LARGE (VISION) LANGUAGE MODELS ARE UNSUPERVISED IN-CONTEXT LEARNERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advancements in large language and vision-language models have made it possible to solve new tasks via zero-shot inference without task-specific training. Various adaptation techniques, such as In-Context Learning (ICL), supervised fine-tuning, and prompt engineering, can further enhance the model's performance on a given task. However, these methods require either labeled examples or substantial manual effort to construct effective prompts. In this work, we introduce a *joint inference* framework extending the standard zero-shot inference. In contrast to independent zero-shot predictions, joint inference makes predictions simultaneously for all inputs for a given task. Since direct joint inference involves a computationally expensive optimization, we develop efficient approximation techniques resulting in *two unsupervised adaptation methods* that are compatible with language and vision-language models: *unsupervised fine-tuning* and *unsupervised ICL*. We demonstrate the effectiveness of both approaches across a broad range of tasks and models, including language-only Llama 3.1, vision-language OpenFlamingo and API-only access GPT-4o models. Our experiments reveal substantial improvements over the standard zero-shot approach. Furthermore, our approach, although *unsupervised*, often performs on par with supervised approaches that use ground truth labels.

1 INTRODUCTION

Recent progress in large language and vision-language models, which we collectively refer to as foundation models (FMs), have made it possible to adapt them to solve different new tasks via zero-shot inference by leveraging their general knowledge obtained during pre-training (Brown et al., 2020). For a given task, e.g., sentiment classification, it obtains the prediction y for an input sentence x by maximizing the probability of the next token, i.e., $\arg\max_y \ p(y|x)$[1]. Various methods have been proposed to enable better task adaptation, with In-Context Learning (ICL) (Brown et al., 2020; Agarwal et al., 2024; Jiang et al., 2024), fine-tuning (Hu et al., 2022; Jia et al., 2022), and prompt engineering (Wei et al., 2023; Snell et al., 2024) emerging as the most prevalent techniques. While these methods improve upon zero-shot inference, they rely on labeled examples or require manual effort to craft effective prompts, which can pose practical limitations.

In this work, we propose an *unsupervised joint inference* framework that enables fully unsupervised adaptation on a new task. Our framework generalizes the standard zero-shot inference to joint inference over $N > 1$ inputs, resulting in the following optimization problem:

$$\arg\max_{y_1,\dots y_N} p(y_1, \dots, y_N | x_1, \dots, x_N). \tag{1}$$

Compared to the zero-shot independent predictions, joint inference can guide the model to make consistent predictions and reason over multiple inputs simultaneously (see Fig.1-Left). To solve this optimization problem that can be intractable for a large number of examples N, we develop two approximate solutions resulting in two unsupervised adaptation methods: *unsupervised fine-tuning* and *unsupervised ICL*.

The unsupervised fine-tuning method is a principled approach for fine-tuning an FM to optimize its own joint predictive probability (Eq. 1), enhancing an FM based on its own feedback. We show that

[1]This usually also involves having a task-specific textual instruction which we omit here for simplicity.

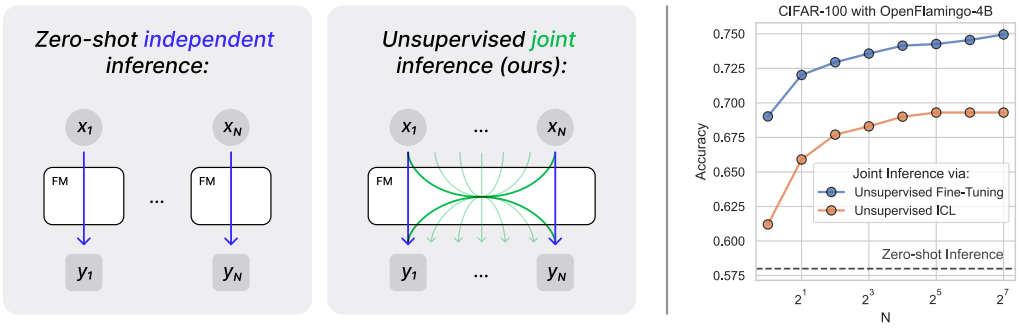

Figure 1: **Unsupervised joint inference framework for foundation models.** ***Left:*** Unlike the standard zero-shot inference that makes a prediction y independently for each input x, our proposed *joint inference* makes predictions for multiple inputs at the same time, leveraging dependencies between all examples. ***Right:*** We develop two methods to perform the unsupervised joint inference that achieve substiantial improvements over traditional zero-shot inference. Their performance also increases as the number of examples N for the joint inference increases, showing the effectiveness of the proposed joint inference framework.

our method matches the performance of supervised fine-tuning in many cases while not having access to any labeled examples. To fine-tune the model, this method requires access to model weights and output probabilities, which might be unavailable in some cases, for example, for close-weight models, such as GPT-4 (Achiam et al., 2023). To enable the applicability of our framework to all existing model types, including closed-weight ones, our unsupervised ICL adaptation method relies only on access to next-token generation and uses the few-shot in-context inference, where instead of ground truth labels, we use the model's own answers from previous iterations. We show that this unsupervised ICL method, in fact, implicitly maximizes the joint probability in Eq.1 and can be seen as an approximate joint inference under the same framework.

We evaluate proposed methods on a range of text and image classification, natural language inference, (visual) question-answering, and math problem-solving (via GSM8K (Cobbe et al., 2021)) tasks using language-only and vision-language FMs, including open-weight Llama 3.1 (Dubey et al., 2024) and OpenFlamingo (Awadalla et al., 2023) models and close-weight GPT-4o via the corresponding API. We show that both proposed methods significantly outperform zero-shot inference and even approach their corresponding fully-supervised counterparts in many cases.

2 Related Work

Adapting FMs via fine-tuning. Pre-training generalist foundation models followed by task-specific fine-tunning was shown to be an effective approach to solving different language and vision tasks (Raffel et al., 2023; Radford et al., 2021; Beyer et al., 2024; Chen et al., 2022; McKinzie et al., 2024; Dubey et al., 2024). The first pre-training stage usually involves optimizing an unsupervised objective, *e.g.*, next-token prediction for language or contrastive loss for vision, on a large-scale dataset (Raffel et al., 2023; Cherti et al., 2023; Radford et al., 2021; Kaplan et al., 2020a; Schuhmann et al., 2022). The second stage involves either full-weights training or parameter-efficient fine-tunning (Hu et al., 2022; Yosinski et al., 2014; Jia et al., 2022; Chen et al., 2023; Houlsby et al., 2019; Pfeiffer et al., 2020). Similar to the second stage, our unsupervised fine-tuning method updates the weights of a pre-trained FM to adapt to a specific task. However, unlike other fine-tuning methods, our approach is based on a *self-improvement mechanism* and does not require labeled examples.

Adapting FMs via prompting. Prompting-based approaches emerged as an alternative optimization-free way to adapt an FM to a new task (Wei et al., 2022; Radford et al.; Brown et al., 2020; Alayrac et al., 2022). A standard zero-shot inference provides input and a task description as a context for a model and generates the answer via next-token prediction. A large line of works develop methods to improve this zero-shot inference by, *e.g.*, prompting a model to generate additional

"reasoning" steps (Wei et al., 2023; Yao et al., 2023; Snell et al., 2024) or providing a few labeled examples as a context (Brown et al., 2020). Similarly, our unsupervised ICL method improves upon the zero-shot inference by using a few *self-generated examples as a context* that are labeled by the model itself, thus without requiring any labeled examples.

Reinforcement learning for FMs. This line of work uses reinforcement learning algorithms Sutton (2018); Schulman et al. (2017) to fine-tune FMs to optimize a non-differentiable reward function. These reward functions are either based on a human feedback (Ouyang et al., 2022; Christiano et al., 2017), a metric (Pinto et al., 2023) or the output the same or another FM (Zheng et al., 2023; Bai et al., 2022; Lee et al., 2024). Related to this, here we *use the model's own feedback based on the joint probability* (Eq. 1) to fine-tune its weights via a reinforcement learning algorithm.

Probabilistic Inference in FMs. Recently, there has been a significant interest in adapting general probabilistic inference techniques to perform inference in a probabilistic models defined by a foundation model. For example, Zhao et al. (2024) build upon Sequential Monte-Carlo (Doucet et al., 2013) to sample from an unnormalized target distribution defined by a foundation model. Another line of works (Hu et al., 2024; Yu et al., 2024) employ GFlowNets framework (Bengio et al., 2023) to solve the probabilistic inference problems. While these general probabilistic inference techniques could be possibly extended to perform the proposed joint inference, we develop the principled unsupervised fine-tuning approach that effectively leverages the structure of our optimization problem.

3 Background

3.1 Foundation Models Pre-training

In this work, we study the class of foundation models (FMs) that are pre-trained on a huge amount of data to model probabilities of a next token given the preceding ones, also known as the *next-token prediction* objective. In particular, given maximal context length L of a foundation model, it models probabilities of token sequences as follows:

$$p_{\text{FM}}(t_1, \ldots, t_L) = \prod_{l=1}^{L} p_{\text{FM}}(t_l | t_{i<l}), \tag{2}$$

where $t_i \in \mathcal{V}$ and $\mathcal{V}$ is the model's vocabulary. Such pre-training has shown remarkable scaling laws (Kaplan et al., 2020b), resulting in the predictable gains that can be delivered by increasing model size, the amount of available training data or compute budget. Furthermore, a separately trained vision adapter can be integrated in such models to enable performing multimodal tasks (Alayrac et al., 2022). It allows a foundation model to ingest a multimodal sequence containing images and/or videos interleaved with text and produce text.

3.2 Solving Downstream Tasks with Foundation Models

This subsection discusses illustrative set of approaches to perform a downstream task given a pre-trained foundation model p_{FM}.

Supervised Fine-tuning. Supervised fine-tuning is the prevalent approach to improve model performance on a downstream task. Specifically, given labeled examples $\mathcal{D}_{\text{train}}$, a model is trained to maximize the probability of the correct outputs, *i.e.*, cross-entropy:

$$\sum_{(x, y_{\text{GT}}) \in \mathcal{D}_{\text{train}}} \log p_{\text{FM}}(y_{\text{GT}} | x). \tag{3}$$

Although being the most performant, supervised fine-tuning requires having access to labeled data, model weights, and, given tremendous model size of p_{FM}, parameter-efficient optimization methods.

Zero-shot Inference and In-Context Learning. Brown et al. (2020) have shown that large-scale pre-training via next-token prediction enables so-called zero-shot inference. Specifically, without any additional training, a foundation model can be prompted with an input instance of a task x and the task description C to generate the corresponding solution via next-token prediction:

$$\arg\max_{y} p_{\text{FM}}(y | x, C). \tag{4}$$

It was also demonstrated that the model's performance is susceptible to the chosen prompts, giving rise to manual prompt engineering to produce more accurate solutions (Liu et al., 2021). Another way to improve the predictions is so-called In-Context Learning (ICL), where the model is provided with a set of input instances and their corresponding ground truth answers:

$$\arg\max_{y} p_{\text{FM}}(y|x, \{(x_n, y_n^{\text{GT}})\}_{n=1}^{N}), \tag{5}$$

where (x_n, y_n^{GT}) denote ground truth (GT) in-context examples and N denotes number of in-context examples. Although such approach has proven itself effective, it requires having access to the set of labeled examples, thus, reflecting the conventional supervised learning setting.

Chain-of-Thought (CoT). Kojima et al. (2023) have recently proposed the off-the-shelf prompting technique that surprisingly improves the performance of a model. In particular, a model is prompted with C_{CoT} = "Think step by step" phrase, that, in turn, triggers it to generate a problem solving reasoning. Subsequently, conditioning on such reasoning chain results into more accurate solutions:

$$\begin{gathered} r_1, \ldots, r_m \sim p_{\text{FM}}(\cdot|C_{\text{CoT}}, x) \\ \arg\max_{y} p_{\text{FM}}(y|r_1, \ldots, r_m, C_{\text{CoT}}, x), \end{gathered} \tag{6}$$

where $r_1, \ldots, r_m$ represent the reasoning chain. The authors have also demonstrated that such approach brings improvements upon both supervised ICL and zero-shot inference.

4 Unsupervised Joint Inference Framework

In this section, we first formally introduce the problem setting and then present a general form of our framework.

Definitions and Problem Setting. We refer to a task $\tau : \mathcal{X} \to \mathcal{Y}$ as a mapping from the space of input instances $\mathcal{X}$ to the set of plausible answers $\mathcal{Y}$. For example, for the task of question answering, the elements of $\mathcal{X}$ and $\mathcal{Y}$ correspond to questions and the corresponding plausible answers to these questions, respectively. Another example can be the sentiment classification task, where $x \in \mathcal{X}$ are sentences, and the set of plausible answers is as simple as $\mathcal{Y} = \{\text{Positive}, \text{Negative}\}$. We assume that we are given a set of input instances $\mathcal{D} = \{x_m\}_{m=1}^{M},\ x_m \in \mathcal{X}$ to perform a task τ on these instances with a foundation model $p_{\text{FM}}(\cdot)$.

The question that we aim to answer in our work is what is a principled approach to improve the predictions of $p_{\text{FM}}(\cdot)$ on a given task τ *in an unsupervised way*, *i.e.*, without having demonstrations of input instances x with their corresponding correct answers y? To simplify the narration, we consider close-ended tasks with K plausible answers, *i.e.*, $\mathcal{Y} = \{y_1, \ldots, y_K\}$, with each $y \in \mathcal{Y}$ comprising a single token. We discuss the general case in detail in Section 6 and Appendix A.

4.1 General Formulation for the Joint Inference

Here, we propose to perform joint inference to produce answers for a set of instances $\mathcal{D}$. In particular, we define the joint likelihood of $y_1, \ldots, y_M$ autoregressively given a set of instances $\mathcal{D}$ and aim to optimize the following objective:

$$\arg\max_{y_1, \ldots, y_M \in \mathcal{Y}^M} \log p(y_1, \ldots, y_M|x_1, \ldots, x_M), \text{where} \tag{7}$$

$$p(y_1, \ldots, y_M|x_1, \ldots, x_M) \stackrel{\text{def}}{=} \prod_{m=1}^{M} p_{\text{FM}}(y_m|x_m, \{(x_i, y_i)\}_{i<m}).$$

Given that foundation models have limited context length and processing the entire $\mathcal{D}$ might be infeasible, we consider the limited number of instances in a single model pass as follows:

$$\arg\max_{y_1, \ldots, y_M \in \mathcal{Y}^M} \mathcal{J}^N(y_1, \ldots, y_M), \text{where} \tag{8}$$

$$\mathcal{J}^N(y_1, \ldots, y_M) \stackrel{\text{def}}{=} \mathbb{E}_{x_1, \ldots, x_N \sim \mathcal{D}} \frac{1}{N} \sum_{n=1}^{N} \log p_{\text{FM}}(y_n|x_n, \{(x_i, y_i)\}_{i<n}),$$

where N limits the number of instances to be processed in a single model pass. Besides the fact that such formulation makes it possible to efficiently estimate the objective via Monte Carlo sampling, it also incorporates the important inductive bias. Indeed, it is easy to note that that Eq. (7) imposes the particular order when processing the sequence $(x_1, y_1, \ldots, x_M, y_M)$ with $p_{\text{FM}}(\cdot)$. However, ground truth answers should not depend on the particular order, and the expectation over different sequences $x_1, \ldots, x_N$ allows to effectively embed this constraint into the objective. One can readily observe that our objective is a strict generalization of the standard zero-shot inference since $\mathcal{J}^1$ reduces to it. Furthermore, Figure 1 demonstrates that increasing N leads to obtaining more accurate answers for the set of instances $\mathcal{D}$ compared to the standard zero-shot inference.

In the subsequent sections, we propose two approaches to optimize the proposed joint inference objective, namely, *unsupervised fine-tuning* and *unsupervised In-Context Learning*.

4.2 Unsupervised Fine-tuning as a Principled Approach

Although the objective $\mathcal{J}^N$ admits efficient Monte Carlo estimation, optimizing it requires $K^M M^N$ model calls which is infeasible in practice. To address this challenge, we resort to the following amortization:

$$\max_{y_1,\ldots,y_M \in \mathcal{Y}^M} \mathcal{J}^N(y_1, \ldots, y_M) \geq \max_{\theta} \mathbb{E}_{y_n \sim \tau_\theta(\cdot|x_n)} \mathcal{J}^N(y_1, \ldots, y_M), \tag{9}$$

where we refer to a $\tau_\theta(\cdot|x_n)$ as a task encoder which defines a distribution over $\mathcal{Y}$ parametrized by continuous parameters θ. As a result, instead of solving the difficult combinatorial optimization problem, we can apply efficient stochastic optimization techniques to learn the parameters of a task encoder. In principle, given flexible enough τ_θ would result in the strict equality in Eq. (9). After the optimization is done, $\arg\max_{y \in \mathcal{Y}} \tau_\theta(y|x_n)$ provides us with answers y_n to the corresponding input instance x_n independently from all other input instances $x_{i \neq n}$, allowing for the efficient inference.

Efficient optimization. Enabling efficient optimization requires obtaining an unbiased stochastic gradient estimator of the objective in Eq. (9). Given that p_{FM} is a black-box function, *i.e.*, it can be evaluated on any given input but the gradients with respect to inputs are unavailable, the prevalent approach in such scenarios is the REINFORCE gradient estimator (Williams, 1992). Despite its generality, a naive implementation of REINFORCE suffers from the high variance when used for the optimization over combinatorial spaces (Gadetsky et al., 2020; Paulus et al., 2021; Struminsky et al., 2021). To address this challenge, we develop an effective stochastic gradient estimator that leverages the structure of our objective to substantially improve the convergence speed. We provide the complete derivation of this estimator and compare it to REINFORCE in Appendix B.1.

Task encoder parametrization. We employ a foundation model itself to serve as our task encoder $\tau_\theta(\cdot|x_n)$. In particular, we constrain p_{FM} to model a distribution over $\mathcal{Y}$ as follows:

$$\tau_\theta(y|x_n) = \frac{p^\theta_{\text{FM}}(y|x_n)[\![y \in \mathcal{Y}]\!]}{\sum_{\hat{y} \in \mathcal{Y}} p^\theta_{\text{FM}}(\hat{y}|x_n)}, \tag{10}$$

where $[\![\cdot]\!]$ denotes Iverson bracket and p^θ_{FM} denotes the same foundation model parametrized by LoRA (Hu et al., 2022) with the corresponding trainable parameters θ. The LoRA parameters θ are set such that, at the beginning of training, p^θ_{FM} corresponds to the zero-shot predictions of p_{FM}, providing a good initialization for our REINFORCE-based optimization, which is known to lead to faster convergence (Greensmith et al., 2004). Noteworthy, this parametrization, coupled with our unsupervised objective, can be seen as an instantiation of self-training, in which a model improves by obtaining feedback from itself.

Regularization. Optimizing Eq. (9) can lead to degenerate solutions, *i.e.*, converging to a single answer for all the input instances, which is common in unsupervised learning. This happens because p_{FM} assigns high probabilities to a single answer after observing the same answer for all input instances in its context. To avoid such trivial solutions, we regularize our task encoder τ. In particular, let $\tau^{\text{prior}}_\theta(y) = \mathbb{E}_{x \in \mathcal{D}} \tau_\theta(y|x)$, then the regularization term is as follows:

$$\mathcal{R}(\tau_\theta) = -\sum_{y \in \mathcal{Y}} \tau^{\text{prior}}_\theta(y) \log \tau^{\text{prior}}_\theta(y). \tag{11}$$

Putting it all together, our final optimization objective to train τ_θ is as follows:

$$\max_{\theta} \mathbb{E}_{y_n \sim \tau_\theta(\cdot|x_n)} \mathcal{J}^N(y_1, \ldots, y_M) + \gamma \mathcal{R}(\tau_\theta), \tag{12}$$

where we found $\gamma = 10$ is a good default choice for the regularization strength. We refer to this principled approach as the joint inference via *unsupervised fine-tuning*. The pseudocode and the implementation details are provided in Appendix B.1.

4.3 Unsupervised In-Context Learning

Although amortization offers a principled approach to optimize the objective in Eq. (8), it requires access to model weights, *i.e.*, to define a task encoder τ_θ, and output probabilities of p_{FM}. This makes our model suitable to open-weight models, but limit its applicability to most close-weight models, such as GPT-4 (Achiam et al., 2023). To make our approach broadly applicable, our key insight is that each summand in Eq. (8) can be seen as ICL predictions:

$$\arg\max_{y_1, \ldots, y_n \in \mathcal{Y}^n} \log p_{\text{FM}}(y_n | x_n, \{(x_i, y_i)\}_{i<n}). \tag{13}$$

Unlike conventional supervised ICL, which relies on ground truth answers, our approach also optimizes answers.

We employ this insight to develop the *unsupervised ICL* approach to optimize Eq. (8) in the multi-turn fashion. Specifically, we iteratively refine answers for the set of instances $\mathcal{D}$ via conditioning on the answers from the previous round, where at the beginning they are initialized by the zero-shot predictions. In particular, for a given $x \in \mathcal{D}$, let $y_x^0 \sim p_{\text{FM}}(\cdot|x)$ be the answers at the 0-th round. Then, for every consecutive refinement round t, we update the answers for $x \in \mathcal{D}$ as follows:

$$y_x^t \sim p_{\text{FM}}(\cdot|x, \{(x_n, y_{x_n}^{t-1})\}_{n=1}^N), \tag{14}$$

where $x_1, \ldots, x_N \sim \mathcal{D}$. In such a way, our approach self-improves answers through the number of iteration steps. It is important to note that such approach is readily applicable to all existing foundation models, since it only requires obtaining samples from a model. Figure 2 hightlights that the proposed approach indeed optimizes the joint inference objective (Eq. (8)). We provide the complete algorithm in Appendix B.2.

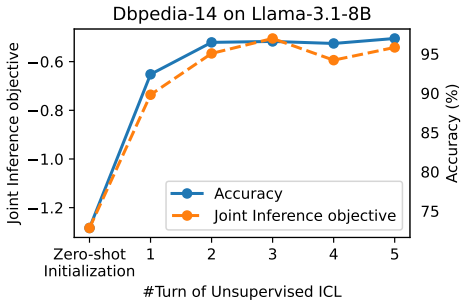

Figure 2: **Unsupervised ICL implicitly optimizes the join inference objective.** Both the joint inference objective and the performance improve with more optimization turns of the unsupervised ICL method.

5 Experiments

Datasets and evaluation metric. We evaluate the performance of our joint inference framework across a wide range of tasks, including text classification, image classification, question answering, visual question answering, natural language inference, common-sense reasoning, and math problem-solving. A detailed description of each dataset, along with the prompts used, is provided in Appendix C.1. We use accuracy as the evaluation metric for all the experiments.

Foundation models. We utilize two open-source foundation models to evaluate our method, namely, Llama-3.1 (Dubey et al., 2024) for text-based experiments and OpenFlamingo (Awadalla et al., 2023; Alayrac et al., 2022) for vision-language experiments. Specifically, Llama-3.1-8B is used as the default model for our primary text experiments, with extensions to the instruction-tuned version and the larger 70B instruction-tuned model. For vision experiments, we use OpenFlamingo-4B as the default model. Furthermore, we employ GPT-4o (Achiam et al., 2023) to serve us as a close-weight foundation model in our experiments.

Baselines. We incorporate the following baselines and upper bounds for our evaluations. (1) *Zero-shot* inference makes the predictions independently for each input example without task-specific

fine-tuning or demonstrations. (2) *Zero-shot with Chain-of-Thought (CoT)* incorporates CoT reasoning prompts to generate intermediate reasoning steps before the final answer (Kojima et al., 2022; Wei et al., 2023). We only use it for our language experiments, as we found that CoT does not show any benefit for the OpenFlamingo model, often significantly degrading the performance. (3) *Supervised In-Context Learning (ICL)* uses labeled training examples to provide them as demonstrations to the model. Consequently, this serves as an upper bound to our unsupervised ICL method, which does not use any labeled data. Similarly, (4) *Fully-Supervised Fine-tuning (FT)* employs LoRA (Hu et al., 2022) supervised fine-tuning using all labeled training examples and serves as an upper bound to our unsupervised fine-tuning method.

Hyperparameters. Unless mentioned otherwise, we use the context length $N = 16$ as the default value for unsupervised fine-tuning, unsupervised ICL, and supervised ICL. We ablate the influence of N in Section 5.3. For unsupervised fine-tuning and supervised fine-tuning, we fine-tune the model with LoRA (Hu et al., 2022) for $6,000$ iterations. For unsupervised ICL, we initialize the labels with the zero-shot predictions and iteratively update it during 5 turns. We refer the reader to Appendix C.2 for the additional implementation details.

5.1 Results on NLP Tasks

Table 1: **Results of the unsupervised fine-tuning and unsupervised in-context learning methods on NLP tasks.** For each dataset, we show the accuracy (in %) of the zero-shot inference, the proposed unsupervised fine-tuning (FT), and ICL methods, and their corresponding supervised counterparts are shown in gray, which represent the upper bound. We use the Llama-3.1-8B model in all cases. Both proposed unsupervised adaptation methods outperform zero-shot inference and approach the performance of the corresponding supervised methods in most cases.

	Text Classification						Language Inference			Question Answering				
Adaptation Method	**SST2**	**Amazon**	**AGNews**	**TREC**	**DBPedia**	**SUBJ**	**RTE**	**QNLI**	**MNLI**	**COPA**	**BoolQ**	**PIQA**	**HellaSwag**	**Avg.**
Zero-shot	77.7	65.5	74.6	42.7	72.4	42.9	62.7	55.5	34.3	81.0	66.7	59.0	46.0	60.1
Zero-shot + CoT	78.8	76.1	58.3	28.7	63.1	54.1	55.6	52.1	47.5	69.0	64.4	58.2	34.6	57.0
Fine-tunning (via LoRA):														
Unsupervised FT	92.3	96.1	89.3	61.9	98.7	95.4	81.7	78.2	72.0	88.1	81.7	80.0	65.5	83.1
Fully-Supervised	92.1	96.0	90.4	93.7	98.8	96.3	89.0	89.5	84.7	85.7	85.6	82.1	87.1	90.1
In-Context Learning (no weight updates):														
Unsupervised ICL	92.4	96.6	86.2	59.0	97.9	74.2	78.8	67.4	65.9	93.5	82.6	78.4	58.2	79.3
Supervised ICL	93.3	96.6	88.0	72.3	97.6	89.2	80.8	74.5	66.6	92.3	84.1	79.1	59.1	82.6

Results on the benchmark datasets. To study the performance of the joint inference framework on language tasks, we evaluate our methods on 13 benchmark datasets, spanning various NLP tasks. Our results highlight the effectiveness of our unsupervised joint inference framework (Table 1). First, we can observe that unsupervised fine-tuning substantially outperforms the standard zero-shot inference. In particular, it brings 23% absolute improvement on average over 13 considered datasets, with remarkable 52.5% 30.6%, 26.3% and 19.5% on the SUBJ, Amazon, DBPedia and HellaSwag datasets, respectively. Furthermore, it often approaches the performance of its fully supervised counterpart, closely matching it on 6 out of 13 considered datasets. Secondly, unsupervised In-Context Learning also exhibits remarkable performance gains compared to the zero-shot inference, bringing 19.2% absolute improvement on average over 13 considered datasets. Remarkably, it is on par with the supervised ICL on 10 out of 13 considered datasets, overall demonstrating the effectiveness of the proposed joint inference framework.

The influence of instruction-tuning and model size. Further, we study the performance of our methods, unsupervised fine-tuning and unsupervised ICL, when applied to the instruction-tuned Llama-3.1-8B and the larger scale Llama-3.1-70B models. Results show that our joint inference framework is effectively applicable across different model sizes compared to Chain-of-Thought (Figure 3), which can improve a foundation model only for large-scale models. In addition, even for the large-scale Llama-3.1-70B model, our unsupervised fine-tuning and unsupervised ICL significantly outperform the Chain-of-Thought prompting technique. In particular, it surpasses CoT by 5% and 4% on the SST-2 and RTE datasets, respectively, providing a principled approach to enhance predictions for models across different sizes.

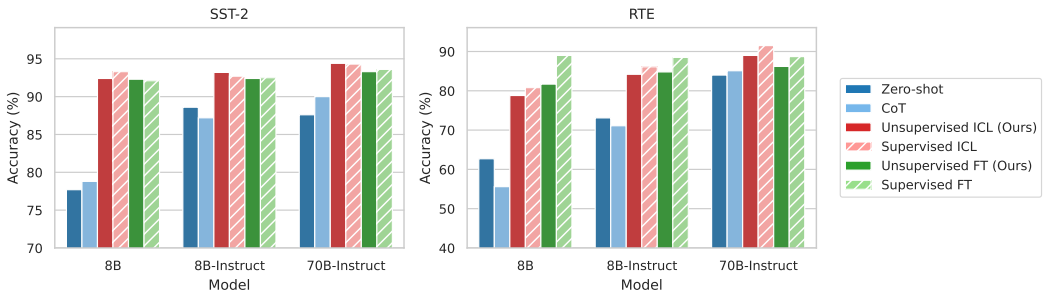

Figure 3: **Using instruction-tuned and larger scale models.** We evaluate our methods on the base 8B, instruction-tuned 8B-Instruct, and a larger 70B-Instruct from the Llama-3.1 family. We find that both proposed methods scale to instruction-tuned and larger-scale models consistently outperform zero-shot baselines. Notably, *our methods applied to the base non-tuned 8B model outperform or work closely to the zero-shot methods on a $\times 9$ larger 70B-Instruct that also benefits from additional training.*

Open-ended tasks. To demonstrate the applicability of our framework to open-ended tasks, we study the performance of our unsupervised ICL on the GSM8K dataset (Cobbe et al., 2021) that contains math problems. We employ large-scale instruction-tuned Llama-3.1-80B for this experiment. Following the official evaluation protocol Dubey et al. (2024) and to demonstrate that our unsupervised ICL can be coupled with modern prompting techniques, we employ Chain-of-Thought for our method and all the baselines. Both our method and supervised ICL use $N = 8$ in-context examples, where supervised ICL utilizes ground truth labels and our method refines the labels in a fully unsupervised manner. We report the performance of unsupervised ICL after 3 refinement turns. Table 2 demonstrates that unsupervised ICL brings remarkable 5.1% absolute improvements upon the zero-shot inference on this challenging benchmark, approaching its supervised counterpart. Overall, these results indicate that our joint inference framework is also applicable to open-ended problems.

Table 2: **Unsupervised ICL successfully scales to the open-ended GSM8K math reasoning task.** We use the Llama-3.1-70B-Instruct model. Our unsupervised ICL method outperforms zero-shot with CoT and approaches the performance of the supervised ICL (as reported in Dubey et al. (2024).

	Accuracy (%)
Zero-shot + CoT	88. 9
Unsupervised ICL	94.0
Supervised ICL	*95.1

5.2 Results for Image Classification and VQA Tasks

Results on the benchmark datasets. To study the performance of the joint inference framework on tasks that require visual comprehension, we evaluate our methods on five vision datasets, spanning both image classification tasks (CIFAR-10, CIFAR-100, Food101) and visual question-answering tasks (COCO-Color and COCO-Number). Our results demonstrates that both unsupervised fine-tuning and unsupervised ICL consistently outperform the standard zero-shot inference (Table 3). In particular, unsupervised fine-tuning brings substantial absolute improvements of 14% on average over the considered datasets with the remarkable gains of 23% on the Food101 dataset, which is the challenging fine-grained image classification task for a vision-language foundation model. Furthermore, reflecting our language experiments, unsupervised ICL closely matches the performance of its supervised counterpart on 4 out of 5 considered datasets, overall demonstrating the applicability of our joint inference framework to vision-language foundation models.

Closed-weight GPT-4o results. To demonstrate the applicability of our framework to closed-weight models, we employ GPT-4o[2] and study the performance of unsupervised ICL on a subset of ImageNet (Deng et al., 2009). In particular, we construct a support set containing 1000 images corresponding to 100 classes and we sample 5000 images for the evaluation purposes only. Specifically, to assess the generalization, we refine the support set with our unsupervised ICL for two rounds,

[2]We employ the GPT-4o version `gpt-4o-2024-08-06`.

Table 3: **Results for image classification and VQA tasks**. For each dataset, we report the accuracy (in %) of zero-shot inference, the proposed unsupervised fine-tuning and unsupervised ICL, and their corresponding supervised counterparts shown in gray. We use OpenFlamingo-4B in all cases. Both unsupervised fine-tuning and unsupervised ICL methods consistently outperform zero-shot inference and approach the performance of the corresponding supervised methods in most cases.

	Image Classification			Visual Question Answering		
Adaptation Method	**CIFAR10**	**CIFAR100**	**Food101**	**COCO-Color**	**COCO-Number**	**Avg.**
Zero-shot	87.2	58.0	58.4	55.8	25.6	57.0
Fine-tunning (via LoRA):						
Unsupervised FT	96.0	74.1	81.0	62.0	42.3	71.1
Fully-Supervised	97.5	84.9	91.5	94.5	85.4	90.7
In-Context Learning (no weight updates):						
Unsupervised ICL	92.6	69.0	61.8	57.5	36.8	63.5
Supervised ICL	93.0	69.1	61.7	58.2	47.1	65.8

Table 4: **Our unsupervised ICL method improves the performance of closed-weight GPT-4o on image classification.** We use 100 randomly sampled classes to construct the IN-100 dataset.

	IN-100 (Top-1, %)
Zero-shot + CoT	76.1
Unsupervised ICL	79.0
Supervised ICL	79.5

and, then, examine the performance on the evaluation set conditioned on the refined support set. As before, we compare our unsupervised ICL to the zero-shot inference and to the suppervised ICL that employs support set with ground truth labels. Table 4 illustrates that unsupervised ICL brings substantial improvement of 3% compared to the zero-shot inference with Chain-of-Thought prompting and approaches supervised ICL, overall demonstrating that our joint inference framework is also applicable to closed-weight models.

5.3 Analysis and Ablations

The influence of the context length N. We examine the impact of the context length N on the performance of our method across both language-only and vision-language tasks. As shown in Figure 4, increasing the context length consistently improves the performance for both our methods, demonstrating the benefits of the joint inference framework to improve the predictions of a foundation model upon the zero-shot inference. Remarkably, unsupervised ICL closely matches the performance of its corresponding supervised upper bound for different values of N. It is also worth noting that the well-known self-training principle, *e.g.* (Huang et al., 2022), resembles as the special case of our unsupervised fine-tuning with $N = 1$.

Convergence rate of the multi-turn unsupervised in-context learning. In addition, we investigate the performance of our unsupervised ICL with respect to the number of turns. The results are shown in Figure 5. Interestingly, we find that the method often converges to near-optimal performance with only a few turns, approaching the supervised ICL upper bound.

6 Conclusion and Limitations

In our work, we proposed the unsupervised joint inference framework that brings substantial improvements over the standard independent zero-shot inference on a given task. The key idea behind our approach is to simultaneously make predictions for multiple input instances of a task. To perform such joint inference, which involves infeasible optimization, we develop two approximations resulting in two efficient unsupervised methods: *unsupervised fine-tuning* and *unsupervised ICL*. We

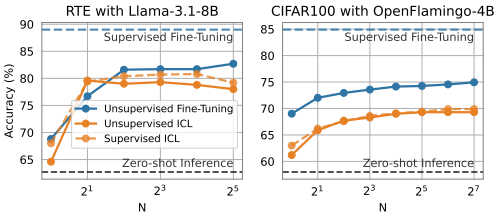

Figure 4: **The effect of the context length N.** We show the performance of both our methods for different context lengths (N). For both text (left) and image (right) classification tasks, our method displays consistent improvement as N increases. This demonstrates the benefits of making joint predictions for multiple examples under the proposed joint inference framework.

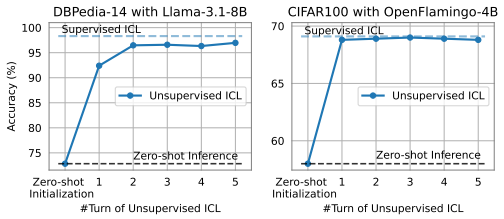

Figure 5: **The convergence analysis of the multi-turns unsupervised ICL method.** We study the number of relabeling turns need for the unsupervised ICL method to converge. We find that the proposed method converges to near-optimal performance after only a few turns and approaches the upper bound supervised ICL performance.

show their effectiveness on a range of datasets and tasks using large language and vision-language models. Below, we discuss both methods and their corresponding limitations.

Unsupervised fine-tuning. Unsupervised fine-tuning is a principled approach to optimize the proposed joint inference objective. Remarkably, although being unsupervised, it often approaches its supervised upper bound, which uses labeled examples for fine-tuning. This approach has two main limitations. First, in its current form, it is limited to close-ended tasks with a finite set of plausible answers $\mathcal{Y}$. This stems from the fact that we need to constrain the output of the task encoder to $\mathcal{Y}$, which greatly benefits the optimization. One potential solution to this could be using more advanced amortization optimization techniques such as (Hu et al., 2024; Zhao et al., 2024). Second, this approach is not applicable to closed-weight proprietary models (Achiam et al., 2023; Team et al., 2023) since it requires access to model weights and output probabilities. We address this limitation with our unsupervised ICL method.

Unsupervised ICL. Unsupervised ICL offers a simple yet powerful approximation to perform the joint inference *compatible with any task and model*. It requires only obtaining samples from a model conditioned on the provided input, that is readily available for all the existing foundation models. Moreover, it can be easily coupled with modern prompting techniques such as Chain-of-Thought to further improve the performance *in an unsupervised manner*. The main limitation of this approach lies in the ICL capabilities of the original model. Indeed, if the original model does not exhibit in-context capabilities, it will not be able to self-refine given the provided context. Conversely, our method can benefit from (constantly) improved capabilities of newly released foundation models.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning, 2024. URL https://arxiv.org/abs/2404.11018.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning, November 2022. URL http://arxiv.org/abs/2204.14198. arXiv:2204.14198 [cs].

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI Feedback, December 2022. URL http://arxiv.org/abs/2212.08073. arXiv:2212.08073 [cs].

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations, 2023. URL https://arxiv.org/abs/2111.09266.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pp. 446–461. Springer, 2014.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision Transformer Adapter for Dense Predictions, February 2023. URL http://arxiv.org/abs/2205.08534. arXiv:2205.08534 [cs].

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

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.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Arnaud Doucet, Nando Freitas, Kevin Murphy, and Stuart Russell. Sequential monte carlo methods in practice. 01 2013. doi: 10.1007/978-1-4757-3437-9_24.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Artyom Gadetsky, Kirill Struminsky, Christopher Robinson, Novi Quadrianto, and Dmitry Vetrov. Low-Variance Black-Box Gradient Estimates for the Plackett-Luce Distribution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10126–10135, April 2020. ISSN 2374-3468. doi: 10.1609/aaai.v34i06.6572. URL https://ojs.aaai.org/index.php/AAAI/article/view/6572.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning. *Journal of Machine Learning Research*, 5(Nov): 1471–1530, 2004. ISSN ISSN 1533-7928. URL https://www.jmlr.org/papers/v5/greensmith04a.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.

Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing Intractable Inference in Large Language Models. In *International Conference on Learning Representations*, 2024.

Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning, July 2022. URL http://arxiv.org/abs/2203.12119. arXiv:2203.12119 [cs].

Yixing Jiang, Jeremy Irvin, Ji Hun Wang, Muhammad Ahmed Chaudhry, Jonathan H. Chen, and Andrew Y. Ng. Many-shot in-context learning in multimodal foundation models, 2024. URL https://arxiv.org/abs/2405.09798.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020a.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020b. URL https://arxiv.org/abs/2001.08361.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in neural information processing systems*, 2022.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL https://arxiv.org/abs/2205.11916.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback, September 2024. URL http://arxiv.org/abs/2309.00267. arXiv:2309.00267 [cs].

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172, 2013.

Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14236. URL https://www.nature.com/articles/nature14236.

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.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*, 2004.

Max B. Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris J. Maddison. Gradient Estimation with Stochastic Softmax Tricks, February 2021. URL http://arxiv.org/abs/2006.08063. arXiv:2006.08063 [cs, stat].

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.

André Susano Pinto, Alexander Kolesnikov, Yuge Shi, Lucas Beyer, and Xiaohua Zhai. Tuning computer vision models with task rewards. In *International Conference on Machine Learning*, pp. 33229–33239. PMLR, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, February 2021. URL http://arxiv.org/abs/2103.00020. arXiv:2103.00020 [cs].

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, September 2023. URL http://arxiv.org/abs/1910.10683. arXiv:1910.10683 [cs, stat].

P Rajpurkar. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI spring symposium series*, 2011.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters, August 2024. URL http://arxiv.org/abs/2408.03314. arXiv:2408.03314 [cs].

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Kirill Struminsky, Artyom Gadetsky, Denis Rakitin, Danil Karpushkin, and Dmitry Vetrov. Leveraging Recursive Gumbel-Max Trick for Approximate Inference in Combinatorial Spaces, October 2021. URL http://arxiv.org/abs/2110.15072. arXiv:2110.15072 [cs].

Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Ellen M Voorhees and Dawn M Tice. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 200–207, 2000.

Alex Wang. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent Abilities of Large Language Models, October 2022. URL http://arxiv.org/abs/2206.07682. arXiv:2206.07682 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, January 2023. URL http://arxiv.org/abs/2201.11903. arXiv:2201.11903 [cs].

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, May 1992. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00992696. URL http://link.springer.com/10.1007/BF00992696.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models, December 2023. URL http://arxiv.org/abs/2305.10601. arXiv:2305.10601 [cs].

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks?, November 2014. URL http://arxiv.org/abs/1411.1792. arXiv:1411.1792 [cs].

Fangxu Yu, Lai Jiang, Haoqiang Kang, Shibo Hao, and Lianhui Qin. Flow of reasoning: Efficient training of llm policy with divergent thinking, 2024. URL https://arxiv.org/abs/2406.05673.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Baker Grosse. Probabilistic Inference in Language Models via Twisted Sequential Monte Carlo. In *International Conference on Machine Learning*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, December 2023. URL http://arxiv.org/abs/2306.05685. arXiv:2306.05685 [cs].

A GENERALIZATION TO MULTI-TOKEN LABELS

In the main paper, we assume for simplicity that each $y \in \mathcal{Y}$ comprise a single token, which might not be the case for many datasets. Let $y_k = [t_1^k, \dots, t_{l_k}^k] \in \mathcal{Y}$ be a multi-token label comprising l_k tokens. Then, to compute $\log p_{\text{FM}}(y_k | x_m, \{(x_i, y_i)\}_{i<m})$ one would need to sum over all the tokens comprising y_k:

$$\log p_{\text{FM}}(y_k | x_m, \{(x_i, y_i)\}_{i<m}) = \sum_{i=1}^{l_k} \log p_{\text{FM}}(t_i^k | t_{j<i}^k, x_m, \{(x_i, y_i)\}_{i<m}). \tag{15}$$

Given that our task encoder τ_θ involves renormalization in Eq. (10), summation over all the tokens for all $y \in \mathcal{Y}$ would require impractical multiple model calls.

First token approximation. In case of absence of labels $y \in \mathcal{Y}$ sharing their first corresponding token t_1^k, we found that the following approximation of Eq. (15) performs well in practice:

$$\sum_{i=1}^{l_k} \log p_{\text{FM}}(t_i^k | t_{j<i}^k, x_m, \{(x_i, y_i)\}_{i<m}) \approx \log p_{\text{FM}}(t_1^k | x_m, \{(x_i, y_i)\}_{i<m}). \tag{16}$$

Bag-of-Tokens (BoT) approximation. First token approximation would not work in case there are labels $y_i, y_j \in \mathcal{Y}$ that share prefix. Such scenario mostly occurs for fine-grained image classification problems. To address this challenge, we, first, find the minimal prefix $\hat{y}_k = [t_1^k, \dots, t_{\hat{m}_k}^k]$, $\hat{m}_k \leq m_k$ that allows to distinguish $y_k \in \mathcal{Y}$ from the rest labels. Then, we propose to consider $\hat{y}_k$ as a Bag-of-Tokens, effectively ignoring the order of $t_1^k, \dots, t_{\hat{m}_k}^k$:

$$\sum_{i=1}^{l_k} \log p_{\text{FM}}(t_i^k | t_{j<i}^k, x_m, \{(x_i, y_i)\}_{i<m}) \approx \sum_{t \in \hat{y}_k} \log p_{\text{FM}}(t | x_m, \{(x_i, y_i)\}_{i<m}). \tag{17}$$

It is easy to note that Bag-of-Tokens approximation reduces to the first token approximation for datasets without labels that share a prefix. Consequently, we use it by default for all the datasets.

B Implementation Details of the Proposed Approaches

B.1 Amortized Approach

For the close-ended tasks it is feasible to enumerate all $y \in \mathcal{Y}$, thus we renormalize conditional likelihoods over the entire set $\mathcal{Y}$, resulting in:

$$\log p(y_n|x_n, \{(x_i, y_i)\}_{i<n}) \stackrel{\text{def}}{=} \log \frac{p_{\text{FM}}(y_n|x_n, \{(x_i, y_i)\}_{i<n})}{\sum_{y \in \mathcal{Y}} p_{\text{FM}}(y|x_n, \{(x_i, y_i)\}_{i<n})}, \tag{18}$$

where it is important to note that this renormalization does not require additional model calls. It is well-known that rescaling the objective is beneficial for the faster convergence of REINFORCE-based optimization methods (Mnih et al., 2015; Schulman et al., 2017; Sutton, 2018). We use this renormalization for all the summands in $\mathcal{J}^N$ in Eq. (12).

Low-variance Gradient Estimator. Our objective in Eq. (9) has the following form:

$$\mathbb{E}_{x_1,\ldots,x_N \sim \mathcal{D}} \mathbb{E}_{y_n \sim \tau_\theta(\cdot|x_n)} \sum_{n=1}^{N} \mathcal{J}_n^N(y_1, \ldots, y_n), \text{where} \tag{19}$$

$$\mathcal{J}_n^N(y_1, \ldots, y_n) = \frac{1}{N} \log p(y_n|x_n, \{(x_i, y_i)\}_{i<n}).$$

Without loss of generality, let's consider particular samples $\hat{x}_1, \ldots, \hat{x}_N \sim \mathcal{D}$, since averaging over multiple samples does not introduce any bias. Thus, after rearranging terms, we need to obtain the unbiased gradients for the following objective:

$$\sum_{n=1}^{N} \nabla_\theta \mathbb{E}_{y_1,\ldots,y_n \sim \prod_{i=1}^n \tau_\theta(\cdot|\hat{x}_n)} \mathcal{J}_n^N(y_1, \ldots, y_n). \tag{20}$$

Considering only n-th term, let's note that:

$$\nabla_\theta \mathbb{E}_{y_1,\ldots,y_n \sim \prod_{i=1}^n \tau_\theta(\cdot|\hat{x}_n)} \mathcal{J}_n^N(y_1, \ldots, y_n) = \nabla_\theta \mathbb{E}_{y_1,\ldots,y_{n-1}} \sum_{y \in \mathcal{Y}} \mathcal{J}_n^N(y_1, \ldots, y_{n-1}, y) \tau_\theta(y|\hat{x}_n). \tag{21}$$

The key insight here is that marginalization over $y \in \mathcal{Y}$ can be done efficiently without additional model calls as before. Let's denote $\tilde{\mathcal{J}}(y_1, \ldots, y_{n-1}, \theta) = \sum_{y \in \mathcal{Y}} \mathcal{J}_n^N(y_1, \ldots, y_{n-1}, y) \tau_\theta(y|\hat{x}_n)$, then

$$\nabla_\theta \mathbb{E}_{y_1,\ldots,y_{n-1}} \tilde{\mathcal{J}}(y_1, \ldots, y_{n-1}, \theta) = \tag{22}$$

$$\mathbb{E}_{y_1,\ldots,y_{n-1}} \Big[\tilde{\mathcal{J}}(y_1, \ldots, y_{n-1}, \theta) \sum_{j=1}^{n-1} \nabla_\theta \log \tau_\theta(y_j|\hat{x}_j) \Big] + \mathbb{E}_{y_1,\ldots,y_{n-1}} \frac{\partial}{\partial \theta} \tilde{\mathcal{J}}(y_1, \ldots, y_{n-1}, \theta),$$

where the first term can be seen as the REINFORCE gradient estimator for $\tilde{\mathcal{J}}(y_1, \ldots, y_{n-1}, \theta)$ and the second term is low-variance pathwise derivative. To reduce the variance of the overall estimator even further, we introduce simple yet effective control variate for the first term. In particular, let $y_j^* = \arg\max_{y \in \mathcal{Y}} \tau_\theta(y|\hat{x}_j),\ j = 1, \ldots, (n-1)$, then our final gradient estimator is:

$$\mathbb{E}_{y_1,\ldots,y_{n-1}} \Bigg[\Big[\tilde{\mathcal{J}}(y_1, \ldots, y_{n-1}, \theta) - \mathcal{B}(\hat{x}_1, \ldots, \hat{x}_n) \Big] \times \Big[\sum_{j=1}^{n-1} \nabla_\theta \log \tau_\theta(y_j|\hat{x}_j) \Big] \Bigg] + \tag{23}$$

$$+ \mathbb{E}_{y_1,\ldots,y_{n-1}} \frac{\partial}{\partial \theta} \tilde{\mathcal{J}}(y_1, \ldots, y_{n-1}, \theta), \text{where}$$

$$\mathcal{B}(\hat{x}_1, \ldots, \hat{x}_n) = \sum_{y \in \mathcal{Y}} \mathcal{J}_n^N(y_1^*, \ldots, y_{n-1}^*, y) \tau_\theta(y|\hat{x}_n).$$

The obtained estimator admits the unbiased estimate by sampling $y_1, \ldots, y_N \sim \tau_\theta(\cdot|\hat{x}_n)$ and calculating what is inside expectations. Figure B1 demonstrates the effectiveness of the proposed gradient estimator on several tasks.

B.2 Multi-Turn for Unsupervised In-Context Learning

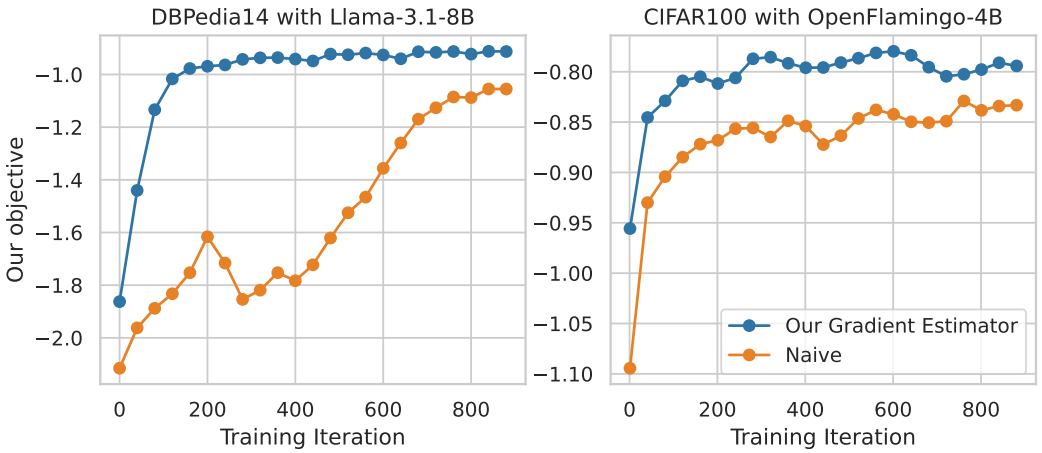

Figure B1: **Comparison of our gradient estimator with the naive approach.** The plot shows the convergence rate during optimization of the joint inference objective. Our proposed gradient estimator achieves faster convergence and leads to the higher values of the objective.

Algorithm B1 Amortized Approach

Input: Dataset $\mathcal{D}$, Foundation model $p_{\text{FM}}(\cdot)$, hyperparameter N, LoRA task encoder $\tau_\theta(\cdot)$ with parameters θ, regularization strength γ, number of iterations T, learning rate α, batch size B
Initialize θ_0 such that $\tau_{\theta_0} = p_{\text{FM}}$
for $t = 0$ to $T - 1$ **do**
  Sample mini-batch $x_1^b, \ldots, x_N^b \sim \mathcal{D},\ b = 1, \ldots, B$
  Sample answers $y_n^b \sim \tau_{\theta_t}(\cdot|x_n^b),\ n = 1, \ldots, N; b = 1, \ldots, B$
  Estimate $\tau_{\theta_t}^{\text{prior}}(\cdot) = \frac{1}{N \times B} \sum_{b=1}^{B} \sum_{n=1}^{N} \tau_{\theta_t}(\cdot|x_n^b)$
  Compute the objective $\mathcal{O}_t = \frac{1}{B} \sum_{b=1}^{B} \sum_{n=1}^{N} \mathcal{J}_n^N(y_1, \ldots, y_n) + \gamma \mathcal{R}(\tau_{\theta_t}^{\text{prior}})$
  Compute the gradient estimator g_t via Eq. (23)
  Updante the parameters: $\theta_{t+1} = \theta_t + \alpha g_t$
end for
Produce answers $y_n = \arg\max_{y \in \mathcal{Y}} \tau_{\theta_T}(y|x)$for all $x \in \mathcal{D}$
Output: Answers for $\mathcal{D}$

Algorithm B2 Multi-Turn Approach

Input: Dataset $\mathcal{D}$, Foundation model $p_{\text{FM}}(\cdot)$, hyperparameter N, number of turns T, number of repeats N_r
Initialize answers with zero-shot predictions: $\mathcal{D}_0 = \{(x, y) \mid x \in \mathcal{D},\ y \sim p_{\text{FM}}(\cdot|x)\}$
for $t = 1$ to T **do**
  Initialize $\mathcal{D}_t = \varnothing$
  for $x \in \mathcal{D}$ **do**
    for $n = 1$ to N_r **do**
      Sample support examples labeled by previous turn: $(x_1, y_1^{t-1}), \ldots, (x_N, y_N^{t-1}) \sim \mathcal{D}_{t-1}$
      Obtain answer: $y_n^x \sim p_{\text{FM}}(\cdot|x, (x_1, y_1^{t-1}), \ldots, (x_N, y_N^{t-1}))$
    end for
    Take majority vote over N_r options: $y^x = \text{MAJ}(y_1^x, \ldots, y_{N_r}^x)$
    Update answers: $\mathcal{D}_t = \mathcal{D}_t \cup \{y^x\}$
  end for
end for
Take answers from the last turn: $\{y_n \mid (x_n, y_n) \in \mathcal{D}_T\}$
Output: Answers for $\mathcal{D}$

C EXPERIMENTAL DETAILS

C.1 DATASETS AND PROMPTS

Text. We evaluate our method on 13 NLP datasets covering various tasks. For sentiment analysis, we use *SST2* (Socher et al., 2013), which contains movie reviews classified as positive or negative, and *Amazon* (McAuley & Leskovec, 2013), a dataset of product reviews with similar labels. For topic classification, we use *AG-News* (Zhang et al., 2015), which consists of news articles categorized into four topics (World, Sports, Business, and Technology), *TREC* (Voorhees & Tice, 2000) for classifying questions into six types, and *DBpedia-14* (Lehmann et al., 2015), which includes Wikipedia articles grouped into 14 categories. *SUBJ* (Pang & Lee, 2004) is used for classifying sentences as subjective or objective. For natural language inference, we use *RTE* (Wang, 2018) to assess entailment relationships, *QNLI* (Rajpurkar, 2016) for sentence-answering tasks, and *MNLI* (Williams et al., 2017), which involves classifying sentence pairs into entailment, contradiction, or neutral. We also include *COPA* (Roemmele et al., 2011) and *HellaSwag* (Zellers et al., 2019) for story completion, *BoolQ* (Clark et al., 2019) for yes/no question answering, and *PIQA* (Bisk et al., 2020) for physical commonsense reasoning. For open-ended questions, *GSM8K* (Cobbe et al., 2021) assesses mathematical reasoning through multi-step word problems.

For each dataset, we randomly sample $2,000$ examples as the train split for unsupervised learning, and $1,000$ examples as the test split for evaluation (except for COPA where there are only 500 examples in total). We balance labels in both train split and test split. For GSM8K (Cobbe et al., 2021), we use the whole test set which contains 1319 examples for the evaluation. The datasets and corresponding prompts are summarized at Table C1.

Vision. We evaluate our method on five vision datasets, including three image classification tasks and two visual question-answering tasks. For image classification, we use *CIFAR10* (Krizhevsky et al., 2009), a benchmark dataset with color images across 10 different classes, and *CIFAR100* (Krizhevsky et al., 2009), which provides a more detailed classification challenge with 100 classes. We also include *Food101* (Bossard et al., 2014), a large-scale dataset featuring a wide variety of food categories. For visual question answering, we use *COCO-Color* and *COCO-Number*, both derived from VQAv2 (Goyal et al., 2017). *COCO-Color* focuses on questions about the dominant colors of objects in images, testing the model's ability to understand color attributes, while *COCO-Number* involves predicting numerical attributes such as object counts, evaluating the model's numeric reasoning based on visual input.

For all vision datasets, we train the model on the entire training set and report performance on the test set. Details of the prompts used for each dataset can be found in Table C1.

C.2 IMPLEMENTATION DETAILS / HYPERPARAMETERS

Unsupervised Fine-tuning. We use LoRA (Hu et al., 2022) for parameter-efficient fine-tuning on NLP and vision tasks. For NLP tasks with Llama-3.1, we also use flash-attention (Dao et al., 2022) and 4-bit quantization of the model provided by the Unsloth library [3] to improve efficiency. We found that with improved gradient estimator, the training is less sensitive to the hyper-parameters. Thus we do not customize hyperparamters for each datasets, and instead using a learning rate of 1e-5 with Adam optimizer for all datasets. The model is fine-tuned for 6,000 iterations and usually the training converges at around 2,000 iterations. We train our model with 64 examples at each mini-batch. We use context-length $N = 16$ for the main experiments and provide ablation study on the effect of N at Section 5.3. Similarly, for vision experiments, we train our model for 3,000 iterations with a learning rate of 1e-4, and 256 examples at each iteration. The typical training time is 12h for text tasks and 4h for vision tasks, on one NVIDIA H100 GPU.

Unsupervised ICL. For unsupervised in-context learning (ICL), we initialize pseudo-labels using zero-shot predictions and iteratively refine them based on ICL predictions. At each iteration, the label of a query example is updated based on the ICL prediction from N support examples. For both supervised and unsupervised ICL, we manually balance the labels when sampling support examples, as this helps prevent biased predictions. Additionally, we sample 5 support sequences per iteration

[3] The library could be found at https://github.com/unslothai/unsloth

and apply a majority vote to reduce variance. The labels are updated across 5 turns, after which we report the accuracy on the test set.

GPT-4o evaluation. We use the version of "gpt-4o-2024-08-06" for evaluation. We experiment on a subset of the ImageNet dataset with 1000 support images and 5000 evaluation images corresponding to 100 classes. We perform two-turn pseudo labeling for unsupervised ICL and 16-shot for evaluation. The total cost for the API call and evaluation is $200.

Table C1: Datasets and corresponding prompts used in this paper.

Dataset	Prompts
SST2	*<sentence>* The sentiment of the sentence is *<label>*.
Amazon	*<title><content>* The sentiment of the sentence is *<label>*.
AG-News	*<text>* The topic of the sentence is about *<label>*.
TREC	*<text>* The topic of the sentence is about *<label>*.
DBpedia-14	*<title><content>* The topic of the sentence is about *<label>*.
SUBJ	*<text>* The sentence is *<label>*.
RTE	*<premise>* Question: Does this imply that "*<hypothesis>*", yes or no? Answer: *<label>*.
QNLI	*<sentence>* Question: Does that sentence have all you need to answer the question "*<question>*", yes or no? Answer: *<label>*.
MNLI	*<premise>* Based on the previous passage, is it true that "*<hypothesis>*"? Answer: *<label>*.
COPA	Consider the following premise: "*<premise>* " Choice 1: *<choice1>* Choice 2: *<choice2>* Q: Which one is more likely to be the *<question>*, choice 1 or choice 2? A: *<label>*.
BoolQ	*<passage>* Question: After reading this passage, the answer to the question *<question>* is yes or no? Answer: *<label>*.
PIQA	Goal: *<goal>* Solution 1: *<sol1>* Solution 2: *<sol2>* Question: Given the goal, what is the correct solution, solution 1 or solution 2? Answer: *<label>*.
HellaSwag	Consider the following description: "*<ctx>*" Choice 1: *<endings1>* Choice 2: *<endings2>* Choice 3: *<endings3>* Choice 4: *<endings4>* Question: Which is the most plausible ending, choice 1, choice 2, choice 3 or choice 4? Answer: *<label>*.
GSM8K	Given the following problem, reason and give a final answer to the problem. Problem: *<question>* Answer: *<label>*
CIFAR10	*<image>*An image of *<label>*.<|*endofchunk*|>
CIFAR100	*<image>*An image of *<label>*.<|*endofchunk*|>
Food101	*<image>*An image of *<label>*.<|*endofchunk*|>
COCO-Color	*<image>*Question: *<question>*? Short answer: *<label>*<|*endofchunk*|>
COCO-Number	*<image>*Question: *<question>*? Short answer: *<label>*<|*endofchunk*|>
ImageNet-100	*<image>*Please identify the class of the image provided. The class has to belong to one of the classes specified in the system prompt