FiD-ICL: A Fusion-in-Decoder Approach for Efficient In-Context Learning

Qinyuan Ye[1†] **Iz Beltagy**[2] **Matthew E. Peters**[2] **Xiang Ren**[1,2] **Hannaneh Hajishirzi**[2,3]
[1]University of Southern California [2]Allen Institute for AI [3]University of Washington
{qinyuany,xiangren}@usc.edu {beltagy,matthewp,hannah}@allenai.org

Abstract

Large pre-trained models are capable of few-shot in-context learning (ICL), *i.e.*, performing a new task by prepending a few demonstrations before the test input. However, the concatenated demonstrations are often excessively long and induce additional computation. Inspired by fusion-in-decoder (FiD) models which efficiently aggregate more passages and thus outperforms concatenation-based models in open-domain QA, we hypothesize that similar techniques can be applied to improve the efficiency and end-task performance of ICL. To verify this, we present a comprehensive study on applying three fusion methods—concatenation-based (early fusion), FiD (intermediate), and ensemble-based (late)—to ICL. We adopt a meta-learning setup where a model is first trained to perform ICL on a mixture of tasks using one selected fusion method, then evaluated on held-out tasks for ICL. Results on 11 held-out tasks show that FiD-ICL matches or outperforms the other two fusion methods. Additionally, we show that FiD-ICL (1) is 10x faster at inference time compared to concat-based and ensemble-based ICL, as we can easily precompute the representations of in-context examples and reuse them; (2) enables scaling up to meta-training 3B-sized models, which would fail for concat-based ICL.[1]

1 Introduction

Large pre-trained models demonstrated remarkable performance in learning new language tasks via few-shot fine-tuning (FT)—initializing a model with pre-trained weights and optimizing it based on a few examples (Zhang et al., 2021). FT-based approaches currently achieve state-of-the-art performance (Liu et al., 2022), yet they require back-propagating and computing gradients over the full models, which can be prohibitive under memory and resource constraints.

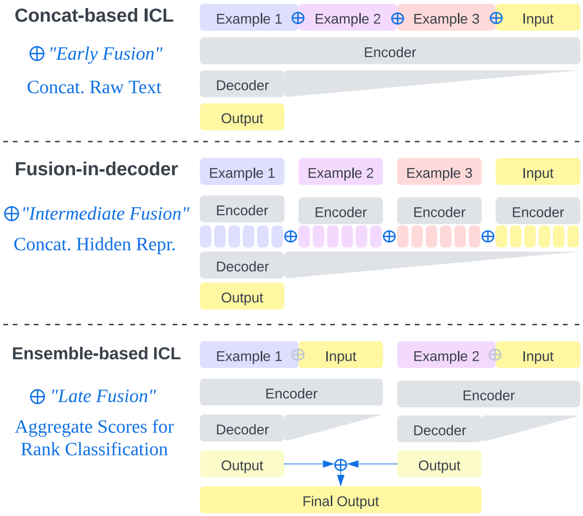

Figure 1: **Overview.** In this study we compare different methods to incorporate examples for in-context learning. We term these as "fusion methods". ⊕ marks where and how fusion is implemented.

An alternative approach to few-shot learning is in-context learning (ICL). By concatenating a few examples and prepending them before the test instance, the model can perform a new task readily (Brown et al., 2020). ICL is more efficient at its "learning" stage, as it only uses one forward-pass and does not require gradients at all. Yet it is less efficient at inference stage, as the concatenated examples can become overly long and induce excessive computational costs. Additionally, ICL performance typically falls short of FT-based methods (Liu et al., 2022).

These limitations and trade-offs between ICL and FT motivate our exploration of methods that are efficient at *both* few-shot learning and inference time. In particular, we aim to achieve this by exploring different methods that incorporate (or "fuse") the in-context examples during inference. We draw connections between open-domain QA (Chen and Yih, 2020) and ICL, since both problems task a model with reading long context (multiple retrieved

[†] Work done while interning at Allen Institute for AI.
[1] Code: https://github.com/INK-USC/FiD-ICL

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8158–8185
July 9-14, 2023 ©2023 Association for Computational Linguistics

passages vs. multiple examples) and making a prediction based on the context (answer a relevant question vs. infer about a new input). Further, we draw inspiration from fusion-in-decoder (FiD; Izacard and Grave 2021), a method that can efficiently "aggregate evidence" from many retrieved passages to answer open-domain questions. Given that FiD models significantly outperforms concatenation-based methods for open-domain QA, we hypothesize that FiD can be applied to ICL analogously to improve its efficiency and end-task performance.

To verify this, we present a systematic comparison of three different methods to incorporate in-context examples: concatenation-based, fusion-in-decoder, and ensemble-based (Fig. 1). We term them as "fusion" methods, and characterize them as *early*, *intermediate*, and *late* fusion, based on their formulation. We conduct comprehensive experiments with the P3 dataset (Sanh et al., 2022) in a meta-learning setting similar to Min et al. (2022b)– *i.e.*, a model is first trained to perform ICL on a mixture of tasks using one selected fusion method, then evaluated on held-out tasks for ICL.

Our empirical results suggest that, while being significantly more efficient on computation complexity and memory usage, FiD-ICL is comparable to or outperforms the other two fusion methods on the 11 P3 held-out tasks. This observation is consistent across three different model sizes. Notably, given the memory efficiency of FiD-ICL, we are able to meta-train 3B-sized ICL models within an academic budget, which would lead to out-of-memory errors and fail in the case of concatenation-based ICL. Our best model, FiD-ICL trained from T5-LM-XL (3B), narrows the gap with T-Few (Liu et al., 2022)–a state-of-the-art few-shot FT method–to 3% difference in accuracy.

Moreover, our formulation of FiD-ICL decouples the computation of few-shot examples and the test input, allowing the computation over the few-shot examples to be pre-computed and reused. As a result, FiD-ICL is up to 10x faster at inference time compared to the other two fusion methods. We further support this argument with computation cost analysis and inference speed tests.

However, it is still questionable whether the ICL models we investigate learn effectively from the few-shot examples. We replicate a set of diagnostic experiments in Min et al. (2022c) by perturbing the in-context examples (*e.g.*, using fewer/more shots, replacing correct labels with random labels). We found that ICL methods, regardless of what fusion method is employed, are still rather insensitive to these perturbations, and do not rely on input-label mapping as much as expected. These observations call for further investigation and efforts to improve the effectiveness of in-context learning.

2 Related Work

Few-shot Fine-tuning. It has been shown that fine-tuning a large pre-trained model with only a few examples yields strong performance on a wide range of NLP tasks (Zhang et al., 2021). The performance can be further improved by incorporating prompts and demonstrations in the input (Schick and Schütze, 2021; Gao et al., 2021). Moreover, parameter-efficient fine-tuning methods can be applied to improve memory and storage costs (Liu et al., 2022). However, these methods are still relatively expensive at training time as they require back-propagating through the full model.

In-Context Learning. In-context learning (ICL) is an alternative approach for few-shot learning by simply concatenating the few-shot examples and using them as a prompt before the actual inference example. Very large pre-trained models, such as GPT-3 (Brown et al., 2020) and PaLM (Chowdhery et al., 2022), are capable of ICL *off the shelf* (*i.e.*, without any gradient update) and achieve competitive performance. Smaller models can be meta-trained to obtain this capability (Chen et al., 2022; Min et al., 2022b). We follow the latter problem setting and focus on smaller models (up to 3B size). One disadvantage of ICL is that the *inference cost* grows rapidly as the number of few-shot examples increases. Researchers also find that ICL models do not rely on input-label mapping as much as expected, casting doubts on the effectiveness of ICL (Min et al., 2022c).

Zero-shot/Few-shot Task Generalization. Towards the goal of building a generalist NLP system, recent works adopt a meta-learning paradigm (Schmidhuber, 1987) and propose to meta-train a model on a set of given tasks (*i.e.*, meta-train set). The resulting model is expected to solve novel tasks (in a meta-test set) in a zero-shot or few-shot setting. This is made possible by unifying task format with prompts (Zhong et al., 2021; Sanh et al., 2022; Wei et al., 2022), providing task instructions/descriptions (Weller et al., 2020; Mishra et al., 2022; Wang et al., 2022b), and scaling up and diversifying the meta-train set (Ye et al., 2021; Chung et al., 2022). The meta-training set is typically utilized with multi-task learning (Caruana, 1997) or model-agnostic meta-learning (Finn et al., 2017). In this work, we compare different fusion methods in such meta-learning setting. In particular, we focus on applying fusion-in-decoder technique to ICL and investigate the benefits and limitations of it.

3 Investigating Fusion Methods for ICL

3.1 Problem Setting

Overview. Our goal is to build models that are capable of *few-shot in-context learning* without gradient updates when handling an unseen task. Prior work show that such capabilities can be obtained by learning from a collection of seen tasks and training the model on *concatenation-based* in-context learning (Chen et al., 2022; Min et al., 2022b). In this work, we examine alternative ways to synthesize and incorporate information in multiple shots (*i.e.*, "fusion method").

Data. We use three non-overlapping sets of tasks, meta-train ($\mathcal{T}_{train}$), meta-valid ($\mathcal{T}_{valid}$), and meta-test ($\mathcal{T}_{test}$). We assume all tasks are in text-to-text format. Each task T in $\mathcal{T}_{train}$ contains a set of training examples, *i.e.*, $T = \{(x, y)\}$. Tasks in $\mathcal{T}_{valid}$ and $\mathcal{T}_{test}$ are *few-shot*. Each task T in $\mathcal{T}_{valid}$ or $\mathcal{T}_{test}$ contains a support set and a query set.[2] Models in this study are expected to *learn* from the k-shot support set $\{(x_i^{(s)}, y_i^{(s)})\}$ without gradient updates, and do inference on the query set $\{(x_i^{(q)}, y_i^{(q)})\}$. Additionally, we assume all tasks in $\mathcal{T}_{valid}$ and $\mathcal{T}_{test}$ can be evaluated with rank classification, with a set of choices C given for each query example $(x^{(q)}, y^{(q)})$. In this case, the model does inference by ranking the probabilities assigned to each choice $c \in C$.

Meta-Training and Inference Procedure. We closely follow the procedure described in MetaICL (Min et al., 2022b). In the **meta-training** phase, we first sample one task T from $\mathcal{T}_{train}$, then sample k support examples $\{(x_i^{(s)}, y_i^{(s)})\}$ and m query examples $\{(x_i^{(q)}, y_i^{(q)})\}$ from the task. We update the model (using a selected fusion method) to minimize the loss of generating the correct target sequences $y_i^{(q)}$. In the **meta-test/inference** phase, for each unseen task in $\mathcal{T}_{test}$, we are given a fixed set of k-shot support examples, and the model is expected to do inference on all query examples $\{(x_i^{(q)}, y_i^{(q)})\}$.

3.2 Fusion Methods

Previously in Fig. 1 we provide the visualizations of fusion methods that we compare. In this section, we reinstate our motivations and describe them more formally.

Overview. Fusion-in-decoder (Izacard and Grave, 2021) is a competitive method for incorporating multiple retrieved documents for open-domain QA, and it significantly outperforms concatenation-based methods (Lewis et al., 2020b). Bringing these insights to few-shot learning, in-context learning can be viewed as concatenating the raw text of few-shot examples and doing *"early fusion"*. We investigate whether doing *fusion* at later stages, such as fusion-in-decoder (*i.e.*, *"intermediate fusion"*) or ensemble (*i.e.*, *"late fusion"*) will bring additional benefits.[3]

Early Fusion: Concatenation-based ICL. This refers to the method of concatenating $(x_1^{(s)}, y_1^{(s)}, ..., x_k^{(s)}, y_k^{(s)}, x^{(q)})$ into a long text input and feeding this sequence to a model. The model is expected to generate $y^{(q)}$. Specifically, we compute $\arg\max_{c \in C} P(c|x_1^{(s)}, y_1^{(s)}, ..., x_k^{(s)}, y_k^{(s)}, x^{(q)})$. Note that in transformer models the computation cost typically grow quadratically with sequence length (and thus the number of shots).

Intermediate Fusion: Fusion-in-decoder (FiD). In fusion-in-decoder, the support examples $(x_1^{(s)}, y_1^{(s)}), ..., (x_k^{(s)}, y_k^{(s)})$ and the query $x^{(q)}$ are encoded *separately* by the *same* encoder layers in the transformer model. The representations produced by the last encoder layer are then concatenated (*i.e.*, "fused") and sent to the decoder layers. In this way, the computation cost grows linearly with the number of shots.

Note that our formulation is slightly different from the original fusion-in-decoder models for open-domain QA (ODQA). In ODQA, the question ($x^{(q)}$) is first concatenated with each retrieved paragraph ($x_i^{(s)}$) and then encoded separately by

[2] Support/query set are the same as few-shot train/test set. We adopt these terms to distinguish from meta-train/meta-test.

[3] The terms of early/intermediate/late fusion are inspired by multi-modal literature, but their meanings are slightly different in this work.

Method	Meta-Train		Meta-Test	
	T0	ICL	Fine-tune	# shots
Initialize from T5-LM				
Zero-shot	✗	✗	✗	0
Concat/FiD/Ensemble-ICL	✗	✓	✗	k
Simple/TFew Fine-tune	✗	✗	✓	k
Initialize from T0				
Zero-shot	✓	✗	✗	0
Concat/FiD/Ensemble-ICL	✓	✓	✗	k
Simple/TFew Fine-tune	✓	✗	✓	k

Table 1: Meta-training and inference procedure for all compared methods.

the model. For ICL, we decouple the computation of support examples and the query example. In this way, the support examples can be encoded only once and re-used throughout the inference phase. See §5.2 for discussion.

Late Fusion: Ensemble-based ICL. Early fusion and intermediate fusion naturally bring us to the idea of ensemble-based approaches, which are effectively doing *"late fusion"*. They are previously explored in Min et al. (2022a) for classification tasks and demonstrate competitive performance. We implement this by training one-shot concat-based ICL models and aggregating the k different predictions at inference time. More specifically, we compute $\arg\max_{c\in C} \sum_{i=1}^{k} P(c|x_i^{(s)}, y_i^{(s)}, x^{(q)})$. Theoretically, the cost of ensemble-based ICL grows linearly with the number of shots.

Other Variants. Adapting FiD in open-domain QA for ICL is *non-trivial*. In the early stages of this work, we also examined two more variants named as FiD-Pairwise and FiD+Ensemble. FiD-Pairwise is closer to the original FiD implementation for open-domain QA. FiD+Ensemble a hybrid method that combines the techniques in FiD and Ensemble. Details are elaborated in Fig 6 and §A.1. The fusion-in-decoder design illustrated in Fig. 1 is the best one in our preliminary study, and therefore we adopt it in the main experiments.

4 Experiment Settings

4.1 Data

We use Public Pool of Prompts (P3) dataset (Sanh et al., 2022). The dataset includes a collection of diverse NLP tasks with crowd-sourced prompt templates. The tasks are partitioned into a Meta-Train set and a Meta-Test set. In the main experiments we use all 11 tasks in the meta-test set (Meta-Test-11). For analysis experiments, we use a subset of 7 tasks for faster experimentation (Meta-Test-7). We use 16 shots for all few-shot experiments, unless specified otherwise. Additionally, we use 14 BIG-bench tasks (Srivastava et al., 2023) as a Meta-Validation set for selecting the best checkpoint. We provide the full list of datasets and more details in Table 4 and §B.

4.2 Model

We limit our scope to encoder-decoder models for our experiments.[4] We use T5-LM-Adapt models[5] and T0 models (Sanh et al., 2022) as initializations in our experiments. The two model groups have the same model architecture but different weights; T0 is trained to multi-task on the P3 meta-train set using T5-LM-Adapt as initialization. We experiment with models of three different sizes: Base (250M), Large (800M), XL (3B).[6]

4.3 Compared Methods

The goal of using few-shot ICL methods is to learn from the few-shot examples so that it improves on top of zero-shot performance; further, we aim to close its gap to few-shot fine-tuning, which requires gradient updates. To quantify these, we include zero-shot inference and few-shot fine-tuning in our experiments, in addition to the three fusion methods that we compare. We provide an overview of the training and evaluation procedure of these methods in Table 1.

Zero-shot. We directly evaluate T5-LM-Adapt and T0 models on the Meta-Test, in the zero-shot setting.

Few-shot ICL. We initialize from either T5-LM-Adapt or T0, meta-train it with the three fusion methods (concatenation, fusion-in-decoder, ensemble) described in §3.2. We evaluate all saved checkpoints on Meta-Validation, then evaluate the one selected checkpoint on Meta-Test. Unless specified otherwise, we use 16 shots during training and evaluation.

[4] We elaborate our discussion on encoder-decoder vs. decoder-only models in §A.4.

[5] `https://huggingface.co/google/t5-xl-lm-adapt`

[6] We replicate the experiment setting in Sanh et al. (2022) and trained our own T0-Base/Large/3B model for this work. Notably, our reproduction of T0-3B outperforms the public checkpoint by a large margin, suggesting that the public T0-3B checkpoint may be undertrained. See §C.1 for details on training these models.

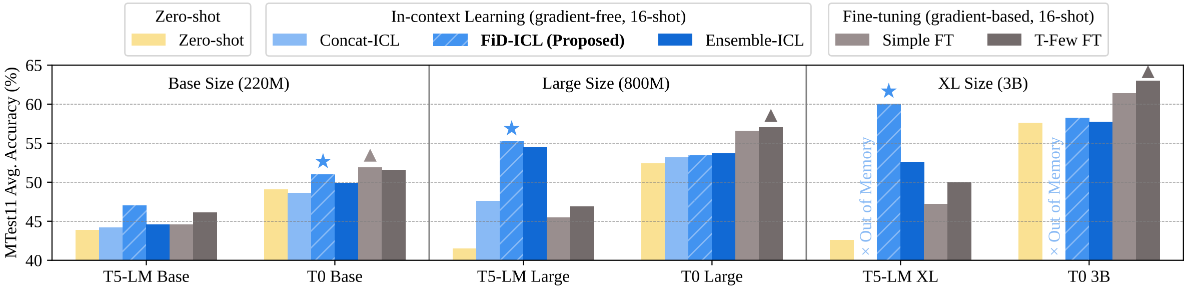

Figure 2: **Main Results on Meta-Test-11.** Bar height represents average accuracy on Meta-Test-11. X-tick labels represent the model size and initialization. Methods within each size group should be compared together (see Table 1 for difference in training procedure). ★ marks the best ICL method in each size group, ▲ marks the best fine-tuning method in each size group. **Observations: (1)** FiD-ICL outperforms Concat-ICL and Ensemble-ICL in all three size groups. **(2)** FiD-ICL (T5-LM XL) narrows the performance gap between ICL and T-Few to be 3%.

Few-shot Fine-tuning. For each meta-test task, we fine-tune either T5-LM-Adapt or T0 with the few-shot examples. Apart from simple fine-tuning, we also experiment with the T-Few fine-tuning recipe (Liu et al., 2022), which updates only a small portion of parameters, and includes an unlikelihood loss and a length normalization loss during fine-tuning.

4.4 Reporting the Results

For a high-level comparison across different methods, we report Meta-Test-11 average accuracy. Note that this one number is taking average on three levels: (1) averaging over the 11 held-out tasks; (2) for each task, averaging over all prompts associated with the task (in P3 dataset, each task is accompanied with multiple prompts); (3) for each (task, prompt) pair, averaging over 5 different samples of few-shot examples, to mitigate the influence brought by a specific set of few-shot examples.

We also report detailed per-task performance of Meta-Test-11 for more fine-grained analysis.

5 Experiment Results

5.1 Performance on Held-out Tasks

Following our experiment settings, we present the results of all compared methods in Fig. 2. We have the following observations. **Firstly**, we highlight that the efficient design of FiD-ICL and Ensemble-ICL enables us to train them in a larger scale (*e.g.*, 3B models) on an academic budget. We fail to do so for Concat-ICL as training with a max sequence length of 4096 results in out-of-memory errors.[7]

Secondly, when comparing the three fusion methods, FiD is comparable or outperforms the other two fusion methods in all three model sizes. Izacard and Grave (2021) attribute the success of FiD to "scaling to large number of contexts" in the encoder and "better aggregating evidence from multiple passages" in the decoder. We conjecture that the same inductive biases are also beneficial for few-shot ICL. **Thirdly**, our best FiD-ICL model (trained from T5-LM XL) achieves an average accuracy of 60.0% on Meta-Test-11. As a gradient-free method, this leaves a 1.4% gap compared to simple fine-tuning, and a 3.0% gap to T-Few fine-tuning (using T0-3B). This demonstrates the great potential of gradient-free ICL methods, and we hope future work can further improve ICL to close the gap.

5.2 Efficiency

One major motivation of our work is *efficiency*—to find a few-shot learning method that is efficient at *both* few-shot learning and inference time. In this section we estimate and compare the computational cost and inference speed of all methods.

Computation Complexity. Our estimation is based on the following assumptions: (1) the output length l_{out} is much smaller than the input length l_{in}, *i.e.*, $l_{out} \ll l_{in}$, so that the cost for an input (l_{in}) and a complete in-context example ($l_{in} + l_{out}$) are roughly comparable, *i.e.*, $l_{out} + l_{in} \approx l_{in}$; (2) training (forward and backward pass) requires 3 times the cost of inference (forward pass) (Liu et al., 2022). We use M_1, M_2, M_3 to represent the *baseline cost* for one forward pass using a zero-shot model over one example in the encoder self-attention layers, decoder cross-attention layers and

[7] 4096 tokens = 16 examples × 256 tokens/example. We are able to train 3B models when reducing k to 4, and we include the result in Table 7 for completeness. Our conclusion on performance and efficiency remains the same.

	Zero-shot	Concat.	FiD	Ensemble	Simple FT
Complexity Analysis					
Pre-Inference	0	0	kM_1	0	$> 3kN(M_1 + M_2 + M_3)$
Inference (Encoder Self Attn)	M_1	$(k+1)^2 M_1$	M_1	$4kM_1$	M_1
Inference (Decoder Cross Attn)	M_2	$(k+1)M_2$	$(k+1)M_2$	$2kM_2$	M_2
Inference (Decoder Self Attn)	M_3	M_3	M_3	kM_3	M_3
Run Time: RTE (277 test examples)					
Pre-Inference (time; sec)	-	-	0.2	-	151.2
Inference (speed; #examples/sec)	46.2	2.7	24.0	1.8	46.2
Pre-Inference + Inference (time)	1x	17x	2x	26x	26x
Run Time: StoryCloze (1871 test examples)					
Pre-Inference (time; sec)	-	-	0.1	-	126.0
Inference (speed; #examples/sec)	72.6	2.7	28.1	2.5	72.6
Pre-Inference + Inference (time)	1x	27x	3x	29x	6x
Performance (Meta-Test-11 Avg.)					
Performance (Large)*	52.4	53.2	55.2	54.5	56.6
Performance (XL)*	51.0	N/A	60.0	57.7	61.4

Table 2: **Computation Cost Comparison.** $M_1/M_2/M_3$ stands for the unit computation costs used for one forward pass over one example. N is the number of epochs over the k shots during fine-tuning. See §5.2 for assumptions and details. Run time is measured when evaluating large-size (800M) models. *We list the performance of the better model between T5-LM and T0 initialization.

decoder self-attention layers, respectively. Computation costs of other methods will be represented in multipliers of M_1, M_2, M_3.

We summarize our estimation in the top section of Table 2. **(1)** We use "pre-inference cost" in the table to represent the one-time costs. For FiD-ICL, this refers to the cost of pre-computing the representations of examples using the encoder. For few-shot FT, this refers to the cost of applying gradient-based optimization. FiD-ICL has a significantly smaller pre-inference cost compared to few-shot FT. **(2)** In terms of inference cost, FiD-ICL is more efficient than the other two fusion methods in all the layers that we list. It uses kM_2 more computation in the decoder cross attention layers compared to a zero-shot or fine-tuned model.

Inference Speed. Additionally, we select two tasks (RTE and StoryCloze) in the meta-test set and measure the run time. For few-shot FT, we optimize the model for 300 updates, which is the recommended value in T-Few (Liu et al., 2022). In Table 2, we show that FiD-ICL is up to 10x faster than the other two fusion methods. Moreover, FiD-ICL, while achieving competitive performance, is faster than few-shot FT when pre-inference and inference time are combined.

Note that inference speed comparison above is dependent on number of test examples. In practice, when the test set is larger, the pre-inference cost will be amortized and FT will become faster when pre-inference cost and inference cost are summed. The break-even point for FiD-ICL and FT appears at 3.4k test instances for RTE and 5.6k test instances for Story Cloze. Therefore we believe FiD-ICL is most useful when the test set is small or when fast prototyping is needed.

6 Analysis

In this section we evaluate our ICL models in various scenarios, in hope to better understand their behavior and limitations. In §6.1 we evaluate the models to perform ICL with varying number of shots, when they were originally meta-trained to do 16-shot ICL. In §6.2 we study the influence to performance when the in-context examples are perturbed. In §6.3 we try to understand where ICL methods lie among other recent advances by comparing the performance of different model families.

6.1 Evaluate with Varying Number of Shots

Performance. One advantage of fusion-in-decoder models is that they may be trained with a small number of passages (*e.g.*, 5 passages), but evaluated with a larger number of passages (*e.g.*, 100 passages) (Izacard and Grave, 2021). For few-shot learning, this enables *flexibility* in the number of shots used. We conduct a similar analysis by changing the number of shots available at meta-test time. All our models are originally meta-trained to perform 16-shot in-context learning, and here we evaluate them with $\{2, 4, 8, 16, 32\}$ shots. Results

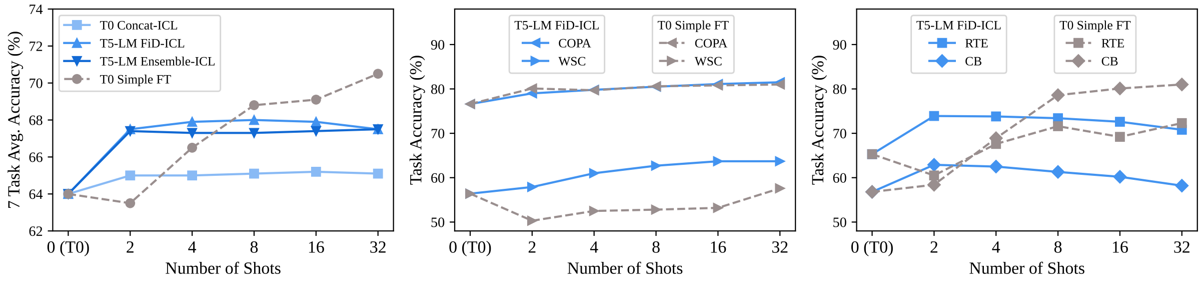

Figure 3: **Performance with varying number of shots at meta-test time.** Large size models (800M) are evaluated on Meta-Test-7. **Left:** Average performance of ICL methods does not improve significantly with more shots. **Middle:** When FiD-ICL is used, performance gradually improves when more shots are available for COPA and WSC. **Right:** When FiD-ICL is used, performance drops when more shots are available for RTE and CB.

	0-shot	2-shot			4-shot			8-shot			16-shot			16-shot
	ZS	Concat.	FiD	Ens.	Concat.	FiD	Ens.	Concat.	FiD	Ens.	Concat.	FiD	Ens.	FT
Run Time: RTE (277 test examples)														
Pre-Inference (time; sec)	-	-	<0.1	-	-	<0.1	-	-	0.1	-	-	0.2	-	151.2
Inference (speed; #examples/sec)	46.2	20.6	39.2	7.87	13.2	36.1	4.6	4.2	31.4	2.3	2.8	24.0	1.8	46.2
Pre-Inference + Inference (time)	1x	2.2x	1.2x	5.9x	3.5x	1.3x	10.1x	11.0x	1.5x	20.0x	16.9x	2.0x	26.1x	26.2x
Run Time: StoryCloze (1871 test examples)														
Pre-Inference (time; sec)	-	-	<0.1	-	-	<0.1	-	-	<0.1	-	-	0.1	-	126.0
Inference (speed; #examples/sec)	72.6	23.5	59.8	19.1	14.1	51.5	9.78	4.6	40.2	4.9	2.7	28.1	2.5	72.6
Pre-Inference + Inference (time)	1x	3.1x	1.2x	3.8x	5.1x	1.4x	7.4x	15.7x	1.8x	14.8x	27.2x	2.6x	29.0x	5.9x

Table 3: **Run time (pre-inference + inference) comparison when $\mathbf{k = 2, 4, 8, 16}$.** FiD-ICL has substantial efficiency benefits at inference even when k is small.

are visualized in Fig. 3 and reported in Table 9.

While the performance of fine-tuning method consistently increases when more shots become available, the performance of in-context learning methods is less sensitive to the number of shots. We further look at per-task performance and find two distinctive patterns: (1) On COPA and WSC, the performance gradually improves with more shots, as expected. Interestingly, FiD-ICL outperforms simple fine-tuning on WSC, suggesting that FiD-ICL is somehow "good at" learning WSC in particular. (2) On NLI tasks such as RTE and CB, performance surprisingly drops with more shots. These two patterns together lead to the unchanging performance on average (in Fig. 3 Left).

These observations suggest that ICL may be more suitable to certain task types than others. This may be relevant to the intrinsic task hardness (Zhao et al., 2022) or the difference between inductive biases exhibited by ICL and FT methods (Chan et al., 2022). One relevant observation is that on RTE, GPT-3 few-shot performance is not always better than zero-shot or one-shot performance (Brown et al. 2020, Appendix H), suggesting that RTE may have some unique characteristics. We leave further investigation as future work.

Inference Speed. Previously in §5.2, our run time analysis has been limited to the case of $k = 16$. As shown in the Complexity Analysis section in Table 2, efficiency is dependent on the number of in context example k, and the efficiency benefit of FiD is more significant with larger k. To provide a full picture of the efficiency benefits of FiD-ICL with smaller k, we report the run time when $k = 2, 4, 8$ in Table 3. We observe that FiD-ICL is constantly faster than Concat-ICL and Ensemble-ICL.

6.2 Perturbation to In-Context Examples

Min et al. (2022c) show that ICL models are rather insensitive to perturbations in in-context examples. Even with 100% wrong labels, little performance drop is observed with ICL models.[8] This is unexpected as the performance of fine-tuning would be drastically worse when labels are incorrect.

To investigate whether the fusion methods we use in this work help resolve these issues, we conduct a similar ablation study. We compare the performance of the following: **(1) No Perturbation**; **(2) No Input**, remove the inputs but keep the la-

[8] A more recent work (Wei et al., 2023) suggest that extremely large LMs can override semantic priors when these perturbations are applied.

bels; **(3) Random Label**, randomly select one of the valid options as the output; **(4) Wrong Label**, randomly select one of the wrong options; **(5) No Label**, remove the labels but keep the inputs. We examine both large-size (800M) and XL-size (3B) models, selecting the better model between T5-LM and T0 initialization.[9] We visualize the results in Fig. 4.

As expected, we observe a clear trend of No Perturbation > Random Label > Wrong Label for the T0-FT method. For ICL methods, performance drops in most cases when No Label perturbation is applied, suggesting that the presence of labels is essential. FiD-ICL suffers from No Label perturbation more than other two methods, suggesting that it may be capturing more information from the labels. However, performance does not change significantly with Random Label or Wrong Label perturbation, suggesting that FiD-ICL also struggle to learn from input-label mapping, despite their improved performance over Concat-ICL. Enabling ICL models to learn effectively and faithfully from examples remains a challenging problem.

6.3 Comparing with Other Model Families

Previously, we limit our scope to encoder-decoder models meta-trained to perform in-context learning. It is also necessary to have contextualized understanding by referencing and comparing with performance of other model families. We plot performance of various models in Fig. 5 and Fig. 7. We hope this can explain *where FiD-ICL lies* among other recent advances, and partly disentangle factors such as model architecture, training procedure.

Meta-trained vs. Not Meta-trained. In Fig. 5(a) we show the performance of T5-LM models that do not go through any meta-training. We show that in our problem setting, meta-training is crucial for the model to acquire the capability of zero-shot learning or few-shot in-context learning. Surprisingly, T5-LM models demonstrate little zero-shot or few-shot in-context learning capabilities on our meta-test tasks. We further try to quantify the effect of model architecture (encoder-decoder vs. decoder-only) and prompts used (P3 prompts or GPT-3 prompts), which we visualize in Fig. 7 and discuss in §A.2.

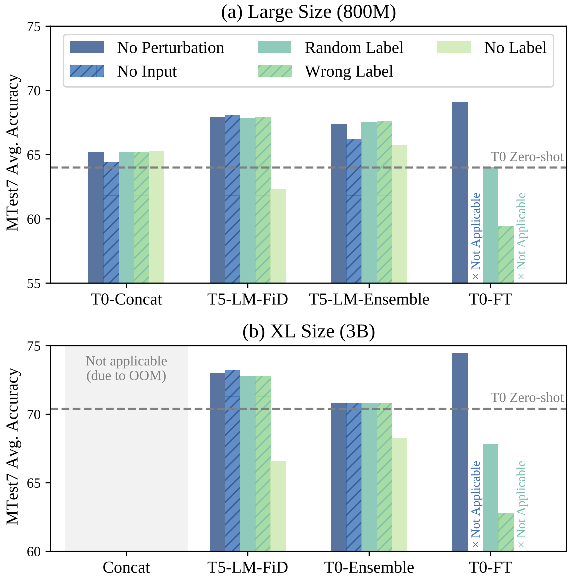

Figure 4: **Performance when Perturbing In-Context Examples.** Large models (800M) and XL models (3B) are evaluated on Meta-Test-7. **Observation:** All ICL methods are still rather insensitive to perturbations. FiD-ICL suffers from No Label perturbation more than the other two methods.

vs. GPT-3 models. We quote the GPT-3 performance (Brown et al., 2020) on Meta-Test-7 tasks in Table 13 and visualize them in Fig. 5(b). Note that the performance is not directly comparable as the number of shots vary from 20 to 70 for GPT-3 models, while our experiments are using 16 shots.[10] We would like to highlight that meta-trained encoder-decoder models outperforms off-the-shelf decoder models by a large margin, which aligns with the findings in Sanh et al. (2022); Wang et al. (2022a). Further, few-shot ICL models improves on top of zero-shot methods.

7 Conclusion

Motivated by the train-test efficiency differences between few-shot in-context learning and few-shot fine-tuning, we aim to find a balance and benefit from the strengths of both approaches. Towards this goal, we introduce FiD-ICL, a fusion-in-decoder approach for ICL, inspired by fusion-in-decoder models for open-domain QA (Izacard and Grave, 2021). With extensive experiments, we show that fusion-in-decoder ICL (intermediate fusion) is more favorable compared to concatenation-

[9] Note that all four perturbations can be applied to ICL models, but only (3)(4) can be applied to FT-based method.

[10] This comparison is less fair due to differences in model architecture, pre-training procedure, and prompts used. Yet, we think GPT-3 performance provide a reasonable reference.

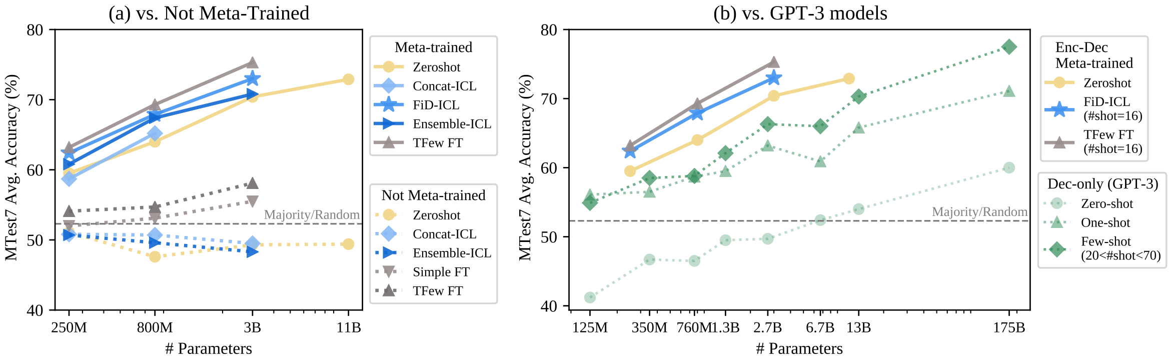

Figure 5: **Performance Comparison with (a) Not-Meta-Trained Models and (b) GPT-3 Models.** (a) Meta-training is crucial for the model to acquire zero-shot and few-shot ICL capabilities. (b) Meta-trained encoder-decoder models outperforms off-the-shelf decoder-only models by a large margin, consistent with findings in Sanh et al. (2022); Wang et al. (2022a).

based ICL (early fusion) and ensemble-based ICL (late fusion), in terms of *both* performance and computation efficiency. Moreover, fusion-in-decoder ICL partly closes the gap between gradient-free ICL methods and gradient-based fine-tuning methods, highlighting the potential of approximating gradient-based optimization with efficient forward-only methods (Phang et al., 2022). Future work may build upon our insights to further improve the computation efficiency of few-shot learning. However, similar to the findings in Min et al. (2022c), our analysis on ICL models suggest that they barely learn the input-label mapping from the in-context examples. We also have mixed results when more shots become available for the ICL model. We hope future work can further improve the performance of ICL by enabling it to learn from input-label mapping effectively and faithfully.

Limitations

Firstly, following the work of T0 (Sanh et al., 2022), we mainly focus on NLP tasks that can be formulated as rank classification. This covers classification and multiple-choice tasks, but not other task categories such as generation or regression. We hope to extend our training and evaluation to encompass a wider range of task categories, and hope the research community will collaborate in creating resources for such study.

Secondly, though we showed that FiD-ICL outperforms Concat-ICL, we still lack clear understanding on the source of such improvement. We hypothesized that FiD enables the model to learn from in-context examples more effectively, yet our perturbation experiments show that FiD-ICL models still learn little from input-label mapping (§6.2). Much more work is needed to further understand of the working mechanism of ICL models.

Thirdly, given the complexity of our study, we limit the scope to encoder-decoder models. We made this decision due to the superior performance of encoder-decoder models in task-level generalization (Wang et al., 2022a) and their compatibility with fusion-in-decoder method. Also, our important baselines, T0 (Sanh et al., 2022) and T-Few (Liu et al., 2022), are implemented with the T5 model family. We include more discussion in §A.4.

Acknowledgements

We thank Hamish Ivison, Zhaofeng Wu, Ian Magnusson, Dirk Groeneveld for insightful discussions. In addition, we thank anonymous reviewers, members of AllenNLP, USC INK Lab, the USC NLP community for their valuable feedback.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*.

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*, volume 6, pages 6–4. Venice.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI:

Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

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 *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Stephanie CY Chan, Ishita Dasgupta, Junkyung Kim, Dharshan Kumaran, Andrew K Lampinen, and Felix Hill. 2022. Transformers generalize differently from information stored in context vs in weights. *ArXiv preprint*, abs/2210.05675.

Danqi Chen and Wen-tau Yih. 2020. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37, Online. Association for Computational Linguistics.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *ArXiv preprint*, abs/2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *ArXiv preprint*, abs/2210.11416.

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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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 preprint*, abs/1803.05457.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models secretly perform gradient descent as meta optimizers. *ArXiv preprint*, abs/2212.10559.

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 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

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

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational*

Linguistics, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

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*, pages 1–9, Prague. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

David Ha, Andrew M. Dai, and Quoc V. Le. 2017. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. Structured prompting: Scaling in-context learning to 1,000 examples. *ArXiv preprint*, abs/2212.06713.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Hamish Ivison, Akshita Bhagia, Yizhong Wang, Hannaneh Hajishirzi, and Matthew Peters. 2022. Hint: Hypernetwork instruction tuning for efficient zero-shot generalisation. *arXiv preprint arXiv:2212.10315*.

Hamish Ivison and Matthew Peters. 2022. Hyperdecoders: Instance-specific decoders for multi-task NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1715–1730, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576, Online. Association for Computational Linguistics.

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 *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A dataset for question answering via sentence composition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8082–8090. AAAI Press.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of* *the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

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

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Bill Yuchen Lin, Kangmin Tan, Chris Scott Miller, Beiwen Tian, and Xiang Ren. 2022. Unsupervised cross-task generalization via retrieval augmentation. In *Advances in Neural Information Processing Systems*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

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*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

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*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 165–172. ACM.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022b. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022c. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. 2022. Hypertuning: Toward adapting large language models without back-propagation. *ArXiv preprint*, abs/2211.12485.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2022. Parallel context windows improve in-context learning of large language models. *ArXiv preprint*, abs/2212.10947.

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

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8722–8731. AAAI Press.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. 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 Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

Jürgen Schmidhuber. 1987. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. Ph.D. thesis, Technische Universität München.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cesar Ferri, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Christopher Waites, Christian Voigt, Christopher D Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, C. Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-Lopez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Francis Anthony Shevlin, Hinrich Schuetze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh Dhole, Kevin Gimpel, Kevin Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng

He, Luis Oliveros-Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje Ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramirez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael Andrew Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Andrew Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Russ Salakhutdinov, Ryan Andrew Chi, Seungjae Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel Stern Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Shammie Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven Piantadosi, Stuart Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, vinay uday prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

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. *Transactions of the Association for Computational Linguistics*, 7:217–231.

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 *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7063–7071. AAAI Press.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019b. QuaRTz: An open-domain dataset of qualitative relationship questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5941–5946, Hong Kong, China. Association for Computational Linguistics.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6076–6085, Hong Kong, China. Association for Computational Linguistics.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2022. Transformers learn in-context by gradient descent. *ArXiv preprint*, abs/2212.07677.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022a. What language model architecture and pretraining objective work best for zero-shot generalization? *ArXiv preprint*, abs/2204.05832.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

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*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1361–1375, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qinyuan Ye and Xiang Ren. 2021. Learning to generate task-specific adapters from task description. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 646–653, Online. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *ArXiv preprint*, abs/1810.12885.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015a. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* *Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinran Zhao, Shikhar Murty, and Christopher D Manning. 2022. On measuring the intrinsic few-shot hardness of datasets. *ArXiv preprint*, abs/2211.09113.

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 the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878, Punta Cana, Dominican Republic. Association for Computational Linguistics.

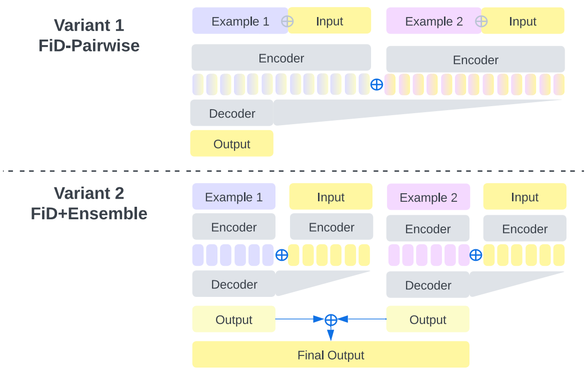

Figure 6: **Illustration of two altenative fusion methods: FiD-Pairwise and FiD+Ensemble.** See §A.1 for discussion.

A Additional Experiments and Findings

A.1 Alternative Fusion Methods

In addition to the three methods we investigate in the main paper, we experimented with two other methods, which we refer to as FiD-Pairwise and FiD+Ensemble. We illustrate these two methods in Fig. 6. FiD-Pairwise is closer to the original fusion-in-decoder model for open-domain QA. In FiD-Pairwise, the input is appended to *each* example individually. FiD+Ensemble is a hybrid model that employs the "fusion" operations in both FiD and Ensemble-based ICL. It applies fusion-in-decoder with one example and the input. Then it aggregates the predictions similar to ensemble-based ICL.

We conduct experiments with these two methods with base-size model (250M). We report the performance in Table 8. FiD (with T0 initialization) remains the best method among all compared fusion methods. Given the fact that FiD-Pairwise and FiD+Ensemble are less efficient than the FiD we use in the main paper, we stop investigating them in larger model scales.

A.2 Influence of Using P3 Prompted Data

In Fig. 7(a) we report performance on P3 held-out tasks using P3 prompted data and T5-LM models. Surprisingly we the performance does not grow with model scale. Also it does not grow when more shots become available. For all models the performance is close to majority/random baseline. We were skeptical about these results so we further evaluate public GPT models with the same P3 prompted data, which is reported in Fig. 7(b). The public GPT models include GPT-2 of various sizes, GPT-Neo-2.7B and GPT-J-6B. We still observe similar trends as in Fig. 7(a).

In Fig. 7(b), the main differences between the two groups were the prompt templates used. We hypothesize that P3 prompts may appear unnatural for GPT models and thus leads to the near random performance. We did some initial experiments with the COPA dataset using GPT-J-6B model, where using GPT-3 prompts yields zero-shot accuracy of 81%, but using P3 prompts gives accuracy ranging from 47% to 54%. We hope future work conducts rigorous comparisons about this.

A.3 T5-LM or T0 as initialization?

In Fig. 2, we observe that for large/XL models, T5-LM is a better initialization than T0 for ICL meta-training. Our hypothesis is that T0 training (*i.e.*, training T5-LM to become T0) may cause the model to forget general pre-train knowledge *or* lose the capabilities in modeling long context. This results in meta-training ICL being less effective. However this observation is dependant on model size. For base-size models, T0 is a better initialization.

A.4 Discussion on Enc-Dec vs. Dec-only Models

Prior work suggest that in a similar meta-learning setting, enc-dec models outperform dec-only models (Wang et al. 2022a, Sec 4.2). Another supporting evidence is that a 3B FLAN-T5 (enc-dec) model outperforms 175B OPT-IML (dec-only) on few-shot in-context learning (Longpre et al. 2023, Figure 1). Given the competitive performance of enc-dec models on the problem of interest, we focus on enc-dec models in this work.

To give our best efforts in make fair and comprehensive comparisons between enc-dec and dec-only models, we evaluated our meta-test tasks with publicly-available GPT models (Fig. 7, Table 12),

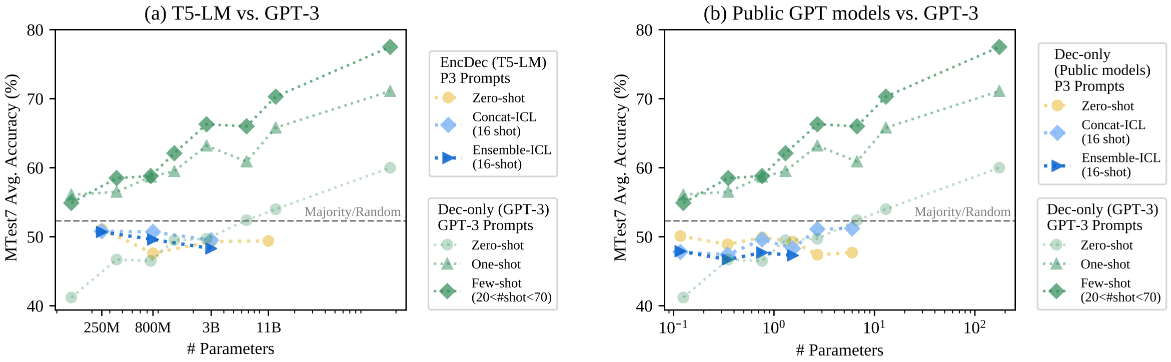

Figure 7: **Performance comparison when comparing model architectures and prompts.** In this figure, all models do not go through a meta-training phase. We find that P3 prompts are less effective (near random performance) when no meta-training is applied.

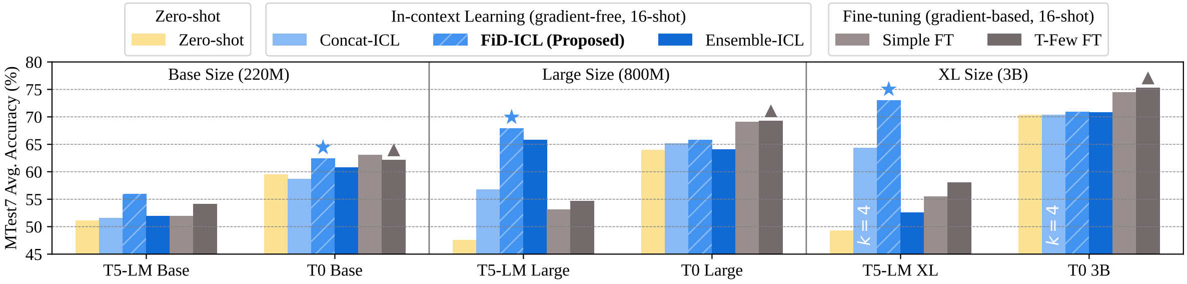

Figure 8: **Main Results on Meta-Test-7.** Differences with Fig. 2: (1) Reporting Meta-Test-7 instead of Meta-Test-11; (2) For Concat-ICL models with 3B size (T5-LM-XL, T0-3B), we reduce k to 4 to avoid OOM issue. Our conclusions in §5.1 remain the same.

quoted the performance on these tasks using GPT-3 models (Fig. 5, Table 13), and discussed our findings in §A.2.

One may argue that the computation for ICL using *decoder-only* models is also partly cache-able and thus the conclusions on computation efficiency may vary when decoder-only models are used. We agree with this argument. To account for this, we experimented with a key-value caching mechanism for a dec-only Concat-ICL model.[11] Caching enables a 3-5x speed-up, demonstrating that this is a promising direction. We hope a rigorous comparison between enc-dec and dec-only models regarding this matter can be done in future work.

B Data

B.1 Meta-Train and Meta-Test

We use the data and prompts provided in P3 (Sanh et al., 2022). We use the meta-train and meta-test partition used to train T0 (as opposed to the ones use for training T0+ or T0++). We provide the full list of these datasets and their reference in Table 4.

For meta-train, we use all the prompts associated with meta-train tasks. For meta-test, Sanh et al. (2022) provide the list of prompts for evaluation.[12] We want to point out a caveat here that this list is only a portion of all prompts associated with meta-test tasks. For example, `hellaswag_Topic_of_the_context` is a prompt name associated with the HellaSwag dataset, but it is not relevant to the original HellaSwag task, and should not used in evaluation.

We consider ANLI R1/R2/R3 as three separate tasks. Therefore the original meta-test in P3 has 11 tasks (Meta-Test-11).

B.2 Meta-Validation

In our preliminary experiments we observe that ICL methods may suffer from meta-overfitting: meta-test performance drops when the model is trained for more steps on meta-train. To ensure a fair eval-

[11] We used the catwalk library: https://github.com/allenai/catwalk/tree/prefix-caching

[12] https://github.com/bigscience-workshop/t-zero/blob/master/evaluation/template_list.py

uation set up, we additionally use 14 BIG-bench Task used in Sanh et al. (2022) as meta-validation (listed in Table 4). We use this meta-validation set for selecting intermediate checkpoints saved during meta-training. Apart from this, we do not tune any other hyper-parameters.

B.3 Few-shot Sampling

The performance of few-shot learning is highly subject to the sample of few-shot examples. To mitigate its influence in evaluation, our evaluations are based on 5 different few-shot samples. We first obtain the 5 samples of T0 held-out tasks used in Liu et al. (2022).[13] We then further sub-sample 16 examples to be our few-shot support set. We report the average of the 5 samples for all few-shot methods.

B.4 Data Sources

We obtain all our data from huggingface datasets (Lhoest et al., 2021). In the following we provide the links:

- P3 (meta-train/meta-test): `https://huggingface.co/datasets/bigscience/P3`
- BIG Bench (meta-validation): `https://huggingface.co/datasets/bigbench`

The full list of datasets and their citations are in Table 4.

C Training Details

C.1 Training T0-Base/Large/3B

Sanh et al. (2022) only provide model checkpoints in sizes of 3B and 11B. For a thorough investigation of different fusion methods, we aim to conduct experiments across different model sizes. Therefore, we replicate training procedure of T0-3B/T0-11B and train our own T0 models. We also largely reference the practice in Lin et al. (2022), in which the authors trains a BART0 model using BART-Large (Lewis et al., 2020a).

Specifically, we sub-sample at most 50k examples for each prompted task, following Lin et al. (2022). We combine all examples as a large dataset for multi-task learning, and do not apply any task sampling re-weighting technique. We list the key hyper-parameters in Table 5.

Sanh et al. (2022) reported an average of 51.0 on P3 held-out tasks. Our re-evaluation of the public checkpoint yields the same value of 51.0. Our T0-Base achieves 49.1, our T0-Large achieves 52.4, and our replication of T0-3B achieves 57.6. We hypothesize that the publicly released T0-3B may be under-trained, which corroborates with the findings in Lin et al. (2022) and Ivison et al. (2022).

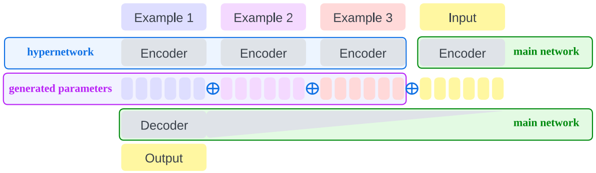

Figure 9: **A hypernetwork view of fusion-in-decoder.** The encoder is generating prefix parameters for the decoder.

C.2 ICL methods

Hyperparameters are listed in Table 6. Gradient checkpointing is enabled when training Concat-ICL-Large, FiD-ICL-3B and Ensemble-ICL-3B models.

C.3 Few-shot Fine-tuning

Hyperparameters are listed in Table 5.

C.4 Implementation

Our implementations are based on huggingface transformers (Wolf et al., 2020).

D Extended Related Work

Sparse Attention for In-Context Learning. Concurrent to our work, Ratner et al. (2022) proposed parallel context window (PCW) and Hao et al. (2022) proposed structured prompting for in-context learning. In a broader sense, these two works and our FiD-ICL can be viewed as applying sparse attention mask to the in-context examples. Ratner et al. (2022) and Hao et al. (2022) mainly focus on (1) applying such sparse masks to *off-the-shelf* decoder-only models and (2) incorporating more in-context examples than what one context window can typically fit. Our work differs in (1) problem settings, as we mainly compare different fusion methods in a meta-learning setting; (2) experiment settings, as we fix the number of shots available, and investigate the performance and efficiency of the models. Despite these differences, the shared intuitions and findings invite future research in adopting efficient architectures for improving different aspects of ICL.

[13] `https://github.com/r-three/t-few`

Fusion-in-decoder and Hypernetworks. In recent years, hypernetworks (Ha et al., 2017) are explored for various NLP problems (Ivison and Peters, 2022; Karimi Mahabadi et al., 2021), including zero-shot and few-shot task generalization (Ye and Ren, 2021; Phang et al., 2022). We believe the encoder in our fusion-in-decoder approach can be viewed as a hypernetwork. The encoder is effectively generating prefix parameters for the decoder, as demonstrated in Fig. 9. In Table 7 we compare with HyperT5 (Phang et al., 2022), a concurrent work that trains a hypernetwork to produce adaptation parameters.[14] Our fusion-in-decoder ICL is comparable with HyperT5.

Related to the concept of hypernetworks, recent work also suggest that in-context learning can be viewed as applying implicit optimization to the model itself (Akyürek et al., 2023; von Oswald et al., 2022; Dai et al., 2022).

E Extended Results

- Table 7 reports the per-task performance and average accuracy reported in Fig. 2.
- Table 9 includes the numbers in Fig. 3.
- Table 10 and Table 11 includes the numbers in Fig. 4.
- Table 13 includes the GPT-3 results quoted from the original paper (Brown et al., 2020). They were visualized in Fig. 5 and Fig. 7.
- Table 12 includes the numbers of our evaluation with GPT-style models. They were visualized in in Fig. 7.
- Table 14 includes the performance of not-meta-trained encoder-decoder models, also visualized in Fig. 5.

Dataset	Reference
Meta-Train (from P3)	
adversarial_qa dbert	Bartolo et al. (2020)
adversarial_qa dbidaf	Bartolo et al. (2020)
adversarial_qa droberta	Bartolo et al. (2020)
ag_news	Zhang et al. (2015a)
ai2_arc ARC-Challenge	Clark et al. (2018)
ai2_arc ARC-Easy	Clark et al. (2018)
amazon_polarity	McAuley and Leskovec (2013)
cnn_dailymail 3.0.0	See et al. (2017)
common_gen	Lin et al. (2020)
cos_e v1.11	Rajani et al. (2019)
cosmos_qa	Huang et al. (2019)
crows_pairs	Nangia et al. (2020)
dbpedia_14	Lehmann et al. (2015)
dream	Sun et al. (2019)
duorc ParaphraseRC	Saha et al. (2018)
duorc SelfRC	Saha et al. (2018)
gigaword	Graff et al. (2003)
glue mrpc	Dolan and Brockett (2005)
glue qqp	(link)
imdb	Maas et al. (2011)
kilt_tasks hotpotqa	Yang et al. (2018)
multi_news	Fabbri et al. (2019)
openbookqa main	Mihaylov et al. (2018)
paws labeled_final	Zhang et al. (2019)
piqa	Bisk et al. (2020)
qasc	Khot et al. (2020)
quail	Rogers et al. (2020)
quarel	Tafjord et al. (2019a)
quartz	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)
samsum	Gliwa et al. (2019)
sciq	Welbl et al. (2017)
squad_v2	Rajpurkar et al. (2016)
super_glue axg	Rudinger et al. (2018)
super_glue boolq	Clark et al. (2019)
super_glue multirc	Khashabi et al. (2018)
super_glue record	Zhang et al. (2018)
trec	Li and Roth (2002)
trivia_qa unfiltered	Joshi et al. (2017)
web_questions	Berant et al. (2013)
wiki_bio	Lebret et al. (2016)
wiki_hop original	Welbl et al. (2018)
wiki_qa	Yang et al. (2015)
wiqa	Tandon et al. (2019)
xsum	Narayan et al. (2018)
yelp_review_full	Zhang et al. (2015b); (link)
Meta-Validation (from BIG-bench, Srivastava et al. 2023)	
conceptual_combinations	code_line_description
hindu_knowledge	known_unknowns
language_identification	logic_grid_puzzle
logical_deduction	misconceptions
movie_dialog_same_or_different	novel_concepts
strategyqa	formal_fallacies_syllogisms_negation
vitaminc_fact_verification	winowhy
Meta-Test-11 (from P3; Meta-Test-7 marked with †)	
†hellaswag	Zellers et al. (2019)
†super_glue cb	De Marneffe et al. (2019)
†super_glue copa	Roemmele et al. (2011)
†super_glue rte	Dagan et al. (2005) Bar-Haim et al. (2006) Giampiccolo et al. (2007) Bentivogli et al. (2009)
†super_glue wic	Pilehvar and Camacho-Collados (2019)
†super_glue wsc.fixed	Levesque et al. (2012)
†story_cloze	Mostafazadeh et al. (2016)
anli (r1/r2/r3)	Nie et al. (2020)
winogrande winogrande_xl	Sakaguchi et al. (2020)

Table 4: Datasets used in this study: P3 and part of BIG-bench.

[14] Though the performance is not directly comparable (*e.g.*, the in-context examples used are different), we believe they provide reasonable references.

	T0-Base	T0-Large	Few-shot FT
Initialization	t5-base-lm-adapt	t5-large-lm-adapt	-
Max Input Len	1024	1024	256
Max Output Len	256	256	64
Optimizer	adafactor	adafactor	adafactor
Learning Rate	0.001	0.001	0.0003
# Training Steps	50000	50000	300
Batch Size	16	8	4
Gradient Accumulation	2	4	2
Effective Batch Size	32	32	8
Train Time	30 hours	60 hours	-

Table 5: Hyperparameters for Training T0-Base/Large and Hyperparameters for Few-shot Fine-tuning Experiments.

	Concat	FiD	Ensemble
Max Input Len	256	256	256
Max Output Len	64	64	64
Optimizer	adamw	adamw	adamw
Learning Rate	Base:5e-5; Large:1e-4; XL:1e-4		
# Training Steps	Base:50k; Large:50k; XL:10k		
# Warmup Steps	6% of total training steps		
Validation Interval	Base:10k; Large:5k; XL:2k		
k	16	16	1
m	1	16	1
Batch Size	4	1	16
Gradient Accumulation	2	8	2

Table 6: Hyperparameters for Training ICL Models. k/m represents the number of support/query examples in a forward pass.

Task	ANLI♦	(R1)	(R2)	(R3)	HSwag	CB	COPA	RTE	WiC	WSC	WGD	SCloze	MTest11 Avg.	MTest7 Avg.	HyperT5 Avg.
Majority / Random	33.4	33.4	33.4	33.4	25.0	50.0	50.0	52.7	50.0	63.5	50.0	50.0	44.7	52.3	46.8
Base (250M)															
T5-LM	33.4	33.3	33.5	33.5	24.7	44.3	54.3	47.9	49.7	57.9	49.8	54.1	43.9	51.1	45.2
T5-LM-Concat-ICL	33.3	33.0	33.4	33.3	25.6	45.1	55.0	48.7	50.2	55.9	48.8	57.5	44.2	51.6	45.3
T5-LM-FiD	33.0	32.4	33.1	33.4	26.7	42.5	58.8	54.6	51.1	57.9	50.3	76.3	47.0	55.9	46.9
T5-LM-Ensemble-ICL	32.6	31.5	34.0	32.4	25.8	44.5	56.5	47.7	50.2	56.4	49.4	62.6	44.6	52.5	45.4
T5-LM Simple Fine-tune	33.8	34.5	33.4	33.5	24.8	66.5	45.7	51.1	53.7	46.3	49.8	50.9	44.6	52.0	46.5
T5-LM T-Few Fine-tune	34.0	34.7	33.9	33.6	26.2	66.1	49.2	52.7	53.8	50.2	48.1	58.7	46.1	54.1	47.5
T0♥	32.3	31.5	32.4	33.1	26.5	45.8	65.9	69.3	51.6	56.7	51.2	76.1	49.1	59.5	49.9
T0-Concat-ICL	32.5	31.0	32.6	33.9	26.2	43.6	65.1	65.1	51.6	57.7	50.8	77.1	48.6	58.7	49.1
T0-FiD	32.7	31.7	32.9	33.6	26.2	54.9	68.2	68.1	51.9	60.3	51.3	82.3	51.0	62.4	51.7
T0-Ensemble-ICL	32.3	31.0	32.4	33.5	25.7	51.3	68.8	68.5	50.9	58.7	50.4	77.2	49.9	60.8	50.8
T0 Simple Fine-tune	33.5	32.6	33.9	33.9	29.1	73.2	66.3	68.0	53.1	50.9	51.0	79.0	51.9	63.1	53.1
T0 T-Few Fine-tune	33.1	30.5	35.1	33.6	32.2	73.6	59.8	64.3	51.9	50.6	54.2	81.3	51.6	62.2	52.5
Large (800M)															
T5-LM	32.7	32.1	33.4	32.7	25.3	33.8	50.5	49.0	51.0	50.4	50.5	47.8	41.5	47.6	42.9
T5-LM-Concat-ICL	33.4	33.0	33.9	33.3	25.7	49.7	63.4	47.3	50.0	63.4	51.1	73.0	47.6	56.8	48.0
T5-LM-FiD	34.4	33.9	33.4	35.8	28.3	60.2	81.1	72.6	50.7	63.7	55.6	91.6	55.2	67.9	55.8
T5-LM-Ensemble-ICL	33.5	32.2	33.1	35.3	27.0	62.1	77.5	77.9	50.9	61.0	55.0	87.5	54.5	67.4	55.6
T5-LM Simple Fine-tune	34.1	35.1	33.6	33.6	26.1	65.4	47.1	51.7	53.5	47.5	49.9	56.5	45.5	53.1	46.9
T5-LM T-Few Fine-tune	34.3	34.6	34.1	34.1	30.3	65.4	49.6	51.6	52.4	50.4	49.2	64.3	46.9	54.7	47.9
T0♥	34.1	32.2	34.2	36.0	26.1	56.8	76.6	65.3	50.8	56.4	53.9	88.4	52.4	64.0	52.5
T0-Concat-ICL	33.7	32.1	33.2	35.9	27.0	58.4	80.1	65.2	50.9	60.6	52.2	89.2	53.2	65.2	53.5
T0-FiD	33.4	31.8	32.8	35.7	26.1	60.7	77.6	67.1	52.1	59.1	54.7	89.5	53.4	65.8	53.9
T0-Ensemble-ICL	34.4	32.8	34.0	36.5	26.6	62.3	79.0	65.6	51.4	59.6	53.6	89.3	53.7	65.8	54.1
T0 Simple Fine-tune	35.3	34.5	35.4	36.2	33.1	80.1	80.8	69.2	54.1	53.2	56.3	90.0	56.6	69.1	57.8
T0 T-Few Fine-tune	35.2	33.2	37.3	34.9	36.6	79.6	79.0	69.5	53.9	56.4	56.2	90.6	57.0	69.3	58.3
HyperT5-Prefix$^{\beta}$	33.4	-	-	-	32.3	60.1	73.9	71.5	51.1	63.0	51.1	-	-	-	54.6
HyperT5-LoRA$^{\beta}$	33.6	-	-	-	33.0	49.5	74.2	67.4	52.0	64.0	52.9	-	-	-	53.3
XL (3B)															
T5-LM	32.7	32.2	33.4	32.7	24.6	32.7	53.1	48.8	50.8	57.6	50.9	51.4	42.6	49.3	43.9
T5-LM-Concat-ICL	OOM														
T5-LM-Concat-ICL (k=4)	-	-	-	-	-	56.3	83.2	65.2	50.3	54.9	54.6	86.4	-	64.4	-
T5-LM-FiD	39.3	39.8	37.6	40.4	31.4	67.0	92.3	78.8	50.4	64.5	61.2	96.5	60.0	73.0	60.6
T5-LM-Ensemble-ICL	34.1	33.9	33.8	34.6	27.2	51.8	89.5	51.2	50.2	58.9	53.8	93.3	52.6	64.1	52.1
T5-LM Simple Fine-tune	34.6	35.5	34.3	33.9	27.1	67.8	54.8	50.7	53.7	47.7	50.7	63.3	47.2	55.5	48.4
T5-LM T-Few Fine-tune	35.5	37.2	35.4	33.8	37.1	79.3	62.0	48.7	52.3	51.4	45.4	67.9	50.0	58.1	51.5
T0$^{\alpha}$	33.4	33.8	33.1	33.3	27.2	45.4	73.1	64.6	50.7	65.1	51.0	84.0	51.0	62.0	51.3
T0-Concat-ICL	OOM														
T0-FiD	37.8	39.1	36.7	37.6	30.0	61.2	90.8	71.6	51.8	63.1	59.6	96.0	58.0	70.6	58.2
T0-Ensemble-ICL	36.9	38.1	36.0	36.6	28.7	54.5	86.2	76.0	54.1	57.4	56.2	94.1	56.2	68.4	56.3
T0 Simple Fine-tune	37.1	39.3	36.6	35.4	35.1	75.0	75.8	72.8	53.2	55.6	52.1	88.0	56.3	67.5	57.1
T0 T-Few Fine-tune	40.1	42.4	40.7	37.1	51.9	81.8	84.6	71.7	55.1	57.2	57.5	93.5	61.2	71.6	62.5
T0♥	38.0	38.4	35.7	40.0	26.5	67.7	82.2	80.1	53.5	57.3	57.8	94.0	57.6	70.4	57.9
T0-Concat-ICL	OOM														
T0-Concat-ICL (k=4)	-	-	-	-	-	62.7	86.4	78.9	51.3	63.2	56.5	93.5	-	70.4	-
T0-FiD	38.6	39.0	36.5	40.5	28.5	62.9	87.4	74.6	52.1	62.7	61.0	95.5	58.2	70.9	58.5
T0-Ensemble-ICL	37.3	37.2	35.8	39.0	27.1	63.4	87.6	76.2	51.6	65.1	56.8	95.0	57.7	70.8	58.2
T0 Simple Fine-tune	38.5	37.5	38.8	39.2	38.7	81.9	88.0	80.1	55.9	59.5	61.4	95.0	61.4	74.5	63.0
T0 T-Few Fine-tune	40.2	41.2	40.0	39.5	44.9	82.1	88.4	81.3	56.9	64.1	59.6	94.8	63.0	75.3	64.7
HyperT5-Prefix$^{\beta}$	38.7	-	-	-	33.6	69.6	88.4	79.5	53.1	57.6	56.6	-	-	-	59.6
HyperT5-LoRA$^{\beta}$	35.3	-	-	-	30.8	66.4	83.3	68.5	50.3	60.0	56.1	-	-	-	56.4
XXL (11B)															
T5-LM	33.5	33.0	33.8	33.8	27.0	33.9	55.0	53.0	50.3	54.1	51.2	48.2	43.0	49.4	44.8
T0$^{\alpha}$	41.0	43.2	38.7	41.3	33.6	70.1	90.0	81.0	56.1	61.1	59.9	92.4	60.7	72.9	61.6

Table 7: **Main Results.** All few-shot methods are using 16 shots. “-” means not reported. ♥Trained by us. See §C.1 for details. ♦Computed as the average of R1/R2/R3 (except for HyperT5 rows where the numbers are quoted). $^{\alpha}$Sanh et al. (2022) $^{\beta}$Phang et al. (2022)

	CB	COPA	RTE	WiC	WSC	WGD	SCloze	MTest7 Avg.
Base (250M)								
T5-LM	44.3	54.3	47.9	49.7	57.9	49.8	54.1	51.1
T5-LM FiD	42.5	58.8	54.6	51.1	57.9	50.3	76.3	55.9
T5-LM FiD-Pairwise	54.0	60.4	65.9	51.1	54.0	51.1	77.2	59.1
T5-LM FiD+Ensemble	48.5	65.0	65.9	52.3	58.6	51.5	79.5	60.2
T0	45.8	65.9	69.3	51.6	56.7	51.2	76.1	59.5
T0 FiD	54.9	68.2	68.1	51.9	60.3	51.3	82.3	62.4
T0 FiD-Pairwise	46.1	68.7	70.4	51.9	61.5	50.4	79.8	61.3
T0 FiD+Ensemble	51.1	69.5	67.9	51.7	60.8	50.7	78.4	61.4

Table 8: **Performance using two alternative fusion methods: FiD-Pairwise and FiD+Ensemble.** Base-size (250M) models are trained evaluated.

	CB	COPA	RTE	WiC	WSC	WGD	SCloze	MTest7 Avg.
0-shot								
T0	56.8	76.6	65.3	50.8	56.4	53.9	88.4	64.0
2-shot								
T0-Concat-ICL	57.5	80.5	64.8	51.2	59.8	52.5	88.5	65.0
T5-LM-FiD	62.9	79.0	73.9	50.9	57.9	55.9	91.6	67.5
T5-LM-Ensemble-ICL	62.3	77.2	77.7	50.9	61.5	54.8	87.4	67.4
T0 Simple Fine-tune	58.4	80.1	60.5	51.9	50.3	54.4	89.0	63.5
4-shot								
T0-Concat-ICL	57.1	80.2	64.6	51.1	60.7	52.5	88.9	65.0
T5-LM-FiD	62.5	79.8	73.8	50.8	61.0	55.6	91.7	67.9
T5-LM-Ensemble-ICL	61.7	77.2	77.9	50.9	60.8	54.8	87.5	67.3
T0 Simple Fine-tune	68.9	79.7	67.6	52.4	52.5	55.1	89.3	66.5
8-shot								
T0-Concat-ICL	57.8	80.1	64.9	50.8	60.8	52.3	89.1	65.1
T5-LM-FiD	61.3	80.5	73.4	50.8	62.7	55.6	91.7	68.0
T5-LM-Ensemble-ICL	62.1	77.2	78.0	51.0	60.7	54.9	87.5	67.3
T0 Simple Fine-tune	78.6	80.6	71.6	52.2	52.8	55.6	89.7	68.8
16-shot								
T0-Concat-ICL	58.4	80.1	65.2	50.9	60.6	52.2	89.2	65.2
T5-LM-FiD	60.2	81.1	72.6	50.7	63.7	55.6	91.6	67.9
T5-LM-Ensemble-ICL	62.1	77.5	77.9	50.9	61.0	55.0	87.5	67.4
T0 Simple Fine-tune	80.1	80.8	69.2	54.1	53.2	56.3	90.0	69.1
32-shot								
T0-Concat-ICL	58.7	78.7	65.5	50.9	60.3	52.3	89.3	65.1
T5-LM-FiD	58.2	81.5	70.8	50.6	63.7	56.0	91.5	67.5
T5-LM-Ensemble-ICL	62.1	77.3	78.0	51.0	61.4	55.0	87.5	67.5
T0 Simple Fine-tune	81.0	81.0	72.3	55.1	57.6	56.3	90.2	70.5

Table 9: **Performance when using varying number of shots at meta-test time.** Large (800M) models trained to perform ICL with 16 shots are evaluated.

	CB	COPA	RTE	WiC	WSC	WGD	SCloze	MTest7 Avg.
Zero-shot Baselines (Large/800M)								
T5-LM	33.8	50.5	49.0	51.0	50.4	50.5	47.8	47.6
T0$^\heartsuit$	56.8	76.6	65.3	50.8	56.4	53.9	88.4	64.0
T0-ICL (Large/800M)								
No Perturbation	58.4	80.1	65.2	50.9	60.6	52.2	89.2	65.2
Random Label	58.4	80.2	65.2	50.9	60.5	52.2	89.2	65.2
Wrong Label	58.4	80.2	65.2	50.9	60.6	52.2	89.2	65.2
No Label	58.7	80.2	65.1	50.8	60.9	52.2	89.1	65.3
No Input	56.6	79.5	65.0	51.2	58.7	51.7	88.5	64.4
T5-LM-FiD (Large/800M)								
No Perturbation	60.2	81.1	72.6	50.7	63.7	55.6	91.6	67.9
Random Label	59.6	81.0	72.7	50.7	63.1	55.6	91.6	67.8
Wrong Label	59.6	81.0	72.9	50.7	64.2	55.6	91.6	67.9
No Label	45.7	81.2	66.0	52.5	43.8	55.5	91.8	62.3
No Input	64.4	79.2	74.7	51.0	59.7	55.9	91.5	68.1
T5-LM-Ensemble (Large/800M)								
No Perturbation	62.1	77.5	77.9	50.9	61.0	55.0	87.5	67.4
Random Label	63.0	77.5	77.9	50.9	61.0	55.0	87.5	67.5
Wrong Label	63.1	77.4	77.9	51.0	61.2	55.0	87.5	67.6
No Label	59.2	75.0	78.6	50.8	56.8	53.4	86.3	65.7
No Input	61.4	75.5	76.3	50.7	60.0	53.9	86.0	66.2
T0 Simple Fine-tune (Large/800M)								
No Perturbation	80.1	80.8	69.2	54.1	53.2	56.3	90.0	69.1
Random Label	48.6	79.9	68.1	52.1	52.8	56.3	90.0	64.0
Wrong Label	24.3	76.8	64.9	50.5	53.3	56.3	90.0	59.4
No Label	Not Applicable							
No Input	Not Applicable							

Table 10: **Performance with perturbation to in-context examples at meta-test time.** Large size (800M) models are compared.

	CB	COPA	RTE	WiC	WSC	WGD	SCloze	MTest7 Avg.
Zero-shot Baselines (XL/3B)								
T5-LM	32.7	53.1	48.8	50.8	57.6	50.9	51.4	49.3
T0$^\alpha$	45.4	73.1	64.6	50.7	65.1	51.0	84.0	62.0
T0$^\heartsuit$	67.7	82.2	80.1	53.5	57.3	57.8	94.0	70.4
T5-LM-FiD (XL/3B)								
No Perturbation	67.0	92.3	78.8	50.4	64.5	61.2	96.5	73.0
Random Label	65.5	92.2	78.8	50.5	64.7	61.2	96.5	72.8
Wrong Label	65.5	92.2	78.9	50.5	64.5	61.2	96.5	72.8
No Label	55.8	92.2	68.5	52.0	38.8	62.6	96.3	66.6
No Input	71.1	92.8	79.5	51.0	59.4	62.5	96.2	73.2
T0-Ensemble (XL/3B)								
No Perturbation	63.4	87.6	76.2	51.6	65.1	56.8	95.0	70.8
Random Label	63.3	87.6	76.2	51.6	65.2	56.8	94.8	70.8
Wrong Label	63.3	87.6	76.2	51.6	65.2	56.8	94.8	70.8
No Label	59.4	83.5	69.3	50.8	65.6	54.8	94.7	68.3
No Input	64.0	86.0	78.2	51.3	65.1	56.3	94.7	70.8
T0 Simple Fine-tune (XL/3B)								
No Perturbation	38.7	81.9	88.0	80.1	55.9	59.5	61.4	74.5
Random Label	46.7	84.0	75.7	53.7	57.9	61.4	94.9	67.8
Wrong Label	23.2	79.8	71.1	50.7	58.1	61.4	95.0	62.8
No Label	Not Applicable							
No Input	Not Applicable							

Table 11: **Performance with perturbation to in-context examples at meta-test time.** XL size (3B) models are compared.

	CB	COPA	RTE	WiC	WSC	WGD	SCloze	MTest7 Avg.
Majority / Random	50.0	50.0	52.7	50.0	63.5	50.0	50.0	52.3
GPT2-Small (117M)								
Zero-shot	37.3	56.0	48.1	51.6	54.1	49.6	54.3	50.1
Concat-ICL	42.2	46.8	51.8	51.1	42.2	49.1	51.4	47.8
Ensemble-ICL	41.3	46.8	51.2	50.6	42.0	50.2	53.4	47.9
GPT2-Medium (345M)								
Zero-shot	30.5	54.4	47.9	52.2	54.1	49.7	53.2	48.9
Concat-ICL	33.6	48.6	48.2	51.5	54.2	50.2	45.7	47.4
Ensemble-ICL	32.8	45.1	48.4	51.0	51.4	49.7	48.6	46.7
GPT2-Large (762M)								
Zero-shot	36.1	55.5	47.5	50.9	56.1	49.7	53.6	49.9
Concat-ICL	43.6	50.4	48.5	51.0	54.0	49.8	50.2	49.6
Ensemble-ICL	34.5	46.7	47.8	50.5	52.1	49.8	52.2	47.7
GPT2-XL (1542M)								
Zero-shot	34.9	52.5	47.2	51.2	55.7	49.4	54.2	49.3
Concat-ICL	30.8	50.8	47.6	50.5	55.3	49.6	53.6	48.3
Ensemble-ICL	31.9	51.4	48.5	50.7	46.0	48.8	53.5	47.3
GPT-Neo (2.7B)								
Zero-shot	25.8	55.9	47.7	51.6	48.1	48.9	53.9	47.4
Concat-ICL	46.6	56.6	54.2	50.7	48.9	50.2	50.6	51.1
GPT-J (6B)								
Zero-shot	24.8	55.1	50.3	52.1	48.7	49.1	53.6	47.7
Concat-ICL	45.7	57.4	54.6	52.6	45.6	49.7	53.2	51.2

Table 12: **Performance using public decoder-only models (without meta-training).** We evaluate these public checkpoints using P3 formatted data. For all ICL methods, 16 shots are used.

	CB	COPA	RTE	WiC	WSC	WGD	SCloze	MTest7 Avg.
Majority / Random	50.0	50.0	52.7	50.0	63.5	50.0	50.0	52.3
GPT-3 (Small/125M)								
Zero-shot	0.0	66.0	47.7	0.0	59.6	52.0	63.3	41.2
One-shot	55.4	62.0	53.1	50.0	58.7	51.3	62.3	56.1
Few-shot*	42.9	67.0	52.3	49.8	58.7	51.3	62.3	54.9
GPT-3 (Medium/350M)								
Zero-shot	32.1	68.0	49.8	0.0	56.7	52.1	68.5	46.7
One-shot	53.6	64.0	47.3	50.3	58.7	53.0	68.7	56.5
Few-shot*	58.9	64.0	48.4	55.0	60.6	52.6	70.2	58.5
GPT-3 (Large/760M)								
Zero-shot	8.9	73.0	48.4	0.0	65.4	57.4	72.4	46.5
One-shot	53.6	66.0	49.5	50.3	60.6	58.3	72.3	58.7
Few-shot*	53.6	72.0	46.9	53.0	54.8	57.5	73.9	58.8
GPT-3 (XL/1.3B)								
Zero-shot	19.6	77.0	56.0	0.0	61.5	58.7	73.4	49.5
One-shot	48.2	74.0	49.5	49.2	62.5	59.1	74.2	59.5
Few-shot*	69.6	77.0	50.9	53.0	49	59.1	76.1	62.1
GPT-3 (2.7B)								
Zero-shot	19.6	76.0	46.6	0.0	66.3	62.3	77.2	49.7
One-shot	57.1	76.0	54.9	49.4	66.3	61.7	77.3	63.2
Few-shot*	67.9	83.0	56.3	51.6	62.5	62.6	80.2	66.3
GPT-3 (6.7B)								
Zero-shot	28.6	80.0	55.2	0.0	60.6	64.5	77.7	52.4
One-shot	33.9	82.0	54.9	50.3	60.6	65.8	78.7	60.9
Few-shot*	60.7	83.0	49.5	53.1	67.3	67.4	81.2	66.0
GPT-3 (13B)								
Zero-shot	19.6	84.0	62.8	0.0	64.4	67.9	79.5	54.0
One-shot	55.4	86.0	56.3	50.0	66.3	66.9	79.7	65.8
Few-shot*	66.1	86.0	60.6	51.1	75.0	70.0	83.0	70.3
GPT-3 (175B)								
Zero-shot	46.4	91.0	63.5	0.0	65.4	70.2	83.2	60.0
One-shot	64.3	87.0	70.4	48.6	69.2	73.2	84.7	71.1
Few-shot*	82.1	92.0	72.9	55.3	75.0	77.7	87.7	77.5

Table 13: **Performance of GPT-3 models (without meta-training).** Numbers are quoted from Brown et al. (2020). *In the GPT-3 paper the number of shots is task-specific and vary from 20 to 70.

	CB	COPA	RTE	WiC	WSC	WGD	SCloze	MTest7 Avg.
Majority / Random	50.0	50.0	52.7	50.0	63.5	50.0	50.0	52.3
T5-Base-LM-Adapt (250M)								
Zero-shot	44.3	54.3	47.9	49.7	57.9	49.8	54.1	51.1
Concat-ICL	45.1	49.9	47.3	50.0	58.0	49.4	56.2	50.8
Ensemble-ICL	38.0	52.0	47.7	50.1	63.1	50.2	53.6	50.7
T5-Large-LM-Adapt (800M)								
Zero-shot	33.8	50.5	49.0	51.0	50.4	50.5	47.8	47.6
Concat-ICL	43.9	54.5	47.6	50.0	58.0	49.7	50.9	50.7
Ensemble-ICL	42.5	51.0	47.2	49.9	52.6	49.9	54.4	49.6
T5-XL-LM-Adapt (3B)								
Zero-shot	32.7	53.1	48.9	50.8	57.6	51.0	51.4	49.3
Concat-ICL	40.3	55.5	48.1	50.1	50.6	49.6	52.3	49.5
Ensemble-ICL	43.2	48.9	52.3	50.2	40.4	50.0	53.0	48.3
T5-XXL-LM-Adapt (11B)								
Zero-shot	34.3	54.9	53.0	50.3	54.1	50.7	48.2	49.4

Table 14: **Performance using encode-decoder models for ICL (without meta-training).** As opposed to results in Table 7, models in this tables are evaluated directly and do not go through a meta-training phase.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitation section after conclusion.

☑ A2. Did you discuss any potential risks of your work?
Limitation section after conclusion.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
Section 4, Appendix B

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix B

C ☑ Did you run computational experiments?

Left blank.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Table 3 and Table 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Table 3 and Table 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix C

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.