with TTP ≈ 5.4 DAPs. Under the same parameters, VA$(1.5)\circ I(2,1)\circ L$ earns +26% with a TTP of 0.2 DAPs—around 5× improvement in revenue advantage and around 27× reduction in break-even time. Removing the incentive component from the strategy shows a clear collapse, both in terms of revenue that falls to near zero and TTP that rises to unreliable levels.

A natural question arises on why value-awareness alone is insufficient for incentivized strategies to achieve gains. In a nutshell, value-aware publication only changes *when* an attacker releases blocks it has already mined; it does not impact *who* mines on the attacker's branch during a fork. Carlsten's improved selfish mining demonstrated that fee-conditional publication modestly improves the return per high-fee block, but the underlying fork-resolution probability is still bounded by γ. The incentive mechanism, by contrast, affects fork-resolution itself. By posting conditional fees on its branch being extended, the attacker recruits honest-but-rational hashrate that would otherwise default to the honest chain. Thus, in general these two mechanisms improve selfish mining in two different directions, but the fork-resolution gains from incentivized strategy is where the larger gains lie.

It is important to note that the result is contingent on honest-but-rational miners being present, i.e., $\kappa > 0$, and on a reward model that admits per-block fee

variation. In the block-reward-only model, the incentive mechanism is unavailable and so the strategy landscape collapses back to the map of Figure 2.

5 TTP in the Multi-attacker Setting

We next examine TTP in the two-attacker setting, where each attacker holds individual hash rate α. All simulations assume whale transactions with $p = 0.05$ and $m = 4\times$, $\gamma \in \{0.0, 0.5\}$, and $\kappa \in \{0, 0.25\}$. Each parameterization run 100 times over 24,192 blocks, which is equivalent to 12 DAPs. Analogous to the conditional mean TTP, the conditional JTTP is defined over runs where both attacking miners, attacker miner 0 (M_0) and attacker miner 1 (M_1) reach positive cumulative revenue advantage.

Figure 5 reports conditional mean TTP and JTTP at $\gamma = 0.5$, $\kappa = 0.25$. Rows fix M_1's strategy: $M_1 = S$ (first row), M_1 uses the same strategy as M_0 (second row), and M_1 deploys VA(1.5)$\circ I(2,1)\circ L$ (third row). Columns vary M_0's strategy across six strategies of increasing sophistication (and profitability). Lines are drawn only where the strategy achieves positive revenue advantage in at least 50% of runs (i.e., $P(\mathsf{profit}) \geq 50\%$); no line indicates the strategy fails to clear that threshold, i.e., the reliability threshold. The *joint-profitability lag* (shaded band) is the gap between M_0's individual TTP and JTTP.

Against a selfish attacker (first row). Non-incentivized strategies S, VA(1.5)$\circ S$,VA(1.5)$\circ T_1$, and VA(1.5)$\circ L$ show no TTP line below $\alpha \approx 22.5$–27.5%. Where lines do appear, TTP clusters between 5.6 and 7.2 DAPs at entry, with VA(1.5)$\circ S$ (TTP = 7.2 DAPs at $\alpha = 22.5\%$) trailing classic selfish mining S (TTP = 5.6 DAPs). As before, block-value-awareness alone does not help, and modestly delays the time to break-even without the incentive component.

Incentivized strategies break sharply from this pattern. VA(1.5)$\circ I(2,1)$ is profitable starting at $\alpha = 17.5\%$ with TTP = 3.3 DAPs, reaching 1 DAP at $\alpha = 22.5\%$ and 0.84 DAPs at $\alpha = 25\%$. VA(1.5)$\circ I(2,1)\circ L$ becomes profitable starting at $\alpha = 17.5\%$ with TTP = 1.4 DAPs, converging to 0.35 DAPs at $\alpha = 22.5\%$ and 0.28 DAPs at $\alpha = 25\%$, which is roughly 15$\times$ faster than S at the same hash rate.

The joint-profitability lag for an incentivized M_0 against selfish M_1 is large. For VA(1.5)$\circ I(2,1)$, M_1 never achieves majority profitability across the tested range. This is due to the fact that the incentive mechanism captures high-fee transactions first, suppressing M_1's revenue below the reliability threshold, so JTTP is undefined. For VA(1.5)$\circ I(2,1)\circ L$, M_1 achieves profitability only starting at $\alpha = 25\%$ with TTP = 7.5 DAPs, where at this point M_0 has already been profitable for 7.2 DAPs, with JTTP = 7.47.

Self-competition (second row). When M_0 and M_1 deploy identical strategies, the picture changes. Non-incentivized strategies are largely unaffected: S and VA(1.5)$\circ L$ retain profitability entry thresholds near $\alpha = 22.5\%$ and TTP in the 4–7 DAP range, similar to the first row. Incentivized strategies benefit from symmetry: VA(1.5)$\circ I(2,1)\circ L$ and VA(1.5)$\circ I(2,1)$ are both profitable starting

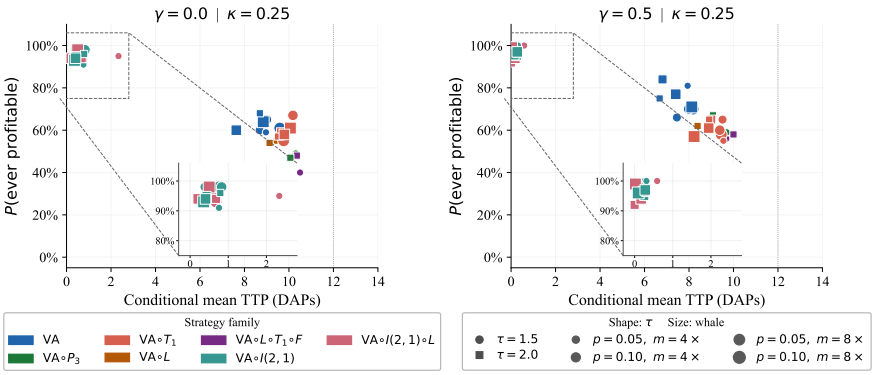


Fig. 6: Value-aware strategies in the multi-attacker setting; profitability versus TTP of M_0 when competing against a classic selfish miner M_1 ($\alpha = 0.25$).

at $\alpha = 15\%$ with TTP ≈ 1 DAP for each. The joint-profitability lag shrinks to 0.4–0.5 DAPs at $\alpha = 15\%$ and narrows further at higher α, against the 7.2 DAP lag observed in the first row at $\alpha = 25\%$.

Against the strongest adversary (third row). When M_1 adopts VA$(1.5)\circ I(2,1)\circ L$, non-incentivized strategies fare worst. In particular, S shows no profitable line below $\alpha = 25\%$ (TTP = 7.9 DAPs), and VA$(1.5)\circ L$ fails to establish majority profitability within 12 DAPs, appearing only marginally at $\alpha = 32.5\%$ with TTP = 5.0 DAPs. On the other hand, incentivized strategies fare better. As shown in the figure, VA$(1.5)\circ I(2,1)$ and VA$(1.5)\circ I(2,1)\circ L$ are both profitable starting at $\alpha = 15\%$ with TTP ≈ 1 DAP, closely matching their second row curves. Thus, we observe that a strong incentivized opponent does not raise the TTP or profitability threshold of a competing selfish miner.

Whale dynamics under competition. Figure 6 plots conditional mean TTP against $P(\mathsf{profit})$ for M_0 across all 38 value-aware strategy configurations against a classic selfish miner M_1 at $\alpha = 25\%$, $\kappa = 0.25$, spanning seven strategy families, two threshold values $\tau \in \{1.5, 2.0\}$, and four whale regimes $p \in \{0.05, 0.10\}$, $m \in \{4\times, 8\times\}$.

The figure reveals a sharp separation between incentivized and non-incentivized families. Non-incentivized strategies VA, VA$\circ T_1$, VA$\circ L$, VA$\circ P_3$, and VA$\circ L\circ T_1\circ F$ cluster in the upper-right region at TTP = 6–11 DAPs with $P(\mathsf{profit}) = 40$–84%. Also, several VA$\circ P_3$ and VA$\circ L\circ T_1\circ F$ configurations fall below $P(\mathsf{profit}) = 50\%$ at $\gamma = 0.0$, meaning that they fail to achieve majority block-share profitability within 12 DAPs against the classic selfish opponent. Incentivized families VA$\circ I(2,1)$ and VA$\circ I(2,1)\circ L$ occupy the top-left corner (visible in the zoom inset) with TTP < 1 DAP and $P(\mathsf{profit}) > 90\%$ across nearly all configurations for both γ values.

The parameter γ has a varying impact across the two strategy groups. For $\gamma = 0.5$, the non-incentivized cluster shifts leftward and upward relative to

$\gamma = 0.0$, where VA strategies improve from TTP $\approx$ 8–10 DAPs to 6.7–8.2 DAPs, reflecting the benefit of better connectivity. The incentivized cluster is largely unaffected; both families already achieve TTP < 0.6 DAPs and $P(\mathsf{profit}) > 92\%$ at $\gamma = 0.0$ and improve marginally to TTP ≈ 0.3 DAPs and $P(\mathsf{profit}) \geq 95\%$ at $\gamma = 0.5$. Within this cluster, larger whale parameters ($m = 8\times$ and $p = 0.10$) push points toward the top-left corner, confirming that both higher whale transaction frequency and magnitude independently accelerate capturing fees.

Finding: Asymmetric incentive adoption produces the dominant multi-attacker lag. The single attacker setting finding from Section 4, i.e., incentivization is the dominant driver of reducing TTP, extends into the multi-attacker setting in a specific way. That is, the joint-profitability lag is large when one attacker deploys the incentive mechanism and the other does not, and is roughly absent otherwise. In the first row in Figure 5, the only pairings producing substantial lag are those where M_0 runs an incentivized strategy against non-incentivized $M_1 = S$: VA$(1.5)\circ I(2,1)\circ L$ at $\alpha = 25\%$ has a TTP of just$= 0.28$ with M_1 not becoming also profitable until $= 7.47$ DAPs while VA$(1.5)\circ I(2,1)$ suppresses M_1 entirely. The four non-incentivized M_0 strategies in the first row in Figure 5 show small lag; both attackers converge together because neither has an advantage in capturing the honest-but-rational miner's hashrate. The second row in the figure confirms this reading, where under symmetric strategy choice the lag becomes very small regardless of whether the shared strategy is incentivized.

Underpinning this observation is the same fork resolution mechanism impact that found beneficial in the single-attacker setting. When one attacker posts branch-conditional fees and the other does not, the rational hashrate κ flows preferentially to the incentivizing party, depriving the other of any boost that would come from honest-but-rational miners. Conversely, in the symmetric setting, competition over honest-but-rational miners is restored as now both attackers can post incentives. In general, we see two implications of this finding. First, a sustained gap between the observed miner revenue trajectories may be a signal of one-sided use of incentives by an attacker running a non-incentivized/value-aware strategy. Second, incentivized strategies reach the symmetric joint-profitability regime with a individual hashrate as low as 15%.

6 Temporal Strategy Composition

Building on the destabilization/capture intuition introduced in Section 2.4, we evaluate temporal compositions that alternate between strategies maximizing orphan production and strategies maximizing revenue capture under reduced difficulty. We highlight that the static-strategy maps in Figures 1 and 2 (generalizable to their value-aware and multi-attacker analogues) already characterize TTP-minimizing and profit-maximizing choice for every cell of the (α, γ) grid. As a result, we do not separately evaluate compositions reacting to shifts in α or γ, since an attacker observing such a shift can simply remap to the corresponding static dominant strategy. Our focus instead is on whether alternating strategy

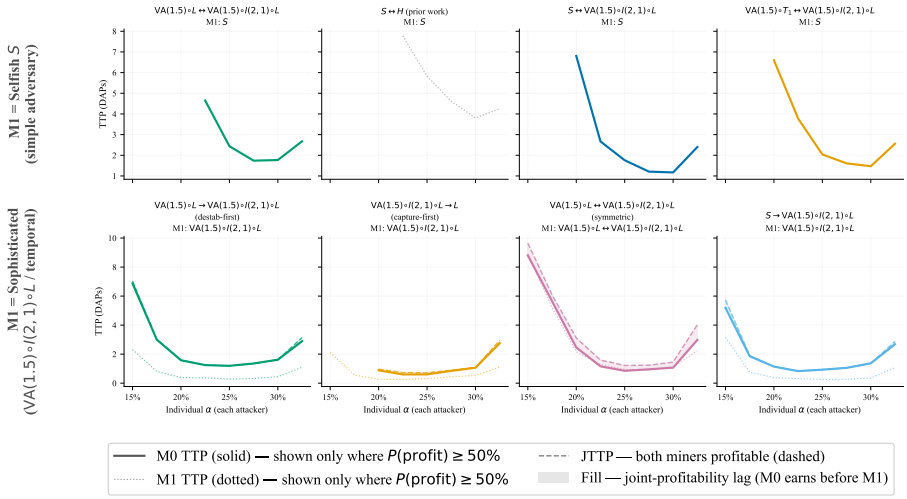


Fig. 7: TTP and JTTP vs. α for eight temporal compositions in the multi-attacker setting ($\alpha = 0.25, \gamma = 0.5, \kappa = 0.25$). Lines appear only where $P(\mathsf{profit}) \geq 50\%$.

choices at fixed DAP intervals improve TTP or profitability over the best static strategy for the same parameter configuration.

We evaluate alternating temporal composition in the multi-attacker setting across two sub-settings that capture the realistic multi-miner environment covering transaction fees, honest-but-rational miners, and whale transactions. In particular, we have (1) attacker 0 M_0 against a classic selfish attacker $M_1 = S$, and (2) M_0 against a sophisticated attacker $M_1 = \mathrm{VA}(1.5){\circ}I(2,1){\circ}L$ or another temporal strategy. Given their advantage in the transaction-fee setting, we focus primarily on value-aware strategy variants.

For each sub-setting, we examine how TTP and JTTP vary with α (shown in Figure 7), then we measure cumRA (shown in Figure 8). All experiments assume whale transactions with $p = 0.05$ and $m = 4\times$, $\gamma \in \{0.0, 0.5\}$, 100 independent runs of 12 DAPs per parameter setting, and $\alpha = 25\%$ per attacker unless stated otherwise. In Figure 7, lines are drawn only where $P(\mathsf{profit}) \geq 50\%$, and so an absent line indicates that the strategy is too unreliable to characterize a meaningful TTP.

Temporal composition against a simple adversary. The first row of Figure 7 shows TTP and JTTP curves for four representative temporal compositions played against a classic selfish opponent $M_1 = S$. Examining classic intermittent selfish mining $S{\leftrightarrow}H$, as introduced in [18], we find that in the multi-attacker setting this strategy produces no visible line at any tested α; less than half of the games yield a positive cumRA, so reliable profitability is never established. The three novel temporal compositions $\mathrm{VA}(1.5){\circ}L{\leftrightarrow}\mathrm{VA}(1.5){\circ}I(2,1){\circ}L$, $S{\leftrightarrow}\mathrm{VA}(1.5){\circ}I(2,1){\circ}L$, and $\mathrm{VA}(1.5){\circ}T_1{\leftrightarrow}\mathrm{VA}(1.5){\circ}I(2,1){\circ}L$ reach TTP with $P(\mathsf{profit}) \geq 50\%$ starting at $\alpha \approx 22.5\%$, 20%, and 20%, respectively. At $\alpha = 25\%$, all three achieve

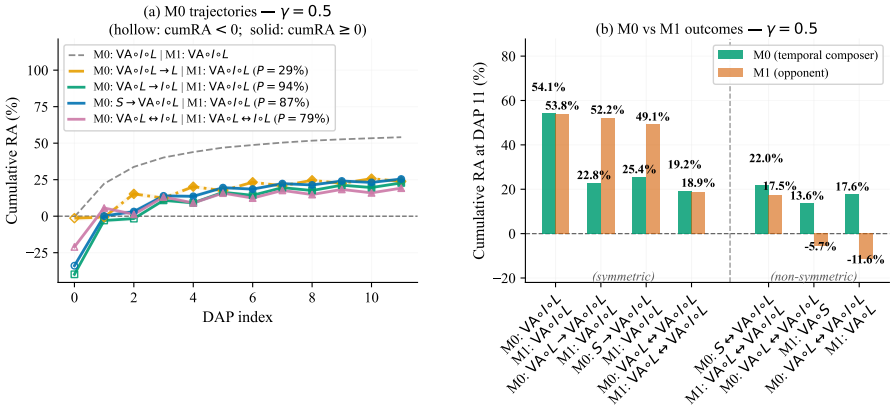


Fig. 8: Cumulative revenue advantage trajectories at $\alpha = 25\%$, $\gamma = 0.5$ for five temporal compositions (left), and M_0 vs. M_1 cumRA for symmetric and non-symmetric strategy pairs (right).

$P(\text{profit}) = 1$, with TTPs of 2.4, 1.8, and 2.0 DAPs, respectively. For reference, static VA(1.5)$\circ I(2,1)\circ L$ is reliably profitable starting at $\alpha = 17.5\%$ and converges in only 0.6 DAPs at $\alpha = 25\%$. Thus, it has an additional 1–2 DAP break-even lag in the temporal variants.

Figure 8 confirms that this break-even cost translates into lower final cumRA. As shown, none of the five temporal compositions in the figure exceeds the static VA(1.5)$\circ I(2,1)\circ L$ variant (+45.5% at $\gamma = 0.5$). The $S\leftrightarrow$VA(1.5)$\circ I(2,1)\circ L$ composition initially tracks the static VA(1.5)$\circ I(2,1)\circ L$ trajectory, then falls progressively short as the classic selfish phase suppresses fee-capture every other DAP.

Temporal composition against a sophisticated adversary. The second row of Figure 7 replaces M_1 with VA(1.5)$\circ I(2,1)\circ L$ (static) or an opponent running a temporal strategy.

Two structural differences from the first row emerge here. First, the hashrate threshold is universally lower: all four compositions reach TTP with $P(\text{profit}) \geq 50\%$ starting at $\alpha \approx 15$–20%, compared to 20–22.5% against $M_1 = S$. Second, phase order is consequential. Destabilize-first (i.e., VA(1.5)$\circ L\rightarrow$VA(1.5)$\circ I(2,1)\circ L$) enters the reliability region at $\alpha = 15\%$ but requires 2.4 DAPs for M_0 to converge at $\alpha = 25\%$. Capture-first (i.e., VA(1.5)$\circ I(2,1)\circ L\rightarrow L$) does not become reliably profitable until $\alpha = 20\%$ yet converges in only 1.2 DAPs once it does. The left panel in Figure 8 illustrates why VA(1.5)$\circ I(2,1)\circ L\rightarrow L$ is the weaker choice. As shown, the trajectory remains near zero for most of the game and the strategy is profitable in only 29% of runs at $\alpha = 25\%$, $\gamma = 0.5$, making it unreliable in practice despite its lower mean TTP. Where as when $M_0 =$VA(1.5)$\circ L\rightarrow I(2,1)\circ L$ or $M_0 = S\rightarrow$ VA(1.5)$\circ I(2,1)\circ L$ and $M_1 =$ VA(1.5)$\circ I(2,1)\circ L$, M_0 is effective

in capturing high-fee transactions and becoming profitable within 0.6 DAPs. In both instances, we find that JTTP is dictated entirely by the TTP of M_0.

The right panel in Figure 8 shows cumRA for both symmetric and non-symmetric strategy pairs. Under symmetric temporal adoption both miners earn $\approx 39.5\%$ cumRA at $\gamma = 0.5$, compared to $\approx 58\%$ each under symmetric static case of VA(1.5)$\circ I(2,1)\circ L$. The symmetric case, when both miners using VA(1.5)$\circ L \leftrightarrow$VA(1.5)$\circ I(2,1)\circ L$, yields M_0 TTP = 1.7 DAPs and M_1 TTP = 1.9 DAPs at $\alpha = 25\%$. The bar chart highlights this finding explicitly, with mutual adoption of the temporal strategies reducing per-miner revenue below what either could achieve by unilaterally adopting the static strategy.

The $S \rightarrow$VA(1.5)$\circ I(2,1)\circ L$ composition against M_1 running VA(1.5)$\circ I(2,1)\circ L$ offers an alternative result trend, starting as S and alternating with VA(1.5) $\circ$ $I(2,1) \circ L$. At $\alpha = 25\%$, M_1 converges in 0.5 DAPs and M_0 in 1.9 DAPs, with M_0's delayed transition accounting for the JTTP lag. Despite this, $S \rightarrow$VA(1.5) $\circ$ $I(2,1) \circ L$ is reliably profitable starting at $\alpha = 15\%$ and offers comparable cumRA to VA(1.5)$\circ L \leftrightarrow$VA(1.5)$\circ I(2,1)$ against M_1 =VA(1.5)$\circ I(2,1)\circ L$.

Finding: Temporal composition rarely outperforms the best static strategy. Across the regimes tested, no alternating temporal composition matches either the TTP or the asymptotic revenue of static VA(1.5)$\circ I(2,1)\circ L$. Against a classic selfish miner at $\alpha = 25\%$, the four temporal compositions' lowest TTP across hashrate is still 1.8–2.4 DAPs against the static strategy's 0.6 DAPs, which is a 3–4$\times$ slowdown, and none of the five temporal variants in Figure 8 reach this TTP nearly as frequently across the examined hashrates as the static strategy. Against a sophisticated adversary, the picture does not improve much despite producing a lower TTP and smaller lag on average (reinforcing our findings in Section 5). As shown, all miners TTP still fall short of matching that of VA(1.5)$\circ I(2,1)\circ L$.

Where temporal composition does help. The framework still admits compositions over observable shifts in α or γ, as noted at the beginning of this section; that is, an attacker whose connectivity or hashrate genuinely changes mid-attack does benefit from remapping its strategy. The negative finding here is narrower and specific. In particular, alternating between strategies at fixed DAP intervals, in a stationary environment, does not outperform the best static strategy identified in Sections 4 and 5.

7 Adaptive Strategy Selection at the DAP Level

The temporal composition results from the previous section show that fixed-schedule strategy alternation rarely beats the best static strategy. A natural follow-up question is whether *state-conditioned* strategy selection, i.e., in response to observed network conditions, would perform better. Existing controllers that targeted similar problem are typically hand-derived policies or per-environment DRL pipelines [5, 13], both of which require substantial engineering investment and, for DRL, per-environment retraining.

In this section, we investigate whether a per-DAP selector can instead be built directly from the strategy knowledge developed in Sections 5–6, and how much the selector's *implementation*, i.e., the mechanism that turns encoded knowledge into a decision, matters once that knowledge is established. We study two selectors that share an identical interface (state summary in, strategy name or NO-CHANGE out) but differ in how they apply the encoded strategy knowledge. The first is a deterministic fixed policy that we manually crafted using the findings of prior work [11] and those in Sections 5 to 6. In particular, we map the current state into an ordered decision tree according to various applicability conditions and thresholds. While the second is an LLM agent that exercises judgment over the material at each call.

Both selectors draw on the same underlying domain knowledge but receives it at different levels of specification. That is, the fixed policy requires that prior knowledge be hand-translated into explicit, checkable decision rules, while the AI agent receives it as high-level context and performs that translation itself on per-call basis. Varying this level of encoding, under the same knowledge and surrounding control loop, isolates what depends on the knowledge itself from what depends on how, and by whom, it is turned into an action. To the best of our knowledge, no prior work has examined either LLM-driven selfish mining policies or deterministic strategy-level adaptive strategy selectors.

Motivation and impact. We frame this as a proof-of-concept exploration with three motivations and one broader implication as follows.

Feasibility. We want to understand whether such an adaptive selector can identify dominant selfish mining strategies from observable signals alone, react to mid-run shifts, and perform against an unknown competing attacker.

Implementation impact. Given identical inputs and encoded knowledge, we aim to investigate whether an agentic selector reach different decisions and profitability compared to a selector that mechanically applies a fixed rule set.

Selector limiting factors. We wish to understand what limits a per-DAP selector regardless of implementation: whether starting with a non-performant strategy imposes unrecoverable costs, where a mid-run network shift can be recovered once detected, and whether these limits are environment or selector related properties.

Implication: real-world traction. Custom DRL-based selfish mining controllers require DRL expertise, and manually-crafted selectors require extensive exploration to identify decision boundaries. LLM agents lower this barrier substantially: a strategy-selector agent requires only a system prompt and an API key. Even if at some performance cost (in terms of lower profitability) compared to hand-trained DRL/manually-crafted policies, such agents enlarge the population of attackers for whom adaptive selfish mining is operationally feasible.

Architecture. Figure 9 illustrates the system. The left panel shows the per-game setup, where we have four miner classes: selector-controlled attacker M_0, a fixed-strategy attacker M_1, honest-but-rational miner M_2, and honest miner M_3. They are instantiated when the game begins. The right panel shows the per-DAP control loop. A lightweight daemon runs alongside the simulator, triggered once

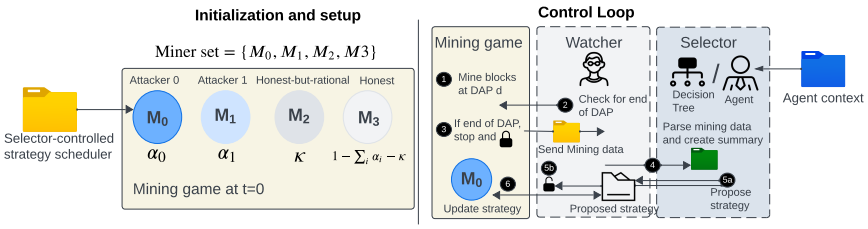


Fig. 9: Per-DAP control loop for adaptive strategy selection.

per completed DAP via a file-system lock (steps 1–3). At each trigger, this daemon reads two rolling CSV logs: a 200-block sliding window ($\approx 10\%$ of one DAP, ≈ 33 hours at target difficulty) supplying fee-variance and whale-event signal, and an epoch-level log retaining the two most recently completed DAPs, supplying orphan rate, block fraction, and seconds-per-block. From these, the daemon assembles a state summary—orphan rate, hashrate shares, γ, fee distribution, cumulative rewards—and passes it to the selector (step 4). The decision is written to a handoff file (step 5a); the simulator resumes on lock deletion and applies the strategy to M_0 (step 5b).

Selectors. For the AI agent, we use Anthropic's `claude-sonnet-4-6` [1], invoked with a fixed system prompt, i.e., the "Agent context" input in Figure 9. From a developer's perspective, this context needs only supplying a high-level specification of the strategy space, applicability conditions, and a reference to prior known performance results under various model settings. As such, based on this context, the agent will reason over the current state at call time.

Concretely, the given prompt consists of four parts: (1) the full strategy space (base, value-aware, and incentivized variants) with per-strategy applicability conditions, e.g., incentivized variants require $\kappa > 0$ and per-block fee variation; (2) a reference section giving the joint-profitability thresholds for two attackers across the (γ, κ) space, expected per-epoch revenue at representative α, and structural orphan-rate baselines separating competition noise from signal worth acting on; (3) the daemon's observation summary; and (4) an output constraint. This material is supplied as background context rather than as an executable rule list (as in the fixed policy); the prompt characterizes when strategies tend to apply and how they tend to perform. The exact translation from context to action is left to the model at time of call. Moreover, within a run, the agent also carries forward a record of which strategies it has already selected and how they performed, and can revise course mid-run if a previously chosen strategy under performs.[2]

[2]In terms of monetary cost, per-DAP inference costs $\approx$ \$0.000005 (9,909-token cached system prompt, 455 uncached input and around 26 output tokens per call at `claude-sonnet-4-6` pricing of \$3/\$15 per MTok). Hence, a twelve-epoch run costs around \$0.0001, which is considered negligible compared to mining pool operational costs in practice.

For the fixed policy, it is implemented as an ordered list of conditional rules derived from the same base knowledge provided to the agent (i.e., strategy space, applicability conditions, past reference section, and current blockchain state). This policy demands considerably more integration work than the agent, as this background knowledge must be manually translated into an ordered list of rules, each with an explicit, checkable condition (organized as a decision tree) for when it fires. We hand-derived this rule set from findings of the present work as well as individual thresholds of prior work [11]. As conditions must be checkable from the state summary alone, they rely on coarse interpretations, i.e., interpreting κ as present or not, and defaulting to $\gamma = 0.5$ when it cannot be estimated.

The base knowledge was fixed before either selector was run and was not tuned against either selector's results. Neither selector currently carries information *across* runs—the agent's adaptivity is confined to the strategies it has tried within a single run, and decision tree's rule of the fixed policy are static once set. We leave fine tuning the agent's context or the decision tree's conditions based on accumulated outcomes across runs as future work.

Experimental design. We construct two scenarios, each bounded by static baselines, and run both selectors on every configuration:

- In the *stable environment* ($\gamma = 0.5$, $\kappa = 0.25$, whale fees ($p = 0.05$ and $m = 4\times$), $\alpha \in \{0.10, 0.20, 0.30\}$, $n = 10$ runs each), the selector must arrive at the dominant strategy from the observation summary alone, without direct visibility of M_1's strategy. We compare against static $VA(1.5) \circ I_{2,1} \circ L$ and report head-to-head win rates against M_1.
- In the γ-*switching* scenario (γ steps from 0.25 at DAPs 0–5 to 0.75 at DAPs 6–11; $\kappa = 0.25$; whale fees ($p = 0.05$ and $m = 4\times$); $\alpha \in \{0.10, 0.20, 0.30\}$; $n = 10$ each), we test whether a selector detects and exploits a mid-run propagation shift.

We use two static strategies to bound performance: $VA(1.5) \circ I_{2,1} \circ L$ from block 0 (upper bound, the best-performing strategy in the multi-attacker setting) and $VA(1.5) \circ L$ (lower bound, fee-aware but not incentive-aware). We run each scenario under two initial conditions (seeded on $VA(1.5) \circ I_{2,1} \circ L$ and cold-started on $VA(1.5) \circ L$) since the selector's opening commitment turns out to be decisive.

Stable environment. Section 5 identifies $VA(1.5) \circ I_{2,1} \circ L$ as strongly dominant under whale fees, and both selectors (when seeded with it) never leave; NO-CHANGE in all 330 decisions for each, tracking the static run's cumRA within run-to-run variance and winning 10/10 against M_1 at $\alpha \geq 0.20$ (Table 3). Instead, starting on $VA(1.5) \circ L$, the picture depends sharply on α. In particular, both selectors adopt the incentive variant only when $\alpha = 0.30$, at the first DAP boundary, and issue NO-CHANGE for all twelve epochs at $\alpha \in \{0.10, 0.20\}$, leaving those runs as static $VA(1.5) \circ L$ measurements. Neither selector diverges from the other in this scenario, where given the same base knowledge, both reach the same decisions at every trigger.

γ-switching. What neither selector can resolve is the asymmetry introduced by the initial seed. Figure 10 shows the per-epoch cumRA when γ changes from 0.25

Table 3: Stable environment ($\gamma = 0.5$, $\kappa = 0.25$), $n = 10$ runs, cumRA is computed over 12 epochs; parentheses = represents ± 1 std. (— = never profitable within 12 epochs). Win rate: runs (of 10) in which M_0 beats $M_1 = S$ head-to-head (recall S is classic selfish mining). Paired rows are the LLM agent and the fixed policy; slashed entries read LLM / fixed.

	α		
	0.10	0.20	0.30
Static VA$\circ I \circ L$	$-33.4\%_{(4.0)}$	$+24.4\%_{(4.1)}$	$+55.7\%_{(3.8)}$
TTP (epochs)	—	0.9	0.8
LLM, ($VA \circ I \circ L$ seed)	$-33.5\%_{(3.0)}$	$+25.1\%_{(4.3)}$	$+56.7\%_{(3.4)}$
Fixed policy, ($VA \circ I \circ L$ seed)	$-33.3\%_{(2.9)}$	$+24.2\%_{(2.2)}$	$+55.5\%_{(3.6)}$
TTP (epochs)	—	0.8/0.9	0.7/0.7
LLM, ($VA \circ L$ seed)	$-15.6\%_{(2.6)}$	$-2.5\%_{(1.5)}$	$+7.4\%_{(3.1)}$
Fixed policy, ($VA \circ L$ seed)	$-15.1\%_{(1.9)}$	$-3.0\%_{(1.4)}$	$+7.3\%_{(1.8)}$
TTP (epochs)	—	4.0/6.7	2.6/2.7
Win rate /10, ($VA \circ I \circ L$ seed)	0/0	**10/10**	**10/10**
Win rate /10, ($VA \circ L$ seed)	**8/8**	**7**/3	4/5

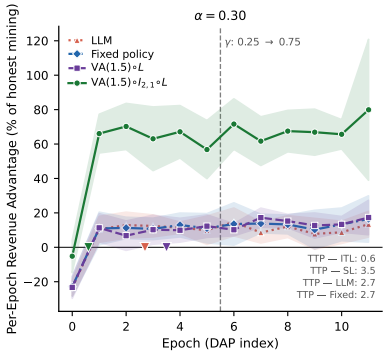


Fig. 10: Per-epoch cumRA in the γ-switching scenario at $\alpha = 0.30$. Dashed line marks the γ: 0.25 $\rightarrow$ 0.75 transition at epoch 5. Band represents ± 1 std. Both selectors are seeded on $VA(1.5) \circ L$.

to 0.75 during epoch 5. The two selectors behave similarly and reach profitability at the same epoch (TTP= 2.7). In general, we find that switching mid-run forfeits gains relative to a strategy determined dominant from the start. That is, seeded on $VA(1.5) \circ L$, cumRA is $-1.8\% \pm 1.4$ at $\alpha = 0.20$ and $+7.0\% \pm 2.6$ at $\alpha = 0.30$. While a selector seeded on the dominant strategy correctly issues NO-CHANGE in every epoch of every run and closely tracks it: having a cumRA of $+25.1\% \pm 4.3\%$ at $\alpha = 0.20$ and $+56.7\% \pm 3.4\%$ at $\alpha = 0.30$. When combining this asymmetry with γ-switching, the picture is less optimistic. As shown, seeded on $VA(1.5) \circ L$, the per-epoch cumRA for epochs 6–11 is 1–2%, matching $VA(1.5) \circ L$ rather than the static incentive strategy's $\approx 6\%$/epoch from epoch 1. cumRA at $\alpha = 0.3$ is $+6.3\% \pm 2.0\%$, which is near the `va-L` lower bound ($+7.2\% \pm 2.4\%$) and far below the $VA(1.5) \circ I_{2,1} \circ L$-seed upper bound. Thus, we observe that both selector correctly notice the network change (despite a DAP delay), and switch strategy accordingly, but this switch cannot recover the cumRA gains forfeited by not having started on the dominant strategy.

Insight: Automated strategy selection trades profitability for adaptability, independent of implementation. Both selectors succeed at identifying the dominant strategy from observable signals without direct visibility of M_1. Also, each detects and responds to mid-run parameter shifts—capabilities unavailable to any static strategy. The LLM agent achieves that with only high-level specification, and without requiring an expert-derived deterministic policy, which significantly simplifies integration. Moreover, both selectors incur the same two costs: a cumRA shortfall from from the initial seed asymmetry, and a delay in TTP in responding to the γ-shift. An automated selector that does not know the

right strategy in advance must spend epochs discovering it, and one that does know cannot anticipate a shift before it happens. This is not a defect of any one selector; it is inherent to per-DAP adaptive selection, whether the decision procedure is a fixed policy or an agentic one. As Section 6 shows, the bar to dominate a static strategy is already high. That is, the gains from adaptability have to exceed the cost of discovery and the profit lag caused by strategy switching, and neither implementation we test changes where that bar sits.

8 Conclusion

In this paper, we revisited selfish mining through the lens of TTP, asking not only what an attacker can earn asymptotically but when and how reliably they realize such profits. We provided a holistic study across several dimensions covering the single-attacker setting, multi-attacker competition, temporal composition, and automated per-DAP strategy selection. Our results show that profitability timing behaved differently compared to long-term revenue trends. In particular, TTP-minimizing and profit-maximizing strategies frequently diverge, making attack horizon a leading metric rather than an afterthought. Also, incentivized recruitment of honest-but-rational miners, rather than value-awareness or temporal composition, is the main driver of shortening the path to profitability. Furthermore, the performance of a fixed rule-based adaptive per-DAP strategy selector matches that of an LLM agent one when both are given the same strategy knowledge. Together, these results point future work away from richer temporal schedules and toward state-conditioned strategy selection, as well as studying TTP across varying network assumptions and models.

Acknowledgments

This work is supported by the National Science Foundation (NSF) under grant No. CNS-2226932.

References

1. Anthropic: Claude Sonnet 4.6 System Card. Tech. rep., Anthropic (February 2026), https://www-cdn.anthropic.com/78073f739564e986ff3e28522761a7a0b4484f84.pdf
2. Bahrani, M., Weinberg, S.M.: Undetectable selfish mining. In: ACM Conference on Economics and Computation (2024)
3. Bai, Q., Xu, Y., Liu, N., Wang, X.: Blockchain mining with multiple selfish miners. IEEE Transactions on Information Forensics and Security **18**, 3116–3131 (2023)
4. Bai, Q., Zhou, X., Wang, X., Xu, Y., Wang, X., Kong, Q.: A deep dive into blockchain selfish mining. In: IEEE international conference on communications (ICC) (2019)
5. Bar-Zur, R., Abu-Hanna, A., Eyal, I., Tamar, A.: Werlman: To tackle whale (transactions), go deep (rl). In: IEEE Symposium on Security and Privacy (SP) (2023)

6. Carlsten, M., Kalodner, H., Weinberg, S.M., Narayanan, A.: On the instability of bitcoin without the block reward. In: ACM SIGSAC Conference on Computer and Communications Security (CCS) (2016)
7. Davidson, M., Diamond, T.: On the profitability of selfish mining against multiple difficulty adjustment algorithms. Cryptology ePrint Archive (2020)
8. Doger, M., Ulukus, S.: Incentive attacks in btc: Short-term revenue changes and long-term efficiencies. arXiv preprint arXiv:2511.11538 (2025)
9. Doger, M., Ulukus, S.: When should selfish miners double-spend? IEEE Transactions on Information Theory (2026)
10. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: International Conference on Financial Cryptography and Data Security (2014)
11. Finkbeiner, C., Najd, M.E., Guskind, J., Almashaqbeh, G.: Sok: Time to be selfless?! demystifying the landscape of selfish mining strategies and models. In: IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (2025)
12. Grunspan, C., Pérez-Marco, R.: On profitability of selfish mining. arXiv preprint arXiv:1805.08281 (2018)
13. Hou, C., Zhou, M., Ji, Y., Daian, P., Tramèr, F., Fanti, G., Juels, A.: Squirrl: Automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning. In: Network and Distributed Systems Security (NDSS) (2021)
14. Keller, P., Bissias, G.: Generic selfish mining mdp for dag protocols. arXiv preprint arXiv:2309.11924 (2023)
15. Liu, H., Ruan, N., Du, R., Jia, W.: On the strategy and behavior of bitcoin mining with n-attackers. In: Asia Conference on Computer and Communications Security (AsiaCCS) (2018)
16. Marmolejo-Cossío, F.J., Brigham, E., Sela, B., Katz, J.: Competing (semi-) selfish miners in bitcoin. In: ACM Conference on Advances in Financial Technologies (AFT) (2019)
17. Nayak, K., Kumar, S., Miller, A., Shi, E.: Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In: IEEE European Symposium on Security and Privacy (EuroS&P) (2016)
18. Negy, K.A., Rizun, P.R., Sirer, E.G.: Selfish mining re-examined. In: Financial Cryptography and Data Security (2020)
19. Sapirshtein, A., Sompolinsky, Y., Zohar, A.: Optimal selfish mining strategies in bitcoin. In: Financial Cryptography and Data Security (2017)
20. Sarenche, R., Zhang, R., Nikova, S., Preneel, B.: Selfish mining time-averaged analysis in bitcoin: Is orphan reporting an effective countermeasure? IEEE Transactions on Information Forensics and Security **20**, 449–464 (2024)
21. Schwarz-Schilling, C., Li, S.N., Tessone, C.J.: Stochastic modelling of selfish mining in proof-of-work protocols. Journal of Cybersecurity and Privacy **2**(2), 292–310 (2022)
22. TheMinerMag: Bitcoin transaction fees drop below 1% of block rewards (Jun 2025), `https://www.theminermag.com/news/2025-06-16/bitcoin-transaction-fees`
23. Xia, Q., Dou, W., Xi, T., Zeng, J., Zhang, F., Wei, J., Liang, G.: The impact analysis of multiple miners and propagation delay on selfish mining. In: IEEE Annual Computers, Software, and Applications Conference (COMPSAC) (2021)
24. Yu, J., Gao, S., Song, R., Cai, Z., Xiao, B.: Partial selfish mining for more profits. arXiv preprint arXiv:2207.13478 (2022)
25. Zhang, R., Preneel, B.: Publish or perish: A backward-compatible defense against selfish mining in bitcoin. In: Topics in Cryptology–CT-RSA (2017)

26. Zhang, S., Zhang, K., Kemme, B.: Analysing the benefit of selfish mining with multiple players. In: IEEE International Conference on Blockchain (Blockchain) (2020)
27. Zhang, Y., Liu, M., Guo, J., Wang, Z., Wang, Y., Liang, T., Singh, S.K.: Optimal revenue analysis of the stubborn mining based on markov decision process. In: International Conference on Machine Learning for Cyber Security (2022)

A Robustness of Parameter Configuration

We assess the robustness of two parameter choices underlying our single-attacker results—the fraction of honest-but-rational miners, κ, and the transaction-fee ratio—both of which support the incentive mechanism's findings. Fixing $\alpha = 0.25$ to match Section 4, we sweep $\kappa \in \{0.05, 0.10, 0.15, 0.20, 0.25, 0.33\}$ and the fee ratio (fees as a fraction of total block reward) over $\{1, 5, 10, 17.5, 30\}\%$ (baseline 17.5%), each at $\gamma \in \{0, 0.5\}$ over 100 games of 12 DAPs. We report the cumRA between the value-aware incentive strategy VA(1.5)$\circ I(2,1)\circ L$ and VA(1.5)$\circ L$ under whale transactions (p=0.05, m=4×), as well as $I(2,1)\circ T_1$ versus T_1.

Honest-but-rational fraction. The advantage is positive and significant at every κ, and is not unique to $\kappa = 0.25$: it measures +36.9% at $\kappa = 0.05$ and declines gently to +30.4% at $\kappa = 0.33$ ($\gamma = 0.5$) as the non-incentive baseline increasingly captures rational hash rate on its own. Profitability simultaneously becomes more reliable and faster as the pool of honest-but-rational miners grows—over this range the value-aware incentive strategy's probability of profitability rises from 0.48 to 1.00 and its TTP falls from $\approx$ 10 to 0 DAPs—confirming that the mechanism is actively recruiting the honest-but-rational miner. On the other hand, in the case of $I(2,1)\circ T_1$ versus T_1, a declining trend is found with profitability increasing as κ approaches 0.25 ($\gamma = 0$) and then turning slightly negative (−4.0%) by $\kappa = 0.33$—consistent with the observation in [11] that too many honest-but-rational miners become self-defeating (shown at $\kappa = 0.5$).

Fee ratio. We adopt the elevated 17.5% baseline following prior fee-aware selfish-mining analyses [5, 6], which study regimes where transaction fees form a substantial share of block value—highlighting the future impact of Bitcoin increasingly dependent on transaction fees for incentives. Across the full 1–30% range the cumulative advantage is positive and significant at every level (+46.4% to +23.1% at $\gamma = 0.5$; +64.4 to +39.2 pp at $\gamma = 0$), and the incentive strategy is reliably profitable where its non-incentive analog is not: at $\gamma = 0$ the baseline turns a profit in under 11% of runs at *every* fee level, against roughly 50% for the incentive strategy. This examination highlights that again our incentivized strategies' findings are robust to parameter choice, both for Bitcoin today and well into its maturity.