Weight Copying in Bittensor (Working Paper)

Opentensor

May 29, 2024

Abstract

We consider the design and operation of decentralized and anonymous markets where money and query-responses are exchanged. When we have access to the true quality of these responses, we can easily penalize bad responses, but we may also want to penalize good responses that are obtained by copying other participants. We propose to do so via the commit-reveal approach, which prevents copying during a "commit reveal weights interval." Under appropriate conditions, this reduces the rewards to the copier so as to make copying unappealing. When the true quality of responses is not readily available, verifying the quality of responses is arguably more difficult, but still achievable through the peer-prediction approach.

1 Introduction

Bittensor incentivises decentralized actors to do honest and intelligent work. Miners provide utility to the ecosystem. For example, ML model inferences, compute, storage and many more freely defined by the subnet owners. Validators should validate miner's works and score their performance accordingly. Nominators select validators and delegate their voting power (stake) to these validators. The incentivized are currently calculated through the Yuma Consensus mechanism [Ope23].

We consider a problem that arises from the fact that validation is costly, and reporting the outcome of the validation may not coincide with the best interests of the validator. Despite our best intentions, we have observed a free-rider problem, where some validators do not do the work of evaluating miners, but copy on the work of others and receive equal (if not higher[1]) credit. When left unchecked, free-riders have led to the demise of previous decentralized systems [HCW05], but countermeasures do exist [JA05, FC05].

Section 2 models the interactions between validators as a repeated game with private signals at each round. Section 3 defines copying strategies, highlights

[1]The free-rider can even get greater credit than honest validators by copying recent consensus scores (or weights for short).

the ambiguity in the definition, and presents approaches to circumvent this ambiguity. If the ground-truth (miner performance) is available, we propose the commit-reveal approach (Section 4); else, we propose to adopt peer-prediction approaches (Section 6). In Section 5, we simulate the effect of implementing the commit-reveal solution on actual Bittensor subnets. At last, we situate our work relative to related works (Section 7) and outline open problems (Section 8).

2 Model

The problem of weight copying arises in the interaction between validators. The simplest model that describes this interaction is a repeated game between a set of validators. In such a setting, weight copying is simply the tit-for-tat strategy [RC65] where one participant j picks another participant k, and at every round t plays the same action that k played at the previous round $t-1$. This tit-for-tat strategy has the nice property of creating consensus and improving the long-term outcome for all players. In Bittensor, however, we do not simply want consensus: we also want fair weights for miners. Our model must therefore include miners.

We consider the following model of a repeated game with additional signals from miners. The set of players is the set of validators $\mathcal{Z}$. At every round $t = 1, 2, \ldots$, each validator j observes a signal $\mathbf{v}_t^j \in \mathcal{X} = \mathbb{R}^d$, where each of the d elements corresponds to a miner. We assume that the signals[2] $\mathbf{v}_t^j$ are independent and identically distributed for every validator j and every round t. The distribution of $\mathbf{v}_t^j$ is however unknown. Each validator j also has a stake value S_j associated to it and stays fixed over time, which is publicly known. Without loss of generality, we assume that $\sum_{j \in \mathcal{Z}} S_j = 1$.

At round t, each validator j takes a pair of actions

1. $a_t^j \in \{0, 1\}$, representing whether the validator does make the effort of evaluating all the miners. The signal v_t^j is only observed by j if $a_t^j = 1$.
2. $w_t^j \in \Delta^d$, which is the reported weight vector, where Δ^d denotes the simplex of probability vectors $\{z \in \mathbb{R}_+^d : \sum_{i=1}^d z_i = 1\}$.

The report profile at round t is

$$\mathcal{X}^{|\mathcal{Z}|}. \tag{1}$$

There pairs of actions are generated by a strategy of the following form.

The set of possible histories observed by j up to round t is denoted

$$\mathcal{H}_{t-1} = (\underbrace{\mathcal{X}^{|\mathcal{Z}|}}_{\text{reports}} \times \underbrace{\mathcal{X}}_{\text{observed signal}})^{t-1}. \tag{2}$$

[2]The sequence $\mathbf{v}_t^j$ essentially replaces the notion of a static type in Bayesian games.

A strategy σ for j is a sequence of mappings from history and current signal to an action-pair:

$$\sigma_t : \mathcal{H}_{t-1} \times \underbrace{\mathcal{X}}_{\text{current signal}} \to \{0,1\} \times \mathcal{X}, \quad \text{for all } t \geq 1. \tag{3}$$

Observe that j does not observe the action a_t^k for other validators, nor the signals $\mathbf{v}_t^k$ for other validators. The action-pair can be written as the strategy applied to the history H_{t-1} and signal v_t^j:

$$(a_t^j, w_t^j) = \sigma(H_{t-1}, v_t^j). \tag{4}$$

In particular, we can define the honest strategy of a validator as follows.

Definition 1 (Honest strategy)**.** *The honest strategy σ^* for a validator j is*

$$\sigma^*(H_{t-1}, v_t^j) = (1, v_t^j), \quad \textit{for all } t. \tag{5}$$

2.1 Yuma consensus and utility functions

To define the utility function for validators, we first introduce the calculation of dividends. Let $w_t^j(i)$ denote the validator j's weight on miner i at time t. The consensus $\bar{w}_i^t$ of miner i is a function of the form

$$\bar{w}_t(i) = F((S_j, w_t^j)_{j \in \mathcal{Z}}). \tag{6}$$

Next, we define the consensus-clipped weight from validator j to miner i.

$$\bar{w}_t^j(i) = \min(w_t^j(i), \bar{w}_t(i)). \tag{7}$$

Validator trust is definded as the total validator weight after consensus-clipping:

$$T^j = \sum_{i=1}^{d} \bar{w}_t^j(i). \tag{8}$$

With bonds penalty $\beta \in [0,1]$, e.g., $\beta = 1$ in the implementation, we define stake weighted as

$$\tilde{w}_t^j(i) = (1-\beta) w_t^j(i) + \beta \bar{w}_t^j(i). \tag{9}$$

For a fixed sequence α_t, the validator bond that validator j has over miner i is

$$\Delta_t^j(i) = \frac{S_i \tilde{w}_t^j(i)}{\sum_{k \in \mathcal{Z}} S_k \tilde{w}_t^k(i)}, \tag{10}$$

$$B_t^j(i) = \alpha_t \Delta_t^j(i) + (1-\alpha_t) B_{t-1}^j(i) \tag{11}$$

$$= \alpha_t \Delta_t^j(i) + (1-\alpha_t)\alpha_{t-1}\Delta_{t-1}^j(i) + \ldots + (1-\alpha_t)\alpha_{t-1}\ldots\alpha_1 B_1^j(i), \tag{12}$$

$$B_1^j(i) = 0, \quad \text{for all } i. \tag{13}$$

Lastly, validator j's dividend is the sum of bonds scaled by miner incentives I_t^i:

$$D_t^j(w_t^j, \mathbf{w}_t^{-j}) = \sum_{i \in \mathcal{M}} B_t^j(i) \cdot I_t^i. \tag{14}$$

Remark 1 (Miner incentives)**.** *Each miner i's incentive are actually a function of the weight profile of the form*

$$I_t^i = G_i(\mathbf{w}_t). \tag{15}$$

Finally, when the sequence of signals observed by all validators is $\{\mathbf{v}\}$, the utility function of j adopting a strategy σ is

$$u_j(\sigma^j, \sigma^{-j}; \mathbf{v}) = \sum_{t=1}^{\infty} \gamma^t (D_t^j(w_t^j, \mathbf{w}_t^{-j}) - \mu a_t^j), \tag{16}$$

where μ is the cost of the effort of evaluating all the miners in one round.

2.2 Single-round dividend

Although we really care about strategies that maximize the long-term total utility u_j, in this section, we get some insights by restricting our attention on the dividend of a single round t, and the reported weight that maximizes it. Namely, given stake S_t and a fixed profile of opponent reports $\mathbf{w}_t^{-j}$ with the resulted consensus $(\bar{w}_t(i))$, what is the best response?

The following proposition says that the validator can maximize its one-round dividend by reporting weights equal to the consensus $\bar{w}_t(i)$ for each miner i. This is because the dividend is a non-decreasing function of the weight for small values of the weight below the consensus; moreover, weight values above the consensus are "lost" due to clamping. For simplicity, we first make two assumptions.

Assumption 1 (No clamp)**.** *We assume that that $\beta = 1$ and that the profile of opponent reports $\mathbf{w}_t^{-j}$ is such that the clamp of (7) is not in effect, i.e.,*

$$\min(w_t^j(i), \bar{w}_t(i)) = w_t^j(i). \tag{17}$$

Assumption 2 (Constant miner incentives)**.** *We assume that for every miner i, and for every element z of the profile of weights $\mathbf{w}$, we have $\frac{d}{dz} I_t^i(\mathbf{w}) = 0$.*

Proposition 1 (Monotone non-decreasing dividend)**.** *For a fixed profile of opponent reports $\mathbf{w}_t^{-j}$, the dividend $D_t^j(z, \mathbf{w}_t^{-j})$ to j is monotone non-decreasing in each of the elements of z.*

Proof. Consider without loss of generality the case of two miners ($d = 2$). By definition, we have

$$D_t^j((w_1, w_2) = B_t^j(1) \cdot I_t^1 + B_t^j(2) \cdot I_t^2 \tag{18}$$

and

$$\frac{d}{dw_i} D_t^j((w_1, w_2), \mathbf{w}_t^{-j}) = I_t^1 \frac{d}{dw_i} B_t^j(1) + B_t^j(1) \frac{d}{dw_i} I_t^1 \tag{19}$$

$$+ I_t^2 \frac{d}{dw_i} B_t^j(2) + B_t^j(2) \frac{d}{dw_i} I_t^2. \tag{20}$$

It follows from Assumption 2 and $\frac{d}{dw_2} B_t^j(1) = \frac{d}{dw_1} B_t^j(2) = 0$ that

$$\frac{d}{dw_i} D_t^j((w_1, w_2), \mathbf{w}_t^{-j}) = I_t^1 \frac{d}{dw_i} B_t^j(1) + I_t^2 \frac{d}{dw_i} B_t^j(2) \tag{21}$$

$$\frac{d}{dw_1} D_t^j((w_1, w_2), \mathbf{w}_t^{-j}) = I_t^1 \frac{d}{dw_1} B_t^j(1), \tag{22}$$

$$\frac{d}{dw_2} D_t^j((w_1, w_2), \mathbf{w}_t^{-j}) = I_t^2 \frac{d}{dw_2} B_t^j(2). \tag{23}$$

Next, observe that

$$\frac{d}{dw_i} B_t^j(i) = \alpha_t \frac{d}{dw_i} \Delta_t^j(i), \tag{24}$$

since $\Delta_{t-1}^j(i), \ldots, \Delta_1^j(i)$ are fixed at time $t-1$.

$$\frac{d}{dw_i} \Delta_t^j(i) = 0, \quad i = 1, 2. \tag{25}$$

By definition and by Assumption 1, we have

$$\frac{d}{dw_i} \Delta_t^j(i) = \frac{d}{dw_i} \left[\frac{S_j w_i}{S_j w_i + \sum_{k \neq j} S_k w_t^k(i)} \right] \tag{26}$$

$$= \frac{S_j}{S_j w_i + \sum_{k \neq j} S_k w_t^k(i)} + S_j w_i \frac{-S_j}{(S_j w_i + \sum_{k \neq j} S_k w_t^k(i))^2} \tag{27}$$

$$= \frac{S_j}{S_j w_i + \sum_{k \neq j} S_k w_t^k(i)} \left(1 - \frac{S_j w_i}{S_j w_i + \sum_{k \neq j} S_k w_t^k(i)} \right) > 0. \tag{28}$$

Finally, the claim follows from the facts that $I_t^1, I_t^2 > 0$ and $\alpha_t > 0$. □

Figure 1 illustrates Proposition 1: it shows that the dividend rate D^j / S_j decreases monotonically as validator j's reported weight diverges from the consensus in mean squared error (MSE).

3 Copying strategies

The goal of the dividends in Bittensor is to encourage validators to adopt the honest strategy of Definition 1. The strongest level of encouragement would be if the honest strategy is dominant, in the sense that for every profile of

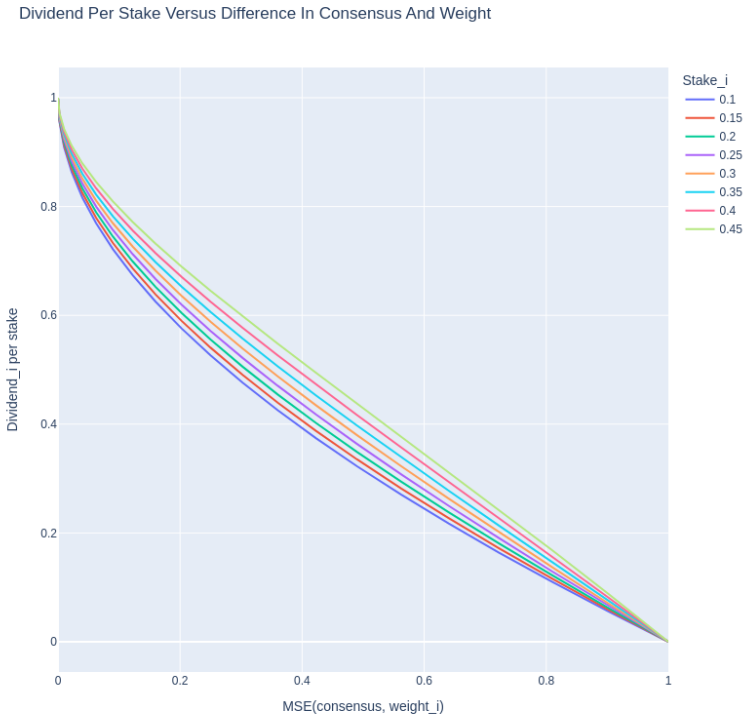

Figure 1: Dividend rate for validator j as a function of the mean squared error between its reported weight and the consensus.

opponent strategies σ^{-j}, the validator j gets more dividend by following the honest strategy. Namely, for every set of strategies σ^j, σ^{-j}, every sequence of signals $\mathbf{v}$, and every σ^j, we have

$$u_j(\sigma^*, \sigma^{-j}; \mathbf{v}) \geq u_j(\sigma^j, \sigma^{-j}; \mathbf{v}). \tag{29}$$

The weaker level of encouragement is that the honest strategy profile $(\sigma^*, \ldots, \sigma^*)$ is an equilibrium, in the sense that for every sequence of signals $\mathbf{v}$, we have

$$u_j(\sigma^*, (\sigma^* \ldots \sigma^*); \mathbf{v}) \geq u_j(\sigma^j, (\sigma^* \ldots \sigma^*); \mathbf{v}). \tag{30}$$

In this paper, we do not consider the ultimate goal of encouraging adoption of the honest strategy, but an intermediate goal of discouraging the adoption of the copying strategy. Next, we define the notion of a copying strategy, making a distinction between doing the validation work or not.

Remark 2 (Occasional tests)**.** *To detect that penalize collusion between validators, miners, and subnet owners, one approach to randomly insert occasional tests, where validators are presented with outputs for which we know the ground truth. If these tests are inserted with probability p, and penalties for failing these tests are of the order of* $1/p$*, then coalitions do not gain from doing dishonest work. This approach is left as a future work.*

3.1 Model of copying

We consider the problem of one validator j copying the weights that another validator k assigns on the set of miners. Each validator j is characterized by a sequence of scores (or weights) that we denote

$$w^j = w_0^j, w_1^j, \ldots, \tag{31}$$

where $w_t^j \in \mathbb{R}^n$. For simplicity, we call w_j a weight process. These weights are for $n = 64$ miners in this paper, but can be extended to the $n = 32$ subnets in a straightforward way.

Before defining the notion of copying, we first consider how to compare two weight vectors such as w_t^j from validator j at time t and w_{t-1}^k from validator k at time $t - 1$.

Let

$$\Delta(x, y) \tag{32}$$

denote the distance between two weight vectors in $\mathbb{R}^n$. We define the notion of copying as follows.

Definition 2 ((ϵ, δ)-copy of weight process)**.** *Let $\epsilon > 0, \delta > 0$ denote two fixed thresholds. We say that validator j's weight process y_t is an (ϵ, δ)-copy of the weight process x_t of validator k if there exist a sequence of delays*

$$d_t > \delta, \quad \textit{for all } t, \tag{33}$$

such that

$$\Delta(y_t, x_{t-d_t}) \leq \epsilon, \quad \textit{for all } t. \tag{34}$$

Definition 3 (Lazy-copying, Active-copying)**.** *A strategy $\tilde{\sigma}$ for j is a lazy-copying if there exists another validator k and a sequence $\{y_t\}$ of reports, such that y_t is an (ϵ, δ)-copy of r_t^k, and*

$$\tilde{\sigma}(H_{t-1}, v_t^j) = (0, y_t), \quad \textit{for all } t. \tag{35}$$

Likewise, a strategy $\tilde{\sigma}$ is active-copying if the right-hand side of (35) *is replaced by $(1, y_t)$.*

Lazy-copying strategy is dominated by the honest strategy σ^* if for every sequence of signals $\mathbf{v}$,

$$u_j(\sigma^*, \sigma^{-j}; \mathbf{v}) \geq u_j(\tilde{\sigma}, \sigma^{-j}; \mathbf{v}). \tag{36}$$

Observe that Definition 3 only allows copying weights from a single validator k, but one validator may copy from a combination of other validators. For instance, we define next a version of the fictitious play strategy [Bro51] against the most recently observed profile of all opponents' actions.

Definition 4 (Consensus-copying)**.** *A strategy $\tilde{\sigma}$ for j is a consensus-copying if it reports the best-response against $\mathbf{r}_{t-1}^{-j}$ for all t, i.e.,*

$$\tilde{\sigma}(H_{t-1}, v_t^j) = (a_t^j, r_t^j), \tag{37}$$

$$r_t^j \in \arg\max_{r \in \mathcal{X}} D_t^j(r, \mathbf{r}_{t-1}^{-j}) \quad \textit{for all } t. \tag{38}$$

By setting a larger threshold value ϵ, we allow more weight processes to be labeled as copies. Similarly, by setting a smaller threshold value δ, we label a weight process as a copy even if it is slightly behind another one (at every time step).

There is a flaw with this definition, as illustrated in the following figure. Consider two validators j and k: whose weight processes w^j and w^k are represented by the blue and red plots respectively in Figure 2. At first, we may be tempted to label j as the copier because its weight process (blue) appears to be a copy of the weight process of k (red) with a small delay. However, observe that the weight process of k (red) is itself a copy of the weight process of j (blue) with a much larger delay. Therefore, both validators can be labeled as copiers. One approach to avoid the extremely wide applicability of the label of "copying" would be to add constraints on the delay sequence d_t. If we take inspiration from protection of copyright material, we can put a limit on the duration of the protection. However, nothing prevents a validator from briefly reporting an extensive set of different weights for the sole purpose of labeling all other validators as copiers.

Furthermore, it is conceivable that one validator has a weight process that is labeled as a copy, even if it does validation work honestly and diligently.

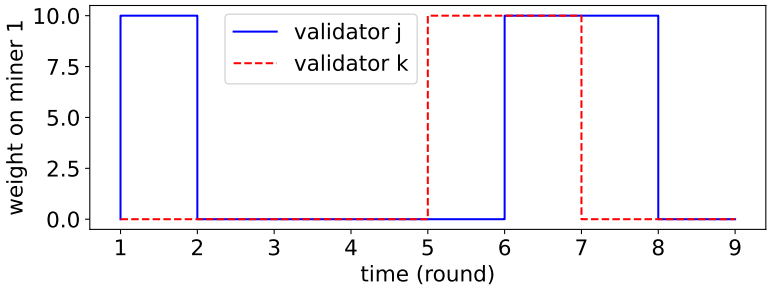

Figure 2: Weight processes for two validators: which one is the copier? The blue sequence copies recent values of the red sequence, but the red sequence also copies the much earlier values of the blue sequence.

For instance, this validator could be located in a farther location and incur consistently higher communication delay.

We can divide the problem into two settings:

1. The Verifiable setting, where the ground truth—miner performance—is reveal with some delay (e.g., weather prediction). In this setting, we can resort to the commit-then-reveal approach, where validator essentially send their weights in sealed envelopes that are opened simultaneously once they are all received. We want to simply make copying computationally intractable, without penalizing validators who report the same weight.

2. The Non-verifiable setting, where there is no ground truth on miner performance (e.g., image generation, or text summary). We design a dividend rule that makes the honest strategy dominant or at least an equilibrium.

We consider the Verifiable setting next, and the Non-verifiable setting in Section 6.

4 Commit-Reveal approach for verifiable setting

In this section, we propose an approach to disincentivize weight copying by replacing the weight communication step with the two-step commitment scheme [Gol01]: first, committing (and communicate) to a hashed version of a set of weights, then revealing a set of weights matching the hash. This scheme is analogous to sending a locked box containing a set of weights, and then sending the key to the lock at a later time.

The same commitment scheme between every validator j and the chain is described in Algorithm 1 for a single round t corresponding to a single emission event. The same scheme is then repeated for every round t. Our commitment scheme differs from classical commitment schemes in setting deadlines for the commitment and reveal steps, and handling the situation where the revealed value is rejected.

Algorithm 1 Commit-Reveal scheme for validator j

Let h_j denote the function that generates the commitment value for validator j: it is composed of a hash function with a fixed security parameter (BLAKE2 in our release [**?**]) and j's digital signature. It is known by all parties.
Let t denote the current round (emission event) index, and w_t^j denote the weight that the validator j intends to set for round t. Let w_{t-1}^j denote the weight for the previous round.
Let time $m = 0$ denote the start of a new round. Fix a time deadline $T_1 > 0$ to receive the commitment and time deadline $T_2 > T_1$ to receive the revealed weight. Once time reaches $m = T_2$, it is reset to 0 and a new round starts.
(Commit Step: Validator-side) Validator j computes a set of weights w, and samples o uniformly at random, and sends the commitment value $h_j(w, o)$ before time T_1. This $h_j(w, o)$ is observed by all participants.
if (Commit Step: Chain-side) If chain receives a new commitment value z from Validator j before time T_1 **then**
  the chain records $C_t = z$,
else
  the chain records the previous commitment as the fall-back value: $C_t = C_{t-1}$.
end if
(Reveal Step: Validator-side): Validator j sends (w, o) to the chain (with digital signature) before time T_2. This pair (w, o) is observed by all participants.
if (Reveal Step: Chain-side) Chain receives (w, o) from the validator before time T_2 and $C_t = h_j(w, o)$ **then**
  the weight vector w is accepted and the chain records $w_t^j = w$.
else
  the weight vector is left unchanged from the previous round: $w_t^j = w_{t-1}^j$.
end if

Observe that the commitment scheme guarantees two properties:

1. hiding: between the first step and the second step of validator j (i.e., the commit reveal weights interval of duration at least $T_2 - T_1$), no participant can gain any knowledge of the weight vector of j;

2. binding: there exists only one value that can be accepted as the revealed weight vector at the second step.

Our design principle is to rely on the hiding property to limit a validator's ability to copy weights in a timely matter. Copying is only computationally feasible with delay greater than $T_2 - T_1$. We will show that, under appropriate assumptions, this delay makes copying generate less dividend than the honest strategy for every round.

The commitment scheme of Algorithm 1 has the following computationally hiding and perfectly binding guarantees [Dam99].

Theorem 1 (Hiding and (τ, ϵ)-Binding)**.** *For uniform random variables o, o' and any w, w', we have that the distributions of $h_j(w, o)$ and $h_j(w', o')$ are computationally indistinguishable. For any algorithm running in time at most τ, the probability that it computes w, o, w', o' such that $h_j(w, o) = h_j(w', o')$ and $w \neq w'$, is at most ϵ.*

Remark 3 (Removing r, r')**.** *What could go wrong if we do not use random variables r, r'? In this case, since BLAKE2 is a deterministic function, the commitment value will be the same for all rounds where a validator gives the same weight. If a copier keeps track of a list of commitment values, then it can copy weights every time it observes a value already in that list.*

In the following theorem, we show that under the commitment scheme, the dividend received by a validator j is at least an amount η less than the validator k if j copies k's weight.

Proposition 2 (Copying strategy dominated)**.** *Suppose that there exist two opponent report-profiles Q_1, Q_2 and an $\epsilon > 0$ such that*

$$\left| \max_{r \in \mathcal{X}} D_t^j(r, Q_1) - \max_{s \in \mathcal{X}} D_t^j(s, Q_2) \right| > \epsilon. \tag{39}$$

Suppose that the cost of validation satisfies $\mu < \epsilon/2$. Consider a validator j, there exists a sequence of opponent report profiles $\mathbf{r}^{-j}$ where the lazy-, active-, and consensus-copying strategies are dominated by the honest strategy.

It is important to note that we only give a guarantee on the suboptimality the copying strategy on a single round (emission event); the extension to the utility over a time horizon is an open problem.

Proof. Consider the case where j adopts the consensus-copying strategy. We construct $\mathbf{r}^{-j}$ as follows. Suppose that all opponents of j are honest validators,

and that every honest validator observes the same sequence of signals b_t from the miners. Suppose that b_t is an i.i.d. sequence of Bernoulli random variables with $\mathbb{P}(b_t = 1) = 1/2$. Therefore, at the commit deadline of each round t, the opponent report profile updates to

$$\mathbf{r}_t^{-j} = Q_{b_t}. \tag{40}$$

Since j is consensus-copying, during the period between commit-deadline and reveal-deadline (of duration $T_2 - T_1$), j optimizes against the most recent weight $r_{t-1}^{-j} = Q_{b_{t-1}}$ by reporting

$$r_t^j = \arg\max_{r \in \mathcal{X}} D_t^j(r, Q_{b_{t-1}}), \tag{41}$$

and receiving the dividend

$$D_t^j(r_t^j, Q_{b_t}). \tag{42}$$

If j instead did the honest work, it would observe the realization of b_t and receive dividend (non-random)

$$\max_{r \in \mathcal{X}} D_t^j(r, Q_{b_t}). \tag{43}$$

Observe that, by (39)

$$\begin{aligned}
\mathbb{E}\left[D_t^j(r_t^j, Q_{b_t})\right] &\leq \max_{r \in \mathcal{X}} D_t^j(r, Q_{b_t}) \mathbb{P}(b_t = b_{t-1}) && (44)\\
&\quad + \left(\max_{r \in \mathcal{X}} D_t^j(r, Q_{b_t}) - \epsilon\right) \mathbb{P}(b_t \neq b_{t-1}) && (45)\\
&\leq \max_{r \in \mathcal{X}} D_t^j(r, Q_{b_t}) - \epsilon/2. && (46)
\end{aligned}$$

Hence, j incurs an expected loss relative to the honest strategy of $\epsilon/2$ per round. Since the cost of validation μ is assumed less than $\epsilon/2$, the copying strategy is dominated as claimed.

The argument is similar for the cases of active- and lazy-copying, because a copier's report is a function of the sequence $b_1, \ldots, b_{t-1}$, whereas b_t is independent. □

5 Experiment

In this section, we show that the commit-reveal approach penalizes the dividend of a validator who adopts the consensus-copying strategy. We do so via simulation using historical Bittensor data, cf. weights and stake from block[3] 2987500 to 3001180.

[3]Note that subnets 11 and 25 are omitted from the data sample because they were newly registered during that time.

We simulate an artificial validator j who possesses 5% of the stake on each subnet and who follows the consensus-copying strategy by always copying the most recently observed consensus as its reported weight.

We use the relative dividend rate of the copier j,

$$G^j = \frac{D^j/S^j}{\underset{i\in\mathcal{Z}\setminus\{j\}}{\text{median}}\{D^i/S^i\}}, \tag{47}$$

to measure the success of the commit-reveal approach. Here, validator dividend is normalized by the corresponding validator stake as dividend is linear in the amount of stake. Further, we use median as the baseline for comparison.

5.1 Experiment result

Let $\tau \in [0, 1]$ be validator take, which is 18% by default in the implementation; N^j be the set of norminators that stake to validator j; E^j be the emission to validator j; and L^n be the emission to norminator n who stake to j:

$$E^j = 0.5(D^j + \sum_{n\in N^j} \tau D^n), \tag{48}$$

$$L^n = 0.5(1-\tau)D^n. \tag{49}$$

The following three ranges of values for G^j lead to three distinct regimes.

1. When $G > 1$, validator j who adopts the lazy consensus-copying strategy is having an advantage over other honest validators, which they can gain more dividend per stake. Such a validator j would also attracts norminators to stake into them, as a result validator j would be gaining influence over the subnet as their stake grow. Moreover, validator j would earn τ from the norminator's dividend D^n (48).

2. When $(1-\tau) < G \leq 1$, validator j is losing to other honest validators in dividend per stake. Norminators would stop staking to valdiator j and shift to stake to the honest validators who gives a higher dividend per stake. Validator j would be losing their influence through stake to honest validators and stop earning from dividend take. Validator j can still benefit from consensus copying as they avoid becoming a nominator and losing the validator take τ to the validator (49) .

3. When $G \leq (1-\tau)$, consensus copying would no longer be beneficial for validator j, validator j should opt to become a nominator.

Out of 30 subnets, 10 subnets who cannot pass the $G = 1$ threshold was listed on Figure 3 - plot 1, and 20 subnets who can pass the $G = 1$ threshold were listed in Figure 3 - plot 2. For these 20 subnets, nominators would no longer be incentivised to stake to validator j. Yet, none of the 30 subnets can reach the $1-\tau$ threashold, so we can expect these consensus copying validators would still exist in the subnets.

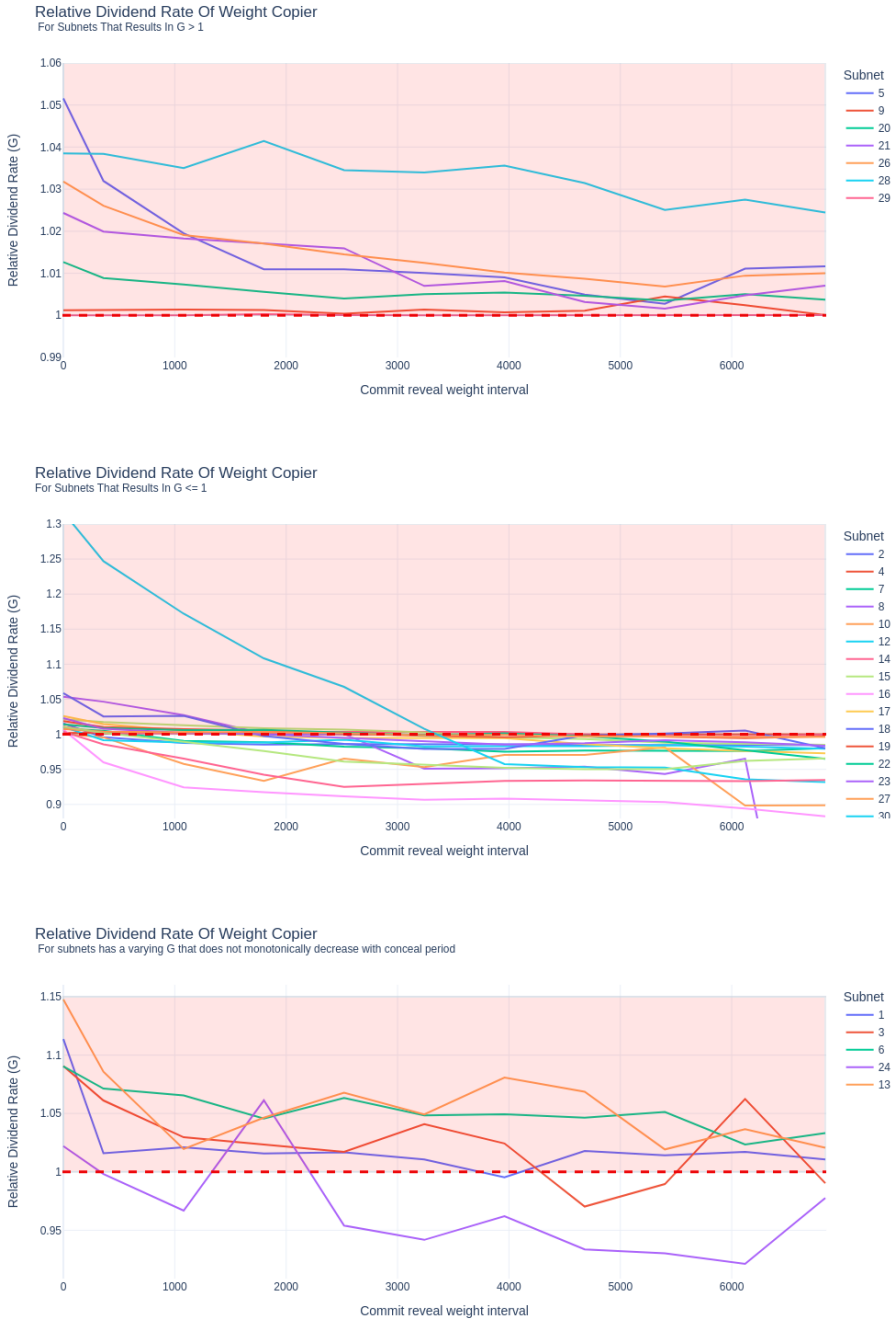

Figure 3: Relative dividend rate (G) of the weight copier as the commit reveal weights interval increases, with commit reveal weights interval ranging from 0 tempo (no conceal at all) to 19 tempo (22.8 hours). The red dotted line is where the weight copier receives the same dividend return as the median validator, while the red area highlight the event $G > 1$. The scenario where these plots was situated are indicated in the subtitle.

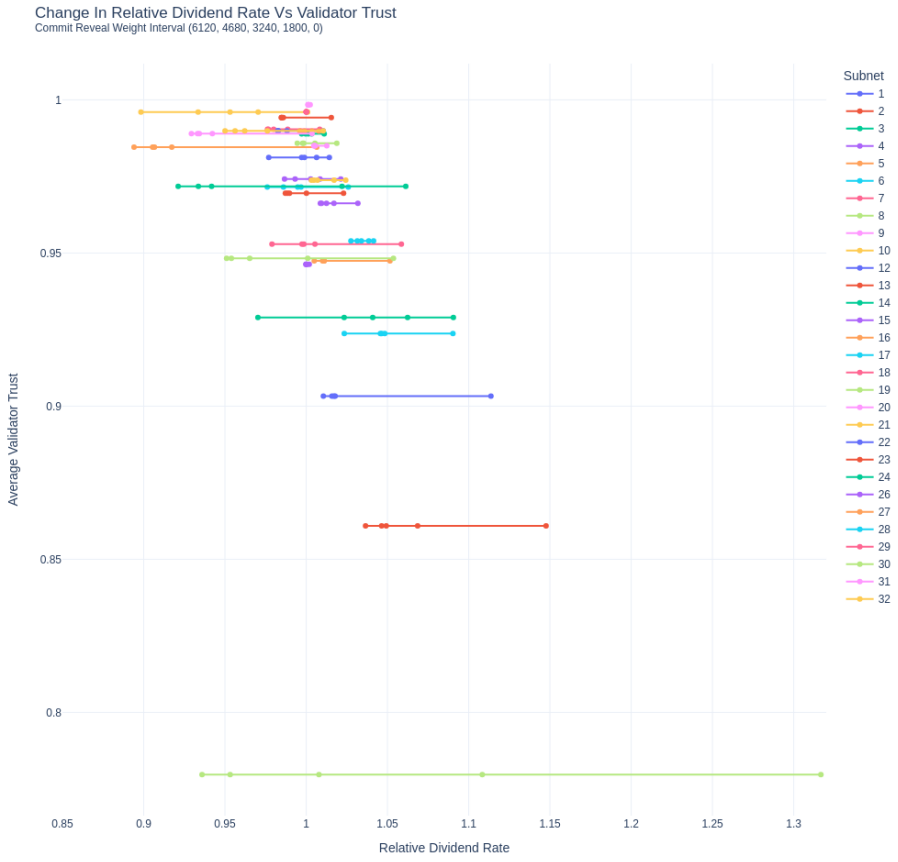

Figure 4: Relationship between relative dividend rate and validator trust T^j for each subnet, where we sample (from left to right) the values $6120, 4680, 3240, 1800, 0$ for the commit reveal weights interval.

Subnet	commit_reveal_weight_interval	num of tempos	hours	immunity_period
16	360	1	1.2	14400
2	360	1	1.2	5000
31	360	1	1.2	7200
12	360	1	1.2	2440
10	360	1	1.2	5000
7	1080	3	3.6	10800
32	1080	3	3.6	7200
17	1800	5	6.0	5000
4	2520	7	8.4	5000
23	2520	7	8.4	5000
24	2520	7	8.4	5000
8	3240	9	10.8	65535
19	3240	9	10.8	7000
27	3240	9	10.8	9000
30	3960	11	13.2	10000
15	3960	11	13.2	6000
14	4680	13	15.6	5000
22	4680	13	15.6	7200
3	6840	19	22.8	5000
18	6840	19	22.8	5000
29	7560	21	25.2	7200
28	7560	21	25.2	10800
26	7560	21	25.2	5000
20	7560	21	25.2	5000
6	7560	21	25.2	5000
13	7560	21	25.2	12000
5	7560	21	25.2	5000
9	7560	21	25.2	5000
21	7560	21	25.2	10800
1	7560	21	25.2	7200

Table 1: Minumum required commit reveal weights interval for each subnet such that the relative dividend rate of the weight-copier j drops below 1. Our simulation did not find the required commit reveal weights interval for the subnets 1, 5, 6, 9, 13, 20, 21, 26, 28, 29. Immunity period is the amount of time (the length of conceal blocks) it takes for a new miner or validator to gain enough weight to avoid deregistration.

Moreover, we can observe from Figure 3, plot 2 that, on most of the subnets, G^j has converged when commit reveal weights interval reaches 1800 (5 tempos). A longer commit reveal weights interval value yields little benefit, but slows down the evaluation of miners.

In Figure 3, plot 3, it shows some examples where relative dividend rate G would not be monotonically decreasing with an increased commit reveal weights interval. There is no strict forward explanation for such event.

Figure 3, plot 2 shows that it is very profitable to run a consensus-copying strategy on subnet 30, whereas consensus-copying only gives 0.95 relative dividend rate on subnets 10, 16, 24, 31, 32 with a long enough commit reveal weights interval.

Figure 4 shows a strong correlation between valdiator trust and relative dividend rate. When the validator trust is higher, relative dividend rate tends to be lower before and after a certain amount of commit reveal weights interval. This could be explained by the intuition that validator trust indicates the difference between the weight and the consensus. When this difference is large, then validator j can easily outperform honest validators by a larger margin through setting weight equals to consensus, which was as well shown previously in Figure 1.

Table 1 shows the minumum required commit reveal weights interval for each subnet such that the relative dividend rate from a weight copier drops below 1. It is important that we compare this number with the immunity period: increasing commit reveal weights interval slows down the discovery of new miner and puts them at a higher risk to be deregistered. We advise subnet owners to increment the immunity period by the number of conceal blocks.

6 Peer-prediction for the non-verifiable setting

In the non-verifiable setting, to avoid labeling these honest validators as copiers, we design the payments to validators such that

1. strategies that do the validation work ($a_t^j = 1$) dominate those that do not do the work ($a_t^j = 0$), including copying other participants' validation reports,

2. strategies that report the true observed signal ($\mathbf{w}_t^j = \mathbf{v}_t^j$) dominate those that report anything else (including false reports coordinated with other participants).

This dominance holds at least for one profile of all opponent strategies, but ideally for all such profiles.

This can be done by computing validator rewards according to peer prediction rules [MRZ05, SAFP16, KSL16, FJR17] approaches. The peer-prediction approach uses monetary incentives to encourage honest validation in the absence of ground truth.

The most basic setting is as follows. Suppose first that all validators score the same miner m during each block. Consider a fixed validator j. Let S denote

a finite set of weight values and let $s_m^j \in S$ denote the signal that j observes about the quality of miner m's work. We also assume that obtaining this signal requires an effort on the part of j, which incurs a cost c_j that is constant across every miner m. The validator j then submits a report r_m^j, which may be different from the observed signal. Finally, validator j receives a payment $\pi(r^j, \mathbf{r}^{-j})$. The peer-prediction approach is shown in Algorithm 2.

Algorithm 2 Peer prediction

Let $R : \Delta^{|S|} \times S \to \mathbb{R}$ be a strictly proper scoring rule as defined in [MRZ05]. Let the joint distribution f of $\mathbf{s}$ be known by all participants:

$$f(z) = \mathbb{P}(\mathbf{s} = z). \tag{50}$$

for $j \in \mathcal{Z}$ **do**

Fix a reference validator $X_j \in \mathcal{Z} \setminus \{j\}$ uniformly at random, and send reference X_j to validator j,

Validator j outputs its report r_j,

Calculate the marginal distribution of s^{X_j} conditional on $s^j = r^j$:

$$p^j(z) = \mathbb{P}(s^{X_j} = z \mid s^j = r^j) = \frac{\mathbb{P}(s^{X_j} = z)}{\mathbb{P}(s^j = r^j)}, \quad \text{for all } z \in S, \tag{51}$$

$$\mathbf{p}^j = (p^j(z))_{z \in S}. \tag{52}$$

end for

Send each validator j a payment of $\pi(r^j, \mathbf{r}^{-j}) = R(\mathbf{p}^j, r^{X_j})$.

6.1 Multi-task mechanism for peer prediction

Next, we consider the case where each validator scores multiple miners in each block. We present the multi-task mechanism of [SAFP16].

These mechanisms are parameterized by score matrix S.

1. Assign each agent to two or more tasks, with at least one task in common, and at least three tasks total.

2. Let r_k^1 denote the report received from agent 1 on task k (and similarly for agent 2). Designate one or more tasks assigned to both agents as bonus tasks (set M_b). Partition the remaining tasks into penalty tasks M_1 and M_2, where $|M_1| > 0$ and $|M_2| > 0$ and M_1 tasks have a report from agent 1 and M_2 a report from agent 2.

3. For each bonus task $k \in M_b$, pick a random $L \in M_1$ and $L' \in M_2$. The payment to both agent 1 and agent 2 for task k is

$$S(r_k^1, r_k^2) - S(r_L^1, r_{L'}^2). \tag{53}$$

4. The total payment to an agent is the sum total payment across all bonus tasks.

7 Related works

Our model of interaction between validators is most similar to the model of peer prediction [MRZ05]. The distinguishing features of our problem are the repeated interactions of the same set of participants over many rounds, and the division of a fixed amount of reward each round—instead of arbitrary rewards. Moreover, in contrast to use cases in the literature [SAFP16], the rewards on the line are monetary and very substantial—of the order of millions of dollars per month. In this setting, selecting the correct strategy and safeguarding against adversarial attacks are very consequential. In the peer prediction literature, the free-rider problem is also called "arbitrage" [FJR17, Section 5.2][4].

In the current implementation, we have a repeated game [FT91] with incomplete information[5] (signals are private to each validator). Unlike extensive-form games, there is no predetermined order in which validators take their actions: each validator chooses a time at which to take their action by considering a trade-off between avoiding missing a deadline and preventing others from copying. The proposed commit-reveal approach essentially imposes simultaneous validator actions each round.

The commit-reveal approach is preemptive solution approach to tackle the problem of copiers. Digital watermarking presents a reactive solution by detecting instances of copying [Che00]. Other solutions are proposed for crowdsourcing applications [DBES09, JNX+19, FSS+23, WZC+].

8 Open problems

This work scratches the surface of the design and operation of decentralized and anonymous markets where money and query-responses are exchanged. When we have access to the true quality of these responses, we can easily penalize bad responses, but we would also like to penalize good responses that are obtained by copying other participants. We propose to do so via the commit-reveal approach, which prevents copying during a "commit reveal weights interval." Under appropriate conditions, this reduces the rewards to the copier so as to make copying unappealing. When the true quality of responses is not readily available, verifying the quality of responses is arguably more difficult, but still achievable through the peer-prediction approach.

Along the way, we have identified many additional important open problems:

1. designing *both* incentives and registration rules[6] to encourage desirable

[4] This is different from arbitrage in the context of efficient markets [Dav08].

[5] Our setting is one of perfect information since all histories of action profiles are observed and recorded on-chain.

[6] A set of registration rules currently act as a barrier of entry to validators and miners.

behaviour,

2. comparing the honest strategy to other heuristic strategies such as fictitious play [Bro51] and dynamic best-response [SMK18],
3. modeling the effect of miner incentives on validator dividends through (15)—allowing us to analyze the occasional-tests approach of Remark 2,
4. extending Proposition 2 from the dividend in a single round to the utility over the entire horizon,
5. how should validators choose the time at which they send their commitment and revelation values[7]?
6. how do we set immunity periods (cf. Section 5) informed by the duration of the commit reveal weights interval?

References

[Bro51] George W. Brown. Iterative solution of games by fictitious play. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*. Wiley, New York, 1951.

[Che00] Brian Chen. *Design and analysis of digital watermarking, information embedding, and data hiding systems*. PhD thesis, Massachusetts Institute of Technology, 2000.

[Dam99] Ivan Damgård. *Commitment Schemes and Zero-Knowledge Protocols*, pages 63–86. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[Dav08] Mark H. A. Davis. Louis bachelier's "theory of speculation". 2008.

[DBES09] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. Truth discovery and copying detection in a dynamic world. *Proc. VLDB Endow.*, 2(1):562–573, aug 2009.

[FC05] Michal Feldman and John Chuang. Overcoming free-riding behavior in peer-to-peer systems. *SIGecom Exch.*, 5(4):41–50, jul 2005.

[FJR17] Boi Faltings, Radu Jurca, and Goran Radanovic. Peer truth serum: Incentives for crowdsourcing measurements and opinions, 2017.

[FSS+23] Xiu Fang, Suxin Si, Guohao Sun, Wenjun Wu, Kang Wang, and Hang Lv. A domain-aware crowdsourcing system with copier removal. In Jian Dong and Long Zhang, editors, *Proceedings of the International Conference on Internet of Things, Communication and Intelligent Technology*, pages 761–773, Singapore, 2023. Springer Nature Singapore.

[7] If they send too early, they have less time to perform the validation work or give information to their opponents, if they send too late, they risk missing the deadline.

[FT91] Drew Fudenberg and Jean Tirole. *Game theory*. MIT press, 1991.

[Gol01] Oded Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2001.

[HCW05] D. Hughes, G. Coulson, and J. Walkerdine. Free riding on gnutella revisited: the bell tolls? *IEEE Distributed Systems Online*, 6(6), 2005.

[JA05] Seung Jun and Mustaque Ahamad. Incentives in bittorrent induce free riding. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems*, P2PECON '05, page 116–121, New York, NY, USA, 2005. Association for Computing Machinery.

[JNX+19] Lingyun Jiang, Xiaofu Niu, Jia Xu, Dejun Yang, and Lijie Xu. Incentivizing the workers for truth discovery in crowdsourcing with copiers. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1286–1295, 2019.

[KSL16] Yuqing Kong, Grant Schoenebeck, and Katrina Ligett. Putting peer prediction under the micro(economic)scope and making truth-telling focal, 2016.

[MRZ05] Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

[Ope23] OpenTensor. Yuma consensus, 2023.

[RC65] A. Rapoport and A.M. Chammah. *Prisoner's Dilemma: A Study in Conflict and Cooperation*. Ann Arbor paperbacks. University of Michigan Press, 1965.

[SAFP16] Victor Shnayder, Arpit Agarwal, Rafael Frongillo, and David C. Parkes. Informed truthfulness in multi-task peer prediction, 2016.

[SMK18] Brian Swenson, Ryan Murray, and Soummya Kar. On best-response dynamics in potential games, 2018.

[WZC+] Shuang Wang, He Zhang, Long Chen, Xiaoping Li, Taotao Cai, and Quan Z Sheng. Dependent truth discovery from multiple sources. *Available at SSRN 4691039*.