# PERFORMATIVE PERSONALIZATION INCENTIVIZES TRUTHFULNESS IN FEDERATED LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We study collaborative learning with strategic clients who may misreport oracle outputs to steer the learned model. Under simultaneous realizability and leave-one-out identifiability, we show that a leave-one-out consensus mechanism prevents harmful unilateral misreports and identifies the deviator. We also propose a one-shot alternative that uses a minimal cluster-recovering personalization oracle and preserves incentive compatibility without an $M$-fold increase in computation.

## 1 INTRODUCTION

Consider $M$ clients, collaborating through a server, and each client $m \in [M]$ seeking to minimize

$$F_m(x) := \mathbb{E}_{z\sim\mathcal{D}_m}\big[f(x;z)\big] \ ,$$

where $\mathcal{D}_m \in \Delta(\mathcal{Z})$ denotes the client data distribution, $f : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ is a shared loss, and $\mathcal{X} \subseteq \mathbb{R}^d$ is the model space. In many deployments, the server uses the learned model as an input to a downstream decision rule: it predicts demand to allocate scarce resources, estimates utilities to select policies, or produces rankings that shape attention and revenue Hwang et al. (2021); Vuppalapati et al. (2023); Fischer-Abaigar et al. (2024).

Because downstream decisions depend on the learned model, a client can benefit when the mechanism returns a predictor that favors its own distribution or induces a favorable decision rule, even when this shift slightly worsens aggregate performance. This incentive can drive a client to push the learned solution toward its own objective while harming others. To formalize this we first define

$$S_m^\star := \arg\min_{x\in\mathcal{X}} F_m(x) \qquad \text{and} \qquad S_m^\epsilon := \Big\{x \in \mathcal{X} : F_m(x) - F_m^\star \le \epsilon\Big\} \ .$$

We then model each client by a utility function $u_m : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ that enforces a lexicographic priority: first achieve $\epsilon$-optimality for $F_m$, then improve $F_m$ within that set, then (when tied at optimality) worsen other clients according to a competitive preference.

**Assumption 1** ($\epsilon$-hierarchically-competitive utilities)**.** *The utilities $(u_1, \dots, u_M)$ are $\epsilon$-hierarchically-competitive if for every $m \in [M]$:*

1. *(Feasibility) If $x \notin S_m^\epsilon$, then $u_m(x) = \infty$ .*
2. *(Within-client ordering) For any $x, y \in S_m^\epsilon$, if $F_m(x) < F_m(y)$ then $u_m(x) > u_m(y)$ .*
3. *(Competitive tie-break on $S_m^\star$) For any $x, y \in S_m^\star$, if there exists $n \in [M] \setminus \{m\}$ such that*

$$F_n(x) > F_n(y) \quad \text{and} \quad F_\ell(x) \ge F_\ell(y) \quad \text{for all } \ell \in [M] \setminus \{m, n\} \ ,$$

*then $u_m(x) > u_m(y)$ .*

Clients may hold different competitive preferences, as long as their utilities satisfy Assumption 1. Our goal is to design a collaborative learning mechanism that makes **truthful participation incentive-compatible**, so the server can reliably use the learned model in downstream decisions.

### 1.1 SETUP AND ASSUMPTIONS

Strategic incentives in Assumption 1 hinge on comparisons across clients: a client prefers outcomes that remain near-optimal for itself while shifting losses for others. To act on such preferences, a

client must estimate how the current model performs on other clients' data distributions. We capture this information access by a zeroth-order (value) oracle.

**Definition 1** (Value oracle)**.** For each $m \in [M]$, a value oracle $\mathcal{O}_m^0 : \mathcal{X} \to \Delta(\mathbb{R})$ takes as input a model $x \in \mathcal{X}$ and returns a random value $\mathcal{O}_m^0(x)$ such that $\mathbb{E}\big[\mathcal{O}_m^0(x)\big] = F_m(x)$ .

We assume clients can query $\mathcal{O}_m^0$ for other clients, for instance via public evaluation data that approximates other clients' distributions. We next impose two structural assumptions that make collaboration well-posed and allow us to avoid strategic effects. First, to ensure that the server has a meaningful global target and that collaboration can in principle benefit all parties, we assume that all clients share at least one common minimizer.

**Assumption 2** (Simultaneous realizability)**.** *The intersection of the individual minimizer sets is nonempty, i.e.,* $W^\star := \bigcap_{m=1}^M S_m^\star \neq \varnothing$ .

Second, we require a redundancy condition: removing any single client does not enlarge the $\epsilon$-optimal target set. This condition prevents a single strategic client from becoming pivotal.

**Assumption 3** (Leave-one-out identifiability)**.** *For every* $n \in [M]$, $\bigcap_{m \neq n} S_m^\epsilon = \bigcap_{m \in [M]} S_m^\epsilon$ .

Assumption 3 implies that no single client can change the feasible $\epsilon$-optimal set available to the mechanism; any deviation that attempts to move the outcome outside the common intersection becomes detectable through leave-one-out consistency checks.

A simple scenario that yields such redundancy arises when clients come from a small set of archetypes and each archetype appears at least twice among the $M$ clients.

**Proposition 1** (Redundancy from archetypes)**.** *Assume the* $M$ *clients are sampled i.i.d. uniformly from* $k$ *archetypes. Let* $N_j$ *denote the number of sampled clients of archetype* $j$*, and define the event* $E := \big\{ \min_{j \in [k]} N_j \geq 2 \big\}$*. Then,* $\mathbb{P}(E) \geq 1 - ke^{-M/k} - Me^{-(M-1)/k}$ .

In practice, the server can enforce Assumption 3 by actively maintaining coverage across archetypes (for example, by admission control or by soliciting additional clients from underrepresented groups) rather than relying on random sampling. Thus, one can view Assumption 3 as a coverage requirement. In the next section, we show how a simple mechanism leverages this coverage to rule out strategic manipulation while still producing a good model for the server.

## 2 Reciprocity and Redundancy Ensure Truthfulness

A canonical approach to collaboration is to minimize the averaged objective $\frac{1}{M} \sum_{m \in [M]} F_m(x)$. Under Assumption 2, if all clients are truthful and the server solves this problem exactly, it will recover a solution in $W^\star$. This motivates the following abstract oracle.

**Definition 2** (Consensus Oracle)**.** Assume there are $M$ clients, each with objective $F_m : \mathcal{X} \to \mathcal{R}$ and an oracle $\mathcal{O}_m : \mathcal{X} \to O$, where $O$ is some vector space. Then, $\mathcal{C} : \big(O^{\otimes M}\big)^\star \to \mathcal{X}$ is an $\epsilon$-consensus oracle if it takes as input $M$ oracles $(\mathcal{O}_1, \ldots, \mathcal{O}_M)$ and outputs $\hat{x} \in W^\epsilon := \cap_{m \in [M]} S_m^\epsilon$ .

**Remark 1** (Federated optimization algorithms)**.** A standard instantiation uses stochastic first-order oracles: for each $m \in [M]$, an oracle $\mathcal{O}_m^{(1)} : \mathcal{X} \to \mathbb{R}^d$ satisfies $\mathbb{E}\big[\mathcal{O}_m^{(1)}(x)\big] = \nabla F_m(x)$. One can simulate $\mathcal{O}_m^{(1)}$ by drawing fresh samples $z \sim \mathcal{D}_m$ and returning $\nabla f(x; z)$. This interface underlies mini-batch SGD (Federated SGD) Dekel et al. (2012); Woodworth et al. (2020) and Local SGD (Federated Averaging) McMahan & Ramage (2017); Patel et al. (2024). For example, under convexity and smoothness, and assuming bounded oracle variance, mini-batch SGD requires $\Theta(M/\epsilon)$ communication rounds and $\Theta(M/\epsilon^2)$ stochastic gradient evaluations per client to output in $W^\epsilon$.

The difficulty is that Definition 2 presumes truthful oracle access, but our model permits strategic clients whose utilities depend on other clients' losses (Assumption 1). A client can therefore manipulate the information it returns to the server. With first-order feedback, for instance, a client can rescale its gradients, add a systematic bias, or rotate directions so as to steer the iterate away from $W^\epsilon$ while still ensuring the final model remains inside its own $\epsilon$-optimal set.

At the same time, we do not attempt to prohibit or detect every false oracle output. Many deviations remain benign for the final outcome, including stochastic noise, approximate computation, delayed

or quantized updates, or privacy-preserving perturbations. We instead focus on deviations that matter for agreement with the server's target. Under Assumption 2, the server ultimately wants to recover a point in the exact agreement set $W^\epsilon$, whereas clients may still prefer outcomes that lie in $W^\epsilon \setminus W^\star$ (see Assumption 1). We therefore call a deviation by client $m$ *harmful* if it can cause the protocol to output a final model $\hat{x} \notin W^\epsilon$. Accordingly, we do not require each reported oracle response to equal the true gradient or value. We require only that clients lack an incentive to deviate in a way that can move the final outcome outside $W^\epsilon$. We now formalize this requirement in our oracle model.

**Definition 3** ($\epsilon$-agreement incentive compatibility)**.** Fix utilities $(u_1, \ldots, u_M)$ satisfying Assumption 1. Let $\mathcal{A}$ be an interactive protocol that can make repeated, adaptive queries to reported oracle interfaces and outputs $\hat{x} = \mathcal{A}(\tilde{\mathcal{O}}_1, \ldots, \tilde{\mathcal{O}}_M)$. Let $\mathcal{O}_m$ denote client $m$'s truthful oracle, and let $\tilde{\mathcal{O}}_m$ denote any (possibly randomized, adaptive) reporting strategy. We say that $\mathcal{A}$ is *$\epsilon$-agreement incentive compatible* if for every client $m \in [M]$ and every alternative strategy $\tilde{\mathcal{O}}_m$,

$$\mathbb{E}\Big[u_m\Big(\mathcal{A}(\tilde{\mathcal{O}}_m, \mathcal{O}_{-m})\Big)\Big] \;>\; \mathbb{E}[u_m(\mathcal{A}(\mathcal{O}_m, \mathcal{O}_{-m}))] \quad\Longrightarrow\quad \mathbb{P}\left(\mathcal{A}(\tilde{\mathcal{O}}_m, \mathcal{O}_{-m}) \in W^\epsilon\right) = 1 \;,$$

where $\mathcal{O}_{-m}$ denotes truthful reporting by all clients other than $m$, and the expectation is over protocol and oracle randomness.

We consider a simple mechanism that prevents any single client from unilaterally compromising agreement. The mechanism relies on redundancy through reciprocity. Under Assumption 3, any collection of $M-1$ honest clients already suffices to identify a solution in $W^\epsilon$. The server, however, does not know which client (if any) acts strategically, and the server cannot verify whether a client reports truthful oracle outputs during execution. This information asymmetry motivates a leave-one-out design. The server runs $M$ consensus instances in parallel, each excluding a different client. A strategic client can influence every instance in which it participates, but it cannot influence the instance that excludes it. That excluded run therefore yields a fallback candidate that remains in the common agreement set, which removes the incentive to deviate in a way that threatens agreement.

**Theorem 1.** *Assume Assumptions 2 and 3, and let $\mathcal{C}$ be an $\epsilon$-consensus oracle (Definition 2). Suppose at most one client deviates arbitrarily from truthful oracle reporting. Consider the following mechanism: for each $n \in [M]$, the server computes*

$$\hat{x}^{(-n)} \;\leftarrow\; \mathcal{C}\big(\tilde{\mathcal{O}}_1, \ldots, \tilde{\mathcal{O}}_{n-1}, \tilde{\mathcal{O}}_{n+1}, \ldots, \tilde{\mathcal{O}}_M\big),$$

*and then outputs any $\hat{x}^{(-\hat{n})}$ that lies in $W^\epsilon = \bigcap_{m=1}^M S_m^\epsilon$ (verified using the value oracle in Definition 1). Then the mechanism is $\epsilon$-agreement incentive compatible (Definition 3).*

**Remark 2** (Identifying the strategic client)**.** The leave-one-out protocol can also identify a strategic client. Consider the candidates $\{\hat{x}^{(-n)}\}_{n=1}^M$ and evaluate each candidate using the value oracles. Under Assumption 3, if all clients except $m^\star$ report truthfully, then the candidate $\hat{x}^{(-m^\star)}$—which excludes the strategic client—lies in $W^\epsilon$ and therefore satisfies the $\epsilon$-optimality constraints for every client. In contrast, any candidate $\hat{x}^{(-n)}$ with $n \neq m^\star$ includes client $m^\star$, and any *harmful* deviation must cause at least one such candidate to violate the $\epsilon$-optimality condition for some honest client (otherwise it would still lie in $W^\epsilon$). Hence, by checking which leave-one-out candidates fail the $\epsilon$-sublevel tests, the server can identify a client whose exclusion restores agreement. This identifiability creates an additional deterrent: collaborative learning deployments often sit inside contractual relationships, where detected manipulation can trigger audits, termination of participation, or legal remedies. In such settings, the risk of attribution further discourages unilateral strategic behavior.

This section presents a simple mechanism that uses only a consensus optimization oracle (Definition 2) to enforce $\epsilon$-agreement incentive compatibility. The leave-one-out design follows naturally from Assumption 3. The main drawback is computational. The mechanism runs consensus optimization $M$ times, which multiplies total work by a factor of $M$. The server cannot safely skip instances instance either, since it cannot predict in advance which client, if any, may act strategically. In the next section, we show how to remove this bottleneck by augmenting the protocol with a personalization oracle, which achieves the same incentive guarantee with a lower overhead.

## 3 Simulating Counterfactuals through Personalization

The leave-one-out mechanism in Theorem 1 explicitly simulates counterfactual worlds in which one client does not participate. We now show that a suitable personalization primitive can approximate

the same counterfactual reasoning in a single run. The key point is that a client's oracle reports encode which solution concepts it is compatible with. Personalization can exploit this information to infer which groups of clients can share a common model, without explicitly rerunning consensus under $M$ different exclusions. We formalize this by defining a personalization oracle that returns a small collection of models together with an assignment of clients to models. Intuitively, each model represents the solution concept for one agreement cluster, and the oracle seeks the smallest number of such clusters that keeps every client $\epsilon$-satisfied.

**Definition 4** (A minimal personalization oracle)**.** Assume there are $M$ clients, each with objective $F_m : \mathcal{X} \to \mathcal{R}$ and an oracle $\mathcal{O}_m : \mathcal{X} \to O$, where $O$ is some vector space. A mapping

$$\mathcal{P} : (O^{\otimes M})^\star \to \bigcup_{k=1}^{M} \left(\mathcal{X}^k \times [k]^M\right)$$

is an *$\epsilon$-personalization oracle* if, on input $(\mathcal{O}_1, \ldots, \mathcal{O}_M)$, it outputs a set of models $(x^{(1)}, \ldots, x^{(k)})$ and an assignment $a : [M] \to [k]$ such that for every $i \in [k]$, if we define the induced cluster $C_i := \{m \in [M] : a(m) = i\}$, then the cluster model satisfies $x^{(i)} \in \bigcap_{m \in C_i} S_m^\epsilon$. We call $\mathcal{P}$ *minimal cluster-recovering* if it returns a feasible output with the smallest possible $k$ among all pairs $(\{x^{(i)}\}_{i=1}^k, a)$ satisfying the above cluster-feasibility condition.

The definition only enforces within-cluster agreement. When global agreement exists, the oracle should collapse to a single cluster; when global agreement fails, the oracle should separate incompatible clients into different clusters. The next remark records the behavior under truthful reporting.

**Remark 3** (Behavior under truthful reports)**.** Assume all clients report truthfully. If $W^\epsilon = \cap_{m=1}^M S_m^\epsilon$ is non-empty, then there exists a feasible output with $k = 1$, namely one model $x^{(1)} \in W^\epsilon$ assigned to every client. By minimality, a minimal cluster-recovering oracle returns $k = 1$ in this case, and it reduces to consensus. If there exists an outlier $n$ such that $\cap_{m \neq n} S_m^\epsilon \neq \varnothing$ but $W^\epsilon = \varnothing$, then there exists a feasible output with $k = 2$: one model shared by the $M - 1$ compatible clients and one model for client $n$. Minimality forces the oracle return these two models. More generally, suppose the clients partition into disjoint clusters $(C_1, \ldots, C_k)$ such that each $\cap_{m \in C_i} S_m^\epsilon$ is non-empty, and no single model lies in $S_m^\epsilon$ for clients drawn from two different clusters. Then the oracle outputs one model per cluster and assigns clients accordingly.

We now use this oracle as a one-shot replacement for the $M$ leave-one-out consensus calls. The server runs the personalization oracle once, and then selects a model that satisfies all clients whenever such a model exists. Under realizability and leave-one-out identifiability, such a model exists even in the presence of a unilateral misreport.

**Theorem 2.** *Assume Assumptions 2 and 3. Let $\mathcal{P}$ be a minimal cluster-recovering $\epsilon$-personalization oracle (Definition 4). Suppose at most one client deviates arbitrarily from truthful oracle reporting. Consider the following mechanism: compute*

$$\left((x^{(1)}, \ldots, x^{(k)}), a\right) \leftarrow \mathcal{P}(\tilde{\mathcal{O}}_1, \ldots, \tilde{\mathcal{O}}_M),$$

*and output any $x^{(i)}$ that lies in $W^\epsilon = \bigcap_{m=1}^M S_m^\epsilon$ (verified using the value oracle in Definition 1). Then the mechanism is $\epsilon$-agreement incentive compatible (Definition 3).*

The proof uses two structural facts. Realizability implies $W^\epsilon \neq \varnothing$. Leave-one-out identifiability implies that the intersection of the truthful clients' $\epsilon$-sublevel sets already equals $W^\epsilon$, so the truthful clients always form a feasible cluster that admits a single shared model in $W^\epsilon$. Minimality then forces the personalization oracle to include such a shared model among its outputs. The selection rule extracts it, which makes every unilateral deviation benign with respect to agreement.

**Future work.** A natural next step is to move beyond unilateral deviations and study coalition deviations, for example through core-stability style requirements where no subset of clients can jointly deviate to obtain a strictly preferred outcome. At a high level, one can extend our approach by strengthening Assumption 3 to an identifiability condition that remains valid after removing any subset of up to $k$ clients, and then combining this with a personalization oracle that recovers the minimal number of agreement clusters. This yields a direct generalization of our counterfactual reasoning, although it is significantly more restrictive. A second direction concerns computation. Our results treat the minimal cluster-recovering personalization oracle as an abstract primitive, but implementing it efficiently for broad problem classes remains challenging, especially under noisy oracle access, privacy constraints, and non-convexity Ghosh et al. (2020); Vardhan et al. (2024).

## REFERENCES

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(1), 2012.

Unai Fischer-Abaigar, Christoph Kern, Noam Barda, and Frauke Kreuter. Bridging the gap: Towards an expanded toolkit for ai-driven decision-making in the public sector. *Government Information Quarterly*, 41(4):101976, 2024.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in neural information processing systems*, 33:19586–19597, 2020.

Dawsen Hwang, Patrick Jaillet, and Vahideh Manshadi. Online resource allocation under partially predictable demand. *Operations Research*, 69(3):895–915, 2021.

Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data, Apr 2017. URL `https://ai.googleblog.com/2017/04/federated-[]learning-[]collaborative.html`.

Kumar Kshitij Patel, Margalit Glasgow, Ali Zindari, Lingxiao Wang, Sebastian U Stich, Ziheng Cheng, Nirmit Joshi, and Nathan Srebro. The limits and potentials of local sgd for distributed heterogeneous learning with intermittent communication. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 4115–4157. PMLR, 2024.

Harsh Vardhan, Avishek Ghosh, and Arya Mazumdar. An improved federated clustering algorithm with model-based clustering. *Transactions on machine learning research*, 2024.

Midhul Vuppalapati, Giannis Fikioris, Rachit Agarwal, Asaf Cidon, Anurag Khandelwal, and Éva Tardos. Karma: Resource allocation for dynamic demands. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pp. 645–662, 2023.

Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33:6281–6292, 2020.


## A APPENDIX

### A.1 PROOF OF PROPOSITION 1

*Proof.* We sample $M$ archetype labels independently, each uniformly distributed on $[k]$. For each archetype $j \in [k]$, define the count

$$N_j := \sum_{t=1}^{M} \mathbf{1}\{\text{the } t\text{-th sampled label equals } j\}.$$

By construction, $N_j$ is a binomial random variable with parameters $M$ and $1/k$, namely

$$N_j \sim \mathrm{Bin}(M, 1/k).$$

The event $E = \{\min_{j\in[k]} N_j \geq 2\}$ means that *no* archetype appears zero or one times.

**Step 1: Reduce to a union bound over archetypes.** The complement event is

$$E^c = \Big\{\exists j \in [k] \text{ such that } N_j \leq 1\Big\} = \bigcup_{j=1}^{k} \{N_j \leq 1\}.$$

Applying the union bound gives

$$\Pr(E^c) \leq \sum_{j=1}^{k} \Pr(N_j \leq 1). \tag{1}$$

**Step 2: Compute $\Pr(N_j \le 1)$ for a fixed archetype.** Fix any $j \in [k]$. Since $N_j \sim \mathrm{Bin}(M, 1/k)$, we can compute

$$\Pr(N_j \le 1) = \Pr(N_j = 0) + \Pr(N_j = 1).$$

For $N_j = 0$, none of the $M$ draws equals $j$, which happens with probability

$$\Pr(N_j = 0) = \Big(1 - \frac{1}{k}\Big)^M.$$

For $N_j = 1$, exactly one of the $M$ draws equals $j$. Choose which draw hits $j$ (there are $\binom{M}{1} = M$ choices), multiply by the probability that this draw equals $j$ (which is $1/k$), and multiply by the probability that all remaining $M-1$ draws do not equal $j$ (which is $(1-1/k)^{M-1}$). This gives

$$\Pr(N_j = 1) = \binom{M}{1}\frac{1}{k}\Big(1 - \frac{1}{k}\Big)^{M-1} = M \cdot \frac{1}{k}\Big(1 - \frac{1}{k}\Big)^{M-1}.$$

Therefore,

$$\Pr(N_j \le 1) = \Big(1 - \frac{1}{k}\Big)^M + M \cdot \frac{1}{k}\Big(1 - \frac{1}{k}\Big)^{M-1}. \tag{2}$$

**Step 3: Sum the bound over $j$ and simplify.** Plugging equation 2 into equation 1, and using that the right-hand side does not depend on $j$, we obtain

$$\Pr(E^c) \le k\Big(1 - \frac{1}{k}\Big)^M + k \cdot M \cdot \frac{1}{k}\Big(1 - \frac{1}{k}\Big)^{M-1} = k\Big(1 - \frac{1}{k}\Big)^M + M\Big(1 - \frac{1}{k}\Big)^{M-1}.$$

Finally,

$$\Pr(E) = 1 - \Pr(E^c) \ge 1 - k\Big(1 - \frac{1}{k}\Big)^M - M\Big(1 - \frac{1}{k}\Big)^{M-1},$$

which is the first inequality in the statement.

**Step 4: Convert to exponential form.** Using the standard inequality $(1-x)^r \le e^{-rx}$ for $x \in [0,1]$ and $r \ge 0$, with $x = 1/k$ and $r \in \{M, M-1\}$, we get

$$\Big(1 - \frac{1}{k}\Big)^M \le e^{-M/k}, \qquad \Big(1 - \frac{1}{k}\Big)^{M-1} \le e^{-(M-1)/k}.$$

Substituting these into the previous bound yields

$$\Pr(E) \ge 1 - ke^{-M/k} - Me^{-(M-1)/k}.$$

□

### A.2 Proof of Theorem 1

*Proof.* Fix a client index $m \in [M]$ and assume that all clients in $[M] \setminus \{m\}$ report truthfully, that is, they use $(\mathcal{O}_{-m})$. Let $\tilde{\mathcal{O}}_m$ be an arbitrary (possibly randomized, adaptive) reporting strategy for client $m$. We will show that the output of the leave-one-out mechanism lies in $W^\epsilon$ with probability one. This will imply $\epsilon$-agreement incentive compatibility by Definition 3.

**Step 1: $W^\epsilon$ is non-empty.** Assumption 2 states that $W^\star = \cap_{i=1}^M S_i^\star$ is non-empty. Since $S_i^\star \subseteq S_i^\epsilon$ for every $i$, we have $W^\star \subseteq W^\epsilon$, hence $W^\epsilon$ is also non-empty.

**Step 2: The leave-one-out instance that excludes $m$ yields a point in $W^\epsilon$.** Consider the candidate produced by excluding client $m$:

$$\hat{x}^{(-m)} \leftarrow \mathcal{C}\big(\mathcal{O}_1, \ldots, \mathcal{O}_{m-1}, \mathcal{O}_{m+1}, \ldots, \mathcal{O}_M\big).$$

This call to $\mathcal{C}$ uses only truthful oracles, so by the definition of an $\epsilon$-consensus oracle (Definition 2),

$$\hat{x}^{(-m)} \in \bigcap_{i \ne m} S_i^\epsilon.$$

Assumption 3 implies that removing any single client does not change the $\epsilon$-agreement set, hence

$$\bigcap_{i \ne m} S_i^\epsilon \;=\; \bigcap_{i=1}^M S_i^\epsilon \;=\; W^\epsilon,$$

so $\hat{x}^{(-m)} \in W^\epsilon$.

**Step 3: The mechanism outputs a point in $W^\epsilon$ with probability one.** The mechanism computes the $M$ candidates $\{\hat{x}^{(-n)}\}_{n=1}^M$ and then outputs any $\hat{x}^{(-\hat{n})}$ that lies in $W^\epsilon$, where the server verifies membership using the value oracle (Definition 1). Step 2 shows that, regardless of the choice of $\tilde{\mathcal{O}}_m$, the candidate $\hat{x}^{(-m)}$ lies in $W^\epsilon$. Therefore, at least one candidate passes the server's verification test, and any output selected by the mechanism satisfies

$$\mathbb{P}\big(\hat{x} \in W^\epsilon\big) = 1.$$

**Step 4: Incentive compatibility.** Since $\mathbb{P}(\mathcal{A}(\tilde{\mathcal{O}}_m, \mathcal{O}_{-m}) \in W^\epsilon) = 1$ holds for every client $m$ and every alternative reporting strategy $\tilde{\mathcal{O}}_m$, the implication in Definition 3 holds trivially: whenever a deviation strictly increases client $m$'s expected utility (or even if it does not), the resulting output still lies in $W^\epsilon$ with probability one. Hence the mechanism is $\epsilon$-agreement incentive compatible. □

### A.3 Proof of Theorem 2

*Proof.* Fix a client index $m \in [M]$ and assume all clients in $[M] \setminus \{m\}$ report truthfully, that is, they use $\mathcal{O}_{-m}$. Let $\tilde{\mathcal{O}}_m$ be an arbitrary (possibly randomized, adaptive) reporting strategy for client $m$. Let

$$\big((x^{(1)}, \ldots, x^{(k)}), a\big) \;\leftarrow\; \mathcal{P}(\tilde{\mathcal{O}}_m, \mathcal{O}_{-m})$$

denote the oracle output, with induced clusters $C_i = \{j : a(j) = i\}$, and let the mechanism output any $x^{(\hat{i})} \in W^\epsilon$ after value-oracle verification.

**Step 1: $W^\epsilon$ is non-empty.** Assumption 2 states that $W^\star = \cap_{j=1}^M S_j^\star$ is non-empty. Since $S_j^\star \subseteq S_j^\epsilon$ for every $j$, we have $W^\star \subseteq W^\epsilon$, hence $W^\epsilon$ is non-empty.

**Step 2: The truthful clients admit a shared model in $W^\epsilon$.** By Assumption 3,

$$\bigcap_{j \neq m} S_j^\epsilon \;=\; \bigcap_{j=1}^{M} S_j^\epsilon \;=\; W^\epsilon.$$

In particular, there exists a point $x^\dagger \in W^\epsilon$ that lies in $S_j^\epsilon$ for every truthful client $j \neq m$.

**Step 3: Minimality forces $\mathcal{P}$ to output a model in $W^\epsilon$.** Consider the following feasible clustering: assign all truthful clients $[M] \setminus \{m\}$ to one cluster, and assign client $m$ to either that same cluster or to a second singleton cluster. Step 2 shows that the truthful-clients cluster is feasible with cluster model $x^\dagger \in \cap_{j \neq m} S_j^\epsilon = W^\epsilon$. Therefore, there exists a feasible output of Definition 4 that uses at most two clusters and includes a cluster model in $W^\epsilon$.

Now consider the output of $\mathcal{P}$, which is minimal cluster-recovering. Let $i^\star$ be any index such that $[M] \setminus \{m\} \subseteq C_{i^\star}$ if such an index exists. If no such index existed, then the truthful clients would be split across at least two clusters. Merging all clusters that contain truthful clients into a single cluster remains feasible, because $x^\dagger$ lies in every truthful client's $S_j^\epsilon$. This merge would strictly reduce $k$, which contradicts minimality. Hence there exists an index $i^\star$ with $[M] \setminus \{m\} \subseteq C_{i^\star}$. By cluster-feasibility in Definition 4,

$$x^{(i^\star)} \in \bigcap_{j \in C_{i^\star}} S_j^\epsilon \subseteq \bigcap_{j \neq m} S_j^\epsilon = W^\epsilon.$$

Therefore, at least one model returned by $\mathcal{P}$ lies in $W^\epsilon$.

**Step 4: The mechanism outputs a point in $W^\epsilon$ with probability one.** Step 3 shows that the returned list contains at least one candidate in $W^\epsilon$. The mechanism outputs any candidate in $W^\epsilon$ after verification, hence

$$\mathbb{P}\Big(\mathcal{A}(\tilde{\mathcal{O}}_m, \mathcal{O}_{-m}) \in W^\epsilon\Big) = 1,$$

where $\mathcal{A}$ denotes the one-shot mechanism in Theorem 2.

**Step 5: Incentive compatibility.** Since the conclusion in Step 4 holds for every client $m$ and every alternative reporting strategy $\tilde{\mathcal{O}}_m$, the implication in Definition 3 holds: any deviation that strictly increases expected utility must still yield an output in $W^\epsilon$ with probability one. Therefore, the mechanism is $\epsilon$-agreement incentive compatible. □