# Does it Harm Science to Suppress Dissenting Evidence? - Online Appendix

## 1 Other Results

Figure 1 shows the effect of increasing the number of tests performed each round. In this case, both *Random* and *Hiding Dissent* do better with more tests performed each round, because there is less chance of outlier results. However, the dynamics of *Random* reaching correct consensus more often, but *Hiding Dissent* reaching correct consensus faster, remain robust. The effect of increased results improving success when consensus is reached within a single round is slightly decreased, because with more tests the likelihood of large outlier results is reduced anyway.

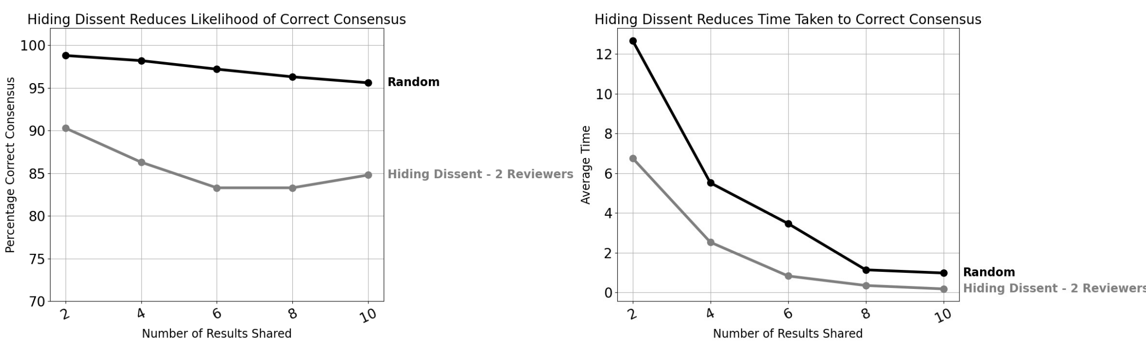


Figure 1: This figure demonstrates that the results are robust under changes in number of tests run each round. As the amount of evidence considered for publication increases (a) Percentage of runs correct consensus is reached and (b) Average time until correct consensus. T = 1000, r = 2.

Figure 2 shows the effect of increasing the size of the community. In this case, whilst the success of both strategies increases, the dynamics of *Random* reaching correct consensus more often, but *Hiding Dissent* reaching correct consensus faster, remain robust. The pattern for each strategy remains the same too. *Random* exhibits a decrease in success as more results are shared. *Hiding*

*Dissent* exhibits a similar decrease, until it reaches consensus within a single round and then the effect reverses. For *Hiding Dissent*, the effect of increased results improving success becomes more prominent with a larger community, because an even greater number of results can be shared in the single round before consensus is reached, further increasing the likelihood of outliers being canceled out and that the total results shared will be more accurate.

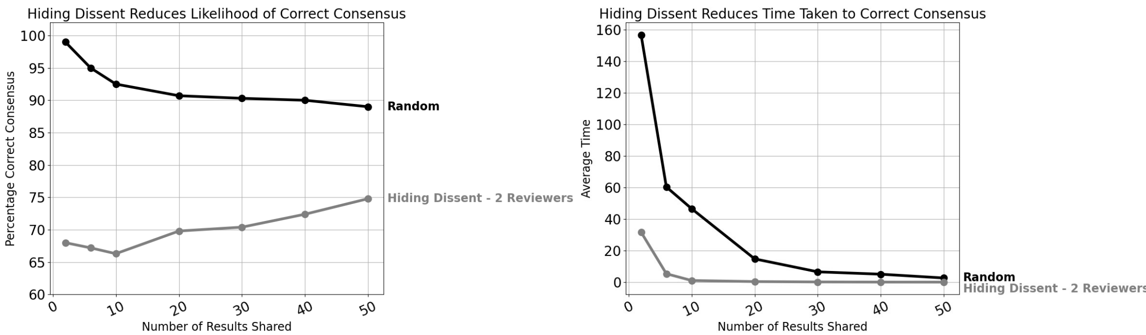


Figure 2: This figure demonstrates that the results are robust under changes in size of network. With a community of size 50, as the amount of evidence considered for publication increases (a) Percentage of runs correct consensus is reached and (b) Average time until correct consensus. T = 100, r = 2.

Figure 3 shows that as the number of biased agents increases, eventually *Hiding Dissent* may do better at protecting the community than *Random*, though both strategies are very unsuccessful in this situation. However, this may just be a modeling artifact. Bandit problems typically struggle to deal with cases where many agents are pulling from different bandit arm distributions with no way of judging which bandit is being pulled from.

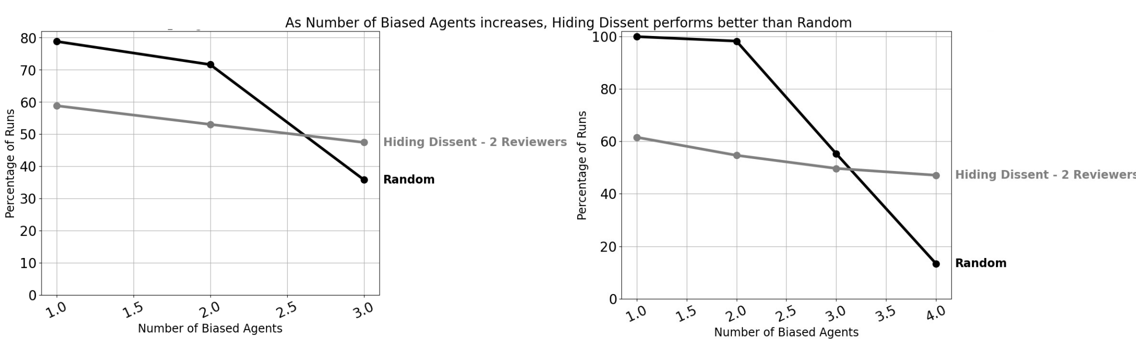


Figure 3: This figure demonstrates that as number of biased agents increases, *Hiding Dissent* may eventually do better than *Random* at protecting the community. Average percentage of runs agents use better arm over 50% of the time during the last 1000 rounds with (a) Myopia. (b) $\varepsilon$-Greedy with $\varepsilon = 0.05$. T = 100, k = 4, r = 4

Finally, I look at altering the initial conditions. As stated in the paper, the model is typically initialized by assigning each agent an $\alpha, \beta \in [0, 4]$ for arm A and then for arm B randomly through a uniform distribution. $\alpha$ tracks the number of successful pulls of that arm, and $\beta$ tracks the number of unsuccessful pulls of that arm. These then determine the shape of two initial beta distributions. Each agent has an expected value for each arm from these beta distributions, given by $\mathbb{E}[X] = \alpha_X/(\alpha_X + \beta_X)$, and thinks the arm with the highest expected value is better.

However, I also test the effect of specific initial conditions. I consider the case where some number of agents are assigned initial $\alpha, \beta \in [0, 4]$ such that their initial expected value of A is greater than their initial expected value for B. The rest of the community get the initial $\alpha, \beta \in [0, 4]$ such that their initial expected value of B is greater than their initial expected value for A. I do this with different numbers in each group.


Figure 4: This figure demonstrates that as the number of agents starting with a preference for arm A increases, both selection strategies typically do worse, though *Random* performs better than *Hiding Dissent*. Percentage of runs correct consensus is reached with (a) Myopia. (b) $\varepsilon$-Greedy with $\varepsilon = 0.05$. N = 10, T = 100, k = 4, r = 2

Figure 4 shows the results for this with both myopic and $\varepsilon$-greedy agents. As expected, across both myopia cases, as well as $\varepsilon$-greedy agents using *Hiding Dissent*, the likelihood of reaching correct consensus decreases as the amount of agents starting with a preference for the the worse arm increases. This is expected because with myopia (and *Hiding Dissent* generally) when more agents start by preferring the worse arm, it becomes less likely agents will test and share evidence for the actually better arm to overcome the discrepancy, so they lock into the worse arm. As previously, it is impossible for $\varepsilon$-greedy agents using *Random* to lock into the worse arm, no matter the starting proportion of agents preferring the worse arm, as they will always eventually experiment and test the actually better arm, sharing those results.

However, even when every agent starts with a preference for the worse theory, so is at a state of incorrect consensus, it is possible for the community to still come to the correct consensus. This occurs far more often with *Random* than with *Hiding Dissent*. The reason for this is because with *Random* it is still possible for agents to receive evidence showing arm A is less successful than they currently think it is, and if some have high enough expectations of arm B they may eventually believe arm B is better and start testing it. In contrast, with *Hiding Dissent* that evidence is much more likely to be hidden so they will never see it and continue preferring arm A.

## 2 Manipulated variables and functions

| Variable | Description | Mathematical Description |
|---|---|---|
| N | Number of Agents | $N \in \mathbb{N}$ |
| T | Number of Arm Pulls | $T \in \mathbb{N}$ |
| k | Number of papers considered for publication | $k \in (0, N]$ & $k \in \mathbb{N}$ |
| r | Number of agents chosen to review each paper | $r \in [0, N)$ & $r \in \mathbb{N}$ |
| $\varepsilon$ | Likelihood of exploring (pull less preferred arm with probability $\varepsilon$) | $\varepsilon \in [0, 0.5]$ |
| $t$ | Tolerance for Dissent (How far from a scientist's own expected value another scientist's evidence needs to be to be considered dissenting.) | $t \in [0, 0.5)$ |
| $\alpha_{A,B}$ | Number of successes of arm A (or B) for an agent | Starting value for each arm determined from a random draw of a Uniform Distribution $X \sim U(a, b)$ |
| $\beta_{A,B}$ | Number of failures of arm A (or B) for an agent | Starting value for each arm determined from a random draw of a Uniform Distribution $X \sim U(a, b)$ |
| $\mathbb{E}[A]$ | Expected success of Arm A for an agent | $\mathbb{E}[A]$ is a function such that $\mathbb{E}[A] : \mathbb{R} \times \mathbb{R} \to [0, 1]$ and $\mathbb{E}[A] = \alpha_A/(\alpha_A + \beta_A)$ |

| | | |
|---|---|---|
| $\mathbb{E}[B]$ | Expected success of Arm B for an agent | $\mathbb{E}[B]$ is a function such that $\mathbb{E}[B] : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ and $\mathbb{E}[B] = \alpha_B/(\alpha_B + \beta_B)$ |

# 3 Pseudocode

I have included Pseudocode for the following Algorithms

- Determining initial beta distributions
- Choosing which arm to pull, and how the pulling works
- How publications are selected for sharing under *Random* and *Hiding Dissent*
- Belief updating at the end of each round

# Pseudocode for Algorithms

| | |
|---|---|
| N | Number of agents |
| Pa | Probability of Success for Arm A |
| Pb | Probability of Success for Arm B |
| T | Number of Tests each agent performs |
| ε | Probability of exploring |
| Rand([a,b]) | A function generating a random number uniformly between a and b |
| k | Number of Results Considered for Publication |
| r | Number of Reviewers |

## #Initial Beliefs

```
FOR I = 1 to N do:
        i.αa = Rand([a,b])
        i.βa = Rand([a,b])
        i.αb = Rand([a,b])
        i.βb = Rand([a,b])
        i.E[A]= i.αa/(i.αa+i.βa)                    Expectation of arm A for agent i
        i.E[B]= i.αb/(i.αb+i.βb)                    Expectation of arm B for agent i
```

## #Algorithm for testing an arm

```
FOR I = 1 to N do:
        i.Sa = 0                                     Remove successes and failures of Arm A
        i.Fa = 0                                     and B from previous round
        i.Sb = 0
        i.Fb = 0
        IF i.E[A]=> i.E[B]do:                        If agent i has a greater expectation of A to B
                Rand([0,1]):
                IF Rand([0,1]) > ε do:               Test arm A with probability (1-ε)
                        counter = 0
                        FOR l = 1 to T:
                                Rand([0,1])          Pull arm A
                        IF Rand([0,1]) < Pa
                                        counter = counter + 1
                                END IF
                        END FOR
                        i.Sa = Counter               Success of Arm A for i this round
                        i.Fa = T - Counter           Failure of Arm A for i this round
```

```
                i.Pa = i.Sa/T                           Percentage Success of Arm A
                i.TestedA = TRUE
                i.TestedB = FALSE
        ELIF Rand([0,1]) < ε do:                        Test arm B with probability ε
                counter = 0
                FOR l = 1 to T:
                        Rand([0,1])                     Pull arm B
                IF Rand([0,1]) < Pb
                                counter = counter + 1
                        END IF
                END FOR
                i.Sb = Counter                          Success of Arm B for i this round
                i.Fb = T - Counter                      Failure of Arm B for i this round
                i.Pb = i.Sb/T                           Percentage Success of Arm B
                i.TestedB = TRUE
                i.TestedA = FALSE
        END IF

IF i.E[A]< i.E[B]do:                                    If agent i has a greater expectation of B to A
        Rand([0,1]):
        IF Rand([0,1]) < ε do:                          Test arm A with probability (ε)
                counter = 0
                FOR l = 1 to T:
                        Rand([0,1])                     Pull arm A
                IF Rand([0,1]) < Pa
                                counter = counter + 1
                        END IF
                END FOR
                i.Sa = Counter                          Success of Arm A for i this round
                i.Fa = T - Counter                      Failure of Arm A for i this round
                i.Pa = i.Sa/T                           Percentage Success of Arm A
                i.TestedA = TRUE
                i.TestedB = FALSE
        ELIF Rand([0,1]) >  ε do:                       Test arm B with probability (1-ε)
                counter = 0
                FOR l = 1 to T:
                        Rand([0,1])                     Pull arm B
                IF Rand([0,1]) < Pb
                                counter = counter + 1
                        END IF
                END FOR
```

```
                i.Sb = Counter                          Success of Arm B for i this round
                i.Fb = T - Counter                      Failure of Arm B for i this round
                i.Pb = i.Sb/T                           Percentage Success of Arm B
                i.TestedB = TRUE
                i.TestedA = FALSE
            END IF
        END IF
    END IF
```

## #Algorithm for selecting publications for sharing

```
IF Random = True do:                                            If Publishing Strategy is Random
    SelectedPapers = random.sample(N, k)                        Randomly select k papers for
                                                                publication from total papers
END IF
ELIF HidingDissent = True do:                                   If Publishing Strategy is Hiding
                                                                Dissent
    SelectedPapers = random.sample(N, k)                        Randomly select k papers for
                                                                publication from total papers
    FOR i in SelectedPapers do:                                 Generate Reviewers for selected
                                                                papers
        ListWithoutSelected = list(range(1, N))

        ListWithoutSelected.remove(i)
        i.r = random.sample(ListWithoutSelected, r)             Select r reviewers for agent i
    END FOR

    FOR i in SelectedPapers do:                                 Count how many reviewers think
                                                                it is dissenting

        DissentCounter = 0
        IF i.TestedA = TRUE do:                                 If agent i tested Arm A
            FOR l in i.r do:
                IF l.Ba >= l.Bb do:                             If the Reviewer prefers Arm A to B
                    IF i.Pa < l.Bb do:                          If Reviewer has a higher
                                                                expectation of Arm B, than agents
                                                                success for Arm A
                            DissentCounter = =
                            DissentCounter + 1

                ELIF l.Ba < l.Bb do:                            If the Reviewer prefers Arm B to A
                    IF i.Pa > l.Bb do:                          If Reviewer has a lower
                                                                expectation of Arm B, than agents
                                                                success for Arm A
```

```
                                        DissentCounter = =
                                        DissentCounter + 1

                              END IF
                         END IF
                    END FOR
               ELIF i.TestedB = TRUE do:                      If agent i tested Arm B
                    FOR l in i.r do:
                         IF l.Ba >= l.Bb do:                  If the Reviewer prefers Arm A to B
                              IF i.Pb > l.Ba do:              If Reviewer has a lower
                                                              expectation of Arm A, than agents
                                                              success for Arm B
                                        DissentCounter = =
                                        DissentCounter + 1

                         ELIF l.Ba < l.Bb do:                 If the Reviewer prefers Arm B to A
                              IF i.Pb < l.Ba do:              If Reviewer has a higher
                                                              expectation of Arm A, than agents
                                                              success for Arm B
                                        DissentCounter = =
                                        DissentCounter + 1

                              END IF
                         END IF
                    END FOR
               END IF

               IF DissentCounter == r do:                     Remove any papers where all
                                                              reviewers think it is dissenting
                    SelectedPapers.remove(i)
               END IF
          END FOR
END IF
```

#Algorithm for Updating Beliefs

```
FOR i = 1 to N do:
     i.αa = i.αa + i.Sa                                       Add successes of pulling arm A to
                                                              previous successes of Arm A
     i.βa = i.αa + i.Fa                                       Add failures of pulling arm A to
                                                              previous failures of Arm A
     i.αb = i.αb + i.Sb                                       Add successes of pulling arm B to
                                                              previous successes of Arm B
```

```
    i.βb = i.αb + i.Fb                          Add failures of pulling arm B to
                                                previous failures of Arm B
    FOR l in SelectedPapers do:                 Update on Shared Results
        IF l != i do:
            i.αa = i.αa + l.Sa
            i.βa = i.αa + l.Fa
            i.αb = i.αb + l.Sb
            i.βb = i.αb + l.Fb
        END IF
    END FOR
    i.E[A]= i.αa/(i.αa+i.βa)                    New Expectation of arm A for
                                                agent i
    i.E[B]= i.αb/(i.αb+i.βb)                    New Expectation of arm B for
                                                agent i
END FOR
```