

Peer Oversight in Collective Decision Making

Sarah Mohsen^a, Pavel Naumov^a

^a*School of Electronics and Computer Science
University of Southampton, United Kingdom*

Abstract

This article introduces peer k -oversight, a property of sequential collective decision mechanisms requiring at least k agents to be responsible for every harmful outcome. It is shown that whenever k -oversight can be achieved by redistributing control over the decisions in a mechanism, it can be achieved using just k agents. A polynomial-time algorithm is also presented that determines whether such a redistribution exists and, when it does, constructs one. These results establish peer oversight as a tractable design principle for multiagent decision-making systems.

Keywords: counterfactual responsibility, responsibility gap, mechanism design, multiagent systems, collective decision making

1. Introduction

In this article, we propose and study *peer oversight*, a formally defined concept of shared responsibility for a harmful outcome of a collective decision. We start with a discussion of the notion of responsibility.

1.1. Responsibility and Strategic Abilities

Consider a variation of Joe Halpern’s fish and polluting factories example (2016, p. 40, Example 2.3.8). In our version of this example, set in the last century, there are three lakes (A, B, and C) and three factories (F1, F2, F3) located at the shores of these lakes, see Figure 1a. Each factory has accumulated a pollutant and must dump it into one of the two lakes it is connected to in the figure. For instance, factory F1 can dump either into Lake A or Lake B. If any two factories dump the pollutant into the same lake, the fish in that lake die. If all three of them dump the pollutant into different lakes, the fish survive.

By a *decision mechanism*, we mean the description of the process used to make a collective decision. First, let us consider a *concurrent* decision mechanism under which each factory chooses its action (the lake where it will dump the pollutant) independently and without knowing the actions chosen by the other factories. This mechanism is used in strategic (or normal form) games.

Suppose that factories F1 and F2 dump the pollutant into Lake B and the fish in Lake B die. Who is responsible for this? Responsibility is a vague concept that can be formally defined in many different ways. The most popular (Widerker and McKenna, 2003) of them is based on Frankfurt’s (1969) principle of alternative possibilities¹: *... a person is morally responsible for what he has done only if he could have done otherwise*. In the literature, “could have done otherwise” has been interpreted as having a strategy to avoid the harmful outcome no matter what the other agents do (Bulling and Dastani, 2013; Yazdanpanah, Dastani, Jamroga, Alechina and Logan, 2019; Naumov and Tao, 2020; Baier, Funke and Majumdar, 2021; Shi, 2024; Shi and Naumov, 2025, 2026).

Email addresses: szrm1g23@soton.ac.uk (Sarah Mohsen), p.naumov@soton.ac.uk (Pavel Naumov)

¹It is worth noting that (Frankfurt, 1969) discusses limitations of this definition and constructs examples when the definition fails to capture the intuitive notion of moral responsibility.

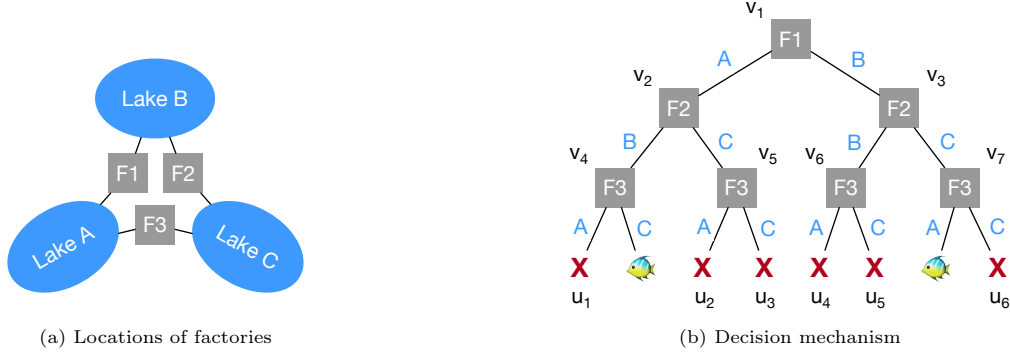


Figure 1: Three factories example.

In this article, we refer to the responsibility defined through this principle as counterfactual responsibility or just *responsibility*. Chockler and Halpern (2004) defined a degree of responsibility in causal models using a similar counterfactual approach.

When applying the definition of counterfactual responsibility to our setting, it is important to specify what we mean by the “harmful outcome”. If the harmful outcome consists of the death of the fish specifically in Lake B, then factory F1 had an individual strategy to avoid such an outcome by dumping the pollutant into Lake A. Similarly, factory F2 could avoid the death of the fish in Lake B by dumping the pollutant into Lake C. Both of these strategies avoid the death of the fish in Lake B by endangering the life of the fish at another lake. To avoid such trolley-like dilemmas, in this article, by “harm” we mean the death of the fish in any of the three lakes. In other words, a strategy that avoids the harmful outcome must guarantee the survival of the fish in all three lakes.

It is easy to see that none of the three factories has a strategy that would guarantee the survival of the fish in all three lakes. Indeed, note, for example, that no matter what action factory F1 chooses, factories F2 and F3 can choose to dump their portions of the pollutant into Lake C and, thus, kill the fish in that lake. Hence, in our example, none of the three factories is responsible for the harmful outcome. Moreover, in this setting, no matter what action either of the factories chooses, none of them is ever responsible for the harmful outcome. If one is aiming to enforce individual responsibility in the fish and three factories setting, then a concurrent decision mechanism is probably the worst option to use!

Let us now consider, for the same setting, a *sequential* decision mechanism under which the factories choose their actions in the order F1, F2, and F3. We visualise this mechanism as a tree depicted in Figure 1b. The labels on the edges denote the lakes that the acting agents decided to dump the pollutant in. The leaf nodes of this tree represent outcomes. We label the harmful outcomes u_1 through u_6 with the red letter X and the rest with the fish emoji.

Consider the *decision path* v_1, v_2, v_4, u_1 in Figure 1b. It corresponds to the scenario under which, first, factory F1 dumps the pollutant into Lake A; then, factory F2 dumps into Lake B; finally, F3 dumps into Lake A. Because two factories dumped into Lake A, the fish in this lake die. Note that factory F3 did not have a strategy to save the fish in all three lakes *upfront*, but it has such a strategy (dump into Lake C) at node v_4 along the decision path. Thus, factory F3 is (counterfactually) responsible for the death of the fish along the path v_1, v_2, v_4, u_1 . It is easy to see that at any node along this path neither factory F1 nor factory F2 has a strategy to save the fish. Hence, neither of them is responsible along this path. Similarly, factory F3 is the only factory responsible along the decision path from node v_1 to node u_6 .

Next, let us consider the decision path from node v_1 to node u_2 . Neither of the three agents has an individual strategy to avoid harm at any of the nodes along this path. Thus, neither of them is (counterfactually) responsible along the path for the death of the fish. We say that this path belongs to the *responsibility gap* of the decision mechanism in Figure 1b. In the literature, the responsibility gap is also sometimes referred to as a responsibility void. One might argue that factory F2 should be blamed along the path from node v_1 to node u_2 because, at v_2 , it has chosen the action (dump into Lake C) that made

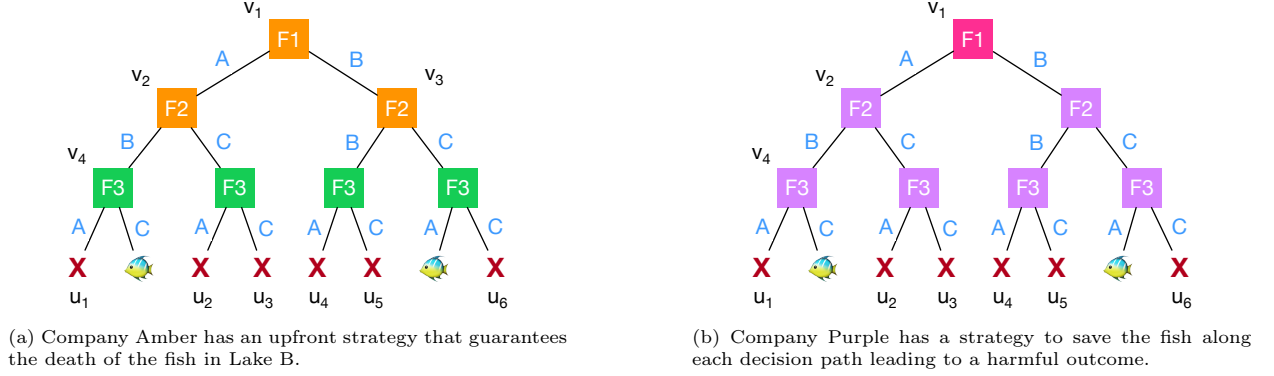


Figure 2: Three factories example.

the death of the fish in at least one of the lakes unavoidable. This argument captures a different form of responsibility that we discuss in Section 5.

The counterfactual responsibility gap can often be eliminated by combining multiple parties into a single agency. To illustrate this, let us first suppose factories F1 and F2 are controlled by the management of company Amber and factory F3 is controlled by company Green, see Figure 2a. By combining the control over factories F1 and F2 in the hands of company Amber, we endow the company with strategic abilities that neither of the two factories has alone. For example, company Amber has an *upfront* strategy to guarantee the death of the fish in Lake B. The strategy is to dump the pollutant into Lake B at nodes v_1 and v_3 , see Figure 2a. However, under the distribution of the control over the factories between companies Amber and Green that we consider, neither of the companies has a strategy to avoid the harm along the decision paths to outcomes u_2 , u_3 , u_4 , and u_5 . Thus, combining the control over the factories F1 and F2 in the hands of company Amber does not eliminate the responsibility gap.

The situation is different in Figure 2b, where company Pink controls factory F1 and company Purple controls factories F2 and F3. In this setting, company Purple has an upfront strategy to avoid harm no matter what company Pink does. Indeed, if company Pink instructs factory F1 to dump the pollutant into Lake A, then company Purple should instruct factories F2 and F3 to dump into lakes B and C, respectively. If company Pink instructs factory F1 to dump the pollutant into Lake B, then company Purple should instruct factories F2 and F3 to dump into lakes C and A, respectively. In either case, the fish in all three lakes survive. This upfront strategy is the strategy at the root node v_1 of the tree. Thus, company Purple has a strategy to avoid harm at some node along each decision path leading to a harmful outcome. Hence, company Purple is responsible along each such path. Therefore, the responsibility gap of the decision mechanism depicted in Figure 2b is empty. We say that this mechanism is *gap-free*.

Together, Figures 1b, 2a, and 2b show that a well-designed aggregation of the control over individual decisions in the hands of larger agencies can potentially eliminate the responsibility gap of the decision mechanism.

1.2. Peer Oversight

From the last century example of Section 1.1, let us now transition to the present day, where the industrial area grew to a six-factory complex, depicted in Figure 3a. All six factories are owned by the same multinational company and discharge the pollutant in the sequential order F1 through F6. The company employs a single AI agent (named Red) to decide which lake each factory dumps the pollutant into. Note that each lake is connected to four factories. We assume that, due to the improved environmental standards, *all four* factories must dump the pollutant into a lake in order to kill the fish in that lake.

It is easy to see that agent Red has an upfront strategy to avoid harm by spreading the pollutant evenly between the three lakes. Thus, agent Red is responsible for the harm each time the fish die. Hence, this mechanism is gap-free.

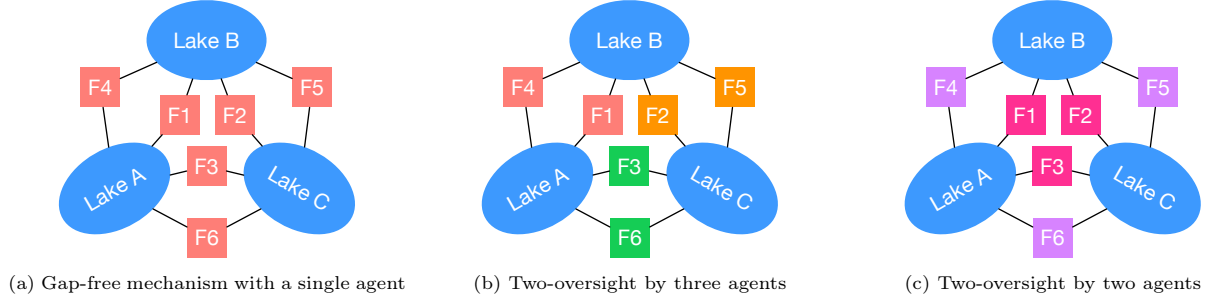


Figure 3: Six factories example.

AI agents have a tendency to be unreliable. To improve *trustworthiness* of the decision-making process, the company decides to replace the single-agent decision-making system with a multi-agent AI system consisting of agents Red, Amber, and Green. The Red agent chooses the lake for factories F1 and F4, the Amber agent chooses for F2 and F5, and the Green agent does this for F3 and F6, see Figure 3b. The company made this change because of the following observation.

Proposition 1. *Under the three-agent arrangement depicted in Figure 3b, if the fish in one of the lakes is killed, then two agents become responsible for the harmful outcome.*

Proof. We split the decision process into two stages. Stage 1 covers the discharge decisions for the first three factories, F1, F2, and F3. Stage 2 covers the decisions for the remaining factories, F4, F5, and F6. Observe that, at the end of Stage 1, one of the following two cases takes place:

Case I: A single portion of the pollutant has been dumped in each of the three lakes. Note that among the remaining factories F4, F5, and F6 only two are connected to each individual lake. Hence, during Stage 2, at most two portions of the pollutant could be added to each lake. Thus, the total amount of the pollutant accumulated in each lake during both stages would not exceed three portions. Therefore, in this case, a harmful outcome is not possible.

Case II: Two portions of the pollutant have been dumped in one lake (call it lake X), one portion dumped in another lake (call it lake Y), and no pollution dumped into the third lake (call it lake Z). Observe, similarly to Case I, that at most two portions of the pollutant could be added to each lake during Stage 2. Thus, the fish in lakes Y and Z are guaranteed to survive no matter where factories F4, F5, and F6 dump the pollutant. Hence, to avoid a harmful outcome, it suffices to save the fish in lake X. Out of factories F4, F5, and F6, two are located at the shore of lake X (no matter which of the three lakes is lake X), see Figure 3b. These two factories are controlled by different AI agents (because all three factories F4, F5, and F6 are controlled by different agents). Thus, two different agents can prevent the harm by instructing their respective factories not to dump the pollutant into lake X on Stage 2. Therefore, these two agents will both be responsible if the harm happens. $\square$

Intuitively, Proposition 1 means that in the setting of Figure 3b, the three AI agents provide a safety net for one another, overseeing one another’s work. We call such an arrangement *peer oversight*. More generally, we say that a decision mechanism ensures *peer k -oversight* (or just *k -oversight*) if, under that mechanism, at least k agents are responsible for each harmful outcome. Gap-free mechanisms, such as in Figure 3a, ensure one-oversight. The mechanism in Figure 3b ensures two-oversight. The concept of *peer k -oversight* establishes an architectural design principle for trustworthy multiagent AI systems. In addition to protecting against error-prone AI agents, *peer two-oversight* also provides a defence mechanism against a single rogue agent, while *peer $(k + 1)$ -oversight* provides a defence mechanism against a collusion of k rogue agents.

As the example in Figure 3b illustrates, for a given k , sometimes the nodes of a decision mechanism can be redistributed between multiple agents to ensure k -oversight. Of course, this is not always possible. For instance, one cannot divide the control over factories in Figure 1a between some number of agents to ensure

two-oversight. In the case when redistribution of control ensuring k -oversight exists, one can naturally ask what the *minimal* number of agents required to ensure k -oversight is. As it turns out, in the case of our example in Figure 3b, two-oversight can be ensured by just two agents. In Figure 3c, we show how this can be done. In this case, agent Pink controls factories F1, F2, and F3, while agent Purple controls factories F4, F5, and F6.

Proposition 2. *Under the two-agent arrangement depicted in Figure 3c, if the fish is killed in one of the lakes, then two agents become responsible for the harmful outcome.*

Proof. Agent Pink has an upfront strategy to avoid the harm by arranging the dumps of factories F1, F2, and F3 in such a way that a single portion of the pollutant is dumped by these three factories into each lake. Note that each lake is connected only to two out of three remaining factories (F4, F5, and F6). Thus, no matter what the remaining three factories do, they can dump at most two portions of the pollutant into each lake. Therefore, after all six dumps, each lake will have at most three portions of the pollutant and the fish in all three lakes will survive.

Agent Purple also has an upfront strategy to avoid the harm by arranging the dumps of factories F4, F5, and F6 in such a way that a single portion of the pollutant is dumped by these three factories into each lake. By the argument similar to the one above, this strategy guarantees survival of the fish in all three lakes.

Note that agent Purple, whose factories dump the pollutant after the factories of agent Pink, also has an *alternative* strategy to guarantee the survival of the fish in all three lakes. The alternative strategy consists in instructing factories F4, F5, and F6 to do the *opposite* of what is done by factories F1, F2, and F3, respectively. For instance, if factory F1 dumps the pollutant into Lake A, then factory F4 should dump its portion of the pollutant into Lake B. Remarkably, this alternative strategy guarantees that only *two* portions of the pollutant are dumped into each lake. $\square$

The above proposition shows that, in our six-factory example, two-oversight can be ensured by just two agents. Of course, two distinct agents cannot be responsible in a single-agent setting. Thus, two is the *minimal* number of agents required to ensure two-oversight in the six-factory example.

1.3. Contribution

In this article, we show that what we observed in the six-factory example is not an exception but a general rule. Specifically, Theorem 1 establishes that, in any decision mechanism ensuring k -oversight, control over the decision nodes can be redistributed among the agents so that the resulting mechanism uses only k agents while preserving k -oversight. Note that if a harmful outcome is reachable, then at least k agents are necessary to achieve k -oversight. Hence, the theorem implies that, in such settings, k is precisely the minimum number of agents required whenever k -oversight is possible.

In addition to Theorem 1, we present a polynomial-time algorithm that check whether a k -oversight mechanism exists. If it exists, the algorithm also constructs a k -oversight mechanism using only k agents.

In Section 5, we discuss an alternative concept of responsibility that combines counterfactual responsibility with responsibility for seeing to it and argue that the same results hold for that form of responsibility as well.

It is worth noting that although the decision mechanisms in both our introductory examples can be represented by a tree (see, for example, Figure 1b), we show our results in a significantly more general setting where the mechanism is a *directed graph* that might contain cycles.

2. Mechanisms, Responsibility, and Peer Oversight

We start with the formal definition of the game-like model that we use to capture the interaction process between agents.

Definition 1. A (decision) *mechanism* is a tuple $M = (V, E, Q, H, A, \{C_a\}_{a \in A})$, where

1. V is a finite set of **nodes** and $E \subseteq V \times V$ is a set of directed **edges**; let $Sink_M$ be the set of nodes without outgoing edges; $V \setminus Sink_M$ is the set of **decision** nodes,
2. $Q \subseteq V$ is a nonempty set of **initial** nodes,
3. $H \subseteq Sink_M$ is a set of **harmful** sink nodes,
4. A is a set of **agents**,
5. $\{C_a\}_{a \in A}$ is a disjoint family of **control** sets of decision nodes such that $\bigcup_{a \in A} C_a \subseteq V \setminus Sink_M$.

We omit the subscript in $Sink_M$ when its value is clear from the context. An example of a mechanism is depicted in Figure 1b. In that example, set V consists of the 15 nodes of the tree. The edges in that tree are assumed to be directed from the parent node to the child node. The set $Sink$ consists of the 8 leaf nodes of the tree. The set H consists of the nodes u_1, u_2, u_3, u_4, u_5 , and u_6 . Set A of agents is $\{F1, F2, F3\}$. Set C_{F1} contains only the root node v_1 , set C_{F2} contains only the nodes v_2 and v_3 , and set C_{F3} contains only nodes v_4, v_5, v_6 , and v_7 . Note again that, unlike the mechanism depicted in Figure 1b, in general, the graph (V, E) might contain cycles.

Definition 2. For a given mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, a **strategy** of an agent $a \in A$ is a function s from C_a to V such that $(u, s(u)) \in E$ for each decision node $u \in C_a$.

An example of a strategy of agent F2 under the mechanism depicted in Figure 1b is a function that maps node v_2 into node v_4 and node v_3 into node v_7 .

By a *decision path* we mean any sequence of nodes $u_1, u_2, \dots, u_n$ such that $n \geq 1$ and $(u_i, u_{i+1}) \in E$ for each $i < n$. Sequence v_2, v_4, u_1 is an example of a decision path for the mechanism depicted in Figure 1b.

Definition 3. For a given mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, a **strategy** of an agent $a \in A$ **avoids harm from a node** $u \in V$ if for each decision path $u_1, \dots, u_n$ from node $u_1 = u$ to a harmful sink node $u_n \in H$ there is $i < n$ such that $u_i \in C_a$ and $u_{i+1} \neq s(u_i)$.

For the example depicted in Figure 1b, any strategy s of agent F3 such that $s(v_4) \neq u_1$ (in other words, at node v_4 , factory F3 dumps the pollutant into Lake C rather than Lake A), avoids harm from node v_4 .

The next three lemmas capture important properties of strategies that avoid harm. These lemmas will be used later in the article.

Lemma 1. For any mechanism, no agent has a strategy that avoids harm from a harmful sink node.

Proof. Towards a contradiction, suppose that agent a has a strategy s that avoids harm from a harmful sink node $u \in H$ under a mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$. Consider the single-element decision path u_1 such that $u_1 = u$. Thus, by Definition 3, there exists $i < 1$ such that $u_i \in C_a$, which is a contradiction because the decision path contains no elements u_i with $i < 1$. $\square$

Lemma 2. For a given mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, if a strategy s of an agent $a \in A$ avoids harm from a node $u \notin C_a$, then strategy s also avoids harm from any node v such that $(u, v) \in E$.

Proof. Consider any decision path $u_1, \dots, u_n$ from node $u_1 = v$ to a harmful sink node $u_n \in H$. By Definition 3, it suffices to prove that there is $i < n$ such that $u_i \in C_a$ and $u_{i+1} \neq s(u_i)$.

Towards this proof, consider the decision path $u_0, u_1, \dots, u_n$, where $u_0 = u$. By Definition 3 and the assumption of the lemma that strategy s avoids harm from node u , there must exist i , such that $0 \leq i < n$, $u_i \in C_a$, and $u_{i+1} \neq s(u_i)$. Observe that $i \neq 0$ by the assumption $u \notin C_a$ of the lemma because $u_0 = u$. $\square$

Although the statement of the next lemma differs from the previous one, the proofs of these two lemmas are surprisingly similar.

Lemma 3. For a given mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, if a strategy s of an agent $a \in A$ avoids harm from a node $u \in C_a$, then strategy s also avoids harm from the node $s(u)$.

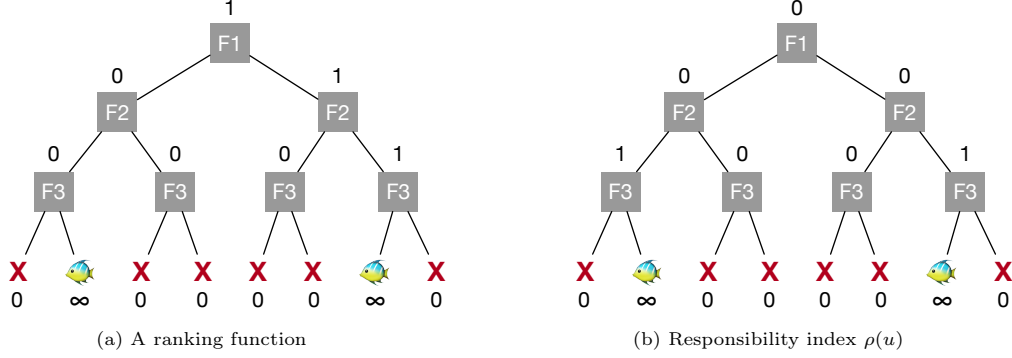


Figure 4: Two examples of ranking functions for the mechanism in Figure 1b.

Proof. Consider any decision path $u_1, \dots, u_n$ from node $u_1 = s(u)$ to a harmful sink node $u_n \in H$. By Definition 3, it suffices to prove that there is $i < n$ such that $u_i \in C_a$ and $u_{i+1} \neq s(u_i)$.

Towards this proof, consider the decision path $u_0, u_1, \dots, u_n$, where $u_0 = u$. By Definition 3 and the assumption of the lemma that strategy s avoids harm from node u , there must exist i , such that $0 \leq i < n$, $u_i \in C_a$, and $u_{i+1} \neq s(u_i)$. Observe that $i \neq 0$ because $u_{0+1} = u_1 = s(u) = s(u_0)$. $\square$

We are now ready to give the core definition of this article, the one that captures the notion of counterfactual responsibility.

Definition 4. An agent is **responsible** along a decision path that terminates at a harmful sink node if the agent has a strategy that avoids harm from at least one node along the path.

For instance, in the setting of Figure 1b, agent F3 is responsible along the path v_2, v_4, u_1 because F3 has a strategy at node v_4 to avoid harm.

Definition 5. For any integer $k \geq 1$, a mechanism ensures (peer) **k -oversight** if there are at least k distinct agents responsible along each decision path from an initial node to a harmful sink node.

Note that the mechanism ensures 1-oversight iff it has no responsibility gap. Thus, for instance, the mechanism depicted in Figure 2b ensures oversight, while the mechanism depicted in Figure 2a does not. Our introduction contains no trees representing the decision mechanism depicted in Figure 3a, Figure 3b, and Figure 3c. This is because these trees contain 2^6 leaf (sink) nodes and, thus, are simply too large to draw. However, as we proved in Proposition 1 and Proposition 2, the last two mechanisms ensure 2-oversight. The single-agent mechanism in Figure 3a only ensures 1-oversight.

Definition 6. Mechanisms $(V, E, Q, H, A, \{C_a\}_{a \in A})$ and $(V', E', Q', H', A', \{C'_a\}_{a \in A'})$ are **structurally equivalent** if $V = V'$, $E = E'$, $Q = Q'$, and $H = H'$.

The mechanisms depicted in Figure 2a and Figure 2b are structurally equivalent. And so are those depicted in Figure 3a, Figure 3b, and Figure 3c.

3. First Result: All That You Need Are k Agents

In this section, we prove our first main result: for any mechanism that ensures k -oversight, there is a structurally equivalent mechanism with just k agents that also ensures k -oversight. In the rest of the article, by *extended* natural numbers we mean the set of non-negative integers extended by the element ∞ . As is common in mathematics, we assume that: (a) ∞ is larger than each of the non-negative integers, (b) $\infty + 1 = \infty$, and (c) the minimal element of the empty set is ∞ .

At the core of the proofs of our two main results is the notion of *ranking function*, defined below.

Definition 7. A **ranking function** φ on a mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$ is an arbitrary function that maps each node $u \in V$ into an extended natural number $\varphi(u)$ such that

1. $\varphi(u) = 0$ for each harmful sink node $u \in H$,
2. $\varphi(u) = \infty$ for each non-harmful sink node $u \in \text{Sink} \setminus H$,
3. for all nodes $u, v \in V$, if $(u, v) \in E$, then $\varphi(u) \leq \varphi(v) + 1$,
4. for each decision node $u \in V \setminus \text{Sink}$, there is a node $v \in V$ such that $(u, v) \in E$ and $\varphi(u) \leq \varphi(v)$.

Figure 4a and Figure 4b show two examples of ranking functions for the mechanism shown in Figure 1b. In these figures, the value $\varphi(u)$ of a ranking function for a node u is displayed next to the node.

Lemma 4. Any two structurally equivalent mechanisms have the same set of ranking functions.

Proof. The statement of the lemma follows from Definition 6 and Definition 7 because the latter definition does not refer to the family of control sets $\{C_a\}_{a \in A}$. $\square$

Definition 8. For a given mechanism, the **responsibility count** of a decision path that terminates at a harmful sink is the number of agents responsible along the path.

For the mechanism depicted in Figure 1b, the responsibility count of the decision path v_2, v_4, u_1 is 1. This is because the only agent responsible along this path is agent F3 (it has a strategy to avoid harm at node v_4).

Definition 9. For a given mechanism, the **responsibility index** $\rho(u)$ of a node u is the minimum responsibility count among all decision paths from u to a harmful sink node.

Of course, there might be no decision paths from node u to a harmful sink node. In this case, $\rho(u) = \infty$ because of our assumption in the preamble to this section that the minimal element of the empty set is ∞ . Figure 4b shows the index $\rho(u)$ for each node u of the mechanism from Figure 1b. As we stated earlier, the function ρ in Figure 4b is a ranking function. This is not a coincidence. As we show in Lemma 9, the responsibility index of any decision mechanism is a ranking function. The four lemmas below prove each of the four properties of a ranking function from Definition 7 separately.

Lemma 5. $\rho(u) = 0$ for each harmful sink node $u \in H$ of a mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$.

Proof. Consider the single-element decision path u . By Lemma 1, no agent can have a strategy to avoid harm from any of the nodes along this path. Hence, by Definition 4, no agent is responsible along this path. Thus, by Definition 8, the responsibility count of this path is zero. Therefore, $\rho(u) = 0$ by Definition 9. $\square$

Lemma 6. $\rho(u) = \infty$ for each non-harmful sink node $u \in \text{Sink} \setminus H$ of a mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$.

Proof. The assumption $u \in \text{Sink} \setminus H$ implies that there are no paths from u to a harmful sink. Therefore, $\rho(u) = \min \emptyset = \infty$, by Definition 9. $\square$

Lemma 7. For any nodes $u, v \in V$ of a decision mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, if $(u, v) \in E$, then $\rho(u) \leq \rho(v) + 1$.

Proof. We consider the following two cases separately.

Case I: there is no decision path from node v to a harmful sink node. Thus, $\rho(v) = \min \emptyset = \infty$ by Definition 9. Therefore, $\rho(u) \leq \infty = \infty + 1 = \rho(v) + 1$.

Case II: there is at least one decision path from node v to a harmful sink node. Consider a decision path $u_1, u_2, \dots, u_n$ from node $u_1 = v$ to a harmful sink node $u_n \in H$ with minimal responsibility count. Such a decision path does not have to be unique. Let k be the responsibility count of this decision path. Thus, $\rho(v) = k$ by Definition 9. By Lemma 2, the responsibility count of the path $u, u_1, \dots, u_n$ is at most $k + 1$. Thus, $\rho(u) \leq k + 1$ by Definition 9. Therefore, $\rho(u) \leq \rho(v) + 1$. $\square$

Lemma 8. For any non-sink node $u \in V \setminus \text{Sink}$ of a decision mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, there is node $v \in V$ such that $(u, v) \in E$ and $\rho(u) \leq \rho(v)$.

Proof. We start the proof with the following claim:

Claim 1. *There is a node $v \in V$ such that $(u, v) \in E$ and any agent that has a strategy to avoid harm from node u also has a strategy to avoid harm from v .*

Proof of Claim. We consider the following two cases separately:

Case 1: There is no agent $a \in A$ such that $u \in C_a$ and agent a has a strategy to avoid harm from node u . The assumption $u \in V \setminus \text{Sink}$ implies that there is at least one node $v \in V$ such that $(u, v) \in E$. Then, the statement of the claim follows from Lemma 2.

Case 2: There is at least one agent $a \in A$ such that $u \in C_a$ and agent a has a strategy to avoid harm from node u . There could be only one such agent a because the family $\{C_a\}_{a \in A}$ is disjoint by item 5 of Definition 1. Call this agent a_0 and the strategy s . Hence, by Lemma 3, agent a_0 has a strategy to avoid harm from node $v = s(u)$. Furthermore, by Lemma 2, any agent $a \neq a_0$ that has a strategy to avoid harm from node u also has a strategy to avoid harm from node v . $\square$

To finish the proof of the lemma, suppose that v is the node whose existence is proven in the above claim. If there is no decision path from node v to a harmful sink, then $\rho(v) = \infty$ by Definition 9. Therefore, $\rho(u) \leq \rho(v)$.

Assume now that there is at least one decision path from v to a harmful sink node. Let $u_1, \dots, u_n$, where $u_1 = v$ and $u_n \in H$, be such a path with the minimal responsibility count. Let k be the responsibility count of this decision path. Thus, $\rho(v) = k$ by Definition 9. By the above claim, Definition 4, and Definition 8, the responsibility count of the path $u, u_1, \dots, u_n$ is also k . Therefore, $\rho(u) \leq k = \rho(v)$ by Definition 9. This concludes the proof of the lemma. $\square$

The next lemma follows from Definition 7 and the four previous lemmas.

Lemma 9. *Responsibility index on an arbitrary mechanism is a ranking function.*

The proof of the next lemma contains the main construction of this article. It shows how to build a mechanism with just k agents that ensures k -oversight.

Lemma 10. *For any ranking function φ on a mechanism $M = (V, E, Q, H, A, \{C_a\}_{a \in A})$ and any integer $k \geq 1$, if $\varphi(q) \geq k$ for each initial node $q \in Q$, then there is a mechanism structurally equivalent to M that has only k agents and ensures k -oversight.*

Proof. Let $A' = \{1, \dots, k\}$ and, for each agent $a \in A'$,

$$C'_a = \{u \in V \mid \varphi(u) = a\}. \quad (1)$$

It suffices to show that the mechanism $(V, E, Q, H, A', \{C'_a\}_{a \in A'})$ ensures k -oversight. By Definition 5 and Definition 4, it is enough to show that each agent $a \in A'$ has a strategy to avoid harm from each initial node $q \in Q$.

Consider an arbitrary agent $a \in A'$. For each node $u \in C'_a$, define $s(u)$ to be any node v such that $(u, v) \in E$ and $\varphi(u) \leq \varphi(v)$. Such a node v exists by item 4 of Definition 7. Then, for each $u \in C'_a$,

$$\varphi(u) \leq \varphi(s(u)). \quad (2)$$

To finish the proof of the lemma, it suffices to establish that strategy s of agent a avoids harm from each initial node.

Consider any decision path $u_1, \dots, u_n$ from an initial node $u_1 \in Q$ to a harmful sink $u_n \in H$. By Definition 3, it is sufficient to prove that there is $i < n$ such that $u_i \in C'_a$ and $u_{i+1} \neq s(u_i)$. Indeed, $\varphi(u_1) \geq k$ by the assumption of the lemma. Hence, $\varphi(u_1) \geq a$ because $a \in A' = \{1, \dots, k\}$. At the same time, $\varphi(u_n) = 0$ by item 1 of Definition 7. Thus, by item 3 of Definition 7, there must exist $i < n$ such that $\varphi(u_i) = a$ and $\varphi(u_{i+1}) = a - 1$. Hence, $u_i \in C'_a$ by statement (1). Also, $u_{i+1} \neq s(u_i)$ by statement (2). $\square$

The next theorem states the first main result of this article.

Theorem 1. *For any mechanism that ensures k -oversight, there is a structurally equivalent mechanism with just k agents that also ensures k -oversight.*

Proof. If a mechanism ensures k -oversight, then $\rho(u) \geq k$ for each initial node u of the mechanism by Definition 5, Definition 4, and Definition 9. Then, the statement of the theorem follows from Lemma 9 and Lemma 10. $\square$

4. Second Result: Mechanism Ensuring k -oversight is Feasibly Computable

In this section, we present our second result: a polynomial-time algorithm that, for any given mechanism, either constructs a structurally equivalent mechanism ensuring k -oversight or determines that such a mechanism does not exist. To construct such an algorithm, let us first introduce the notion of a closure X^+ of a set of nodes X . Intuitively, X^+ is the set of nodes from which the mechanism will inevitably reach a node in the set X . For the mechanism depicted in Figure 1b, if $X = \{v_4, u_2, u_3, v_3\}$, then $X^+ = X \cup \{v_1, v_2, v_5\}$. Formally, the set X^+ is defined recursively:

Definition 10. *For any set $X \subseteq V$ of nodes of a mechanism $M = (V, E, Q, H, A, \{C_a\}_{a \in A})$, let X_M^+ be the minimal set of nodes such that*

1. $X \subseteq X_M^+$,
2. if $u \notin \text{Sink}$ and $\{v \in V \mid (u, v) \in E\} \subseteq X_M^+$, then $u \in X_M^+$.

We omit the subscript in the expression X_M^+ when its value is clear from the context. The next two lemmas follow from the above definition.

Lemma 11. *For any mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, sink node $u \in \text{Sink}$, and set $X \subseteq V$ of nodes, if $u \in X^+$, then $u \in X$.*

Lemma 12. *For any mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, decision node $u \in V \setminus \text{Sink}$, and set $X \subseteq V$ of nodes such that $u \notin X^+$, there is a node $v \in V$ such that $(u, v) \in E$ and $v \notin X^+$.*

The next lemma captures a connection between the closure operation X^+ and ranking functions.

Lemma 13. *For any mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, ranking function φ , set $X \subseteq V$ of nodes, and node $u \in X^+$,*

$$\varphi(u) \leq \max_{v \in X} \varphi(v).$$

Proof. We prove the statement of the lemma by induction on the recursive construction of the set X^+ specified in Definition 10.

Base Case: $u \in X$. Then, $\varphi(u) \leq \max_{v \in X} \varphi(v)$.

Induction Step: $u \notin \text{Sink}$ and $\{v \in V \mid (u, v) \in E\} \subseteq X^+$. Then, $\varphi(v) \leq \max_{w \in X} \varphi(w)$ for each $v \in V$ such that $(u, v) \in E$ by the induction hypothesis. At the same time, by item 4 of Definition 7, the assumption $u \notin \text{Sink}$ implies that there is a node $v_0 \in V$ such that $(u, v_0) \in E$ and $\varphi(u) \leq \varphi(v_0)$. Therefore, $\varphi(u) \leq \varphi(v_0) \leq \max_{w \in X} \varphi(w)$. $\square$

Definition 11. *For any mechanism $M = (V, E, Q, H, A, \{C_a\}_{a \in A})$ and set $X \subseteq V$ of nodes, let $tl_M(X)$ be the set of all nodes $u \in V$ for which there is a node $v \in X$ such that $(u, v) \in E$.*

We omit the subscript in the expression $tl_M(X)$ when its value is clear from the context. In the setting of Figure 1b, we have $tl(\{u_1, v_4\}) = \{v_4, v_2\}$. The next lemma follows from the above definition.

Lemma 14. *For any mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, sink node $u \in \text{Sink}$, and set $X \subseteq V$ of nodes, $u \notin tl(X)$.*

Definition 12. For any mechanism $M = (V, E, Q, H, A, \{C_a\}_{a \in A})$, let sequence $\mathbb{X}_0^M, \mathbb{X}_1^M, \dots, \mathbb{X}_\infty^M$ of sets of nodes be defined recursively as follows:

$$\mathbb{X}_i^M = \begin{cases} H^+, & \text{if } i = 0, \\ (\mathbb{X}_{i-1}^M \cup tl(\mathbb{X}_{i-1}^M))^+, & \text{if } 0 < i < \infty, \\ V, & \text{if } i = \infty. \end{cases}$$

We omit the superscript in the expression $\mathbb{X}_i^M$ when its value is clear from the context. For the example depicted in Figure 1b,

$$\begin{aligned} \mathbb{X}_0 &= H^+ = \{u_1, u_2, u_3, u_4, u_5, u_6, v_5, v_6\}, \\ \mathbb{X}_1 &= (\mathbb{X}_0 \cup tl(\mathbb{X}_0))^+ = (\{u_1, u_2, u_3, u_4, u_5, u_6, v_5, v_6\} \cup \{v_2, v_3, v_4, v_5, v_6, v_7\})^+ \\ &= \{u_1, u_2, u_3, u_4, u_5, u_6, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}, \\ \mathbb{X}_2 &= (\mathbb{X}_1 \cup tl(\mathbb{X}_1))^+ = \mathbb{X}_1, \\ \mathbb{X}_3 &= (\mathbb{X}_2 \cup tl(\mathbb{X}_2))^+ = \mathbb{X}_1, \\ &\dots \end{aligned}$$

and $\mathbb{X}_\infty$ is the set of all nodes of the tree, including the two unlabelled leaf nodes.

Lemma 15. $\mathbb{X}_0 \subseteq \mathbb{X}_1 \subseteq \mathbb{X}_2 \subseteq \dots \subseteq \mathbb{X}_\infty$ for any mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$.

Proof. By Definition 10, if $X \subseteq Y$, then $X^+ \subseteq Y^+$ for all sets $X, Y \subseteq V$. By Definition 11, if $X \subseteq Y$, then $tl(X) \subseteq tl(Y)$ for all sets $X, Y \subseteq V$. Therefore, the statement of the lemma follows from Definition 12. $\square$

The next lemma connects the chain of sets $\mathbb{X}_0 \subseteq \mathbb{X}_1 \subseteq \dots$ with an *arbitrary* ranking function.

Lemma 16. $\varphi(u) \leq k$, for any mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, ranking function φ , extended natural number k such that $0 \leq k \leq \infty$, and node $u \in \mathbb{X}_k$.

Proof. We prove the statement of the lemma by induction on k .

Base Case: $k = 0$. Then, $u \in H^+$ by the assumption $u \in \mathbb{X}_k$ and Definition 12. Hence, $\varphi(u) \leq \max_{v \in H} \varphi(v)$ by Lemma 13. Thus, $\varphi(u) \leq 0$ by item 1 of Definition 7.

Induction Step: $0 < k < \infty$. Then, $u \in (\mathbb{X}_{k-1} \cup tl(\mathbb{X}_{k-1}))^+$ by the assumption $u \in \mathbb{X}_k$ and Definition 12. Hence, by Lemma 13,

$$\varphi(u) \leq \max\{\varphi(v) \mid v \in \mathbb{X}_{k-1} \cup tl(\mathbb{X}_{k-1})\} = \max(\max\{\varphi(v) \mid v \in \mathbb{X}_{k-1}\}, \max\{\varphi(v) \mid v \in tl(\mathbb{X}_{k-1})\}).$$

Thus, $\varphi(u) \leq \max(k-1, \max\{\varphi(v) \mid v \in tl(\mathbb{X}_{k-1})\})$ by the induction hypothesis. Then, by Definition 11,

$$\varphi(u) \leq \max(k-1, \max\{\varphi(v) \mid (v, w) \in E, w \in \mathbb{X}_{k-1}\}).$$

Hence, $\varphi(u) \leq \max(k-1, \max\{\varphi(v) \mid (v, w) \in E, \varphi(w) \leq k-1\})$ again by the induction hypothesis. Thus, $\varphi(u) \leq \max(k-1, (k-1)+1)$ by item 3 of Definition 7. Therefore, $\varphi(u) \leq k$.

Infinity Case: $k = \infty$. Note that $\varphi(u) \leq \infty$. Thus, $\varphi(u) \leq k$ because $k = \infty$. $\square$

Definition 13. For any mechanism $M = (V, E, Q, H, A, \{C_a\}_{a \in A})$, the **irrecoverability index** $\iota_M(u)$ of a node $u \in V$ is the minimal i such that $u \in \mathbb{X}_i$.

Note that the irrecoverability index is well-defined because $\mathbb{X}_\infty = V$ by Definition 12. We omit the subscript in the expression $\iota_M(u)$ when its value is clear from the context. Figure 5 shows the irrecoverability index for each node of the mechanism depicted in Figure 1b. It is easy to see that, in this example, the irrecoverability index is a ranking function. This is also not a coincidence. As we prove in Lemma 21, the irrecoverability index of any mechanism is a ranking function. The four lemmas below prove each of the four properties of a ranking function from Definition 7 separately.

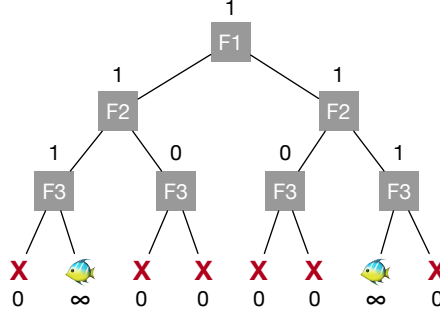


Figure 5: Irrecoverability index $\iota(u)$.

Lemma 17. $\iota(u) = 0$ for each harmful sink node $u \in H$ of a mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$.

Proof. Note that $H \subseteq H^+$ by Definition 10. Hence, $H \subseteq \mathbb{X}_0$ by Definition 12. Therefore, $\iota(u) = 0$ for each node $u \in H$ by Definition 13. $\square$

Lemma 18. $\iota(u) = \infty$ for each non-harmful sink node $u \in \text{Sink} \setminus H$ of a mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$.

Proof. Consider any node $u \in \text{Sink} \setminus H$. By Definition 13, it suffices to show that $u \notin \mathbb{X}_i$ for each $i < \infty$. We prove this by induction on i . For the base case, note that $u \notin H^+$ by Lemma 11 and the assumption $u \in \text{Sink} \setminus H$. Hence, $u \notin \mathbb{X}_0$ by Definition 12.

Consider any $i > 0$ and suppose that $u \notin \mathbb{X}_i$. Thus, $u \notin \mathbb{X}_i \cup \text{tl}(\mathbb{X}_i)$ by Lemma 14 and the assumption $u \in \text{Sink} \setminus H$. Thus, $u \notin (\mathbb{X}_i \cup \text{tl}(\mathbb{X}_i))^+$ by Lemma 11. Therefore, $u \notin \mathbb{X}_{i+1}$ by Definition 12. $\square$

Lemma 19. If $(u, v) \in E$, then $\iota(u) \leq \iota(v) + 1$, for any $u, v \in V$ of a mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$.

Proof. Note that $v \in \mathbb{X}_{\iota(v)}$ by Definition 13. Thus, $u \in \text{tl}(\mathbb{X}_{\iota(v)})$ by the assumption $(u, v) \in E$ and Definition 11. Hence $u \in \mathbb{X}_{\iota(v)} \cup \text{tl}(\mathbb{X}_{\iota(v)})$. Then, $u \in (\mathbb{X}_{\iota(v)} \cup \text{tl}(\mathbb{X}_{\iota(v)}))^+$ by item 1 of Definition 10. Thus, $u \in \mathbb{X}_{\iota(v)+1}$ by Definition 12. Therefore, $\iota(u) \leq \iota(v) + 1$ by Definition 13. $\square$

Lemma 20. For each decision node $u \in V \setminus \text{Sink}$ of a mechanism $(V, E, Q, H, A, \{C_a\}_{a \in A})$, there is node $v \in V$ such that $(u, v) \in E$ and $\iota(u) \leq \iota(v)$.

Proof. We consider the following four cases separately:

Case 1: $\iota(u) = 0$. The assumption $u \in V \setminus \text{Sink}$ implies that there is $v \in V$ such that $(u, v) \in E$. Then, $\iota(u) = 0 \leq \iota(v)$.

Case 2: $\iota(u) = 1$. Thus, $u \notin \mathbb{X}_0$ by Definition 13. Hence, $u \notin H^+$ by Definition 12. Then, by Lemma 12 and the assumption $u \in V \setminus \text{Sink}$ of the lemma, there exists a node $v \in V$ such that $(u, v) \in E$ and $v \notin H^+$. Hence, $v \notin \mathbb{X}_0$ by Definition 12. Thus, $\iota(v) > 0$ by Definition 13. Therefore, $\iota(v) \geq \iota(u)$ by the assumption $\iota(u) = 1$ of the case.

Case 3: $1 < \iota(u) < \infty$. Then, $u \notin \mathbb{X}_{\iota(u)-1}$ by Definition 13. Hence, $u \notin (\mathbb{X}_{\iota(u)-2} \cup \text{tl}(\mathbb{X}_{\iota(u)-2}))^+$ by Definition 12 and the assumption $1 < \iota(u)$ of the case. Thus, by Lemma 12 and the assumption $u \in V \setminus \text{Sink}$ of the current lemma, there exists a node $v \in V$ such that $(u, v) \in E$ and $v \notin (\mathbb{X}_{\iota(u)-2} \cup \text{tl}(\mathbb{X}_{\iota(u)-2}))^+$. Then, $v \notin \mathbb{X}_{\iota(u)-1}$ by Definition 12. Hence, $\iota(v) > \iota(u) - 1$ by Definition 13. Therefore, $\iota(v) \geq \iota(u)$.

Case 4: $\iota(u) = \infty$. The assumption $u \in V \setminus \text{Sink}$ of the lemma implies that there is a node $v \in V$ such that $(u, v) \in E$. It suffices to show that $\iota(v) = \infty$. Towards a contradiction, suppose that $\iota(v) < \infty$. Note that $u \in \text{tl}(\mathbb{X}_{\iota(v)})$ by the assumption $(u, v) \in E$ and Definition 11. Thus, $u \in \mathbb{X}_{\iota(v)} \cup \text{tl}(\mathbb{X}_{\iota(v)})$. Hence, $u \in (\mathbb{X}_{\iota(v)} \cup \text{tl}(\mathbb{X}_{\iota(v)}))^+$ by item 1 of Definition 10. Then, $u \in \mathbb{X}_{\iota(v)+1}$ by Definition 12. Therefore, $\iota(u) \leq \iota(v) + 1 < \infty$ by the assumption $\iota(v) < \infty$, which contradicts the assumption $\iota(u) = \infty$ of the case. $\square$

The next lemma follows from Definition 7 and the four previous lemmas.

Lemma 21. *For any mechanism, the irrecoverability index is a ranking function.*

Recall that Figure 4a, Figure 4b, and Figure 5 visualise three different ranking functions for the mechanism depicted in Figure 1b. It is easy to see that, for each node of the tree, the value of the function in Figure 5 is at least as large as the corresponding values of the two other ranking functions. This too is not a coincidence. The next lemma proves that all ranking functions on all mechanisms are bounded by the irrecoverability index. Since, by Lemma 21, the irrecoverability index itself is a ranking function, this means that the irrecoverability index is equal to the pointwise maximum of all ranking functions.

Lemma 22. *For each ranking function φ and node u of a mechanism, $\varphi(u) \leq \iota(u)$.*

Proof. The statement of the lemma follows from Lemma 16 and Definition 13. $\square$

We are finally ready to state and prove our second main result: the existence of a computationally efficient procedure for finding a k -agent mechanism providing k -oversight.

Theorem 2. *For any mechanism M and integer $k \geq 1$, it can be determined in polynomial time whether there exists a mechanism M' structurally equivalent to M that ensures k -oversight. If such a mechanism M' exists, it can be constructed in polynomial time as well.*

Proof. The family of sets $\{\mathbb{X}_i\}_i$ for the mechanism $M = (V, E, Q, H, A, \{C_a\}_{a \in A})$ can be constructed in polynomial time using the recursive construction from Definition 12. Thus, by Definition 13, the value $\iota(u)$ for each node $u \in V$ can be computed in polynomial time. We consider the following two cases separately:

Case 1: $\min_{q \in Q} \iota(q) < k$. In this case, mechanism M' does not exist. To prove this, assume that a mechanism $M' = (V, E, Q, H, A', \{C'_a\}_{a \in A'})$ ensures k -oversight and is structurally equivalent to mechanism M . Let ρ' be the responsibility index of mechanism M' . By Lemma 9, function ρ' is a ranking function for mechanism M' . By Lemma 4, function ρ' is also a ranking function for mechanism M . Thus, $\rho'(u) \leq \iota(u)$ for each node $u \in V$ by Lemma 22. Hence, $\min_{q \in Q} \rho'(q) < k$ by the assumption $\min_{q \in Q} \iota(q) < k$ of the case. Then, there is an initial node $q_0 \in Q$ such that $\rho'(q_0) < k$. Thus, by Definition 9, there is a decision path $u_1, \dots, u_n$ from the initial node $u_1 = q_0$ to a harmful sink node $u_n \in H$ whose responsibility count is less than k under mechanism M' . Hence, by Definition 8, fewer than k agents are responsible along the path $u_1, \dots, u_n$ under mechanism M' . Therefore, by Definition 5, mechanism M' does not ensure k -oversight.

Case 2: $\min_{q \in Q} \iota(q) \geq k$. Note that ι is a ranking function by Lemma 21. Thus, by Lemma 10, there is a mechanism structurally equivalent to M that has only k agents and ensures k -oversight. The construction of this mechanism, specified by equation (1), can be accomplished in polynomial time because the value $\iota(u)$ for each node $u \in V$ can be computed in polynomial time. $\square$

Note that the construction in the proof of Theorem 2 produces a structurally equivalent mechanism under which some of the nodes might not be controlled by any agents (that is, $\bigcup_{a \in A} C_a \subsetneq V \setminus \text{Sink}$). Intuitively, such nodes can be viewed as controlled by nature or the environment. However, if desired, these nodes can be put under the control of any of the agents. This modification preserves k -oversight.

5. Related Work

Counterfactual responsibility is probably the most widely discussed responsibility concept (Widerker and McKenna, 2003). However, at least four other formal definitions of responsibility have been considered. One of them is the notion of *responsibility for seeing to harm* (Broersen, 2011a,b; Naumov and Tao, 2021, 2023; Shi, 2024), extensively studied in STIT logic (Belnap and Perloff, 1990; Horty, 2001; Horty and Belnap, 1995; Horty and Pacuit, 2017; Olkhovikov and Wansing, 2019) under the name “deliberatively seeing to it”. In the setting where agents act sequentially, as in our Definition 1, an agent is responsible for seeing to harm if the agent took an action that eliminated the last possibility of a non-harmful outcome. For example, although no agent is responsible counterfactually along the decision path v_1, v_2, v_5, u_2 in Figure 1b, factory

F2 is responsible for seeing to harm along that decision path. This is because, along that path, the action “dump into lake C” by factory F2 eliminated the last possibility that the fish in all three lakes survive. In sequential mechanisms, at most one agent could be the one who eliminates the last possibility of a non-harmful outcome. Thus, in such mechanisms, at most one agent can be responsible along any decision path. As a result, for $k \geq 2$, it is impossible to construct a sequential decision mechanism that ensures k -oversight.

One can also consider a “bimodal” responsibility: an agent is bimodally responsible along a decision path if the agent is responsible either counterfactually or for seeing to harm. For any decision path, *bimodal* responsibility count can be one more than counterfactual responsibility count (see Definition 8) because, along the path, in addition to the agents responsible counterfactually, there could be an additional agent responsible for seeing to harm. Nevertheless, it is relatively easy to see that Lemma 9 holds for *bimodal* responsibility index. Thus, by Lemma 22, bimodal responsibility index is no more than the irrecoverability index. Therefore, bimodal responsibility cannot ensure a higher degree of oversight than we produce in Theorem 2 using counterfactual responsibility alone. In the case of our running example from Figure 1b, the bimodal responsibility index is simply equal to the irrecoverability index. Both are shown in Figure 5.

Another commonly discussed responsibility concept is the best-effort responsibility (Braham and van Hees, 2018; Naumov, 2026). As an example, imagine that three agents concurrently (say by a paper ballot) vote on whether to spare the life of a cat. The decision is made by the majority vote. Suppose that all three agents voted to kill the cat. The cat is dead. Who is responsible for this? None of the agents had a strategy to save the life of the cat. Thus, none of the agents is responsible counterfactually. At the same time, no individual vote guarantees the death of the cat. Hence, none of the agents is responsible for seeing to the death of the cat. Nevertheless, most people will agree that all three agents are intuitively responsible for the death. This intuition can be captured by the best-effort responsibility. Each agent is responsible because the cat is dead and did not use the action (vote to spare the cat’s life) that constitutes the best effort to save the cat. Best-effort actions can be defined as non-weakly dominated actions, assuming that a harmful outcome has utility -1 and a non-harmful outcome has utility 0. Note that best-effort responsibility is only meaningful in the setting where agents take actions just once and all of them act simultaneously. It is not clear what a best-effort action could be in the settings of Definition 1. The situation here is similar to classical game theory where the notion of weakly dominated strategy exists for strategic games and does not exist for extensive form games. Best-effort responsibility could also be considered in a probabilistic setting, where an agent might not be minimising the expected chances of a harmful outcome (Duijf and van De Putte, 2022). We do not discuss best-effort responsibility in this article because it is not compatible with the class of decision mechanisms that we consider.

Finally, Shi (2024) introduced the notion of a higher-order responsibility. This notion has also been discussed in (Jiang and Naumov, 2026). An agent a is second-order responsible for the harmful outcome if no agent is counterfactually responsible for the harm and the agent a had a strategy that would guarantee that if harm happens then at least one agent is counterfactually responsible for it. One can similarly introduce third-, fourth-, and higher-order responsibility. We leave the study of k -oversight under higher-order responsibility as a direction for future research.

In addition to the works on formally defining various responsibility concepts, there are three papers (Braham and van Hees, 2018; Duijf and van De Putte, 2022; Naumov and Tao, 2025) that show that gap-free mechanisms do not exist in some special classes of mechanisms. None of them studies how to construct a mechanism with desirable responsibility-related properties, as we do in Theorem 2.

Finally, the concept of k -oversight is closely related to the concept of separation of duty in information security. The latter is a design principle stating that certain crucial steps of a process are carried out by different users (Groll, Fuchs and Pernul, 2025).

6. Conclusion

This article introduced peer k -oversight as a responsibility-related property of sequential collective decision mechanisms. Rather than asking only who is responsible after a harmful outcome occurs, peer oversight treats responsibility as a design requirement: control over the decision process should be distributed so that at least k agents are responsible whenever harm occurs.

The main results show that whenever k -oversight can be achieved by redistributing control over the decision nodes, it can be achieved using exactly k agents. Moreover, whether such a redistribution exists can be determined in polynomial time, and a corresponding k -agent mechanism can be constructed efficiently.

These results suggest peer oversight as an architectural design principle for collective and multiagent decision-making systems, including multiagent AI systems in which agents are intended to oversee one another. A natural direction for future work is to allow the required degree of peer oversight to depend on the harmful outcome. Since harmful outcomes can differ in severity, more serious harms may warrant stronger oversight, with the required value of k varying across harmful sink nodes.

References

- Baier, C., Funke, F., Majumdar, R., 2021. A game-theoretic account of responsibility allocation, in: 30th International Joint Conference on Artificial Intelligence.
- Belnap, N., Perloff, M., 1990. Seeing to it that: A canonical form for agentives, in: Knowledge representation and defeasible reasoning. Springer, pp. 167–190.
- Braham, M., van Hees, M., 2018. Voids or fragmentation: Moral responsibility for collective outcomes. *The Economic Journal* 128, F95–F113.
- Broersen, J., 2011a. Deontic epistemic STIT logic distinguishing modes of mens rea. *Journal of Applied Logic* 9, 137–152.
- Broersen, J.M., 2011b. Making a start with the STIT logic analysis of intentional action. *Journal of Philosophical Logic* 40, 499–530.
- Bulling, N., Dastani, M., 2013. Coalitional responsibility in strategic settings, in: International Workshop on Computational Logic in Multi-Agent Systems, Springer. pp. 172–189.
- Chockler, H., Halpern, J.Y., 2004. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research* 22, 93–115.
- Duijf, H., van De Putte, F., 2022. The problem of no hands: responsibility voids in collective decisions. *Social Choice and Welfare* 58, 753–790.
- Frankfurt, H.G., 1969. Alternate possibilities and moral responsibility. *The Journal of Philosophy* 66, 829–839. doi:10.2307/2023833.
- Groll, S., Fuchs, L., Pernul, G., 2025. Separation of duty in information security. *ACM Computing Surveys* 57, 1–35.
- Halpern, J.Y., 2016. Actual causality. MIT Press.
- Horty, J., Pacuit, E., 2017. Action types in STIT semantics. *The Review of Symbolic Logic* 10, 617–637.
- Horty, J.F., 2001. Agency and deontic logic. Oxford University Press, Oxford, England.
- Horty, J.F., Belnap, N., 1995. The deliberative STIT: A study of action, omission, ability, and obligation. *Journal of Philosophical Logic* 24, 583–644.
- Jiang, J., Naumov, P., 2026. Higher-order responsibility, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 35545–35552. doi:10.1609/aaai.v40i42.40865.
- Naumov, P., 2026. Responsible mechanism design, in: Proceedings of the AAAI Conference on Artificial Intelligence, Senior Member Track: Blue Sky Ideas, pp. 39745–39752.
- Naumov, P., Tao, J., 2020. Blameworthiness in security games, in: Proceedings of Thirty-Fourth AAAI Conference on Artificial Intelligence.

- Naumov, P., Tao, J., 2021. Two forms of responsibility in strategic games, in: 30th International Joint Conference on Artificial Intelligence.
- Naumov, P., Tao, J., 2023. Counterfactual and seeing-to-it responsibilities in strategic games. *Annals of Pure and Applied Logic* 174, 103353.
- Naumov, P., Tao, J., 2025. Responsibility gap in collective decision making, in: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence.
- Olkhovikov, G.K., Wansing, H., 2019. Inference as doxastic agency. Part I: The basics of justification STIT logic. *Studia Logica* 107, 167–194.
- Shi, Q., 2024. Responsibility in extensive form games, in: Proceedings of 38th AAAI Conference on Artificial Intelligence.
- Shi, Q., Naumov, P., 2025. Responsibility in multi-step decision schemes. *Journal of Philosophical Logic* doi:10.1007/s10992-025-09794-7.
- Shi, Q., Naumov, P., 2026. A graph-theoretical perspective on law design for multiagent systems, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 29669–29677.
- Widerker, D., McKenna, M. (Eds.), 2003. *Moral responsibility and alternative possibilities: Essays on the importance of alternative possibilities*. Ashgate, Burlington, VT.
- Yazdanpanah, V., Dastani, M., Jamroga, W., Alechina, N., Logan, B., 2019. Strategic responsibility under imperfect information, in: Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems. pp. 592–600.