The Flan Collection: Designing Data and Methods for Effective Instruction Tuning

Shayne Longpre [1] **Le Hou** [2] **Tu Vu** [2] **Albert Webson** [2] **Hyung Won Chung** [2] **Yi Tay** [2] **Denny Zhou** [2] **Quoc V. Le** [2] **Barret Zoph** [2] **Jason Wei** [2] **Adam Roberts** [2]

Abstract

We study the design decisions of publicly available instruction tuning methods, by reproducing and breaking down the development of Flan 2022 (Chung et al., 2022). Through careful ablation studies on the Flan Collection *of tasks and methods*, we tease apart the effect of design decisions which enable Flan-T5 to outperform prior work by 3-17%+ across evaluation settings. We find task balancing and enrichment techniques are overlooked but critical to effective instruction tuning, and in particular, training with mixed prompt settings (zero-shot, few-shot, chain-of-thought) actually yields equivalent or stronger (2%+) performance in *all* settings. In further experiments, we show Flan-T5 requires less finetuning to converge higher and faster than T5 on single downstream tasks—motivating instruction-tuned models as more computationally-efficient starting checkpoints for new tasks. Finally, to accelerate research on instruction tuning, we make the Flan 2022 collection of datasets, templates, and methods publicly available.[1]

1. Introduction

Large language models such as PaLM (Chowdhery et al., 2022), Chinchilla (Hoffmann et al., 2022), and ChatGPT among others (Brown et al., 2020; Ouyang et al., 2022) have unlocked new capabilities in performing natural language processing (NLP) tasks from reading instructive prompts. Prior art has shown that instruction tuning—finetuning language models on a collection of NLP tasks formatted with instructions—further enhances the ability of language models to perform an unseen task from an instruction (Wei et al., 2021; Sanh et al., 2021; Min et al., 2022).

In this work, we evaluate the methods and results of *open sourced* instruction generalization efforts, comparing their finetuning techniques and methods. And in particular, we identify and evaluate the critical methodological improvements in the "Flan 2022 Collection", which is the term we use for the collection *of data and the methods that apply to the data and instruction tuning process*, first introduced in Chung et al. (2022). Where Chung et al. (2022) focuses on the emergent and state-of-the-art results of combining Flan 2022 with PaLM 540B, this work focuses in on the details of the instruction tuning methods themselves, ablating individual factors, and comparing them directly to prior work by keeping the pretrained model size and checkpoint consistent.

The Flan 2022 Collection offers the most extensive publicly available set of tasks and methods for instruction tuning, which we have compiled in one place, and supplemented with hundreds more high-quality templates and richer formatting patterns. We show that a model trained on this collection outperforms other public collections on all tested evaluation benchmarks, including the original Flan 2021 (Wei et al., 2021), T0++ (Sanh et al., 2021), Super-Natural Instructions (Wang et al., 2022c), and the concurrent work on OPT-IML (Iyer et al., 2022). As shown in Figure 1, this includes a 4.2%+ and 8.5% improvements on the MMLU (Hendrycks et al., 2020) and BIG-Bench Hard (Suzgun et al., 2022) evaluation benchmarks, for equally sized models.

Analysis of the Flan 2022 method suggests the strong results stem both from the larger and more diverse set of tasks, but also from a set of simple finetuning and data augmentation techniques. In particular, training on a mix of examples templatized with zero-shot, few-shot, and chain-of-thought prompts improves or maintains performance in every one of

[1]Media Lab, Massachusetts Institute of Technology, Cambridge, USA [2]Google, Mountain View, USA. Correspondence to: Shayne Longpre <slongpre@media.mit.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1]Data generation code available at: https://github.com/google-research/FLAN/tree/main/flan/v2. Generation code allows users to vary mixtures rates, templates, prompt types and data augmentations techniques, for faster public research.

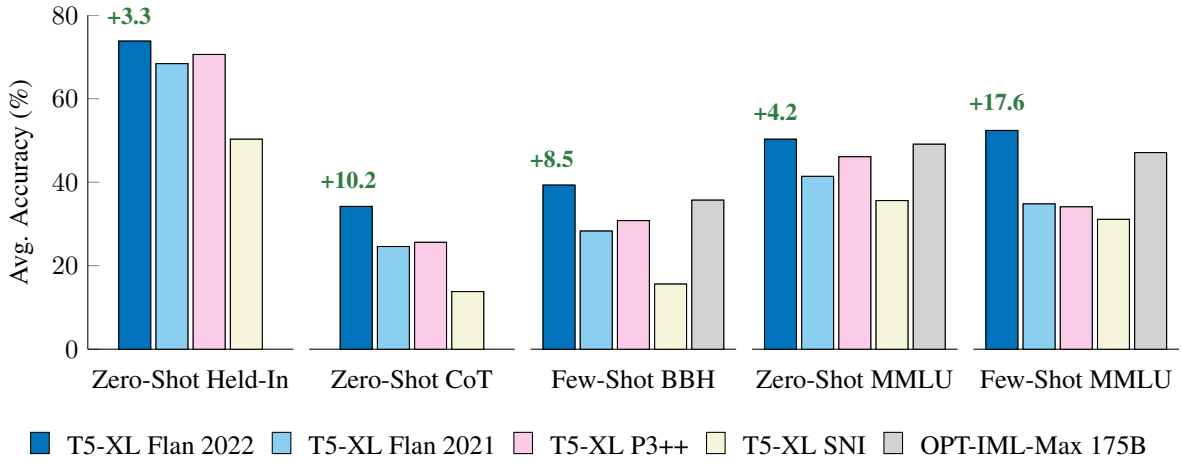

Figure 1: **Comparing public instruction tuning collections** on Held-In, Held-Out (BIG-Bench Hard (Suzgun et al., 2022) and MMLU (Hendrycks et al., 2020)), and Chain-of-Thought evaluation suites, detailed in **??**. All models except OPT-IML-Max (175B) are T5-XL with 3B parameters. Green text indicates improvement over the next best comparable T5-XL (3B) model.

these settings, together. For instance, adding just 10% few-shot prompts improves zero-shot prompting results by 2%+. Additionally, enriching task diversity by inverting input-output pairs, as used in (Sanh et al., 2021; Min et al., 2022), along with balancing task sources, are both shown to be critical to performance. The resulting Flan-T5 model converges faster and at a higher performance than T5 models in single-task finetuning—suggesting instruction-tuned models offer a more computationally-efficient starting checkpoint for downstream applications, corroborating Aribandi et al. (2021); Liu et al. (2022b).

We hope making these findings and resources publicly available will unify resources around instruction tuning and accelerate research into more general-purpose language models. We summarize this work's core contributions as follows:

- Methodological: Show that training with mixed zero- and few-shot prompts yields much better performance in **both** settings (Section 3.2).
- Methodological: Measure and demonstrate the critical techniques to effective instruction tuning: scaling Section 3.3, enriching task variety with input inversion (Section 3.4), adding chain-of-thought training data, and balancing different data sources (Section 3.5).
- Results: Demonstrate these technical choices yield 3-17% Held-Out task improvements over existing open source instruction tuning collections (Figure 1).
- Results: Demonstrate Flan-T5 XL serves as a stronger and more computationally-efficient starting checkpoint for single-task finetuning (Section 4).

2. Public Instruction Tuning Collections

Large Language Models Instruction tuning has emerged as a tool to "unlock" the knowledge and abilities of large language models (LLMs) learned at pretraining time, to make them more useful for interactive dialog and functional tasks. Previous work (Raffel et al., 2020; Liu et al., 2019; Aghajanyan et al., 2021; Aribandi et al., 2021) experimented with large scale multi-task finetuning, to improve downstream single target finetuning, but without instruction prompts. UnifiedQA and others (Khashabi et al., 2020; McCann et al., 2018; Keskar et al., 2019) unified a wide range of NLP tasks into a single generative question answering format, using prompt instructions for multi-task finetuning and evaluation.

The First Wave Since 2020, several instruction tuning task collections have been released in rapid succession, outlined in Figure 2. Natural Instructions (Mishra et al., 2021), Flan 2021 (Wei et al., 2021), PromptSource (a.k.a. P3, Public Pool of Prompts, Bach et al., 2022) aggregated large NLP task collections and templatized them with instructions (*zero-shot prompting*), specifically for finetuning models to generalize to unseen instructions. MetaICL (Min et al., 2022) also consolidated other task collections (Ye et al., 2021; Khashabi et al., 2020) to train models to learn tasks "in-context" – from several input-output examples, known as *few-shot prompting*, but in this case without instructions. Each of these works affirmed the scaling benefits of task and template diversity, and some reported strong benefits from inverting the inputs and outputs in templates to produce new tasks ("noisy channel" in Min et al., 2022).

Release	Collection	Model Details: Model	Base	Size	Public?	Data Collection & Training Details: Prompt Types	Tasks in Flan	# Exs	Methods
2020 05	UnifiedQA	UnifiedQA	RoBerta	110-340M	P	ZS	46 / 46	750k	
2021 04	CrossFit	BART-CrossFit	BART	140M	NP	FS	115 / 159	71.M	
2021 04	Natural Inst v1.0	Gen. BART	BART	140M	NP	ZS / FS	61 / 61	620k	+ Detailed k-shot Prompts
2021 09	Flan 2021	Flan-LaMDA	LaMDA	137B	NP	ZS / FS	62 / 62	4.4M	+ Template Variety
2021 10	P3	T0, T0+, T0++	T5-LM	3-11B	P	ZS	62 / 62	12M	+ Template Variety + Input Inversion
2021 10	MetaICL	MetaICL	GPT-2	770M	P	FS	100 / 142	3.5M	+ Input Inversion + Noisy Channel Opt
2021 11	ExMix	ExT5	T5	220M-11B	NP	ZS	72 / 107	500k	+ With Pretraining
2022 04	Super-Natural Inst.	*Tk*-Instruct	T5-LM, mT5	11-13B	P	ZS / FS	1556 / 1613	5M	+ Detailed k-shot Prompts + Multilingual
2022 10	GLM	GLM-130B	GLM	130B	P	FS	65 / 77	12M	+ With Pretraining + Bilingual (en, zh-cn)
2022 11	xP3	BLOOMz, mT0	BLOOM, mT5	13-176B	P	ZS	53 / 71	81M	+ Massively Multilingual
2022 12	Unnatural Inst.†	T5-LM-Unnat. Inst.	T5-LM	11B	NP	ZS	~20 / 117	64k	+ Synthetic Data
2022 12	Self-Instruct†	GPT-3 Self Inst.	GPT-3	175B	NP	ZS	Unknown	82k	+ Synthetic Data + Knowledge Distillation
2022 12	OPT-IML Bench†	OPT-IML	OPT	30-175B	P	ZS + FS, CoT	~2067 / 2207	18M	+ Template Variety + Input Inversion + Multilingual
2022 10	Flan 2022 (ours)	Flan-T5, Flan-PaLM	T5-LM, PaLM	10M-540B	P NP	ZS + FS, CoT	1836	15M	+ Template Variety + Input Inversion + Multilingual

Figure 2: A **Timeline of Public Instruction Tuning Collections** specifies the collection release date, name, detailed information on the finetuned models (their name, the base model, their size, and whether the model itself is made Public (P) or Not Public (NP)), what prompt specification they were trained for (zero-shot, few-shot, or Chain-of-Thought), the number of tasks contained in the Flan 2022 Collection Flan 2022 (released with this workours), and core methodological contributions in each work. Note that the number of tasks and of examples vary under different assumptions and so are estapproximateions. For instance, the definition of "task" and "task category" vary by work, and are not easily simplified to one ontology. The reported cCounts for the number of taskseach are reported using task definitions from the respective works. † indicates concurrent work.

The Second Wave A second wave of instruction tuning collections expanded prior resources: combining more datasets and tasks into one resource, like Super-Natural Instructions (Wang et al., 2022c) or OPT-IML (Iyer et al., 2022), adding multilingual instruction tuning in xP3 (Muennighoff et al., 2022), and Chain-of-Thought training prompts in Flan 2022 (Chung et al., 2022). Both the Flan Collection and OPT-IML contain most tasks represented in prior collections.[2] Our work is positioned here, coalescing most of these collections (of collections) and their methods, as the strongest starting point for future open source work.

New Directions Concurrent and future work is beginning to explore two new directions: (a) expanding task diversity even more aggressively with synthetic data generation, particularly in creative, and open-ended dialogue (Wang et al., 2022b; Honovich et al., 2022; Ye et al., 2022; Gupta et al., 2022), and (b) offering human feedback signals on model responses (Ouyang et al., 2022; Glaese et al., 2022; Bai et al., 2022a; Nakano et al., 2021; Bai et al., 2022b). We view most of these new directions as likely additive to a foundation of instruction tuning methods.

Tuning with Human Feedback Instruction tuning on human feedback has demonstrated strong results on open-ended tasks, but at the expense of performance on a wide array of more traditional NLP tasks (Ouyang et al., 2022; Glaese et al., 2022; Bai et al., 2022a; Nakano et al., 2021). (See Ouyang et al. (2022)'s discussion of the "alignment tax".) Our work focuses specifically on instruction generalization, without human feedback, for two reasons. First, human feedback datasets are far less publicly available than instruction tuning datasets (and may be model specific). Second, by itself, instruction generalization shows great promise in enhancing human preferred responses on open-

[2]Each work defines datasets, tasks, and task categories differently. For simplicity, we use their own definitions in Section 2.

ended tasks, as well as improving traditional NLP metrics (Chung et al., 2022). The extent of obtainable progress *without* expensive human response demonstrations or ratings remains an open question, and an important pursuit to narrow the gap between public and non-public research.

The Importance of Open Source High profile research is increasingly driven by non-public data, as in the case of Gopher, PaLM, GPT-3 and others (Ouyang et al., 2022; Glaese et al., 2022; Rae et al., 2021; Chowdhery et al., 2022). The inaccessibility of these resources inhibits the research community's ability to analyze and improve these methods in the public domain. We narrow our purview to open source and accessible data collections, motivated by the goal of democratizing accessibility to research.

3. Flan 2022 Instruction Tuning Experiments

Recent research has yet to coalesce around a unified set of techniques, with different tasks, model sizes, and target input formats all represented. We open source a new collection, first introduced in Chung et al. (2022), we denot as " Flan 2022", which combines Flan 2021, P3++[3], Super-Natural Instructions, with some additional reasoning, dialog, and program synthesis datasets. We emulate Chung et al. (2022)'s description of templatization and collection; and in this work we analyse the key methodological improvements with detailed ablations, and compare the collection to other collections, using equivalently-sized models.

In this section, we evaluate the design decisions in Flan 2022 and discuss four in particular that yield strong improvements to the instruction tuning recipe. These design components, outlined in Section 2, are: **(I)** using mixed zero-shot, few-shot, and Chain-of-Thought templates at training (Section 3.2), **(II)** scaling T5-sized models to 1800+ tasks (Section 3.3), **(III)** enriching tasks with input inversion (Section 3.4), and **(IV)** balancing these task mixtures (Section 3.5). In Section 3.1, we begin by measuring the value of each component and compare the final model against alternative instruction tuning collections (and their methods).

Experimental Setup We finetune on the prefix language model adapted T5-LM (Lester et al., 2021), using the XL (3B) size for all models, unless otherwise stated, again following Chung et al. (2022). We evaluate on (a) a suite of 8 "Held-In" tasks represented within the 1800+ training task collection (4 question answering and 4 natural language inference validation sets), (b) Chain-of-Thought (CoT) tasks (5 validation sets), and (c) the MMLU (Hendrycks et al., 2020) and BBH (Suzgun et al., 2022) benchmarks as our set of "Held-Out" tasks, as they are not included as part of Flan 2022 finetuning. The Massivley Multitask Language Understanding benchmark (MMLU) broadly tests reasoning and knowledge capacity across 57 tasks in the sciences, social sciences, humanities, business, health, among other subjects. BIG-Bench Hard (BBH) includes 23 challenging tasks from BIG-Bench (Srivastava et al., 2022) where PaLM under-performs human raters. In our ablations, we also evaluate BBH with Chain-of-Thought inputs, following Chung et al. (2022). Additional finetuning and evaluation details are provided in Appendix B.

[3]"P3++" is our notation for all datasets used in the Public Pool of Prompts (P3): https://huggingface.co/datasets/bigscience/P3

3.1. Ablation Studies

Table 1 summarizes the mean contribution to Held-in, Held-out, and Chain-of-thought tasks, by individually deducting methods: mixture weight balancing ("- Mixture Balancing"), Chain-of-thought tasks ("- CoT"), mixed prompt settings ("- Few Shot Templates"), and Input Inversion ("- Input Inversion"). Flan-T5 XL leverages all four of these methods together. We also finetune T5-XL-LM on other collections, including Flan 2021, P3++, Super-Natural Instructions for comparison.

Each of the ablated components of Flan contributes improvements to different metrics: Chain-of-Thought training to Chain-of-Thought evaluation, input inversion to Held-Out evaluations (MMLU and BBH), few-shot prompt training to few-shot evaluations, and mixture balancing to all metrics.

As compared to T5-XL models trained on alternative instruction tuning collections (and their methods), Flan outperforms in almost every setting. While previous collections are tuned specifically to zero-shot prompts, Flan-T5 XL is tuned for either zero- or few-shot prompts. This yields performance margins of +3-10% for most of the zero-shot settings, and margins of 8-17% for the few-shot settings. Most impressively, Flan 2022 outperforms OPT-IML-Max's much larger (10x) 30B and (58x) 175B models, and GLM-130B (x43). Next, we isolate some of Flan 2022's ablated methods individually, to examine the benefits of each.

3.2. Training with Mixed Prompt Settings

Prior work has shown a wide variety of input templates per task can improve performance. However, separate from the wording of the instruction template, these prior LLMs mostly tune with template sets *targeted to a single prompt setting*: for zero-shot prompting (Wei et al., 2021; Sanh et al., 2021; Aghajanyan et al., 2021; Aribandi et al., 2021) or for few-shot prompting (Min et al., 2022; Wang et al., 2022c).

An underappreciated design decision in InstructGPT (Ouyang et al., 2022) was to mix training templates for

Table 1: **Method Ablations (top)** show the importance of each method for Flan-T5 XL. **Collection Ablations (bottom)** evaluate Flan-T5 XL against T5-XL finetuned on other instruction tuning collections: FLAN 2021, P3++, and Super-Natural Instructions. **Flan 2022 - Next Best T5-XL** shows the improvement of Flan-T5 XL over the next best T5-XL (comparatively sized) finetuned on another collection. Metrics are reported in both zero-shot / few-shot settings across Held-In, Chain-of-Thought, and Held-Out (MMLU, BBH) tasks.
† We also inlcude the results reported by OPT-IML (Iyer et al., 2022) and GLM-130B (Zeng et al., 2022).

MODEL	HELD-IN	COT	MMLU	BBH	BBH-COT
T5-XL Flan 2022	**73.8 / 74.8**	35.8 / **34.1**	**50.3 / 52.4**	26.2 / **39.3**	**33.9 / 35.2**
- CoT	73.3 / 73.2	28.8 / 24.6	47.5 / 46.9	18.2 / 30.0	18.2 / 12.0
- Input Inversion	**73.8** / 74.1	32.2 / 23.5	41.7 / 41.2	18.4 / 24.2	15.7 / 13.0
- Mixture Balancing	71.2 / 73.1	32.3 / 30.5	45.4 / 45.8	15.1 / 24.3	13.8 / 15.4
- Few Shot Templates	72.5 / 62.2	**38.9** / 28.6	47.3 / 38.7	27.6 / 30.8	18.6 / 23.3
T5-XL Flan 2021	68.4 / 56.3	24.6 / 22.7	41.4 / 34.8	**28.1** / 28.3	26.0 / 26.9
T5-XL P3++	70.5 / 62.8	25.6 / 25.6	46.1 / 34.1	26.0 / 30.8	23.4 / 26.1
T5-XL Super-Natural Inst.	50.3 / 42.2	13.8 / 14.3	35.6 / 31.1	10.4 / 15.6	8.0 / 12.5
GLM-130B†	-	-	– / 44.8	-	-
OPT-IML-Max 30B†	-	-	46.3 / 43.2	– / 30.9	-
OPT-IML-Max 175B†	-	-	49.1 / 47.1	– / 35.7	-
Flan 2022 - Next Best T5-XL	+3.3 / +12	+10.2 / +8.5	+4.2 / +17.6	-1.9 / +8.5	+7.9 / +8.3

each of these prompt settings, rather than target a single setting. However, since Ouyang et al. (2022) do not examine this choice, we expected a performance trade-off in finetuning for zero-shot or few-shot prompted performance – particularly for smaller models. Instead, we find training with mixed zero- and few-shot prompts significantly improves performance in **both** settings – most surprisingly, even for models with only 3B parameters.

Figure 3 shows (1) adding as little as 10% few-shot training templates can improve zero-shot performance by 2%, and (2) adding 10%+ of zero-shot data improves few-shot performance by 2-4%. Both Held-In and Held-Out tasks peak between 10-90% of few-shot data, but this range is consistently higher than training with only one prompt setting.

3.3. Scaling Small Models to 1.8k+ Tasks

The most recent and concurrent publicly available instruction tuning efforts, like Flan 2022, train on thousands of tasks (Wang et al., 2022c; Iyer et al., 2022), but operate on different task compositions and underlying training methods. To measure the impact of scaling model sizes and tasks for the Flan 2022 collection, and specifically on T5-sized models, we finetune T5-LM adapted models (Small, Base, Large, XL, XXL) on randomly selected task subsets (8, 25, 50, 100, 200, 400, 800, all 1873). Every finetuning run is guaranteed to include the Held-In tasks, so we can estimate how task scaling impacts the model capacity to maintain performance on a given task its already seen.

Figure 4 demonstrates that both Held-In and Held-Out tasks appear to benefit from adding hundreds of finetuning tasks. Held-in task evaluations peak around 200 total tasks, and diminish in performance as more tasks are added, though larger models peak later and diminish less. Held-out task performance increases log-linearly with the number of tasks, achieving the highest performances with all 1836 tasks. Surprisingly, only T5-Small appears to exceed its Held-Out task performance before 1836 tasks, while larger model sizes continue to improve. These results suggest (a) even T5-Base may not have exhausted its capacity with thousands of tasks, and (b) the largest LMs could benefit from thousands more tasks for Held-In and Held-Out task performance.

One necessary assumption of this analysis is that all tasks are defined and counted equally. Section 3.5 demonstrates how not all task sources are equally beneficial to training, and the model performance may saturate from too many tasks from one source (e.g. Super-Natural Instructions). We would caution conclusions that task scaling beyond 1800 would translate to increased returns without also paying attention to task heterogeneity and quality.

3.4. Task Enrichment with Input Inversion

Prior instruction tuning work has enriched their diversity of tasks by inverting the (x, y) input-output pairs in supervised tasks—referred to as "prompts not intended for the original task" in P3 (Bach et al., 2022) or the "noisy channel" in MetaICL (Min et al., 2022). For example, a dataset may be originally designed for, given a question x, evaluate if a model can answer y. Input inversion instead gives a model the answer y and trains it to generate the question x. This is an easy method to enrich the task variety given a limited set of data sources. However, it isn't clear from prior work

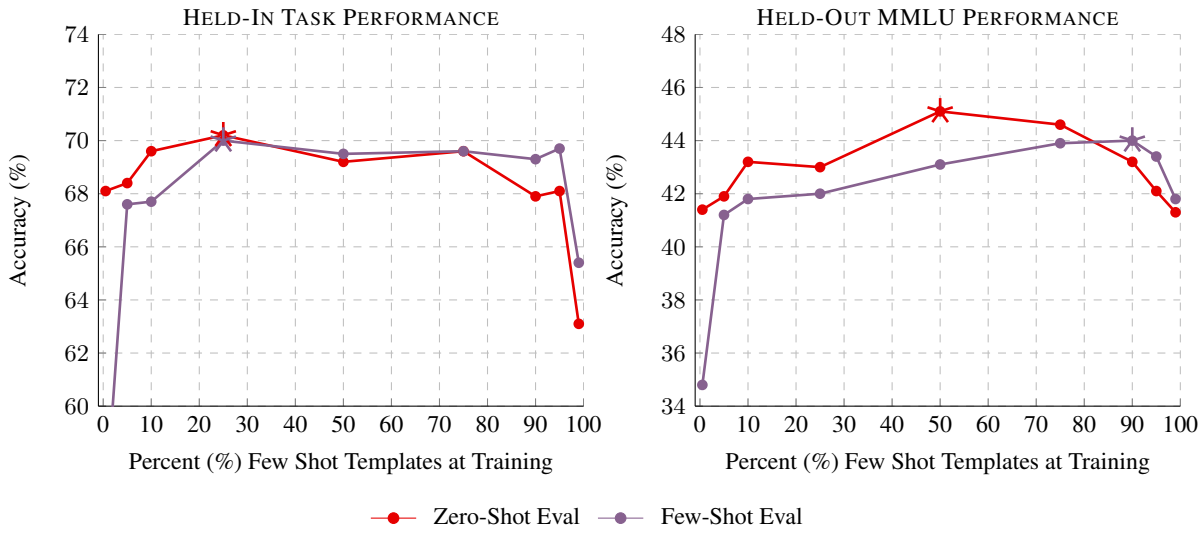

Figure 3: **Training jointly with zero-shot and few-shot prompt templates improves performance** on both Held-In and Held-Out tasks. The stars indicate the peak performance in each setting.

Table 2: Subsets of tasks are left out from an equally weighted mixture to measure their importance. **T0-SF and Flan 2021 finetuning are most important for MMLU, while Chain-of-Thought (CoT) finetuning is most important for Chain-of-Thought evaluation.**

Train Mixtures	Metrics		
	Held-In	CoT	MMLU
All (Equal)	64.9	41.4	47.3
All - T0-SF	63.2	**43.4**	44.7
All - Flan 2021	55.3	38.6	45.7
All - Super-Nat. Inst.	65.9	42.2	46.8
All - CoT	65.6	29.1	46.8
All - Prog. Synth.	66.9	42.3	46.8
All - Dialog	65.4	40.3	47.1
All (Weighted)	**66.4**	40.1	**48.1**

that this method remains helpful when 100s of unique data sources and 1000s of tasks are already available.

To assess this, we enrich our mixtures with input inverted tasks (details and examples in Appendix C) and measure the effect. In Table 1 we find this is not beneficial for Held-In performance, but strongly beneficial for Held-Out performance. These benefits invigorate the prospect of data augmentation techniques for LLM finetuning, which had previously been shown to have diminishing returns the longer models are pretrained (Longpre et al., 2020).

3.5. Balancing Data Sources

Scaling architecture size and the number of tasks are effective, but our results suggest the mixture weighting deserves as much attention to optimize results. To assess a balanced weighting, we omit different sets of task sources, one at a time (Flan 2021, T0-SF, Super-Natural Instructions, Chain-of-Thought, Dialog, and Program Synthesis), and rank their contributions on the MMLU benchmark.[4].

In Table 2 we sample from each of the 6 submixtures equally for "All (Equal)", then remove submixtures individually. We ran these experiment prior to other ablations, so it is trained only with zero-shot training templates.
As shown in Table 2, Flan 2021 and T0-SF are among the most beneficial mixtures, followed by Super-Natural Instructions and Chain-of-Thought, with Dialog and Program Synthesis last. These findings are corroborated by Iyer et al. (2022) who extensively test data mixing proportions, and also determine their Flan 2021, T0-SF, and T5 mixtures are the most broadly beneficial. Additionally, they find Super-Natural Instructions has limited scaling benefits on Held-Out task performance, which they relate to its unique input format and instruction design. Notably, Chain-of-thought finetuning appears beneficial across all our evaluation settings, especially considering they contain far fewer tasks than Flan 2021, T0-SF or Natural Instructions. For the final mixture we follow Chung et al. (2022), which seems to mirror our findings in Table 2.

[4]Following Chung et al. (2022) we refer to the subset of P3++ that is not in Flan 2021 as T0-SF (SF stands for "sans Flan").

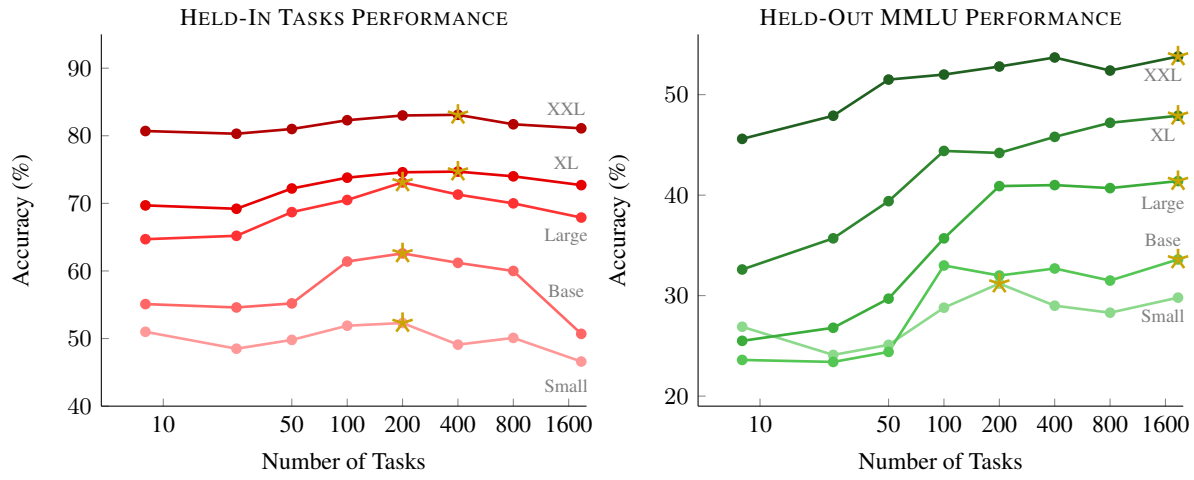

Figure 4: **Performance Scaling Laws for the number of finetuning tasks and model sizes**. Held-In performance (left) and Held-Out MMLU performance (right) are shown. The gold star indicates the peak performance for that model size.

3.6. Discussion

OPT-IML (Iyer et al., 2022) presents the closest comparison to the Flan Collection, including a similar collection of tasks, examples and techniques. However, while their used tasks are all publicly sourced, their collection, with templates, processing, and example mixing, is not released, and as a result cannot be easily compared. Iyer et al. (2022) report that Flan-T5-XL (3B) and XXL (11B) outperforms OPT-IML-Max 175B on both MMLU and BBH. As they discuss, these differences may arise from any combination of pre-training, model architecture, and instruction tuning. Model architecture and pretraining before instruction tuning can play a significant role (Wang et al., 2022a). But there are many other details in instruction tuning that may vary between Flan 2022 and OPT-IML. Likely candidates are: example templatization, how the mixed input prompting procedures are used at training, and task composition.

How significant are each of these difference? While OPT-IML contains more tasks than Flan 2022, we estimate approximately $94\%(2067/2207)$ are also used in the Flan 2022 collection[5], and very few tasks in Flan 2022 are not contained in some format in OPT-IML. This suggests the overall difference in task diversity is not significant when using a shared definition of "task". Task mixture rates also emphasize similar sources, including Flan 2021 (46% vs 20%), PromptSource/P3 (28% vs 45%), and Super-Natural Instructions (25% vs 25%), for Flan 2022 and OPT-IML respectively.[6] OPT-IML's other collections (Crossfit, ExMix, T5, U-SKG) are not weighted significantly: 4%, 2%, 2%, 2% respectively.

We believe example templatization and the mixed prompt formats may pose the largest differences with OPT-IMLs instruction tuning. Our template repository inherits from the source collections, but also significantly extends them, adding variety not just in instructions, but also other dimensions. For instance, the templatization procedure varies where the instruction is placed (before or after few-shot prompts), the spacing and separators between few-shot and Chain-of-Thought exemplars, and the formatting permutations of answer options (and their targets) for multiple-choice examples, which sometimes includes and sometimes excludes answer options in the inputs or exemplars. While we do not have dedicated experiments comparing many iterations of their development, we found these procedures dramatically augment input variety and showed repeated performance improvements. Our example templatizing procedure is open sourced for inspection and future work.

4. Instruction Tuning Enhances Single-Task Finetuning

In applied settings, machine learning practitioners deploy NLP models finetuned (FT) specifically for a single target task, usually where finetuning data is already available. While prior work has shown the benefits of intermediate finetuning (Pruksachatkun et al., 2020; Vu et al., 2020) or multi-task finetuning (Aghajanyan et al., 2021; Aribandi et al., 2021) for downstream tasks, this has not been studied extensively for instruction-tuned models. In this setting, we evaluate Flan 2022 instruction tuning as an intermediary step before single target finetuning, to understand if Flan-T5 would serve as a better starting checkpoint for applied prac-

[5]This is calculated using their definition of "task" (Iyer et al. (2022)'s Table 1), which does not deduplicate across collections.

[6]Note that 46% weight for Flan 2022 is actually on "Muffin" from Chung et al. (2022) which is similar to Flan 2021.

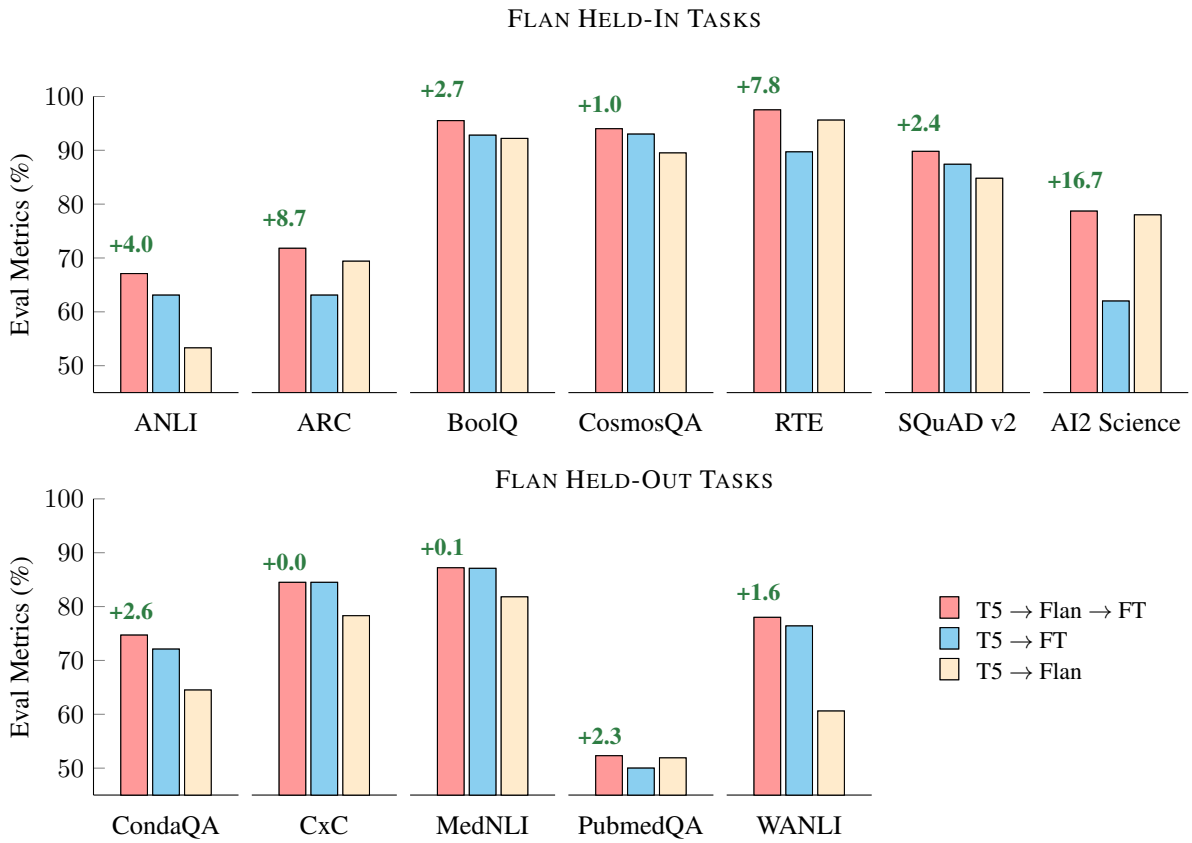

Figure 5: **Flan-T5 Outperforms T5 on Single-Task Finetuning.** We compare single-task finetuned T5, single-task finetuned Flan-T5, and Flan-T5 without any further finetuning.

titioners. We evaluate three settings in Figure 5: finetuning T5 directly on the target task as the conventional baseline, using Flan-T5 without further finetuning, and finetuning Flan-T5 further on the target task.

Pareto Improvements to Single Target Finetuning For both sets of Held-In and Held-Out tasks examined, finetuning Flan-T5 offers a pareto improvement over finetuning T5 directly. In some instances, usually where finetuning data is limited for a target task, Flan-T5 without further finetuning outperforms T5 with target task finetuning.

Faster Convergence Using Flan-T5 as a starting checkpoint has an added benefit in training efficiency. As demonstrated in Figure 6, Flan-T5 converges much more quickly than T5 during single target finetuning, as well as peaking at higher accuracies. These convergence results also suggest there are strong green-AI incentives for the NLP community to adopt instruction-tuned models, like Flan-T5 for single-task finetuning, rather than conventional non-instruction-tuned models. While pretraining and instruction tuning are more financially and environmentally expensive than single-task finetuning, they are a one-time cost. On the contrary, pretrained models that require extensive finetuning become more costly when aggregating over many millions of additional training steps (Wu et al., 2022; Bommasani et al., 2021). Instruction-tuned models offer a promising solution to significantly reduce the amount of finetuning steps across a wide swathe of tasks, if they are adopted as a new standard starting point for single-task finetuning.

5. Conclusions

The new Flan 2022 instruction tuning collection unifies some of the most popular prior public collections and their methods, while adding new templates and simple improvements like training with mixed prompt settings. The resulting collection outperforms Flan 2021, P3++, Super-Natural Instructions, GLM-130B, and OPT-IML-Max 175B on a wide vareity of Held-In and Held-Out benchmarks, often by large margins. Results suggest this new collection serves as a more competitive starting point for generalizing to new instructions, or finetuning on a single new task.

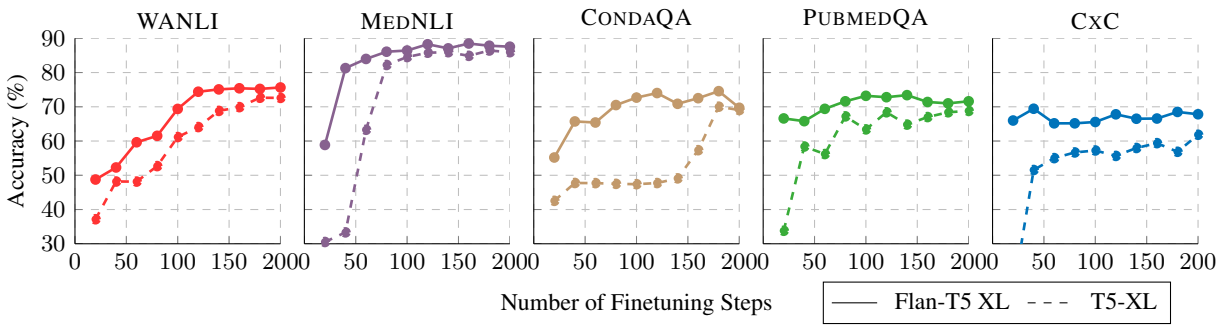

Figure 6: **Flan-T5 convergences faster than T5 on single-task finetuning** for each of 5 Held-Out tasks from Flan finetuning.

Acknowledgements

We would like to thank Ed H Chi, and Xinyun Chen, and Colin Raffel for their advice and feedback on the paper.

References

Aghajanyan, A., Gupta, A., Shrivastava, A., Chen, X., Zettlemoyer, L., and Gupta, S. Muppet: Massive multi-task representations with pre-finetuning. In *EMNLP*, 2021. URL https://aclanthology.org/2021.emnlp-main.468.

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Jauregui Ruano, R., Jeffrey, K., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M., and Zeng, A. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. *arXiv e-prints*, art. arXiv:2204.01691, April 2022.

Aribandi, V., Tay, Y., Schuster, T., Rao, J., Zheng, H. S., Mehta, S. V., Zhuang, H., Tran, V. Q., Bahri, D., Ni, J., et al. Ext5: Towards extreme multi-task scaling for transfer learning. *arXiv preprint arXiv:2111.10952*, 2021.

Bach, S., Sanh, V., Yong, Z. X., Webson, A., Raffel, C., Nayak, N. V., Sharma, A., Kim, T., Bari, M. S., Fevry, T., Alyafeai, Z., Dey, M., Santilli, A., Sun, Z., Ben-david, S., Xu, C., Chhablani, G., Wang, H., Fries, J., Al-shaibani, M., Sharma, S., Thakker, U., Almubarak, K., Tang, X., Radev, D., Jiang, M. T.-j., and Rush, A. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 93–104, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-demo.9. URL https://aclanthology.org/2022.acl-demo.9.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

Bentivogli, L., Clark, P., Dagan, I., and Giampiccolo, D. The fifth pascal recognizing textual entailment challenge. In *TAC*, 2009.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., et al. PaLM: Scaling language modeling with Pathways. *arXiv preprint arXiv:2204.02311*, 2022. URL https://arxiv.org/abs/2204.02311.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Dai, A. M. and Le, Q. V. Semi-supervised sequence learning. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf.

Devaraj, A., Sheffield, W., Wallace, B., and Li, J. J. Evaluating factuality in text simplification. In *Proceedings of*

the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 7331–7345, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.506. URL https://aclanthology.org/2022.acl-long.506.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019. URL https://aclanthology.org/N19-1423.

Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Gao, L., Dai, Z., Pasupat, P., Chen, A., Chaganty, A. T., Fan, Y., Zhao, V. Y., Lao, N., Lee, H., Juan, D.-C., et al. Attributed text generation via post-hoc research and revision. *arXiv preprint arXiv:2210.08726*, 2022.

Geva, M., Khashabi, D., Segal, E., Khot, T., Roth, D., and Berant, J. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9: 346–361, 2021.

Glaese, A., McAleese, N., Trębacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Gupta, P., Jiao, C., Yeh, Y.-T., Mehri, S., Eskenazi, M., and Bigham, J. P. Improving zero and few-shot generalization in dialogue through instruction tuning. *arXiv preprint arXiv:2205.12673*, 2022.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0RDcd5Axok.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *ICLR*, 2020. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. v. d., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Honovich, O., Scialom, T., Levy, O., and Schick, T. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*, 2022.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Huang, L., Le Bras, R., Bhagavatula, C., and Choi, Y. 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)*, pp. 2391–2401, 2019.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Brown, N., Jackson, T., Luu, L., Levine, S., Hausman, K., and Ichter, B. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022.

Iyer, S., Lin, X. V., Pasunuru, R., Mihaylov, T., Simig, D., Yu, P., Shuster, K., Wang, T., Liu, Q., Koura, P. S., Li, X., O'Horo, B., Pereyra, G., Wang, J., Dewan, C., Celikyilmaz, A., Zettlemoyer, L., and Stoyanov, V. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*, 2022. URL https://arxiv.org/abs/2212.12017.

Jin, Q., Dhingra, B., Liu, Z., Cohen, W., and Lu, X. PubMedQA: A dataset for biomedical research question answering. 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)*, pp. 2567–2577, 2019. URL https://aclanthology.org/D19-1259.

Keskar, N. S., McCann, B., Xiong, C., and Socher, R. Unifying question answering, text classification, and regression via span extraction. *arXiv preprint arXiv:1904.09286*, 2019.

Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., and Hajishirzi, H. UnifiedQA: Crossing format boundaries with a single QA system. In *Findings of*

the Association for Computational Linguistics: EMNLP 2020, 2020. URL https://aclanthology.org/2020.findings-emnlp.171.

Laurençon, H., Saulnier, L., Wang, T., Akiki, C., del Moral, A. V., Le Scao, T., Von Werra, L., Mou, C., Ponferrada, E. G., Nguyen, H., et al. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Le Scao, T., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *EMNLP*, 2021. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. 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*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

Liang, P. P., Wu, C., Morency, L.-P., and Salakhutdinov, R. Towards understanding and mitigating social biases in language models. In *ICML*, 2021.

Liu, A., Swayamdipta, S., Smith, N. A., and Choi, Y. Wanli: Worker and ai collaboration for natural language inference dataset creation. *arXiv preprint arXiv:2201.05955*, 2022a. URL https://arxiv.org/abs/2201.05955.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, 2022b. URL https://arxiv.org/abs/2205.05638.

Liu, X., He, P., Chen, W., and Gao, J. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4487–4496, 2019.

Longpre, S., Wang, Y., and DuBois, C. How effective is task-agnostic data augmentation for pretrained transformers? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4401–4411, 2020.

Longpre, S., Perisetla, K., Chen, A., Ramesh, N., DuBois, C., and Singh, S. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7052–7063, 2021.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.173. URL https://aclanthology.org/2020.acl-main.173.

McCann, B., Keskar, N. S., Xiong, C., and Socher, R. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

McGuffie, K. and Newhouse, A. The radicalization risks of gpt-3 and advanced neural language models. *arXiv preprint arXiv:2009.06807*, 2020.

Miao, S.-Y., Liang, C.-C., and Su, K.-Y. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, 2020.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Min, S., Lewis, M., Zettlemoyer, L., and Hajishirzi, H. MetaICL: Learning to learn in context. In *NAACL*, 2022. URL https://aclanthology.org/2022.naacl-main.201.

Mishra, S., Khashabi, D., Baral, C., and Hajishirzi, H. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*, 2021.

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Scao, T. L., Bari, M. S., Shen, S., Yong, Z.-X., Schoelkopf, H., et al. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*, 2022.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, 2020.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022. URL https://arxiv.org/abs/2203.02155.

Parekh, Z., Baldridge, J., Cer, D., Waters, A., and Yang, Y. Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for MS-COCO. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 2855–2870, 2021. URL https://aclanthology.org/2021.eacl-main.249.

Patel, A., Bhattamishra, S., and Goyal, N. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, 2021.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *NAACL*, 2018. URL https://aclanthology.org/N18-1202.

Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., and Bowman, S. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5231–5247, 2020.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., Rutherford, E., Hennigan, T., Menick, J., Cassirer, A., Powell, R., Driessche, G. v. d., Hendricks, L. A., Rauh, M., Huang, P.-S., Glaese, A., Welbl, J., Dathathri, S., Huang, S., Uesato, J., Mellor, J., Higgins, I., Creswell, A., McAleese, N., Wu, A., Elsen, E., Jayakumar, S., Buchatskaya, E., Budden, D., Sutherland, E., Simonyan, K., Paganini, M., Sifre, L., Martens, L., Li, X. L., Kuncoro, A., Nematzadeh, A., Gribovskaya, E., Donato, D., Lazaridou, A., Mensch, A., Lespiau, J.-B., Tsimpoukelli, M., Grigorev, N., Fritz, D., Sottiaux, T., Pajarskas, M., Pohlen, T., Gong, Z., Toyama, D., d'Autume, C. d. M., Li, Y., Terzi, T., Mikulik, V., Babuschkin, I., Clark, A., Casas, D. d. L., Guy, A., Jones, C., Bradbury, J., Johnson, M., Hechtman, B., Weidinger, L., Gabriel, I., Isaac, W., Lockhart, E., Osindero, S., Rimell, L., Dyer, C., Vinyals, O., Ayoub, K., Stanway, J., Bennett, L., Hassabis, D., Kavukcuoglu, K., and Irving, G. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020. URL https://arxiv.org/abs/1910.10683.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, 2018.

Ravichander, A., Gardner, M., and Marasović, A. Condaqa: A contrastive reading comprehension dataset for reasoning about negation. *arXiv preprint arXiv:2211.00295*, 2022. URL https://arxiv.org/abs/2211.00295.

Romanov, A. and Shivade, C. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1586–1596, 2018. URL https://aclanthology.org/D18-1187.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., et al. Multitask prompted training enables zero-shot task generalization. *ICLR 2022*, 2021. URL https://arxiv.org/abs/2110.08207.

Sheng, E., Chang, K.-W., Natarajan, P., and Peng, N. The woman worked as a babysitter: On biases in language generation. 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), pp. 3407–3412, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1339. URL https://aclanthology.org/D19-1339.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., Payne, P., Seneviratne, M., Gamble, P., Kelly, C., Scharli, N., Chowdhery, A., Mansfield, P., Arcas, B. A. y., Webster, D., Corrado, G. S., Matias, Y., Chou, K., Gottweis, J., Tomasev, N., Liu, Y., Rajkomar, A., Barral, J., Semturs, C., Karthikesalingam, A., and Natarajan, V. Large language models encode clinical knowledge, 2022. URL https://arxiv.org/abs/2212.13138.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022. URL https://arxiv.org/abs/2206.04615.

Suzgun, M., Scales, N., Scharli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., ZHou, D., and Wei, J. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022. URL https://arxiv.org/abs/2210.09261.

Talat, Z., Névéol, A., Biderman, S., Clinciu, M., Dey, M., Longpre, S., Luccioni10, A. S., Masoud11, M., Mitchell10, M., Radev12, D., et al. You reap what you sow: On the challenges of bias evaluation under multilingual settings. *Challenges & Perspectives in Creating Large Language Models*, pp. 26, 2022.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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)*, pp. 4149–4158, 2019.

Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Bahri, D., Schuster, T., Zheng, H. S., Houlsby, N., and Metzler, D. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2022a. URL https://arxiv.org/abs/2205.05131.

Tay, Y., Wei, J., Chung, H. W., So, D. R., Shakeri, S., Garcia, X., Tran, V. Q., Zheng, H. S., Rao, J., Zhou, D., Metzler, D., Houlsby, N., Le, Q. V., and Dehghani, M. Transcending scaling laws with 0.1% extra compute. In *arxiv*, 2022b.

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. LaMDA: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022. URL https://arxiv.org/abs/2201.08239.

Vu, T., Wang, T., Munkhdalai, T., Sordoni, A., Trischler, A., Mattarella-Micke, A., Maji, S., and Iyyer, M. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7882–7926, 2020. URL https://aclanthology.org/2020.emnlp-main.635.

Vu, T., Lester, B., Constant, N., Al-Rfou', R., and Cer, D. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 5039–5059, 2022. URL https://aclanthology.org/2022.acl-long.346.

Wallace, E., Feng, S., Kandpal, N., Gardner, M., and Singh, S. Universal adversarial triggers for attacking and analyzing NLP. 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)*, pp. 2153–2162, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1221. URL https://aclanthology.org/D19-1221.

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

Wang, T., Roberts, A., Hesslow, D., Scao, T. L., Chung, H. W., Beltagy, I., Launay, J., and Raffel, C. What language model architecture and pretraining objective work best for zero-shot generalization? *ICML*, 2022a. URL https://arxiv.org/abs/2204.05832.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions, 2022b. URL https://arxiv.org/abs/2212.10560.

Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., Ashok, A., Dhanasekaran, A. S., Naik, A., Stap, D., et al. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*, 2022c. URL https://arxiv.org/abs/2204.07705.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. *ICLR 2022*,

2021. URL https://openreview.net/forum?id=gEZrGCozdqR.

Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Aga, F., Huang, J., Bai, C., et al. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4:795–813, 2022.

Xu, Z., Shen, Y., and Huang, L. Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning, 2022. URL https://arxiv.org/abs/2212.10773.

Ye, Q., Lin, B. Y., and Ren, X. Crossfit: A few-shot learning challenge for cross-task generalization in NLP. In *EMNLP*, 2021. URL https://arxiv.org/abs/2104.08835.

Ye, S., Kim, D., Jang, J., Shin, J., and Seo, M. Guess the instruction! making language models stronger zero-shot learners. *arXiv preprint arXiv:2210.02969*, 2022.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., and Choi, Y. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.

Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Appendix

Table of Contents

A Extended Related Work **16**

B Experimental Details **17**

B.1 Instruction Tuning 17

B.2 Single-Task Finetuning 17

B.3 Evaluation 17

C Input Inversion Details **17**

A. Extended Related Work

We discuss directly related work in Section 2, and other important related works in this section.

Large Language Models As the foundation of instruction tuning, the practice of pretraining one general-purpose language representation that is useful for multiple downstream tasks has a long tradition that goes back at least Mikolov et al. (2013) and Dai & Le (2015). In 2018, Peters et al. (2018) and Devlin et al. (2019) cemented the paradigm of pretraining a large model on a large unsupervised corpus, and the field of NLP quickly converged to using these models which substantially outperform the prior art of non-pretrained task-specific LSTM models on all tasks. However, the dominate way to access that high-quality syntactic and semantic knowledge encoded in pretrained models was not to prompt them with instructions, but to train an additional task-specific linear layer that maps the model activations into numerical class labels. A short year later, Radford et al. (2019), Raffel et al. (2020), and Lewis et al. (2020) popularized the notion that downstream tasks—and multiple tasks—can be jointly learned by directly using the pretrained LM head to generate the answers in natural language (cf. task-specific numerical class labels), the task-general nature of these generative models became the precursor to many multitask transfer learning studies (McCann et al., 2018; Khashabi et al., 2020; Ye et al., 2021; Vu et al., 2020), which in turn led to the first wave of instruction tuning as described in Section 2.

The continuing advancement in research on the pretraining corpora, architectures and pretraining objectives of LMs also has a large impact on instruction tuning. As of 2022, decoder-only left-to-right causal Transformers dominate the market of models larger than 100B (Brown et al., 2020; Thoppilan et al., 2022; Rae et al., 2021; Chowdhery et al., 2022; Hoffmann et al., 2022), and all models of such size class with fully public model parameters are decoder-only (Wang & Komatsuzaki, 2021; Le Scao et al., 2022; Zhang et al., 2022), the decision of which are often due to better hardware and software framework support. However, Raffel et al. (2020), Lewis et al. (2020), and Tay et al. (2022a) have consistently found that left-to-right causal language modeling is a suboptimal objective, while Tay et al. (2022b) and Wang et al. (2022a) particularly showed that a mixture of non-sequential objectives is much superior for downstream tasks with zero-shot and few-shot prompting. An additional factor which remains under-explored is the relationship between pretraining corpora, instruction tuning, and downstream abilities. Typically, public models are all trained on one of a few public corpora: C4 (Raffel et al., 2020), The Pile (Gao et al., 2020), or ROOTs (Laurençon et al.).

Instruction Tuning In Section 2 we outline major developments in instruction tuning. Other important developments include the prospect of complimenting or replacing few-shot in-context learning-the currently predominate method of evaluating pretrained and instruction-tuned models—with parameter-efficient tuning. As standard finetuning of models larger than 100B requires a high number of accelerators with the right interconnects often too expensive even for many industry labs, parameter-efficient tuning (a.k.a. continuous or soft "prompt tuning") shows that only updating a small subset of model parameters can reach comparable performance as fully tuning all model parameters (Lester et al., 2021; Vu et al., 2022; Hu et al., 2021; see He et al., 2022 for a detailed analysis). Notably, Liu et al. (2022b) show that, due to the long sequence length of few-shot ICL and that the few-shot exemplars need to be repeatedly inferenced for evaluating every example, parameter-efficient tuning can be computationally cheaper and higher performing than in-context learning. Further, Liu et al. (2022b), Vu et al. (2022), Wei et al. (2021), and Singhal et al. (2022) collectively show that both single-task and multi-task parameter-efficient tuning can be productively combined with instruction tuning, either before or after regular full-model instruction tuning. This line of work makes it easy for other researchers to build on top of a general-domain instruction-tuned model, and collect a custom instruction-tuning mixture for their use, e.g., with multiple modalities (Ahn et al., 2022; Huang et al., 2022; Xu et al., 2022) or special domains such as science and medicine (Lewkowycz et al., 2022; Singhal et al., 2022).

Problems Addressed by Instruction Tuning & Alignment Techniques Instruction tuning is part of a line of work designed to "align" language models with more useful objectives and human preferences. In the absence of

such methods, language models are known to demonstrate toxic/harmful behaviour (Sheng et al., 2019; Liang et al., 2021; Wallace et al., 2019), generate non-factual information (Maynez et al., 2020; Longpre et al., 2021; Devaraj et al., 2022), and other challenges in deployment and evaluation (Zellers et al., 2019; McGuffie & Newhouse, 2020; Talat et al., 2022). Analyzing, evaluating and mitigating these problems pose a promising direction for future work (Gao et al., 2022; Ganguli et al., 2022). Instruction tuning warrants greater investigation, as it has already demonstrated itself an encouraging remedy in reducing NLP bias metrics, as shown in Chung et al. (2022).

B. Experimental Details

B.1. Instruction Tuning

Our instruction tuning follows the setup described in Chung et al. (2022). For few-shot and few-shot Chain-of-Thought prompts during finetuning our templatizing procedure generates few-shot examples with 2, 3, or 5 exemplars.

B.2. Single-Task Finetuning

For single-task finetuning, described in Section 4, our models are finetuned for 100,000 steps for all tasks. We use a constant learning rate of 0.001, a dropout probability of 0.1, and a batch size of 128 length-512 sequences. We save a checkpoint every 20 steps and report test performance on the model checkpoint corresponding to the highest validation performance. For tasks without a validation split, we hold out 1024 training examples for validation. For tasks without a test split, we hold out 1024 training examples for validation and report results on the original validation set. For PubmedQA, we do not use any of the unlabeled and artificially generated QA instances associated with the dataset. For CxC, we only consider the text-text portion of the dataset, following Vu et al. (2022). For tasks with less than 1K training examples, we report average results across 3 random seeds.

We also evaluate on certain metrics to account for label skew in some of the datasets, as shown in Table 3.

B.3. Evaluation

For Held-In evaluations we use the validation sets from 4 question answering (QA) tasks, BoolQ, ARC Easy, ARC Challenge, and AI2's Middle School Science Exams, and 4 natural language inference (NLI) tasks, including ANLI R1, R2, R3, and RTE. These datasets are contained in the Flan 2022 finetuning collection and represent challenging benchmarks, often used to evaluate LLMs on QA and NLI. The Held-In score is the mean accuracy across these 8 tasks.

For the Chain-of-Thought (CoT) evaluation, we use the mean accuracy across 5 datasets which have been prepared with prompts which request step-by-step explanations in their target answers: GSM8K, StrategyQA, SVAMP, Asdiv, and CommonsenseQA.

For the Held-Out tasks, we use MMLU's suite of 57 exams, and BBH's suite of 23 tasks where PaLM performed worse than the average human annotators. MMLU tasks were removed from the Super-Natural Instructions part of the Flan 2022 collection at training, to ensure they were Held-Out.

C. Input Inversion Details

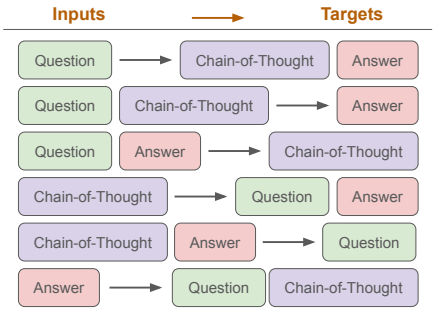

Figure 7: **Input Inversions permutations for a Zero-Shot Chain-of-Thought example.** Each is accompanied by a corresponding instruction template that prompts the model with what the input is, and what to predict as the targets.

For the input inversion experiments we note that Flan 2021, P3++, and Super-Natural Instructions already implicitly include tasks that have been inverted, e.g. question answering to question or context generation. Consequently, we choose to also create input inversions for the remaining datasets in the Flan 2022 collection, following Chung et al. (2022) as closely as possible, including for the Dialog, Program Synthesis, and Chain-of-Thought tasks.

As examples: for Dialog tasks, we write template instructions asking for the previous conversational history from the current dialog turn; for program synthesis we ask for the coding question which the code solves; and for Chain-of-Thought we include every permutation of the query-answer-explanation triple, where at least one of the three appears as the in output. An illustration of Chain-of-Thought input inversion permutations are shown in Figure 7.

These inversions are mixed in with the existing tasks at a rate of 30%, meaning for a Dialog task, 3 inverted examples will be generated for every 10 regular examples. We choose this rate for simplicity, approximately mirroring prior work, and leave the large space of exploration for future work.

Dataset	Metric	Used in Held-In	CoT	ST-FT H-In	ST-FT H-Out	Citation
ARC E+C	Acc	✓		✓		(Clark et al., 2018)
ANLI R1+R2+R3	3-class F1	✓		✓		(Nie et al., 2020)
AI2 Mid. Science	4-class F1	✓		✓		AI2 Science Questions
BoolQ	AUC-ROC	✓		✓		(Clark et al., 2019)
RTE	AUC-ROC	✓		✓		(Bentivogli et al., 2009)
SQuAD V2	F1			✓		(Rajpurkar et al., 2018)
CosmosQA	Acc			✓		(Huang et al., 2019)
GSM8K	Acc		✓			(Cobbe et al., 2021)
StrategyQA	Acc		✓			(Geva et al., 2021)
SVAMP	Acc		✓			(Patel et al., 2021)
Asdiv	Acc		✓			(Miao et al., 2020)
CommonsenseQA	Acc		✓			(Talmor et al., 2019)
WANLI	Acc				✓	(Liu et al., 2022a)
MedNLI	Acc				✓	(Romanov & Shivade, 2018)
CondaQA	Acc				✓	(Ravichander et al., 2022)
PubmedQA	F1				✓	(Jin et al., 2019)
CxC	Spearman				✓	(Parekh et al., 2021)

Table 3: **Datasets used for Various Finetuning and Evaluation Experiments.** ST-FT stands for Single Task Finetuning. H-In stands for Held-In. H-Out stands for Held-Out. The chain-of-thought datasets are held-in as they are included either in the Natural Instructions v2 or Flan submixtures of the Flan Collection. Templates for the evaluations are found in https://github.com/google-research/FLAN/blob/main/flan/v2/templates.py—we use the first template in the list for each dataset. For MMLU evaluation templates we follow the original settings in https://github.com/hendrycks/test.