MetaICL: Learning to Learn In Context

Anonymous ACL submission

Abstract

We introduce MetaICL (**Meta**-training for **I**n-**C**ontext **L**earning), a new meta-training framework for few-shot learning where a pretrained language model is tuned to do in-context learning on a large set of training tasks. This meta-training enables the model to more effectively learn a new task in context at test time, by simply conditioning on a few training examples with no parameter updates or task-specific templates. We experiment on a large, diverse collection of tasks consisting of 142 NLP datasets including classification, question answering, natural language inference, paraphrase detection and more, across seven different meta-training/target splits. MetaICL outperforms a range of baselines including in-context learning without meta-training and multi-task learning followed by zero-shot transfer. We find that the gains are particularly significant for target tasks that have domain shifts from the meta-training tasks, and that using a diverse set of the meta-training tasks is key to improvements. We also show that MetaICL approaches (and sometimes beats) the performance of models fully finetuned on the target task training data, and outperforms much bigger models with nearly 8x parameters.

1 Introduction

Large language models (LMs) have recently been shown to be able to do *in-context learning* (Brown et al., 2020), where they learn a new task simply by conditioning on a few training examples and predicting which tokens best complete a test input. This type of learning is attractive because the model learns a new task through inference alone, without any parameter updates. However, performance significantly lags behind supervised finetuning, results are often high variance (Zhao et al., 2021; Perez et al., 2021), and it can be difficult to engineer the templates that convert existing tasks to this format.

In this paper, we address these challenges by introducing MetaICL: **Meta**-training for **I**n-**C**ontext **L**earning. MetaICL tunes a pretrained language model on a large set of tasks to learn how to in-context learn, and is evaluated on strictly new unseen tasks. Each meta-training example matches the test setup—it includes $k+1$ training examples from one task that will be presented together as a single sequence to the language model, and the output of the final example is used to calculate the cross-entropy training loss. Simply finetuning the model in this data setup directly leads to better in-context learning—the model learns to recover the semantics of the task from the given examples, as must be done for in-context learning of a new task at test time. This approach is related to recent work that uses multi-task learning for better zero-shot performance at test time (Khashabi et al., 2020; Mishra et al., 2021b; Zhong et al., 2021; Wei et al., 2021; Sanh et al., 2021). However, MetaICL is distinct as it allows learning new tasks from k examples alone, without relying on a task reformatting (e.g., reducing everything to question answering) or task-specific templates (e.g., converting different tasks to a language modeling problem).

We experiment on a large, diverse collection of tasks taken from Ye et al. (2021) and Khashabi et al. (2020), including 142 text classification, question answering, natural language inference and paraphrase detection datasets. We report seven different settings, all with no overlap between meta-training and target tasks in types of the task (e.g., classification, question answering) or domains (e.g., news, finance, medical). This leads to 52 unique target tasks in total, which is the largest among all recent related work to the best of our knowledge.

Experimental results show that MetaICL consistently outperforms baselines including (1) a variety of LM in-context learning baselines without meta-training (Brown et al., 2020; Zhao et al., 2021; Holtzman et al., 2021; Min et al., 2021), and (2) multi-task learning followed by zero-shot transfer (Zhong et al., 2021; Wei et al., 2021; Sanh

et al., 2021). Gains over multi-task zero-shot transfer are particularly significant when meta-training tasks and target tasks are dissimilar, e.g. there are large differences in task formats, domains, or required skills. This demonstrates that MetaICL enables the model to recover the semantics of the task in context during inference even when the target does not share similarities with meta-training tasks. MetaICL often gets close to (and sometimes beats) the performance of models trained with supervised finetuning on the target datasets, and perform as well as models with 8x parameters. We also perform extensive ablations to identify key ingredients for success of MetaICL such as the number and diversity of meta-training tasks. Finally, we demonstrate MetaICL without any templates is better than recent work using human-written natural instructions, while the best performance is achieved by combining both approaches.

Codes and data will be publicly released at `anonymous/to-be-updated`.

2 Related Work

In-context learning Brown et al. (2020) proposed to use a language model (LM) conditioned on a concatenation of training examples for few-shot learning with no parameter updates. It has been further improved by later work (Zhao et al., 2021; Holtzman et al., 2021; Min et al., 2021), showing promising results on a variety of tasks. However, in-context learning with an LM achieves poor performance when the target task is very different from language modeling in nature or the LM is not large enough. Moreover, it can have high variance and poor worst-case accuracy (Perez et al., 2021; Lu et al., 2021).

Our paper is based on the core idea of in-context learning by conditioning on training examples. We show that, by explicitly training on an in-context learning objective, MetaICL achieves substantial improvements even with smaller LMs.

Meta-training via multi-task learning Our work is broadly inspired by a large body of work in meta-learning (Vilalta and Drissi, 2002; Finn et al., 2017) and multi-task learning (Evgeniou and Pontil, 2004; Ruder, 2017). Prior work has shown that multi-task learning on a large collection of tasks leads to better performance on a new task, either when tested zero-shot (Khashabi et al., 2020; Mishra et al., 2021b; Zhong et al., 2021; Wei et al., 2021) or when further finetuned (Aghajanyan et al., 2021; Ye et al., 2021). In particular, the former is closely related to our work, as it eliminates the need for parameter updates on a target task. However, these zero-shot models are either limited to tasks sharing the same format as training tasks (e.g., a question answering format) (Khashabi et al., 2020; Zhong et al., 2021), or rely heavily on task-specific templates (Mishra et al., 2021b; Wei et al., 2021; Sanh et al., 2021) which are difficult to engineer due to high variance in performance from very small changes (Mishra et al., 2021a).

In this paper, we propose a meta-training method for better in-context learning that improves few-shot performance. We show that it effectively learns semantics of a new task with no manual effort, significantly outperforming zero-shot transfer methods.[1] Furthermore, while Wei et al. (2021) shows that meta-training helps only when the model has 68B or more parameters, our experiments demonstrate improvements with a much smaller model (770M).

Chen et al. (2021), concurrently to our work, propose meta-training for in-context learning. Our approach differs in a number of ways: we remove requirements of human-written templates or instructions, and include more diverse tasks, stronger baselines, and extensive experiments in much larger scale with many meta-training/target splits.

3 MetaICL

We introduce MetaICL: **Meta**-training for **I**n-**C**ontext **L**earning. Table 1 provides an overview of the approach. The key idea is to use a multi-task learning scheme over a large collection of meta-training tasks, in order for the model to learn how to condition on a small set of training examples, recover the *semantics* of a task, and predict the output based on it. Following previous literature (Brown et al., 2020), the training examples are concatenated and provided as an single input to the model, which is feasible for k-shot learning (e.g., $k = 16$). At test time, the model is evaluated on an unseen target task that comes with k training examples, and inference directly follows the same data format as in meta-training.

[1] We show that MetaICL without instructions is still better than zero-shot transfer with instructions, but by using instructions, performance of MetaICL further improves (Section 5.2).

	Meta-training	Inference
Task	C *meta-training* tasks	An unseen *target* task
Data given	Training examples $\mathcal{T}_i = \{(x_j^i, y_j^i)\}_{j=1}^{N_i}$, $\forall i \in [1, C]$ $(N_i \gg k)$	Training examples $(x_1, y_1), \cdots, (x_k, y_k)$, Test input x
Objective	For each iteration, 1. Sample task $i \in [1, C]$ 2. Sample $k+1$ examples from $\mathcal{T}_i$: $(x_1, y_1), \cdots, (x_{k+1}, y_{k+1})$ 3. Maximize $P(y_{k+1}\|x_{k+1}, x_1, y_1, \cdots, x_k, y_k, x_{k+1})$	$\text{argmax}_{c \in \mathcal{C}} P(c\|x_1, y_1, \cdots, x_k, y_k, x)$

Table 1: Overview of MetaICL (Section 3).

3.1 Meta-training

The model is meta-trained on a collection of tasks which we call meta-training tasks. For every iteration, one meta-training task is sampled, and $k+1$ training examples $(x_1, y_1), \cdots, (x_{k+1}, y_{k+1})$ are sampled from the training examples of the chosen task. We then supervise the model by feeding the concatenation of $x_1, y_1, \cdots, x_k, y_k, x_{k+1}$ to the model as an input and train the model to generate y_{k+1} using a negative log likelihood objective. This simulates in-context learning at inference where the first k examples serve as training examples and the last $(k+1)$-th example is regarded as the test example.

3.2 Inference

For a new target task, the model is given k training examples $(x_1, y_1), \cdots, (x_k, y_k)$ as well as a test input x. It is also given a set of candidates $\mathcal{C}$ which is either a set of labels (in classification) or answer options (in question answering). As in meta-training, the model takes a concatenation of $x_1, y_1, \cdots, x_k, y_k, x$ as the input, and compute the conditional probability of each label $c_i \in \mathcal{C}$. The label with the maximum conditional probability is returned as a prediction.

3.3 Channel MetaICL

We introduce a noisy channel variant of MetaICL called Channel MetaICL, following Min et al. (2021). In the noisy channel model, $P(y|x)$ is reparameterized to $\frac{P(x|y)P(y)}{P(x)} \propto P(x|y)P(y)$. We follow Min et al. (2021) in using $P(y) = \frac{1}{|\mathcal{C}|}$ and modeling $P(x|y)$ which allows us to use the channel approach by simply flipping x_i and y_i. Specifically, at meta-training time, the model is given a concatenation of $y_1, x_1, \cdots, y_k, x_k, y_{k+1}$ and is trained to generate x_{k+1}. At inference, the model computes $\text{argmax}_{c \in \mathcal{C}} P(x|y_1, x_1, \cdots, y_k, x_k, c)$.

Meta-train			Target	
Setting	# tasks	# examples	Setting	# tasks
HR	61	819,200	LR	26
Classification	43	384,022	Classification	20
Non-Classification	37	368,768		
QA	37	486,143	QA	22
Non-QA	33	521,342		
Non-NLI	55	463,579	NLI	8
Non-Paraphrase	59	496,106	Paraphrase	4

Table 2: Statistics of seven different settings. Each row indicates meta-training/target tasks for each setting. '# tasks' in meta-training is equivalent to C in Table 1. 'HR' and 'LR' indicate high resource and low resource, respectively. Full datasets for each split are provided in Appendix A.

4 Experimental Setup

4.1 Datasets

We use a large collection of tasks taken from CROSSFIT (Ye et al., 2021) and UNIFIEDQA (Khashabi et al., 2020). We have 142 unique tasks in total, covering a variety of problems including text classification, question answering (QA), natural language inference (NLI) and paraphrase detection.

We experiment with seven distinct settings as shown in Table 2, where there is no overlap between the meta-training and target tasks. The number of unique target tasks in total is 52, which is significantly larger than other relevant work (Khashabi et al., 2020; Mishra et al., 2021b; Zhong et al., 2021; Wei et al., 2021; Sanh et al., 2021).

HR→LR (High resource to low resource): We experiment with a main setting where datasets with 10,000 or more training examples are used as meta-training tasks and the rest are used as target tasks. We think using high resource datasets for meta-training and low resource datasets as targets is a realistic and practical setting for few-shot learning.

Method	Meta	Target	
	train	train	# samples
LMs			
0-shot	✗	✗	0
PMI 0-shot	✗	✗	0
Channel 0-shot	✗	✗	0
In-context	✗	✗	k
PMI In-context	✗	✗	k
Channel In-context	✗	✗	k
Meta-trained			
Multi-task 0-shot	✓	✗	0
Channel Multi-task 0-shot	✓	✗	0
MetaICL (Ours)	✓	✗	k
Channel MetaICL (Ours)	✓	✗	k
Oracle			
Oracle	✗	✓	k
Oracle w/ meta-train	✓	✓	k

Table 3: Summary of the baselines and MetaICL. 'train' indicates whether the model is trained with parameter updates, and '# samples' indicates the number of training examples used on a target task. Our baselines include a range of recently introduced methods (Holtzman et al., 2021; Zhao et al., 2021; Min et al., 2021; Wei et al., 2021) as described in Section 4.2.

[P]: Time Warner is the world's largest media and Internet company. [H]: Time Warner is the world's largest company. Labels: `entailment`, `not_entailment`
Holtzman et al. (2021) Input [P] question: [H] true or false? answer: Output {true, false}
Wei et al. (2021) Input [P] Based on the paragraph above, can we conclude that [H]? Output {yes, no}
Ours Input [P] [H] Output {entailment, not_entailment}

Table 4: Example input-output pairs for an NLI task. We show human-authored templates taken from prior work as references.

X→X (X={Classification, QA}): We also experiment with two settings with meta-training and target tasks sharing the task format, although with no overlap in tasks.

Non-X→X (X={Classification, QA, NLI, Paraphase}): Lastly, we experiment with four settings where meta-training tasks do not overlap with target tasks in task format and required capabilities. These settings require the most challenging generalization capacities.

Each setting has a subset of target tasks with no domain overlap with any meta-training tasks (e.g., finance, poem, climate or medical). We evaluate both on all target tasks or on target tasks with no domain overlap only. Full details of the settings and datasets with citations are provided in Appendix A.

4.2 Baselines

We compare MetaICL and Channel MetaICL with a range of baselines, as summarized in Table 3.

0-shot: We use a pretrained LM as it is and run zero-shot inference, following Brown et al. (2020).

In-context: We use the pretrained LM as it is and use in-context learning by conditioning on a concatenation of k training examples, following Brown et al. (2020).

PMI 0-shot, PMI In-context: We use the PMI method from Holtzman et al. (2021); Zhao et al. (2021) for 0-shot and In-context learning.

Channel 0-shot, Channel In-context: We use the noisy channel model from Min et al. (2021) for 0-shot and In-context learning.

Multi-task 0-shot: We train the LM on meta-training tasks and use zero-shot transfer on a target task, as done in Khashabi et al. (2020); Zhong et al. (2021); Wei et al. (2021).

Channel Multi-task 0-shot: We have a noisy channel variant of Multi-task 0-shot.

Oracle: We train the LM on a given target task. This is not directly comparable to other methods as parameter updates are required for every target task.

Oracle w/ meta-train: We train the LM on meta-training tasks first and then further finetuned on a target task. This is not directly comparable to other methods for the same reason as above.

4.3 Evaluation

We use Macro-F1[2] and Accuracy as evaluation metrics for classification tasks and non-classification tasks, respectively.

For a target task, we use $k = 16$ training examples, sampled uniformly at random. We relax the assumption of perfect balance between labels on k training examples, following Min et al. (2021). Because in-context learning is known to have high variance (Zhao et al., 2021; Perez et al., 2021; Lu et al., 2021), we use 5 different sets of k training examples. We first compute the average and the worst-case performance over seeds for every target

[2]More suitable than accuracy for imbalanced classification.

Method	HR→LR	Class →Class	non-Class →Class	QA →QA	non-QA →QA	non-NLI →NLI	non-Para →Para
			All target tasks				
0-shot	34.9	34.2	34.2	40.4	40.4	25.5	34.2
PMI 0-shot	34.8	33.2	33.2	40.4	40.4	27.9	39.2
Channel 0-shot	36.8	37.2	37.2	39.2	39.2	33.9	39.5
In-context	38.2/35.4	37.4/33.9	37.4/33.9	40.2/38.8	40.2/38.8	34.0/28.3	33.7/33.1
PMI In-context	38.9/33.3	38.3/29.3	38.3/29.3	40.5/38.9	40.5/38.9	33.0/28.0	38.6/33.4
Channel In-context	42.9/38.5	46.3/40.6	46.3/40.6	40.5/37.9	40.5/37.9	39.9/34.8	45.4/40.9
Multi-task 0-shot	35.4	37.3	36.2	45.7	35.8	40.7	30.6
Channel Multi-task 0-shot	38.6	40.8	42.2	42.1	36.5	36.8	35.1
MetaICL	43.2/41.6	43.4/39.9	38.2/31.8	**45.9**/44.8	38.7/36.9	49.0/44.8	33.1/33.1
Channel MetaICL	**48.7**/46.4	**50.5**/47.7	**49.9**/47.5	45.0/43.6	**42.1**/40.8	**54.6**/51.9	**52.2**/50.3
Oracle	46.4/40.0	50.7/44.0	50.7/44.0	41.8/39.1	41.8/39.1	44.3/32.8	54.7/48.9
Oracle w/ meta-train	52.0/47.9	53.5/48.5	51.2/44.9	46.7/44.5	41.8/39.5	57.0/44.6	53.7/46.9
			Target tasks in unseen domains				
0-shot	32.7	32.7	32.7	45.9	45.9	33.4	38.3
PMI 0-shot	25.8	25.8	25.8	44.4	44.4	33.4	32.9
Channel 0-shot	28.5	28.5	28.5	41.6	41.6	33.1	32.6
In-context	30.4/27.5	30.4/27.5	30.4/27.5	45.6/44.7	45.6/44.7	52.0/41.3	35.8/34.1
PMI In-context	32.0/24.3	32.0/24.3	32.0/24.3	45.4/44.7	45.4/44.7	47.8/35.2	38.5/33.3
Channel In-context	39.4/35.0	39.4/35.0	39.4/35.0	44.7/40.6	44.7/40.6	40.4/35.7	44.1/36.8
Multi-task 0-shot	35.6	28.0	25.5	**71.2**	40.3	33.5	35.0
Channel Multi-task 0-shot	35.1	30.7	34.2	54.4	39.4	50.8	34.1
MetaICL	35.7/32.7	32.3/29.3	28.7/25.1	69.9/68.1	**48.2**/47.2	**80.1**/77.2	34.0/34.0
Channel MetaICL	**44.8**/41.8	**40.9**/36.3	**44.5**/42.2	57.9/56.6	47.2/45.0	62.0/57.3	**51.0**/49.9
Oracle	44.9/37.6	44.9/37.6	44.9/37.6	43.6/39.1	43.6/39.1	56.3/33.4	56.6/51.6
Oracle w/ meta-train	53.3/43.2	53.2/43.7	46.1/36.9	67.9/66.2	44.5/42.8	71.8/58.2	65.6/61.4

Table 5: Main results, using GPT-2 Large. Two numbers indicate the average and the worst-case performance over different seeds used for k target training examples. **Bold** indicates the best average result except oracle. 'Class' indicates 'Classification'.

task, and then report the macro-average of them over all target tasks.

4.4 Experiment Details

As a base LM, we use GPT-2 Large (Radford et al., 2019) which consists of 770M parameters.[3] For baselines without meta-training (raw LMs), we also compare with GPT-J (Wang and Komatsuzaki, 2021), which is the largest public causal LM at the time of writing, consisting of 6B parameters.

Elimination of templates Prior work uses human-authored templates to transform the input-output pair to a natural language sentence (Mishra et al., 2021b; Zhong et al., 2021; Wei et al., 2021; Chen et al., 2021). They require expensive manual effort (as 136 different templates are required for 136 tasks in this paper) and cause unstable model performance due to many different ways of writing (Mishra et al., 2021a). We eliminate templates, using the given input (or a concatenation of inputs if there are multiple) and label words provided in the original datasets. A comparison of input-output schemes from prior work and our approach is shown in Table 4.

More details about preprocessing and training can be found in Appendix B.

5 Experimental Results

5.1 Main Results

Table 5 reports the full results using GPT-2 Large, where we compute the average and the worst-case performance of every target task and report the macro-average over them. The top and the bottom respectively evaluate on all target tasks and target tasks in unseen domains only.

Our baselines are strong We first discuss the results of ours baselines. Among raw LMs without meta-training (the first six rows of Table 5), we observe that channel in-context baselines are the most competitive, consistent with findings from Min et al. (2021). Multi-task 0-shot baselines do not outperform the best raw LM baseline in most settings, despite being supervised on a large set of meta-training tasks. This somewhat contradicts

[3] Appendix C.2 reports performance for other LM sizes.

Method	HR→LR	Class →Class	non-Class →Class	QA →QA	non-QA →QA	non-NLI →NLI	non-Para →Para
All target tasks							
Channel In-context	42.9/38.5	46.3/40.6	46.3/40.6	40.5/37.9	40.5/37.9	39.9/34.8	45.4/40.9
MetaICL	43.2/41.6	43.4/39.9	38.2/31.8	45.9/44.8	38.7/36.9	49.0/44.8	33.1/33.1
Channel MetaICL	**48.7**/46.4	50.5/47.7	49.9/47.5	45.0/43.6	42.1/40.8	**54.6**/51.9	**52.2**/50.3
Channel In-context	48.0/44.0	**50.9**/46.4	**50.9**/46.4	**47.0**/45.2	**47.0**/45.2	47.2/41.7	51.0/47.5
Target tasks in unseen domains							
Channel In-context	39.4/35.0	39.4/35.0	39.4/35.0	44.7/40.6	44.7/40.6	40.4/35.7	44.1/36.8
MetaICL	35.7/32.7	32.3/29.3	28.7/25.1	**69.9**/68.1	48.2/47.2	**80.1**/77.2	34.0/34.0
Channel MetaICL	**44.8**/41.8	**40.9**/36.3	**44.5**/42.2	57.9/56.6	47.2/45.0	62.0/57.3	51.0/49.9
Channel In-context	39.7/35.5	39.7/35.5	39.7/35.5	55.8/54.4	**55.8**/54.4	51.1/40.4	**52.0**/46.5

Table 6: Comparison between raw LM in-context learning (based on GPT-2 Large and GPT-J) and MetaICL (based on GPT-2 Large). GPT-2 Large used unless otherwise specified. Two numbers indicate the average and the worst-case performance over different seeds used for k target training examples. For raw LM baselines, Channel In-context is reported because it is the best raw LM baseline overall across the settings; full results based on GPT-J are provided in Appendix C.1.

findings from Wei et al. (2021); Sanh et al. (2021). This is likely for two reasons. First, our models are much smaller than theirs (770M vs. 11B–137B); in fact, Wei et al. (2021) reports Multi-task 0-shot starts to be better than raw LMs only when the model size is 68B or larger. Second, we compare with much stronger channel baselines which they did not; Multi-task 0-shot outperforms non-channel LM baselines but not channel LM baselines.

MetaICL outperforms baselines MetaICL and Channel MetaICL consistently outperform a range of strong baselines. In particular, Channel MetaICL achieves the best performance in 6 out of 7 settings. Gains are particularly significant in the HR→LR, non-NLI→NLI and non-Para→Para settings (6–14% absolute). This is noteworthy because HR→LR targets the common low-resource case where new tasks have very few labeled examples, and the other two represent large data distribution shifts where the test tasks are relatively different from the meta-training tasks. This demonstrates that MetaICL can infer the semantics of new tasks in context even when there are no closely related training tasks.

While MetaICL significantly outperforms baselines in most settings, it is comparable to Multi-task 0-shot in the QA→QA setting, as an exception. This is likely because the meta-training and target tasks are relatively similar, allowing the Multi-task 0-shot baseline to achieve very strong performance. Nonetheless, performance of Multi-task 0-shot in QA significantly drops when the model is trained on non-QA tasks, while performance of MetaICL drops substantially less.

Gains are larger on unseen domains Gains over Multi-task 0-shot are more significant on target tasks in unseen domains. In particular, Multi-task 0-shot is generally less competitive compared to raw LM baselines, likely because they require more challenging generalization. MetaICL suffers less from this problem and is consistently better or comparable to raw LM baselines across all settings.

Comparison to oracle MetaICL matches or sometimes even outperforms performance of oracle without meta-training. This is a promising signal, given that no prior work has shown models with no parameter updates on the target can match or outperform supervised models. Nonetheless, oracle with meta-training outperforms oracle without meta-training—so meta-training also helps in supervised learning—as well as MetaICL. This hints that there is still room for improvement in methods that allow learning without parameter updates .

Comparison to GPT-J In Table 6, we compare GPT-2 Large based models with raw LM baselines based on GPT-J which consists of 6B parameters. MetaICL, despite being 8x smaller, outperforms or matches GPT-J baselines.

5.2 Ablations

Varying number of training examples We vary the number of training examples (k) from 0, 4, 8, 16 to 32. In-context learning with $k = 0$ is equivalent to the zero-shot method. Results are shown in Figure 1. Increasing k generally helps across all models, and Channel MetaICL outperforms the raw in-context learning over all values of k. We additionally find that the performance tends to saturate

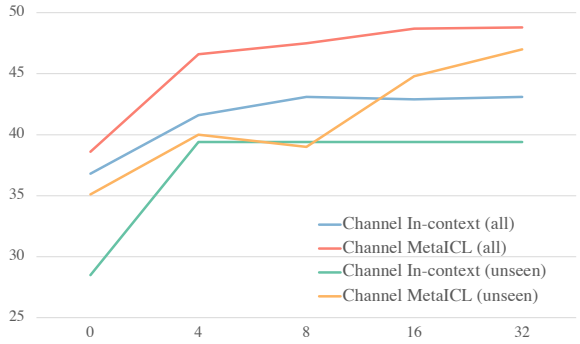

Figure 1: Ablation on the number of training examples (k) in the HR→LR setting. $k = 0$ is equivalent to the zero-shot methods.

when k is closer to 16, likely because the sequence length limit of the language model makes it hard to encode many training examples.

Number of meta-training tasks To see the impact of the number of meta-training tasks, we subsample $\{7, 15, 30\}$ meta-training tasks out of 61 in the HR→LR setting. For each, we use ten different random seeds to additionally see the impact of the choice of meta-training tasks.

Figure 2 reports the results. On average, performance generally increases as the number of tasks increase, which is consistent with results in Mishra et al. (2021b); Wei et al. (2021). Across different numbers of meta-training tasks, Channel MetaICL consistently outperforms other models. Nonetheless, there is nonnegligible variance across different choices of meta-training (the bottom of Figure 2), indicating that a choice of meta-training gives substantial impact in performance.

Diversity in meta-training tasks We hypothesize that the diversity in meta-training tasks may impact performance of MetaICL. To verify this hypothesis, we create two settings by subsampling 13 out of 61 meta-training datasets in the HR→LR setting. One setting is diverse in their task formats and required capacities: QA, NLI, relation extraction, sentiment analysis, topic classification, hate speech detection and more. The other setting is less diverse, including tasks related to sentiment analysis, topic classification and hate speech detection only. A full list of datasets is reported in Appendix A. Using these two settings, we compare multi-task zero-shot transfer baselines and MetaICL.

Results are reported in Table 7. We find that MetaICL with a diverse set outperforms MetaICL with a non-diverse set by a substantial margin. This shows that diversity among meta-training tasks

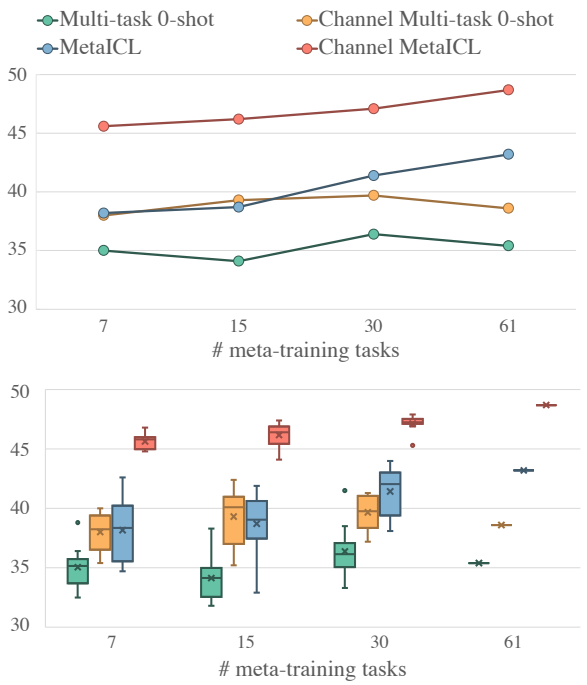

Figure 2: Ablation on the number of meta-training tasks ($\{7, 15, 30, 61\}$). The graph of the average (top) and the box chart (bottom) over different meta-training sets using 10 different random seeds (except for 61).

Method	Diverse	No Diverse
0-shot	34.9	
PMI 0-shot	34.8	
Channel 0-shot	36.8	
In-context	38.2/35.4	
PMI In-context	38.9/33.3	
Channel In-context	42.9/38.5	
Multi-task 0-shot	35.2	29.9
Channel Multi-task 0-shot	41.6	38.3
MetaICL	45.6/43.4	38.8/35.4
Channel MetaICL	**47.2**/44.7	45.3/42.6

Table 7: Ablation on the diversity of meta-training tasks in the HR→LR setting. For both settings, the number of meta-training tasks is 13, and the number of target tasks is 26 as in the original HR→LR setting. A full list of meta-training tasks is shown in Appendix A.

is one of substantial factors for the success of MetaICL. In Appendix C.3, we include ablations that provide more insights on the choice of meta-training tasks.

Are instructions necessary? Most recent work has used human-written natural instructions for zero- or few-shot learning (Mishra et al., 2021b; Wei et al., 2021; Sanh et al., 2021). While we argue for not using instructions to avoid manual engineering and high variance, we also ask: *are instructions still useful with MetaICL?* On one hand, learning to

Method	w/o Instruct	w/ Instruct	
# instruct/task	0	1	8.3
0-shot	33.4	34.2	
PMI 0-shot	33.6	27.8	
Channel 0-shot	32.3	30.6	
In-context	34.4/31.5	45.2/42.3	
PMI In-context	36.7/31.6	41.9/37.6	
Channel In-context	39.0/35.9	39.6/35.3	
MT 0-shot	35.7	32.6	37.1
Channel MT 0-shot	36.6	30.6	36.0
MetaICL	40.6/38.0	42.6/41.0	43.2/41.0
Channel MetaICL	41.3/39.4	45.3/43.9	**46.9**/44.2

Table 8: Ablation on the impact of *natural instructions*. 'w/ Instruct' uses instructions from Sanh et al. (2021), either one per meta-training task or all available ones; 'w/o Instruct' does not use instructions, as in all of our other experiments. '# instruct/task' indicates the number of instructions per meta-training task on average. 'MT 0-shot' indicates 'Multi-task 0-shot' baselines. Both settings have the same meta-training and target tasks, 32 and 12, respectively. A full list of tasks is shown in Appendix A.

condition on k examples may remove the necessity of instructions. On the other hand, instructions may still be complementary and provide the model with extra useful infomration.

We aim to answer this question by using 32 meta-training tasks and 12 target tasks from the HR→LR setting for which human-written instructions are available in Sanh et al. (2021).[4] We have two variants: (a) using one instruction per meta-training task, and (b) using all available instructions, which includes 267 instructions in total (8.3 per meta-training task) which Sanh et al. (2021) found to be better than (a). We then compare MetaICL and a range of baselines with and without instructions.

Results are reported Table 8. As in Wei et al. (2021) and Sanh et al. (2021), Multi-task 0-shot outperforms the raw-LM 0-shot baseline. However, MetaICL with no instructions is better than Multi-task 0-shot with instructions. Furthermore, MetaICL achieves further improvements when instructions are jointly used, significantly outperforming all baselines. In fact, when increasing the number of instructions per task from 0, 1 to 8.3, performance of MetaICL improves much more than performance of Multi-task 0-shot does. To summarize, (1) learning to in-context learn (MetaICL) outperforms learning to learn from instructions; (2) MetaICL and using instructions are largely complementary, and (3) MetaICL actually benefits more from using instructions than Multi-task 0-shot does.

Importantly, Channel MetaICL trained on available tasks and instructions still achieves lower performance than Channel MetaICL without templates/instructions (46.9 from Table 8 vs. 48.7 from Table 5). This is likely because the model with instructions was trained with less meta-training tasks, which was unavoidable since instructions are only available on 32 out of 61 meta-training tasks. This supports our earlier choice of not using human-written templates/instructions, since writing templates and instructions for every task requires extensive effort.

It is worth noting that, it is nonetheless difficult to make direct comparisons with Wei et al. (2021) and Sanh et al. (2021) because there are many moving components: size of LMs, types of LMs (e.g., causal LM vs. masked LM), splits between meta-training and target tasks, and more.

6 Conclusion

In this paper, we introduced MetaICL, a new few-shot learning method where an LM is meta-trained to learn to in-context learn—condition on training examples to recover the task and make predictions. We experiment with a large, diverse collection of tasks, consisting of 142 unique tasks in total and 52 unique target tasks, using seven different settings. MetaICL outperforms a range of strong baselines including in-context learning without meta-training and multi-task learning followed by zero-shot transfer, and outperforms or matches 8x bigger models. We identify ingredients for success of MetaICL such as the number and diversity of meta-training tasks. We also demonstrate that, while MetaICL is better than recent work using natural instructions, they are complementary and the best performance is achieved by combining two ideas.

One critical limitation of in-context learning approaches is that more (and longer) training examples are difficult to use of due to the LM input length limit; we leave this challenge to future work. Other avenues for future work include further improving MetaICL to outperform supervised models with meta-training, identification of which meta-training tasks are helpful on target tasks, and how to better combine human-written instructions and MetaICL.

[4] github.com/bigscience-workshop/promptsource

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. *arXiv preprint arXiv:2101.11038*.

Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: New collection and results. In *Proceedings of the 11th ACM Symposium on Document Engineering*.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *EMNLP*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *ICLR*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.

Michael Boratko, Xiang Li, Tim O'Gorman, Rajarshi Das, Dan Le, and Andrew McCallum. 2020. ProtoQA: A question answering dataset for prototypical common-sense reasoning. In *EMNLP*.

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. 2020. Language models are few-shot learners. In *NeurIPS*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. CODAH: An adversarially-authored question answering dataset for common sense. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. In *ICLR*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2021. Meta-learning via language model in-context tuning. *arXiv preprint arXiv:2110.07814*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *NAACL-HLT*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *EMNLP*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2021. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*.

T. Diggelmann, Jordan L. Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. *ArXiv*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *NAACL*.

Matthew Dunn, Levent Sagun, Mike Higgins, V. U. Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *ArXiv*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *LREC*.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Manaal Faruqui and Dipanjan Das. 2018. Identifying well-formed natural language questions. In *EMNLP*.

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

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *The First Joint Conference on Lexical and Computational Semantics (SemEval)*.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *EMNLP*.

Ari Holtzman, Peter West, Vered Schwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *EMNLP*.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *EMNLP*.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *NAACL-HLT*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL-HLT*.

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 EMNLP*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2019. QASC: A dataset for question answering via sentence composition. In *AAAI*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *AAAI*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *AAAI*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *TACL*.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *EMNLP*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *TACL*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: Large-scale reading comprehension dataset from examinations. In *EMNLP*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, D. Kontokostas, Pablo N. Mendes, Sebastian Hellmann, M. Morsey, Patrick van Kleef, S. Auer, and C. Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *CoNLL*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *EMNLP*.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*.

Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding indirect answers. In *EMNLP*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.*

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *LREC*.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *arXiv preprint arXiv:2012.10289*.

Julian McAuley and J. Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. *Proceedings of the 7th ACM conference on Recommender systems*.

Clara H. McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. 2020. Effective transfer learning for identifying similar questions: Matching user questions to covid-19 faqs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2017. Mixed precision training. In *ICLR*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Noisy channel language model prompting for few-shot text classification. *arXiv preprint arXiv:2108.04106*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021a. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021b. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset. *ArXiv*.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *EMNLP*.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *EMNLP*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *ACL*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*.

Dimitris Pappas, Petros Stavropoulos, Ion Androutsopoulos, and Ryan McDonald. 2020. BioMRC: A dataset for biomedical machine reading comprehension. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

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

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *EMNLP*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *NAACL-HLT*.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *AAAI*.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020a. WINOGRANDE: an adversarial winograd schema challenge at scale. In *AAAI*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020b. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019a. Social IQa: Commonsense reasoning about social interactions. In *EMNLP*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social iqa: Commonsense reasoning about social interactions. In *EMNLP-IJCNLP*.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *EMNLP*.

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining discourse markers for unsupervised sentence representation learning. In *NAACL-HLT*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *TACL*.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019a. Quarel: A dataset and models for answering questions about qualitative relationships. In *AAAI*.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019b. QuaRTz: An open-domain dataset of qualitative relationship questions. In *EMNLP*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL-HLT*.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. In *EMNLP*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Rep4NLP@ACL*.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*.

Ricardo Vilalta and Youssef Drissi. 2002. A perspective view and survey of meta-learning. *Artificial intelligence review*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *ACL*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *TACL*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *TACL*.

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. *arXiv preprint arXiv:2109.01652*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*.

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. Transformers: State-of-the-art natural language processing. In *EMNLP: System Demonstrations*.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A social media focused question answering dataset. In *ACL*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *EMNLP*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

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

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *EMNLP*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *ACL*.

Sheng Zhang, X. Liu, J. Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *ArXiv*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NeurIPS*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *NAACL-HLT*.

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

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. In *Findings of EMNLP*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *EMNLP*.

A Dataset List

Table 13 and Table 14 report a list of datasets used in the settings detailed in Section 4.1. The first 10 rows are for settings described in Section 4.1; the next two rows are for settings used for ablations on the diversity of meta-training tasks (Table 7 of Section 5.2); the last two rows are for settings used for ablations on using natural instructions (Table 8 of Section 5.2). **Bold** datasets are target datasets with no overlap in domain with meta-training tasks. All datasets are taken from CROSSFIT (Ye et al., 2021) (except we exclude datasets that are unavailable from their repository[5] or the scope is notably different from other tasks, e.g., solving math problems or breaking down compositional questions) and UNIFIEDQA (Khashabi et al., 2020).

How meta-training/target splits are determined The HR→LR setting is created based on the training data size as described in Section 4.1. Settings involving Classification, NLI and Paraphrase are taken from CROSSFIT. Settings involving QA are created by combining QA datasets from CROSSFIT and datasets from UNIFIEDQA.

Statistics are reported in Table 2 and Table 9. The number of tasks is the largest among recent related work: we have 142 unique tasks, while Khashabi et al. (2020), Mishra et al. (2021b), Zhong et al. (2021), Wei et al. (2021) and Sanh et al. (2021) use 32, 61, 62, 42 and 62 tasks, respectively. References for all datasets are provided in Table 14. Data and splits are available at `anonymous/to-be-updated`.

B Implementation Details

Preprocessing details For all models with meta-training and the raw GPT-J, we separate the input and the output with one newline (`\n`), and separate between examples with three newlines. For the raw GPT-2, we use spaces instead of newlines. This choice was made in order to report the best baseline performance we were able to achieve: when raw LMs are used, GPT-2 is significantly better with spaces than with newlines, and GPT-J is significantly better with newlines than with spaces.[6] We note that MetaICL is less sensitive to these formatting differences: Channel MetaICL achieves 47.0/43.0 with spaces and 48.7/46.4 with newlines.

When the concatenation of k examples is too long, we truncate each example to have at most 256 tokens, and truncate the earlier tokens of the concatenation so that the LM sees the recent tokens. Additionally, for extractive question answering datasets as meta-training tasks, the input passage is truncated with a guarantee that the groundtruth answer is included in the input passage. We do not do this truncation for target datasets.

Training details All implementation is done in PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020). For meta-training, we use up to 16,384 training examples per task. We use a batch size of 8, learning rate of 1×10^{-5} and a sequence length of 1024. For the baselines with no in-context learning, we use a sequence length of 256. We train the model for $30,000$ steps. To save memory during meta-training, we use an 8-bit approximation (Dettmers et al., 2021) of an Adam optimizer (Kingma and Ba, 2015) and mixed precision (Micikevicius et al., 2017). Training was done for 4.5 hours with eight 32GB GPUs. Note that this is drastically more efficient than recent prior work, e.g., 270 hours of a 512GB TPU in Sanh et al. (2021).

Ablations in using instructions When we choose one instruction per task at meta-training tasks, we choose one by (1) first excluding the instruction if its name contains `no_option`,

Setting	Input		Output	
	Mean	Median	Mean	Median
Meta-training tasks				
HR	81.7	73	2.8	2
Classification	45.8	41	1.1	1
Non-Classification	77.7	69	4.2	3
QA	142.6	137	2.7	2
Non-QA	68.7	56	2.3	2
Non-NLI	44.3	39	1.1	1
Non-Paraphrase	45.0	39	1.1	1
Target tasks				
LR	29.7	25	1.9	1
Classification	44.9	38	1.0	1
QA	74.4	69	4.6	4
NLI	45.4	41	1.0	1
Paraphrase	42.2	41	1.0	1

Table 9: Length statistics of tasks used in different settings, before any truncation. We compute the mean and the median of each task, and report the macro-average over all tasks for each setting.

[5] `github.com/INK-USC/CrossFit`

[6] In the HR→LR setting, the raw GPT-2 achieves 42.0/36.9 with spaces and 38.8/32.6 with newlines, and the raw GPT-J achieves 43.7/38.2 with spaces and 48.0/44.0 with newlines (all with the channel in-context learning method).

Method	HR→LR	{Class,non-Class} →Class	{QA,non-QA} →QA	non-NLI →NLI	non-Para →Para
		All tasks			
0-shot	30.1	27.6	42.9	25.7	30.0
PMI 0-shot	35.4	30.6	41.6	30.2	37.6
Channel 0-shot	36.4	36.8	40.3	36.2	45.0
In-context	43.6/39.0	43.4/34.3	**50.6**/48.1	35.0/27.6	41.3/33.2
PMI In-context	42.9/37.4	44.8/36.7	48.7/46.8	31.5/26.0	38.4/33.6
Channel In-context	**48.0**/44.0	**50.9**/46.4	47.0/45.2	**47.2**/41.7	**51.0**/47.5
		Target tasks in unseen domains			
0-shot	18.0	18.0	47.5	33.5	34.1
PMI 0-shot	16.9	16.9	43.8	36.1	34.4
Channel 0-shot	34.3	34.3	46.9	**53.4**	**54.7**
In-context	32.4/25.4	32.4/25.4	**57.4**/53.1	46.7/36.1	34.1/34.1
PMI In-context	35.5/28.6	35.5/28.6	54.5/50.9	33.9/33.9	32.5/32.4
Channel In-context	**39.7**/35.5	**39.7**/35.5	55.8/54.4	51.1/40.4	52.0/46.5

Table 10: Performance of raw LM baselines using GPT-J. Two numbers indicate the average and the worst-case accuracy over different seeds used for k target training examples. 'Class' indicate 'Classification'.

	All tasks				*Target tasks in unseen domains*			
	S	M	L	XL	S	M	L	XL
Channel In-context	41.5/37.4	42.2/37.7	42.9/38.5	43.5/39.9	40.9/35.9	38.8/34.7	39.4/35.0	40.0/37.2
MT 0-shot	35.4	36.4	35.4	-	34.9	32.2	35.6	-
Channel MT 0-shot	40.4	37.9	38.6	-	33.8	35.9	35.1	-
MetaICL	39.7/36.2	40.3/36.4	43.2/41.6	-	36.9/32.6	38.1/35.0	35.7/32.7	-
Channel MetaICL	46.2/43.1	44.3/41.5	48.7/46.4	-	46.9/42.6	43.1/39.8	44.8/41.8	-

Table 11: Ablation on the size of the LM on the HR→LR setting. We use small, medium, large, and XL variants of GPT-2. We only report the raw LM results for the XL variant: we were unable to meta-train the model due to memory limit.

(2) then taking the instruction which name contains `multiple_choice`, `most_correct` or `most_suitable` if there are any, and (3) if not, then randomly sampling one. We choose one instruction per target task at test time using the same process. This is different Sanh et al. (2021) where the median of the performance over all instructions is reported. We think our choice better reflects the real use-case scenario—choosing one instruction that looks the most reasonable to human.

C Additional Results & Analyses

C.1 GPT-J results

Table 10 reports the full results of raw LM baselines based on GPT-J, the largest publicly available causal LM at the time of writing, consisting of 6B parameters. See Section 5.1 for discussions.

C.2 Varying LM sizes

We vary the size of the GPT-2 models—small, medium, large, and XL—with 124M, 355M, 774M, and 1.5B parameters, respectively. Results are reported in Table 11. We find that (1) for all models, increasing the model size generally helps, (2) for all model sizes, Channel MetaICL significantly outperforms other models, and (3) MetaICL enables a much smaller model to outperform a bigger model, e.g., Channel MetaICL based on GPT-2 Small outperforms the GPT-2 XL baseline (46.2 vs. 43.5).

C.3 Which meta-training tasks are more helpful?

Based on large variance across different choices of meta-training (Figure 2 of Section 5.2), we think certain tasks are more helpful for meta-training than other tasks. In this context, we create 50 sets of seven meta-training tasks using 50 different random seeds. We then measure the correlation between tasks/task pairs/task triples and average performance of Channel MetaICL when the task is included in the meta-training tasks.

Table 12 reports the result. We first find that high quality datasets with diverse domain like GLUE family (Wang et al., 2018) are often helpful. We also find that datasets that are collected adversarially (e.g. `paws`, `art`) or are notably dissimilar from all other tasks (e.g. `wikisql` that requires

Single task	
Helpful:	tweet_eval-offensive, glue-sst2, glue-mnli, wino_grande, kilt_hotpotqa
Unhelpful:	race-middle, cosmos_qa, dbpedia_14, gigaword, wikisql
Task pair	
Helpful:	(yelp_review_full, glue-mnli), (yelp_review_full, wino_grande), (hateexplain, glue-sst2), (hateexplain, glue-mnli), (hateexplain, glue-qqp),
Unhelpful:	(paws, dbpedia_14), (paws, art), (paws, cosmos_qa), (cosmos_qa, dbpedia_14), (quail, art)
Task triple	
Helpful	(yelp_review_full, glue-qqp, glue-mnli), (yelp_review_full, glue-sst2, glue-mnli), (yelp_review_full, hateexplain, glue-mnli), (yelp_review_full, hateexplain, qqp), (yelp_review_full, hate_speech_offensive, glue-mnli),
Unhelpful	(paws, dbpedia_14, art), (paws, dbpedia_14, cosmos_qa), (paws, cosmos_qa, art), (dbpedia_14, cosmos_qa, art), (quail, paws, dbpedia_14)

Table 12: Analysis of which meta-training tasks give good performance in Channel MetaICL. We report five most helpful and the most unhelpful tasks (or task sets), respectively.

semantic parsing) are often unhelpful. Nonetheless, we were not able to find good explanations for other cases, e.g., many sentiment analysis datasets being particularly helpful even though only 3 out of 26 target datasets are sentiment analysis, and `dbpedia_14`/`cosmos_qa`/`race-middle` being unhelpful. Moreover, we think which tasks are helpful largely depends on the choice of target tasks, and we should not make early conclusions that certain tasks are helpful/unhelpful in all cases. We think future work should investigate these impacts in a more systematic way.

D Potential Risks

MetaICL is based on the large language model that is pretrained on a web corpus, which potentially includes harmful and biased context, despite the original authors' best efforts to mine the text. There are also potential risks in privacy and security—for instance, Carlini et al. (2021) reported that it is possible to design the attack algorithm to extract a substantial amount of training data. We thus highlight that MetaICL should be considered as a research prototype rather than a deployable system to real users, and continuing efforts are needed to reduce potential risks of the model.

Setting: HR→LR Meta-train piqa, hate_speech_offensive, google_wellformed_query, social_i_qa, circa, quoref, glue-sst2, scitail, emo, cosmos_qa, freebase_qa, ag_news, art, paws, kilt_ay2, glue-qnli, quail, ade_corpus_v2-classification, sciq, hatexplain, emotion, glue-qqp, kilt_fever, kilt_nq, dbpedia_14, kilt_zsre, hellaswag, squad-with_context, hotpot_qa, glue-mnli, ropes, squad-no_context, kilt_hotpotqa, discovery, superglue-record, race-middle, race-high, lama-trex, swag, gigaword, amazon_polarity, biomrc, tab_fact, tweet_eval-emoji, tweet_eval-offensive, tweet_eval-sentiment, tweet_qa, imdb, lama-conceptnet, liar, anli, wiki_qa, kilt_trex, wikisql, wino_grande, wiqa, search_qa, xsum, yahoo_answers_topics, yelp_polarity, yelp_review_full
Setting: HR→LR Target quarel, **financial_phrasebank**, openbookqa, codah, qasc, glue-mrpc, dream, sick, commonsense_qa, **medical_questions_pairs**, quartz-with_knowledge, **poem_sentiment**, quartz-no_knowledge, glue-wnli, **climate_fever**, ethos-national_origin, ethos-race, ethos-religion, ai2_arc, hate_speech18, glue-rte, superglue-cb, superglue-copa, tweet_eval-hate, tweet_eval-stance_atheism, tweet_eval-stance_feminist
Setting: Classification Meta-train Meta-Train: superglue-rte, tweet_eval-sentiment, discovery, glue-rte, superglue-wsc, glue-mrpc, tweet_eval-stance_hillary, tweet_eval-offensive, emotion, hatexplain, glue-cola, sick, paws, ethos-sexual_orientation, glue-qqp, tweet_eval-emotion, sms_spam, health_fact, glue-mnli, imdb, ethos-disability, glue-wnli, scitail, trec-finegrained, yahoo_answers_topics, liar, glue-sst2, tweet_eval-stance_abortion, circa, tweet_eval-stance_climate, glue-qnli, tweet_eval-emoji, ethos-directed_vs_generalized, ade_corpus_v2-classification, hate_speech_offensive, superglue-wic, google_wellformed_query, tweet_eval-irony, ethos-gender, onestop_english, trec, rotten_tomatoes, kilt_fever
Setting: Non-Classification Meta-train ade_corpus_v2-dosage, art, biomrc, blimp-anaphor_number_agreement, blimp-ellipsis_n_bar_2, blimp-sentential_negation_npi_licensor_present, blimp-sentential_negation_npi_scope, commonsense_qa, crows_pairs, dream, freebase_qa, gigaword, hellaswag, hotpot_qa, kilt_ay2, kilt_hotpotqa, kilt_trex, kilt_zsre, lama-conceptnet, lama-google_re, lama-squad, numer_sense, openbookqa, piqa, proto_qa, qa_srl, quarel, quartz-no_knowledge, race-high, ropes, sciq, social_i_qa, spider, superglue-multirc, wikisql, xsum, yelp_review_full
Setting: Classification Target tweet_eval-stance_feminist, ethos-national_origin, tweet_eval-hate, ag_news, amazon_polarity, hate_speech18, **poem_sentiment**, **climate_fever**, **medical_questions_pairs**, tweet_eval-stance_atheism, superglue-cb, dbpedia_14, wiki_qa, emo, yelp_polarity, ethos-religion, **financial_phrasebank**, tab_fact, anli, ethos-race
Setting: QA Meta-train biomrc, boolq, freebase_qa, hotpot_qa, kilt_hotpotqa, kilt_nq, kilt_trex, kilt_zsre, lama-conceptnet, lama-google_re, lama-squad, lama-trex, mc_taco, numer_sense, quoref, ropes, search_qa, squad-no_context, squad-with_context, superglue-multirc, superglue-record, tweet_qa, web_questions, unifiedqa:squad2, unifiedqa:natural_questions_with_dpr_para, unifiedqa:race_string, unifiedqa:squad1_1, unifiedqa:drop, unifiedqa:newsqa, unifiedqa:narrativeqa, unifiedqa:winogrande_xl, unifiedqa:social_iqa, unifiedqa:quoref, unifiedqa:physical_iqa, unifiedqa:ropes, unifiedqa:commonsenseqa, unifiedqa:boolq
Setting: Non-QA Meta-train hate_speech_offensive, google_wellformed_query, circa, glue-sst2, scitail, emo, ag_news, art, paws, kilt_ay2, glue-qnli, ade_corpus_v2-classification, hatexplain, emotion, glue-qqp, kilt_fever, dbpedia_14, glue-mnli, discovery, gigaword, amazon_polarity, tab_fact, tweet_eval-emoji, tweet_eval-offensive, tweet_eval-sentiment, imdb, liar, anli, wikisql, xsum, yahoo_answers_topics, yelp_polarity, yelp_review_full
Setting: QA Target ai2_arc, codah, cosmos_qa, dream, hellaswag, openbookqa, qasc, quail, quarel, quartz-no_knowledge, quartz-with_knowledge, sciq, superglue-copa, swag, wino_grande, wiqa, unifiedqa:qasc, unifiedqa:qasc_with_ir, unifiedqa:openbookqa, unifiedqa:openbookqa_with_ir, **unifiedqa:mctest**, unifiedqa:ai2_science_middle
Setting: Non-NLI Meta-train ade_corpus_v2-classification, ag_news, amazon_polarity, circa, climate_fever, dbpedia_14, discovery, emo, emotion, ethos-directed_vs_generalized, ethos-disability, ethos-gender, ethos-national_origin, ethos-race, ethos-religion, ethos-sexual_orientation, financial_phrasebank, glue-cola, glue-mrpc, glue-qqp, glue-sst2, google_wellformed_query, hate_speech18, hate_speech_offensive, hatexplain, health_fact, imdb, kilt_fever, liar, medical_questions_pairs, onestop_english, paws, poem_sentiment, rotten_tomatoes, sick, sms_spam, superglue-wic, superglue-wsc, tab_fact, trec, trec-finegrained, tweet_eval-emoji, tweet_eval-emotion, tweet_eval-hate, tweet_eval-irony, tweet_eval-offensive, tweet_eval-sentiment, tweet_eval-stance_abortion, tweet_eval-stance_atheism, tweet_eval-stance_climate, tweet_eval-stance_feminist, tweet_eval-stance_hillary, wiki_qa, yahoo_answers_topics, yelp_polarity Setting: NLI Target anli, glue-mnli, glue-qnli, glue-rte, glue-wnli, **scitail**, sick, superglue-cb
Setting: Non-Paraphrase Meta-train ade_corpus_v2-classification, ag_news, amazon_polarity, anli, circa, climate_fever, dbpedia_14, discovery, emo, emotion, ethos-directed_vs_generalized, ethos-disability, ethos-gender, ethos-national_origin, ethos-race, ethos-religion, ethos-sexual_orientation, financial_phrasebank, glue-cola, glue-mnli, glue-qnli, glue-rte, glue-sst2, glue-wnli, google_wellformed_query, hate_speech18, hate_speech_offensive, hatexplain, health_fact, imdb, kilt_fever, liar, onestop_english, poem_sentiment, rotten_tomatoes, scitail, sick, sms_spam, superglue-cb, superglue-rte, superglue-wic, superglue-wsc, tab_fact, trec, trec-finegrained, tweet_eval-emoji, tweet_eval-emotion, tweet_eval-hate, tweet_eval-irony, tweet_eval-offensive, tweet_eval-sentiment, tweet_eval-stance_abortion, tweet_eval-stance_atheism, tweet_eval-stance_climate, tweet_eval-stance_feminist, tweet_eval-stance_hillary, wiki_qa, yahoo_answers_topics, yelp_polarity
Setting: Non-Paraphrase Target Target: glue-mrpc, glue-qqp, **medical_questions_pairs**, paws
Setting: HR→LR Diverse Meta-train glue-mnli, glue-qqp, glue-sst2, hate_speech_offensive, kilt_hotpotqa, kilt_zsre, lama-trex, race-high, scitail, tweet_eval-offensive, wino_grande, yahoo_answers_topics, yelp_review_full
Setting: HR→LR No Diverse Meta-train ag_news, amazon_polarity, dbpedia_14, emo, emotion, glue-sst2, imdb, tweet_eval-emoji, tweet_eval-offensive, tweet_eval-sentiment, yahoo_answers_topics, yelp_polarity, yelp_review_full
Setting: HR→LR Instructions Meta-train ag_news, amazon_polarity, anli, art, circa, cosmos_qa, dbpedia_14, discovery, emo, emotion, freebase_qa, gigaword, google_wellformed_query, hellaswag, imdb, liar, paws, piqa, quail, quoref, ropes, sciq, scitail, social_i_qa, swag, tab_fact, wiki_qa, wiqa, xsum, yahoo_answers_topics, yelp_polarity, yelp_review_full
Setting: HR→LR Instructions Target ai2_arc, **climate_fever**, codah, commonsense_qa, dream, **financial_phrasebank**, **medical_questions_pairs**, openbookqa, **poem_sentiment**, qasc, quarel, sick

Table 13: Full datasets for all settings. The first 10 rows are for main settings described in Section 4.1; the last four rows are settings used for ablations in Section 5.2. Splits and dataname names consistent to those in Ye et al. (2021) and Khashabi et al. (2020). **Bold** indicates the test dataset with no overlap in domain with meta-training tasks. A prefix `unifiedqa:` indicates that the dataset taken is from UNIFIEDQA; otherwise, from CROSSFIT. References for all datasets are provided in Table 14.

ade_corpus_v2-classification (Gurulingappa et al., 2012), ade_corpus_v2-dosage (Gurulingappa et al., 2012), ag_news Gulli (link), ai2_arc (Clark et al., 2018), amazon_polarity (McAuley and Leskovec, 2013), anli (Nie et al., 2020), art (Bhagavatula et al., 2020), biomrc (Pappas et al., 2020), blimp-anaphor_number_agreement (Warstadt et al., 2020), blimp-ellipsis_n_bar_2 (Warstadt et al., 2020), blimp-sentential_negation_npi_licensor_present (Warstadt et al., 2020), blimp-sentential_negation_npi_scope (Warstadt et al., 2020), boolq (Clark et al., 2019), circa (Louis et al., 2020), climate_fever (Diggelmann et al., 2020), codah (Chen et al., 2019), commonsense_qa (Talmor et al., 2019), cosmos_qa (Huang et al., 2019), crows_pairs (Nangia et al., 2020), dbpedia_14 (Lehmann et al., 2015), discovery (Sileo et al., 2019), dream (Sun et al., 2019), emo (Chatterjee et al., 2019), emotion (Saravia et al., 2018), ethos-directed_vs_generalized (Mollas et al., 2020), ethos-disability (Mollas et al., 2020), ethos-gender (Mollas et al., 2020), ethos-national_origin (Mollas et al., 2020), ethos-race (Mollas et al., 2020), ethos-religion (Mollas et al., 2020), ethos-sexual_orientation (Mollas et al., 2020), financial_phrasebank (Malo et al., 2014), freebase_qa (Jiang et al., 2019), gigaword (Napoles et al., 2012), glue-cola (Warstadt et al., 2019), glue-mnli (Williams et al., 2018), glue-mrpc (Dolan and Brockett, 2005), glue-qnli (Rajpurkar et al., 2016), glue-qqp (data.quora.com/First-Quora-Dataset-Release-Question-Pairs), glue-rte (Dagan et al., 2005; Bar-Haim et al., 2006)(Giampiccolo et al., 2007; Bentivogli et al., 2009), glue-sst2 (Socher et al., 2013), glue-wnli (Levesque et al., 2012), google_wellformed_query (Faruqui and Das, 2018), hate_speech18 (de Gibert et al., 2018), hate_speech_offensive (Davidson et al., 2017), hatexplain (Mathew et al., 2020), health_fact (Kotonya and Toni, 2020), hellaswag (Zellers et al., 2019), hotpot_qa (Yang et al., 2018), imdb (Maas et al., 2011), kilt_ay2 (Hoffart et al., 2011), kilt_fever (Thorne et al., 2018), kilt_hotpotqa (Yang et al., 2018), kilt_nq (Kwiatkowski et al., 2019), kilt_trex (Elsahar et al., 2018), kilt_zsre (Levy et al., 2017), lama-conceptnet (Petroni et al., 2019, 2020), lama-google_re (Petroni et al., 2019, 2020), lama-squad (Petroni et al., 2019, 2020), lama-trex (Petroni et al., 2019, 2020), liar (Wang, 2017), mc_taco (Zhou et al., 2019), medical_questions_pairs (McCreery et al., 2020), numer_sense (Lin et al., 2020), onestop_english (Vajjala and Lučić, 2018), openbookqa (Mihaylov et al., 2018), paws (Zhang et al., 2019), piqa (Bisk et al., 2020), poem_sentiment (Sheng and Uthus, 2020), proto_qa (Boratko et al., 2020), qa_srl (He et al., 2015), qasc (Khot et al., 2020), quail (Rogers et al., 2020), quarel (Tafjord et al., 2019a), quartz-no_knowledge (Tafjord et al., 2019b), quartz-with_knowledge (Tafjord et al., 2019b), quoref (Dasigi et al., 2019), race-high (Lai et al., 2017), race-middle (Lai et al., 2017), ropes (Lin et al., 2019), rotten_tomatoes (Pang and Lee, 2005), sciq (Welbl et al., 2017), scitail (Khot et al., 2018), search_qa (Dunn et al., 2017), sick (Marelli et al., 2014), sms_spam (Almeida et al., 2011), social_i_qa (Sap et al., 2019a), spider (Yu et al., 2018), squad-no_context (Rajpurkar et al., 2016), squad-with_context (Rajpurkar et al., 2016), superglue-cb (de Marneffe et al., 2019), superglue-copa (Gordon et al., 2012), superglue-multirc (Khashabi et al., 2018), superglue-record (Zhang et al., 2018), superglue-rte (Dagan et al., 2005; Bar-Haim et al., 2006)(Giampiccolo et al., 2007; Bentivogli et al., 2009), superglue-wic (Pilehvar and Camacho-Collados, 2019), superglue-wsc (Levesque et al., 2012), swag (Zellers et al., 2018), tab_fact (Chen et al., 2020), trec (Li and Roth, 2002; Hovy et al., 2001), trec-finegrained (Li and Roth, 2002; Hovy et al., 2001), tweet_eval-emoji (Barbieri et al., 2020), tweet_eval-emotion (Barbieri et al., 2020), tweet_eval-hate (Barbieri et al., 2020), tweet_eval-irony (Barbieri et al., 2020), tweet_eval-offensive (Barbieri et al., 2020), tweet_eval-sentiment (Barbieri et al., 2020), tweet_eval-stance_abortion (Barbieri et al., 2020), tweet_eval-stance_atheism (Barbieri et al., 2020), tweet_eval-stance_climate (Barbieri et al., 2020), tweet_eval-stance_feminist (Barbieri et al., 2020), tweet_eval-stance_hillary (Barbieri et al., 2020), tweet_qa (Xiong et al., 2019), unifiedqa:ai2_science_middle (data.allenai.org/ai2-science-questions), unifiedqa:boolq (Clark et al., 2019), unifiedqa:commonsenseqa (Talmor et al., 2019), unifiedqa:drop (Dua et al., 2019), unifiedqa:mctest (Richardson et al., 2013), unifiedqa:narrativeqa (Kociský et al., 2018), unifiedqa:natural_questions (Kwiatkowski et al., 2019), unifiedqa:newsqa (Trischler et al., 2017), unifiedqa:openbookqa (Mihaylov et al., 2018), unifiedqa:physical_iqa (Bisk et al., 2020), unifiedqa:qasc (Khot et al., 2019), unifiedqa:quoref (Dasigi et al., 2019), unifiedqa:race_string (Lai et al., 2017), unifiedqa:ropes (Lin et al., 2019), unifiedqa:social_iqa (Sap et al., 2019b), unifiedqa:squad1_1 (Rajpurkar et al., 2016), unifiedqa:squad2 (Rajpurkar et al., 2018), unifiedqa:winogrande_xl (Sakaguchi et al., 2020a), web_questions (Berant et al., 2013), wiki_qa (Yang et al., 2015), wikisql (Zhong et al., 2017), wino_grande (Sakaguchi et al., 2020b), wiqa (Tandon et al., 2019), xsum (Narayan et al., 2018), yahoo_answers_topics (link), yelp_polarity (Zhang et al., 2015), yelp_review_full (Zhang et al., 2015)

Table 14: References for 142 datasets used in the paper. A prefix `unifiedqa:` indicates that the dataset taken is from UNIFIEDQA; otherwise, from CROSSFIT.