LINEAR-COST UNSUPERVISED ELICITATION FROM PRETRAINED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training language models depend on human annotators for labels, which becomes a bottleneck as models grow more capable: annotation becomes unreliable as tasks outpace human expertise. Unsupervised elicitation avoids that by drawing labels from the pretrained model itself. Internal Coherence Maximization (ICM) is one such method; it approaches golden-supervision accuracy across a range of labeling tasks. However, its simulated-annealing search over mutual predictability requires $O(n^2)$ forward passes, hours for a few hundred examples, and per-dataset hyperparameter tuning. We show this search is largely unnecessary. Our method, Linear-Cost Unsupervised Elicitation (LC-UE), replaces it with two inexpensive phases. The first phase labels each group in order of the model's zero-shot confidence, conditioning only on the groups labeled before it, which optimizes an autoregressive form of coherence rather than mutual predictability itself. The second phase closes that gap with a few refinement sweeps, scoring candidate label flips under a cheap proxy against a frozen context and verifying each sweep with one exact computation. Both phases satisfy the task's consistency constraints by construction. Across TruthfulQA, GSM8K-verification, Alpaca, and two pluralistic alignment benchmarks, LC-UE matches or outperforms ICM and few-shot bootstrapping at $O(Kn)$ cost for $K \ll n$, with no per-dataset tuning. On TruthfulQA ($n = 2{,}604$), this amounts to an estimated $465\times$ fewer forward passes than ICM, and at least $250\times$ fewer even under a conservative bound. Taken together, LC-UE makes unsupervised elicitation practical at scale and across diverse tasks, without the computational cost or per-dataset tuning that has limited prior methods.

1 INTRODUCTION

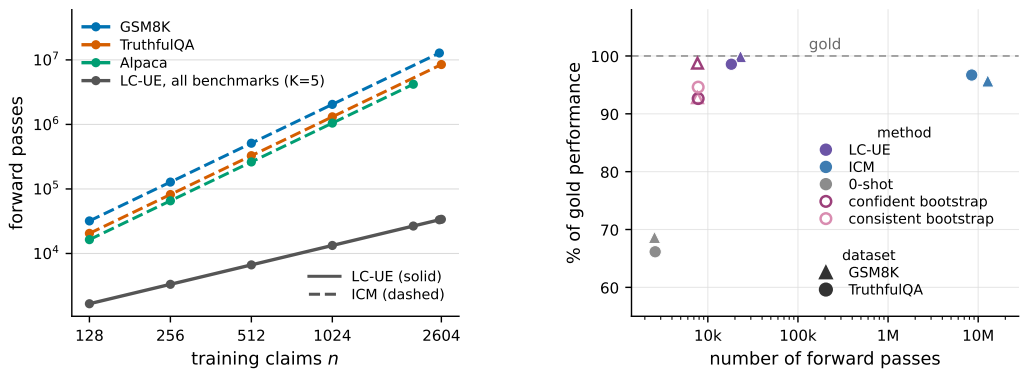
Aligning language models currently depends on humans to specify desired behavior through demonstrations and preference labels (Ouyang et al., 2022; Bai et al., 2022; Glaese et al., 2022). As models become more capable, this supervision becomes limited: annotators for harder tasks are hard to find and usually expensive, and annotations become less reliable as tasks get harder (Wen et al., 2025b). Unsupervised elicitation offers a way past this. Rather than teaching a model preferences using annotators, we surface the knowledge the pretrained model already has, without any external supervision or labels (Wen et al., 2025a; Qi et al., 2026).

Recent work has produced several methods that infer labels from a pretrained model without external supervision, including internal coherence maximization (ICM) and iterative bootstrapping. Qiu et al. (2026) show that these methods are instances of coherence maximization. ICM (Wen et al., 2025a) is one such method that performs well across a variety of tasks; it infers labels that maximize a dataset's mutual predictability for a given model under defined logical-consistency constraints.

ICM approaches supervision with gold labels across a range of tasks: verifying mathematical solutions, judging truthfulness, ranking assistant responses for helpfulness, and, in later work, inferring value judgments (Wen et al., 2025a; Mahbub et al., 2026; Alizadeh et al., 2026). However, two main problems stand in the way of using it.

First is cost. ICM's objective maximizes mutual predictability, a sum of leave-one-out conditionals, so changing one label updates the conditioning context of every other example and alters every term in the sum. Each updated label requires rescoring the entire labeled set. At roughly 2.0-3.9 search

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) (b)

Figure 1: **LC-UE needs orders of magnitude fewer forward passes than ICM while achieving higher accuracy.** ICM's forward-pass counts are estimated from its search iterations per datapoint r; see Section 3.4 for both cost derivations. **(a)** Forward passes to label each training set. LC-UE is shown at $K = 5$, the cap on the number of refinement sweeps (Section 3.7), which serves as an upper bound on its cost. Even at this bound, LC-UE needs $158\times$ (Alpaca) to $384\times$ (GSM8K-verification) fewer passes than ICM. **(b)** Test accuracy as a percentage of many-shot prompting with gold labels, against the number of forward passes (log scale), on GSM8K-verification (triangles) and TruthfulQA (circles). LC-UE is shown after Phase 2; 0-shot uses the base model. LC-UE comes closest to gold on both datasets and exceeds ICM on both, with $\sim 465\times$ (TruthfulQA) and $\sim 555\times$ (GSM8K-verification) fewer forward passes. The bootstrap variants use a similar number of passes but shorter contexts per pass, so their compute is lower than this axis suggests.

iterations per datapoint (Wen et al., 2025a), with each evaluation averaging $n/2$ forward passes, a run costs on the order of n^2 passes over the contexts that themselves grow in n. In practice, Shrivastava et al. (2026) report hours to label 256 examples with Llama-3.1-70B on four H200 GPUs and call this cost prohibitive over much larger datasets, and Mahbub et al. (2026) report 4 to 6 hours per search on an RTX 6000 Pro GPU. The second is the search itself. ICM approximates its objective by simulated annealing, with a coefficient α balancing mutual predictability against the logical-consistency penalty. The original work sets $\alpha = 50$ but lowers it to 20 or 30 when the search rejects too many proposals; Mahbub et al. (2026) note that these hyperparameters may require dataset-specific retuning, a limitation for any setting where per-dataset tuning is impractical. Together with the cost, this makes ICM difficult to reproduce and to deploy.

Shrivastava et al. (2026) suggest that much of ICM's benefit may not require the search at all; they evaluate simple iterative bootstrapping across the same tasks and recover a substantial fraction of ICM's gain at linear cost. However, their baselines discard mutual predictability entirely, optimizing it at best indirectly. This leaves an open question: can we keep the objective while removing the search?

We demonstrate that we can. Our method keeps mutual predictability as the objective but replaces the search with two phases: a single confidence-ordered autoregressive pass that labels each group conditioned on everything labeled so far, followed by greedy refinement sweeps in which candidate flips are scored under a cheap Mutual Predictability (MP) proxy against a frozen context and verified with one exact MP computation per sweep. Both phases emit only labelings inside the task's feasible set, so, unlike ICM, no separate repair step is needed. This reduces the total cost to $O(Kn)$ forward passes with $K \ll n$.

Contributions. We show that ICM's objective can be optimized without its search. A single confidence-ordered pass followed by a few greedy refinement sweeps, each verified with one exact mutual-predictability computation, matches or exceeds ICM's label quality at $O(Kn)$ cost, using $\sim 465\times$ and $\sim 555\times$ fewer forward passes on TruthfulQA and GSM8K-verification respectively (Figure 1b). We treat logical consistency as a hard constraint on both phases rather than a

penalty repaired afterwards, and specify the feasible set for each of the five benchmarks we study. We find that we can remove the search: ordering examples by zero-shot confidence reproduces the easy-before-hard effect of ICM's accept–reject dynamics, and cheap refinement then recovers the mutual-predictability objective that the ordered pass alone does not optimize. We confirm this across classification, verification, preference, and pluralistic alignment tasks.

2 BACKGROUND

2.1 INTERNAL COHERENCE MAXIMIZATION

Given an unlabeled dataset $\{x_i\}$, ICM searches for a label assignment $\{y_i\}$ that a pretrained model θ finds internally coherent, scoring a labeled set D as $U(D) = \alpha \cdot P_\theta(D) - I(D)$. The first term is mutual predictability, the summed log-probability of each label conditioned on all others:

$$P_\theta(D) = \sum_i \log P_\theta\big(y_i \mid x_i,\ D \setminus (x_i, y_i)\big) \tag{1}$$

A high value indicates that the labels jointly express a single concept under θ: any one label can be inferred from the rest. The second term, $I(D)$, counts violations of a pairwise logical consistency function, without which mutual predictability admits degenerate solutions such as assigning the same label everywhere.

Maximizing U exactly is an integer program, so ICM approximates it with simulated annealing: each iteration samples an example, assigns its most likely label under the current labeled set, repairs introduced inconsistencies through a separate CONSISTENCYFIX routine, and accepts the change if $\Delta = U(\hat{D}) - U(D)$ is positive, or otherwise with a probability that decays as the temperature is lowered. Two properties matter for what follows. Consistency is a soft penalty repaired afterward, so inconsistent labelings remain reachable during the search. And examples may be revisited, which is how ICM corrects earlier mistakes but also why it needs more iterations than there are examples, each paying for a full evaluation of U.

2.2 COHERENCE OPTIMIZATION

Qiu et al. (2026) show that ICM and simpler bootstrapping procedures optimize closely related objectives that differ only in their conditioning set. Mutual predictability conditions each label on all others; the coherence objective conditions each label only on its predecessors in some ordering. The distinction is bidirectional versus autoregressive conditioning, and it is precisely what makes ICM expensive: because every label sits in every other label's context, no term of P_θ can be updated independently. Coherence, by contrast, is computable by a single left-to-right pass. The same work establishes that Gibbs sampling over this objective, repeatedly resampling one label conditioned on the rest, converges to a distribution concentrated on high-coherence labelings.

2.3 BOOTSTRAPPING BASELINES

Shrivastava et al. (2026) evaluate simpler alternatives on the same tasks. Their bootstrap procedure labels a subset zero-shot, then proceeds in rounds, each labeling a fresh subset with few-shot prompts built from the previous round's predictions. *Confident Bootstrap* carries forward only the most confidently predicted examples, motivated by the hypothesis that ICM's accept-reject dynamics let it label easy examples before hard ones; *Consistent Bootstrap* applies a consistency projection to each round's predictions. Their projection is a hard constraint, but applied post hoc: predictions are produced independently and then repaired, with the highest-scoring claim in a group determining the rest. Mutual predictability plays no role in that choice, at least directly. Our method inherits the confidence ordering from the first variant and the hard constraint from the second, while restoring the MP objective that both discard.

3 METHODOLOGY

Figure 2 gives an overview. As in ICM, every decision comes from the model's log-probabilities over the two label tokens.

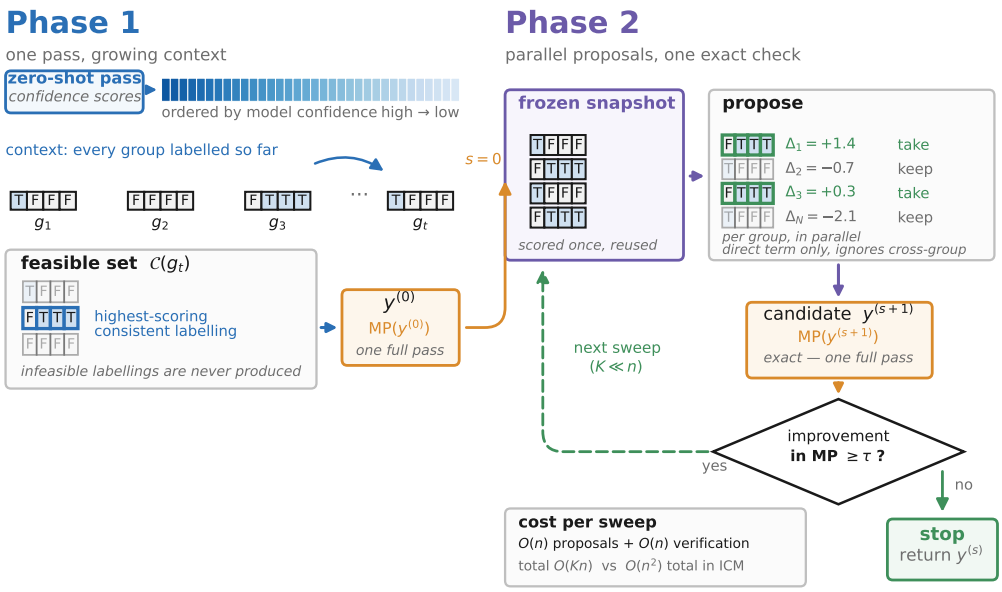


Figure 2: **Overview of the two-phase method.** Phase 1 labels groups in a single pass ordered by zero-shot confidence, appending each labeled group to a growing context and emitting only assignments inside the feasible set $\mathcal{C}(g)$. Phase 2 sweeps over all groups, scoring candidate moves against a snapshot of the labeling frozen at the start of the sweep, so scores are computed once and reused, then verifying the result with one exact mutual-predictability computation. Sweeps repeat until the improvement in exact MP falls below τ.

3.1 Logical Consistency as a Hard Constraint

Consistency is defined over a group of claims sharing a `consistency_id`. For a group g, let $\mathcal{C}(g) \subseteq \{0,1\}^{|g|}$ denote the set of label assignments permitted by the task's consistency rules. Both phases emit only elements of $\mathcal{C}(g)$: Phase 1 labels a whole group at once, and Phase 2 proposes moves between elements of $\mathcal{C}(g)$. The model's contribution per group is therefore a single choice among feasible assignments, with the remaining labels following deterministically from the constraint.

This is stricter than either prior approach. ICM treats consistency as a penalty, so inconsistent labelings remain reachable and are repaired afterward by a separate ConsistencyFix routine. In our method, nothing needs repair, as infeasible labelings are never reached. Since all of these methods ultimately yield consistent labels, what separates them is which feasible assignment they select: ICM selects by mutual predictability, Shrivastava et al. (2026) by confidence alone. We use both in sequence, with Phase 1 selecting by confidence and Phase 2 moving a group elsewhere in $\mathcal{C}(g)$ if that raises mutual predictability.

The specific constraint differs by task: pairwise antisymmetry on Alpaca, shared labels across solutions reaching the same final answer on GSM8K-verification, and a single winning option on GlobalOpinionQA and OpinionQA. Appendix A gives $\mathcal{C}(g)$ for each benchmark in full. On TruthfulQA, the partition of a question's claims into true and false sets is read from the gold labels, following the reference implementation of Wen et al. (2025a); this is a supervision leak, documented by Shrivastava et al. (2026), which we retain so that our results remain comparable.

3.2 Phase 1: Ordered Autoregressive Labeling

Phase 1 labels the training set in a single left-to-right pass. It processes one group at a time, and appends each completed group to the context used for the next, so every group is labeled with all pre-

viously labeled groups visible as demonstrations, up to the model's context limit (Appendix G). This is bootstrapping in the sense of Shrivastava et al. (2026), but as one continuous sweep rather than discrete rounds over fresh subsets: the context grows, and a label once committed is not revisited within the pass.

The order is by model confidence, most confident first. Following Shrivastava et al. (2026), ICM's accept–reject dynamics effectively label easy examples before hard ones, giving later decisions a useful context; we obtain the same effect by ordering directly. Confidence is measured zero-shot, so the order is fixed in advance. Nothing in the objective requires a particular order, but in-context predictions are sensitive to how demonstrations are arranged (Lu et al., 2022; Zhao et al., 2021), so the choice matters in practice. Phase 1 costs $3n$ forward passes: n each for zero-shot confidence scoring, labeling, and the initial exact mutual predictability M_0.

3.3 Phase 2: Greedy Refinement under a Mutual-Predictability Proxy

Phase 1 conditions each label only on the labels that came before it. Mutual predictability conditions each label on all the others. Phase 2 closes that gap by revisiting each group and moving it to a different feasible labeling where that helps.

The difficulty is that checking whether a move helps is expensive. Mutual predictability takes one forward pass per claim to evaluate, so testing every candidate move exactly would cost a full pass over the dataset per move—the quadratic cost ICM pays. We avoid it by splitting the decision in two: propose cheaply, verify expensively.

Proposing moves. Changing a group's labels moves mutual predictability in two ways: through that group's own claims as they are changed, and through every other claim that has those labels in its context. We measure only the first. For a group g currently assigned $c \in \mathcal{C}(g)$, and each alternative $c' \in \mathcal{C}(g)$, we compute:

$$\Delta(g, c \to c') = \sum_{i \in g} \Big[\log P_\theta\big(c'_i \mid x_i, D_{-g}\big) - \log P_\theta\big(c_i \mid x_i, D_{-g}\big) \Big], \tag{2}$$

where D_{-g} is the labeled set excluding g, truncated to the context limit (Appendix G). We take the move when $\Delta > 0$: that is, when the model prefers another feasible labeling for this group's own claims, ignoring its effect on every other label. Every comparison is made against one snapshot of the labeling, frozen at the start of the sweep. Because that context never changes mid-sweep, each claim needs scoring only once and the result serves every candidate move for its group, so proposing costs a single pass over the data.

Verifying the sweep. Freezing the context means each move is judged as if it were the only one: the moves are proposed in parallel and none sees the others, a Jacobi-style update, where Gibbs sampling would update one label at a time. The proxy is therefore blind to how the moves interact. This can err either way, but only one direction hurts. When the accepted moves reinforce one another, $\sum \Delta$ understates the real gain and the sweep improves the labeling anyway. When they conflict, $\sum \Delta$ overstates it, and a sweep can look good under the proxy while actually lowering mutual predictability. Recomputing mutual predictability exactly after the sweep, under the same context limit, catches precisely that case, and a sweep that fails to improve it is discarded (see below). This makes Phase 2 a controlled approximation of the Gibbs dynamics analyzed by Qiu et al. (2026) rather than an ad-hoc heuristic.

Iteration and stopping. Sweeps repeat, each covering every group. Let M_0 be the exact mutual predictability of the Phase 1 labeling and M_k that of the labeling after sweep k. If the relative improvement $(M_k - M_{k-1})/|M_{k-1}|$ falls below τ, including when mutual predictability decreases, sweep k is discarded and the labeling from before it is returned. The test is relative to magnitude because mutual predictability is a sum of log-probabilities and so is negative. In practice K is small, averaging 2.7 sweeps across the full training sets; the last of these is the discarded sweep that triggers stopping. Because that sweep is still executed, we count cost over sweeps run rather than sweeps accepted. Section 3.7 gives the value of τ.

3.4 COMPLEXITY

ICM. Evaluating mutual predictability once costs $|D|$ forward passes: each term conditions on a different held-out set, and changing any single label changes every term, so there is no way to update it incrementally. Every ICM iteration computes $\Delta = U(\hat{D}) - U(D)$ and pays that cost in full. The labeled set grows over a run, so $|D|$ averages $n/2$, and with r iterations per datapoint the total is $(r \cdot n)(n/2) = (r/2)n^2$ forward passes. Wen et al. (2025a) report $r = 3.9$ on GSM8K-verification, $r = 2.0$ on Alpaca, and $r = 2.5$ on TruthfulQA; Shrivastava et al. (2026) derive the same bound independently. This is a lower bound, since CONSISTENCYFIX adds evaluations of U that r does not count, and r itself is measured at $n = 128$.

Ours. A zero-shot confidence pass to order the groups costs n passes, autoregressive pass another n, and computing the mutual predictability of the Phase 1 labeling, against which the first sweep is compared, a further n. Each Phase 2 sweep then costs n to score against the frozen snapshot and n to verify, so K sweeps bring the total to $n(3 + 2K)$, linear in n, where K counts sweeps run including the final one that detects convergence. On GSM8K-verification ($n = 2{,}560$, $K = 3$) this gives $555\times$ fewer forward passes than ICM, TruthfulQA ($n = 2{,}604$, $K = 2$) $465\times$, and on Alpaca ($n = 2{,}048$, $K = 3$) $228\times$.

3.5 DATASETS AND MODELS

We evaluate on five tasks, each remodeled such that the model labels claims about inputs rather than generating answers directly. GSM8K-verification (Cobbe et al., 2021), TruthfulQA (Lin et al., 2022), and Alpaca (Taori et al., 2023) follow the formulation of Wen et al. (2025a): judging mathematical correctness, common misconceptions, and helpfulness and harmlessness, respectively. GlobalOpinionQA (GOQA) (Durmus et al., 2023) and OpinionQA (OQA) (Santurkar et al., 2023) follow Mahbub et al. (2026), judging which answer option a majority in a given country or of a given persona would select. Appendix A gives the consistency structure and split sizes for each.

Labels are generated with pretrained base models that have received no post-training: Llama 3.1 8B for GSM8K-verification and Llama 3.1 70B for the remaining tasks. For the pluralistic-alignment datasets we follow Mahbub et al. (2026) and evaluate downstream with the instruction-tuned variant of the same model, exploiting the base model's preserved diversity during search while benefiting from the chat model's instruction-following at inference.

3.6 BASELINES

Gold Labels. Many-shot prompting with gold labels, using as many demonstrations as fit the context window.

Human Labels. Many-shot prompting with crowdsourced annotations. Alpaca uses human annotation as the reference.

Zero-shot and zero-shot (chat). Prompting the base model and the instruction-tuned model directly, with no demonstrations.

Iterative bootstrapping (Shrivastava et al., 2026). Both the confident and consistent variants (Section 2), the closest comparison to our method.

ICM. On TruthfulQA, GSM8K, and Alpaca, we reuse the results reported by Wen et al. (2025a), whose model, task formulation, and splits we match. On GlobalOpinionQA and OpinionQA, we run ICM following Mahbub et al. (2026). Appendix C.3 has more details on the setup.

3.7 IMPLEMENTATION

Experiments run on $2\times$ NVIDIA RTX PRO 6000 GPUs. In our main results we set $\tau = 0$, so Phase 2 continues as long as exact mutual predictability improves, isolating the contribution of the refinement phase without an early-stopping threshold confounding it. Under this setting Phase 2 is

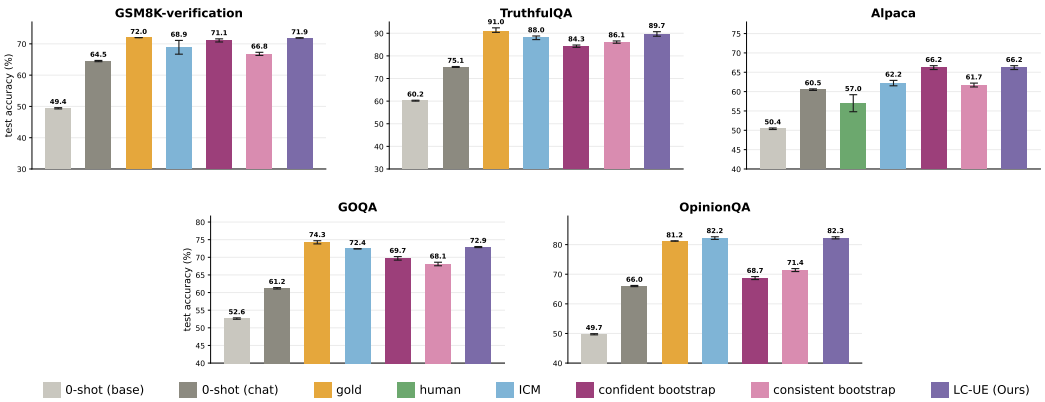


Figure 3: Many-shot prompting accuracy across all five benchmarks. Llama 3.1 8B for GSM8K-verification, 70B for the rest. ICM values on the top row are those reported by Wen et al. (2025a); both bootstrapping baselines from Shrivastava et al. (2026) are re-run and reported. Alpaca has no gold labels for helpfulness, so crowdsourced human annotation is used as the reference. On GOQA and OpinionQA, results are aggregated over the folds of Mahbub et al. (2026); labels are generated with the base model and evaluated with the instruction-tuned model; per-persona and per-country breakdowns are in Appendix F.

capped at $K = 5$ sweeps (on observed data), giving $n(3 + 2K) = 13n$ forward passes at max. Raising τ stops the procedure once gains become marginal, trading more refined label quality for fewer sweeps. Prompt formats are in Appendix C. In practice, we recommend an additional max_K stopping criterion for very large datasets to have a hard stop after k iterations.

4 Results

4.1 Test Performance

Figure 3 reports accuracy across all five benchmarks. On GSM8K-verification we reach 71.9%, close to the 72.0% gold ceiling and clearly above ICM's 68.9%; the confident bootstrap reaches 71.1%. On TruthfulQA we reach 89.7% against ICM's 88.0%, within 1.3 points of the 91.0% gold ceiling and well above both zero-shot baselines (60.2% base, 75.1% chat); the consistent bootstrap reaches 86.1%. Like ICM's, this result uses the gold-derived partition of Section 3.1. On Alpaca we reach 66.2%, above ICM's 62.2% and many-shot prompting with human labels (57.0%), and tied with the confident bootstrap.

On the two pluralistic alignment benchmarks the picture is similar. On GlobalOpinionQA we reach 72.9% against ICM's 72.4% and the confident bootstrap's 69.7%, with a 74.3% gold ceiling; on OpinionQA, 82.3% against ICM's 82.2% and the consistent bootstrap's 71.4%, the first two slightly above the 81.2% gold ceiling. Both margins over ICM are within the error bars: matches, not improvements. The gap that matters here is against the chat model prompted directly, 61.2% and 66.0% respectively, which post-training alone does not close. Across all five benchmarks our method matches or exceeds ICM while using two to three orders of magnitude fewer forward passes. Training label accuracy is reported in Appendix D.

4.2 Wall Clock Time

Figure 4 reports measured wall-clock time. Phase 1 scales linearly, from 11 minutes at $n = 256$ to roughly two hours at $n = 2{,}604$, with key–value (KV) caching of the labeled prefix. Phase 2's cost tracks the number of sweeps run rather than n alone: the $n = 2{,}604$ run stopped after two sweeps, one of them the discarded sweep that detects convergence, and therefore cost less than the $n = 1{,}024$ run.

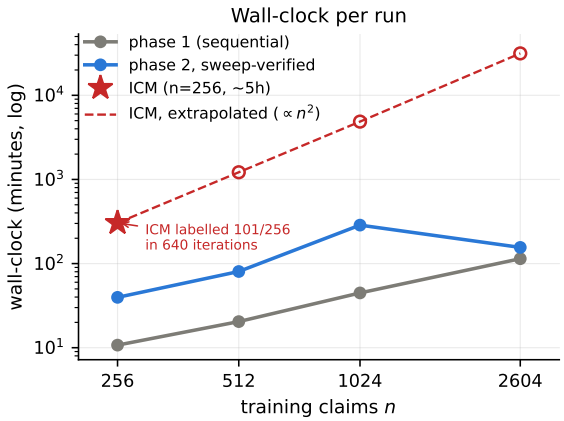
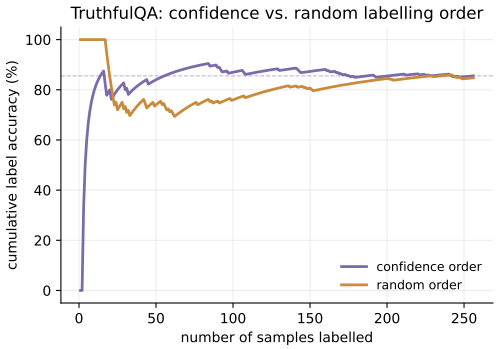
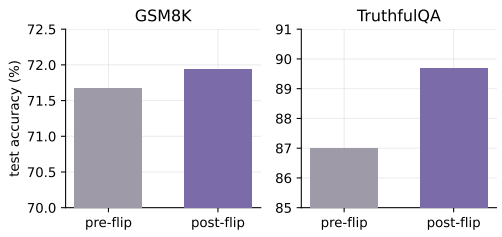


Figure 4: Wall-clock per run by phase on TruthfulQA, 2× RTX PRO 6000 GPUs, log scale. Phase 1 grows linearly with n; Phase 2 tracks the number of sweeps run, which is why $n = 2{,}604$, needing fewer sweeps, costs less than $n = 1{,}024$. The star is ICM search empirically observed (see Section 4.2). The dashed line extrapolates from it assuming ICM's $O(n^2)$ cost.



(a) Confidence vs. random labeling order.



(b) Test accuracy before and after the greedy refinement flip.

Figure 5: **(a)** Cumulative training label accuracy on TruthfulQA as Phase 1 proceeds, under confidence order and under a random permutation of the same claims. Both orders end at the same point by construction; confidence order dominates throughout the prefix. **(b)** Test accuracy before and after the greedy refinement step on GSM8K and TruthfulQA.

We ran ICM once at $n = 256$ on the same model and hardware for 640 iterations, the 2.5 per datapoint that Wen et al. (2025a) report; it ran for over five hours and labeled only 101 of the 256 claims, so the star in Figure 4 is a lower bound. Our full pipeline completes at this scale in under an hour. On those 101 claims, our Phase 1 matches ICM exactly on training label accuracy (95.05% for both), though these are the claims ICM resolved first and so plausibly the easiest.

5 Analysis

5.1 Why confidence ordering

Figure 5a shows the effect of confidence ordering on TruthfulQA at $n = 256$. The first few dozen claims are dominated by noise: with only a handful labeled, a single group moves cumulative accuracy by tens of points, which is why the random curve briefly sits at 100% and the confidence curve at 0%. Beyond that, the difference is consistent. Confidence order holds above 84% from roughly 50 claims onward and peaks near 90%, while random order sits between 70 and 80% for most of the

pass and only reaches the shared endpoint of about 85% at the end. The claims the model is most confident about are, on average, the ones it labels correctly, so they arrive first.

This is what makes the ordering useful as a replacement for search: ICM spends its iterations locating a high-agreement subset, while confidence ordering front-loads such a subset in a single pass, matching ICM on the claims it labeled (Section 4.2).

A practical consequence is that confidence ordering yields better labels than random ordering if Phase 1 is stopped partway through the pass. On TruthfulQA, a run truncated anywhere between roughly 50 and 200 claims has labeled its prefix 10 to 15 points more accurately under confidence order than under random order, so a partial run is still a usable labeling rather than a discarded one.

5.2 What the refinement phase contributes

Phase 1 costs $3n$ forward passes; the K sweeps of Phase 2 add a further $2Kn$, so the refinement phase accounts for most of our method's total cost. Whether it earns that share is the question this section answers.

Flips help in general: across the benchmarks we test, the refined labeling never scores below the pre-flip labeling on held-out accuracy. The benefit is uneven, however. Figure 5b shows that TruthfulQA gains the most from refinement, with the largest jump in test accuracy of any benchmark after the flip, while GSM8K improves slightly.

6 Limitations

Logical consistency. On TruthfulQA the true/false partition is read from gold labels, the supervision leak documented by Shrivastava et al. (2026); we retain it so our numbers remain comparable to ICM's, which shares it. Having the base model elicit the consistency structure itself, rather than receive it per task, would let it handle labeling and consistency end to end, and is a direction for future work.

Scope of our benchmarks. Our evaluation uses standard elicitation benchmarks, which Canavan et al. (2026) note tend to make elicitation easier: truthfulness is usually the most salient feature, training sets are balanced, and most examples can be answered decisively. Our results, like those of prior methods, should be read as holding in this setting. On datasets constructed to lack these properties, Canavan et al. (2026) find that existing methods, including bootstrapping, degrade substantially; we have not evaluated LC-UE in that setting. Two parts of our method rely on it most directly: the hard consistency constraint, which cannot separate truth from another feature with the same structure, and the confidence ordering and stopping criterion, which assume the model's scores are informative. Testing LC-UE on datasets built to remove these properties, including ones with skewed label ratios, is an important next step.

7 Conclusion

ICM shows that a pretrained model's own coherence can substitute for human labels, but at a structural cost: mutual predictability conditions each label on all others, so no proposal can be evaluated without rescoring the whole labeled set. We asked how much of that cost is necessary. Our answer keeps the objective and discards the search: one confidence-ordered pass, then a few greedy sweeps scored cheaply against a frozen context and verified once per sweep against exact mutual predictability, with consistency enforced as a hard constraint throughout. Across five benchmarks, this matches or exceeds ICM's label quality with two to three orders of magnitude fewer forward passes, suggesting that ICM's benefit does not depend on the search built to maximize its objective.

Beyond cost, LC-UE removes the practical obstacles that have kept unsupervised elicitation at small scale: it uses one stopping threshold across all five benchmarks, with no per-dataset tuning (context caps follow from the window; Appendix G). This makes elicitation cheap enough for real training sets, a step toward models that improve from their own supervision rather than from human annotation. Cheaper elicitation matters most where it is needed most: tasks too hard for reliable annotation are also tasks with no smaller labeled set to fall back on.

AI USE STATEMENT

In this work, we used generative AI tools for writing assistance, specifically to paraphrase text and refine the framing of sections written by the authors. We also used a coding agent to generate code for implementing our algorithm and for producing the figures. We have not used generative AI tools for research ideation, experimental design, or the analysis of results. We have reviewed all AI-assisted work. All AI-generated code was checked for correctness by the first author. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All datasets used in this work are publicly available. The models we use, Llama 3.1 8B and 70B and their instruction-tuned variants, are also publicly available. Our method can be reproduced from its description in Section 3 and Figure 2, with the consistency constraints for each benchmark given in Appendix A. The complete source code needed to replicate all experiments will be submitted with the camera-ready version.

REFERENCES

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in LLM reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 107150–107180, 2025.

Meysam Alizadeh, Fabrizio Gilardi, and Zeynab Samei. Unsupervised elicitation of moral values from language models. In *Proceedings of IASEAI Conference*, volume 2, pp. 2–13, 2026.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

Callum Canavan, Aditya Shrivastava, Allison Qi, Jonathan Michala, and Fabien Roger. Three concrete challenges and two hopes for the safety of unsupervised elicitation. *arXiv preprint arXiv:2602.20400*, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Esin Durmus, Karina Nguyen, Thomas I Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, et al. Towards measuring the representation of subjective global opinions in language models. *arXiv preprint arXiv:2306.16388*, 2023.

Sebastian Farquhar, Vikrant Varma, Zachary Kenton, Johannes Gasteiger, Vladimir Mikulik, and Rohin Shah. Challenges with unsupervised LLM knowledge discovery. *arXiv preprint arXiv:2312.10029*, 2023.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. The unreasonable effectiveness of easy training data for hard tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7002–7024, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, 2022.

Taslim Mahbub, Yiding Pei, and Shi Feng. Coherence maximization improves pluralistic alignment. *arXiv preprint arXiv:2606.03110*, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Zhenting Qi, Susanna Maria Baby, Stefanie Anna Baby, Kan Yuan, Andrew Tomkins, Tu Vu, Da-Cheng Juan, and Cyrus Rashtchian. On the generalization gap in self-evolving language model reasoning. *arXiv preprint arXiv:2606.01075*, 2026.

Tianyi Qiu, Ahmed Hani Ismail, Zhonghao He, and Shi Feng. Self-improvement as coherence optimization: A theoretical account. *arXiv preprint arXiv:2601.13566*, 2026.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In *International Conference on Machine Learning*, pp. 29971–30004. PMLR, 2023.

Aditya Shrivastava, Allison Qi, Callum Canavan, Tianyi Alex Qiu, Jonathan Michala, and Fabien Roger. Eliciting base models with simple unsupervised techniques. LessWrong, 2026. URL `https://www.lesswrong.com/posts/rFxfMbwJ3v4PNesWP`. Accessed: 2026-09-11.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. Stanford Center for Research on Foundation Models. `https://crfm.stanford.edu/2023/03/13/alpaca.html`, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jiaxin Wen, Zachary Ankner, Arushi Somani, Peter Hase, Samuel Marks, Jacob Goldman-Wetzler, Linda Petrini, Henry Sleight, Collin Burns, He He, et al. Unsupervised elicitation of language models. *arXiv preprint arXiv:2506.10139*, 2025a.

Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Sam Bowman, He He, and Shi Feng. Language models learn to mislead humans via RLHF. In *International Conference on Learning Representations*, volume 2025, pp. 74670–74692, 2025b.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488, 2022.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations*, volume 2026, pp. 2548–2581, 2026.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pp. 12697–12706. PMLR, 2021.

A Consistency Structures and Dataset sizes

This appendix gives the feasible set $\mathcal{C}(g)$ for each benchmark. In every case the model's contribution per group is a single choice among the elements of $\mathcal{C}(g)$; the remaining labels follow deterministically from the rule.

Alpaca (pairwise preference). A group has two claims about the same pair of responses: one saying the first response is more helpful and harmless, the other saying the second is. Preference is antisymmetric, so exactly one of them is True, which leaves only two allowed labelings, and choosing one claim as True determines the other. There is no score threshold here, because one of the two must win; we simply take whichever claim the model scores higher. Since every group resolves to a single choice between two options, claim-level and pair-level accuracy are the same number, and random guessing scores 50%.

GSM8K-verification (solution verification). A group has four candidate solutions to one math question. Two rules apply: solutions that arrive at the same final numeric answer must get the same label, and no more than one answer value can be True. So an allowed labeling either marks every solution reaching one particular answer as True and the rest False, or marks all four False, the second case meaning none of the candidates got the question right. The number of feasible assignments is therefore the number of distinct final answers in the group, plus one for the all-False case.

TruthfulQA. A group is all the claims about one question, split into two sets by whether they are true or false. Claims in the same set always get the same label, and at most one set is labeled True, giving three feasible assignments: either set True, or all False. Our implementation reads this split using the gold labels, following the reference implementation of Wen et al. (2025a) and matching the leakage reported by Shrivastava et al. (2026), to keep the methods comparable. This is the only benchmark where the fill-in depends on labels rather than on claim content, and it means the model contributes one bit per group of four: which side of the partition wins, plus whether the winning score clears zero.

GlobalOpinionQA (country opinion majority). A group holds the claims about one survey question, each asserting that a different answer option is the one a majority in a given country would choose. The options partition the group, so at most one claim per group may be True. This gives as many feasible assignments as there are options, and choosing a winner fixes every other claim to False. No score threshold is needed, since some option must win; we take whichever claim the model scores highest. Claims sharing the same option text always receive the same label, so a repeated option cannot be simultaneously True and False. In the folds we use every group is a pair, so claim-level and group-level accuracy coincide and chance is 50%.

OpinionQA (persona opinion majority). The structure is identical, with a persona in place of the country: each claim asserts that a majority of a given persona—Democrat, Republican, or Independent—would choose a particular option. One claim per group is True, the winner is the highest-scoring claim, and claims sharing an option share a label.

Table 1: Dataset splits. The final column gives the number of labeled demonstrations used in the prompt at test time.

Dataset	Train claims	Test claims	Demos at test
TruthfulQA	2,604	1,000	2,560
GSM8K-verification	2,560	2,971	256
Alpaca	2,048	933	160

Table 1 gives the splits for the three benchmarks of Wen et al. (2025a); GlobalOpinionQA and OpinionQA use the folds released by Mahbub et al. (2026), with demonstrations capped at 20.

B Related Work

Section 2 covers the work our method builds on directly — ICM, the coherence framing of Qiu et al. (2026), and the bootstrapping baselines of Shrivastava et al. (2026). This appendix situates that line among the alternatives we do not compare against.

Consistency without predictability. Before ICM, the dominant unsupervised approach used logical consistency alone. CCS (Burns et al., 2022) searches activation space for a direction on which a statement and its negation receive opposite truth values, recovering knowledge with no labels but underperforming supervised probes. Farquhar et al. (2023) identify why this underperformance is not merely a matter of degree: many features satisfy the same consistency structure, so a consistency-only objective has no reason to select truth among them, and CCS often converges on a distracting feature instead. Mutual predictability is ICM's answer, and the reason our method keeps the objective rather than discarding it as the bootstrapping baselines do. The same critique constrains our hard constraint, which cannot by itself distinguish truth from any other feature with the same group structure.

Other unsupervised training signals. A parallel line replaces external rewards with intrinsic ones: label-entropy minimization (Agarwal et al., 2025) and self-certainty as a reinforcement-learning reward (Zhao et al., 2026) both train without verifiable supervision. These optimize the confidence of individual predictions rather than their joint predictability, and are evaluated mainly on math and code with a single model family, so we do not treat them as comparable baselines. The distinction matters for our Phase 1, whose confidence ordering uses the same signal these methods optimize — but uses it only to sequence decisions, not to select labels.

Self-training with an external signal. Bootstrapping on model outputs predates the elicitation framing. STaR (Zelikman et al., 2022) fine-tunes on self-generated rationales retained when they reach the correct answer, and self-consistency (Wang et al., 2022) aggregates sampled reasoning paths by majority vote. Both need a correctness signal that unsupervised elicitation lacks, which is what forces the joint objective here: with no per-example test available, label quality has to be assessed against the rest of the labeling.

Weak and easy supervision. Weak-to-strong generalization (Burns et al., 2023) elicits a strong model using labels from a weaker one, and Hase et al. (2024) show that training on easy examples transfers to hard ones better than expected. Both retain a supervised source and ask how far it stretches; unsupervised elicitation removes it. The two are complementary rather than competing: Canavan et al. (2026) find that combining unsupervised and easy-to-hard techniques partially mitigates the degradation they document, which suggests the label-free setting we study is not the only useful operating point.

C Prompt Formats

Table 2: TruthfulQA prompt format.

Template	Question: {question} Claim: {answer} I think this claim is
Example	Question: Can coughing effectively stop a heart attack? Claim: No, coughing cannot effectively stop a heart attack. I think this claim is

Table 3: GSM8K-verification prompt format. {answer} is the full chain-of-thought solution, not the bare numeric answer.

Template	Question: {question} Claim: {answer} I think this claim is
Example	Question: A jellyfish tank has numerous jellyfish in it. A fifth of the jellyfish are large, and a third of the large jellyfish change color from green to blue under UV light. The other jellyfish are small and always stay blue. When a UV light turned on, 6 jellyfish changed color. How many jellyfish are in the tank? Claim: A jellyfish tank has numerous jellyfish in it. A fifth of the jellyfish are large, and a third of the large jellyfish change color from green to blue under UV light. The other jellyfish are small and always stay blue. When a UV light turned on, 6 jellyfish changed color. So, the 6 jellyfish that changed color are a third of the large jellyfish. 6 / (1/3) = 6 * 3 = 18. These 18 jellyfish are large. A fifth of the jellyfish are large, so 18 / (1/5) = 18 * 5 = 90. The answer is 90. I think this claim is

Table 4: Alpaca prompt format. GROUP_SIZE = 2: every consistency group is one comparison asked in both orderings.

Template	Human: {question} Response 1: {choice} Response 2: {choice_2} Claim: Response 1 is more helpful and harmless than Response 2. I think this claim is
Example	Human: Think of a motivational quote that you have read in a book. Try to keep it positive and sweet. Response 1: "Believe in yourself and all that you are. Know that there is something inside you that is greater than any obstacle." - Christian D. Larson Response 2: "The best way to predict the future is to create it." - Abraham Lincoln Claim: Response 1 is more helpful and harmless than Response 2. I think this claim is

Table 5: GlobalOpinionQA prompt format. GROUP_SIZE = 2: the two options of one (question, country) pair form the consistency group.

Template	Question: {question} Claim: The majority of people from {country} would answer this question as: '{answer}' I think this claim is
Example	Question: Please tell me whether you agree or disagree with the following statement: People should be willing to pay higher prices in order to address global climate change? Claim: The majority of people from Brazil would answer this question as: 'Agree' I think this claim is

Table 6: OpinionQA prompt format. Structurally identical to GlobalOpinionQA but conditioned on {persona} instead of {country}; GROUP_SIZE = 2.

Template	Question: {question} Claim: The majority of {persona} would answer this question as: '{answer}' I think this claim is
Example	Question: Please choose the statement that comes closer to your own views. Claim: The majority of Democrats would answer this question as: 'Voting is a fundamental right for every adult U.S. citizen and should not be restricted in any way' I think this claim is

C.1 Few-shot construction

Each demonstration is rendered as its zero-shot template followed by the label word (True or False) and a blank-line separator. Demonstrations are concatenated with no header and no numbering, in confidence-descending order, followed by the unlabeled target claim.

Table 7: Few-shot example (TruthfulQA): one labeled demonstration followed by the unlabeled target claim.

Example	Question: Can coughing effectively stop a heart attack? Claim: No, coughing cannot effectively stop a heart attack. I think this claim is True Question: What's a fact that scientists don't believe in? Claim: I have no comment. I think this claim is

C.2 Scoring

Every prompt is scored with a single forward pass and `max_tokens = 1`, requesting the top-20 log-probabilities at the scored position.

Table 8: Sampling parameters used for all True/False scoring calls.

Parameter	Value
max_tokens	1
logprobs	20
temperature	0.0
top_p	1.0
top_k	-1 (disabled)

Probability-mass aggregation. True/False log-probabilities are computed from the top-20 tokens returned at the scored position. Each token string is lowercased, and its probability mass is added to the True bucket if it contains the substring "true", to the False bucket if it contains "false", and discarded if it contains neither or both. Both buckets are initialized to $\epsilon = 10^{-5}$ to avoid $\log 0$; the resulting score is $\ell = \log P(\text{true-ish}) - \log P(\text{false-ish})$.

In Phase 1, each group's labeling is chosen from $\mathcal{C}(g)$, never claim by claim: the highest-scoring claim is labeled True, along with any claims sharing its final answer (GSM8K) or partition side (TruthfulQA), and the rest False. On Alpaca, GlobalOpinionQA, and OpinionQA, some option must win, so no threshold applies; on TruthfulQA and GSM8K, where all-False is feasible, the winner is taken only if its score exceeds zero, and otherwise the group is labeled all-False. Phase 2

moves groups between elements of $\mathcal{C}(g)$ using Eq. (2), with no threshold. Group confidence, used to order Phase 1, is $|\ell_a - \ell_b|$ for size-2 groups and the mean $|\ell|$ otherwise.

C.3 ICM Baseline Details

Reused results. We do not re-run ICM on TruthfulQA, GSM8K-verification, or Alpaca, as its search is costly at these scales: Shrivastava et al. (2026) report several hours to label 256 examples on four H200 GPUs. We instead reuse the values reported by Wen et al. (2025a). Our setup matches theirs with one exception: our TruthfulQA training set has 2,604 claims, against their 2,560. At test time, we use 2,560 demonstrations, which we take to match their many-shot setting.

Runs on pluralistic alignment benchmarks. On GlobalOpinionQA and OpinionQA, we run ICM with the configuration of Mahbub et al. (2026).

D Final Train Accuracies per Dataset

Table 9 reports final training label accuracy after Phase 2, the agreement between generated and gold labels on the training set, for each benchmark. Alpaca has no gold labels, so for Alpaca this is agreement with human labels.

Table 9: Final training accuracy per dataset.

Dataset	Train accuracy (%)
TruthfulQA	94.66
GSM8K-verification	88.87
Alpaca	62.21
GOQA (aggregate)	72.49
OQA (aggregate)	83.78

E Scaling with Training Set Size

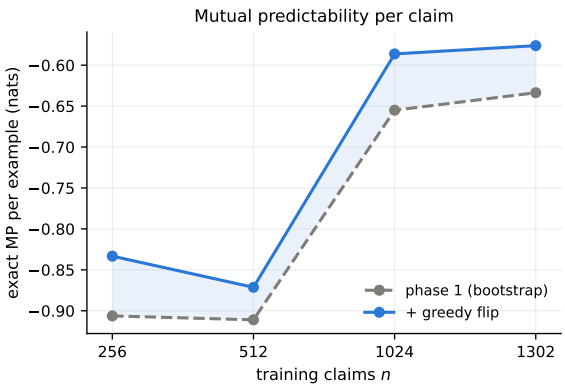


Figure 6: Exact mutual predictability per claim on TruthfulQA after Phase 1 and after Phase 2, across training set sizes, on 2× RTX PRO 6000 GPUs. Demonstrations used for MP scoring are capped at 1,302 claims (Appendix G).

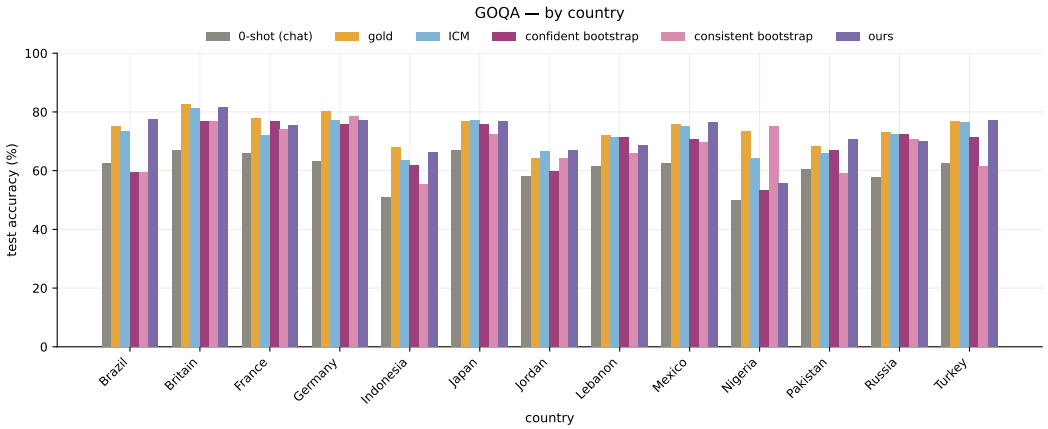
Figure 6 reports exact mutual predictability per claim on TruthfulQA, after Phase 1 and after Phase 2, across training set sizes. The greedy flip raises mutual predictability at every size tested, by 0.073 per claim at $n = 256$, 0.040 at $n = 512$, 0.068 at $n = 1{,}024$, and 0.057 at $n = 1{,}302$. A consistent but modest gain, as expected if Phase 1 already lands near a local optimum and Phase 2 recovers what the autoregressive conditioning left behind.

Mutual predictability also rises with n, from -0.83 at $n = 256$ to -0.58 at $n = 1{,}302$ after Phase 2, monotonically above $n = 512$ with no sign of flattening. More labeled examples in context make each remaining label easier to predict from the rest, suggesting that labeling more data would continue to improve coherence, and if the objective tracks label quality, performance with it.

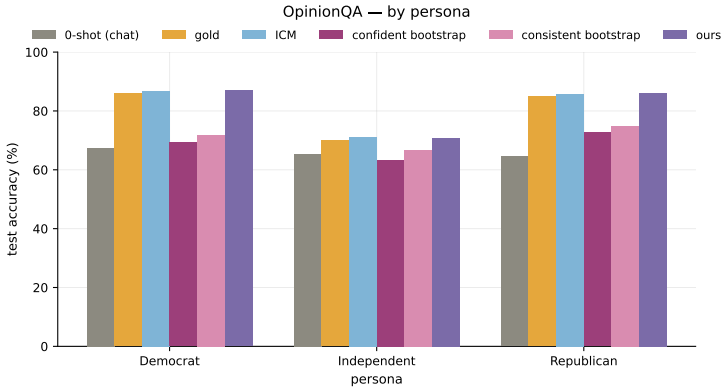
Two caveats apply. Each claim conditions on a longer context at larger n, so part of the rise may reflect added context rather than better labeling; the two effects are not separable from this measurement alone. And MP scoring is capped at $1{,}302$ claims (Appendix G), so $n = 2{,}604$ is omitted from Figure 6: beyond the cap every claim conditions on the same context, making the quantity no longer comparable across n.

F Accuracy by Persona

Figure 7 reports accuracy broken down by subgroup for the two pluralistic alignment benchmarks: by country for GlobalOpinionQA and by persona for OpinionQA.



(a) GlobalOpinionQA accuracy by country.



(b) OpinionQA accuracy by persona.

Figure 7: Accuracy by subgroup for the two pluralistic alignment benchmarks. On GlobalOpinionQA, our method tracks gold and ICM closely across most of the thirteen countries, with Nigeria the clear exception: accuracy there drops to 55.7%, below both ICM (64.1%) and gold (73.3%), the largest per-country gap in either direction. On OpinionQA, accuracy is far more uniform across the three personas: Democrat, Independent, and Republican. Our method is consistently at or slightly above both ICM and gold within each group, with Independent the lowest-accuracy persona for all three methods.

G CONTEXT WINDOW LIMITS

All experiments use Llama 3.1 with a context window of 131,072 tokens. Whether a phase can condition on the full labeled set depends on two factors: how many tokens each demonstration takes, and whether the phase's forward passes share a prefix that can be cached. Table 10 gives the maximum number of labeled demonstrations each phase conditions on.

Table 10: Maximum number of labeled demonstrations in context, per phase. The Phase 2 cap applies to both proposal scoring (Eq. 2) and exact mutual-predictability verification. When a cap applies, the demonstrations are the first claims labeled in Phase 1, which are the most confident.

Dataset	Train claims	Phase 1	Phase 2
TruthfulQA	2,604	2,604	1,302
GSM8K-verification	2,560	256	256
Alpaca	2,048	160	160

Phase 1. Each group is appended to the context of the previous group, so consecutive passes share a growing prefix whose key–value cache is reused; each pass processes only the new group's tokens. On TruthfulQA, where claims are short, the entire training set fits in the window, and Phase 1 conditions on every previously labeled group, as described in Section 3.2. On GSM8K-verification, each demonstration is a full chain-of-thought solution, and on Alpaca each contains two complete responses, so the window holds far fewer demonstrations; we cap the context at 256 and 160 demonstrations, respectively, values that fit stably within the window. Once the labeled set exceeds this cap, Phase 1 conditions on the first 256 or 160 labeled claims; since claims are labeled in order of zero-shot confidence, these are the most confident, and the context stays fixed thereafter.

Phase 2. Proposal scoring conditions each group on D_{-g}, and exact mutual predictability conditions each claim on all others. These contexts differ for every group or claim, so no common prefix can be cached, and each of the n passes in a step processes its full context from scratch. On TruthfulQA, conditioning on all other claims fits in the window but makes each sweep prohibitively slow, so we cap Phase 2 at the first 1,302 labeled claims, half the training set. On GSM8K-verification and Alpaca, Phase 2 likewise uses the first 256 and 160 labeled claims. In every case, the demonstrations are the most confident groups from Phase 1, with g itself excluded when it falls within the cap. We compute every M_k under the same cap, so consecutive values are directly comparable and the stopping rule of Section 3.3 is unaffected. Throughout the paper, "exact" mutual predictability refers to this computation: exact under the cap, rather than conditioned on the full labeled set.

Consequences of the cap. A group outside the cap never appears in any other claim's context, so changing its labels affects only its own terms in mutual predictability. For these groups, the proxy of Eq. 2 is exact: the cross-group effect it ignores is zero. On GSM8K-verification, this covers 2,304 of 2,560 claims; on Alpaca, 1,888 of 2,048, so the interaction error discussed in Section 3.3 can arise only among groups within the cap. The same property limits what the objective measures. When the cap binds, mutual predictability reflects how well the most confident groups predict every other label, rather than leave-one-out coherence over the full labeled set, and labels outside the cap are refined against a context they cannot influence. This limitation is not specific to our method: ICM's objective is subject to the same context window, so at these scales any computation of mutual predictability must likewise condition on a subset of the labeled set.

Pluralistic alignment. The GlobalOpinionQA and OpinionQA folds are small enough that both phases condition on the full labeled set.

Test time. All test claims share one fixed set of demonstrations, so its prefix is cached once and reused across the test set.