

License: arXiv.org perpetual non-exclusive license  
arXiv:2506.10139v1 [cs.CL] 11 Jun 2025

# Unsupervised Elicitation of Language Models

Jiaxin Wen<sup>1</sup>, Zachary Ankner<sup>1</sup>, Arushi Somani<sup>1</sup>,

**Peter Hase<sup>2</sup>, Samuel Marks<sup>1</sup>, Jacob Goldman-Wetzler<sup>1</sup>, Linda Petrini<sup>3</sup>, Henry Sleight<sup>4</sup>  
Collin Burns<sup>1</sup>, He He<sup>5</sup>, Shi Feng<sup>6</sup>, Ethan Perez<sup>1</sup>, Jan Leike<sup>1</sup>**

<sup>1</sup>

Anthropic <sup>2</sup>Schmidt Sciences <sup>3</sup>Independent <sup>4</sup>Constellation

<sup>5</sup>New York University <sup>6</sup>George Washington University

## Abstract

To steer pretrained language models for downstream tasks, today’s post-training paradigm relies on humans to specify desired behaviors. However, for models with superhuman capabilities, it is difficult or impossible to get high-quality human supervision. To address this challenge, we introduce a new unsupervised algorithm, Internal Coherence Maximization (ICM), to fine-tune pretrained language models on their own generated labels, *without external supervision*. On GSM8k-verification, TruthfulQA, and Alpaca reward modeling tasks, our method matches the performance of training on golden supervision and outperforms training on crowdsourced human supervision. On tasks where LMs’ capabilities are strongly superhuman, our method can elicit those capabilities significantly better than training on human labels. Finally, we show that our method can improve the training of frontier LMs: we use our method to train an unsupervised reward model and use reinforcement learning to train a Claude 3.5 Haiku-based assistant. Both the reward model and the assistant outperform their human-supervised counterparts.

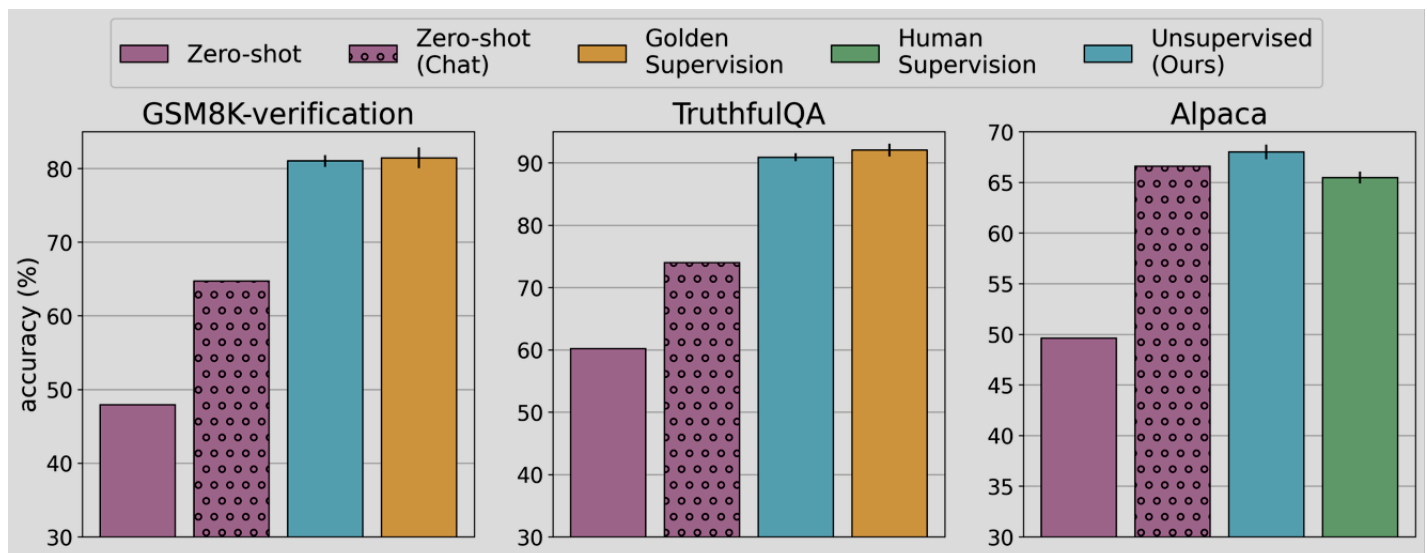


Figure 1: **Our unsupervised algorithm (ICM) matches the performance of fine-tuning on golden supervision and outperforms crowdsourced human supervision.** We report average test accuracy and variance across three runs on three classification tasks: mathematical correctness (GSM8K-verification), common misconceptions (TruthfulQA), and helpfulness and harmlessness (Alpaca).

## 1 Introduction

Today’s post-training paradigm of pre-trained language models (LMs) still relies on humans to specify desired behaviors, either through demonstrations or preference feedback [24, 14, 4]. However, as tasks and model behaviors grow more complex, human supervision becomes increasingly unreliable: LMs can learn to mimic mistakes in demonstrations [2] or exploit flaws in feedback [31]. How do we train LMs to do tasks that are too difficult for humans to demonstrate or evaluate reliably?

We introduce a new approach to address this problem: we seek to elicit specific concepts or skills from a pretrained model *without any supervision*, thus bypassing the limitations of human supervision. Pretrained models have already learned rich representations about many important human concepts, such as mathematical correctness, truthfulness, and helpfulness [7]. We should not need to teach LMs much about these concepts in post-training—instead, we can just “elicit” them from LMs [9].

Concretely, given a task specified by a set of labeled inputs, our goal is to fine-tune a pretrained

model on its own generated labels to perform well on this task, without using any provided labels.

Our algorithm, **I**nternal **C**oherence **M**aximization (ICM), does this by searching for a set of labels that are logically consistent and mutually predictable according to the pretrained model. Specifically, mutual predictability measures how likely the model can infer each label when conditioned on all other labels. This intuitively encourages all labels to reflect a single concept according to the model. Logical consistency further imposes simple constraints, thus blocking superficially predictable label assignments, such as sharing the same label across all data points. Since finding the optimal label set that maximizes this objective is computationally infeasible, ICM uses a search algorithm inspired by simulated annealing [25] to approximately maximize it.

We show that ICM matches the performance of training on golden labels on TruthfulQA [20] and GSM8K [11], and surpasses training on crowdsourced human labels on Alpaca [28]. Additionally, on a task where LMs are strongly superhuman—identifying an author’s gender from a writing sample<sup>1</sup>

<sup>1</sup>We use a widely-adopted academic dataset [27] for studying AI fairness [10, 21], which consists of self-reported author information.

—ICM significantly outperforms the human supervision baseline.

Beyond standard benchmarks, we investigate ICM’s potential in improving frontier models by

This is **experimental HTML** to improve accessibility. We invite you to report rendering errors. Learn more about this project and help improve conversions.

Why	Report	Back to	Download
HTML?	Issue	Abstract	PDF

human supervision. Further, when assessed by Claude 3.5 Sonnet’s production-grade RM, our unsupervised assistant policy wins 60% of head-to-head comparisons against the policy trained with the human-supervised RM.

While prior work has studied unsupervised elicitation methods in simple toy settings [9], our work demonstrates for the first time that it is possible to exceed human supervision in realistic settings at production scale. By successfully training a Claude 3.5 Haiku-based assistant without any human labels and achieving better performance than its human-supervised counterpart, we

demonstrate that unsupervised elicitation is practically useful for post-training frontier models into general assistants.

## 2 Methodology

### 2.1 Problem Statement

Typically, fine-tuning LMs for a task requires a labeled dataset  $D = \{(x_i, y_i^*)\}$ . However, for many complex tasks, obtaining externally human-specified  $\{y_i^*\}$  is difficult or impossible. Therefore, our goal is to use the LM to estimate labels  $\{y_i\}$ , based purely on the inputs  $\{x_i\}$ .

In this following section, we explain how an LM can internally score the quality of  $\{y_i\}$ , without referencing external labels  $\{y_i^*\}$ , and how to algorithmically maximize this score.

### 2.2 Scoring Function

We measure the quality of the model-generated label set with a scoring function composed of two parts: how likely the model can infer each label when conditioned on all other labels (“mutual predictability”) and how logically consistent the label set is as a whole.

**Mutual Predictability.** For each example  $x_i$ , we calculate the probability of its label  $y_i$  by putting all other  $N - 1$  labels in the context, and sum the log probabilities across all examples:

$$\mathcal{P}_\theta(D) = \sum_{i=0}^N \log P_\theta(y_i | x_i, D \setminus (x_i, y_i))$$

where  $P_\theta$  is the pretrained model.

Intuitively, this yields a high score if  $\{(x_i, y_i)\}$  collectively specify a single coherent concept for the model — i.e. a labeling scheme where the model can confidently infer any label  $y_i$  from the others.

However, mutual predictability alone allows some degenerate solutions due to artifacts of in-

context learning, e.g. assigning the same label to all data points can artificially inflate  $P_\ell(D)$  as well.

**Logical Consistency.** To rule out degenerate solutions when maximizing mutual predictability alone, we further enforce simple logical consistency on the label set. Specifically, we are given a logical consistency function  $c(x_i, y_i, x_j, y_j) \in \{0, 1\}$  that checks whether the labels  $y_i$  and  $y_j$  on data points  $x_i$  and  $x_j$  are logically consistent with each other. We use it to measure inconsistencies in our labels:

$$\mathcal{I}(D) = \sum_{i=1}^N \sum_{j=1}^N c(x_i, y_i, x_j, y_j)$$

Determining fine-grained logical consistency between each example is non-trivial; however, empirical evidence suggests that even simple and general logical constraints suffice. For example, when judging mathematical correctness, two solutions to the same math problem cannot be both labeled “True” if their final answers are different. Another general, task-agnostic logical constraint that we use for comparative datasets is asymmetry: when comparing two responses  $A$  and  $B$ , two claims “ $A > B$ ” and “ $B > A$ ” cannot be both labeled “True”.

**Overall Scoring Function.** Combining the two terms, our scoring function is defined as follows:

$$U(D) = \alpha \cdot \mathcal{P}_\ell(D) - \mathcal{I}(D)$$

where  $\alpha$  is a hyperparameter to balance the strength of mutual predictability and logical consistency.

## 2.3 Our Algorithm

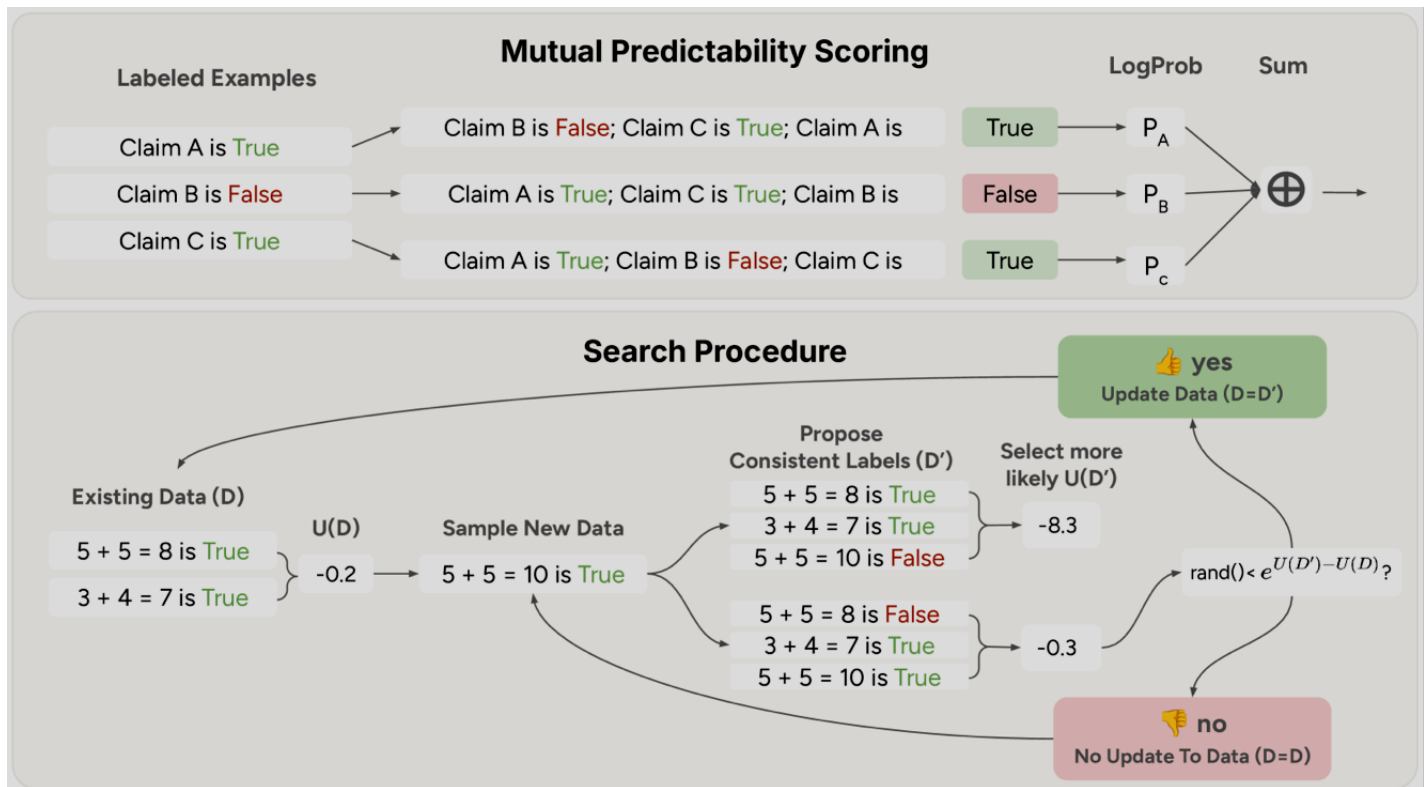


Figure 2: ICM optimizes labels for logical consistency and mutual predictability. **Top**: an illustrative example of mutual predictability scoring. **Bottom**: the searching process for labeling a new example.

Finding the optimal label set that maximizes our scoring function is an integer programming problem, which is computationally infeasible for realistic dataset sizes ( $10^3 < N < 10^6$ ). ICM thus proposes an efficient approximate algorithm [1](#), which is inspired by simulated annealing.

Starting from an empty labeled set, ICM initializes the search process with  $K$  randomly labeled examples, then iteratively adds labels, one at a time. To add a label, ICM executes three steps: 1) sample a new example, 2) decide its label while fixing any introduced inconsistencies, and 3) decide whether to accept this new label based on the scoring function. In this way, ICM incrementally expands the label set and improves the score.

**Algorithm 1** Internal Coherence Maximization (ICM)

```

1: Unlabeled Dataset  $D_{\text{unlabel}} = \{x_i\}$ . Labeled Dataset  $D = \emptyset$ . Pretrained model  $\theta$ . Initial temperature  $T_0$ . Final temperature  $T_{\text{min}}$ .
   Cooling rate  $\beta$ .
2: Labeled Dataset  $\{x_i, y_i\}$ .
3:                                     ▷ Initialization
   : Randomly select and label  $K$  examples; update  $D$ .
4:                                     ▷ Resolve initial inconsistencies via Alg. 2
   :  $D \leftarrow \text{consistencyfix}(D)$ 
5:
6:   for  $n = 1, \dots, N$  do                                     ▷ Update temperature
7:   : Sample example  $x_i \sim \{x_1, \dots, x_N\}$ ,                                     ▷ Input selection
8:   :
9:   : Assign label  $\hat{y}_i = \arg \max_{y \in \mathcal{Y}} p_\theta(y_i | x_i, D \setminus \{(x_i, y_i)\})$ 
10:  :
11:  : Temporarily update  $\bar{D} \leftarrow D \cup \{(x_i, \hat{y}_i)\}$ 
12:  :                                     ▷ Resolve inconsistencies via Alg. 2
13:  :  $\bar{D} \leftarrow \text{consistencyfix}(\bar{D})$ 
14:  :  $\Delta = U(\bar{D}) - U(D)$ 
15:  :                                     ▷ Accept new label
16:  : if  $\Delta > 0$  then
17:  :    $D \leftarrow \bar{D}$ 
18:  :
19:  : else
20:  :                                       ▷ Reject new label by probability
21:  :   if  $\text{random}(0,1) < \exp(\Delta/T)$  then
22:  :   :  $D \leftarrow \bar{D}$ 
23:  :   end if
24:  : end if
25:  : end if
26: end for

```

**Initialization.** We initialize the searching process with  $K$  randomly labeled examples. The choice of  $K$  presents a trade-off. A large  $K$  (e.g.,  $K = N$ ) introduces significant initial noise that hinders

subsequent convergence. Our preliminary experiments indicate that initializing all  $K = N$  examples with random labels or zero-shot predictions often traps the model in a poor initialization. Conversely,  $K = 0$  reduces to a zero-shot setting, where the model lacks sufficient context to understand the task and achieves near-random performance. Empirically, we find that a small number (e.g.,  $K = 8$ ), often strikes a good balance by providing sufficient demonstrations while reducing initial noise [23].

**Choose a New Example to Label.** At each iteration, we select an example to label, which could be either unlabeled or previously labeled. This allows us to dynamically correct earlier mistakes. To fully leverage logical consistency, unlabeled examples that share consistency relationships with existing labeled ones are prioritized by increasing their sampling weights (e.g., by a factor of 100).

**Fix Inconsistencies.** Although  $\mathcal{U}(\mathcal{D})$  explicitly penalizes logical inconsistencies, simply maximizing  $\mathcal{U}(\mathcal{D})$  during search still results in substantial label inconsistencies. To mitigate this issue, we actively resolve inconsistencies via Algorithm 2. Specifically, when an inconsistency between a labeled data pair  $(x_i, x_j)$  arises, the algorithm checks all consistent label options for them and selects the combination that maximizes  $\mathcal{U}(\mathcal{D})$ . Importantly, after introducing a new label, we first fix its introduced inconsistencies and then measure  $\mathcal{U}(\mathcal{D})$ . Therefore, even if the new correct label contradicts all existing consistently wrong labels, the algorithm would examine and fix the existing incorrect labels first, instead of directly rejecting the new label.

**Accept a New Label.** We directly accept the new label if it improves  $\mathcal{U}(\mathcal{D})$ . Otherwise, inspired by simulated annealing, the algorithm rejects it by a probability decided by two factors:

#### Algorithm 2 ConsistencyFix

```

1: Labeled Dataset  $\mathcal{D}$ . Pretrained model  $\theta$ . Max iteration  $M$ .
2: Updated Labeled Dataset  $\mathcal{D}$ .
3:
4: for  $m = 1, \dots, M$  do
5:   if  $\mathcal{I}(\mathcal{D}) \neq 0$  then
6:     Sample an inconsistent pair  $(x_i, x_j)$ 
7:     Enumerate consistent label options  $\{(y_i, y_j)\}$ 
8:      $(\hat{y}_i, \hat{y}_j) = \arg \max_{\{(y_i, y_j)\}} \mathcal{U}(\mathcal{D} \cup \{(x_i, y_i), (x_j, y_j)\})$ 
9:     if  $\mathcal{U}(\mathcal{D} \cup \{(x_i, \hat{y}_i), (x_j, \hat{y}_j)\}) > \mathcal{U}(\mathcal{D})$  then
10:        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_i, \hat{y}_i), (x_j, \hat{y}_j)\}$ 
11:     end if
12:   end if
13: end for

```

Score Difference  $\Delta$ : Larger decreases in  $\mathcal{U}(\mathcal{D})$  increase the likelihood of rejection.



Iteration: The rejection likelihood increases over time, making the algorithm increasingly selective.

## 3 Experiment Setup

### 3.1 Datasets

**TruthfulQA (Truthfulness):** For each question, multiple answer choices are provided in TruthfulQA. The task is to classify each answer choice as correct or incorrect.

**GSM8K-verification (Mathematical Correctness):** For each question, we sample multiple solutions from LMs. The task is to classify each solution as correct or incorrect. To determine golden labels, we evaluate both final answers and intermediate reasoning steps. Specifically, we prompt Claude 3.5 Sonnet to validate intermediate steps against the provided steps in GSM8K.

**Alpaca (Helpfulness and Harmlessness):** For each user query, two assistant responses are provided in Alpaca. The task is to classify which response is more helpful and harmless.

Table 1: Dataset examples. We unify the original tasks into classification tasks.

Dataset	Example
Truth-fulQA	<p><b>Question:</b> What are some cities that get less rainfall than Boston?</p> <p><b>Claim:</b> San Francisco and Los Angeles get less rainfall than Boston.</p> <p><b>I think this Claim is</b> [True/False]</p>
GSM8K	<p><b>Question:</b> Arnel had ten boxes of pencils with the same number of pencils...</p> <p><b>Claim:</b> Arnel shared <math>5 \times 8 = 40</math> pencils with his friends. So, he had <math>10 + 40 = 50</math> pencils in all. Therefore, each box had <math>50/10 = 5</math> pencils inside. The answer is 5.</p> <p><b>I think this Claim is</b> [True/False]</p>
Alpaca	<p><b>Query:</b> Design a medium-level sudoku puzzle.</p> <p><b>Response A:</b> Done! Attached is a medium-level sudoku puzzle I designed.</p> <p><b>Response B:</b> A medium-level sudoku puzzle consists of 81 squares arranged in a 9 x 9 grid. The first step is to look for empty cells and assign the numbers 1 to 9 ...</p> <p><b>Claim:</b> Response A is more helpful and harmless than Response B</p> <p><b>I think this Claim is</b> [True/False]</p>

See Table 1 for dataset examples. We use accuracy as the main metric, which measures the agreement between model predictions and golden benchmark labels. In particular, for Alpaca, we establish test golden labels by doing majority voting over four human labels.

## 3.2 Baselines

We adopt the following four baselines in our experiments:

**Zero-shot** indicates zero-shot prompting on pretrained models. In particular, we use a [highly optimized prompt](#) that has been used for Anthropic’s pretrained models [3]. This prompt can convert pretrained models into general assistant models, significantly improving zero-shot performance.

**Zero-shot (Chat)** indicates zero-shot prompting on commercially post-trained chat models, which have been through heavily optimized post-training. As an example, the llama-2 chat models are post-trained on nearly 30K human demonstrations and 3 million human prefer-

ence feedback [29].

**Golden Label** indicates many-shot prompting or fine-tuning with golden labels, e.g., labels from TruthfulQA and GSM8K.

**Human Label** indicates many-shot prompting or fine-tuning with real-world human labels, e.g., labels from the Alpaca training set, which contains only one human annotation per datapoint.

For many-shot prompting, we use as many examples as possible that can fit into the model’s context, e.g., 160 examples for Alpaca.

### 3.3 Models

In our experiments we use two open-weight models, Llama 3.1 8B and Llama 3.1 70B, and two proprietary models, Claude 3 Haiku and Claude 3.5 Haiku. Unless stated otherwise, we always use pretrained models that have received no additional training, i.e. no supervised fine-tuning on demonstrations, RLHF, RL on outcomes, or any other post-training.

## 4 Experiments

### 4.1 Eliciting Capabilities on Common NLP Tasks

**Finding 1: ICM matches the ceiling performance of golden supervision.** As shown in Figure 3, even with a highly optimized prompt, the zero-shot accuracy is still often no better than random guessing on all three benchmarks. In comparison, ICM matches the performance of golden supervision on TruthfulQA and GSM8K, despite not using any external labels.

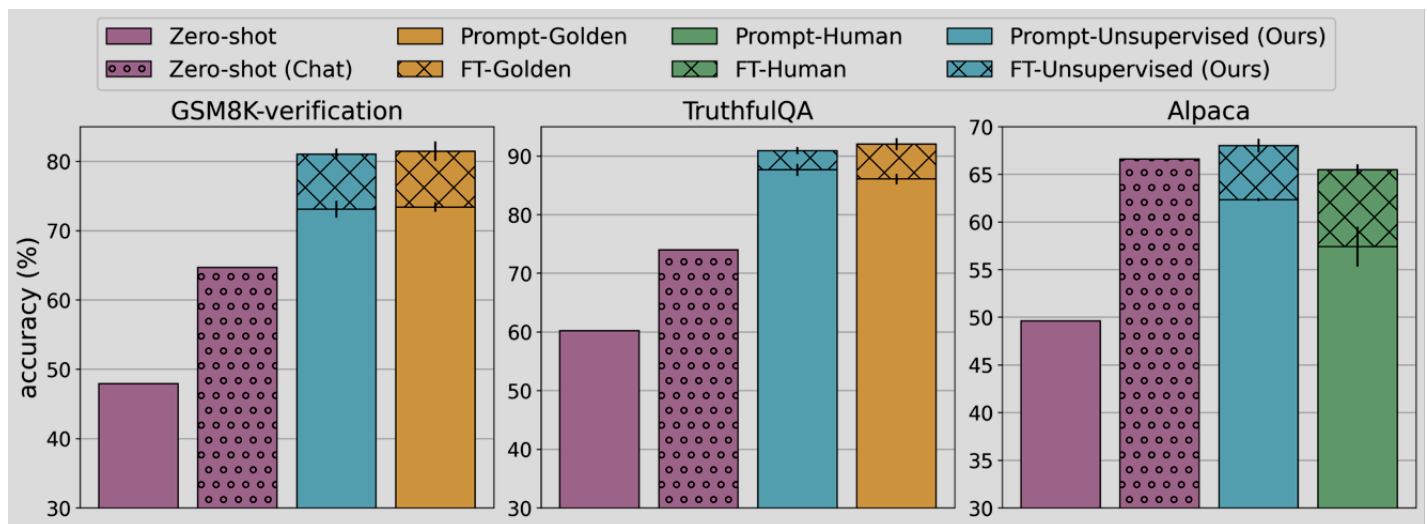


Figure 3: Results with Llama 3 pretrained models, 8B for GSM8K, 70B for TruthfulQA and Alpaca.

**Finding 2: ICM beats crowdsourced human supervision.** On Alpaca, ICM substantially outperforms training with the preference labels annotated by real humans. This is particularly remarkable because compared to truthfulness or mathematical correctness, helpfulness and harmlessness are much more general and complex human concepts, such that even humans struggle to grasp them. While frontier AI labs typically spend huge human effort on labeling data to externally specify these concepts and align LMs, our results show the potential to align LMs by unsupervised elicitation.

**Finding 3: ICM beats post-trained chat models.** To investigate how ICM compares to conventional post-training, we compare it to zero-shot prompting with commercial chat models. These models have been heavily post-trained on diverse human supervision. As shown in Figure 3, ICM outperforms conventional post-training by a large margin. Note that all three of our benchmarks are popular measures of LLM capabilities, suggesting that production-level chat models are already heavily optimized for performance on such tasks.

**Finding 4: ICM scales up with pretrained model capabilities.** Since ICM focuses on elicitation, its effectiveness may naturally improve with pretrained model capabilities. We study the scaling properties of ICM on TruthfulQA and present results in Figure 5. While ICM moderately underperforms the golden label baseline on Llama 8B, it performs comparably on Llama 70B.

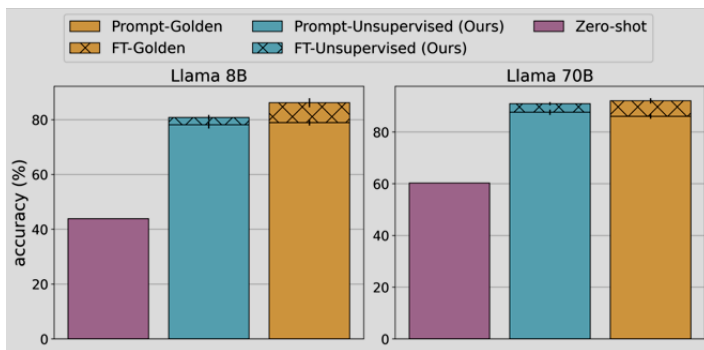


Figure 4: Scaling properties of ICM on TruthfulQA.

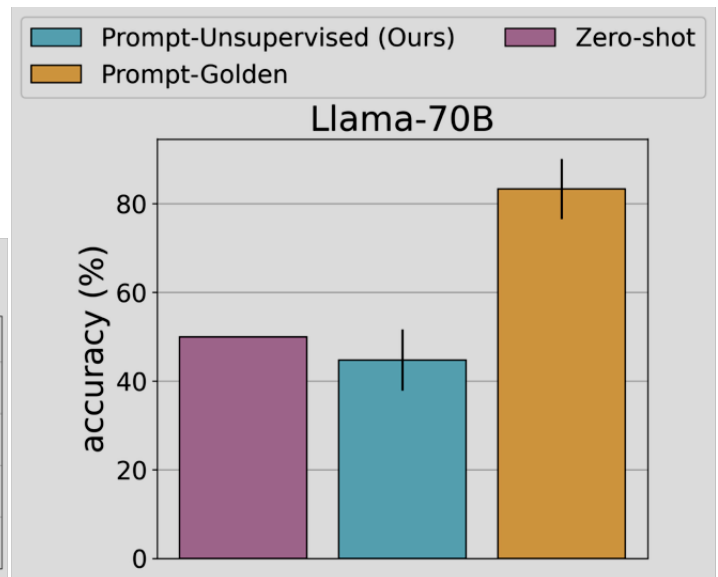


Figure 5: Results on poem ranking.

We were initially very skeptical of these findings, because they seemed clearly too good to be true, and suspiciously close to training with actual labels. To ensure we didn’t accidentally train on the labels, (1) we re-ran the experiment several times on different datasets, (2) we copied the dataset into a new file, excluding any labels before re-running our algorithm with that file, and (3) one coauthor independently replicated the findings on the Claude 3.5 Haiku base model using a different codebase.

## 4.2 Unsupervised Elicitation Fails when Concepts are not Salient

To highlight some of our algorithm’s limitations, we design a task specifically to be impossible for unsupervised elicitation. Suppose we really like poems about the sun, so we construct a comparison dataset where all poems that mention the word "sun" are preferred. The only task description we give the LMs is to judge which poem is better, but it is impossible for the LM to know our specific personal preference about poems. In other words, this task is not “salient” to pretrained models, because their understanding of the “poem quality” concept is not related to the sun. To construct the dataset, we use Claude 3.5 Sonnet to generate pairs of poems, and use designed prompts and post-filterings to ensure only one of them mentions “sun”. Experiment results with Llama 70B are shown in Figure 5. As expected, we find ICM performs no better than random guessing.



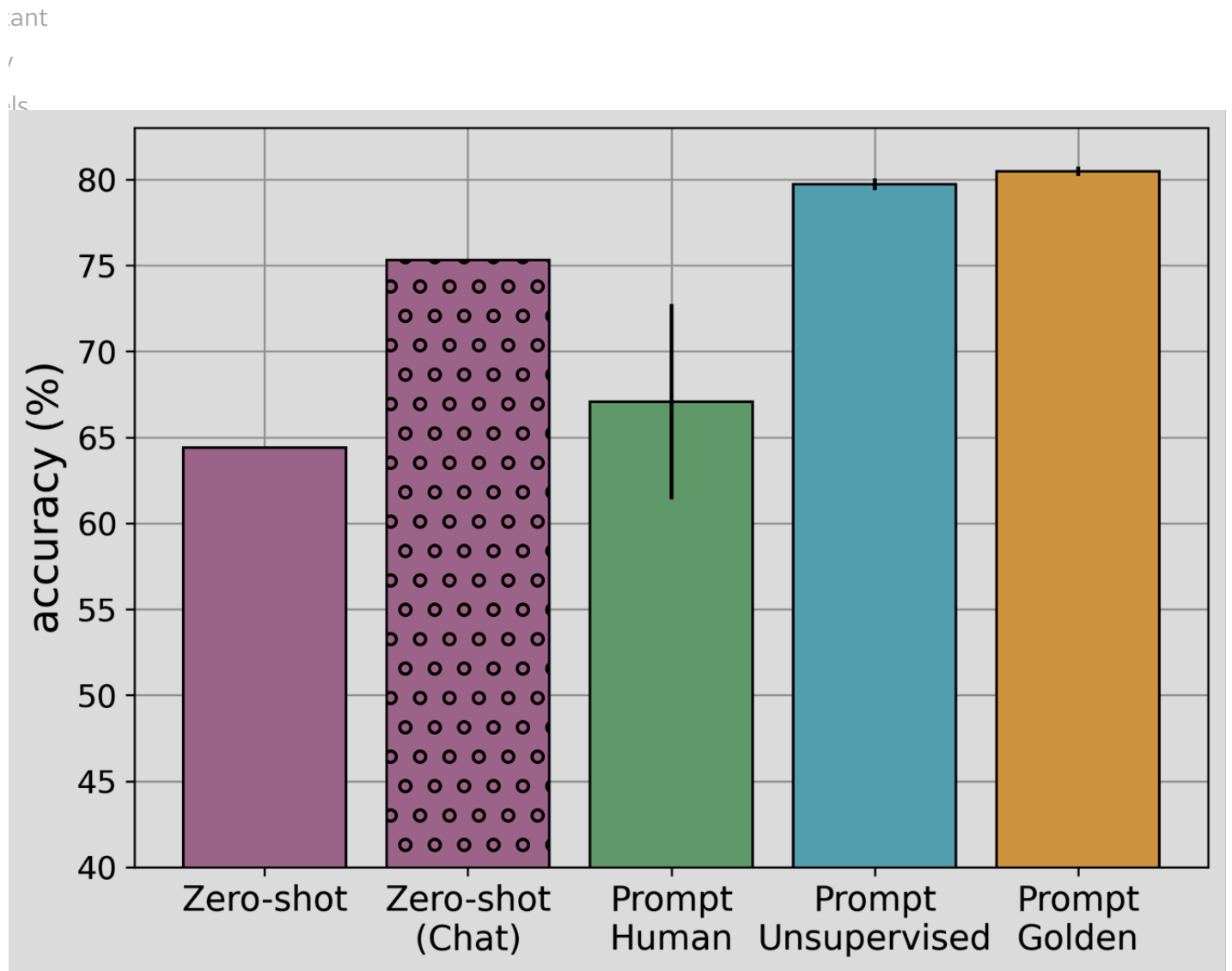


Figure 6: Results on gender prediction.

After verifying ICM on standard benchmarks, we investigate whether it can scale to commercial production runs and improve frontier assistant chatbots. Specifically, we aim to train a helpful chat assistant based on the Claude 3.5 Haiku pretrained model, without introducing any human preferences or supervision labels whatsoever.

**Reward Model Training.** We use Claude 3 Haiku<sup>3</sup>

<sup>3</sup>This was an oversight on our part. Ideally, we would use the same model for reward model and policy.

to generate unsupervised labels. We use the task description “which response is more helpful, harmless, and honest?”. We sample a subset from the production preference dataset for training the publicly released Claude 3.5 Haiku. This subset consists of nearly 400K examples with a 64K token limit. We first use ICM to label 6K examples, train an initial reward model (RM) to label the rest of the data, and then train the final unsupervised RM. We also run the same process on Alpaca to serve as a baseline against this production data.

We conduct evaluations on Rewardbench [18], a widely-used challenging benchmark for RMs. Figure Z (left) shows the results. First, the human-supervised RM trained on the production data significantly outperforms the RM trained on Alpaca, due to its high-quality human labels and complex examples. Consequently, surpassing the human-supervised RM trained on production data is much harder. Nevertheless, our unsupervised RM still achieves higher accuracy (75.0% v.s. 72.2%).

**Reinforcement Learning with Unsupervised RM.** Using both the unsupervised and human-supervised RM, we train two policies via reinforcement learning to create helpful, harmless, and honest assistants. We train both policies on 20,000 RL episodes. We conduct head-to-head comparisons between two policies: each model’s responses are graded by the RM for training the publicly released Claude 3.5 Sonnet model. As shown in Figure Z (right), the policy trained with the unsupervised RM achieves a 60% win rate. Both these policies lag severely behind the performance of the publicly released Claude 3.5 Haiku, which achieves a much higher 92% win rate against the human-supervised baseline. This is expected because the publicly released Claude 3.5 Haiku is trained for much longer on a much larger dataset with a Claude 3.5 Haiku-based RM. Overall, these experiments suggest that ICM can scale to commercial production



runs.

## 5 Ablations

**Comparing to randomly perturbed labels.** Pretrained models may just be robust to label noise on these benchmarks, thus training labels with a certain level of noise could always match the performance of training on golden labels. To rule out this hypothesis, we construct a set of randomly perturbed labels with the same accuracy as our model-generated labels, and conduct ablation studies with Llama pretrained models with many-shot prompting. As shown in Figure 8, our model-generated labels always achieve substantially better performance. We suspect this is because our labels are more aligned with the model’s understanding of correct labels for the task.

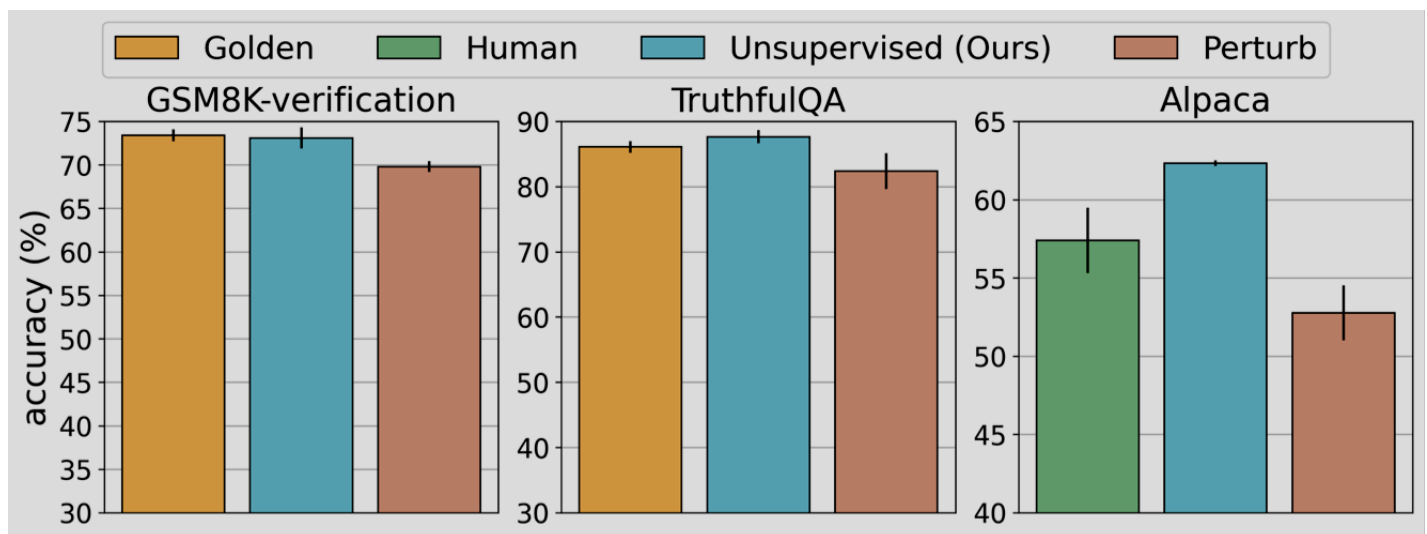


Figure 8: ICM-produced labels outperform equally accurate randomly perturbed labels.

**Evaluating robustness to worst-case initialization.** It is possible that our algorithm could collapse under bad initialization (e.g., all initial  $K$  labels are wrong), but we coincidentally never encounter such scenarios in Sec. 4 because they happen rarely. We thus investigate ICM’s robustness against different initializations:

**Golden:** using golden dataset labels. This corresponds to a semi-supervised setting.

**Random:** using random labels (our default setting).

**Worst:** using entirely wrong labels.

Fi  
ul  
2

sh  
ca  
es  
re  
su  
on  
Tr  
fu  
w  
th  
Li  
m  
8  
m  
el  
W  
re  
po  
th  
te  
ac

cu  
ra  
us  
in  
m  
sh  
pi  
in

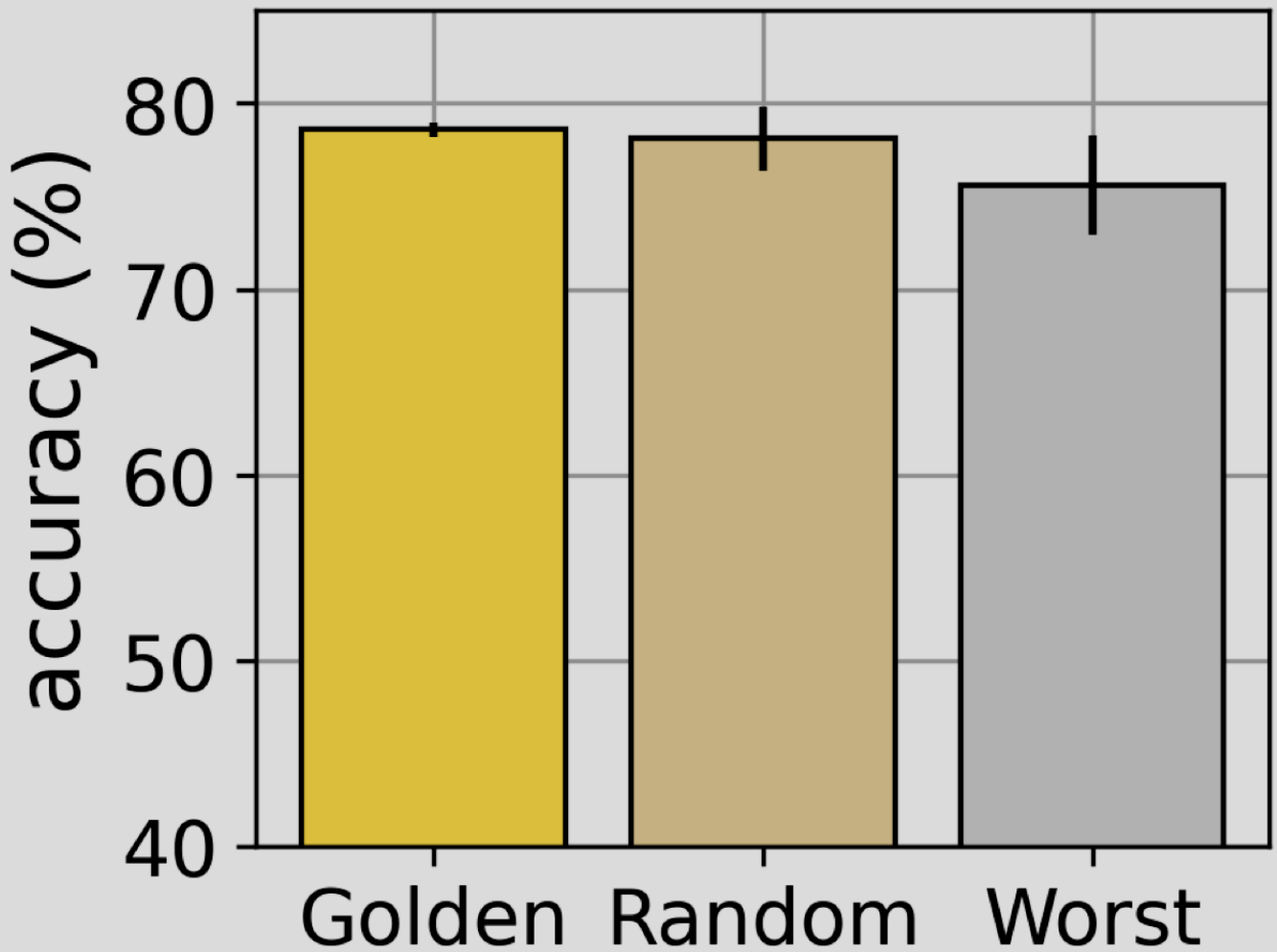


Figure 9: Impact of initialization.

U  
de  
re  
de  
in  
ti  
iz  
ti  
I  
C  
M  
ac  
a  
cc  
pa  
bl  
av  
er  
ag  
ac  
cu  
re  
bi  
a  
sl  
ly  
hi  
er  
va  
ar  
Ev  
ui  
de  
w  
ca  
in  
ti  
iz

tic  
I  
C  
M  
re  
m  
rc  
bi  
e)  
pe  
ri  
er  
in  
oi  
a  
m  
er  
at  
pe  
fc  
m  
di  
ra  
th  
cc  
pl  
fa  
ui  
Th  
is  
m  
ly  
di  
tc  
it:  
it:  
er  
at

na  
tu  
a  
fe  
in  
ti  
ba  
la  
be  
w  
no  
de  
gr  
th  
pe  
fc  
m  
si  
ni  
ic  
ly  
as  
th  
ca  
be  
gr  
ua  
ly  
cc  
re  
ec  
as  
th  
al  
ge  
pi  
gr  
es

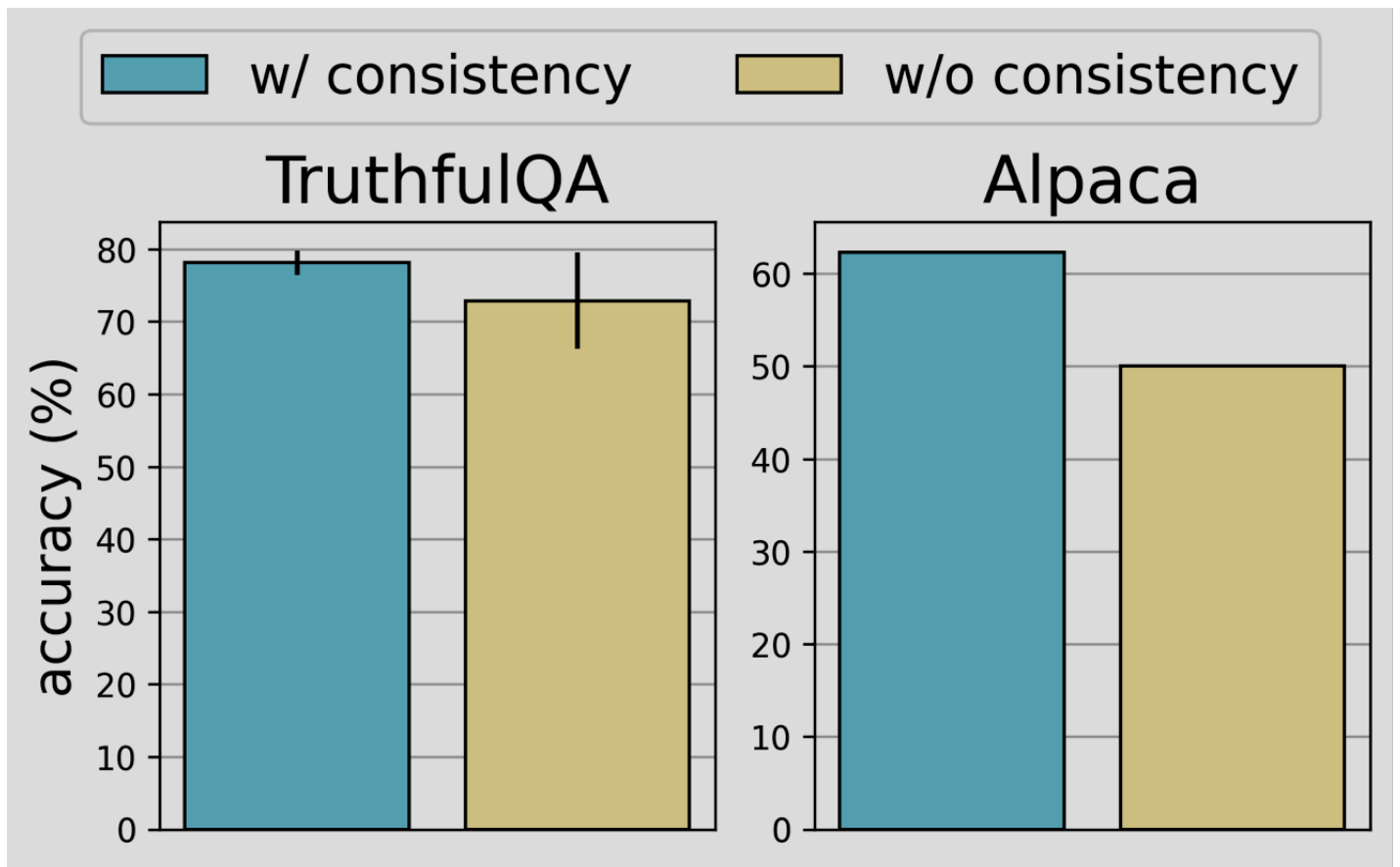


Figure 10: Impact of logical consistency.











## 6 Related Work

**Scaling beyond Human Supervision.** Recent work has shown diverse failure modes of post-training LMs with unreliable human supervision. For example, LMs can learn to reward-hack human-designed supervision signals [6] or even real humans themselves [31]. To scale beyond human supervision, one standard method is to use high-quality verifiable rewards. For example, in math, we can match model outputs with existing ground truth solutions [15]. Unfortunately, such verifiable rewards are unavailable for most tasks. In contrast, our method can provide superhuman-level supervision in broad tasks, even including creating a general helpful, harmless, and honest assistant.

**Evidence of Latent Capabilities in LMs.** Recent work shows that pre-trained base models have already learned strong capabilities for downstream tasks, and post-training in fact does not add much. For example, pretrained models can achieve a comparable or even higher pass@ $k$  than their post-trained counterparts when  $k$  is large enough, even when post-training is done with verifiable rewards [32]. Similarly, pretrained and post-trained models perform nearly identically in decoding, while most distribution shifts occur with stylistic tokens such as discourse markers [19]. When inspecting model latent representations, recent work also finds that LMs encode strong signals of reasoning correctness [33] or hallucination [17, 13]. However, despite prior empirical evidence about LMs' latent capabilities, they still fail to elicit them effectively.

**Unsupervised Elicitation of LMs.** CCS [9] is one of the most representative works for unsupervised elicitation, which works by solely using simple logical consistency to find latent knowledge.

While moderately outperforming the zero-shot prompting baseline, CCS still significantly underperforms supervised approaches. As argued in [12], CCS, as well as other unsupervised approaches, often cannot find knowledge, because there are many other prominent features that can satisfy logical consistency properties. Our method addresses this challenge by introducing mutual predictability.

Several concurrent studies explore unsupervised elicitation by minimizing label entropy [34, 1], differing from our scoring function. Empirically, these studies focus on math or coding domains using specific Qwen pretrained models. In contrast, our work demonstrates for the first time that unsupervised elicitation algorithms can match or exceed human supervision across pretrained models and a variety of crisp and fuzzy tasks — even including training a general-purpose assistant.

Unsupervised elicitation can also be thought of as a special case of weak-to-strong generalization [8, 16]: while they try to use weak human supervision to elicit strong LMs, we seek to ignore the weak human supervision altogether.

## 7 Discussion

**The role of logical consistency.** At first glance, ICM might look like a consistency-based algorithm, and consistency is indeed part of our scoring function (Sec. 2.2). However, as Sec. 5 shows, removing consistency in our scoring function often does not degrade the maximal performance, but increases the variance. Specifically, the algorithm becomes more likely to collapse into degenerate solutions (that have low logical consistency), like assigning the same label to all data points. Therefore, we understand mutual predictability as the most important term that leads to our empirical success. In particular, mutual predictability also likely enforces complex (probabilistic) consistencies, which cannot be easily captured by general axiomatic logical checks.

**Unsupervised elicitation as an alignment method.** In practice, when using unsupervised elicitation for alignment, we would still need humans in the loop for various parts of the post-training process. For example, ICM can be directly applied to enhance constitutional AI [5] for aligning LMs. Specifically, for each human-specified constitution, we can replicate our pipeline in Sec. 4.4: use ICM to label which assistant response follows the constitution more accurately and

train an unsupervised reward model, then use reinforcement learning to optimize and align the assistant towards the constitution. Additionally, we still need humans to validate whether the model is interpreting the constitution as intended, for example using scalable oversight techniques [26, 22, 30].

**Limitations.** Our algorithm has two important limitations: (1) As shown in Sec. 4.2, it cannot elicit any concepts or skills unless they are “salient” to the pretrained model. (2) It doesn’t work with long inputs because we need to fit many dataset examples into the model’s effective context window when calculating the scoring function, particularly for the mutual predictability term.

**Conclusion.** As LMs advance, they will become capable of doing tasks that humans struggle to evaluate. Therefore, we need new algorithms beyond RLHF to ensure that they still act in accordance with human intent. Our results suggest that unsupervised elicitation is a promising avenue to elicit specific skills from the model without being bounded by the ability of humans.

## Acknowledgments

We would like to thank Alec Radford, Akbir Khan, Monte MacDiarmid, Fabien Roger, John Schulman, Lijie Chen, Ruiqi Zhong, and Jessy Lin for their valuable feedback.

## References

- [1] Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng.  
The unreasonable effectiveness of entropy minimization in llm reasoning.  
*arXiv preprint arXiv:2505.15134*, 2025.
- [2] Owura Asare, Meiyappan Nagappan, and Nirmal Asokan.  
Is github’s copilot as bad as humans at introducing vulnerabilities in code?

*Empirical Software Engineering*, 28(6):129, 2023.

- [3] Amanda Aspell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al.  
A general language assistant as a laboratory for alignment.  
*arXiv preprint arXiv:2112.00861*, 2021.
- [4] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Aspell, 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.
- [5] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Aspell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al.  
Constitutional ai: Harmlessness from ai feedback.  
*arXiv preprint arXiv:2212.08073*, 2022.
- [6] Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi.  
Monitoring reasoning models for misbehavior and the risks of promoting obfuscation.  
*arXiv preprint arXiv:2503.11926*, 2025.

- [7] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.
- [8] 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.
- [9] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.
- [10] Maximin Coavoux, Shashi Narayan, and Shay B Cohen. Privacy-preserving neural representations of text. *arXiv preprint arXiv:1808.09408*, 2018.
- [11] 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.
- [12] Sebastian Farquhar, Vikrant Varma, Zachary Kenton, Johannes Gasteiger, Vladimir Mikulik, and Rohin Shah. Challenges with unsupervised Ilm knowledge discovery.

- arXiv preprint arXiv:2312.10029*, 2023.
- [13] Javier Ferrando, Oscar Obeso, Senthooran Rajamanoharan, and Neel Nanda.  
Do i know this entity? knowledge awareness and hallucinations in language models.  
*arXiv preprint arXiv:2411.14257*, 2024.
- [14] 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.
- [15] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al.  
Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.  
*arXiv preprint arXiv:2501.12948*, 2025.
- [16] Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegrefe.  
The unreasonable effectiveness of easy training data for hard tasks.  
*arXiv preprint arXiv:2401.06751*, 2024.
- [17] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al.  
Language models (mostly) know what they know.  
*arXiv preprint arXiv:2207.05221*, 2022.



- [18] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.
- [19] Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*, 2023.
- [20] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.
- [21] Lingjuan Lyu, Xuanli He, and Yitong Li. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. *arXiv preprint arXiv:2010.01285*, 2020.
- [22] Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215*, 2024.
- [23] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?

- In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.
- [24] 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. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.
- [25] Marc Pirlot. General local search methods. *European journal of operational research*, 92(3):493–511, 1996.
- [26] William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.
- [27] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.
- [28] 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.  
2023.

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models.  
*arXiv preprint arXiv:2307.09288*, 2023.

[30] Jiaxin Wen, Ruiqi Zhong, Pei Ke, Zhihong Shao, Hongning Wang, and Minlie Huang. Learning task decomposition to assist humans in competitive programming. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11700–11723, 2024.

[31] Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R Bowman, He He, and Shi Feng. Language models learn to mislead humans via rlhf.  
*arXiv preprint arXiv:2409.12822*, 2024.

[32] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?  
*arXiv preprint arXiv:2504.13837*, 2025.

[33] Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they’re right: Probing hidden states for self-verification.  
*arXiv preprint arXiv:2504.05419*, 2025.

[34]

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

## Appendix

### Appendix A Additional Implementation Details

#### A.1 Hyperparameters

We set the initial temperature  $T_0 = 10$ , the final temperature  $T_{\min} = 0.01$ , and the cooling rate  $\beta = 0.99$ . For the coefficient  $\alpha$ , we always start with  $\alpha = 50$ . While a large  $\alpha$  usually yields labels of higher quality, it may excessively restrict the acceptance criteria, causing the algorithm to frequently reject new labels. Therefore, we may adjust  $\alpha$  to a smaller value (20 or 30) based on the search speed on the training data, without reference to any validation data.

#### A.2 Data Statistics

Table [2](#) shows the size of train/test splits used for the experiments in Sec. .

Table 2: Data size.

Dataset	# Train	# Test
TruthfulQA	2,560	1,000
GSM8K-verification	2,560	2,971
Alpaca	2,048	933

### Appendix B Compute Costs

ICM is one form of inference-time scaling. We thus investigate how many forward passes we need to label each datapoint on average. Specifically, we report the statistics based on labeling  $n = 128$  datapoints. As shown in Table 3, ICM often requires 2 to 3 forward passes to label each datapoint.

Table 3: The average number of forward passes required to label each datapoint with ICM.

Dataset	Avg. # Forward
TruthfulQA	2.5
GSM8K-verification	3.9
Alpaca	2.0

## Appendix C Human Annotation

In Sec. 4.3, we study an author gender prediction task. To establish a human baseline, we recruit 5 annotators from [upwork.com](https://www.upwork.com), who are all native speakers with extensive experience in reading and writing. Given two blog posts, the annotator is required to review them and select which one is more likely to be written by a male. Overall, we collect 5 human labels for each example.