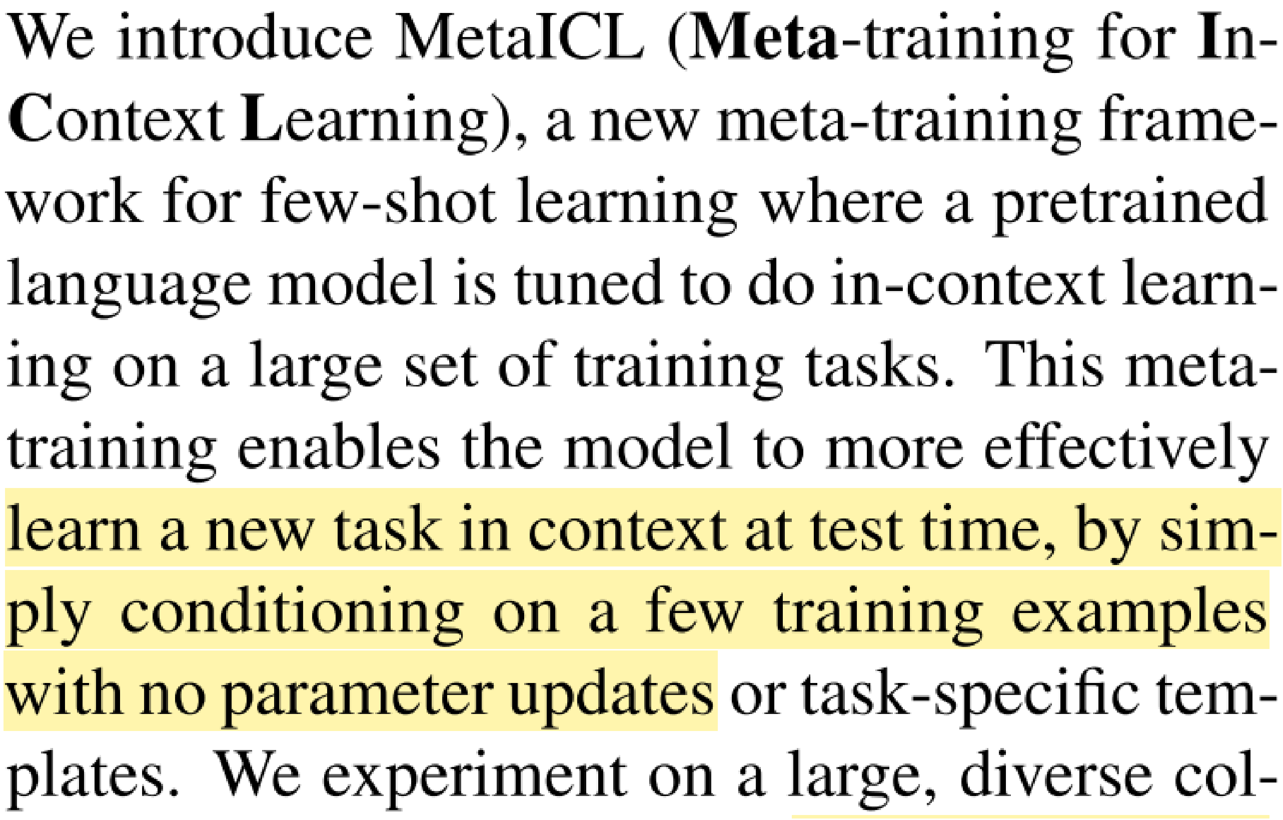
MetaICL: Learning to Learn In Context

Sewon Min[1,2] Mike Lewis[2] Luke Zettlemoyer[1,2] Hannaneh Hajishirzi[1,3]
[1]University of Washington [2]Facebook AI Research [3]Allen Institute for AI
{sewon,lsz,hannaneh}@cs.washington.edu mikelewis@fb.com

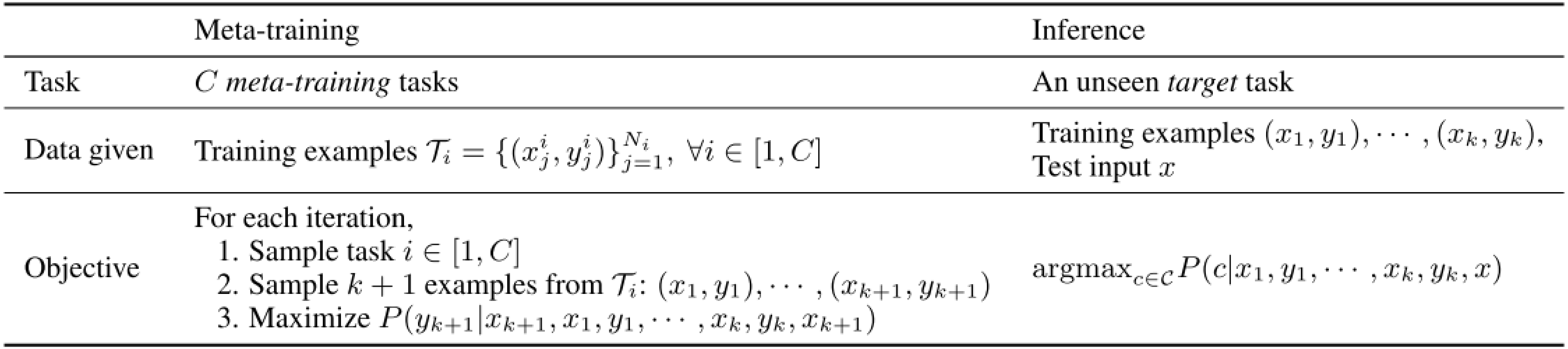
Presented by: Itay Levy

Overview

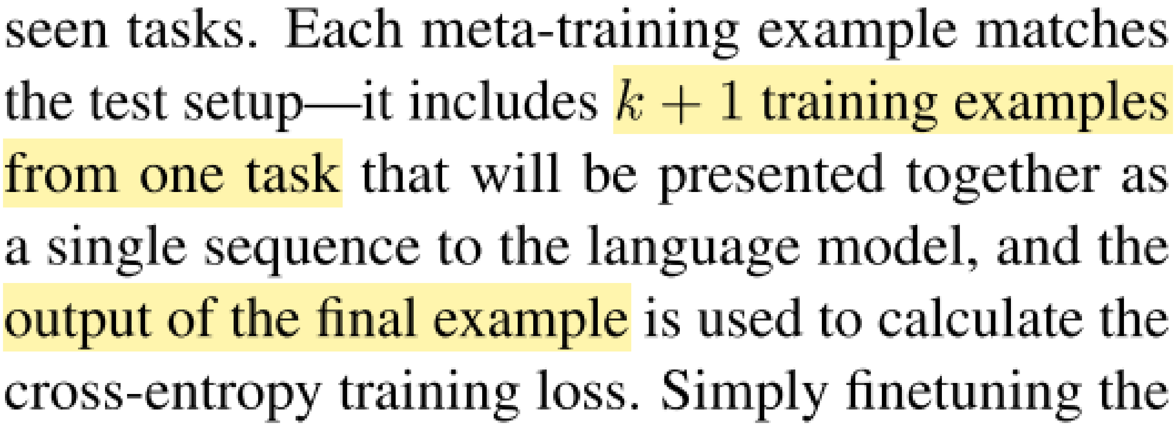


We introduce MetaICL (**Meta**-training for **In**-**C**ontext **L**earning), a new meta-training framework for few-shot learning where a pretrained language model is tuned to do in-context learning on a large set of training tasks. This meta-training enables the model to more effectively learn a new 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 col-

Overview

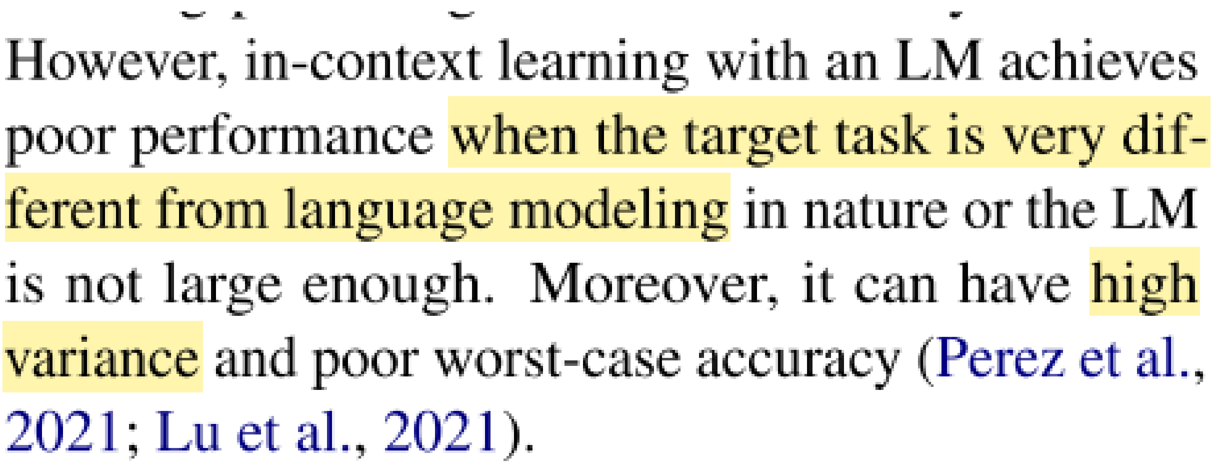


	Meta-training	Inference
Task	C *meta-training* tasks	An unseen *target* task
Data given	Training examples $\mathcal{T}_i = \{(x^i_j, y^i_j)\}_{j=1}^{N_i}, \forall i \in [1, C]$	Training examples $(x_1, y_1), \cdots, (x_k, y_k)$, Test input x
Objective	For each iteration, 1. Sample task $i \in [1, C]$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



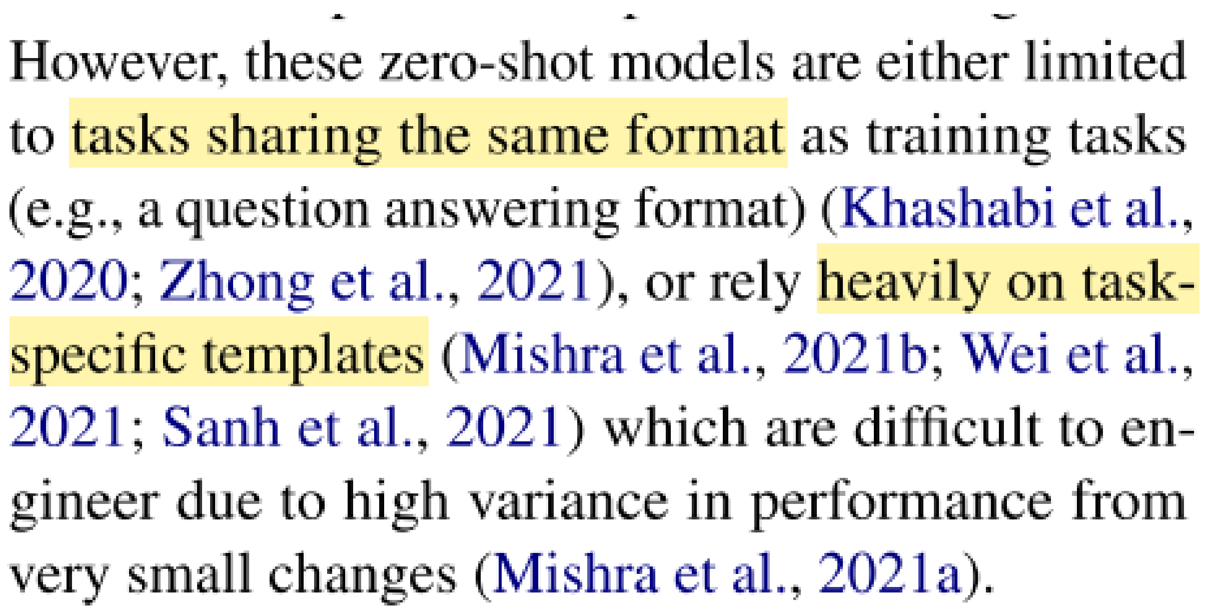
seen tasks. Each meta-training example matches the test setup—it includes $k+1$ training examples from one task that will be presented together as a single sequence to the language model, and the output of the final example is used to calculate the cross-entropy training loss. Simply finetuning the

Background - In context learning



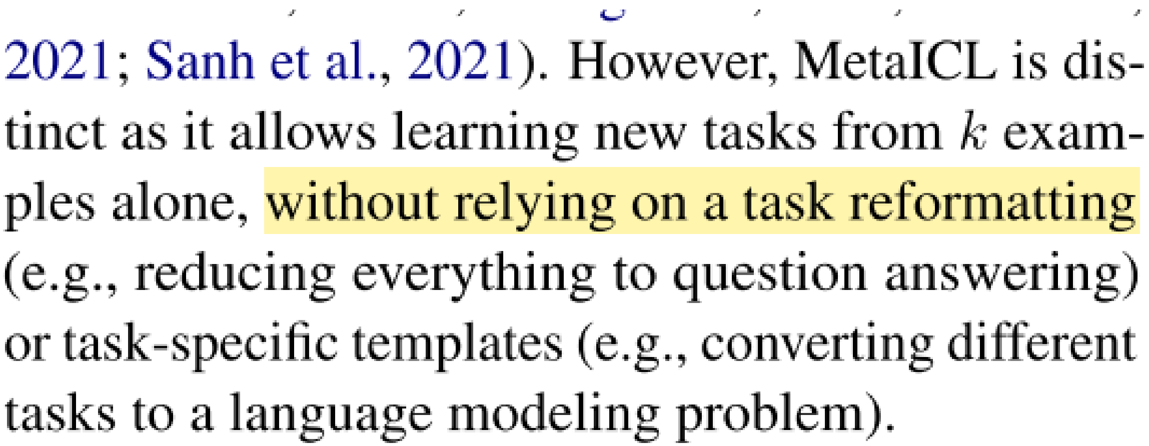
However, in-context learning with an LM achieves poor performance when the target task is very different from language modeling in nature or the LM is not large enough. Moreover, it can have high variance and poor worst-case accuracy (Perez et al., 2021; Lu et al., 2021).

Background – Multi task learning

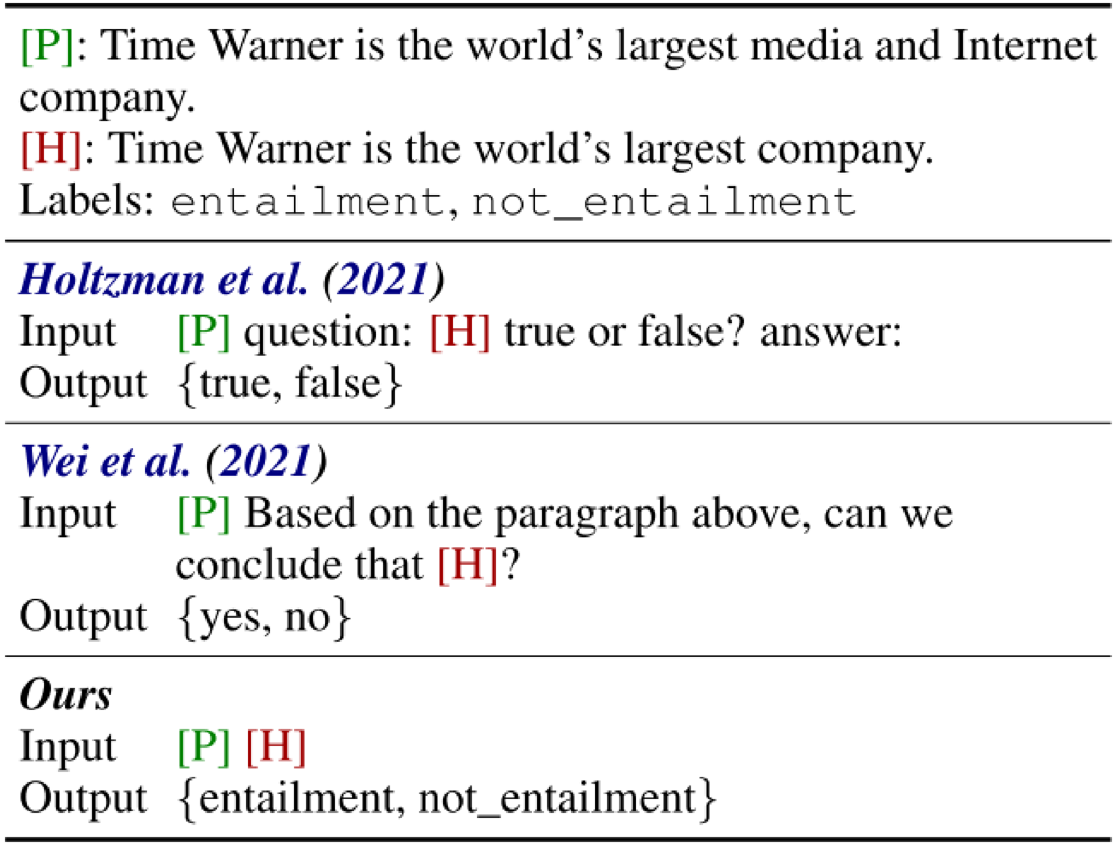


However, these zero-shot models are either limited to tasks sharing the same format as training tasks (e.g., a question answering format) (Khashabi et al., 2020; Zhong et al., 2021), or rely heavily on task-specific templates (Mishra et al., 2021b; Wei et al., 2021; Sanh et al., 2021) which are difficult to engineer due to high variance in performance from very small changes (Mishra et al., 2021a).

What’s unique about MetaICL? No templates



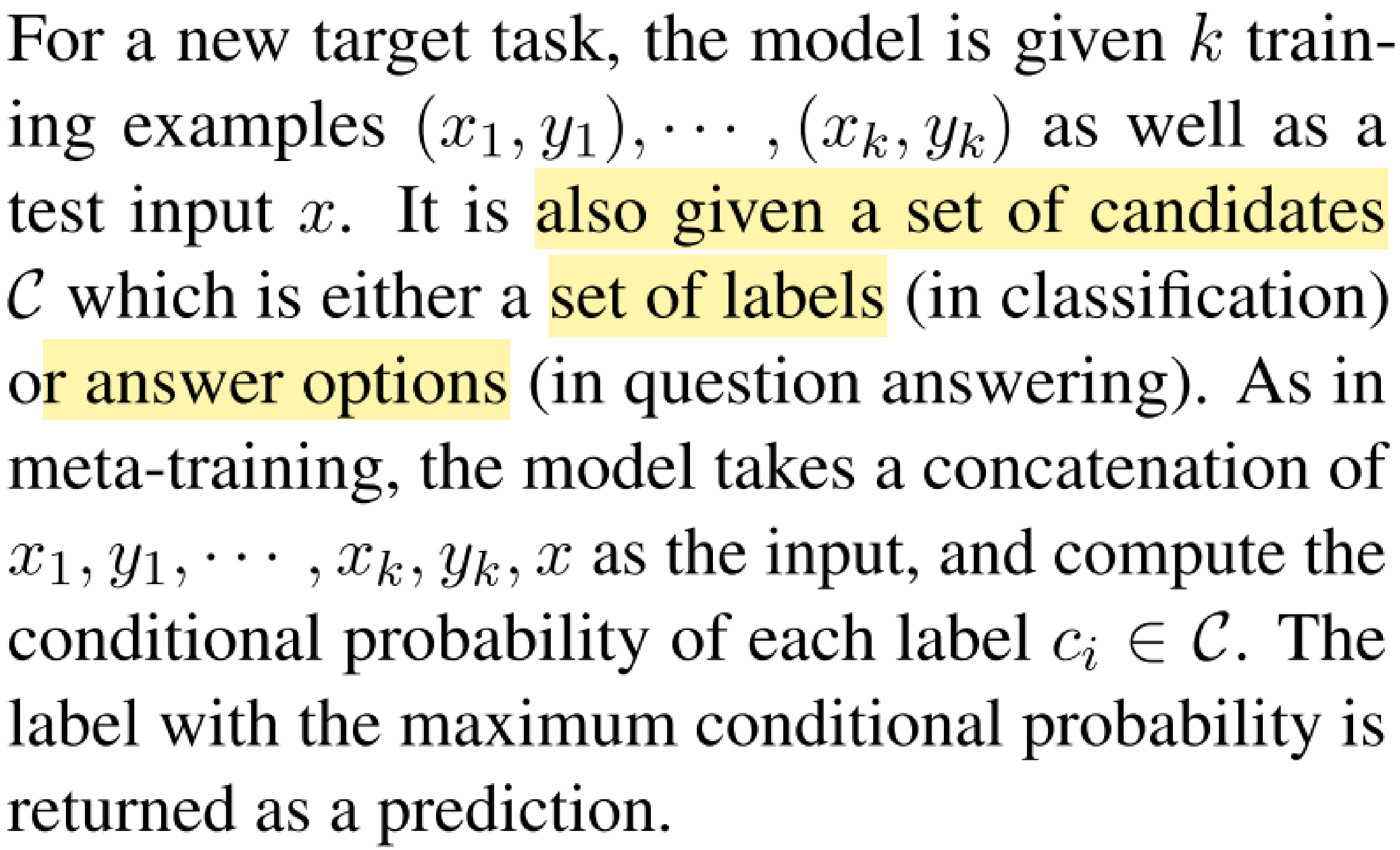
2021; Sanh et al., 2021). However, MetaICL is distinct as it allows learning new tasks from k examples alone, without relying on a task reformatting (e.g., reducing everything to question answering) or task-specific templates (e.g., converting different tasks to a language modeling problem).



[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.

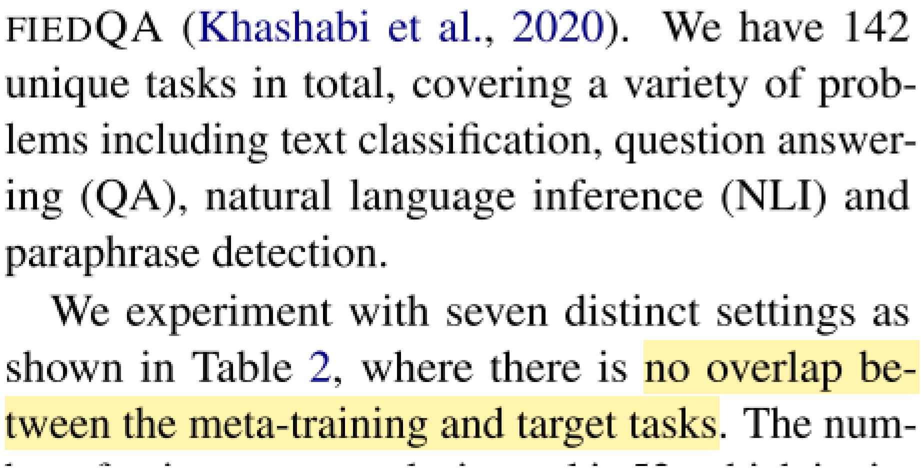
Inference



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.

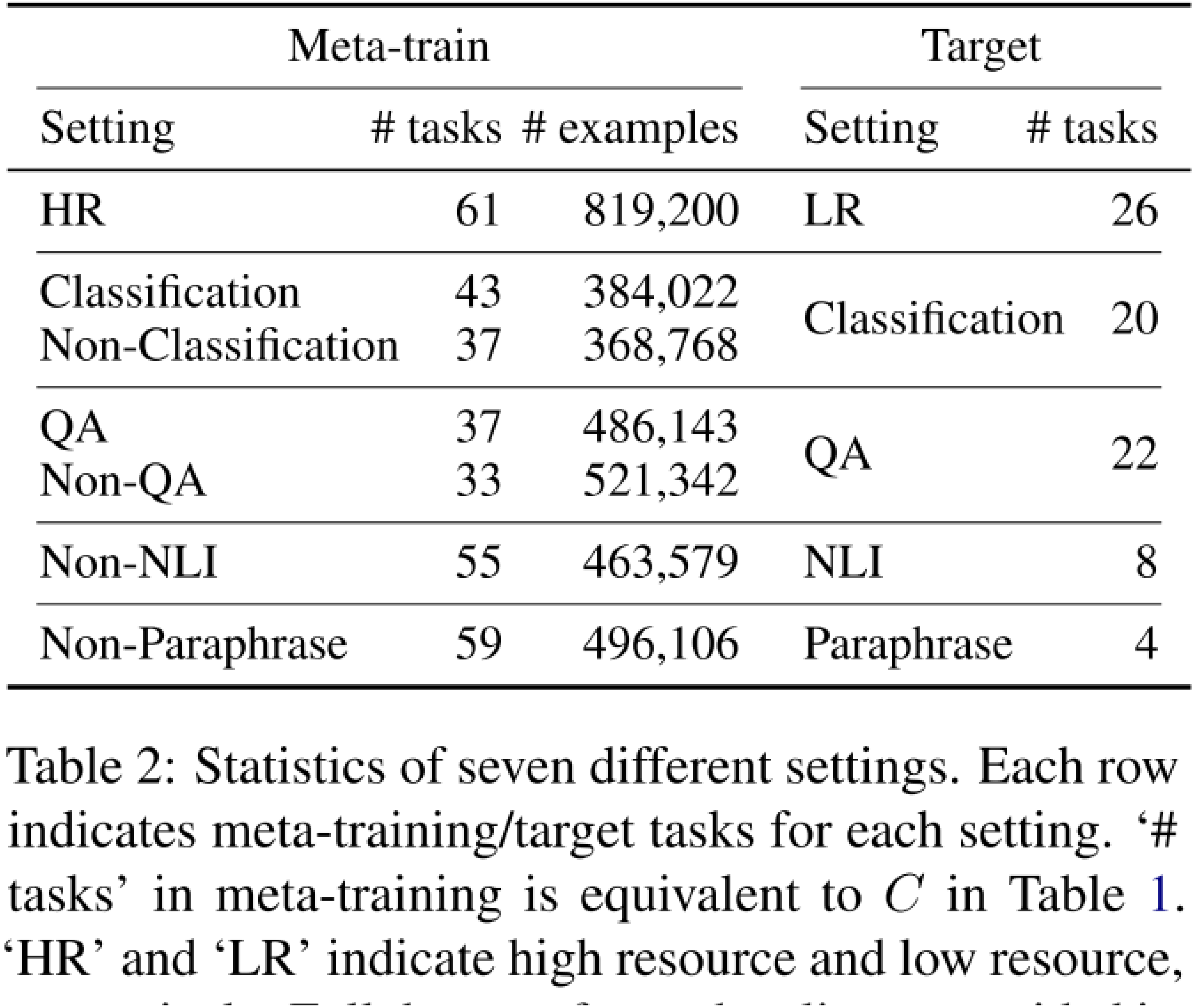
Experimental Setup

Datasets



FIEDQA (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.

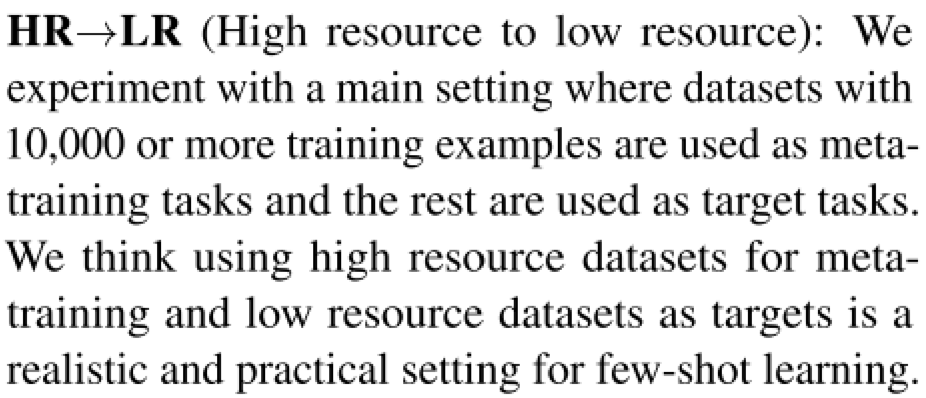
We experiment with seven distinct settings as shown in Table 2, where there is no overlap between the meta-training and target tasks. The num-



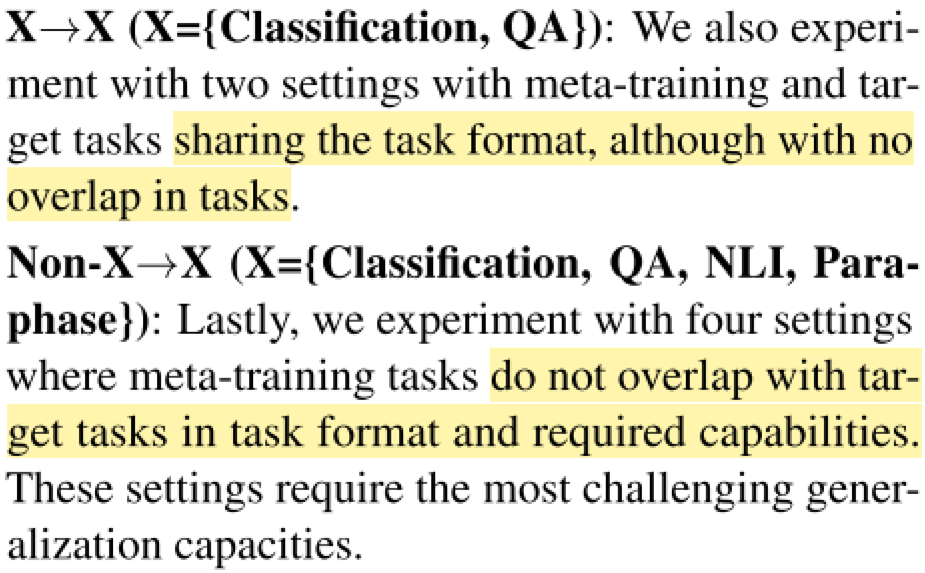
Meta-train			Target	
Setting	# tasks	# examples	Setting	# tasks
HR	61	819,200	LR	26
Classification	43	384,022	Classification	20
Non-Classification	37	368,768		
QA	37	486,143	QA	22
Non-QA	33	521,342		
Non-NLI	55	463,579	NLI	8
Non-Paraphrase	59	496,106	Paraphrase	4

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,

Settings



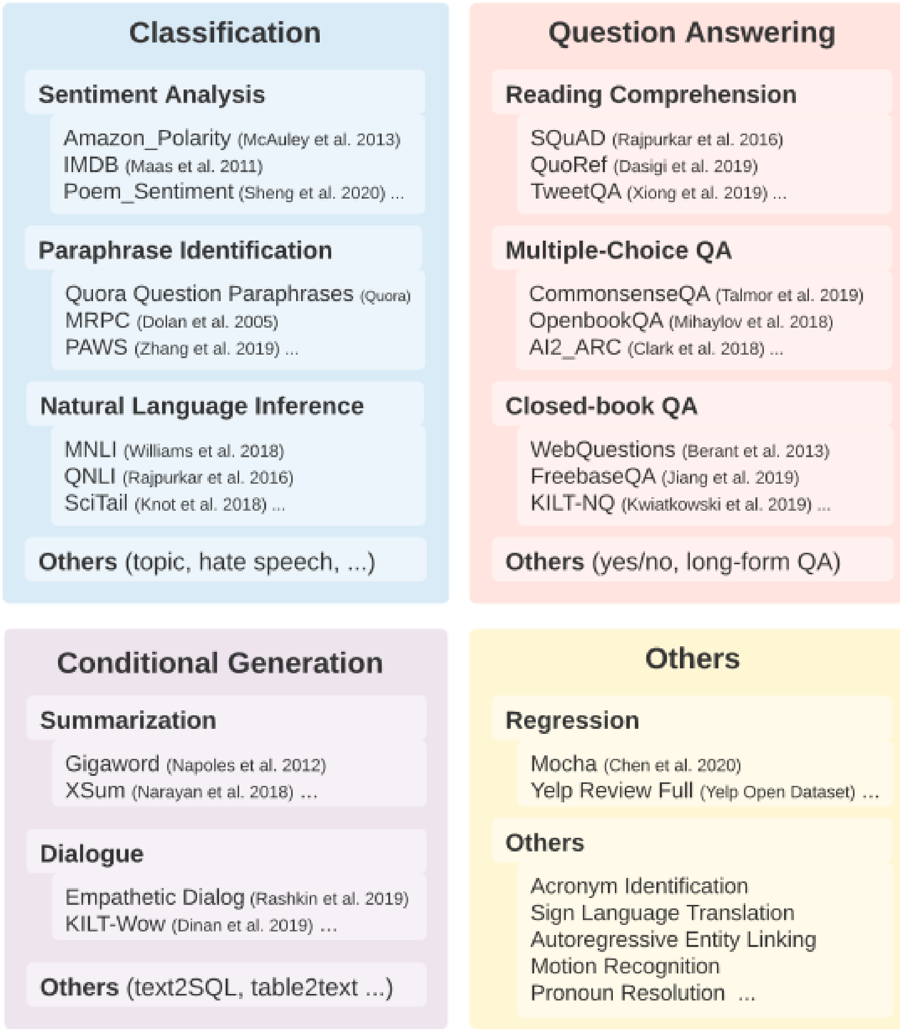
HR→LR (High resource to low resource): We experiment with a main setting where datasets with 10,000 or more training examples are used as meta-training tasks and the rest are used as target tasks. We think using high resource datasets for meta-training and low resource datasets as targets is a realistic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



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.

Datasets



Classification

Sentiment Analysis
- Amazon_Polarity (McAuley et al. 2013)
- IMDB (Maas et al. 2011)
- Poem_Sentiment (Sheng et al. 2020) ...

Paraphrase Identification
- Quora Question Paraphrases (Quora)
- MRPC (Dolan et al. 2005)
- PAWS (Zhang et al. 2019) ...

Natural Language Inference
- MNLI (Williams et al. 2018)
- QNLI (Rajpurkar et al. 2016)
- SciTail (Knot et al. 2018) ...

Others (topic, hate speech, ...)

Question Answering

Reading Comprehension
- SQuAD (Rajpurkar et al. 2016)
- QuoRef (Dasigi et al. 2019)
- TweetQA (Xiong et al. 2019) ...

Multiple-Choice QA
- CommonsenseQA (Talmor et al. 2019)
- OpenbookQA (Mihaylov et al. 2018)
- AI2_ARC (Clark et al. 2018) ...

Closed-book QA
- WebQuestions (Berant et al. 2013)
- FreebaseQA (Jiang et al. 2019)
- KILT-NQ (Kwiatkowski et al. 2019) ...

Others (yes/no, long-form QA)

Conditional Generation

Summarization
- Gigaword (Napoles et al. 2012)
- XSum (Narayan et al. 2018) ...

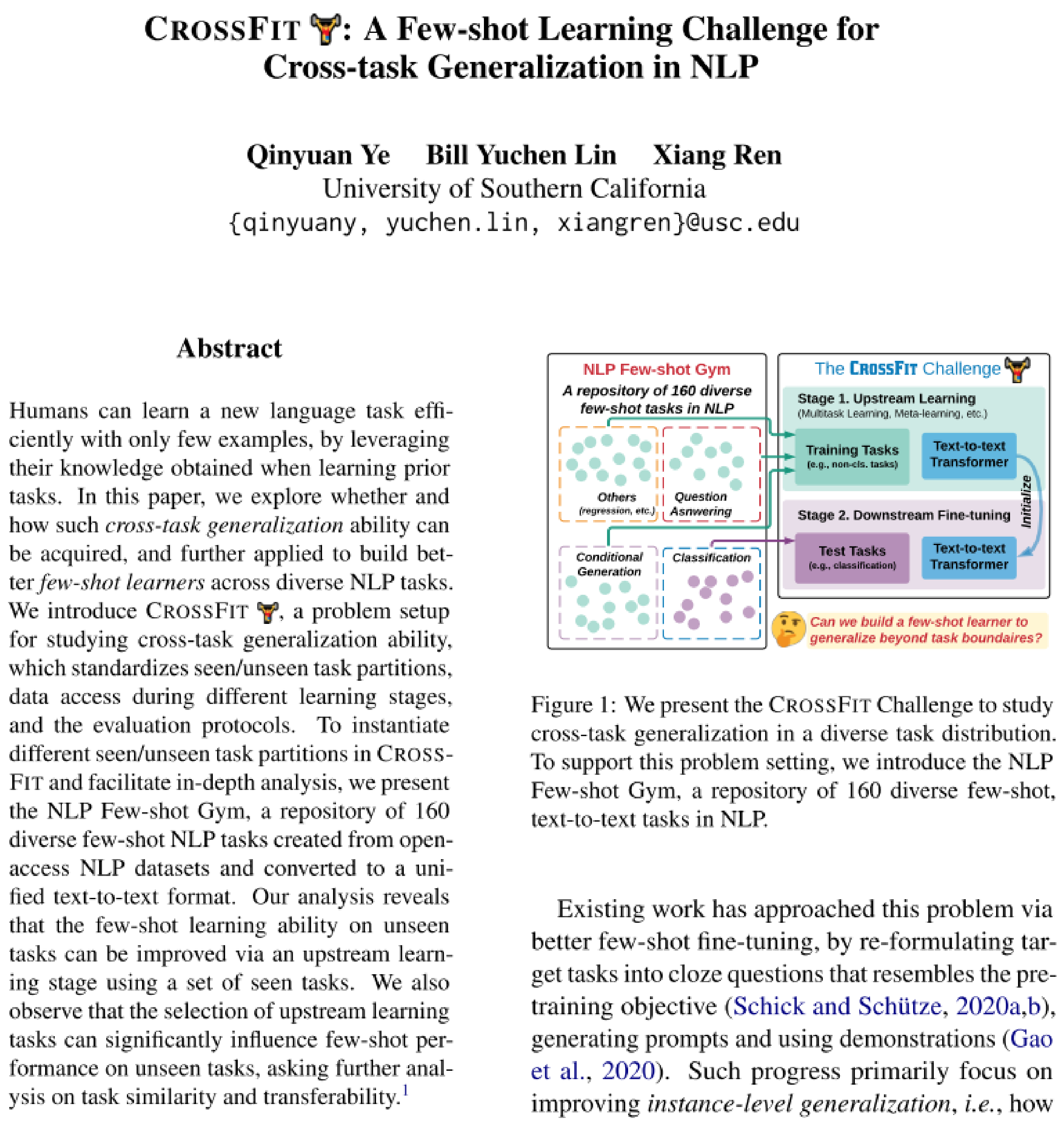
Dialogue
- Empathetic Dialog (Rashkin et al. 2019)
- KILT-Wow (Dinan et al. 2019) ...

Others (text2SQL, table2text ...)

Others

Regression
- Mocha (Chen et al. 2020)
- Yelp Review Full (Yelp Open Dataset) ...

Others
- Acronym Identification
- Sign Language Translation
- Autoregressive Entity Linking
- Motion Recognition
- Pronoun Resolution ...



CrossFit: A Few-shot Learning Challenge for Cross-task Generalization in NLP

Qinyuan Ye **Bill Yuchen Lin** **Xiang Ren**
University of Southern California
{qinyuany, yuchen.lin, xiangren}@usc.edu

Abstract

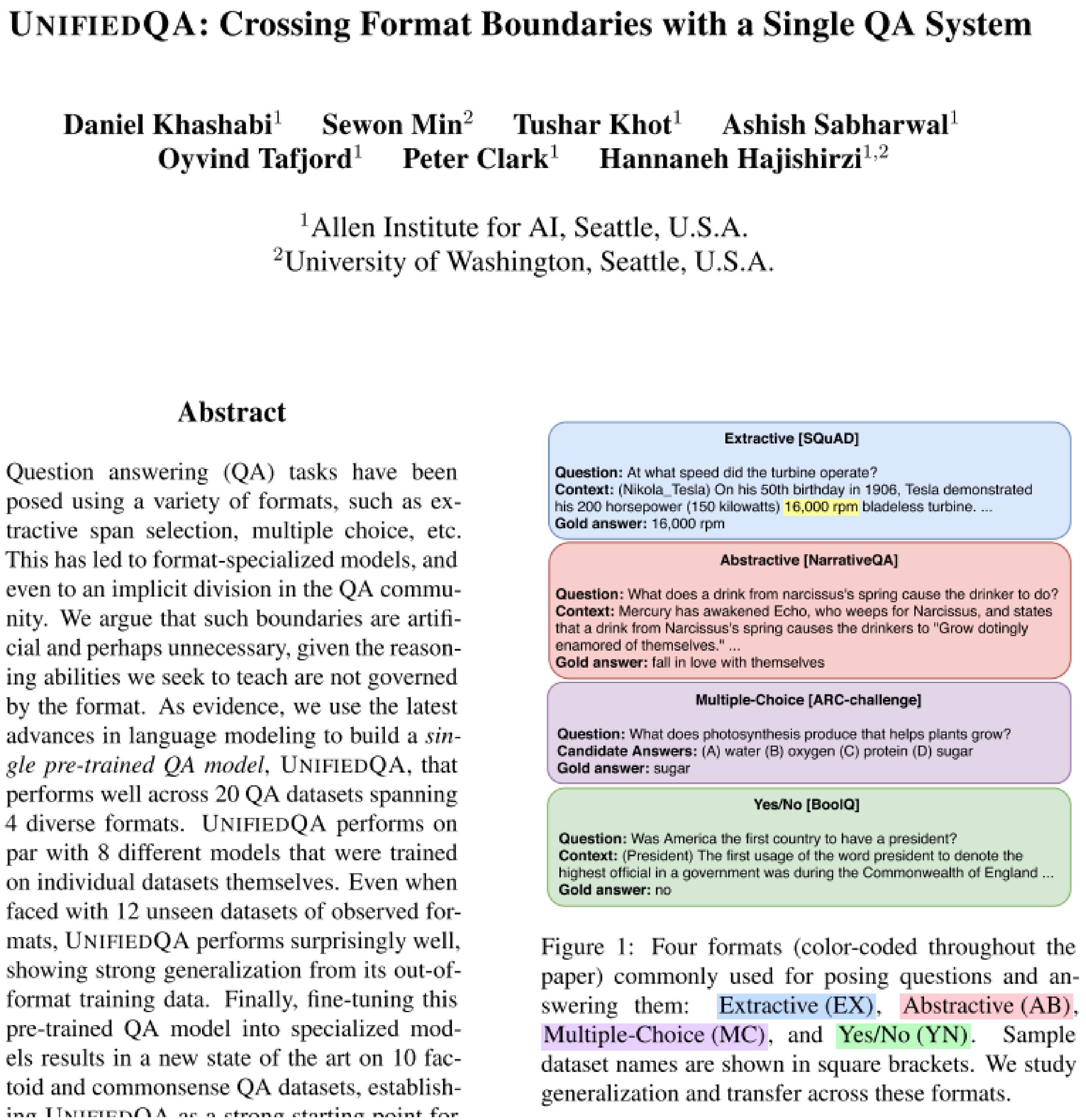
Humans can learn a new language task efficiently with only few examples, by leveraging their knowledge obtained when learning prior tasks. In this paper, we explore whether and how such *cross-task generalization* ability can be acquired, and further applied to build better *few-shot learners* across diverse NLP tasks. We introduce CrossFit, a problem setup for studying cross-task generalization ability, which standardizes seen/unseen task partitions, data access during different learning stages, and the evaluation protocols. To instantiate different seen/unseen task partitions in CrossFit and facilitate in-depth analysis, we present the NLP Few-shot Gym, a repository of 160 diverse few-shot NLP tasks created from open-access NLP datasets and converted to a unified text-to-text format. Our analysis reveals that the few-shot learning ability on unseen tasks can be improved via an upstream learning stage using a set of seen tasks. We also observe that the selection of upstream learning tasks can significantly influence few-shot performance on unseen tasks, asking further analysis on task similarity and transferability.[1]

NLP Few-shot Gym
A repository of 160 diverse few-shot tasks in NLP
Others (regression, etc.)
Question Asnwering
Conditional Generation
Classification
The CrossFit Challenge
Stage 1. Upstream Learning
(Multitask Learning, Meta-learning, etc.)
Training Tasks (e.g., non-cls. tasks)
Text-to-text Transformer
Initialize
Stage 2. Downstream Fine-tuning
Test Tasks (e.g., classification)
Text-to-text Transformer
Can we build a few-shot learner to generalize beyond task boundaires?

Figure 1: We present the CrossFit Challenge to study cross-task generalization in a diverse task distribution. To support this problem setting, we introduce the NLP Few-shot Gym, a repository of 160 diverse few-shot, text-to-text tasks in NLP.

Existing work has approached this problem via better few-shot fine-tuning, by re-formulating target tasks into cloze questions that resembles the pre-training objective (Schick and Schütze, 2020a,b), generating prompts and using demonstrations (Gao et al., 2020). Such progress primarily focus on improving *instance-level generalization*, *i.e.*, how

Datasets



UnifiedQA: Crossing Format Boundaries with a Single QA System

Daniel Khashabi[1] Sewon Min[2] Tushar Khot[1] Ashish Sabharwal[1]
Oyvind Tafjord[1] Peter Clark[1] Hannaneh Hajishirzi[1,2]

[1]Allen Institute for AI, Seattle, U.S.A.
[2]University of Washington, Seattle, U.S.A.

Abstract

Question answering (QA) tasks have been posed using a variety of formats, such as extractive span selection, multiple choice, etc. This has led to format-specialized models, and even to an implicit division in the QA community. We argue that such boundaries are artificial and perhaps unnecessary, given the reasoning abilities we seek to teach are not governed by the format. As evidence, we use the latest advances in language modeling to build a *single pre-trained QA model*, UnifiedQA, that performs well across 20 QA datasets spanning 4 diverse formats. UnifiedQA performs on par with 8 different models that were trained on individual datasets themselves. Even when faced with 12 unseen datasets of observed formats, UnifiedQA performs surprisingly well, showing strong generalization from its out-of-format training data. Finally, fine-tuning this pre-trained QA model into specialized models results in a new state of the art on 10 factoid and commonsense QA datasets, establishing UnifiedQA as a strong starting point for

Extractive [SQuAD]

Question: At what speed did the turbine operate?
Context: (Nikola_Tesla) On his 50th birthday in 1906, Tesla demonstrated his 200 horsepower (150 kilowatts) 16,000 rpm bladeless turbine. ...
Gold answer: 16,000 rpm

Abstractive [NarrativeQA]

Question: What does a drink from narcissus's spring cause the drinker to do?
Context: Mercury has awakened Echo, who weeps for Narcissus, and states that a drink from Narcissus's spring causes the drinkers to "Grow dotingly enamored of themselves." ...
Gold answer: fall in love with themselves

Multiple-Choice [ARC-challenge]

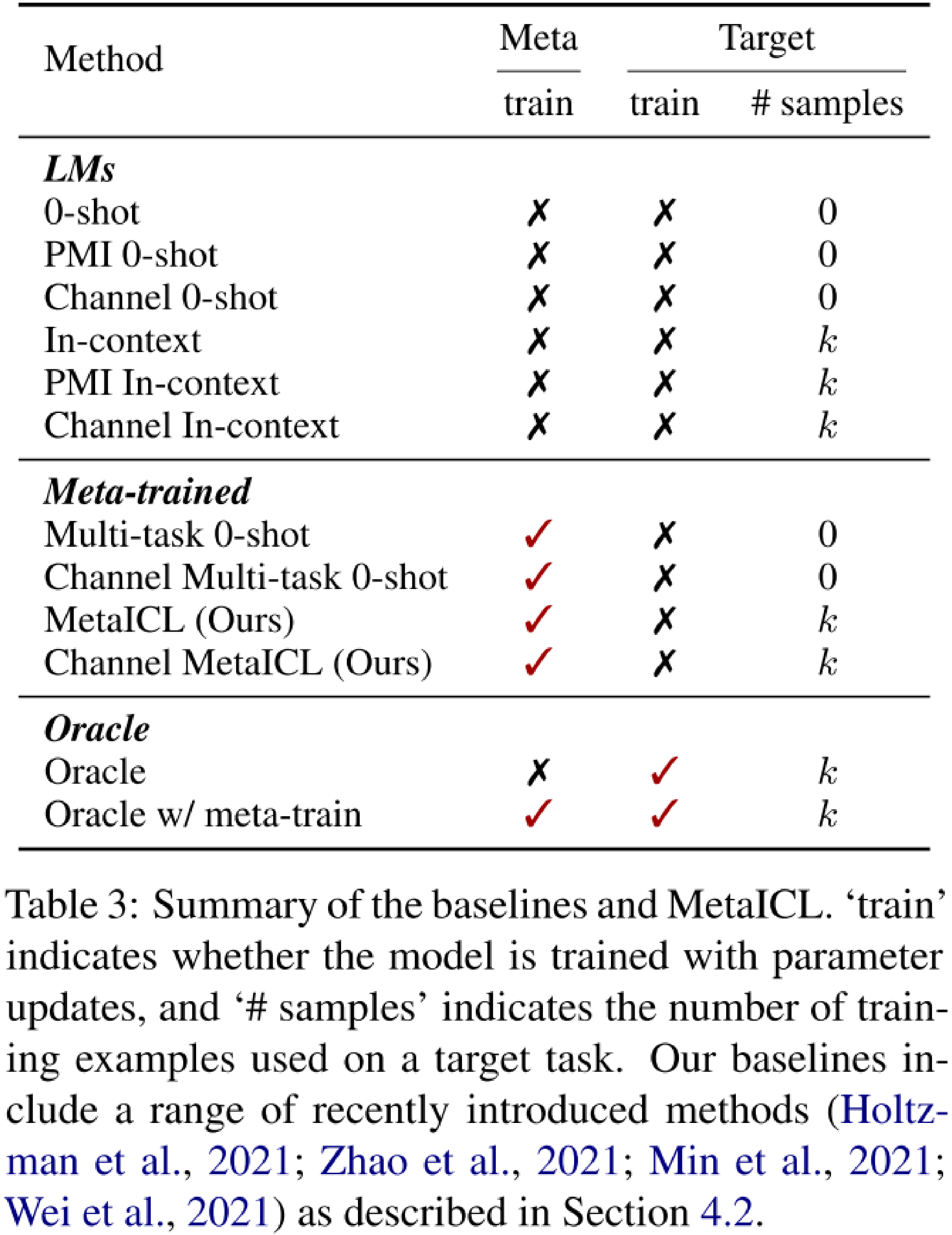
Question: What does photosynthesis produce that helps plants grow?
Candidate Answers: (A) water (B) oxygen (C) protein (D) sugar
Gold answer: sugar

Yes/No [BoolQ]

Question: Was America the first country to have a president?
Context: (President) The first usage of the word president to denote the highest official in a government was during the Commonwealth of England ...
Gold answer: no

Figure 1: Four formats (color-coded throughout the paper) commonly used for posing questions and answering them: Extractive (EX), Abstractive (AB), Multiple-Choice (MC), and Yes/No (YN). Sample dataset names are shown in square brackets. We study generalization and transfer across these formats.

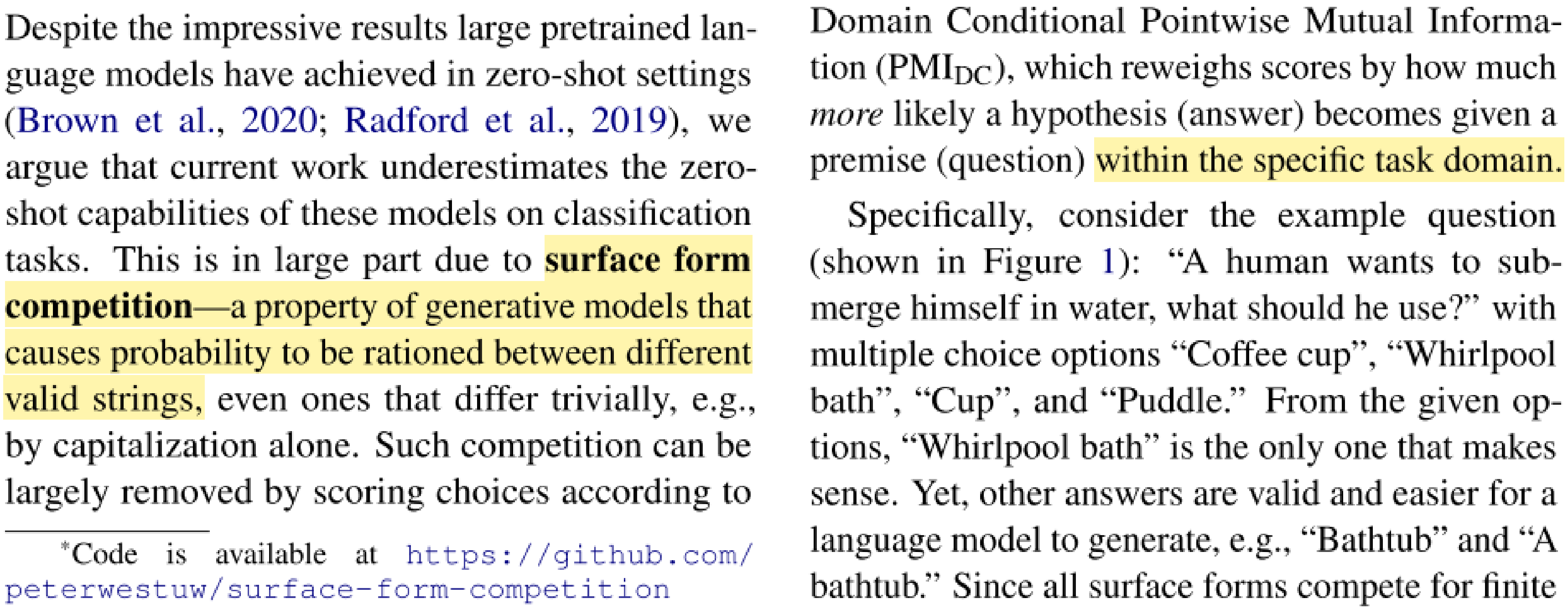
Baselines



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.

Baselines - PMI

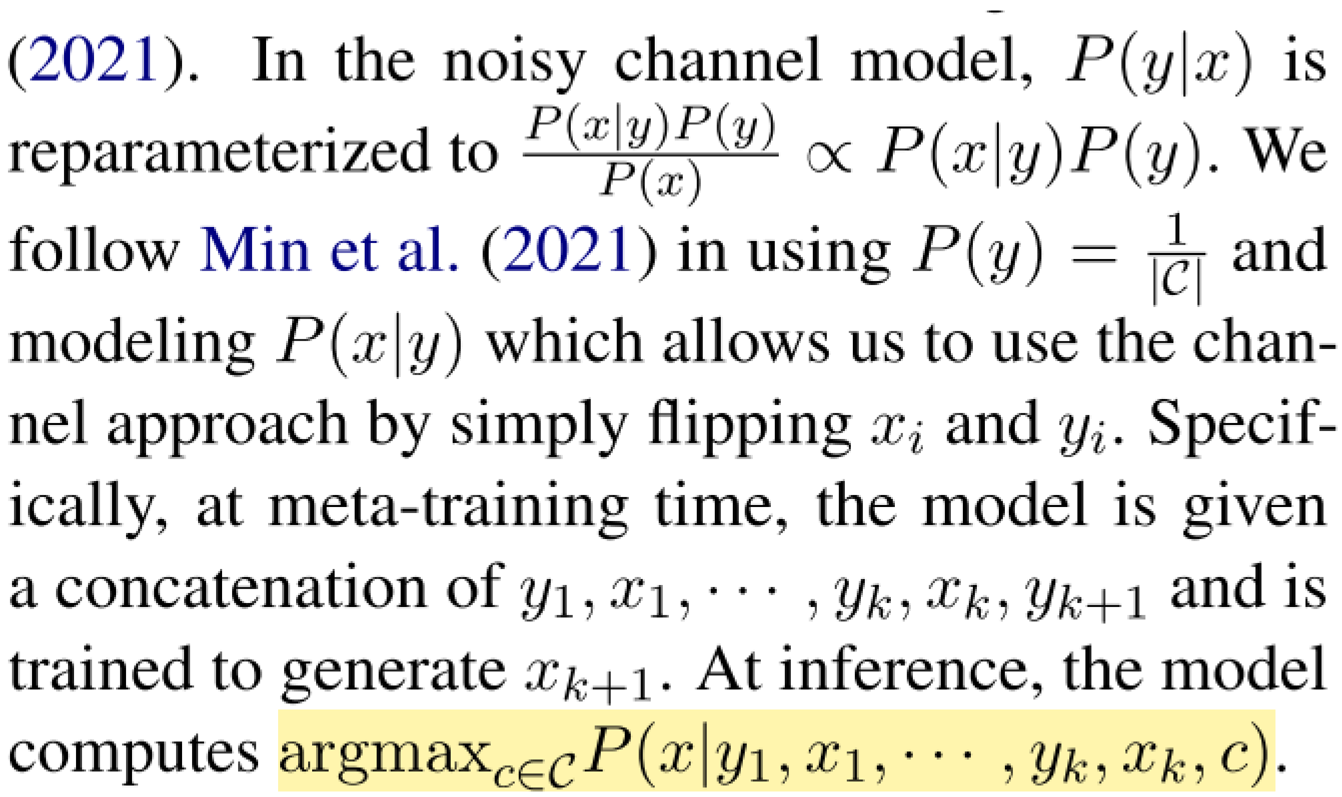


Despite the impressive results large pretrained language models have achieved in zero-shot settings (Brown et al., 2020; Radford et al., 2019), we argue that current work underestimates the zero-shot capabilities of these models on classification tasks. This is in large part due to **surface form competition**—a property of generative models that causes probability to be rationed between different valid strings, even ones that differ trivially, e.g., by capitalization alone. Such competition can be largely removed by scoring choices according to Domain Conditional Pointwise Mutual Information (PMI_{DC}), which reweighs scores by how much *more* likely a hypothesis (answer) becomes given a premise (question) within the specific task domain.

Specifically, consider the example question (shown in Figure 1): "A human wants to submerge himself in water, what should he use?" with multiple choice options "Coffee cup", "Whirlpool bath", "Cup", and "Puddle." From the given options, "Whirlpool bath" is the only one that makes sense. Yet, other answers are valid and easier for a language model to generate, e.g., "Bathtub" and "A bathtub." Since all surface forms compete for finite

*Code is available at https://github.com/peterwestuw/surface-form-competition

Baselines - Noisy channel model



(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 $\text{argmax}_{c \in \mathcal{C}} P(x|y_1, x_1, \cdots, y_k, x_k, c)$.

Pretrained models

Meta train

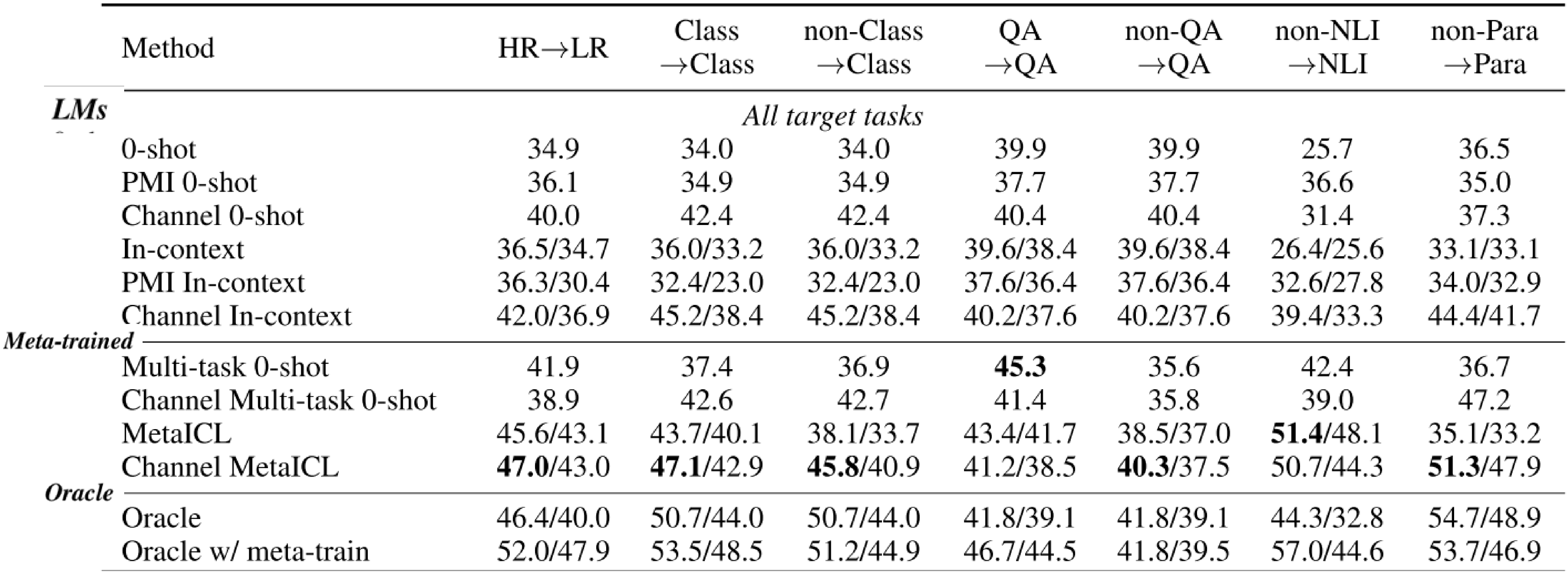
GPT-2 Large (770M parameters)

Inference only

GPT-J (6B parameters)

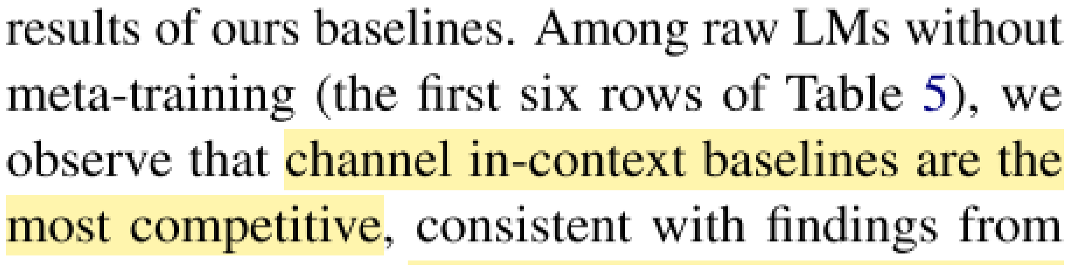
Results

Results



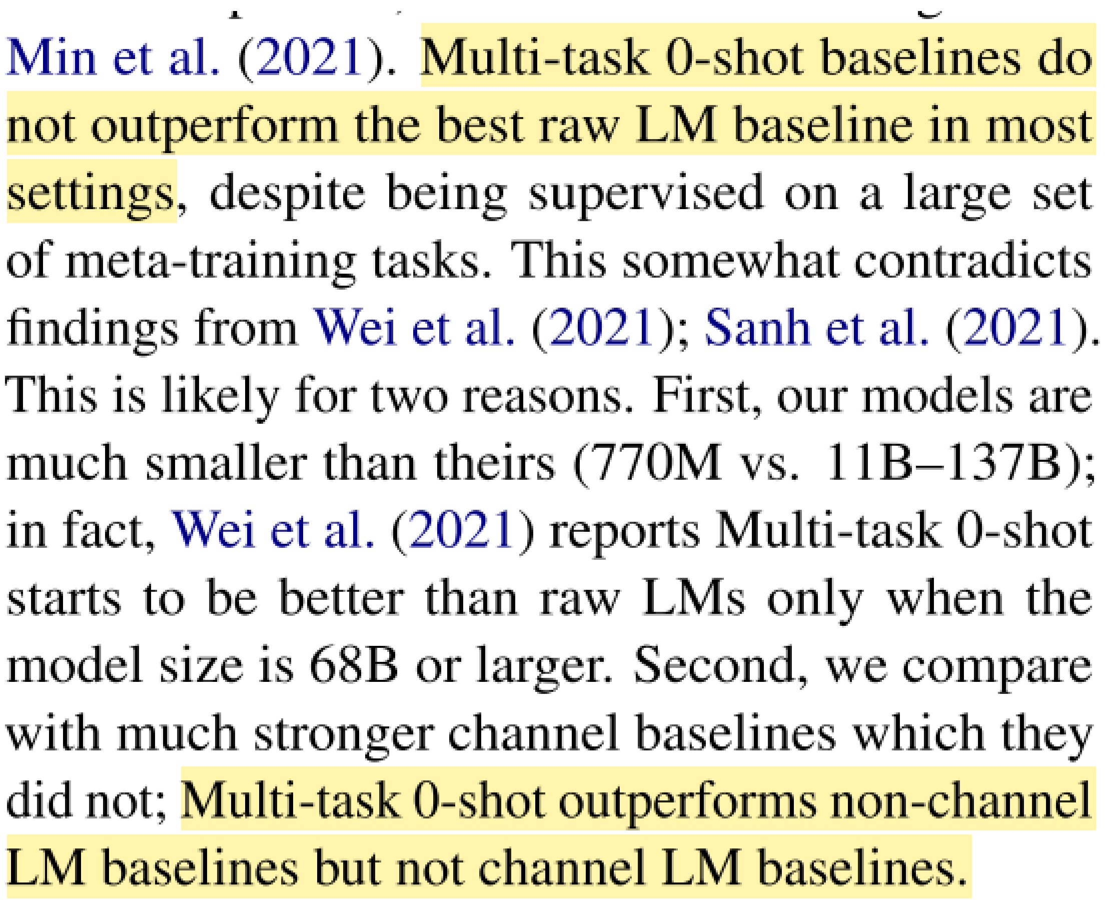
Method	HR→LR	Class →Class	non-Class →Class	QA →QA	non-QA →QA	non-NLI →NLI	non-Para →Para
LMs							
				All target tasks			
0-shot	34.9	34.0	34.0	39.9	39.9	25.7	36.5
PMI 0-shot	36.1	34.9	34.9	37.7	37.7	36.6	35.0
Channel 0-shot	40.0	42.4	42.4	40.4	40.4	31.4	37.3
In-context	36.5/34.7	36.0/33.2	36.0/33.2	39.6/38.4	39.6/38.4	26.4/25.6	33.1/33.1
PMI In-context	36.3/30.4	32.4/23.0	32.4/23.0	37.6/36.4	37.6/36.4	32.6/27.8	34.0/32.9
Channel In-context	42.0/36.9	45.2/38.4	45.2/38.4	40.2/37.6	40.2/37.6	39.4/33.3	44.4/41.7
Meta-trained							
Multi-task 0-shot	41.9	37.4	36.9	**45.3**	35.6	42.4	36.7
Channel Multi-task 0-shot	38.9	42.6	42.7	41.4	35.8	39.0	47.2
MetaICL	45.6/43.1	43.7/40.1	38.1/33.7	43.4/41.7	38.5/37.0	**51.4**/48.1	35.1/33.2
Channel MetaICL	**47.0**/43.0	**47.1**/42.9	**45.8**/40.9	41.2/38.5	**40.3**/37.5	50.7/44.3	**51.3**/47.9
Oracle							
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

Analysis – Channel models



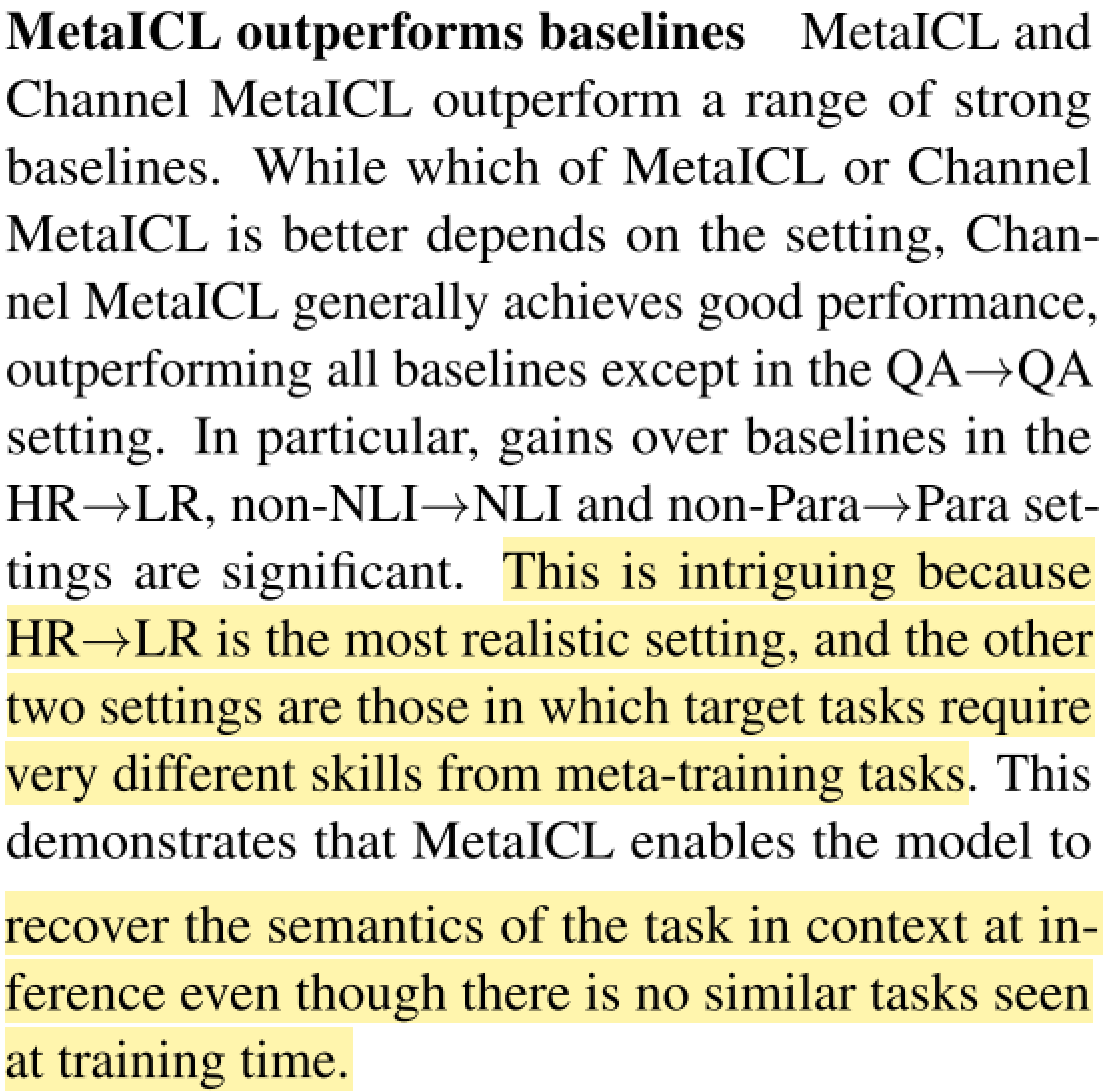
results of ours baselines. Among raw LMs without meta-training (the first six rows of Table 5), we observe that channel in-context baselines are the most competitive, consistent with findings from

Analysis – Multi-task learning



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 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.

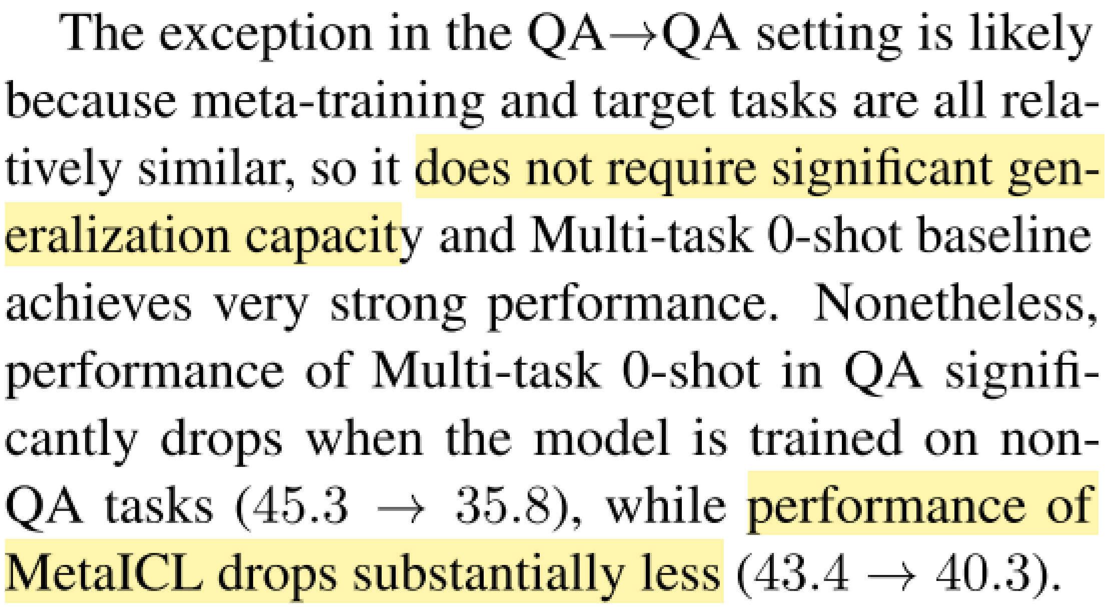
Analysis – MetaICL



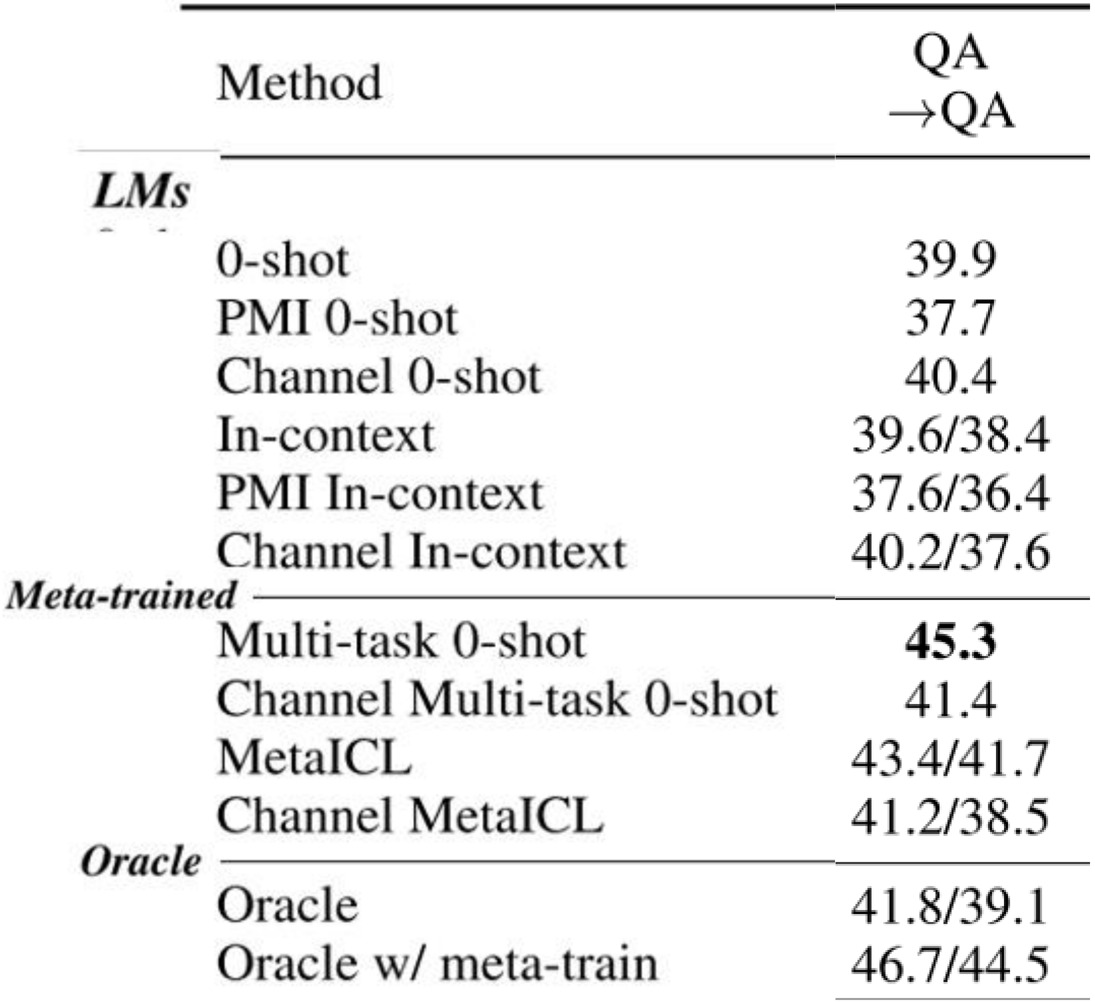
MetaICL outperforms baselines MetaICL and Channel MetaICL outperform a range of strong baselines. While which of MetaICL or Channel MetaICL is better depends on the setting, Channel MetaICL generally achieves good performance, outperforming all baselines except in the QA→QA setting. In particular, gains over baselines in the HR→LR, non-NLI→NLI and non-Para→Para settings are significant. This is intriguing because HR→LR is the most realistic setting, and the other two settings are those in which target tasks require very different skills from meta-training tasks. This demonstrates that MetaICL enables the model to

recover the semantics of the task in context at inference even though there is no similar tasks seen at training time.

Analysis – QA→ QA setting

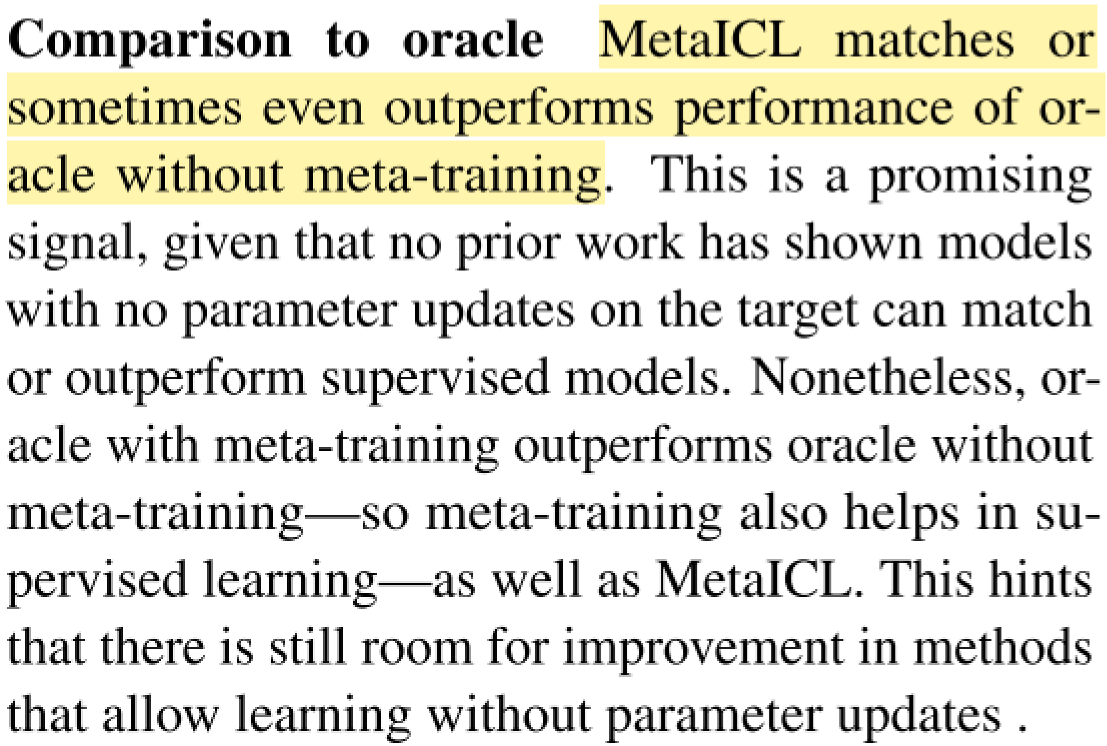


The exception in the QA→QA setting is likely because meta-training and target tasks are all relatively similar, so it does not require significant generalization capacity and Multi-task 0-shot baseline achieves very strong performance. Nonetheless, performance of Multi-task 0-shot in QA significantly drops when the model is trained on non-QA tasks (45.3 → 35.8), while performance of MetaICL drops substantially less (43.4 → 40.3).

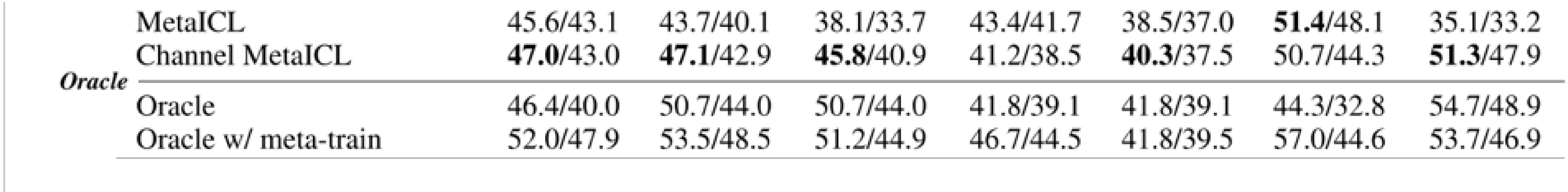


	Method	QA →QA
LMs		
	0-shot	39.9
	PMI 0-shot	37.7
	Channel 0-shot	40.4
	In-context	39.6/38.4
	PMI In-context	37.6/36.4
	Channel In-context	40.2/37.6
Meta-trained		
	Multi-task 0-shot	**45.3**
	Channel Multi-task 0-shot	41.4
	MetaICL	43.4/41.7
	Channel MetaICL	41.2/38.5
Oracle		
	Oracle	41.8/39.1
	Oracle w/ meta-train	46.7/44.5

Analysis – Oracle

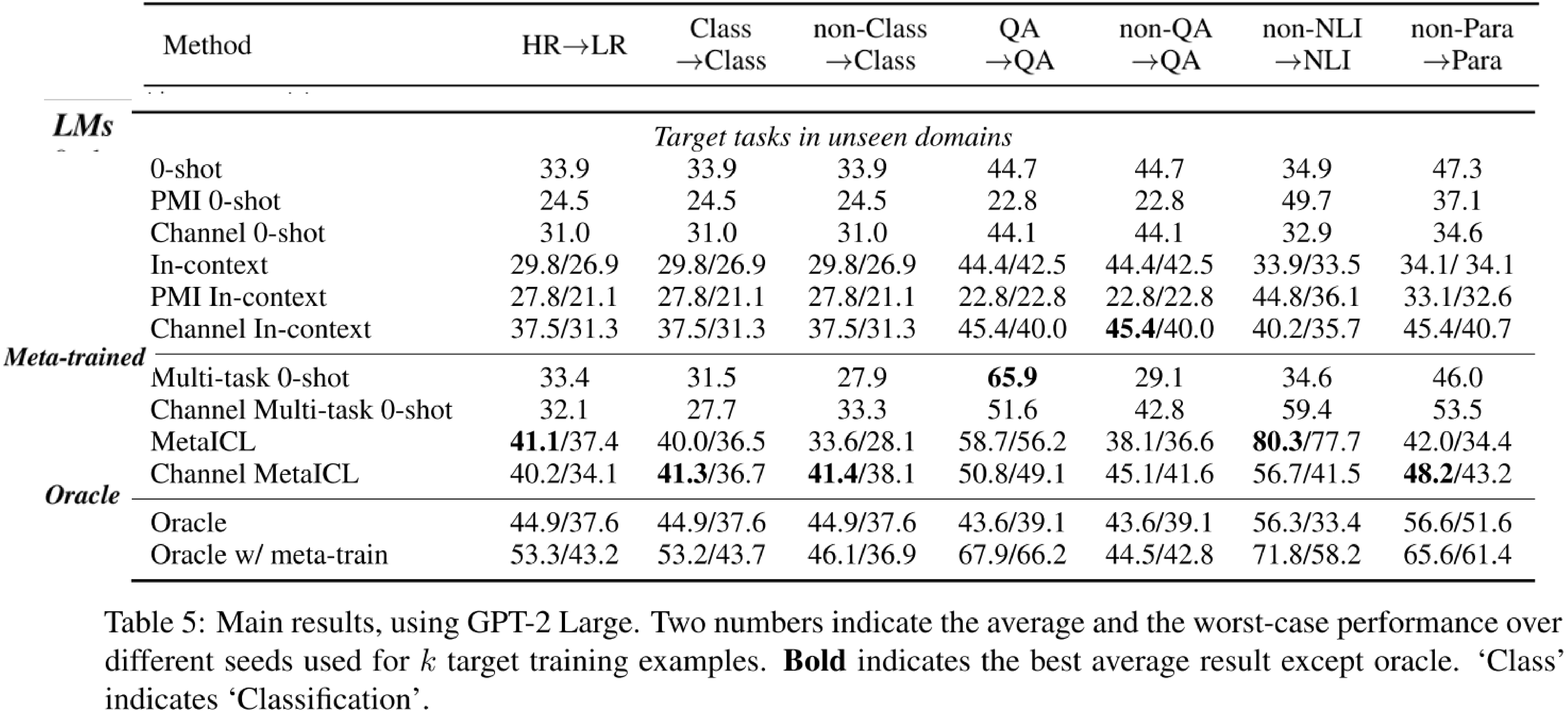


Comparison to oracle MetaICL matches or sometimes even outperforms performance of oracle without meta-training. This is a promising signal, given that no prior work has shown models with no parameter updates on the target can match or outperform supervised models. Nonetheless, oracle with meta-training 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 .



Oracle	MetaICL	45.6/43.1	43.7/40.1	38.1/33.7	43.4/41.7	38.5/37.0	**51.4**/48.1	35.1/33.2
	Channel MetaICL	**47.0**/43.0	**47.1**/42.9	**45.8**/40.9	41.2/38.5	**40.3**/37.5	50.7/44.3	**51.3**/47.9
	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

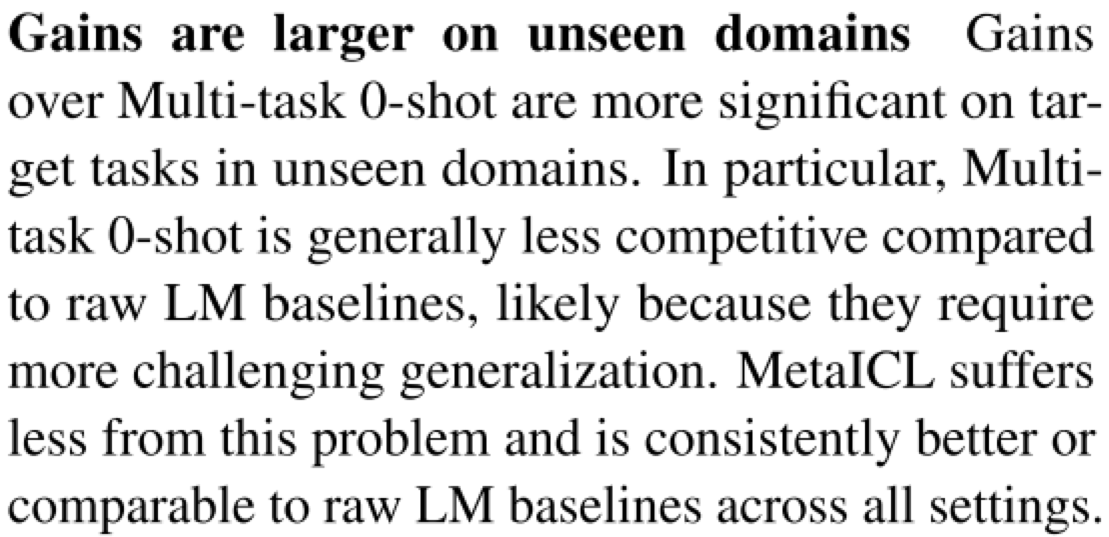
Results – Unseen domains



	Method	HR→LR	Class →Class	non-Class →Class	QA →QA	non-QA →QA	non-NLI →NLI	non-Para →Para
	Target tasks in unseen domains							
LMs	0-shot	33.9	33.9	33.9	44.7	44.7	34.9	47.3
	PMI 0-shot	24.5	24.5	24.5	22.8	22.8	49.7	37.1
	Channel 0-shot	31.0	31.0	31.0	44.1	44.1	32.9	34.6
	In-context	29.8/26.9	29.8/26.9	29.8/26.9	44.4/42.5	44.4/42.5	33.9/33.5	34.1/ 34.1
	PMI In-context	27.8/21.1	27.8/21.1	27.8/21.1	22.8/22.8	22.8/22.8	44.8/36.1	33.1/32.6
	Channel In-context	37.5/31.3	37.5/31.3	37.5/31.3	45.4/40.0	**45.4**/40.0	40.2/35.7	45.4/40.7
Meta-trained	Multi-task 0-shot	33.4	31.5	27.9	**65.9**	29.1	34.6	46.0
	Channel Multi-task 0-shot	32.1	27.7	33.3	51.6	42.8	59.4	53.5
	MetaICL	**41.1**/37.4	40.0/36.5	33.6/28.1	58.7/56.2	38.1/36.6	**80.3**/77.7	42.0/34.4
	Channel MetaICL	40.2/34.1	**41.3**/36.7	**41.4**/38.1	50.8/49.1	45.1/41.6	56.7/41.5	**48.2**/43.2
Oracle	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

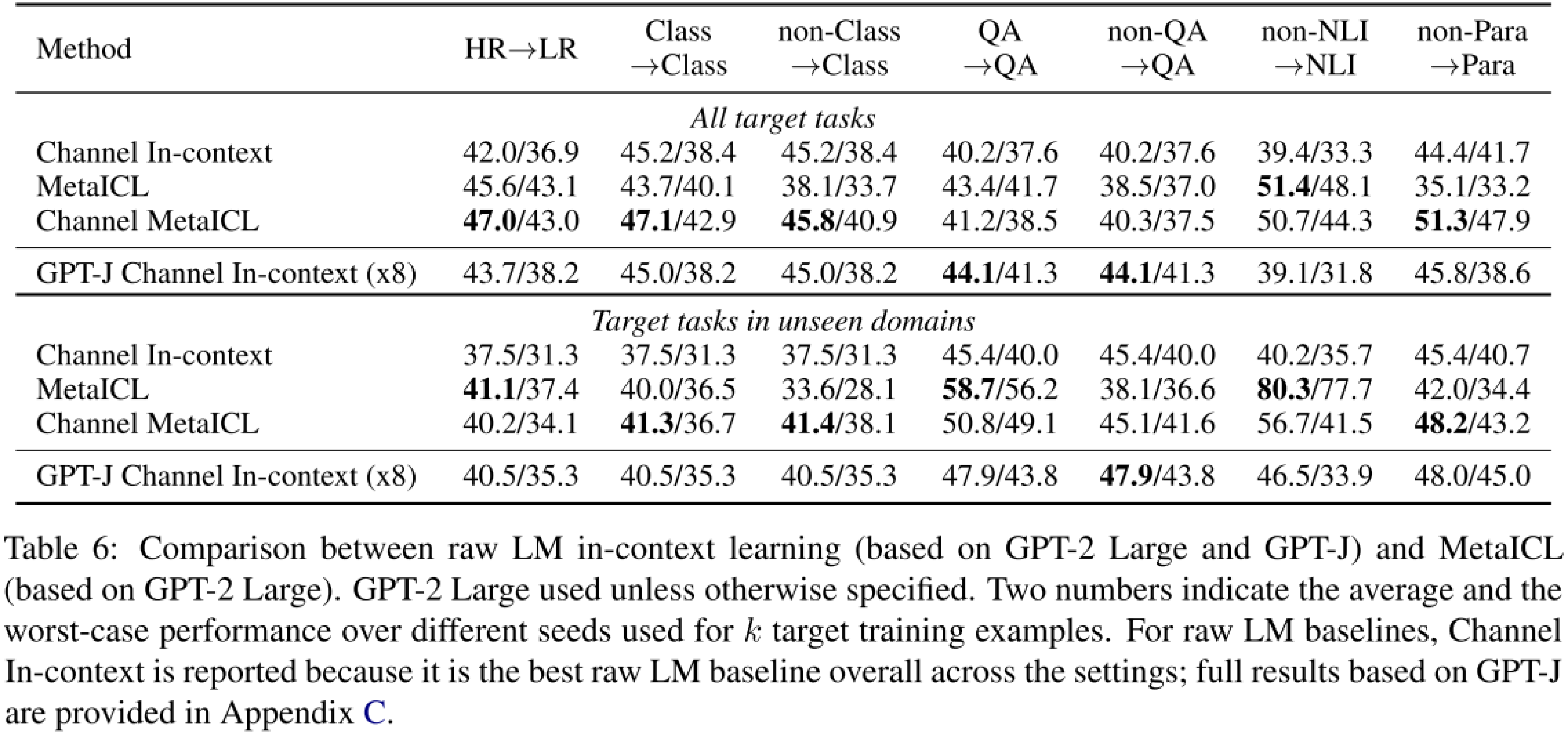
Table 5: Main results, using GPT-2 Large. Two numbers indicate the average and the worst-case performance over different seeds used for k target training examples. **Bold** indicates the best average result except oracle. ‘Class’ indicates ‘Classification’.

Analysis – Unseen domains



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.

Results – comparison to GPT-J (6B parameters)



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	42.0/36.9	45.2/38.4	45.2/38.4	40.2/37.6	40.2/37.6	39.4/33.3	44.4/41.7
MetaICL	45.6/43.1	43.7/40.1	38.1/33.7	43.4/41.7	38.5/37.0	**51.4**/48.1	35.1/33.2
Channel MetaICL	**47.0**/43.0	**47.1**/42.9	**45.8**/40.9	41.2/38.5	40.3/37.5	50.7/44.3	**51.3**/47.9
GPT-J Channel In-context (x8)	43.7/38.2	45.0/38.2	45.0/38.2	**44.1**/41.3	**44.1**/41.3	39.1/31.8	45.8/38.6
				Target tasks in unseen domains			
Channel In-context	37.5/31.3	37.5/31.3	37.5/31.3	45.4/40.0	45.4/40.0	40.2/35.7	45.4/40.7
MetaICL	**41.1**/37.4	40.0/36.5	33.6/28.1	**58.7**/56.2	38.1/36.6	**80.3**/77.7	42.0/34.4
Channel MetaICL	40.2/34.1	**41.3**/36.7	**41.4**/38.1	50.8/49.1	45.1/41.6	56.7/41.5	**48.2**/43.2
GPT-J Channel In-context (x8)	40.5/35.3	40.5/35.3	40.5/35.3	47.9/43.8	**47.9**/43.8	46.5/33.9	48.0/45.0

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.

Ablations

Ablations - Number of meta-training tasks

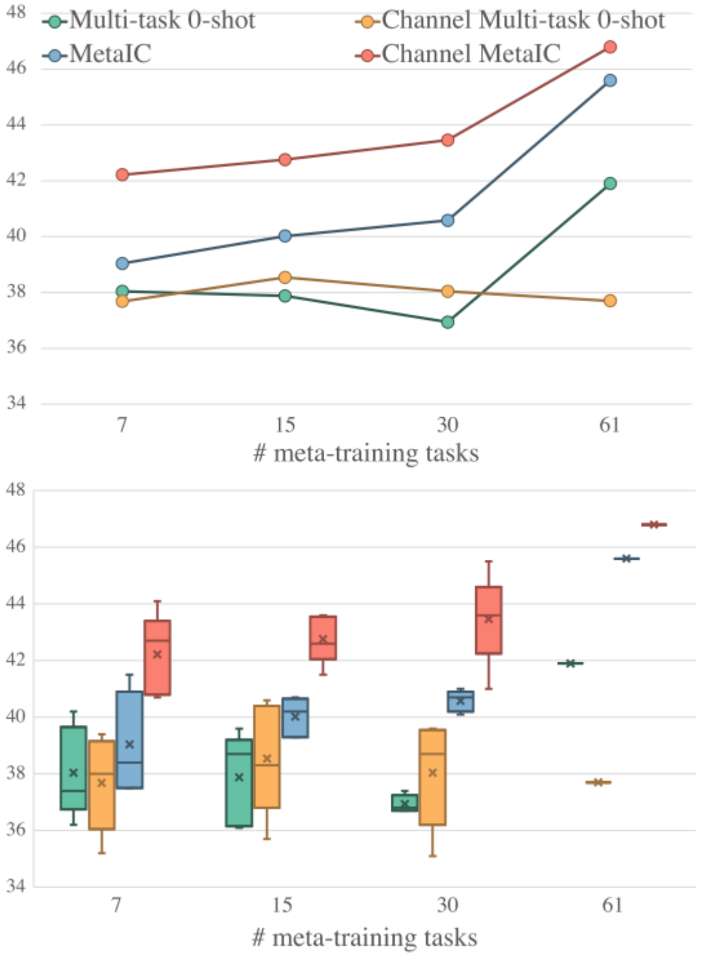
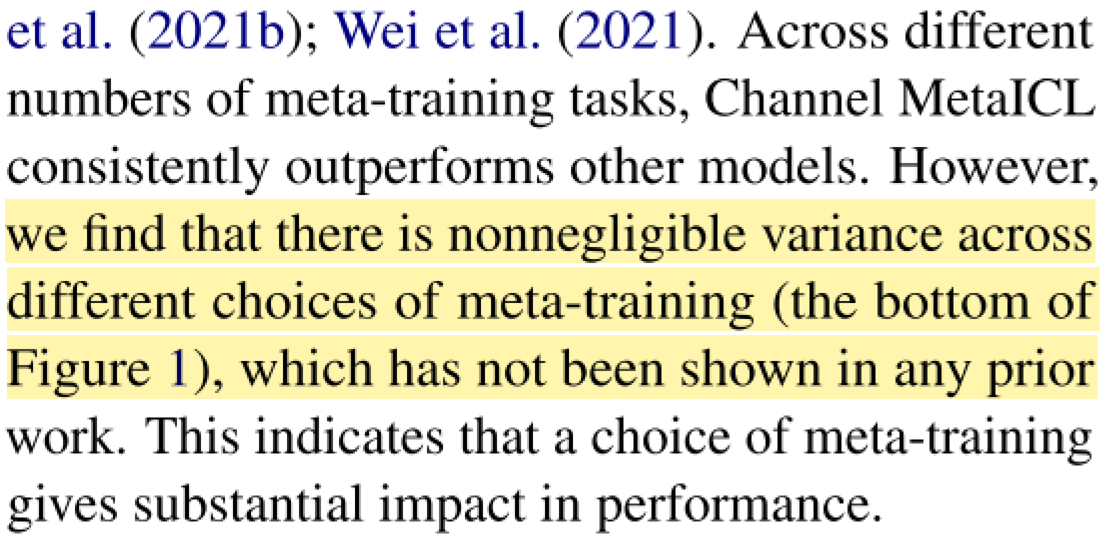
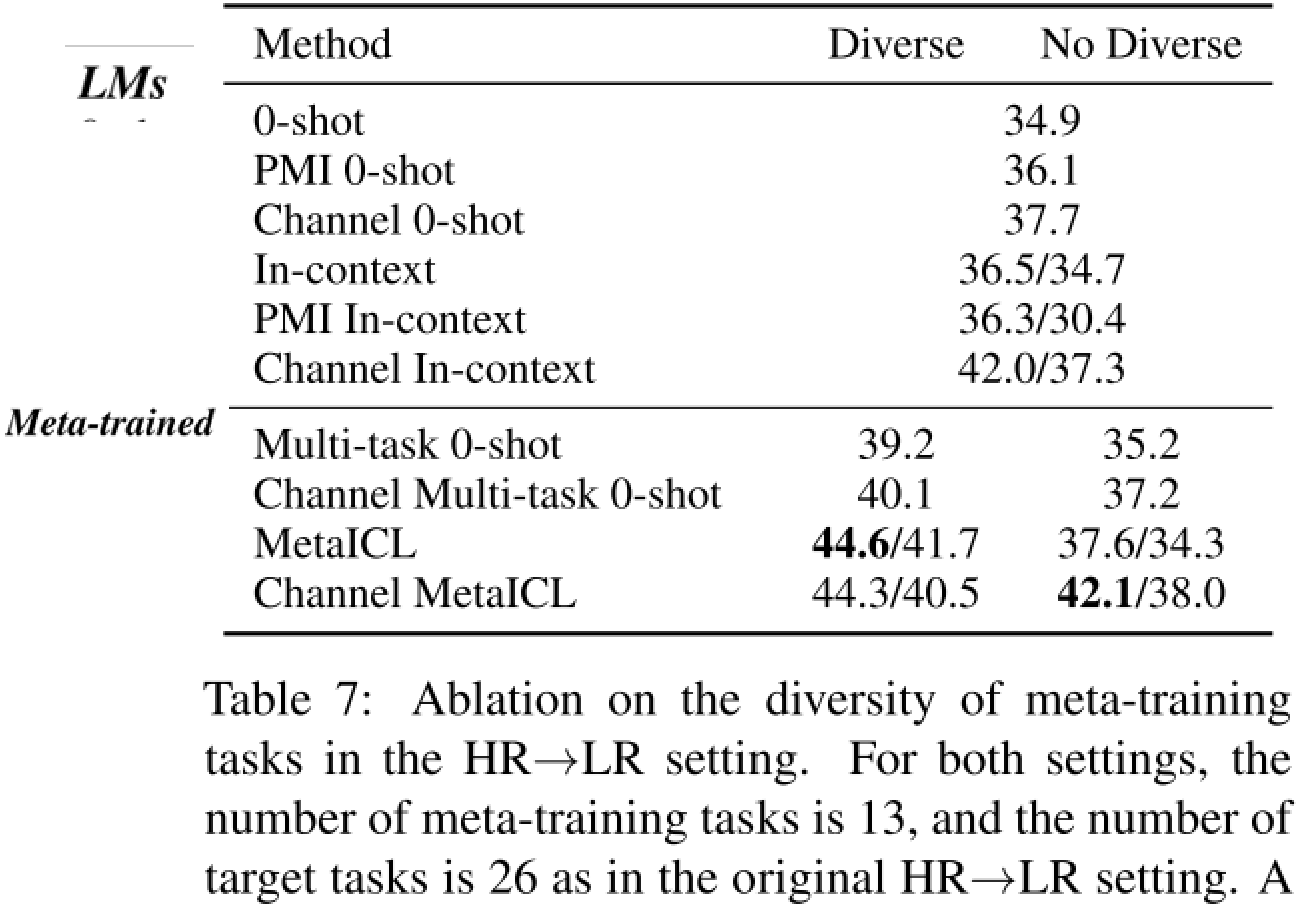


Figure 1: Ablation on the number of meta-training tasks ($\{7, 15, 30, 61\}$). The graph of the average (top) and the box chart (bottom) over different meta-training sets using 5 different random seeds (except for 61).



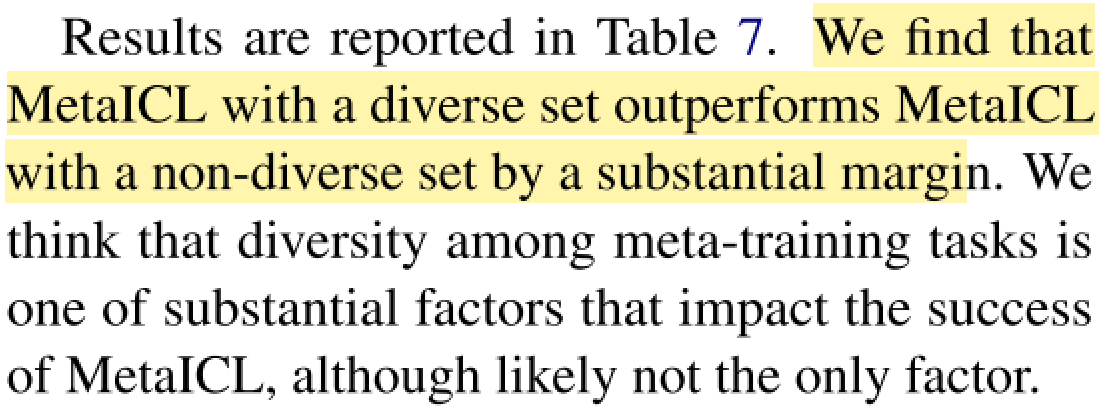
et al. (2021b); Wei et al. (2021). Across different numbers of meta-training tasks, Channel MetaICL consistently outperforms other models. However, we find that there is nonnegligible variance across different choices of meta-training (the bottom of Figure 1), which has not been shown in any prior work. This indicates that a choice of meta-training gives substantial impact in performance.

Ablations - Diversity in meta-training tasks



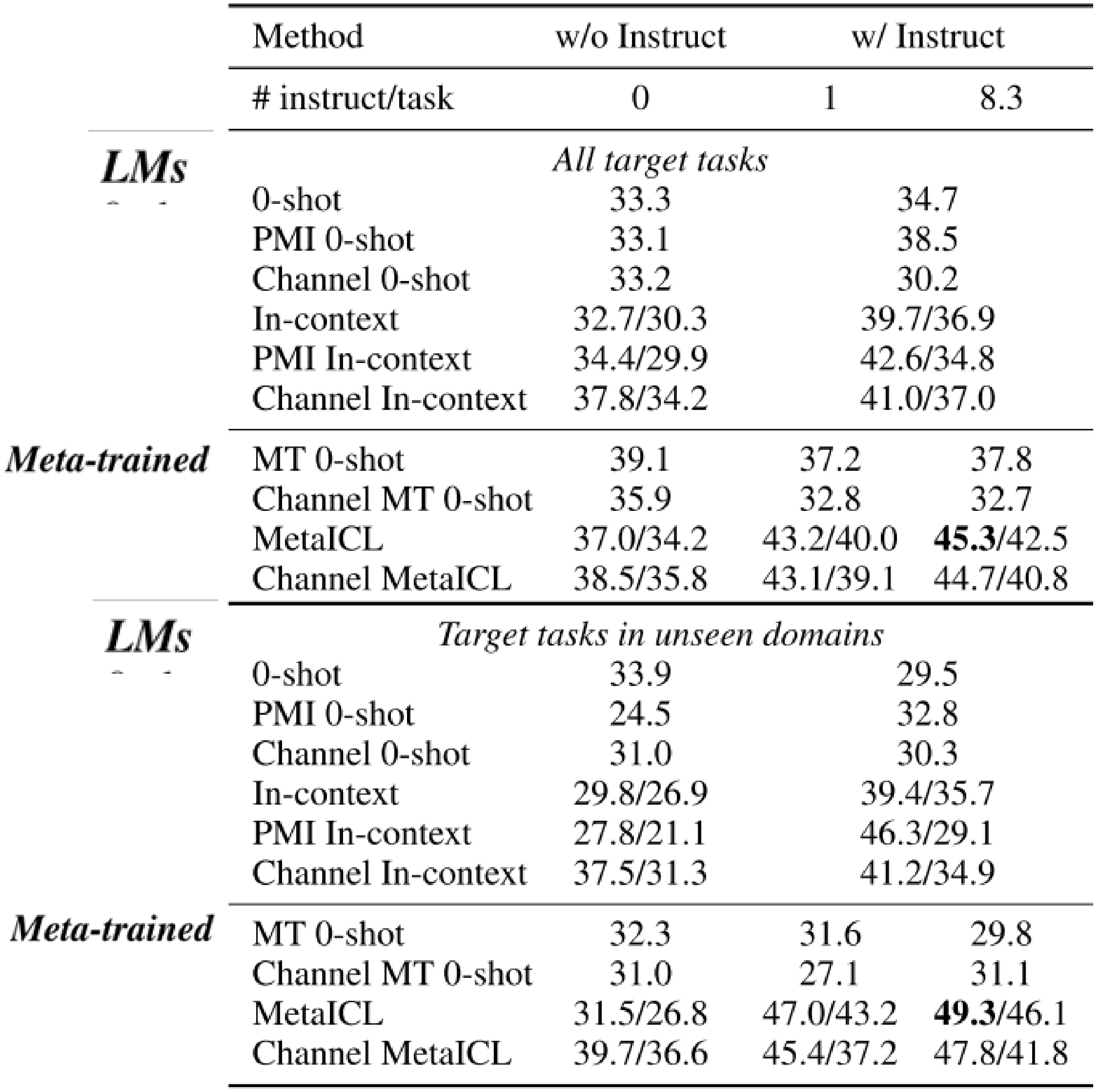
	Method	Diverse	No Diverse
LMs	0-shot	34.9	
	PMI 0-shot	36.1	
	Channel 0-shot	37.7	
	In-context	36.5/34.7	
	PMI In-context	36.3/30.4	
	Channel In-context	42.0/37.3	
Meta-trained	Multi-task 0-shot	39.2	35.2
	Channel Multi-task 0-shot	40.1	37.2
	MetaICL	**44.6**/41.7	37.6/34.3
	Channel MetaICL	44.3/40.5	**42.1**/38.0

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

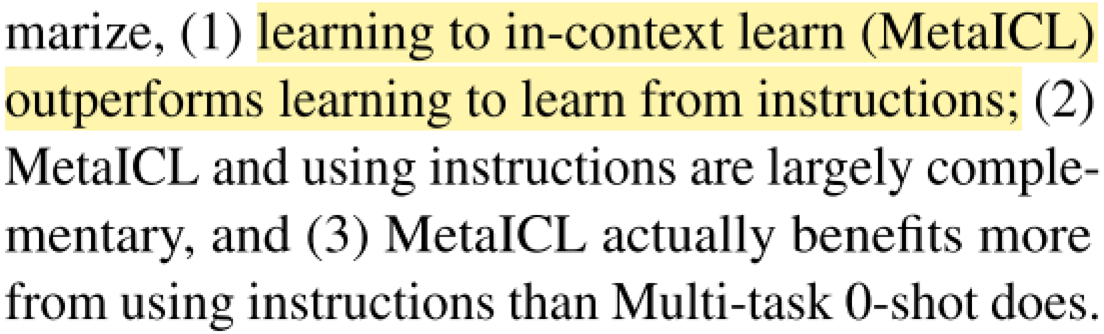


Results are reported in Table 7. We find that MetaICL with a diverse set outperforms MetaICL with a non-diverse set by a substantial margin. We think that diversity among meta-training tasks is one of substantial factors that impact the success of MetaICL, although likely not the only factor.

Ablations – Are instructions necessary?



	Method	w/o Instruct	w/ Instruct	
	# instruct/task	0	1	8.3
	All target tasks			
LMs	0-shot	33.3	34.7	
	PMI 0-shot	33.1	38.5	
	Channel 0-shot	33.2	30.2	
	In-context	32.7/30.3	39.7/36.9	
	PMI In-context	34.4/29.9	42.6/34.8	
	Channel In-context	37.8/34.2	41.0/37.0	
Meta-trained	MT 0-shot	39.1	37.2	37.8
	Channel MT 0-shot	35.9	32.8	32.7
	MetaICL	37.0/34.2	43.2/40.0	**45.3**/42.5
	Channel MetaICL	38.5/35.8	43.1/39.1	44.7/40.8
	Target tasks in unseen domains			
LMs	0-shot	33.9	29.5	
	PMI 0-shot	24.5	32.8	
	Channel 0-shot	31.0	30.3	
	In-context	29.8/26.9	39.4/35.7	
	PMI In-context	27.8/21.1	46.3/29.1	
	Channel In-context	37.5/31.3	41.2/34.9	
Meta-trained	MT 0-shot	32.3	31.6	29.8
	Channel MT 0-shot	31.0	27.1	31.1
	MetaICL	31.5/26.8	47.0/43.2	**49.3**/46.1
	Channel MetaICL	39.7/36.6	45.4/37.2	47.8/41.8



marize, (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.