Mechanism Design for Hashrate Externalities: Implementing Optimal Security in Proof-of-Work

By DANIEL ARONOFF[*]

We develop a mechanism that implements the socially optimal hashrate in a Proof-of-Work system by adjusting block rewards in response to observed network effort. The mechanism embeds a Pigouvian tax/-subsidy schedule directly into the protocol using only public, on-chain data. We prove equilibrium existence, convergence, and robustness to noisy measurement. Applied to Bitcoin, the mechanism yields a reward rule—Targeted Nakamoto—that balances security and carbon externalities while preserving monetary neutrality. The result is a decentralized implementation of optimal effort under limited observability.

Keywords: Proof-of-Work, Blockchain, externalities, mechanism design, network security, carbon emissions.

I. Introduction

Proof-of-Work (PoW) blockchain networks, such as Bitcoin, rely on decentralized miners expending computational effort (hashrate) to validate transactions and secure the network (Nakamoto, 2008). Miners receive block rewards (newly minted cryptocurrency and transaction fees) as compensation for their costly effort. A higher total hashrate improves network security by making attacks (e.g., double-spending or majority attacks) more difficult and expensive (Nakamoto, 2008; Gervais et al., 2016). However, mining is energy-intensive and imposes negative externalities: increased electricity consumption raises carbon emissions and can elevate energy prices for other consumers (Sedlmeir et al., 2020; Gallersdorfer et al., 2020). Thus, there is a social trade-off: additional hashrate reduces the probability of successful attacks (a positive externality) but increases environmental damages (a negative externality).

In current PoW protocols (the Nakamoto protocol), block rewards are fixed exogenously (apart from periodic halvings in Bitcoin) and there is no explicit mechanism to target or cap the total hashrate (Aronoff, 2024). The equilibrium hashrate is determined by miners' private profit motives: miners will keep adding

[*] MIT Department of Economics (daronoff@mit.edu);

hashrate until the marginal revenue (from block rewards) equals the marginal private cost of mining (cf. Kroll et al., 2013; Cong et al., 2021). This decentralized equilibrium generally does not coincide with the social optimum, because miners do not internalize the security benefits conferred to others, nor the environmental costs they impose on society. Indeed, at present the incentive structure may lead to excessive energy usage ((Cambridge Centre for Alternative Finance, 2024; Stoll et al., 2019)) and political backlash over carbon footprints, whereas in the future declining block rewards could lead to insufficient security (Carlsten et al., 2016).

This paper proposes a mechanism design approach to regulate PoW mining such that the equilibrium hashrate aligns with the socially optimal level. We introduce a reward adjustment mechanism—a modification to the Nakamoto protocol—that dynamically adjusts the block reward based on the observed total hashrate. Intuitively, the mechanism acts like a Pigouvian subsidy or tax: if hashrate is below the optimal target, the protocol temporarily increases miners' rewards (a subsidy) to encourage more mining; if hashrate is above the target, the protocol reduces rewards (a tax) to discourage excessive mining. By construction, this mechanism guides the decentralized system to the hashrate that balances security and environmental externalities.

We provide a formal model of strategic miners and derive conditions for equilibrium and optimal hashrate. We then construct a simple reward adjustment rule that implements the optimal hashrate as a Nash equilibrium. We prove that under this mechanism, the target hashrate is the unique equilibrium outcome. We also analyze the dynamic behavior: as miners respond to reward adjustments over time, the hashrate converges to the target (under mild assumptions on best-response dynamics). Our analysis highlights the role of block rewards as an instrument for protocol designers to achieve desired economic outcomes in a decentralized system.

After presenting the general model, we illustrate the approach with an application to Bitcoin. The proposed "Targeted Nakamoto" protocol, originally put forward by Aronoff (2024), specifies a target hashrate (or range) and modifies Bitcoin's block reward policy accordingly. When the network's hashrate exceeds the target range, the protocol imposes a cap on the block reward (lowering the reward to miners); when hashrate falls below the target, the protocol sets a floor on the block reward (increasing the reward). Crucially, these adjustments are offset by modifications to the supply held by users (UTXO holders) to maintain monetary neutrality: any additional rewards given to miners when hashrate is low are effectively taken from all currency holders, and any rewards withheld when hashrate is high are redistributed so that the total money supply and its distribution are unchanged in net present value. We show that Targeted Nakamoto can be viewed as a special case of our general mechanism, and we discuss its implementation details and potential impact on Bitcoin's long-run security and sustainability.

ROADMAP.. — Section II introduces the formal model of the mining game with externalities and characterizes the competitive mining equilibrium and the social optimum. Section III presents the reward adjustment mechanism and provides theoretical results on equilibrium implementation and stability (Propositions III.B–III.B). Section IV connects the general mechanism to the Targeted Nakamoto proposal for Bitcoin, including a description of how the policy can be implemented in practice. Section V discusses related literature. Section VI concludes. All proofs are collected in the Appendix.

II. Model

We consider a Proof-of-Work mining setting with a continuum of strategic miners. Time is modeled in epochs (discrete numbers of blocks); within each epoch, a large number of mining rounds occur and the block reward is held fixed. We focus on the equilibrium within a representative epoch given a certain block reward policy, and later discuss dynamics across epochs when the reward is adjusted.

Each miner i chooses a nonnegative hashrate effort $h_i \geq 0$, incurring a cost $C(h_i)$ measured in fiat currency (e.g., USD). We assume $C(h)$ is increasing and convex, and $C(0) = 0$. For tractability, one may assume a linear cost $C(h) = c\,h$ where $c > 0$ is the unit cost of hashing power (reflecting electricity and hardware costs).

We assume that miners myopically adjust hashrate between epochs, to maximize profit within the epoch. Formally, this corresponds to zero profit representing the shadow profit from an alternative use of computing power and the ability to costlessly shift between computing projects.

Let $H = \sum_i h_i$ denote the total network hashrate. The probability that miner i wins the mining reward (i.e., finds the next valid block) is h_i/H, proportional to their share of hashrate (if $H = 0$, the network is inactive and no rewards are won). The reward for mining a block is specified by the protocol. In a given epoch, let R denote the total reward per block, in cryptocurrency units (this includes the block subsidy and possibly aggregated transaction fees, converted to a common value unit). We assume all miners value rewards in monetary terms (implicitly, they can convert cryptocurrency to USD at some exchange rate).[1]

[1] Miner revenue R and cost c are both valued in USD.

Thus, given total hashrate H, miner i's expected payoff (per block) is:

$$u_i(h_i, H) = \frac{h_i}{H} \cdot R \;-\; C(h_i), \tag{1}$$

for $H > 0$ (and $u_i(0, 0) = 0$ by convention). Each miner chooses h_i to maximize u_i, taking other miners' efforts (hence H) as given. We focus on symmetric Nash equilibria where all active miners choose the same effort.

Mining equilibrium without intervention.. — Under the baseline Nakamoto protocol, R is exogenous (determined by the protocol's monetary policy and the prevailing transaction fees). In a competitive mining equilibrium, miners will enter or adjust h_i until profits are driven to zero (if positive profit opportunities exist, new hashrate will be added; if negative, miners will drop out). Assuming an interior equilibrium with $H > 0$, the equilibrium condition is that each miner's marginal profit is zero:

$$\frac{\partial u_i}{\partial h_i} = \frac{R}{H} - C'(h_i) = 0. \tag{2}$$

In a symmetric equilibrium with n identical miners each choosing $h_i = h^*$ (so $H = nh^*$), $C'(h^*) = R/H$. If C is linear ($C'(h) = c$ constant), this yields $R/H = c$, or

$$H^{\mathrm{N}} = \frac{R}{c}, \tag{3}$$

as the equilibrium total hashrate in the baseline protocol (assuming free entry of miners) (cf. Kroll et al., 2013; Cong et al., 2021). More generally, if $C'(\cdot)$ is increasing, Eq. (2) defines an equilibrium H^{N} implicitly via $C'^{-1}(R/H)$. We assume conditions such that a unique equilibrium hashrate H^{N} exists.

This decentralized equilibrium H^{N} typically does not account for external costs or benefits of mining. We now formalize these externalities and the socially optimal hashrate.

Externalities and social optimum.. — Mining activity H affects two key external metrics:

- Network security: A higher hashrate improves security by increasing the cost for an attacker to overpower honest miners. Let $S(H)$ denote the security cost to society (e.g., expected loss from attacks) as a function of H. We assume $S(H)$ is decreasing in H (diminishing marginal benefit of extra hashrate) and differentiable, with $S'(H) < 0$.

- Environmental damage: Higher H means more energy consumption and carbon emissions. Let $E(H)$ be the total environmental cost (social cost of carbon, pollution, etc.) associated with hashrate H. We assume $E(H)$ is increasing in H (and $E'(H) > 0$).

We can define the total externality cost as $X(H) = S(H) + E(H)$, which is the sum of a decreasing and an increasing function. Under standard assumptions, there exists an interior minimizer of $X(H)$, where the marginal security benefit equals the marginal environmental cost:

$$-S'(H^*) = E'(H^*). \tag{4}$$

The level H^* satisfying (4) is the socially optimal hashrate (or one such optimum, if the minimizer is not unique).[2] At H^*, the total external cost $X(H)$ is minimized, achieving a balance between security and environmental impacts. We assume H^* exists and is unique for clarity of exposition.[3]

A key technical feature of proof-of-work systems is that the total network hashrate H_t can be estimated directly from public blockchain data, without any external oracle. Given the mining difficulty parameter D_t in epoch t and the stochastic structure of proof-of-work, the expected inter-block time satisfies $\mathbb{E}[T_t] = D_t/H_t$. This yields the standard estimator

$$\widehat{H}_t \ = \ \frac{D_t}{\overline{T}_t}, \tag{5}$$

where $\overline{T}_t$ is the empirically observed average inter-block time in epoch t. This relationship follows from modeling mining as a Bernoulli/Poisson process and is well documented in the literature (Decker and Wattenhofer, 2013; Gervais et al., 2016; Rosenfeld, 2014). Because both D_t and block timestamps are recorded in block headers, the mechanism designer can compute $\widehat{H}_t$ entirely from on-chain data. No trusted oracle is required: the mechanism is constructed purely from network-generated information. Empirical studies show that Bitcoin's resulting electricity consumption and associated CO_2 emissions are on the order of those of a medium-sized country (Stoll et al., 2019; Cambridge Centre for Alternative Finance, 2024), and the broader energy-economics literature demonstrates that large, inelastic industrial loads can place upward pressure on electricity prices faced by other consumers (Fowlie et al., 2016). These findings underscore that

[2] We do not include miner expenditure on hashrate in the computation of the social optimum because miners are already internalizing this cost.

[3] It is possible that there is a flat interval of H in which total cost is minimized. In that case, any H in that range could be considered socially optimal, and a mechanism might target an interval rather than a point. Our analysis can be extended to target a range $[H_{\text{LB}}, H_{\text{UB}}]$ of hashrate by capping or flooring rewards at the boundaries of that interval; see Aronoff (2024) for discussion.

excessive hashrate imposes real environmental and economic costs, motivating a protocol-level solution that internalizes these externalities.

Importantly, the competitive equilibrium H^{N} need not equal H^*. In fact, two types of inefficiencies may arise:

1) $H^{\mathrm{N}} > H^*$ (over-provision of hashrate): When block rewards are high (e.g., currently in Bitcoin, with substantial mining incentives), miners ignore the negative externality $E(H)$, potentially resulting in excessive hashing from society's perspective (too much energy use).

2) $H^{\mathrm{N}} < H^*$ (under-provision of hashrate): In future scenarios where block rewards diminish (e.g., after many halvings or if price declines), the private incentive to mine may be too low. Miners do not account for the positive externality of security, leading to an equilibrium hashrate that is below the socially optimal level needed to secure the network (Carlsten et al., 2016).

In either case, a mechanism is needed to correct the externality. Traditional solutions to such externality problems in economics include Pigouvian taxes or subsidies (Pigou, 1920) or cap-and-trade systems (Fowlie et al., 2016). In our context, the protocol itself can implement an analogous solution by adjusting the block reward.

III. Mechanism: Reward Adjustment Scheme

We now design a mechanism to implement the optimal hashrate H^* as an equilibrium outcome. The mechanism is a rule for adjusting the miners' block reward based on the observed total hashrate in the previous epoch. Conceptually, it adds a feedback loop to Nakamoto's protocol: when H deviates from H^*, the reward is altered to push H back towards H^*.

A. Reward Adjustment Rule

The mechanism specifies a simple threshold policy. Let H^* be the target hashrate (social optimum). The mechanism sets two thresholds around H^*, which could coincide with H^* or form a small interval $[H_{\mathrm{LB}}, H_{\mathrm{UB}}]$ containing H^*:

$$H_{\mathrm{LB}} \leq H^* \leq H_{\mathrm{UB}}.$$

For now, assume $H_{\text{LB}} = H_{\text{UB}} = H^*$ (a single target point); we discuss an interval target later. The block reward in epoch $t+1$, denoted R_{t+1}, is set as a function of the total hashrate observed in epoch t:

$$R_{t+1} = \begin{cases} R_t \cdot (1 + \delta(H^* - H_t)) & \text{if } H_t < H^*, \\ R_t \cdot (1 - \delta(H_t - H^*)) & \text{if } H_t > H^*, \\ R_t & \text{if } H_t = H^*, \end{cases} \tag{6}$$

where $\delta(\cdot) \in (0,1)$, $\delta' < 0$ and $\delta(0) = 0$. Thus, the mechanism raises the reward by a factor $(1 + \delta)$ whenever hashrate is below target, and lowers it by factor $(1 - \delta)$ whenever hashrate is above target. This is analogous to a proportional controller in engineering or a tatonnement process in economics. It effectively implements a subsidy for mining when H is too low and a tax when H is too high.

In practice, the protocol can observe H_t via the difficulty adjustment statistics: blockchain data provides an indicator of average hashrate in each epoch (e.g., in Bitcoin the difficulty parameter and block timestamps allow computing H_t). Formally, the protocol uses the on-chain estimator in equation (5), which computes $\widehat{H}_t$ from the difficulty parameter and the empirically observed inter-block times. This estimator is derived entirely from blockchain header data, requiring no trusted oracle: all inputs to the reward-adjustment mechanism are endogenous and publicly verifiable. Because $\widehat{H}_t$ responds mechanically to deviations in block arrival intervals, it provides a timely and protocol-native feedback signal for updating rewards and steering the system toward the target hashrate.

One important consideration is that the cryptocurrency's monetary supply or distribution could be affected by these reward adjustments. If we simply give miners a higher reward when H is low, that creates extra coins (inflation); if we give less when H is high, it removes coins. To preserve monetary neutrality, the mechanism should offset these adjustments elsewhere. A practical solution is to redistribute the "excess" or "shortfall" to all coin holders (for instance, by adjusting coinbase outputs or using a fee pool) such that the total money supply and each holder's proportional share remain unchanged over time (Aronoff, 2024). We do not model the monetary neutrality adjustment in detail here, as it does not affect miners' incentives (it is akin to lump-sum transfers to non-miners). We simply assume any reward adjustment is paired with an opposite adjustment to a public account or distributed pro rata to users, so that the net issuance over time is unchanged. This ensures the mechanism focuses purely on incentivizing hashrate, without altering the long-term supply of currency.

B. *Equilibrium Analysis*

We now show that the reward adjustment mechanism can achieve the target H^* as an equilibrium. First, consider a static one-shot analysis: suppose the mechanism sets a block reward R as a function of H according to some rule $R(H)$. We want to design $R(H)$ such that H^* is a Nash equilibrium of the miners' game. That means if the total hashrate is H^*, miners have no profitable deviation, and any other $H \neq H^*$ cannot be sustained as an equilibrium.

Intuitively, we want H^* to satisfy the miners' zero-profit condition given the reward, and for any H above H^* the reward should be lower so that miners would earn negative profit (making such H unstable), and for any H below H^* the reward should be higher so that there is an incentive for additional mining (making H increase). The threshold rule (6) is a discrete approximation of this idea. For theoretical analysis, one can consider a continuous version: define a reward function $R(H)$ that is high when $H < H^*$ and low when $H > H^*$. For example, let $R(H) = \bar{R} \cdot \phi(H)$ where $\phi(H)$ is a decreasing function that crosses a baseline of 1 at H^*.

The simplest case is a piecewise-constant function:

$$R(H) = \begin{cases} R^+ & \text{if } H < H^*, \\ R^- & \text{if } H > H^*, \\ R^* & \text{if } H = H^*, \end{cases} \tag{7}$$

with $R^+ > R^* > R^-$. We can choose R^* such that if the network hashrate is exactly H^*, miners earn zero profit (this pins down R^* using (2) and H^*). Meanwhile, R^+ is set slightly higher so that if H were below H^*, miners would earn positive profit (inducing entry or expansion of hashrate), and R^- is lower so that if H exceeds H^*, miners earn negative profit (inducing exit or reduction of hashrate). In a game with many miners, no single miner can influence H by a large amount, so they take H as given. Thus:

- At $H = H^*$ with reward R^*, each miner is indifferent to marginally increasing or decreasing their hashpower (zero profit condition holds). So H^* can be an equilibrium outcome.
- If $H > H^*$, the reward would be $R^- < R^*$. At such H, miners' marginal cost is higher than the marginal revenue ($C' > R^-/H$ given H is high and reward is low), so mining is unprofitable for at least one miner (and for all miners if $C(H)$ is linear); hence an $H > H^*$ cannot persist – some miners would drop out until H returns to H^*.

- If $H < H^*$, the reward is $R^+ > R^*$. Then miners have an incentive to increase hashrate (new miners may enter since reward is higher relative to cost), pushing H up toward H^*.

The mechanism thus creates H^* as the sole stable point. This intuitive reasoning can be formalized. We now state our main results.

Proposition 1 (Equilibrium Implementation of H^*)**.** *There exists a block reward function $R(H)$ (with $R(H^*) = R^*$ appropriately chosen) such that H^* is a Nash equilibrium of the mining game. In fact, under the threshold policy* (6) *(with sufficiently small adjustment increments δ), the only Nash equilibrium of the dynamic game is at $H = H^*$. At this equilibrium, miners' expected profit is zero and the social optimum hashrate is achieved.*

Proposition 1 formalizes that our mechanism implements the socially optimal hashrate in equilibrium. The mechanism uses the block reward as a control lever, akin to how a social planner would impose a corrective tax or subsidy to equate private incentives with social optima. Notably, this is achieved without requiring any direct communication or collusion among miners – the protocol itself embeds the incentive.

Next, we consider the dynamic behavior of the system as the reward is adjusted over time. We want to ensure that starting from an arbitrary state, the policy will lead the hashrate to converge to H^*.

Proposition 2 (Convergence of Hashrate Dynamics)**.** *Starting from any initial hashrate H_0, the repeated application of the reward adjustment rule* (6) *will eventually guide the total hashrate into the neighborhood of H^*. In particular, if miners myopically best-respond to the current reward each epoch, then $H_t \to H^*$ as $t \to \infty$. The hashrate may oscillate around H^* in the short run, but H^* (or the target interval $[H_{LB}, H_{UB}]$) is globally attractive and locally stable under the adjustment process.*

We can extend the analysis to the case where there is noise in the estimation of hashrate. This occurs when hashrate is estimated from on-chain data, as it equation (5).

Proposition 3 (Robustness to Estimation Noise)**.** *Let $\widehat{H}_t$ be the observed hashrate estimator defined in equation* (5)*, and suppose $\widehat{H}_t = H_t + \varepsilon_t$, where $\{\varepsilon_t\}$ is a bounded mean-zero error process with $\mathbb{E}[\varepsilon_t] = 0$ and $|\varepsilon_t| \leq \bar{\varepsilon}$. If the adjustment step δ in the update rule* (6) *satisfies $\delta > k\bar{\varepsilon}$ for the corresponding best-response slope $k > 0$, then:*

1) The equilibrium hashrate H^* remains a Nash equilibrium of the mining game.

2) Any perturbed dynamic trajectory $\{H_t\}$ converges to an $O(\bar{\varepsilon})$-neighborhood of H^*.

Thus, small statistical errors in estimating hashrate do not change the equilibrium or the qualitative con-

vergence behavior of the mechanism.

For the next results we restrict attention to the linear-cost benchmark $C(h) = ch$. In this case the free-entry equilibrium condition implies $H = g(R) = R/c$ (equation (3)), so for any $R, R' \geq 0$,

$$|g(R') - g(R)| = \frac{1}{c}\,|R' - R|.$$

Hence g is globally Lipschitz with constant $k = 1/c$, and its (aggregate) best-response slope is $g'(R) = 1/c$.[4]

Lemma 1 (Bound and Scaling of Convergence Error)**.** *Under the assumptions of Theorem* **??** *and equation* (6)*, let $H_{t+1} = f(H_t, \widehat{H}_t)$ denote the aggregate best-response induced by the reward-update rule. Suppose that in a neighborhood of H^*, the mapping f is Lipschitz in its first argument with constant $L \in (0, 1)$ and Lipschitz in its second argument with constant $K > 0$. Then:*

(i) (band.) For all $t \geq 1$,

$$|H_t - H^*| \;\leq\; L^t|H_0 - H^*| \;+\; K\,\bar{\varepsilon}\,\frac{1 - L^t}{1 - L},$$

and therefore

$$\limsup_{t\to\infty} |H_t - H^*| \;\leq\; \frac{K}{1 - L}\,\bar{\varepsilon}.$$

(ii) (with the observation window.) Let m denote the number of blocks used in computing the empirical inter-block time $\overline{T}_t$ in the estimator $\widehat{H}_t = D_t/\overline{T}_t$ of equation (5)*. If the estimator error satisfies $\mathbb{E}[\varepsilon_t^2] = \sigma^2/m$, then*

$$\limsup_{t\to\infty} \mathbb{E}[\,|H_t - H^*|\,] \;\leq\; \frac{K}{1 - L}\,\frac{\sigma}{\sqrt{m}}.$$

In particular, if $\bar{\varepsilon}$ is small and/or m is large, the realized hashrate remains arbitrarily close to the target H^*.

Taken together, these results show that the reward-adjustment mechanism behaves exactly like a well-designed negative-feedback controller for the decentralized mining game. Proposition 1 establishes that the target H^* is the unique zero-profit point consistent with miners' best responses; Theorem **??** adds that no miner can gain by deviating from it. Proposition 2 shows that the reward updates generate directional drift toward H^* whenever the system is away from it. The robustness theorem clarifies that these properties survive imperfect measurement of hashrate: the mechanism reacts correctly whenever the estimation error is not large enough to flip the sign of the deviation, and even when such misclassifications occur, they only

[4] We note, but do not prove, that the block reward adjustment function in equation 6 ensures that g is Lipschitz under less restrictive conditions.

perturb the system within a narrow band. Lemma 1 formalizes this band as a contraction region whose width depends only on the noise magnitude $\bar{\varepsilon}$ (or, under standard Poisson timing, on the sampling variation $\sigma/\sqrt{m}$) and the local Lipschitz constants of the induced best-response dynamics. In particular, the convergence band shrinks at the classical $m^{-1/2}$ rate as the number of observed blocks per epoch grows. Thus, without inspecting any of the proofs, the reader may view the mechanism as creating a stable "gravitational field" around H^*: large departures are pulled back quickly, small perturbations dissipate, and greater sample precision produces tighter control around the socially optimal hashrate.

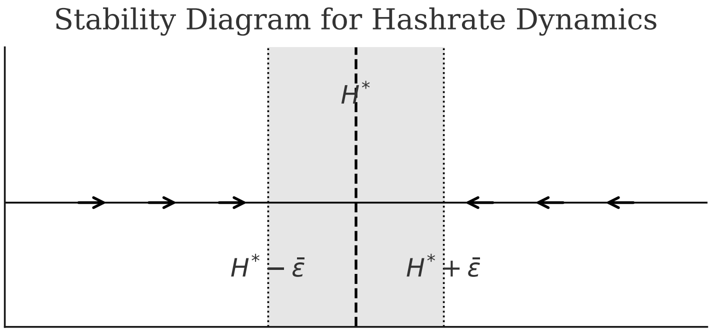
Stability Diagram for Hashrate Dynamics
H^*
$H^* - \bar{\varepsilon}$
$H^* + \bar{\varepsilon}$

Figure 1. : *Stability diagram for the hashrate dynamics.* The figure shows directional drift toward the target hashrate H^* and the noise band $[\,H^* - \bar{\varepsilon},\ H^* + \bar{\varepsilon}\,]$, within which estimation errors may temporarily reverse the adjustment but cannot overturn the long-run stability of the mechanism.

IV. Application: Targeted Nakamoto Protocol

We now translate the general mechanism into a concrete proposal for Bitcoin and similar PoW cryptocurrencies. The *Targeted Nakamoto* protocol, as proposed by Aronoff (2024), is a realization of the reward adjustment mechanism.

A. Protocol Design

Targeted Nakamoto augments Bitcoin's existing rules by introducing a target hashrate range $[H_{\text{LB}}, H_{\text{UB}}]$. For example, this range could be chosen around the estimated social optimum hashrate, or to maintain a certain security level (e.g., cost-of-attack threshold) while limiting energy usage. Suppose Bitcoin's current block reward (set by its halving schedule) is R_{base} BTC. Targeted Nakamoto modifies the coinbase reward paid to miners as follows:

- If the measured hashrate in the last difficulty adjustment period is below H_{LB}, then **increase** the block reward for the next period to $R_{\text{high}} > R_{\text{base}}$. This is a floor on the reward to prevent hashrate from dropping too low.

- If the measured hashrate is above H_{UB}, then **decrease** the block reward for the next period to $R_{\text{low}} < R_{\text{base}}$. This caps the reward to dis-incentivize excessive hashrate.

- If hashrate is within the target range ($H_{\text{LB}} \leq H \leq H_{\text{UB}}$), the block reward remains at the baseline R_{base}.

The parameters R_{high} and R_{low} (or equivalently the adjustment factors) can be calibrated. For instance, the protocol might set $R_{\text{high}} = (1 + \delta)R_{\text{base}}$ and $R_{\text{low}} = (1 - \delta)R_{\text{base}}$ for some δ (and possibly allow further incremental adjustments if deviation persists for multiple epochs).

MONETARY NEUTRALITY.. — When R_{high} is in effect, extra BTC are minted relative to the original schedule. The protocol counterbalances this by effectively "charging" existing holders: one way to implement this is to impose a small proportional decrease in the value of all UTXOs (unspent outputs) when they are spent, proportional to the UTXO balance of each holder (or "address") and in aggregate equal to the extra BTC minted. Conversely, when R_{low} is in effect, the BTC not paid out to miners can be proportionately redistributed to holders (for instance, by increasing every UTXO by a small percentage or by adding the difference to a pool that reduces transaction fees) and added to the available balance at the time a transaction is spent. The precise method is technical, but the result is that (i) the aggregate spending potential is unchanged by the adjustment to the miner block reward and (ii) the change in spending potential is adjusted proportionately for each holder (Aronoff (2024)).

INCENTIVE COMPATIBILITY.. — Targeted Nakamoto leaves intact the core structure of Bitcoin's mining incentives. It does not alter the proof-of-work algorithm, the block time target (10 minutes), or the distribution of rewards among miners and transactors aside from the proportional scaling of all rewards. Transaction fees still function the same way: importantly, the protocol scales the fees and coinbase by the same factor during adjustments, so the ordering of transactions by fee per byte remains unchanged (Huberman et al., 2021; Aronoff, 2024). This means miners still prioritize transactions by highest fee and the allocation of block space remains market-driven and efficient. Because the adjustment is protocol-wide and automatic, there is no scope for individual miners to manipulate it (they cannot misreport hashrate; it is inferred from difficulty and timestamps, which are globally verifiable).

B. Simulation and Practical Considerations

To build intuition and confidence in the Targeted Nakamoto mechanism, Aronoff (2024) provides an open-source implementation and simulation. In collaboration with Praizner, an online API and dashboard[5] allow users to set a target hashrate interval and control parameters and to run counterfactual experiments with alternative time-paths of model variables. These simulations show that the hashrate can be maintained in a tight band around the chosen target, despite large fluctuations in Bitcoin price or mining costs. For example, if Bitcoin's block reward halves and would normally cause a steep drop in hashrate, the policy automatically boosts the reward (temporarily above the scheduled amount) to incentivize miners to stay, thereby preventing a security cliff. Conversely, if Bitcoin's price surges and mining becomes extremely profitable, instead of an unbounded hashrate increase (and energy spike), the protocol would trim the reward to hold hashrate near the target, saving energy with only a slight reduction in miners' revenue relative to the boom scenario.

One concern might be how miners react to a reduction in rewards—could this undermine consensus or lead to a fork? Since the policy is transparent and applies universally, honest miners have no benefit in deviating; indeed, the reduction in reward is exactly what keeps their profit at zero equilibrium—if they attempted a fork without the policy, they'd either face the same equilibrium or worse if others don't follow. Meanwhile, users and society benefit from the controlled hashrate (less environmental impact and sustained security). By design, the protocol changes are backward-compatible with Bitcoin's ethos: they do not rely on any external oracles, do not change the proof-of-work or maximum supply, and do not alter transaction processing rules. In essence, Targeted Nakamoto is a modest tweak to the monetary policy rule, adding a feedback mechanism to what was a fixed schedule. This aligns with prior work on consensus mechanism design that stresses maintaining incentive compatibility and decentralization (Leshno and Strack, 2020).

V. Related Literature

This work connects several strands of literature in economics and the study of blockchain protocols. First, it contributes to the economics of cryptocurrencies by addressing the externalities of mining. Budish (2018) highlighted the tension between ensuring security through massive mining expenditure and the economic cost of that expenditure. Our mechanism can be seen as a way to internalize the externality, akin to proposals for taxing miners or adjusting rewards to reflect social costs. Pagnotta (2021) models the feedback loop between Bitcoin's price, mining investment, and network security; in that framework, our approach introduces a policy intervention that fixes the security level exogenously (via a target hashrate) rather than

[5] See `https://targetednakamoto.com/About` for an interactive simulation tool and the code repository at `https://github.com/Krisp140/TargetedNakomoto`.

letting it fluctuate with market cycles.

Second, our approach is related to the environmental economics literature on regulating externalities in decentralized systems. Traditional tools include Pigouvian taxes (Pigou, 1920) and cap-and-trade schemes (e.g., Fowlie et al., 2016). The Targeted Nakamoto mechanism effectively implements a Pigouvian-like adjustment endogenously within the blockchain protocol, akin to a cap on energy usage (via capping hashrate) but enforced through economic incentives rather than direct quantity restrictions.

Third, this paper builds on a growing literature applying mechanism design to blockchain consensus. Chen et al. (2020) and Cong et al. (2021) provide overviews of how blockchain protocols can be viewed through the lens of incentive compatibility and mechanism design. In particular, Leshno and Strack (2020) prove an impossibility theorem that under certain axioms (such as free entry and decentralization), the only viable reward scheme is one that resembles Bitcoin's Nakamoto protocol (i.e., rewarding each block finder with a fixed payout, as opposed to more complex schemes). Our proposal stays within that paradigm: it does not alter how rewards are allocated to blocks, only how much reward is allocated, as a function of system-wide variables. This ensures that fundamental properties like miner anonymity and free entry are preserved (Leshno and Strack, 2020). Meanwhile, Huberman et al. (2021) analyze the equilibrium of transaction fees in Bitcoin and show that transactions get sorted by fee in a way that reflects users' time preferences. Targeted Nakamoto explicitly preserves this fee ordering by scaling all fees and rewards proportionally, leaving the relative incentive for any single block's transactions unchanged (Aronoff, 2024). This highlights the mechanism's careful design to not distort the micro-incentives of block formation, focusing only on the macro-level incentive (the total hashpower).

Finally, alternative approaches to reducing Bitcoin's energy footprint have been proposed. One is shifting away from proof-of-work entirely (e.g., to Proof-of-Stake), which involves a very different security model beyond the scope of this paper. Another approach is the "Proof-of-Delay" or sharded mining idea by Mirkin et al. (2024), which aims to reduce energy consumption by making block production partly depend on a time delay (so miners cannot use unlimited parallelization to increase success probability). That approach lowers energy use for a given security level but does not solve the issue of choosing the right security level. Our mechanism is complementary: it could be used in conjunction with any mining method (PoW or hybrid) to target an optimal security level. In summary, our work adds to the literature by providing a concrete mechanism design solution that directly addresses the joint security-environmental trade-off in PoW networks, with rigorous theoretical grounding and practical viability.

VI. Conclusion

We have presented a mechanism design framework for managing hashrate in Proof-of-Work cryptocurrencies, treating the protocol as a designer that can set incentives to internalize externalities. By adjusting block rewards in response to total mining effort, the mechanism achieves a socially optimal balance between network security and environmental sustainability. Our theoretical results show that such a mechanism can implement the optimal hashrate in equilibrium and remains stable under dynamic adjustments. The Targeted Nakamoto proposal exemplifies how these ideas can be implemented in practice for Bitcoin, and potentially for other PoW-based systems facing similar challenges.

The implications of this work are significant for the long-term evolution of cryptocurrencies. As block subsidies diminish over time, there is growing concern that transaction fees alone may not sustain adequate security (Carlsten et al., 2016). Rather than passively hoping that fees or price increase to compensate, our approach offers a proactive policy tool: the community or protocol governance can decide on a target security level (hashrate) and let the protocol adjust rewards to maintain it. This could ensure a minimum security threshold even in the face of adverse economic conditions. Conversely, as society grapples with climate change, there may be pressure to reduce Bitcoin's carbon footprint. Our mechanism provides a way to cap the energy usage of mining without fundamentally altering the decentralized nature of the system or requiring bans or external regulation; it simply uses economic incentives to guide the network to a collectively desirable outcome.

There are many avenues for future research. One is to empirically estimate the optimal hashrate range $[H_{\text{LB}}, H_{\text{UB}}]$ for Bitcoin, which requires quantifying the marginal security benefit and marginal environmental cost of hashpower. Another avenue is exploring more sophisticated adjustment rules (e.g., using control theory or machine learning to fine-tune reward adjustments) and analyzing their convergence properties. It would also be worthwhile to study the governance aspects: how the community could agree on a target and update it over time, and how to implement the mechanism in a backwards-compatible way. Finally, extending the mechanism design perspective to other blockchain externalities (such as congestion or miner extractable value issues) could yield further improvements to protocol efficiency. We hope this work stimulates a fruitful dialogue between economists and protocol designers on marrying economic optimality with decentralized consensus.

References

ARONOFF, D. (2024): "Targeted Nakamoto: A Bitcoin Protocol to Balance Network Security and Carbon Emissions," ArXiv:2405.15089 (submitted May 23, 2024; last revised Aug 21, 2025).

BUDISH, E. (2018): "The economic limits of Bitcoin and the blockchain," Tech. Rep. 24717, National Bureau of Economic Research.

CAMBRIDGE CENTRE FOR ALTERNATIVE FINANCE (2024): "Cambridge Bitcoin Electricity Consumption Index (CBECI)," `https://ccaf.io/cbnsi/cbeci`, cambridge Judge Business School, University of Cambridge. Accessed 2024-12-04.

CARLSTEN, M., H. KALODNER, A. NARAYANAN, AND S. M. WEINBERG (2016): "On the instability of Bitcoin without the block reward," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS), 154–167.

CHEN, L., L. W. CONG, AND Y. XIAO (2020): "A Brief Introduction to Blockchain Economics," in World Scientific Book Chapter, World Scientific, 1–40.

CONG, L. W., Z. HE, AND J. LI (2021): "Decentralized mining in centralized pools," Review of Financial Studies, 34, 1191–1235.

DECKER, C. AND R. WATTENHOFER (2013): "Information Propagation in the Bitcoin Network," in IEEE P2P 2013 Proceedings, 1–10.

FOWLIE, M., M. REGUANT, AND S. P. RYAN (2016): "Market-based emissions regulation and industry dynamics," Journal of Political Economy, 124.

GALLERSDORFER, U., L. KLASSEN, AND C. STOLL (2020): "Energy consumption of cryptocurrencies beyond Bitcoin," Joule, 4, 1839–1851.

GERVAIS, A., V. GLYKANTZIS, G. O. KARAME, K. W. RITZDORF, AND S. CAPKUN (2016): "On the security and performance of proof-of-work blockchains," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS), 3–16.

HUBERMAN, G., J. D. LESHNO, AND C. MOALLEMI (2021): "Monopoly without a monopolist: An economic analysis of the Bitcoin payment system," Review of Economic Studies, 88, 3011–3040.

KROLL, J. A., I. C. DAVEY, AND E. W. FELTEN (2013): "The economics of Bitcoin mining, or Bitcoin in the presence of adversaries," in Proceedings of the 12th Workshop on the Economics of Information Security (WEIS).

LESHNO, J. D. AND P. STRACK (2020): "Bitcoin: An axiomatic approach and an impossibility theorem," American Economic Review: Insights, 2, 269–286.

MIRKIN, M., L. ZHOU, I. EYAL, AND F. ZHANG (2024): “Sprints: Intermittent blockchain PoW mining,” in Proceedings of the 33rd USENIX Security Symposium, Philadelphia, PA, 6273–6289.

NAKAMOTO, S. (2008): “Bitcoin: A Peer-to-Peer Electronic Cash System,” White paper (accessible at https://bitcoin.org/bitcoin.pdf).

PAGNOTTA, E. S. (2021): “Decentralizing Money: Bitcoin Prices and Blockchain Security,” The Review of Financial Studies, 35, 866–907.

PIGOU, A. C. (1920): The Economics of Welfare, London: Macmillan.

ROSENFELD, M. (2014): “Analysis of Hashrate-Based Double-Spending,” arXiv preprint arXiv:1402.2009.

SEDLMEIR, J., H. U. BUHL, G. FRIDGEN, AND R. KELLER (2020): “The energy consumption of blockchain technology: Beyond myth,” Business & Information Systems Engineering, 62, 599–608.

STOLL, C., L. KLAASSEN, AND U. GALLERSDÖRFER (2019): “The Carbon Footprint of Bitcoin,” Joule, 3, 1647–1661.

*

Appendix: Proofs

Proof of Proposition 1

PROOF:

Consider the threshold policy defined by (6), and suppose the target H^* is announced. First, note that if $H = H^*$ in some epoch, then R remains at R^* (which by design yields zero profit for miners). Thus, no miner has an incentive to deviate by increasing or decreasing hashrate: increasing would incur a cost without additional expected reward (since marginal profit is zero at optimum), and decreasing would forgo reward without saving cost (because they were at zero profit already). Therefore, H^* is a Nash equilibrium at reward R^*.

Now consider $H > H^*$. Under the mechanism, the next period reward will be scaled down (roughly to $R^- = (1 - \delta)R$ in the discrete version). Given the current H was only sustained by reward R, a lower reward means miners are operating at negative profit (since their cost was equal to R/H per unit hash at equilibrium, and now reward per hash is lower). In a Nash equilibrium, miners would anticipate this and some will drop out before the next period, i.e. H cannot remain above H^*. Formally, if H were to persist above H^*, each miner's payoff would be negative (because $C'(h_i) > R/H$ would imply $C(h_i) > (R/H)h_i$ for small h_i by convexity, leading to $u_i < 0$). Thus no miner would choose a positive h_i, contradicting $H > 0$. Hence no equilibrium with $H > H^*$ exists.

A symmetric argument holds for $H < H^*$. If H is below target, the mechanism increases R to $R^+ = (1+\delta)R$. Then at the current hashrate, miners would enjoy positive profits (their cost is $C'(h_i) < R/H$, so adding a small amount of hashrate yields $u_i > 0$). Thus some miner can profitably increase h_i, implying H cannot stay below H^*. There will be entry or expansion until H rises.

Therefore, the only fixed point of this adjustment process is H^*. The argument above essentially uses a best-response dynamic, but it also shows that any putative Nash equilibrium must have $H = H^*$. This completes the proof.

Proof of Proposition 2

PROOF:

We model the dynamic as follows. Let $f(H)$ be the aggregate best-response of miners when the reward is set such that last period's hashrate was H. That is, $f(H)$ is the total H' chosen in response to the reward policy triggered by H. Under our threshold rule, if $H < H^*$, the reward is increased by factor $(1+\delta)$, so miners will supply a new hashrate $H' > H$ in response (since higher reward shifts the zero-profit condition to $H' \approx (1+\delta)H$ in the linear-cost case). If $H > H^*$, reward is cut to $(1-\delta)R$, leading to $H' < H$ (approximately $H' = (1-\delta)H$). If $H = H^*$, reward is unchanged and $H' = H^*$. Thus, $f(H)$ pushes H toward H^* from either side:

$$f(H) - H^* \approx \begin{cases} (1+\delta)H - H^* > 0 & \text{if } H < H^*, \\ (1-\delta)H - H^* < 0 & \text{if } H > H^*, \end{cases}$$

with $f(H^*) = H^*$. For δ small, this mapping is a contraction around H^*, ensuring convergence by the Banach fixed-point theorem. Even if δ is moderate, one can show that $|H_{t+1} - H^*| < |H_t - H^*|$ whenever $H_t \neq H^*$ (under reasonable assumptions like monotonic best-responses). Therefore H_t approaches H^* over time.

In summary, hashrate either converges exactly to H^* or enters a small neighborhood and possibly oscillates within it if the policy overshoots; by refining the adjustment granularity, the oscillation can be made arbitrarily small. Thus, in the long run the system achieves the target level.

Proof of Proposition 3

PROOF:

Part (i): Equilibrium. The Nash equilibrium analysis in Proposition III.B is static and depends only on the reward level R that prevails at the target hashrate H^* in an epoch (recall that miners are myopic and re-set hashrate each epoch). In the mechanism, the threshold that triggers upward versus downward reward adjustments is the target H^*, not the estimator $\widehat{H}_t$ itself. At $H = H^*$ the designer chooses the reward R^* such that miners' marginal profit is zero and aggregate hashrate is H^*.[6] The presence of estimation noise ε_t affects the dynamics of how R_t is updated when $H_t \neq H^*$, but it does not alter the definition of R^* nor miners' payoff function at H^*. Hence, the same zero-profit and best-response conditions continue to hold at H^*, and no miner can profitably deviate given others choose hashrate consistent with H^*. This establishes that H^* remains a Nash equilibrium of the mining game under noisy observation.

Part (ii): Convergence to an $O(\bar{\varepsilon})$ neighborhood. Write the estimator as $\widehat{H}_t = H_t + \varepsilon_t$ with $|\varepsilon_t| \leq \bar{\varepsilon}$ and $\mathbb{E}[\varepsilon_t] = 0$. Consider first the region

$$H_t \;\geq\; H^* + \bar{\varepsilon}.$$

For any such H_t and any admissible ε_t we have $\widehat{H}_t = H_t + \varepsilon_t \geq H^*$, so the sign of $\widehat{H}_t - H^*$ is correctly classified as "too high." The mechanism therefore applies the same "downward" reward adjustment as in the noise-free case. Similarly, if $H_t \leq H^* - \bar{\varepsilon}$, then $\widehat{H}_t \leq H^*$ for all admissible ε_t, so the state is correctly classified as "too low" and the reward is adjusted upward exactly as in the noise-free case. Thus outside the band

$$B \;\equiv\; [H^* - \bar{\varepsilon},\, H^* + \bar{\varepsilon}],$$

the noisy mechanism induces the same direction of drift for H_t as the noiseless mechanism.

Let $g(R)$ denote the aggregate best-response hashrate when the reward is R, and suppose g is differentiable with Lipschitz constant $k > 0$ around R^*, i.e.,

$$|g(R') - g(R)| \;\leq\; k\,|R' - R| \quad \text{for } R,\, R' \text{ near } R^*.$$

Under the threshold rule, a correct classification outside B implies that in one epoch the reward moves by at

[6] Formally, R^* is pinned down by the zero-profit condition $C'(h^*) = R^*/H^*$ for a symmetric equilibrium h^* with $H^* = \sum_i h_i^*$.

least δ in the appropriate direction and therefore the hashrate moves by at least $k\,\delta$ in the direction of H^*:

$$|H_{t+1} - H^*| \;\le\; |H_t - H^*| - k\,\delta \qquad \text{whenever} \quad |H_t - H^*| > \bar{\varepsilon}.$$

Hence, starting from any H_0, the sequence $\{H_t\}$ reaches the band B in at most $\lceil |H_0 - H^*|/(k\delta) \rceil$ steps.

Inside the band B, estimation noise can in principle flip the sign of $\widehat{H}_t - H^*$ and occasionally trigger an "incorrect" adjustment. Let $f(H, \widehat{H})$ denote the aggregate best-response mapping induced by the reward-update rule, and suppose that for H in a neighborhood of H^*, f is Lipschitz in H with constant $L \in (0,1)$ and Lipschitz in $\widehat{H}$ with constant $K > 0$. Then for all sufficiently large t,

$$\text{(A1)} \quad |H_{t+1} - H^*| \;=\; \left|f(H_t, \widehat{H}_t) - f(H^*, H^*)\right| \;\le\; L\,|H_t - H^*| + K\,|\varepsilon_t| \;\le\; L\,|H_t - H^*| + K\,\bar{\varepsilon}.$$

Iterating inequality (A1) yields, for any $t \ge 1$,

$$\text{(A2)} \qquad |H_t - H^*| \;\le\; L^t |H_0 - H^*| \;+\; K\,\bar{\varepsilon} \sum_{j=0}^{t-1} L^j \;=\; L^t |H_0 - H^*| \;+\; K\,\bar{\varepsilon}\,\frac{1 - L^t}{1 - L}.$$

Taking the limit superior as $t \to \infty$ and using $L \in (0,1)$, we obtain the explicit bound

$$\text{(A3)} \qquad \limsup_{t \to \infty} |H_t - H^*| \;\le\; \frac{K}{1 - L}\,\bar{\varepsilon}.$$

Thus the asymptotic deviation of realized hashrate from the target H^* is bounded above by a constant multiple of the maximal estimation error $\bar{\varepsilon}$, which establishes part (ii) and hence the theorem.

Proof of Lemma 1

PROOF:

Part (i). Let $\widehat{H}_t = H_t + \varepsilon_t$, where $|\varepsilon_t| \le \bar{\varepsilon}$ for all t. By the Lipschitz property of the best-response mapping in the first argument, there exists $L \in (0,1)$ such that for all H in a neighborhood of H^*,

$$|f(H, \widehat{H}_t) - f(H^*, \widehat{H}_t)| \;\le\; L\,|H - H^*|.$$

Moreover, since f is Lipschitz in its second argument with constant $K > 0$, we have

$$|f(H^*, \widehat{H}_t) - f(H^*, H^*)| \;\leq\; K\,|\varepsilon_t| \;\leq\; K\,\bar{\varepsilon}.$$

Combining these two inequalities and using $f(H^*, H^*) = H^*$ yields the one-step bound

$$|H_{t+1} - H^*| \;\leq\; L\,|H_t - H^*| + K\,\bar{\varepsilon}. \tag{A4}$$

Iterating (A4) gives

$$|H_t - H^*| \;\leq\; L^t|H_0 - H^*| \;+\; K\,\bar{\varepsilon}\sum_{j=0}^{t-1} L^j \;=\; L^t|H_0 - H^*| \;+\; K\,\bar{\varepsilon}\,\frac{1 - L^t}{1 - L}.$$

Since $L \in (0, 1)$, $\lim_{t\to\infty} L^t = 0$, and therefore

$$\limsup_{t\to\infty} |H_t - H^*| \;\leq\; \frac{K}{1 - L}\,\bar{\varepsilon}.$$

This proves part (i).

Part (ii). Suppose $\mathbb{E}[\varepsilon_t^2] = \sigma^2/m$. Taking expectations in (A4) and using Jensen's inequality gives

$$\mathbb{E}[|H_{t+1} - H^*|] \;\leq\; L\,\mathbb{E}[|H_t - H^*|] \;+\; K\,\mathbb{E}[|\varepsilon_t|] \;\leq\; L\,\mathbb{E}[|H_t - H^*|] \;+\; K\,\frac{\sigma}{\sqrt{m}},$$

where the last inequality uses $\mathbb{E}[|\varepsilon_t|] \leq \sqrt{\mathbb{E}[\varepsilon_t^2]} = \sigma/\sqrt{m}$. Iterating yields

$$\mathbb{E}[|H_t - H^*|] \;\leq\; L^t|H_0 - H^*| \;+\; K\,\frac{\sigma}{\sqrt{m}}\sum_{j=0}^{t-1} L^j \;=\; L^t|H_0 - H^*| \;+\; \frac{K}{1 - L}\,\frac{\sigma}{\sqrt{m}}\,(1 - L^t).$$

Taking $\limsup_{t\to\infty}$ gives

$$\limsup_{t\to\infty} \mathbb{E}[|H_t - H^*|] \;\leq\; \frac{K}{1 - L}\,\frac{\sigma}{\sqrt{m}}.$$

This proves part (ii).

Numerical Example of Hashrate Adjustment

This appendix provides a simple numerical example of the dynamic behavior of the reward-adjustment mechanism. The exercise is not intended as a calibrated model of Bitcoin mining, but rather as a concrete illustration of how hashrate can converge to the target H^* under the mechanism.

Setup

We consider a discrete-time version of the model with linear costs and static best responses. Let the unit cost of hashing be $c = 1$, and normalize the baseline reward to $R_0 = 1$. The target hashrate is $H^* = 1$. The reward is updated according to

$$R_{t+1} = \begin{cases} R_t(1+\delta) & \text{if } \widehat{H}_t < H^*, \\ R_t(1-\delta) & \text{if } \widehat{H}_t > H^*, \\ R_t & \text{if } \widehat{H}_t = H^*, \end{cases}$$

with adjustment step $\delta = 0.08$. Miners myopically best-respond each period, yielding the static best-response hashrate

$$H_{t+1} = \frac{R_{t+1}}{c}.$$

We initialize the system far from the target at $H_0 = 2.2$ and iterate this recursion for $T = 100$ epochs. For simplicity, the numerical example sets $\widehat{H}_t = H_t$, i.e., it abstracts from estimation noise and focuses on the deterministic dynamics induced by the mechanism.

Result

Figure B1 plots the resulting path $\{H_t\}_{t=0}^{T}$ together with the target level H^*. The example exhibits a rapid initial contraction toward the target region followed by damped oscillations around H^* as the reward adjustments become smaller in absolute terms. The realized hashrate remains close to the target after a short transient phase, consistent with the convergence analysis in Proposition 2 and Theorem 1.

This numerical example illustrates the negative-feedback nature of the mechanism: reward cuts when $H_t > H^*$ and reward increases when $H_t < H^*$ jointly stabilize the decentralized mining game near the socially optimal hashrate.

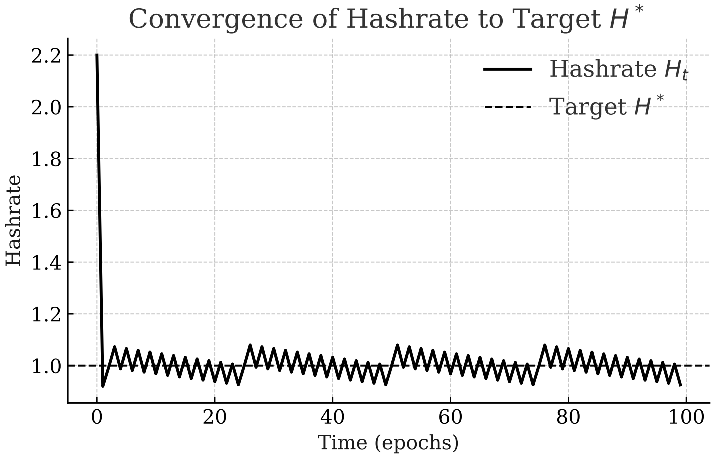

Figure B1. : Numerical example of convergence of hashrate H_t to the target H^* under the reward-adjustment mechanism with $c = 1$, $R_0 = 1$, $H^* = 1$, $\delta = 0.08$, and $H_0 = 2.2$.