Meta-learning via Language Model In-context Tuning

Anonymous ACL submission

Abstract

The goal of meta-learning is to learn to adapt to a new task with only a few labeled examples. Inspired by the recent progress in large language models, we propose *in-context tuning* (ICT), which recasts task adaptation and prediction as a simple sequence prediction problem: to form the input sequence, we concatenate the task instruction, labeled in-context examples, and the target input to predict; to meta-train the model to learn from in-context examples, we fine-tune a pre-trained language model (LM) to predict the target label given the input sequence on a collection of tasks.

We benchmark our method on two collections of text classification tasks: LAMA and BinaryClfs. Compared to MAML which adapts the model through gradient descent, our method leverages the inductive bias of pre-trained LMs to perform pattern matching, and outperforms MAML by an absolute 6% average AUC-ROC score on BinaryClfs, gaining more advantage with increasing model size. Compared to non-fine-tuned in-context learning (i.e. prompting a raw LM), in-context tuning meta-trains the model to learn from in-context examples. On BinaryClfs, ICT improves the average AUC-ROC score by an absolute 10%, and reduces the variance due to example ordering by 6x and example choices by 2x.

1 Introduction

Few-shot learning (FSL) refers to a system's ability to quickly adapt to new tasks when very few labeled examples are available for training. FSL is a key feature of human learning (Lake et al., 2016), but current machine learning systems often rely on large amounts of labeled training data (Silver et al., 2016; He et al., 2016; Adiwardana et al., 2020).

Recently, prompting large pre-trained language models (LMs) for FSL has achieved remarkable progress (Brown et al., 2020; Schick and Schütze, 2021a). LM prompting with in-context learning reduces the "task learning and predict" process to a simple sequence prediction problem. To perform a new task, Brown et al. (2020) prompt a *raw* LM (i.e., a pre-trained LM not fine-tuned on any labeled data) with the concatenation of the task instruction, some input-output examples, and the target input to be predicted on; then they extract the answer from the LM's continuation of the concatenated sequence (Figure 1 left). For example, to coax the model into performing sentiment classification on the target input "*This movie is a waste of time*", we prompt the LM with the sequence "*I like the movie! Positive review? Yes. Horrible Movie! Positive review? No. This movie is a waste of time. Positive review?* ___", and predict "positive" if the next word is more likely to be "*Yes*" rather than "*No*".

However, raw LMs are not optimized for in-context FSL during pre-training, and exhibit undesirable behavior when used for FSL. For example, Zhao et al. (2021) observed that LMs suffer from the "recency bias", which assigns higher probability to labels that appear closer to the target input. As a result, the accuracy becomes extremely sensitive to the ordering of the in-context examples. Previous work has also shown that prompting raw LMs is often oversensitive to example choices and instruction wording (Schick and Schütze, 2021a; Jiang et al., 2020; Gao et al., 2021; Liu et al., 2021).

We address this weakness through a meta-learning lens and directly fine-tune the LM for FSL. Under the meta-learning framework, we meta-train a model to learn to adapt to new tasks from a few examples on a wide range of tasks, so that it learns to leverage the few-shot examples to adapt to new tasks at test time. Since LM prompting already reduces the "task learning and predict" process to a simple sequence prediction problem, we meta-train a LM by directly fine-tuning it to optimize for this sequence prediction problem on a wide range of tasks (Figure 1 left). Since we fine-tune our model to learn in-context learning, we

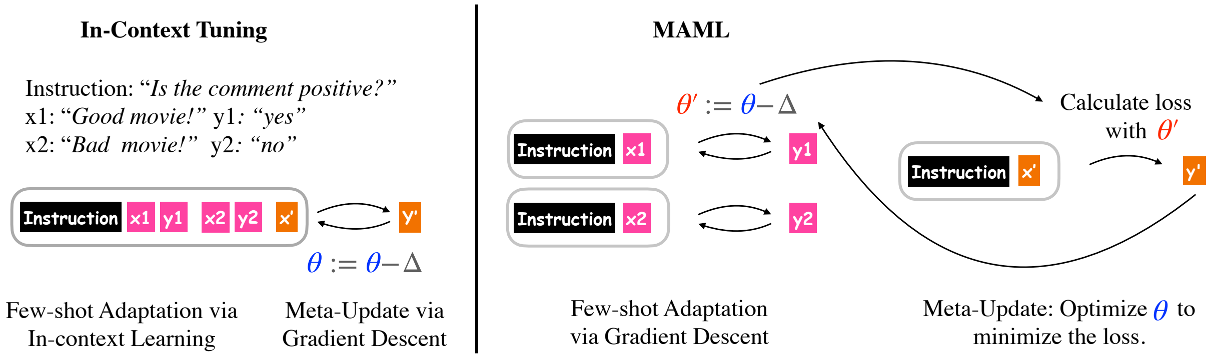

Figure 1: **MAML** (right): MAML aims to learn a task-agnostic model initialization θ that can adapt fast to new tasks. To adapt the model initialization to a new task $\tilde{T}$, a task-specific model θ' initialized with θ is updated with gradient descent using task examples from $\tilde{T}$. Meta-training of MAML involves bi-level optimization, where the inner optimization learns a task-specific model θ' using task examples from $\tilde{T}$, and the outer optimization learns a meta-initialization θ to minimize few-shot prediction loss of θ' on task $\tilde{T}$. **In-context Tuning (ours)** (left): our approach adapts to new tasks via in-context learning, and learns a single model θ shared across all tasks that is directly optimized with the FSL objective (Section 2.2). Because model parameters are frozen during task adaptation, our approach does not involve bi-level optimization during meta-training.

call our approach ***in-context tuning*** (ICT). Unlike optimization-based meta learning approaches such as MAML (Finn et al., 2017), in-context tuning adapts to new tasks through in-context learning where model parameters are frozen, thus it avoids the challenging nested optimization problem in MAML (Figure 1).

We benchmark our algorithm on LAMA (Petroni et al., 2019), a dataset for testing models' factual knowledge, and BinaryClfs (Zhong et al., 2021), a wide range of binary classification tasks each annotated with a few language descriptions of the task. Compared to prompting raw LMs, in-context tuning improves performance by 7.6 Precision@1 points on LAMA and 10.6% AUC-ROC score on BinaryClfs. In addition, in-context tuning mitigates the over-sensitivity of raw LM prompting, significantly reducing the variance of the performance with respect to example ordering (by 68% on LAMA and 83% on BinaryClfs), example choices (by 56% on LAMA and 40% on BinaryClfs), and instruction wording (by 19% on LAMA).

Our approach also out-performs MAML, which adapts the model by gradient descent on a few examples and learns an initialization that can adapt to a new task through a few gradient steps (Finn et al., 2017; Nichol et al., 2018). Since our approach better takes advantage of the inductive bias of LMs to extrapolate from in-context examples, our approach out-performs first-order MAML by 2.8 points on LAMA and 5.1 points on BinaryClfs, with increasing advantage as models become larger.

Given the empirical effectiveness of in-context tuning (Section 4.1), we conjecture that the few-shot learning potential of large LMs (e.g., GPT-3) may be broadly underestimated if prompted without any direct optimization for FSL. We also conjecture that in-context tuning can mitigate various undesirable properties of LM prompting, such as over-sensitivity to example ordering, example choices, and instruction wording (Section 4.2).

2 Approach

We introduce the problem setup (Section 2.1), describe our in-context tuning algorithm (Section 2.2), compare our algorithm to gradient-based adaptation methods (Section 2.3) and other baselines (Section 2.4).

2.1 Problem Setup

We focus on the few-shot classification problem, where the model first learns from a set of training tasks $T \in T_{\text{train}}$, each associated with its natural language instructions I_T and a large amount of task input-output examples $D_T = \{(x_T^i, y_T^i)\}$ (see Figure 1 left for examples). At test time, we ask the model to learn a new task $\tilde{T}$ given its instruction and only a few (K) labeled examples, i.e. $S_{\tilde{T}} \subseteq D_{\tilde{T}}, |S_{\tilde{T}}| = K$. We denote the task input to be predicted at test time as $x_{\tilde{T}}^{\text{target}}$.

Note that "task input" is different from "model input". For example, on the left panel of Figure 1, the task input is "*Good movie!*" while the model input can be a concatenation of the instruction, task inputs and task outputs.

2.2 In-context Tuning Algorithm

In-context tuning directly optimizes pre-trained LMs with the few-shot in-context learning objective (Brown et al., 2020): task-agnostic LMs are meta-trained to perform few-shot in-context learning on a wide variety of training tasks. Similar to in-context learning, LMs trained with in-context tuning adapt to a new task by using few-shot training examples as the input prefix.

Formally, during meta-training, we build the model input by concatenating the task instruction I_T, task input-output pairs $S_T \subseteq D_T$, and the task input x_T^{target}[1] to be classified. We then fine-tune a pre-trained LM to predict y_T^{target} and hope that the model learns to use the in-context examples S_T. Here is the few-shot in-context tuning objective $\mathcal{L}$:

$$\mathcal{L}_T(\theta) := \sum_{(x_T^{\text{tgt}}, y_T^{\text{tgt}}) \in D_T} [-\log p_\theta(y_T^{\text{tgt}} | x_T^{\text{tgt}}, S_T, I_T)] \quad (1)$$

$$\mathcal{L}(\theta) := \sum_{T \in T_{\text{train}}} \mathcal{L}_T(\theta) \quad (2)$$

To adapt to a new task $\tilde{T}$ at test time, we directly concatenate the few-shot examples $S_{\tilde{T}}$ with the instruction $I_{\tilde{T}}$ and the target task input $x_{\tilde{T}}^{\text{target}}$ to be classified to form the model input, and ask the model to predict its corresponding output. No gradient update is performed during adaptation.

2.3 Gradient-based Task Adaptation

We compare in-context tuning with two classical few-shot learning methods: multi-task fine-tuning (instruction tuning + fine-tuning) and MAML. Both methods adapt the model parameters to new tasks by gradient descent on few-shot examples.

Instruction Tuning + Fine-tuning (InsT + FT) We extend the recent work on zero-shot instruction tuning (Wei et al., 2021) to the FSL setting as a *multi-task fine-tuning* baseline. During meta-training, the model is optimized to predict the task output given the task instruction and the task input on a wide range of tasks (Zhong et al., 2021). Formally, we train the model parameter θ to predict y_T^i given $I_T \circ x_T^i$, where θ is shared across all tasks and $\circ$ represents the concatenation operation. During the few-shot adaptation phase, the model is presented with a new task $\tilde{T}$, its natural language instruction $I_{\tilde{T}}$ and a small set of (K) task input-output examples $S_{\tilde{T}} = \{(x_{\tilde{T}}^i, y_{\tilde{T}}^i) | i \in [K]\}$. We then fine-tune the model to predict the task output $y_{\tilde{T}}^i$ from the new task given $I_{\tilde{T}} \circ x_{\tilde{T}}^i$ and update θ with a few gradient steps to get $\theta_{\tilde{T}}$. Finally, we use the updated model $\theta_{\tilde{T}}$ to predict the output from the task input $x_{\tilde{T}}^{\text{target}}$ and the instruction $I_{\tilde{T}}$ under the test task $\tilde{T}$.

MAML The few-shot adaptation stage of MAML is the same as instruction tuning + fine-tuning, where we update the model parameters (initialized with θ) by gradient descent on K examples $S_{\tilde{T}} \subseteq D_{\tilde{T}}$. However, during meta-training, MAML aims to learn a task-agnostic model initialization θ such that, θ_T, which is to be found by initializing with θ and performing gradient descent on S_T, would lead to good performance (Finn et al., 2017).

Therefore, MAML involves two levels of optimization, an inner optimization to learn θ_T given θ and $S_T \subseteq D_T$, and an outer optimization to learn θ given θ_T. Due to the bi-level structure in this optimization problem, MAML has been found to be empirically unstable, sensitive to hyperparameters, and computationally expensive (Finn et al., 2017; Nikolaev et al., 2020). Even worse, few-shot task adaptation is known to be highly sensitive to optimization hyperparameters (Antoniou et al., 2019), while a large labeled validation set for hyperparameter tuning may not be available under a FSL setting (Perez et al., 2021).

In comparison, in-context tuning simplifies the two-stage process of (1) few-shot task adaptation and (2) task-specific prediction as one sequence prediction problem, where task-specific examples are concatenated to the model input to provide information about the task. Hence, in-context tuning removes the bi-level optimization during meta-training, which can be empirically unstable and expensive. Additionally, since model weights are frozen during task adaptation, it is not sensitive to adaptation hyperparameters.

2.4 Other Baselines

Raw In-context Learning (Raw IC-L) We directly evaluate a raw LM on a new task using the same evaluation set-up for in-context tuning, without fine-tuning the LM on any labeled data.

Instruction Tuning (InsT) The model learns to predict the target output only based on the instruc-

[1]We sometimes abbreviate "target" as "tgt" to save space.

Method	Adaptation	Meta-train
In-context Tuning	In-context	Few-shot
MAML	Gradient	Few-shot
InsT	None	Zero-shot
InsT + FT	Gradient	Zero-shot
Raw IC-L	In-context	LM

Table 1: We categorize our approach and the baselines according to 1) how the few-shot examples (if any) are used for adaptation, and 2) the meta-training objective. Ins-T refers to instruction tuning.

tion and the target input. Only the instruction is available during the adaptation phase, and this setup is also known as zero-shot learning.

We categorize all approaches in our paper based on their meta-training objective and how they use task-specific examples in Table 1. In-context tuning is the *only* method that directly optimizes the FSL objective without gradient-based adaptation.

3 Experimental Setup

3.1 Datasets and Metrics

We experiment with two meta-datasets that contain a wide range of tasks, LAMA and BinaryClfs. Each task is associated with several different natural language descriptions, and we call them *instructions* for convenience, even though some of them are realized as questions.

LAMA **LA**nguage **M**odel **A**nalysis (Petroni et al., 2019) is a dataset that tests the factual and commonsense knowledge learned by LMs. In our experiments, we use the TREx-UHN portion of LAMA (Poerner et al., 2020), which consists of (subject, relation, object) triples from Wikidata. LAMA is an entity prediction task, where a model is asked to predict the object entity given the subject entity and the relation. In our experiments, we treat one relation as a task as in Perez et al. (2021).

Initial experiments on LAMA showed that LMs take significant advantage of "majority label bias" (Zhao et al., 2021), where they assign higher probability to object entities that have appeared in the in-context examples, thus inflating the accuracy. To reflect the improvement due to few-shot learning rather than this simple heuristic to copy answers, for all tasks we prune the LAMA dataset so that all object entities appear less than 2.5% of times. Our final filtered LAMA dataset consists of 29 relations (tasks) and 12k (subject, relation, object) examples.

We use task instructions from two datasets: LAMA and LPAQA (Jiang et al., 2020). LAMA contains one task instruction for each task, and the auxiliary LPAQA dataset contains on average 10 additional instructions for each LAMA task.

We use the same evaluation protocol as in Petroni et al. (2019): 1) the object entity is predicted from a pre-defined vocabulary set of 21k words (each LAMA task is 21k-way classification); 2) we compute mean precision at one (P@1) for each task, and report the average across tasks. We report the train-dev-test split in Appendix B.

BinaryClfs This dataset contains a wide range of **binary cl**assi**f**ication task**s**, and each task can be described by 1-4 "yes/no" questions, which we concatenate to the input context as instructions. There are in total 204 different tasks, and 73 of them are used for testing, which include sentiment classification, topic classification, definition detection, stance classification, etc. We use the same evaluation protocol as in Zhong et al. (2021): 1) we group the tasks by similarity and do not allow training tasks to be similar to testing tasks; 2) we treat "Yes" answer as the positive class and calculate the AUC-ROC score for each instruction of each task.

To fit model inputs (concatenation of in-context examples and task input to classify) within the maximum context length (1024) of our LMs, we leave out five evaluation tasks where the maximum task input length exceeds 230 BPE tokens. We also leave out the spam classification task due to its small test set. BinaryClfs does not come with an official validation set. To perform hyperparameter tuning, for each testing group, we randomly sample another testing group as its validation group.

3.2 Implementation Details

Architecture We use BERT models for LAMA (BERT-Base [110M parameters], BERT-Large [340M] and DeBERTa-XLarge-V2 [900M]) and GPT2 models for BinaryClfs (GPT2-Medium [345M] and GPT2-Large [774M]). We use the Huggingface implementation (Wolf et al., 2020).

Hyperparameters We select hyperparameters based on few-shot classification accuracy on validation tasks. Our validation tasks and testing tasks are disjoint, so hyperparameter tuning on validation tasks does not use extra labeled examples on the testing tasks (Perez et al., 2021). See Appendix A for the hyperparameters we tuned.

	LAMA												BinaryClfs			
	BERT-Base				BERT-Large				DeBERTa-xlarge				GPT2-M		GPT2-L	
	0-S	1-S	2-S	5-S	0-S	1-S	2-S	5-S	0-S	1-S	2-S	5-S	0-S	5-S	0-S	5-S
Raw IC-L	10.3	8.5	10.8	14.1	12.7	12.1	15.4	18.6	11.2	12.6	20.6	23.7	50.5	57.8	51.0	58.3
InsT + FT	/	**17.5**	**18.6**	**20.0**	/	**21.6**	22.6	23.9	/	24.7	25.6	27.0	/	67.0	/	69.4
ICT	**14.6**	16.3	17.6	19.6	**18.0**	**21.6**	**23.4**	**24.3**	**21.9**	**26.0**	**27.5**	**28.8**	**62.9**	**67.4**	**66.3**	**69.8**
Raw IC-L w/o Ins	1.5	4.9	8.7	12.3	1.4	3.5	7.0	12.5	2.7	13.0	19.5	22.6	/	/	/	/
ICT w/o Ins	7.1	14.6	17.0	18.2	9.3	19.4	19.9	22.9	10.6	23.5	26.0	27.6	/	/	/	/

Table 2: Few-shot learning accuracy of our in-context tuning approach (ICT) compared to in-context learning with raw LMs (Raw IC-L) and instruction tuning + fine-tuning (InsT + FT). K-S: K-shot learning. GPT2-M: GPT2-Medium. GPT2-L: GPT2-Large. Task instructions are used except the last two rows labeled with "w/o Ins". By definition, InsT + FT is the same as ICT for 0-S. We only experiment with the no-instruction setting on the LAMA dataset. Since we modify the LAMA dataset and BinaryClfs dataset (Section 3.1, Appendix B), the numbers reported in our work are not directly comparable to other work.

	LAMA		BinaryClfs	
	BB	BL	GPT2-M	GPT2-L
MAML	16.9	21.4	63.3	63.9
ICT	**19.6**	**24.3**	**67.4**	**69.8**

Table 3: In-context tuning consistently out-performs MAML on both datasets and all model sizes under the 5-shot setting. BB: BERT-Base. BL: BERT-Large. GPT2-M: GPT2-Medium. GPT2-L: GPT2-Large.

Sampling Different instructions and few-shot example choices can lead to different predictions (Section 2.2). At training time, we expose the model to diverse task instructions and few-shot choices by randomly sampling task instructions and few-shot examples for each target example.

At test time, we report the average accuracy across task instructions and few-shot choices. Since computing the average across all few-shot choices is intractable (there are combinatorically many distinct few-shot choices), we thus calculate the average accuracy of multiple random samplings of few-shot choices as approximation.

4 Results

In-context tuning out-performs MAML and various baselines on the two text classification meta-datasets (Section 4.1). It also significantly reduces model sensitivity to instruction wording, example choices, and example ordering compared to prompting raw LMs (Section 4.2).

4.1 Few-shot Learning Performance

In-context tuning improves in-context learning accuracy over raw LMs. We compare ICT with Raw IC-L in Table 2. In-context tuning consistently out-performs raw LM prompting by 7.6 points on LAMA and 10.6 points on BinaryClfs (averaged across model size and number of few-shots). As expected, directly optimizing the few-shot in-context learning objective (Section 2.2) improves the few-shot in-context learning accuracy.

Few-shot examples lead to more effective task adaptation. We compare few-shot in-context tuning with instruction tuning (equivalent to 0-shot ICT) in Table 2. Few-shot in-context tuning consistently out-performs instruction tuning on both LAMA and BinaryClfs, with increasing performance gains as number of shots increases. Specifically, we observe that 5-shot in-context tuning out-performs instruction tuning by 6.1 points on LAMA and 4.0 points on BinaryClfs. Results show that demonstration examples besides task instructions facilitate more effective task adaptation.

In-context tuning better leverages the inductive bias for pattern matching. By comparing MAML (the first row of Table 3) to instruction tuning (equivalent to 0-shot ICT) of Table 2, we see that MAML out-performs instruction tuning in most evaluation settings, which indicates that MAML is indeed able to take advantage of the few-shot task examples for task adaptation. However, Table 3 shows that our approach of 5-shot in-context tuning out-performs 5-shot MAML consistently on both datasets with an accuracy gain of 2.8 points on LAMA and 5.1 points on BinaryClfs (averaged across model size). We argue that in-context tuning out-performs MAML because in-context tuning better leverages the existing inductive bias of pre-trained LMs to perform pattern matching with in-context examples.

We also compare in-context tuning to the pipeline of instruction tuning + task-specific fine-tuning (Table 2). Surprisingly, fine-tuning an instruction-tuned model on as few as one task-specific example significantly improves task accuracy, without over-fitting to the few labeled examples. We observe that instruction tuning + 1-shot fine-tuning out-performs instruction tuning (equivalent to 0-shot ICT) by 3.1 points on LAMA (Table 2). Our in-context tuning approach performs comparable or better than instruction tuning + fine-tuning, with increasing accuracy gains as models get bigger (Table 2). For DeBERTa-XLarge-v2 (the largest models we use in this work), in-context tuning out-performs InsT + FT across all numbers of shots, achieving an accuracy gain of 1.7 points on LAMA (averaged across all numbers of shots). We conjecture that in-context tuning will be increasingly effective for bigger models that have a stronger inductive bias of pattern matching.

In-context tuning reduces the need of task instructions. As coming up with good task instructions can be hard (Schick and Schütze, 2021a; Jiang et al., 2020), we further investigate the effectiveness of in-context tuning without task instructions (Table 2). In-context tuning is effective in the no-instruction setting as well, consistently out-performing raw in-context learning with no instructions by an average margin of 9.5 points on LAMA. Comparing raw in-context learning with (Raw IC-L) and without instructions (Raw IC-L w/o Ins) (Table 2), we observe that task instructions yield the most significant performance gains when model size is relatively small (+2.5 points on BERT-Base, +7.7 points on BERT-Large, only +0.6 points on DeBERTa-xlarge). We conjecture that smaller models may be weaker at inferring patterns from in-context examples alone compared to larger models, which is why instructions yield larger performance gains on smaller models. On BERT-Base and BERT-Large models where task instructions are most helpful, in-context tuning reduces the improvement gain from task instructions from 5.1 points (raw in-context learning) to 1.8 points (averaged across BERT-Base and BERT-Large), which indicates that in-context tuning reduces the need of task instructions compared to raw in-context learning. However, we note that instructions still yield performance improvement even if in-context tuning is applied.

	LAMA		BinaryClfs	
	BB	**BL**	**GPT2-M**	**GPT2-L**
Raw IC-L	1.82	2.14	9.26	8.84
ICT	**0.66**	**0.61**	**1.41**	**1.58**

Table 4: In-context tuning is significantly less sensitive to example ordering compared to in-context learning with raw LMs.

	LAMA		BinaryClfs	
	BB	**BL**	**GPT2-M**	**GPT2-L**
Raw IC-L	3.74	6.30	18.52	20.33
ICT	**1.78**	**2.57**	**11.46**	**11.62**

Table 5: In-context tuning is significantly less sensitive to example choices compared to in-context learning with raw LMs.

4.2 Sensitivity Analysis

We analyze the sensitivity of in-context tuning accuracy with respect to example ordering, example choices, and instruction wording, and compare it with prompting raw LMs. Let I denote a random selection of task instruction, S_T a random unordered set of few-shot training examples with size K, σ a random permutation of K examples. The accuracy μ is a function of these three random variables, i.e. $\mu : (S_T, \sigma, I) \mapsto [0, 1]$. We can decompose the total variance of μ into its variance w.r.t. each of the three random variables, since they are independent (order variance is independent to choice variance because S_T is *unordered*):

$$
\begin{aligned}
\mathrm{Var}_{S_T,\sigma,I}[\mu] = \; & \underbrace{\mathrm{Var}_I[\mathbb{E}_{S_T,\sigma}[\mu|I]]}_{\text{instruction wording variance}} \\
& + \underbrace{\mathbb{E}_I[\mathrm{Var}_{S_T}[\mathbb{E}_\sigma[\mu|I, S_T]]]}_{\text{example choice variance}} \\
& + \underbrace{\mathbb{E}_{I,S_T}[\mathrm{Var}_\sigma[\mu|I, S_T]]}_{\text{example order variance}}
\end{aligned}
$$

We analyze each type of variance below.

In-context tuning is significantly less sensitive to example ordering. We compare the variance

	BERT-Base		BERT-Large	
	Raw IC-L	ICT	Raw IC-L	ICT
1-shot	35.38	**26.31**	34.03	**28.78**
2-shot	33.79	**25.40**	**17.71**	19.35
5-shot	24.90	**15.64**	6.36	**5.16**

Table 6: In-context tuning is much less sensitive to task instruction wording compared to in-context learning with raw LMs.

with respect to example ordering for in-context tuning and in-context prompting with raw LMs in Table 4. Results show that in-context tuning is significantly less sensitive to ordering of in-context examples compared to in-context prompting with raw LMs, reducing the sensitivity by 68% on LAMA and 83% on BinaryClfs.

In-context tuning is significantly less sensitive to example choices. We compare the variance with respect to example choices for in-context tuning and in-context prompting with raw LMs in Table 5. Results show that in-context tuning is significantly less sensitive to selection of in-context examples compared to in-context prompting with raw LMs across both datasets and all model sizes, reducing the sensitivity by 56% on LAMA and 40% on BinaryClfs (averaged across model sizes). We conjecture that in-context tuning is significantly less sensitive to example ordering and selection because the model is exposed to various example orderings and selections during in-context tuning.

In-context tuning is less sensitive to instruction wording. We report the variance with respect to instruction wording for in-context tuning and in-context prompting with raw LMs in Table 6. Results show that in-context tuning is less sensitive to instruction wording compared to in-context prompting with raw LMs in five out of six evaluation settings, reducing the variance by 19% on LAMA (averaged across model size and number of shots).

We also observe that in-context tuning is especially effective on task instructions with low accuracy under raw in-context learning. For each task, we compute the Pearson correlation between the raw in-context learning accuracy and the accuracy gain from in-context tuning (over raw in-context learning) on all instructions. On the LAMA dataset, we see a strong negative correlation of -0.563 (averaged across all tasks), with p-value < 0.05 on 63% of the tasks. We conjecture that in-context tuning is much less sensitive to instruction wording because the model is exposed to a wide variety of different task instructions during in-context tuning.

In-context examples are complementary to instructions. We observe that in-context tuning is especially effective on task instructions with low accuracy under instruction tuning. For each task, we compute the Pearson correlation between the instruction tuning accuracy and the accuracy gain from in-context tuning (over instruction tuning) on all instructions. On the LAMA dataset, we see a strong negative correlation of -0.910 (averaged across all tasks), with p-value < 0.01 on 91% of the tasks. We conjecture that in-context tuning is much less sensitive to instruction wording because few-shot in-context examples provide additional task information besides the task instructions.

5 Related Work

LM Prompting for FSL Pre-trained LMs can be used to perform various FSL tasks when prompted with a natural language task instruction and several task examples (Radford et al., 2019; Brown et al., 2020; Schick and Schütze, 2021b; Li and Liang, 2021; Lester et al., 2021; Qin and Eisner, 2021). However, prompting pre-trained LMs directly for FSL is known to be sensitive to various artifacts, such as the wording of the task instruction and the selection and ordering of few-shot training examples (Schick and Schütze, 2021a; Jiang et al., 2020; Zhao et al., 2021; Gao et al., 2021; Liu et al., 2021). Our work is the first to show that meta-learning with an explicit FSL objective significantly reduces the sensitivity of LM prompting.

Meta-learning for FSL Meta-learning is a widely used technique in NLP to improve cross-domain transfer (Yu et al., 2018; Geng et al., 2019; Holla et al., 2020; Deng et al., 2020) and cross-task transfer (Gu et al., 2018; Bansal et al., 2020; Dou et al., 2019). Existing optimization-based meta-learning methods mostly perform task adaptation by fine-tuning a task-agnostic model on task-specific examples using gradient descent (Finn et al., 2017; Jiang et al., 2019; Nichol et al., 2018). However, fine-tuning on few-shot task examples is sensitive to hyperparameters (Antoniou et al., 2019) and nested optimization during meta-training is often unstable (Nichol et al., 2018; Antoniou et al., 2019; Rajeswaran et al., 2019). In contrast, our approach performs few-shot task adaptation by using task-specific examples as part of the model input while keeping the model parameters frozen.

Multi-task Learning In multi-task learning, a single model is trained on the union of training sets of multiple tasks to learn a shared representation (Liu et al., 2019). The multi-task model is then fine-tuned on task-specific examples to adapt to new tasks. Multi-task learning is shown to improve performance on various downstream tasks, especially tasks with small training sets (Khashabi et al., 2020; Ye et al., 2021; Aghajanyan et al., 2021). Compared to meta-learning, multi-task learning does not optimize task adaptation directly.

Fine-tuned LMs for Instruction Learning Recent work shows that fine-tuning LMs to learn task instructions on a wide variety of tasks can further leverage the inductive bias of LMs to perform instruction learning (Zhong et al., 2021; Mishra et al., 2021; Wei et al., 2021). Our work is partially inspired by this line of work, but we work under the more generic few-shot meta-learning setting, and show that our approach out-performs both instruction tuning and existing few-shot meta-learning methods (e.g., MAML). While previous work focuses on the accuracy improvement gained from instruction fine-tuning, our work also looks into the well-known over-sensitivity issue of FSL and shows that in-context tuning effectively reduces the sensitivity of FSL with respect to various factors.

Concurrent to our work, Min et al. (2021) also explores in-context tuning under more general Seq2Seq tasks. In comparison, our work compares in-context tuning to a meta-learning baseline MAML, and shows that in-context tuning mitigates the well-known oversensitivity issue of LM prompting. Contrary to our paper, Min et al. (2021) finds that in-context tuning under-performs InsT + FT. This might be because they use many more shots (16-shot), which could give gradient-based methods more advantage.

6 Future Directions

Scaling Up and Broader Applications Our work only considers simple binary classification and knowledge retrieval tasks, at most 5 in-context examples, and models with fewer than 1 billion parameters. Nevertheless, it is straightforward to scale up our framework to a wider and more diverse range of general sequence-to-sequence tasks (Ye et al., 2021), more few-shot examples (which requires a longer context size (Dai et al., 2019; Wang et al., 2020)), and larger models (Brown et al., 2020; Kaplan et al., 2020). It is also straightforward to apply in-context tuning to a broader range of scenarios that require adapting to a new setup, e.g., adapting to a new label in classification tasks (Xia et al., 2021), an unseen database in semantic parsing tasks (Suhr et al., 2020; Lee et al., 2021), or a new language pair in machine translation (Gu et al., 2018; Aharoni et al., 2019), etc.

Meta-learning for Robustness Our work assumed that the few-shot training examples come from the same distribution as the test examples, but this assumption does not necessarily hold in practice. For example, the test distribution might constitute new input compositions (Lake and Baroni, 2018), rare subgroups (Sagawa et al., 2019), other types of distribution shifts (Hendrycks and Dietterich, 2019), or even adversarial examples (Kang et al., 2019). More effective meta-learning methods might learn a more robust learning mechanism and combat these generalization challenges.

Understanding In-context Learning Many properties of in-context learning are still unknown. Is in-context learning more robust to distribution shift (Lester et al., 2021)? Can we combine in-context learning and gradient learning to get the benefit of both worlds (Wortsman et al., 2021)?

7 Conclusion

In this work, we propose meta-learning via in-context tuning, which recasts the few-shot learning process of task adaptation and task-specific prediction as a simple sequence prediction problem, where few-shot labeled examples are concatenated with the target example to form the model input. In-context tuning out-performs a wide variety of baselines in terms of accuracy, including raw LM prompting, MAML and instruction tuning. Meanwhile, sensitivity study shows that our FSL approach of in-context tuning is significantly less sensitive to few-shot examples and instruction wording compared to raw LM prompting.

Given the empirical effectiveness of in-context tuning, we conjecture that the few-shot learning potential of large LMs (e.g., GPT-3) might be broadly underestimated, and that in-context tuning can eliminate well-known artifacts of few-shot LM prompting such as over-sensitivity to example ordering, example selection and instruction wording.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot.

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Antreas Antoniou, Harrison Edwards, and Amos Storkey. 2019. How to train your MAML. In *International Conference on Learning Representations*.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2020. Learning to few-shot learn across diverse natural language classification tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5108–5123, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. *Proceedings of the 13th International Conference on Web Search and Data Mining*.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1192–1197, Hong Kong, China. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Learning to learn to disambiguate: Meta-learning for few-shot word sense disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4517–4533, Online. Association for Computational Linguistics.

Xiang Jiang, Mohammad Havaei, Gabriel Chartrand, Hassan Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados, and Stan Matwin. 2019. Attentive task-agnostic meta-learning for few-shot text classification.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. 2019. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*.

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

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2016. Building machines that learn and think like people.

Chia-Hsuan Lee, Oleksandr Polozov, and Matthew Richardson. 2021. KaggleDBQA: Realistic evaluation of text-to-SQL parsers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2261–2273, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3?

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms.

Dmitry Nikolaev, Ofir Arviv, Taelin Karidi, Neta Kenneth, Veronika Mitnik, Lilja Maria Saeboe, and Omri Abend. 2020. Fine-grained analysis of cross-linguistic syntactic divergences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1159–1176, Online. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489.

Alane Suhr, Ming-Wei Chang, Peter Shaw, and Kenton Lee. 2020. Exploring unexplored generalization challenges for cross-database semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8372–8388, Online. Association for Computational Linguistics.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2021. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*.

Congying Xia, Wenpeng Yin, Yihao Feng, and Philip Yu. 2021. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1351–1360, Online. Association for Computational Linguistics.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. *arXiv preprint arXiv:2104.08835*.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections.

A Hyperparameters

In this section, we report the hyperparameters we tuned for our approach and each baseline.

In-Context Tuning (ours) We tune number of training epochs ([10, 15, 30] for LAMA and [1e-7, 3e-7, 1e-6, 3e-6] for BinaryClfs) and learning rate ([1e-7, 3e-7, 1e-6, 3e-6] for LAMA and [3e-6, 1e-5, 3e-5, 1e-4] for BinaryClfs).

MAML We assume that inner optimization and outer optimization use the same learning rate. We tuned number of adapt steps ([1, 2, 4] for both datasets) and learning rate ([3e-7, 1e-6, 3e-6, 1e-5, 3e-5, 1e-4, 3e-4, 1e-3] for LAMA and [3e-6, 1e-5, 3e-5, 1e-4, 3e-4, 1e-3] for BinaryClfs).

Instruction-Tuning + Fine-tuning For instruction tuning we tuned the same set of hyperparameters as in in-context tuning. The instruction tuning model with the highest validation performance are used for downstream task fine-tuning. For task fine-tuning, we tuned number of training epochs ([5, 10, 15, 30, 40] for LAMA and [5, 10, 15, 30, 40] for BinaryClfs) and learning rate ([1e-7, 3e-7, 1e-6, 3e-6, 1e-5, 3e-5] for LAMA and [3e-6, 1e-5, 3e-5, 1e-4, 3e-4, 1e-3] for BinaryClfs).

B Dataset Split of LAMA

Because LAMA does not have an official train-validation-test split, we use 8-fold cross-validation in our experiments. We randomly partition the 29 tasks into 8 groups of similar sizes. For each cross-validation split, we use six groups for training, one group for validation, and one group for testing. The test sets of the eight folds are disjoint and their union is the set of all tasks.