Multi-Task Transfer Matters During Instruction-Tuning

David Mueller **Mark Dredze** **Nicholas Andrews**
Johns Hopkins University
{dam,noa}@jhu.edu

Abstract

Instruction-tuning trains a language model on hundreds of tasks jointly to improve a model's ability to learn in-context, either from task descriptions, task samples, or both; however, the mechanisms that drive in-context learning are poorly understood and, as a result, the role of instruction-tuning on in-context generalization is poorly understood as well. In this work, we study the impact of instruction-tuning on multi-task transfer: how well a model's parameters adapt to an unseen task via fine-tuning. We find that instruction-tuning negatively impacts a model's transfer to unseen tasks, and that model transfer and in-context generalization are highly correlated, suggesting that this catastrophic forgetting may impact in-context learning. We study methods to improve model transfer, finding that multi-task training—how well the training tasks are optimized—can significantly impact ICL generalization; additionally, we find that continual training on unsupervised pre-training data can mitigate forgetting and improve ICL generalization as well. Finally, we demonstrate that, early into training, the impact of instruction-tuning on model transfer to tasks impacts in-context generalization on that task. Overall, we provide significant evidence that multi-task transfer is deeply connected to a model's ability to learn a task in-context.[1]

1 Introduction

The surprising ability of large language models to follow natural language instructions and perform a variety of complex tasks is attributed in large part to *instructing tuning*, the process of fine-tuning a language model with supervised demonstrations of instructions or in-context examples paired with the desired output (Chung et al., 2022; Wang et al., 2022). However, the mechanisms behind the success of instruction tuning remain poorly understood. In principle, a good in-context learner may perform a task by implementing a learning algorithm over the examples passed in-context (Akyurek et al., 2023); in practice, it is not clear how much of the task is learnt versus *recognized* via memorization or implicit latent inference over a pre-existing mixture of experts (Xie et al., 2022; Olsson et al., 2022). Perhaps most importantly, while instruction tuning explicitly teaches a model to predict a task given it's context, it may also benefit in-context learning via *multi-task transfer* from training on hundreds of supervised tasks jointly.

In this paper, we seek to better understand instruction tuning from the perspective of multi-task transfer. Put simply, instruction-tuning involves continued training of a language model on supervised examples drawn from a variety of tasks. However, a common challenge when attempting to fit a single model to multiple tasks is interference between tasks during optimization, which can lead to models which generalize worse than single task models (Sener and Koltun, 2018; Yu et al., 2020). Despite these challenges inherent to multi-task learning, which are exacerbated with the number of tasks, the ability of an instruction-tuned model to perform in-context learning—learning the task to predict from examples or instructions provided in the context—*improves* with the number of tasks seen during training, because the model sees more demonstrations of context (Sanh et al., 2022).

We aim to characterize the extent to which multi-task transfer dynamics influence the resulting instruction-tuned model, and whether improving multi-task transfer can yield benefits to the transfer and in-context learning (ICL) performance of instruction-tuned models towards unseen tasks.

Contributions We provide an analysis of instruction tuning as a multi-task transfer problem, and we evaluate the effect of multi-task transfer on a model's in-context learning ability. Leveraging

[1]We release our code at https://github.com/davidandym/Multitask-Transfer-Instruction-Tuning

Findings of the Association for Computational Linguistics ACL 2024, pages 14880–14891

few-shot fine-tuning to measure general model transfer, we demonstrate that instruction-tuning harms model transfer to unseen tasks over pre-training; further, we show that model transfer and in-context learning are highly correlated, suggesting that this catastrophic forgetting may negatively impact in-context generalization (§3). Next, we explore different multi-task sampling schemes during instruction-tuning to study how multi-task training impacts in-context generalization (§4); we find that methods which generalize best to the training tasks achieve better ICL generalization, suggesting that multi-task training matters during instruction-tuning. We study the trajectory of catastrophic forgetting, finding that model transfer and ICL ability are both highly correlated in the latter portion of instruction-tuning, again suggesting they are connected (§5); finally, we show that mixing instruction-tuning data with unsupervised pre-training data mitigates catastrophic forgetting and improves ICL generalization. Our results reveal that model transfer is deeply connected to in-context learning ability and suggest future directions to improve instruction-tuning for LLMs.

2 Background & Preliminaries

2.1 Multi-Tasking Learning & Transfer

Transfer learning—the adaptation of knowledge or decision rules learned under one environment to another—has played a critical role in deep learning: *fine-tuning* neural network parameters trained in high-resource environments and tasks to specific tasks of interest has led to massive improvements in generalization in nearly every domain of machine learning (Donahue et al., 2014; Peters et al., 2017, 2018). Intuitively, fine-tuning succeeds when pre-training moves the parameters of a model into regions of the parameter space where solutions to the target task exist, which single-task learning alone would not discover (Juneja et al., 2023).

A key driver in deep transfer learning is *multi-task learning* (MTL; Caruana, 1997): fitting a single neural network to multiple tasks jointly can lead to a solution that is more general than single-task models, and can therefore *a priori* transfer better to unseen tasks and domains (Finn et al., 2017; Aribandi et al., 2022). This paradigm of adapting multi-task parameters to tasks has led to improvements in NLP for lower resource tasks and languages (Mueller et al., 2020; Gheini et al., 2023), and may even explain the success of unsupervised pre-training (Weber et al., 2021).

Despite these successes, multi-task optimization often struggles to fit large, diverse sets of tasks jointly due to inter-task conflicts and task imbalances (Sener and Koltun, 2018; Kendall et al., 2018), and several methods have been proposed that attempt to address this by mitigating task conflict and balancing task losses during training (Yu et al., 2020; Chen et al., 2018; Wang et al., 2020). However, when the ultimate goal is to transfer to specific target tasks, prior work has shown that increasing the number of pre-training tasks is beneficial despite increasing task conflicts (Aghajanyan et al., 2021; Aribandi et al., 2022); thus, it is not clear if task conflicts or imbalances are necessarily harmful to multi-task transfer.

2.2 Mechanisms Behind In-Context Learning

While fine-tuning, particularly parameter-efficient fine-tuning (Xu et al., 2023), is the predominant method of transfer in NLP, recently *in-context learning* has been proposed as a method to combine transferable knowledge from pre-training with an LLMs ability to infer *what task to perform* over the inputs based on the context provided (Brown et al., 2020). ICL is a particularly attractive form of model transfer because, unlike fine-tuning, it requires no parameter updates or additional training to adapt to an unseen task; instead, given an input $(r_k; x)$ where the input x is concatenated with task-specifying context r_k, a strong in-context learner will infer that is must perform task k from r_k and then accurately predict task k over the input x.

However, the mechanisms behind ICL in LLMs are not yet understood: one thread of work suggests that, when r_k is a set of task exemplars, neural networks—specifically attention-based models (Vaswani et al., 2017)—can approximate learning algorithms over those exemplars (Akyurek and Andreas, 2023), or leverage induction heads to perform a prefix-match and copy mechanism (Olsson et al., 2022). Another direction suggests that ICL operates as a latent mixture over expert functions learned during pre-training (Xie et al., 2022), i.e. that ICL first performs task inference by inferring a mixture of pre-trained experts, and then uses that mixture to make a prediction. This latter perspective, in particular, suggests that, in addition to the task inference mechanisms, general multi-task knowledge learned during pre-training may play a critical role in in-context learning ability (§3).

2.3 The Benefits of Instruction Tuning

Despite non-trivial generalization to unseen contexts, the performance of fully unsupervised LLMs on ICL tasks can often be poor. *Instruction tuning* (IT) has emerged as an effective way to improve a model's in-context generalization (Sanh et al., 2022; Wei et al., 2022). In instruction tuning, multiple tasks are first converted to Text-to-Text (T2T) problems (Raffel et al., 2020) and contextualized via task-specifications (Sanh et al., 2022; Wang et al., 2022). Language models are then fine-tuned on all tasks jointly, such that the model must use the specification, r_k, to infer which task to predict. Recent work has shown that as the number of training tasks and unique task specifications increases during instruction-tuning, ICL generalization to unseen tasks improves (Chung et al., 2022).

Formally, instruction tuning considers a collection of pairs of task specifications and datasets, $\{(r_k, S^k)\}_{k=1}^K$ where $S^k = \{(x_i^k, y_i^k)\}_{i=1}^{N^k}$ is the training dataset for task k. During training, the input to the language model, F_θ, is a concatenation of the specification and input, $(r_k; x)$, and the objective of IT is to minimize:

$$\widehat{\mathcal{L}}_{IT}(\theta) = \mathop{\mathbb{E}}_{k\sim P(K)} \left[\mathop{\mathbb{E}}_{(x,y)\in S^k} \left[\ell(F_\theta(r_k; x), y) \right] \right] \quad (1)$$

where ℓ is typically the negative log-likelihood loss and $P(K)$ is a distribution over all training tasks. Although the goal of IT is to improve ICL generalization, the loss that it minimizes (1) is a *multi-task* objective, and therefore instruction-tuned models are explicitly trained with multi-task supervision. However, because the mechanisms behind ICL generalization are poorly understood, it is not clear if multi-task transfer from instruction-tuning is a key benefit to in-context learning, or if it impacts ICL generalization at all. In this work, we study how instruction-tuning impacts multi-task transfer, and whether multi-task transfer matters for ICL.

2.4 Experimental Setup

Instruction-Tuning Data We consider Super-Natural Instructions (Wang et al., 2022) as our large set of instruction-tuning tasks, which consists of 1,600+ tasks in several languages. For the purposes of our experiments—specifically, our focus on multi-task rather than multilingual transfer—we limit our tasks to those which are both English and represent Natural Language tasks (i.e. we do not use synthetically generated tasks). We select 58 task categories (a subset of task categories representing natural language tasks), resulting in 619 training tasks; all of the training task categories can be seen in Table 1. For each task, we select a hold out 10% of it's data as test data and 10% of it's data as validation data. This set of 619 tasks serves as our massively multi-task training objective for instruction-tuning.

When evaluating transfer to unseen tasks, we test on 93 held out tasks which fall under the 11 remaining task categories that are held out from the training set Table 2. We hold out 100 samples from each task as a test set, and 40 additional samples for few-shot fine-tuning. When performing ICL, we use 3 contextual samples provided by SNI as metadata for our in-context exemplars in r_k.

Modeling & Evaluation We use the LM-Adapted T5 models for all our experiments, which are a family of T5-Large models pre-trained only on the C4 dataset with the T5 denoising objective, and then fine-tuned on C4 with a standard language modeling objective.[2] We fine-tune these models on our instruction-tuning tasks for 3 epochs (roughly 300,000 steps), using a linear learning rate schedule with a warmup to an initial learning rate of 5e-4, a batch-size of 16, and using the standard LM objective (Wang et al., 2022). For more details, see Appendix B. When performing instruction-tuning with in-context exemplars as r_k, we use 3 samples provided by SNI as task-metadata for our exemplars; when instruction-tuning without instructions, we set r_k to the unique task-ID number in SNI Figure 5. Following Wang et al. (2022), we use the 'Rouge-L r' metric to evaluate all tasks.

3 Does Transfer From Instruction-Tuning Matter for In-Context Learning?

Instruction-tuning is, in essence, multi-task fine-tuning of a pre-trained LM, which has been shown to improve transfer in some settings (Aghajanyan et al., 2021); however, it is not clear if instruction-tuning improves transfer to *unseen* tasks, or even if model transfer is relevant to in-context learning ability. In this section we ask: *does instruction-tuning impact model transfer to unseen tasks, and does it's impact on task transfer correlate with it's impact on in-context learning accuracy?*

[2]LM-Adapted T5 models adapt much quicker to prompt-based tuning (Lester et al., 2021), as well as instruction-tuning and LoRA fine-tuning in our experiments, ostensibly because SuperNatural Instructions frames all tasks as conditional language-modelling problems.

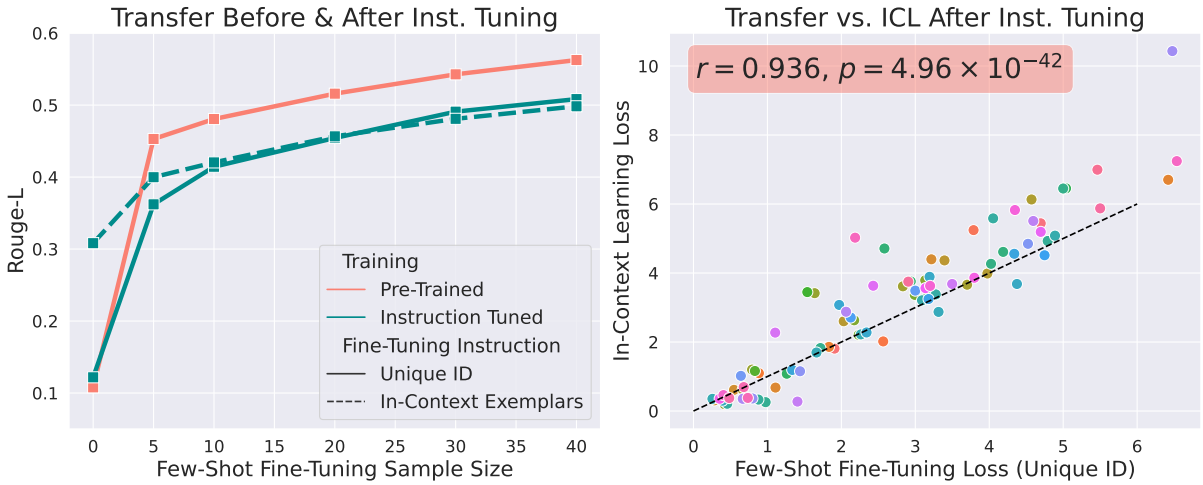

Figure 1: **(Left): Instruction-Tuning Harms Transfer to Unseen Tasks** when performing few-shot fine-tuning. Pre-trained models adapt faster and with higher accuracy to unseen tasks than instruction-tuned models, even when fine-tuning leverages in-context exemplars. **(Right): Few-Shot Transfer is Correlated with In-Context Learning Accuracy**: we see a high correlation between model performance after 5-shot fine-tuning and model performance when leveraging in-context learning, suggesting that *transfer matters for in-context learning performance*.

3.1 IT Harms Transfer To Unseen Tasks

We are interested in how multi-task learning impacts transfer to unseen tasks: because pre-trained models are generally not capable of performing in-context learning very well, we rely on *few-shot fine-tuning* to measure model transfer before and after instruction-tuning. More specifically, we fine-tune task-specific LoRA layers (Hu et al., 2022)—a parameter efficient fine-tuning technique known to achieve strong fine-tuning performance across a wide variety of NLP tasks—on 0 to 40 samples of each task in our set of unseen tasks. We compare Pre-Trained models to Instruction Tuned models when setting the distribution over training tasks, $P(K)$, to be proportional to the task training set size, as in Wang et al. (2022).

In Figure 1 (Left), we plot unseen task fine-tuning transfer, as the number of fine-tuning samples increases, for both Pre-Trained and Instruction-Tuned models. On instruction-tuned models, we perform fine-tuning in two ways: first, as with the pre-trained models, we provide no task context during fine-tuning; second, we set the task context to match the context provided during instruction-tuning (In-context Exemplars), so that fine-tuning may benefit from in-context learning as well.[3] We see that instruction-tuning *harms* model transfer to unseen tasks compared to pre-trained models, i.e. instruction-tuning results in some amount of *catastrophic forgetting* (Kirkpatrick et al., 2017) of general task knowledge from pre-training. This is surprising because Aghajanyan et al. (2021) and Aribandi et al. (2022) suggest that supervised multi-task training can improve model transfer to seen tasks over just pre-training; we find that it does not necessarily improve transfer to *unseen* tasks.

A second take-away from Figure 1 (Left) is that, rather than benefiting from multi-task transfer, in-context learning ability improves via instruction-tuning *in spite of* it's impact on model transfer. However, it is still unclear whether how well a model's parameters transfer, via fine-tuning, to a specific task has any relevance to how well that model can perform in-context learning on that task. To answer this, in Figure 1 (Right) we plot 5-shot fine-tuning test loss by in-context learning test loss for each unseen tasks using the instruction-tuned model; note that we use no in-context exemplars during fine-tuning, so the fine-tuning generalization is not impacted by in-context learning. Surprisingly, we see a very strong correlation (Spearman-R of 0.936 with a p-value of 4.96×10^{-42}) between a model's generalization to unseen tasks when leveraging few-shot fine-tuning vs. using in-context learning. Such a strong correlation suggests that model transfer, as measured by how easily parameters fine-tune or *adapt* to a new task, may be meaningful for in-context learning.

[3]There is no overlap between the examples used for ICL and Fine-tuning.

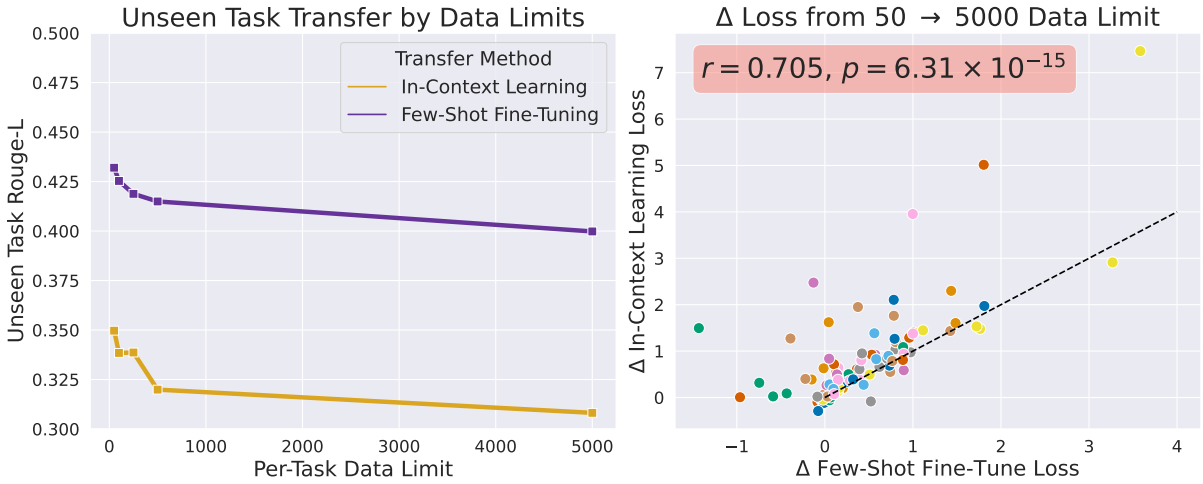

Figure 2: **(Left): More Data Per Task Harms Transfer and In-Context Learning** during Instruction Tuning. Using only 10 samples per task is enough to get strong ICL generalization, and more samples leads to degradation in both fine-tuning transfer and in-context learning ability. **(Right): Changes to Few-Shot Transfer are Correlated with Changes to In-Context Learning Generalization**: when comparing the change, from a $5,000$ data limit model to a 50 data limit model, in fine-tuning test loss to in-context test learning loss, we see a relatively high Spearman-R correlation, suggesting that impacts to few-shot parameter transfer matter for in-context learning.

3.2 Less Data Improves Transfer *and* ICL

Wang et al. (2022) find that using the full dataset of each task during training can harm in-context learning generalization, i.e. that only a few (~ 60) samples per-task are needed to impart strong ICL abilities; we study whether this can be understood, in part, through the lens of our results in §3.1. Namely, we hypothesize that fewer samples per task may reduce catastrophic forgetting, leading to stronger model transfer and improving ICL accuracy.

To test this, we instruction-tune several T5 Large models on increasing limitations of samples per-task, from 10 up to $5,000$ (the maximum number of samples for any task in SuperNatural Instructions), and measure fine-tuned performance and in-context learning performance (using in-context exemplars). When the per-task limit exceeds the training dataset size of a task, we simply use the entire task dataset; additionally, as in §3.1, we set $P(K)$ to be proportional to the training dataset size of each task under each setting (i.e. for a limit of 10 samples per-task, $P(K)$ is uniform). Note that, using 10 samples per task, a model is instruction-tuned on $6,190$ samples in total during instruction-tuning, a significant reduction from the 1.5-million samples that a $5,000$ data-limit model trains on.

Despite a significant reduction in data, we see that, as the number of samples per task decreases, both few-shot fine-tuning performance and in-context learning performance increase significantly. This result not only confirms that only a few samples per-task are necessary to achieve strong ICL capabilities, but also suggests that further training on additional data per-task harms ICL performance via the same mechanisms by which it harms fine-tuning transfer, i.e. by catastrophic forgetting of general task knowledge.

To study whether this trend is corroborated at the individual task-level, in Figure 2 (Right) we plot the *change* in In-Context Learning Loss by the change in Few-Shot Fine-Tuning Loss on Unseen Tasks, when using 50 vs. $5,000$ samples per-task. Again, we see a surprisingly strong correlation (in this case, a Spearman-R of 0.705 with a p-value of 6.31×10^{-15}) between individual task behavior with respect to fine-tuning and ICL transfer. This correlation is surprising because it is not obvious why a change to a task's fine-tuning performance, which measures how easily a model's *parameters* can adapt to a task, would be tied to the task's generalization under in-context learning, which uses ostensibly separate mechanisms of task inference over the context (Olsson et al., 2022; Akyurek and Andreas, 2023); nevertheless, our results suggest that these factors are tied together in some way.

Discussion Together, the results of §3.1 and §3.2 show that Instruction-Tuning generally harms model transfer to unseen tasks, causing catastrophic forgetting of general task knowledge, when transfer is measured via *fine-tuning*: how easily

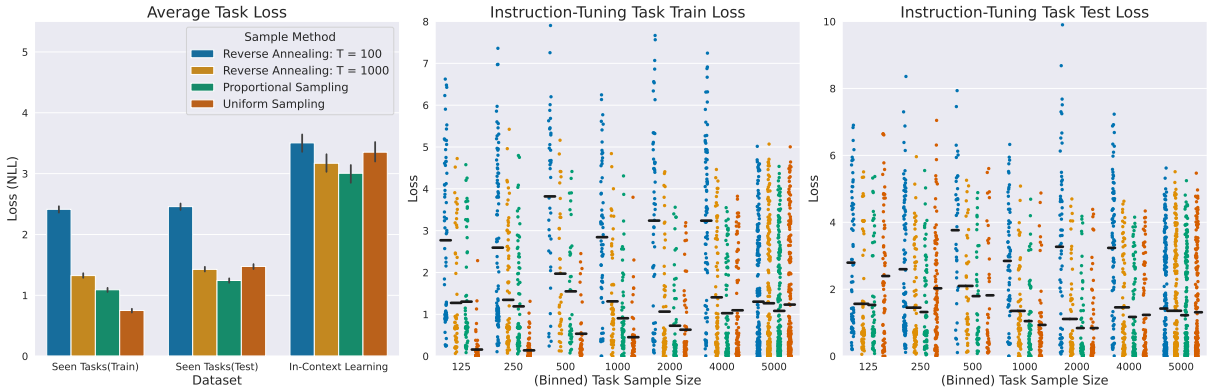

Figure 3: **(Left): Models which exhibit the strongest multi-task generalization also demonstrate the best ICL ability.** Despite being trained on the same amount of data and tasks, MTL models whose sample methods result in the best *generalization* to the training tasks also have the stronger in-context learning abilities. **(Center & Right):** Striking the right balance between high-resource and low-resource tasks during training is important to in-context learning. Despite having comparable generalization on higher resource tasks, models which over- or under-fit lower resource tasks exhibit worse ICL generalization: fitting the training tasks well matters during instruction-tuning.

the parameters of the model can adapt to a new, unseen task. Furthermore, while instruction-tuning significantly improves a model's in-context learning ability, we show that individual task in-context learning performance is tied to the model's ability to transfer to that task via fine-tuning. This is significant for a couple reasons: first, it sheds light on the mechanisms driving in-context learning, suggesting that in-context learning relies, to some extent, on how close a model's *parameters* are to a good solution for each task; second, it suggests that one way to improve ICL ability is to improve multi-task transfer, which we explore in the next sections.

4 Multi-Task Generalization vs. ICL

Our results in §3 indicate that, surprisingly, the ability of a model to transfer to an unseen task may be indicative of a model's ability to learn the task *in-context*, after instruction-tuning. This result suggests that, by improving multi-task transfer during instruction-tuning, we may improve it's in-context learning ability to unseen tasks. In this section, we ask *how the multi-task training of instruction-tuning, specifically how well the model fits and generalizes to the tasks seen during instruction-tuning, impacts ICL performance*.

As described in §2.2, instruction-tuning minimizes $\widehat{\mathcal{L}}_{IT}(\theta)$, computed as the expected task loss over $k \sim P(K)$, where $P(K)$ is a distribution over the set of training tasks K; during training, $P(K)$ reflects the rate at which tasks are sampled for each batch. In standard multi-task training, this distribution is often directly modified using different heuristics or methods to explicitly balance different task losses and mitigate task conflicts (Chen et al., 2018; Yu et al., 2020; Wang et al., 2020, *inter alia.)*. However, the goal of these methods is often to improve generalization to the training tasks; it is not clear if specialized optimization methods are necessary to improve ICL during instruction-tuning.[4]

Instead, in instruction-tuning, $P(K)$ is proportional to the dataset size of each task, biasing training towards tasks that have more training data (Chung et al., 2022; Wang et al., 2022). To study whether multi-task training impacts ICL generalization during instruction-tuning, we consider two additional distributions for $P(K)$: Uniform Sampling, which treats $P(K)$ as the uniform distribution, and Reverse Annealing, inspired by Choi et al. (2023), which initializes $P(K)$ to the proportional distribution and gradually heats the temperature of the distribution during training, up to some max temperature T. We sweep over $T = 10^{\{2,3,4,5\}}$ and report the two highest performing methods based on seen task validation data.

In Figure 3 (Left) we plot the training and test loss of the (seen) instruction-tuning tasks, as well as the in-context learning loss on unseen tasks, for instruction-tuned models using 4 different sampling schemes (Uniform Sampling, Proportional Sampling, and Reverse Annealing with $T = 100$ and $T = 1000$). We find that multi-task gener-

[4]Many of these methods are unsuited for instruction-tuning with LLMs. For instance, some methods require storing the model gradients for each task, which would require 2 TB of memory for T5-Large; other methods require a fully mixed batch, which requires a minimum batch-size of 619.

alization (how well the model generalizes to the training tasks) is correlated with in-context learning generalization to unseen tasks: Proportional Sampling achieves the lowest Test Loss on the seen tasks, and correspondingly has the lowest test loss of unseen tasks via In-Context Learning, followed by the model using Reverse Annealing with $T = 1000$ and Uniform Sampling. This is notable, as it indicates that the multi-task training aspect of instruction-tuning has a significant impact on how well a model can perform in-context learning.

In Figure 3 (Center & Right) we plot the individual task train loss and test loss, respectively, for tasks when binned by dataset size. The sampling scheme which achieves the best in-context generalization (Proportional Sampling) has the highest parity, for both training loss and test loss, across tasks of different resources. Conversely, while all the methods we consider have comparable loss on the very high resource tasks ($\sim 5,000$ samples), they have significantly higher test losses on the lower resource tasks, suggesting that generalizing well to the long-tail of tasks in the instruction-tuning dataset is important for in-context learning generalization. Importantly, striking the right balance during training seems critical: while Reverse Annealing sampling underfits lower resource tasks, Uniform Sampling *overfits* lower resource tasks; only Proportional Sampling manages to achieve strong test loss on lower resource tasks by not over- or under-fitting them during training.

Discussion The results of §4 indicate that multi-task *training*, i.e. how the training tasks are optimized and how well the final solution generalizes to them, impacts in-context learning ability during instruction-tuning. This result is not intuitive, as it is not obvious that task imbalances should impact in-context learning abilities after training given recent results on scaling up the number of tasks for both instruction-tuning (Chung et al., 2022; Wang et al., 2022) as well as multi-task transfer (Aribandi et al., 2022). While none of these works consider specialized MTL methods as a component of transfer learning, our findings suggest that ICL generalization can be improved by stronger MTL methods which better balance seen task generalization during training. However, we also find that proportional sampling is a very strong baseline.

5 The Trajectory of ICL & Forgetting

In §4 we focus on improving multi-task transfer by explicitly attempting to address task imbalances and balance generalization across training tasks. In this section, we instead focus on improving model transfer by mitigating the catastrophic forgetting we observe in §3. More specifically, we ask *can early-stopping and continual learning improve ICL generalization by mitigating the effects of catastrophic forgetting on model transfer?*

5.1 The Trajectory of Catastrophic Forgetting

We begin by studying whether early-stopping can mitigate catastrophic forgetting, and improve ICL performance, by studying the *trajectory* of these values during instruction-tuning. Specifically, in Figure 4 (Left) we plot few-shot fine-tuning transfer and in-context performance during the instruction-tuned model trajectory, evaluating checkpoints every $50,000$ steps of instruction-tuning. Interestingly, we see that early-stopping is not an effective method to improve ICL generalization because ICL generalization increases throughout the entire training process; even more surprising, we find that fine-tuning transfer increases over time as well. While we expect catastrophic forgetting to get worse as we continue to fit the training tasks, we instead see that, after an initial phase of significant catastrophic forgetting, transfer begins to *increase* with continued multi-task training.

The trajectories in Figure 4 suggest that, while ICL performance and fine-tune transfer are initially at odds, they may improve together as training progresses.[5] In Figure 4 (Center) we plot the correlation, across tasks, between the change in fine-tune transfer and ICL performance in the first $50,000$ steps of training: we see that the correlation is very low (Spearman-R of 0.245 with a p-value of 0.019). In contrast, the correlation between the change in fine-tune transfer and ICL performance from the $50,000$th step to the end of training (Figure 4 (Right)) is much higher, with a Spearman-R correlation of 0.621. This correlation suggests that, during the latter portion of instruction-tuning, the mechanisms which improve fine-tuning transfer are also driving improvements to ICL performance.

[5] However, we note that this trajectory of fine-tuning transfer occurs even in a standard multi-task setup (i.e. no task contexts are included), meaning the behavior of non-monotonic forgetting is not *because of* in-context learning.

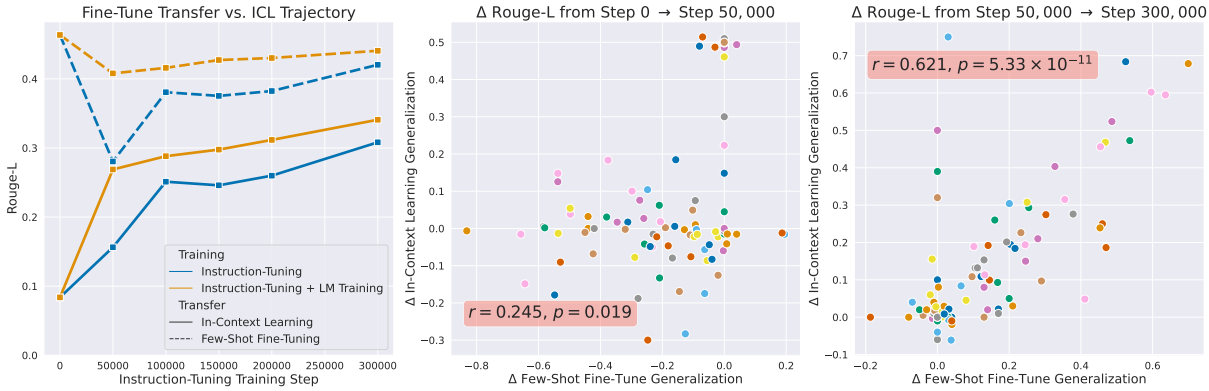

Figure 4: **(Left) Fine-Tuned vs. ICL Transfer over time** during instruction-tuning with and without mixing in unsupervised data. Early into training, few-shot fine-tuning performance degrades while in-context learning performance increases significantly; after this period, both fine-tune transfer and in-context learning begin to improve together. Additionally, mixing in C4 data alleviates catastrophic forgetting and *improves* ICL generalization. **(Center)** During the first phase of training there is little correlation between the effects on transfer and in-context learning performance. **(Right)** After the first $50,000$ steps of training, the changes to fine-tuning transfer and ICL performance become much more tightly correlated, implying multi-task transfer impacts ICL performance *after this initial phase of instruction-tuning*.

5.2 Continual Learning on Pre-Training Data

Finally, we ask whether or not we can mitigate catastrophic forgetting by continuing to train on the data used for pre-training while performing instruction-tuning; specifically, whether continuing to train on C4 (Raffel et al., 2020), the large, unsupervised dataset that our T5 models are pre-trained on, can help in-context learning generalization. To study this, we train an additional instruction-tuned model where, for every 3 steps of instruction-tuning, we mix in an additional step on a sample of unsupervised C4 data; every step has the same batch-size, and models with and without the C4 data take the same number of steps on *instruction-tuning data*.

We plot the comparison between instruction-tuned models and models trained on instruction-tuning and C4 data in Figure 4 (Left). As expected, mixing in C4 data into instruction-tuning results in significantly less catastrophic forgetting, with respect to few-shot fine-tuning, throughout training. However, continued training on C4 data leads to a significant increase in *ICL generalization as well*. This is unexpected because it is not clear why continued training on C4 data should improve in-context learning ability: it adds no additional examples of task specifications or task supervision and consists solely of data which the pre-trained model has *already seen*. We conjecture that the benefit to in-context learning of continual training on unsupervised data, is due to effects on fine-tuning transfer and the strong connection we observe (throughout this paper) between a model's degree of fine-tuning transfer and in-context learning strength across tasks.

6 Conclusion

Summary of findings This paper studies the impact of multi-task transfer from instruction-tuning on in-context generalization. We find that instruction-tuning has a surprisingly negative impact on transfer to unseen tasks and, moreover, that a models ability to transfer to a task and it's in-context generalization on that task are highly correlated. However, we also observe that techniques to improve general model transfer—such as training with less data per-task, better balancing multi-task losses during training, and continuing training on unsupervised data during instruction-tuning—can all improve in-context generalization by mitigating the negative impact of instruction-tuning on model transfer. Finally, we additionally find that the negative impact of instruction-tuning on transfer happens early into training, after which both model transfer and in-context generalization improve in a highly correlated manner.

Future Work Overall, our findings highlight several under-appreciated consequences of instruction tuning and point to promising directions for future work. One promising direction is to incorporate ideas from multi-task learning into the instruction tuning process, as a way to increase positive trans-

fer between seen tasks, which may in turn improve multi-task transfer to unseen tasks, thereby improving ICL generalization. Our findings also imply that multi-task transfer, and in particular the impact of the specific tasks used during instruction-tuning, may have a much more significant effect on ICL generalization than previously thought.

Limitations Due to a limited compute budget and desire for others to easily reproduce our results, we focus our evaluations on a relatively small family of pre-trained models. Although we expect our findings to hold for larger model sizes, conducting additional experiments with larger models would be necessary to confirm this. We also focus our experiments on a single dataset, namely Super-Natural Instructions, itself composed of hundreds of tasks. We hold-out particular categories of tasks for unseen evaluations and our conclusions would be strengthened by repeated trials with further held-out categories, although this would significantly increase the required compute budget.

Acknowledgements

We thank three anonymous reviewers for their recommendations and feedback that helped improve the paper. This work was supported in part by the Human Language Technologies Center of Excellence at Johns Hopkins University.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ekin Akyurek and Jacob Andreas. 2023. LexSym: Compositionality as lexical symmetry. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 639–657, Toronto, Canada. Association for Computational Linguistics.

Ekin Akyurek, 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*.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. volume 80 of *Proceedings of Machine Learning Research*, pages 794–803, Stockholmsmässan, Stockholm Sweden. PMLR.

Dami Choi, Derrick Xin, Hamid Dadkhahi, Justin Gilmer, Ankush Garg, Orhan Firat, Chih-Kuan Yeh, Andrew M. Dai, and Behrooz Ghorbani. 2023. Order matters in the presence of dataset imbalance for multilingual learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 647–655, Bejing, China. PMLR.

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

Mozhdeh Gheini, Xuezhe Ma, and Jonathan May. 2023. Know where you're going: Meta-learning for parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2023*,

pages 11602–11612, Toronto, Canada. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. 2023. Linear connectivity reveals generalization strategies. In *The Eleventh International Conference on Learning Representations*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Mueller, Nicholas Andrews, and Mark Dredze. 2020. Sources of transfer in multilingual named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8093–8104, Online. Association for Computational Linguistics.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 527–538. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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. 2022. 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.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. 2020. Gradient vaccine: Investigating and improv-

ing multi-task optimization in massively multilingual models.

Lucas Weber, Jaap Jumelet, Elia Bruni, and Dieuwke Hupkes. 2021. Language modelling as a multi-task problem. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2049–2060, Online. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent 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 *International Conference on Learning Representations*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning.

A Dataset Details

In this paper, we leverage the pre-existing SuperNatural-Instructions dataset (Wang et al., 2022), which consists of a collection of 1,616 NLP and synthetic tasks, along with expert-written instructions and selected examples for each task. Our instruction-tuning training set is collected by first filtering all tasks that are not in English, as well as all synthetically generated tasks. We next select a set of evaluation domains ({ `Healthcare`, `Social Media`, `Social Media -> Twitter`, `Social Media -> Reddit`, `Sports`, `Animals`, `Reviews -> Books`, `Reviews -> Trip Advisor`, `Reviews -> Restaurants`, `Reviews -> Music`, `Public Places -> Restaurants`, `Scientific Research Papers` }) and exclude all tasks from those domains from the training set. Finally, all tasks constructed from a set of adversarial sources ({ `anli`, `paws`, `hans`, `hellaswag`, `codah`, `adversarial_qa` }) are excluded. The remaining 619 tasks, representing the categories listed in Table 1 are used as our training tasks.

The remaining task categories not used for training are split into two different evaluation categories: NLU if their sources are NLP datasets, and Synthetic if they are synthetically generated. We leverage all tasks that fall under each category for these evaluations, so long as the task has more than 140 total examples (necessary for evaluation and few-shot fine-tuning). Finally, any task that falls under a training task category but was held out because it is from an adversarial source (held-out domain) is used for the adversarial (domain generalization) test set, similarly filtering for 140 minimum samples. The task categories and task counts for each evaluation set are shown in Table 2.

Unique ID

r_k
```
task_040_qasc:
```
x
```
Fact: rain helps plants to survive.
```

Instruction

r_k
```
Turn the given fact into a question by a
simple rearrangement of words. This
typically involves replacing some part of
the given fact with a WH word.
```
x
```
Fact: rain helps plants to survive.
```

Context

r_k
```
input: Fact: pesticides can harm animals.
output: What can harm animals?
input: Fact: rain can help form soil.
output: Rain can help form?
input:
```
x
```
Fact: rain helps plants to survive.
```

Figure 5: Examples of different types of r_k which may specify the task k to a language model, F_θ, at the input. The language model then generates from the conditional distribution $F_\theta(y|r_k; x)$, where ; denotes concatenation.

B Model & Training Details

For all of our experiments, we use the LM-Adapted T5-Large model checkpoint found at https://huggingface.co/models?other=t5-lm-adapt. These are T5-architecture models trained exclusively on the C4 Language Modeling corpus (i.e. no additional supervised data, unlike the traditional T5 model) on the T5 denoising objective, and then fine-tuned on C4 using a standard, auto-regressive language modeling objective. As a result, these models have stronger in-context learning capabilities out of the box, and are similarly much more suited to pattern-based fine-tuning and instruction-tuning, which both leverage auto-regressive training objectives.

When performing instruction-tuning, we train

Seen Categories	Num. Tasks
Text Categorization	24
Speaker Identification	9
Question Generation	49
Sentiment Analysis	29
Misc.	29
Question Answering	150
Text Matching	15
Program Execution	12
Summarization	11
Question Understanding	11
Fact Verification	2
Gender Classification	7
Information Extraction	15
Poem Generation	1
Sentence Composition	15
Story Composition	9
Discourse Connective Identification	1
Named Entity Recognition	11
Textual Entailment	19
Text Completion	9
Wrong Candidate Generation	15
Commonsense Classification	23
Paraphrasing	5
Dialogue Generation	11
Explanation	4
Coherence Classification	6
Linguistic Probing	9
Pos Tagging	8
Stereotype Detection	7
Punctuation Error Detection	1
Text Simplification	4
Word Semantics	10
Sentence Ordering	3
Code to Text	4
Fill in The Blank	6
Text Quality Evaluation	4
Answer Verification	3
Intent Identification	4
Dialogue State Tracking	4
Text to Code	12
Number Conversion	2
Spam Classification	1
Word Relation Classification	4
Stance Detection	2
Speaker Relation Classification	2
Grammar Error Detection	1
Preposition Prediction	1
Negotiation Strategy Detection	7
Style Transfer	2
Discourse Relation Classification	1
Question Decomposition	2
Sentence Perturbation	4
Sentence Compression	1
Entity Relation Classification	1
Translation	2
Sentence Expansion	1
Entity Generation	1
Toxic Language Detection	1

Table 1: All training categories.

Unseen Categories: Language Understanding	Num Tasks
Coreference Resolution	14
Data to Text	9
Question Rewriting	11
Title Generation	18
Dialogue Act Recognition	7
Answerability Classification	11
Keyword Tagging	5
Overlap Extraction	2
Word Analogy	8
Cause Effect Classification	7
Grammar Error Correction	1

Table 2: Our Unseen Task Evaluation Setup, leveraging the Task category taxonomy presented in SuperNatural-Instructions.

all models for 3 epochs, using a batch-size of 16 and a learning-rate of $5e{-}4$, using a linear learning rate with a warm-up period of 500 steps. We train on the standard conditional generation loss, using cross-entropy to minimize the NLL of the label y given the input x and the task context r_k.

For LoRA fine-tuning, we use the PEFT implementation (https://huggingface.co/docs/peft/index). We fine-tune for 5 epochs, using a constant learning rate of $1e{-}3$ and a batch-size of 32. We set r to 16, alpha to 32, and we use a dropout of 0.05.

For Reverse Annealing, we begin with an initial temperature of 1 and linearly increase that temperature up to $1,000$. This was chosen, based on validation data, as the best maximum temperature out of $\{100, 1,000, 10,000, 100,000\}$. For DWA, we use a temperature of 1 for the softmax, which was chosen, based on validation data, as the best temperature out of $\{0.01, 0.1, 1, 10, 100, 1,000, 10,000, 100,000\}$.