Deep Bribe: Predicting the Rise of Bribery in Blockchain Mining with Deep RL

Roi Bar-Zur* *Technion, IC3* *roi.bar-zur@campus.technion.ac.il*
Danielle Dori* *Technion*
Sharon Vardi* *Technion*
Ittay Eyal *Technion, IC3*
Aviv Tamar *Technion*

***Abstract*—Blockchain security relies on incentives to ensure participants, called *miners*, cooperate and behave as the protocol dictates. Such protocols have a *security threshold* – a miner whose relative computational power is larger than the threshold can deviate to improve her revenue. Moreover, blockchain participants can behave in a *petty compliant* manner: usually follow the protocol, but deviate to increase revenue when deviation cannot be distinguished externally from the prescribed behavior. The effect of petty compliant miners on the security threshold of blockchains is not well understood. Due to the complexity of the analysis, it remained an open question since Carlsten et al. identified it in 2016.**

In this work, we use *deep Reinforcement Learning* (RL) to analyze how a rational miner performs *selfish mining* by deviating from the protocol to maximize revenue when petty compliant miners are present. We find that a selfish miner can exploit petty compliant miners to increase her revenue by bribing them. Our method reveals that the security threshold is lower when petty compliant miners are present. In particular, with parameters estimated from the Bitcoin blockchain, we find the threshold drops from the known value of 25% to only 21% (or 19%) when 50% (or 75%) of the other miners are petty compliant. Hence, our deep RL analysis puts the open question to rest; the presence of petty compliant miners exacerbates a blockchain's vulnerability to selfish mining and is a major security threat.

***Index Terms*—Blockchain, Selfish Mining, Bitcoin, Petty Compliant, Transaction Fees, Deep Reinforcement Learning**

1. Introduction

Cryptocurrencies such as Bitcoin [1] continue to gain traction, reaching a market capitalization of over 1 trillion US dollars [2] and attracting the interest of financial organizations (e.g., [3]). Their popularity has been accompanied by a rise in interest in their underlying technology, *blockchain* [4]–[8]. Bitcoin and other prominent blockchains [2] use *Proof of Work* (PoW): They are maintained by *miners*, who extend the blockchain by expending computational power to *mine* new *blocks*, which record *transactions* gathered from system users.

*. Equal contribution.

The security of blockchains critically depends on incentives to motivate miners to participate and behave according to the protocol. These incentives are provided by two types of *block rewards*. The first type, called *subsidy*, is newly minted tokens miners receive for each block they mine. In addition, miners receive *transaction fees* from users for including their posted transactions in a block.

If everyone behaves as the protocol dictates, all rewards should be distributed fairly among miners based on their *size*, the relative amount of computational power they contribute. However, the equilibrium only holds if all miners are smaller than some *security threshold* [9]–[11]. A miner larger than the threshold can engage in *selfish mining*, deviating from the protocol and gaining more revenue than her fair share. Such deviation will disrupt the equilibrium and can lead to unpredictable changes in miner behavior, making the blockchain vulnerable to other attacks [12].

Carlsten et al. [13] identified that although selfish mining might not be profitable for small miners and following the protocol is their best response when considering subsidy only, if fees are available then the situation is different. There are cases when even small miners are better off deviating in a small way from the protocol to take advantage of available fees, seemingly without deviating from the protocol. Miners exploiting such opportunities are called *petty compliant*.

Carlsten et al. [13] also show a selfish miner can *undercut* an existing block by creating a conflicting block and intentionally leaving transactions out of it. This allows the miner of the subsequent block to include the leftover transactions and earn the transaction fees. The leftover fees are a bribe to petty-compliant miners who can increase their revenue by favoring the block created by the selfish miner.

Since petty compliant mining has been introduced over 6 years ago [13], it remained an open question how much revenue can be gained by a selfish miner performing undercutting. This question is important as it leads to a better understanding of the effect of petty compliant miners on the security threshold. But to answer the question one would need a model that can analyze the behavior of miners in a complex blockchain environment. The model would have to allow arbitrary miner strategies and take into account the effect of transaction fees. Even if one could design such a model, solving it would be a challenging task due to its complexity. And thus, to the best of our knowledge, the

question remained open.

In this work, we answer this question by utilizing *deep Reinforcement Learning* (RL) [14]. We focus on *Nakamoto blockchains* [11], Bitcoin and similar blockchains that are based on it, such as Bitcoin Cash, Litecoin and Zcash. We model the mining process as a *Markov Decision Process* (MDP) [14]–[16] from the viewpoint of a rational miner trying to maximize revenue. Modeling selfish mining in an MDP allows for a wide range of strategies, in contrast to Carlsten et al. [13] where only a particular set of strategies was considered. As modeling transaction fees requires a complex model, our MDP contains over 100 million states. We thus cannot solve the MDP exactly, and instead use deep RL to approximate the optimal policy. We utilize the state-of-the-art deep RL framework *WeRLman* [11] suited specifically for complex selfish mining models.

Furthermore, unlike Carlsten et al. [13], we do not assume all other miners are petty compliant. Instead, we assume some fraction β are petty compliant and the rest are *honest*, always following the prescribed protocol. We find that a selfish miner can benefit from undercutting even if only a fraction of non-rational miners are petty compliant. In some cases the advantage can be over 10%. But more importantly – the increase in revenue of a selfish miner results in a decrease in the security threshold of the blockchain.

We instantiate our model with parameters estimated from the Bitcoin blockchain. Assuming that half of the miners are petty compliant, we show a decrease of the state-of-the-art security threshold from 0.25 [11] to 0.21. In one decade from the time of writing, the estimated security threshold will decrease from 0.21 to 0.17. If we assume that 75% of the non-rational miners are petty compliant, the security threshold estimations decrease to 0.19, at the time of writing, and to 0.13, in one decade. This is well below common miner sizes [17].

After reviewing related work (§2) and providing background on blockchain protocols, on selfish mining and on petty-compliant mining (§3), we present our main contributions:

1) a model of selfish mining in a Nakamoto blockchain with petty-compliant miners (§4),
2) a deep RL based analysis of the behavior of a rational miner when other petty-compliant miners are present (§5), and
3) an analysis of the impact of petty-compliant miners on the revenue of selfish mining and the security threshold of Bitcoin (§6).

We conclude with a discussion of the results and of future work (§7).

2. Related Work

Petty Compliant Mining. The concept of petty compliant mining was introduced in 2016 by Carlsten et al. [13]. Carlsten et al. [13] analyze a specific set of selfish mining strategies, which are always more profitable than the honest strategy, resulting in a security failure of the protocol. However, their model assumes that the subsidy is negligible compared to transaction fees and that all miners apart from the selfish miner are petty compliant. These simplifications result in a simpler model, which is easier to analyze.

In contrast, our model allows the analysis of realistic scenarios, tuning both the fraction of petty-compliant miners and the ratio between subsidy and transaction fees. In addition, our model allows the consideration of arbitrary strategies, not limited to a particular set. These differences enable us to analyze the effect of petty compliant miners on the security threshold under a wide range of conditions. Furthermore, we use deep RL to analyze our model, allowing us to find approximately optimal strategies.

Selfish Mining. Selfish mining is a well-studied problem [9]–[12], [18]–[22] and many solutions have been proposed to mitigate it [23]–[25]. Earlier work focused on specific selfish mining strategies and their impact on the security of the blockchain [12], [18], [19]. Later work provided a more comprehensive analysis of the problem by modeling the mining process as an MDP and solving it [9], [10], [20], [21]. This method is constrained by the computational complexity of solving the MDP, and thus cannot be used to analyze complex selfish mining strategies. To overcome this limitation, two deep RL based approaches have been proposed: SquirRL [22] and WeRLman [11].

To the best of our knowledge we are the first to analyze the effect of petty compliant miners on the security of the blockchain besides Carlsten et al. [13]. Due to the complexity of our model it cannot be solved directly, calling for analysis with deep RL. We utilize the method presented in WeRLman [11] using its open-source library.

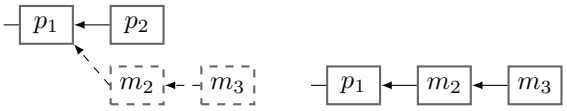
In addition, our model is influenced by the model presented in WeRLman [11] as it is the first to model selfish mining when considering both subsidy and transaction fees. However, WeRLman's model doesn't consider other miners can be petty compliant, and thus doesn't capture the undercutting strategy by allowing the selfish miner to bribe other miners.

3. Preliminaries

In this section we provide background on blockchains, selfish mining, petty compliant mining and how to analyze arbitrary selfish mining strategies. A reader familiar with these topics may skip to the next section (§4).

Blockchain. In this work, we focus on *Nakamoto blockchains*, Bitcoin [1] and similar protocols. A Nakamoto blockchain is maintained by a peer-to-peer network of nodes, called *miners*. Each miner has a copy of the entire ledger. In the ledger, transactions are batched into blocks and each block is linked to the previous block by including its cryptographic hash. This forms a chain of blocks, imaginatively called the *blockchain*.

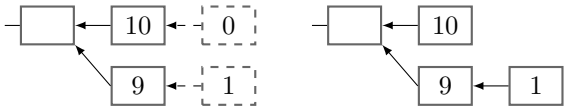
Miners maintain the blockchain through a process called *mining*. Mining involves solving complex cryptographic problems and compiling a list of unverified transactions, which are then added to a new block along with the



(a) The selfish miner creates an alternative secret chain (dashed).

(b) The miner publishes the secret chain altogether, causing other nodes to adopt it.

Figure 1. An illustration of selfish mining.



(a) Two chains of the same length where one chain leaves more fees for potential subsequent blocks (dashed).

(b) A petty compliant miner will favor the chain that leaves her more fees.

Figure 2. An illustration of undercutting.

solution. The solution is a proof that the miner has expended a significant amount of computational power to create the block. If a miner finds a valid solution, it can broadcast the block to the network and be rewarded with a reward, called *subsidy*. This process continues indefinitely, creating an ever-growing chain of blocks.

Nakamoto blockchains operate under the *longest chain rule*. The true version of the blockchain is the chain with the most blocks, and therefore, the most computational power behind it. This rule ensures that the blockchain remains secure and tamper-proof, as it is computationally hard for an attacker to change the state of the blockchain by creating a longer alternative chain. The reason for this is that the computational power required to add blocks to the chain is substantial, and it becomes increasingly difficult to do so as the chain grows longer.

Users pay a *transaction fee* in order to have their transactions included in a block. This fee serves as an incentive for miners to prioritize the transaction and add it to the block they are working on.

Selfish Mining. If each miner only controls less computational power than the *security threshold*, following the protocol is a *Nash equilibrium* – no participant can profit from deviating [9], [10]. Miners larger than the threshold, i.e., with enough computational power, are incentivized to deviate from the protocol and perform *selfish mining* [12]. The threshold characterizes the resistance of a blockchain to selfish mining. The higher the threshold, the more secure the blockchain is.

We illustrate a basic example of a selfish miner (Fig. 1). A miner performing selfish mining selectively chooses to not broadcast blocks it mines to the rest of the network, and instead adds these blocks to a secret chain only she knows about (Fig. 1a). If the miner's secret chain becomes longer than the current public chain, the miner can then broadcast its secret chain and cause other nodes to adopt it instead (Fig. 1b). This behavior allows the selfish miner to increase her revenue at the expense of other miners and lead other miners to deviate as well. This can lead to the deterioration of the blockchain's security.

Petty Compliant Mining. Carlsten et al. [13], show that when considering transaction fees, even miners smaller than the security threshold can benefit from deviating. More so, they can do it while seeming to others as if they were following the protocol. Miners engaging in such behavior are termed *petty compliant*.

In particular, in case of a tie in the longest chain rule, miners are prescribed to follow the first chain they heard of. As this tie-break depends on the local view of the miner, a petty compliant miner can choose which chain to adopt in such a case no matter her size. To increase revenue, a petty compliant miner will choose the chain that leaves the most transaction fees for subsequent miners.

This behavior opens the door for a selfish miner *undercutting* an existing chain by intentionally not including transactions. By doing so, the selfish miner bribes petty compliant miners to adopt her chain in case of a tie in the longest chain protocol.

We illustrate a simple example of undercutting (Fig. 2). A selfish miner can publish an alternative chain (below) equal in length to the public chain (above) (Fig. 2a). The miner can leave more transaction fees to the subsequent block (dashed) to encourage petty compliant miners to favor her chain (Fig. 2b).

Optimal Selfish Mining. To calculate the security threshold, an analysis of the optimal selfish mining strategy is required. To do so, previous work model the blockchain mining process as a Markov Decision Process (MDP) [14], [16] from the perspective of a rational miner controlling a fraction of α of the computational power [9]–[12], [26]. The rational miner mines on a secret chain she withholds from everyone else, while everyone else mines on a single public chain [9]–[12].

The rational miner can choose when to wait for more blocks to be mined, when to reveal blocks from her secret chain, and when to forfeit her secret chain and adopt blocks from the public chain instead [9]–[11]. If she reveals strictly more blocks than the public chain, everyone else will adopt her chain and mine on it.

In addition, when a non-rational miner mines a new block on the public chain, the rational miner can *rush* to reveal blocks from her secret chain before the new block propagates to some fraction γ of the network. If the rational miner had enough blocks prepared in advance, she can reveal a chain that is equal in length to the public chain. In this case, honest miners who receive the rational miner's chain first will adopt it and mine on it, while honest miners who receive the new block first will mine on the public chain. The fraction of the non-rational miners that will receive the rational miner's chain first is called the *rushing factor* and those miners are called *rushable*. This scenario results in an *active fork* If this occurs, the network will split:

a fraction of γ of the $1-\alpha$ non-rational miners will mine on the selfish miner's chain, and the rest will mine on the public chain until a newly mined block is published and resolves the fork.

The model presented in WeRLman [11], also incorporates transaction fees in the form of *whale transactions*, transaction offering exceptionally high fees worth F times the subsidy. The model simulates the arrival of whale transaction from an external Poisson Process. Whenever the longest chain is extended, a new whale transaction will arrive with probability δ.

Furthermore, WeRLman's approach can also encapsulate *Miner Extractable Value* (MEV) [27], an additional form of reward miners can extract from the blockchain by changing the order of transactions, or by adding transactions that exploit the blockchain's state. By treating MEV opportunities as whale transactions, the impact of MEV can also be captured.

4. Model

In this section, we explain our MDP model, and detail its objective function, state space, action space, and transitions. We omit the reasoning behind the elements of the model we adopt from previous work, and only explain the reasoning behind the new elements we introduce. For more details on previous models, we refer the reader to Bar-Zur et al. [11].

Core Model. As in previous work [7], [9]–[11], [20], [22], we model a single rational miner controlling a fraction of α of the computational power. The miner works on a secret chain while everyone else works on a single public chain. The miner can wait for more blocks, reveal blocks or choose to adopt blocks from the public chain.

Unlike previous work, we assume that out of the remaining $1-\alpha$ fraction of the computational power, a fraction of β miners are petty compliant, and the remaining fraction of $1-\beta$ miners are honest. Both the petty compliant and honest miners always mine on the longest known chain, and never mine on a secret chain.

In our model, petty compliant miners deviate from the protocol in how they break ties between chains of equal length. Honest miners, which follow the protocol, mine on the chain that they heard of first, while petty compliant miners mine on the chain that leaves them the most fees. Only in the case of a tie in the remaining fees too, they mine on the chain they heard of first.

Thus, except for the rational miner, a miner can be either honest or petty compliant, and can be either rushable or non-rushable, resulting in four types of non-rational miners. We assume that whether a miner belongs to the rushable or non-rushable group is independent of whether the miner is honest or petty compliant (Table 1). This is a major difference of our model compared to previous models, which consider only two types of non-rational miners: rushable honest miners and non-rushable honest miners.

TABLE 1. PARTITION OF MINERS IN THE MODEL.

Rational	α	
Petty	$\gamma\beta(1-\alpha)$	$(1-\gamma)\beta(1-\alpha)$
Honest	$\gamma(1-\beta)(1-\alpha)$	$(1-\gamma)(1-\beta)(1-\alpha)$
	Rushable	**Non-rushable**

Action Space. We model the rational miner's actions as three parameterized actions, as follows:

- `Adopt` x – Abandon the secret chain and adopt the first x blocks of the public chain.
- `Reveal` x – Reveal the first x blocks of the secret chain.
- `Mine` f – Keep mining on the secret chain. If f is 1, include a whale transaction in the new block, if a whale transaction is available. If f is 0, try to create an empty block, even if a whale transaction is available, leaving it to the miner of the next block.

This is similar to the model presented in WeRLman [11], which is based on the model presented by Sapirshtein et al. [9]. While the permitted actions are similar to WeRLman [11], the existence of petty compliant miner change the optimal strategy of the rational miner. In the model of WeRLman [11], undercutting would not be profitable as there were no petty compliant miners. However, in our new model, the optimal strategy may involve undercutting the public chain by performing `Mine` 0 to leave a transaction fee as a bribe to petty compliant miners, and then reveal her chain.

Furthermore, our model allows a more general form of undercutting compared to Carlsten et al. [13]. In their model, the rational miner can only undercut one public block, while our model permits undercutting the entire public chain which may be longer than a single block.

State Space. The state space is defined as a tuple of four elements: the secret chain $\underline{a}$, the public chain $\underline{h}$, the fork state *fork*, and the pool of available whale transactions *pool*. The secret chain and public chain are lists of boolean values indicating whether a block contains a whale transaction. Both chains are capped at length L. All actions resulting in a chain longer than L are illegal. The pool is the number of whale transactions that were accumulated in the network and have not been confirmed yet. The pool is also capped at L and all overflowing transactions are discarded.

The elements $\underline{a}$, $\underline{h}$, and *pool* are similar to those of WeRLman [11]. However, *fork* is different. We model the fork state to take one of 4 values.

- PRIORRATIONAL – The miner of the latest block mined is the rational miner.
- PRIORNON – The miner of the latest block mined is non-rational.
- ACTIVEPETTY – The network is experiencing an active fork after the latest block was mined by the rational miner.

- ACTIVERUSHING – The network is experiencing an active fork after the latest block was mined by a non-rational miner and the rational miner rushed to reveal blocks.

A simple calculation, reveals that the number of states is:

$$|\mathcal{S}| = \left(2^0 + 2^1 + ... + 2^L\right)^2 \cdot 4 \cdot (L+1) \approx 16 \cdot L \cdot 4^L \,. \quad (1)$$

This is about 33% larger than the state space of the model in the original WeRLman study [11].

Objective Function. Bitcoin and other similar blockchain protocols employ a mechanism for difficulty adjustment. The mechanism ensures that the longest chain grows at intervals of constant expected time. As in previous work, we model the rational miner's objective function as maximizing the expected revenue per unit of time [10], [12]. Due to the difficulty adjustment mechanism, the objective function takes the form of a ratio between accumulated rewards and the number of blocks in the longest chain, which we call *difficulty contributions*. Hence, each transition we later describe results in a reward and a difficulty contribution.

Transitions. We adopt some transitions from the model of WeRLman [11] and add new transitions related to the new fork state. We summarize all the transitions in the model in Appendix A.

5. Methods

As in previous work [11], we instantiate the model using $L = 10$. This results in over 10^8 states (comes from (1)), which is too large to solve using exact methods (e.g., [9], [10]). To overcome this we adopt the method of WeRLman [11], which was the first to successfully solve large-scale selfish mining models. In this section, we proceed to broadly describe the method. We refer the reader to [11] for a more detailed description.

MDP Transformation. The objective function in our model is a ratio between two cumulative sums of step rewards. Since this form is not standard, we use *Probabilistic Termination Optimization* (PTO) [10] to transform it into a linear reward MDP with a transition dependent discount factor. However, the transformed MDP only provides an approximation of the revenue in the original MDP. To receive a good approximation, we choose the *expected horizon* of PTO to be $10,000$, as in WeRLman [11].

Deep RL Algorithm. To solve the transformed MDP, we use *WeRLman* [11], a deep RL framework specialized for selfish mining analysis. WeRLman is based on *AlphaGo Zero* [28] a novel deep RL algorithm to that uses a combination of *Monte Carlo Tree Search* (MCTS) and *Deep Q-network* (DQN) [29], a deep learning extension of the Q-learning algorithm [14].

WeRLman novelty lies in the following three aspects. First, the Bellman operator [16] in WeRLman is based on the expectation of the next state's value, using the knowledge of transition probabilities, as opposed to only considering the next sampled state as standard RL algorithms do. Second, WeRLman interprets the value neural network as the difference from a baseline value of the current policy, rather than the value itself. The baseline is updated using a moving average of the revenue of the current policy. And third, WeRLman normalizes the target values used for training the value neural network to have zero mean.

Implementation Details. We fork the WeRLman repository [30] and implement the model described in Section 4 in Python. We run the code with the newly implemented model with the same hyperparameters used in WeRLman [11]. Each run we perform is on 64 CPU cores and takes about 12 hours. In all our runs, we use $L = 10$, $\gamma = 0.5$ and $\delta = 0.1$ and vary α, β and F.

6. Impact of Petty Compliant Miners

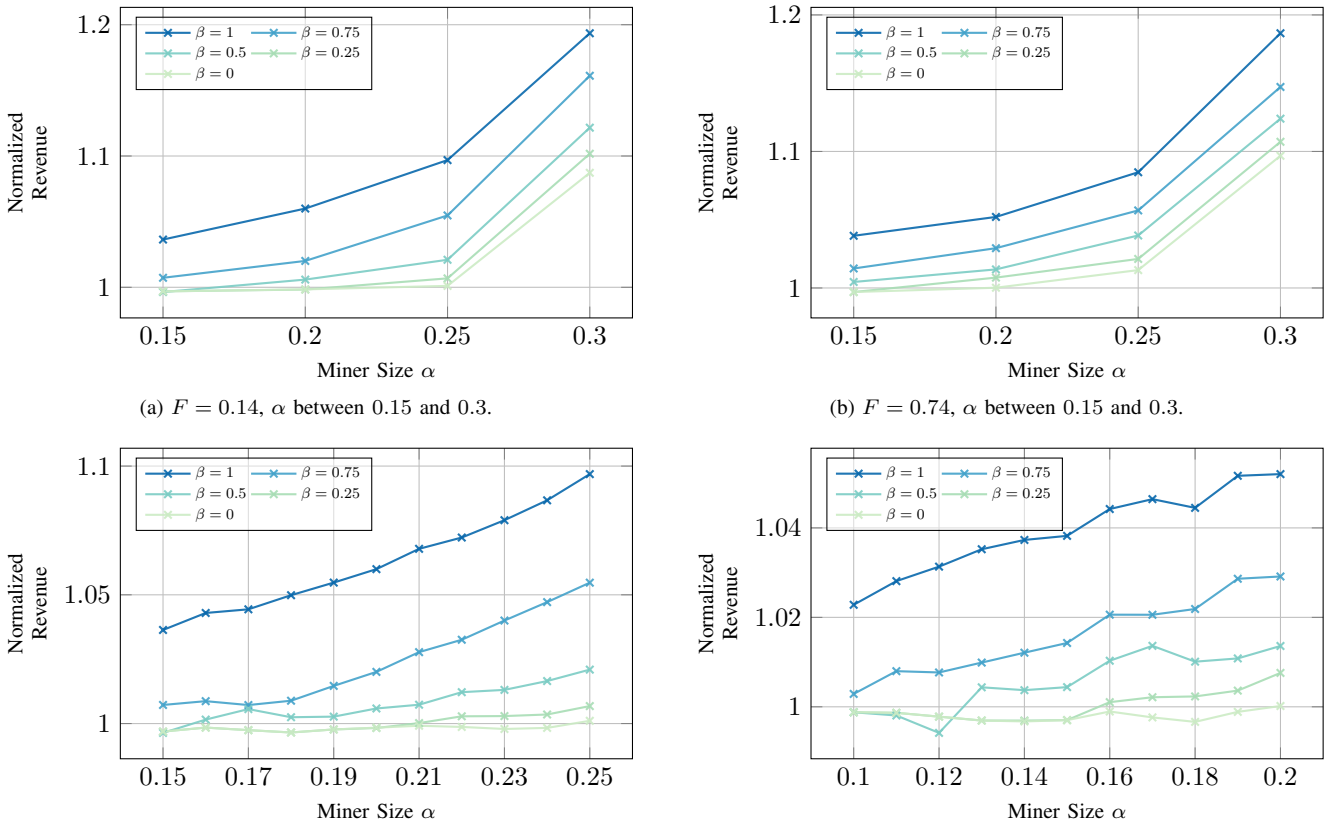
In this section we first highlight the impact of the fraction β of petty-compliant miners on the revenue of a selfish miner. We then characterize how larger values of β result in a reduction of the security threshold.

Impact on Revenue. We approximate the potential revenue of a rational miner with relative computational power α between 0.15 and 0.3 for various values of β, the fraction of petty compliant miners (Fig. 3). See Appendix B for tables with the exact values.

We consider the whale transaction fee to be $F = 0.14$ or $F = 0.74$, as these best estimate the transaction fees of Bitcoin at the time of writing and in one decade, respectively [11]. To ease interpretation, we plot the *normalized revenue*, the ratio between the revenue from selfish mining and the revenue of an honest miner with the same mining power. A ratio of more than 1, means that selfish mining is more profitable than honest mining in that scenario, and that the security threshold, the minimum size required for selfish mining, must be lower than miner's size in said scenario.

We observe that increasing β, the fraction of petty compliant miners results in an increase in the revenue of selfish mining in both cases. This is unsurprising, as the more petty compliant miners there are, the more successful undercutting is likely to be.

Furthermore, we find that the normalized revenue from selfish mining is higher when F is lower. This is an artifact of the model we use, where undercutting costs at least a transaction fee. Thus, bribing petty compliant miners becomes more expensive and less profitable when the transaction fee is higher. This is in contrast to the model by Carlsten et al. [13], where the amount used to bribe a petty compliant can be chosen more freely, so the selfish miner would retain some transaction fees.



(a) $F = 0.14$, α between 0.15 and 0.3.

(b) $F = 0.74$, α between 0.15 and 0.3.

(c) $F = 0.14$, α between 0.15 and 0.25.

(d) $F = 0.74$, α between 0.1 and 0.2.

Figure 3. The normalized revenue as a function of the miner size α.

Impact on Security Threshold. Since our deep RL based method can only approximate the optimal policy, we cannot directly compute the security threshold. We instead only compute an upper bound on the security threshold by finding the smallest α such that the revenue of selfish mining is better than honest mining. As we can only estimate the revenue policies we find using Monte Carlo simulations, we consider a policy better than honest mining only if it is better by a margin large enough to pass a Z-test with 99% confidence [11]. We detail the confidence intervals of our results in Appendix B.

For $F = 0.14$, matching the state of Bitcoin at the time of writing, we find that when $\beta \geq 0.75$, the security threshold drops from the state-of-the-art threshold of 0.25 [11] to below 0.19 (Fig. 3c). Taking $F = 0.74$, which represents an appropriate transaction fee in one decade, we find that when $\beta \geq 0.75$, the security threshold drops from the state-of-the-art threshold of 0.2 [11] to below 0.13 (Fig. 3d).

7. Conclusion

We are the first to fully characterize selfish mining when there are petty compliant miners. The complexity of the problem calls for a deep RL based approach, which was not possible prior to recent advances [11].

Our analysis shows that even if a fraction of the miners are petty compliant, a selfish miner can profit more by bribing them, resulting in a decrease of the security threshold. Since petty compliant mining dominates honest mining, this is a reasonable scenario considering that some miners are rational, and are willing to cheat to maximize revenue [26].

Our work highlights the importance of analyzing selfish mining under more realistic models and shows that deep RL can be used overcome model complexity. Our approach can be used for analyzing other blockchain protocols or potential mitigating mechanisms with the goal of preventing selfish mining from becoming a problem in the near future.

Acknowledgments

The research was supported by the Israel Science Foundation (1641/18), Ava Labs, the Technion Hiroshi Fujiwara Cyber Security Research Center and the Israel National Cyber Directorate. A. Tamar is funded by the European Union (ERC, Bayes-RL, Project No. 101041250). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008.

[2] coinmarketcap.com. (2023) Cryptocurrency market capitalization. [Online]. Available: https://coinmarketcap.com/

[3] B. Allison and CoinDesk, "JPMorgan on its crypto plans: 'the overall goal is to bring these trillions of dollars of assets into DeFi'," *fortune.com*, 12 June 2022.

[4] Y. Guo and C. Liang, "Blockchain application and outlook in the banking industry," *Financial innovation*, vol. 2, pp. 1–12, 2016.

[5] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—a systematic review," *PloS one*, vol. 11, no. 10, p. e0163477, 2016.

[6] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, pp. 183–187, 2017.

[7] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International journal of web and grid services*, vol. 14, no. 4, pp. 352–375, 2018.

[8] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117 134–117 151, 2019.

[9] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security: 20th International Conference, FC 2016, Christ Church, Barbados, February 22–26, 2016, Revised Selected Papers 20*. Springer, 2017, pp. 515–532.

[10] R. Bar-Zur, I. Eyal, and A. Tamar, "Efficient mdp analysis for selfish-mining in blockchains," in *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, 2020, pp. 113–131.

[11] R. Bar-Zur, A. Abu-Hanna, I. Eyal, and A. Tamar, "Werlman: To tackle whale (transactions), go deep (rl)," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2022, pp. 271–288.

[12] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Communications of the ACM*, vol. 61, no. 7, pp. 95–102, 2018.

[13] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 154–167.

[14] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[15] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[16] D. P. Bertsekas, "Dynamic programming and optimal control 4th edition, volume ii," *Athena Scientific*, 2015.

[17] blockchain.com. (2023) Hashrate distribution. [Online]. Available: https://www.blockchain.com/explorer/charts/pools

[18] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: Generalizing selfish mining and combining with an eclipse attack," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2016, pp. 305–320.

[19] C. Feng and J. Niu, "Selfish mining in ethereum," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1306–1316.

[20] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 3–16.

[21] R. Zhang and B. Preneel, "Lay down the common metrics: Evaluating proof-of-work consensus protocols' security," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 175–192.

[22] C. Hou, M. Zhou, Y. Ji, P. Daian, F. Tramer, G. Fanti, and A. Juels, "Squirrl: Automating attack analysis on blockchain incentive mechanisms with deep reinforcement learning," *arXiv preprint arXiv:1912.01798*, 2019.

[23] M. Saad, L. Njilla, C. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 360–364.

[24] R. Pass and E. Shi, "Fruitchains: A fair blockchain," in *Proceedings of the ACM symposium on principles of distributed computing*, 2017, pp. 315–324.

[25] I. Abraham, D. Dolev, I. Eyal, and J. Y. Halpern, "Colordag: An incentive-compatible blockchain," *Cryptology ePrint Archive*, 2022.

[26] A. Yaish, G. Stern, and A. Zohar, "Uncle maker:(time) stamping out the competition in ethereum," *Cryptology ePrint Archive*, 2022.

[27] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 910–927.

[28] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[30] R. Bar-Zur, A. Abu-Hanna, I. Eyal, and A. Tamar, "Werlman github repository," March 2022. [Online]. Available: https://github.com/roibarzur/pto-selfish-mining

Appendix

1. Model Transitions

We present all transitions in the model (Table 2). Transitions unique to our model are colored in light gray. Table 2 uses notation we now introduce.

- $T(\underline{u})$ – The number of transactions in the chain $\underline{u}$.
- $\underline{u}_i^j$ – The subchain of $\underline{u}$ from block j to block j.
- $\underline{u}||f$ – The chain $\underline{u}$ with a new block appended to it. If a transaction is available and $f = 1$, the new block includes a whale transaction.
- $\underline{u} \ll n$ – The chain $\underline{u}$ with the first n blocks removed.
- $|\underline{u}|$ – The length of the chain $\underline{u}$.
- $\underline{0}$ – The empty chain.
- $P_{+?}$ – If the longest chain extended, increment by 1 with probability δ.

2. Results Table

We detail all the results of the simulations in Table 3 and Table 4. The honest values are calculated using the formula: $(1+\delta F)\alpha$ [11]. We also provide the 99% confidence intervals for the estimated revenue of the policies we obtain and mark policies that are statistically significantly better than honest mining in light gray.

TABLE 2. TRANSITIONS IN THE MODEL WITH PETTY COMPLIANT BEHAVIOR.

State, Action		New State	Probability	Reward	Difficulty
$(\underline{a}, \underline{h}, \mathit{fork}, \mathit{pool})$,	`Adopt` x	$\left(\underline{0}, \underline{h} \ll x, \text{PRIORRATIONAL}, \mathit{pool} - T\left(\underline{h}_1^x\right)\right)$	1	0	0
$(\underline{a}, \underline{h}, \mathit{fork}, \mathit{pool})$, when $x > \lvert\underline{h}\rvert$	`Reveal` x	$\left(\underline{a} \ll x, \underline{0}, \mathit{fork}, \mathit{pool} - T\left(\underline{a}_1^x\right)\right)$	1	$x + T\left(\underline{a}_1^x\right) \cdot F$	x
$(\underline{a}, \underline{h}, \text{PRIORNON}, \mathit{pool})$, when $x = \lvert\underline{h}\rvert \leq \lvert\underline{a}\rvert$ and $T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right) < T\left(\underline{h}\right)$	`Reveal` x	$(\underline{a}, \underline{h}, \text{ACTIVEPETTY}, \mathit{pool})$	1	0	0
$(\underline{a}, \underline{h}, \text{PRIORRATIONAL}, \mathit{pool})$, when $x = \lvert\underline{h}\rvert \leq \lvert\underline{a}\rvert$	`Reveal` x	$(\underline{a}, \underline{h}, \text{ACTIVERUSHING}, \mathit{pool})$	1	0	0
$(\underline{a}, \underline{h}, \text{PRIORRATIONAL}, \mathit{pool})$,	`Mine` f	$\left(\underline{a}\lvert\lvert f, \underline{h}, \text{PRIORRATIONAL}, \mathit{pool}_{+?}\right)$	α	0	0
$(\underline{a}, \underline{h}, \text{PRIORNON}, \mathit{pool})$,	`Mine` f	$\left(\underline{a}\lvert\lvert 1, \text{PRIORNON}, \mathit{pool}_{+?}\right)$	$1 - \alpha$	0	0
$(\underline{a}, \underline{h}, \text{ACTIVERUSHING}, \mathit{pool})$, when $T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right) > T\left(\underline{h}\right)$	`Mine` f	$\left(\underline{a}\lvert\lvert f, \underline{h}, \text{ACTIVERUSHING}, \mathit{pool}_{+?}\right)$	α	0	0
		$\left(\underline{a} \ll \lvert\underline{h}\rvert, \underline{0}\lvert\lvert 1, \text{PRIORNON}, \left[\mathit{pool} - T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right)\right]_{+?}\right)$	$\gamma(1-\beta)(1-\alpha)$	$\lvert\underline{h}\rvert + T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right) \cdot F$	$\lvert\underline{h}\rvert$
		$\left(\underline{a}, \underline{h}\lvert\lvert 1, \text{PRIORNON}, \mathit{pool}_{+?}\right)$	$(1 - \gamma + \gamma\beta)(1-\alpha)$	0	0
$(\underline{a}, \underline{h}, \text{ACTIVERUSHING}, \mathit{pool})$, when $T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right) = T\left(\underline{h}\right)$	`Mine` f	$\left(\underline{a}\lvert\lvert f, \underline{h}, \text{ACTIVERUSHING}, \mathit{pool}_{+?}\right)$	α	0	0
		$\left(\underline{a} \ll \lvert\underline{h}\rvert, \underline{0}\lvert\lvert 1, \text{PRIORNON}, \left[\mathit{pool} - T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right)\right]_{+?}\right)$	$\gamma(1-\alpha)$	$\lvert\underline{h}\rvert + T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right) \cdot F$	$\lvert\underline{h}\rvert$
		$\left(\underline{a}, \underline{h}\lvert\lvert 1, \text{PRIORNON}, \mathit{pool}_{+?}\right)$	$(1-\gamma)(1-\alpha)$	0	0
$(\underline{a}, \underline{h}, \text{ACTIVERUSHING}, \mathit{pool})$, when $T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right) < T\left(\underline{h}\right)$	`Mine` f	$\left(\underline{a}\lvert\lvert f, \underline{h}, \text{ACTIVERUSHING}, \mathit{pool}_{+?}\right)$	α	0	0
		$\left(\underline{a} \ll \lvert\underline{h}\rvert, \underline{0}\lvert\lvert 1, \text{PRIORNON}, \left[\mathit{pool} - T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right)\right]_{+?}\right)$	$[\gamma(1-\beta)+\beta](1-\alpha)$	$\lvert\underline{h}\rvert + T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right) \cdot F$	$\lvert\underline{h}\rvert$
		$\left(\underline{a}, \underline{h}\lvert\lvert 1, \text{PRIORNON}, \mathit{pool}_{+?}\right)$	$(1-\gamma)(1-\beta)(1-\alpha)$	0	0
$(\underline{a}, \underline{h}, \text{ACTIVEPETTY}, \mathit{pool})$,	`Mine` f	$\left(\underline{a}\lvert\lvert f, \underline{h}, \text{ACTIVEPETTY}, \mathit{pool}_{+?}\right)$	α	0	0
		$\left(\underline{a} \ll \lvert\underline{h}\rvert, \underline{0}\lvert\lvert 1, \text{PRIORNON}, \left[\mathit{pool} - T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right)\right]_{+?}\right)$	$\beta(1-\alpha)$	$\lvert\underline{h}\rvert + T\left(\underline{a}_1^{\lvert\underline{h}\rvert}\right) \cdot F$	$\lvert\underline{h}\rvert$
		$\left(\underline{a}, \underline{h}\lvert\lvert 1, \text{PRIORNON}, \mathit{pool}_{+?}\right)$	$(1-\beta)(1-\alpha)$	0	0

TABLE 3. REVENUE FROM SELFISH MINING OBTAINED BY DEEP BRIBE FOR $F = 0.14$.

Miner Size α	Petty Compliant Fraction β	Fee F	Honest Mining Revenue	Selfish Mining Revenue	Confidence Radius
0.15	0.00	0.14	0.152 10	0.151 63	0.001 15
0.16	0.00	0.14	0.162 24	0.161 99	0.001 36
0.17	0.00	0.14	0.172 38	0.171 94	0.001 47
0.18	0.00	0.14	0.182 52	0.181 89	0.001 49
0.19	0.00	0.14	0.192 66	0.192 22	0.001 42
0.20	0.00	0.14	0.202 80	0.202 48	0.001 47
0.21	0.00	0.14	0.212 94	0.212 76	0.001 37
0.22	0.00	0.14	0.223 08	0.222 79	0.001 43
0.23	0.00	0.14	0.233 22	0.232 73	0.001 50
0.24	0.00	0.14	0.243 36	0.242 95	0.001 88
0.25	0.00	0.14	0.253 50	0.253 77	0.002 07
0.15	0.25	0.14	0.152 10	0.151 63	0.001 15
0.16	0.25	0.14	0.162 24	0.161 99	0.001 36
0.17	0.25	0.14	0.172 38	0.171 94	0.001 47
0.18	0.25	0.14	0.182 52	0.181 89	0.001 49
0.19	0.25	0.14	0.192 66	0.192 22	0.001 42
0.20	0.25	0.14	0.202 80	0.202 45	0.001 23
0.21	0.25	0.14	0.212 94	0.212 98	0.001 41
0.22	0.25	0.14	0.223 08	0.223 71	0.001 48
0.23	0.25	0.14	0.233 22	0.233 90	0.001 68
0.24	0.25	0.14	0.243 36	0.244 22	0.001 90
0.25	0.25	0.14	0.253 50	0.255 22	0.002 16
0.15	0.50	0.14	0.152 10	0.151 55	0.001 42
0.16	0.50	0.14	0.162 24	0.162 49	0.001 76
0.17	0.50	0.14	0.172 38	0.173 35	0.001 16
0.18	0.50	0.14	0.182 52	0.182 97	0.000 96
0.19	0.50	0.14	0.192 66	0.193 18	0.001 21
0.20	0.50	0.14	0.202 80	0.203 99	0.001 34
0.21	0.50	0.14	0.212 94	0.214 50	0.001 49
0.22	0.50	0.14	0.223 08	0.225 80	0.001 31
0.23	0.50	0.14	0.233 22	0.236 27	0.001 66
0.24	0.50	0.14	0.243 36	0.247 38	0.001 93
0.25	0.50	0.14	0.253 50	0.258 81	0.001 55
0.15	0.75	0.14	0.152 10	0.153 20	0.001 16
0.16	0.75	0.14	0.162 24	0.163 65	0.001 80
0.17	0.75	0.14	0.172 38	0.173 62	0.001 69
0.18	0.75	0.14	0.182 52	0.184 14	0.001 64
0.19	0.75	0.14	0.192 66	0.195 48	0.001 62
0.20	0.75	0.14	0.202 80	0.206 87	0.001 46
0.21	0.75	0.14	0.212 94	0.218 85	0.001 38
0.22	0.75	0.14	0.223 08	0.230 34	0.001 66
0.23	0.75	0.14	0.233 22	0.242 55	0.002 07
0.24	0.75	0.14	0.243 36	0.254 84	0.001 83
0.25	0.75	0.14	0.253 50	0.267 37	0.002 44
0.15	1.00	0.14	0.152 10	0.157 63	0.001 10
0.16	1.00	0.14	0.162 24	0.169 21	0.001 60
0.17	1.00	0.14	0.172 38	0.180 03	0.001 71
0.18	1.00	0.14	0.182 52	0.191 62	0.001 67
0.19	1.00	0.14	0.192 66	0.203 21	0.001 66
0.20	1.00	0.14	0.202 80	0.214 96	0.001 62
0.21	1.00	0.14	0.212 94	0.227 39	0.001 60
0.22	1.00	0.14	0.223 08	0.239 20	0.001 95
0.23	1.00	0.14	0.233 22	0.251 64	0.002 16
0.24	1.00	0.14	0.243 36	0.264 46	0.002 11
0.25	1.00	0.14	0.253 50	0.278 06	0.002 43

TABLE 4. REVENUE FROM SELFISH MINING OBTAINED BY DEEP BRIBE FOR $F = 0.74$.

Miner Size α	Petty Compliant Fraction β	Fee F	Honest Mining Revenue	Selfish Mining Revenue	Confidence Radius
0.10	0.00	0.74	0.107 40	0.107 27	0.000 82
0.11	0.00	0.74	0.118 14	0.117 98	0.000 93
0.12	0.00	0.74	0.128 88	0.128 60	0.001 16
0.13	0.00	0.74	0.139 62	0.139 19	0.001 13
0.14	0.00	0.74	0.150 36	0.149 87	0.001 22
0.15	0.00	0.74	0.161 10	0.160 62	0.001 34
0.16	0.00	0.74	0.171 84	0.171 66	0.001 58
0.17	0.00	0.74	0.182 58	0.182 15	0.001 71
0.18	0.00	0.74	0.193 32	0.192 67	0.001 72
0.19	0.00	0.74	0.204 06	0.203 83	0.001 71
0.20	0.00	0.74	0.214 80	0.214 84	0.001 28
0.10	0.25	0.74	0.107 40	0.107 27	0.000 82
0.11	0.25	0.74	0.118 14	0.117 98	0.000 93
0.12	0.25	0.74	0.128 88	0.128 60	0.001 16
0.13	0.25	0.74	0.139 62	0.139 19	0.001 13
0.14	0.25	0.74	0.150 36	0.149 90	0.001 25
0.15	0.25	0.74	0.161 10	0.160 62	0.001 34
0.16	0.25	0.74	0.171 84	0.172 02	0.001 80
0.17	0.25	0.74	0.182 58	0.182 97	0.001 30
0.18	0.25	0.74	0.193 32	0.193 77	0.001 49
0.19	0.25	0.74	0.204 06	0.204 80	0.001 43
0.20	0.25	0.74	0.214 80	0.216 43	0.001 47
0.10	0.50	0.74	0.107 40	0.107 27	0.000 82
0.11	0.50	0.74	0.118 14	0.117 91	0.001 20
0.12	0.50	0.74	0.128 88	0.128 13	0.001 00
0.13	0.50	0.74	0.139 62	0.140 23	0.000 69
0.14	0.50	0.74	0.150 36	0.150 92	0.001 54
0.15	0.50	0.74	0.161 10	0.161 81	0.001 49
0.16	0.50	0.74	0.171 84	0.173 61	0.002 01
0.17	0.50	0.74	0.182 58	0.185 07	0.001 30
0.18	0.50	0.74	0.193 32	0.195 27	0.001 32
0.19	0.50	0.74	0.204 06	0.206 27	0.001 44
0.20	0.50	0.74	0.214 80	0.217 72	0.001 52
0.10	0.75	0.74	0.107 40	0.107 71	0.001 06
0.11	0.75	0.74	0.118 14	0.119 08	0.001 58
0.12	0.75	0.74	0.128 88	0.129 87	0.001 41
0.13	0.75	0.74	0.139 62	0.141 00	0.001 22
0.14	0.75	0.74	0.150 36	0.152 18	0.001 55
0.15	0.75	0.74	0.161 10	0.163 40	0.001 59
0.16	0.75	0.74	0.171 84	0.175 38	0.001 87
0.17	0.75	0.74	0.182 58	0.186 34	0.001 91
0.18	0.75	0.74	0.193 32	0.197 55	0.001 71
0.19	0.75	0.74	0.204 06	0.209 90	0.001 46
0.20	0.75	0.74	0.214 80	0.221 06	0.001 63
0.10	1.00	0.74	0.107 40	0.109 85	0.000 71
0.11	1.00	0.74	0.118 14	0.121 46	0.000 97
0.12	1.00	0.74	0.128 88	0.132 92	0.001 27
0.13	1.00	0.74	0.139 62	0.144 54	0.001 27
0.14	1.00	0.74	0.150 36	0.155 97	0.001 58
0.15	1.00	0.74	0.161 10	0.167 26	0.001 46
0.16	1.00	0.74	0.171 84	0.179 44	0.002 26
0.17	1.00	0.74	0.182 58	0.191 06	0.002 06
0.18	1.00	0.74	0.193 32	0.201 92	0.001 92
0.19	1.00	0.74	0.204 06	0.214 60	0.001 40
0.20	1.00	0.74	0.214 80	0.225 98	0.001 74