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

Qinyuan Ye
qinyuany@usc.edu

Advisor: Prof. **Xiang Ren**
xiangren@usc.edu

Department of Computer Science, University of Southern California. *Jan 11, 2022*

Motivation

- Humans can learn a new task ***efficiently*** with only few examples, by leveraging their knowledge obtained when learning prior tasks.

???

$$\int_0^1 x = \frac{1}{2}(1^2 - 0^2) = \frac{1}{2}$$

$$\int_1^2 x = \frac{1}{2}(2^2 - 1^2) = \frac{3}{2}$$

$$\int_2^3 x = ?$$

Studied **counting**, **arithmetic**, **fraction**, **geometry**, ..., **physics**, **geography**, ...
Done a lot of puzzles, brain teasers, crosswords, ...

$$\int_0^1 x = \frac{1}{2}(1^2 - 0^2) = \frac{1}{2}$$

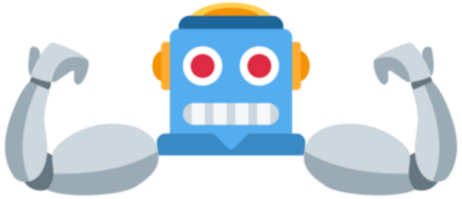
$$\int_1^2 x = \frac{1}{2}(2^2 - 1^2) = \frac{3}{2}$$

$$\int_2^3 x = \frac{1}{2}(3^2 - 2^2) = \frac{5}{2}$$

Motivation

- Humans can learn a new task ***efficiently*** with only few examples, by leveraging their knowledge obtained when learning prior tasks.
- In this work, we refer to this ability as ***cross-task generalization***.
- We explore whether and how such ability can be ***acquired***, and further ***applied*** to build better few-shot learners across ***diverse NLP tasks***.

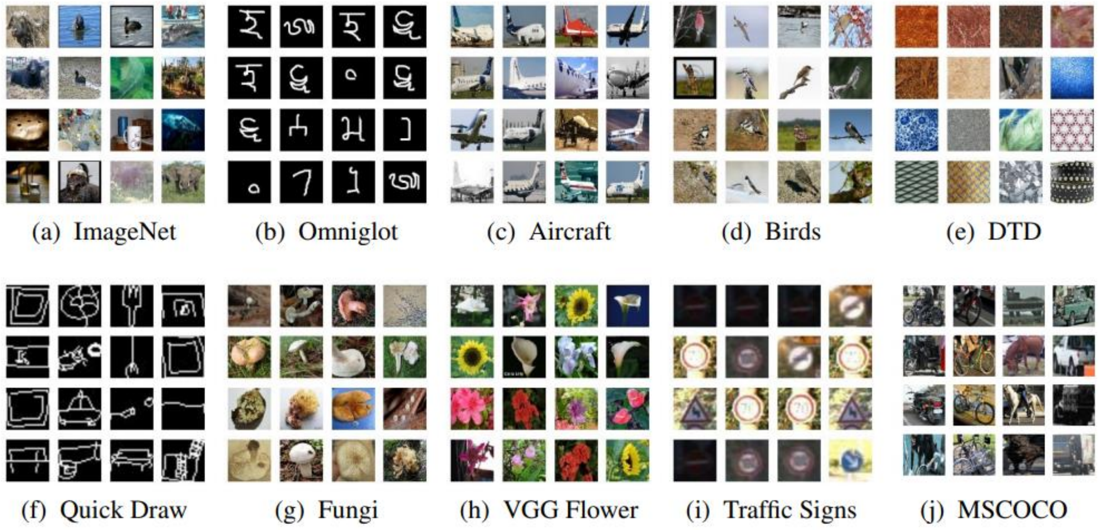
Studied **sentiment classification**, **topic classification**, **reading comprehension**



Summarize the following article: USC will move the first week of spring semester classes online, Provost and Senior Vice President ...

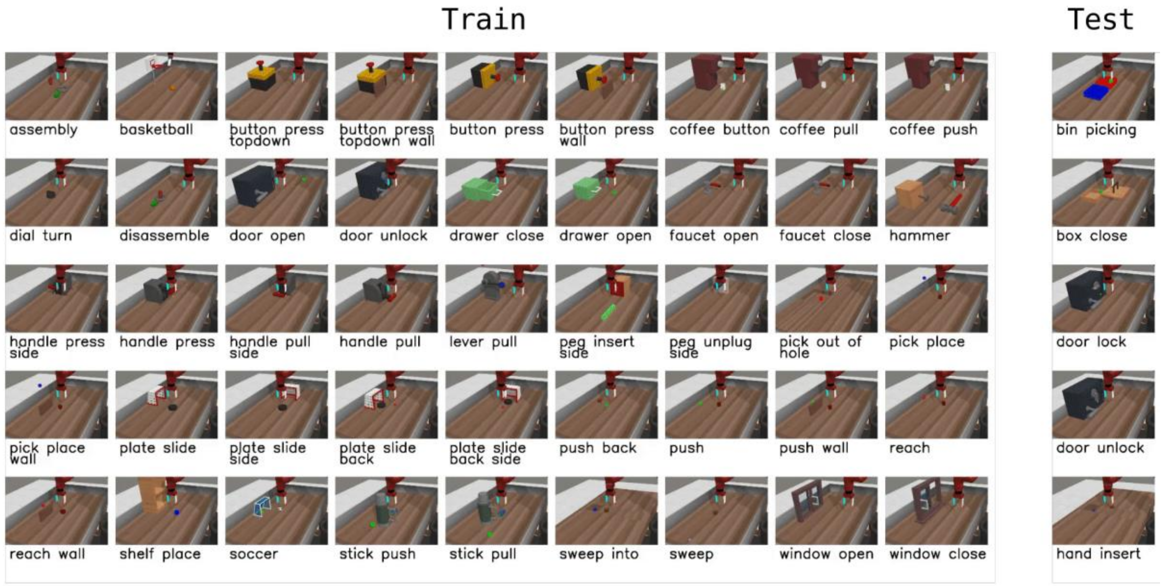
Prior Work

Meta-learning in Computer Vision



***Meta-Dataset**: A Dataset of Datasets for Learning to Learn from Few Examples*
Triantafillou et al., 2020

Meta-learning in Robotics



***Meta-World**: A Benchmark and Evaluation for Multi-task and Meta Reinforcement Learning*
Yu et al., 2019

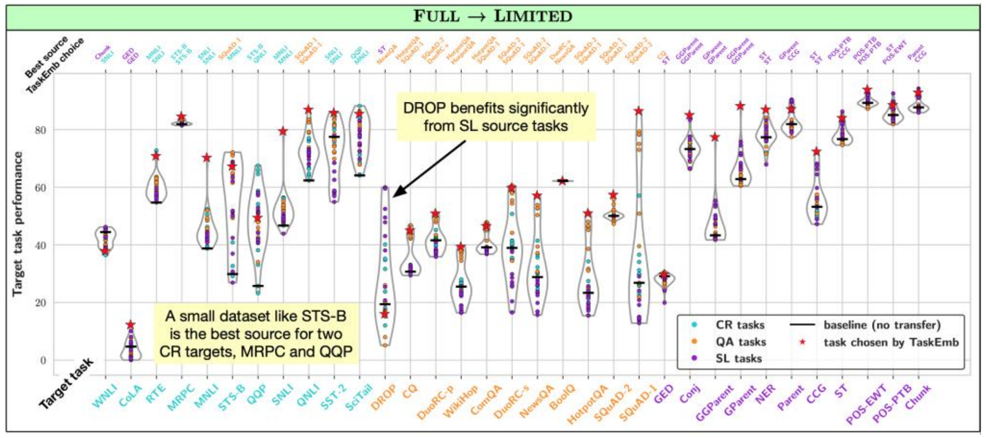
Prior Work

Intermediate Task Transfer in NLP

Supplementary Training on Intermediate Labeled-data Tasks (STILT)
(Phang et al., 2018)

Model	RTE accuracy
GPT → RTE	54.2
GPT → MNLI → RTE	**70.4**
GPT → {MNLI, RTE}	68.6
GPT → {MNLI, RTE} → RTE	67.5

Exploring and Predicting Transferability across NLP Tasks
(Vu et al., 2020)



Mainly focusing on ***one-to-one*** transfer: ***one*** source task, ***one*** target task

In this work

We are interested in having ***multiple source tasks***.

Prior Work

Meta-learning in NLP

Few-shot Relation Classification
(Han et al., 2018, Gao et al., 2019)

Train	(country, father, director) (residence, characters, instrument)
Test	(creator, cast member, author)

Tasks are ***synthetic***

Few-shot Learning Across NL Classification Tasks
(Bansal et al., 2020)

Train	SST-2, CoLA, MNLI QNLI, QQP, RTE
Test	SciTail, Amazon Review (Books)

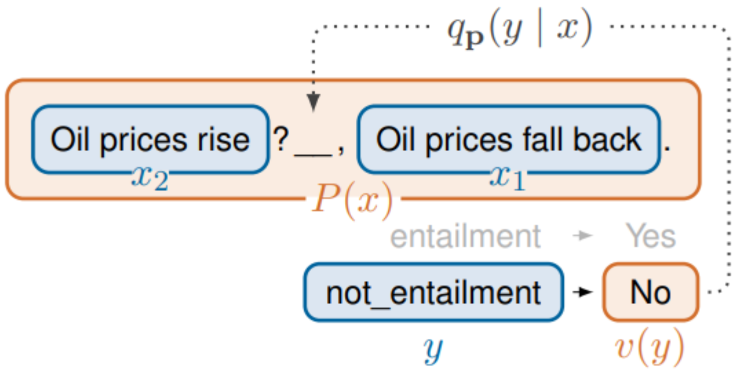
Tasks are drawn from a rather ***narrow*** distribution

In this work

Tasks have ***diverse formats and goals***, to simulate the real human learning environment

Prior Work

Few-shot Fine-tuning

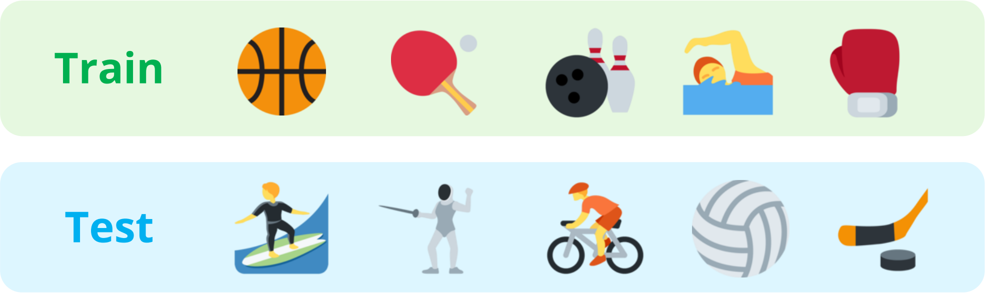


Small Language Models Are Also Few-Shot Learners
Schick and Schütze, 2020

Better **Instance-level Generalization**

Generalize from a few ***seen training instances***,
To multiple ***unseen test instances***.

In this work

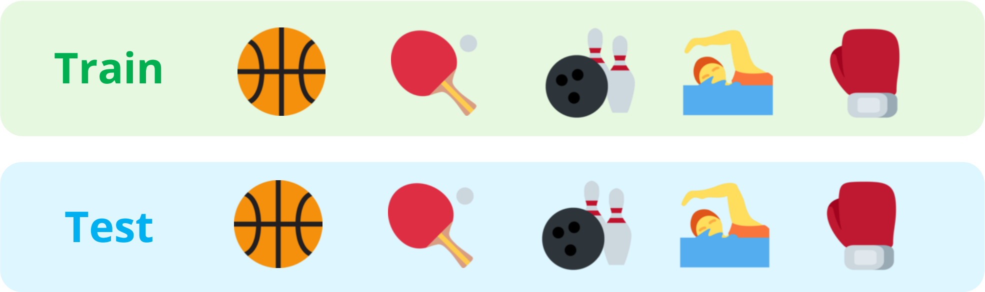


Better **Cross-task Generalization**

Generalize from several ***seen tasks***,
To ***unseen tasks***.

Prior Work

Multi-task Pre-finetuning

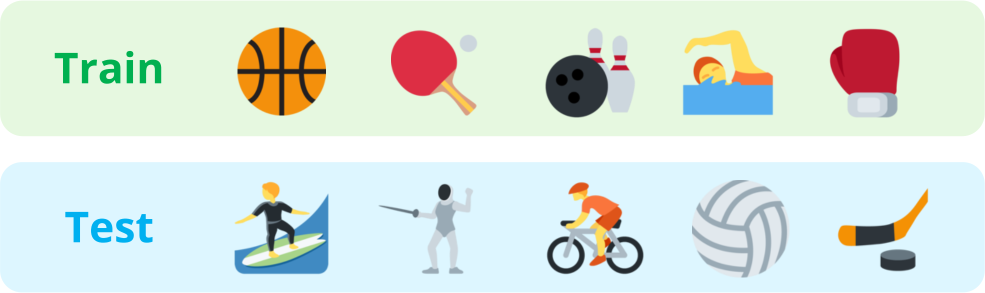


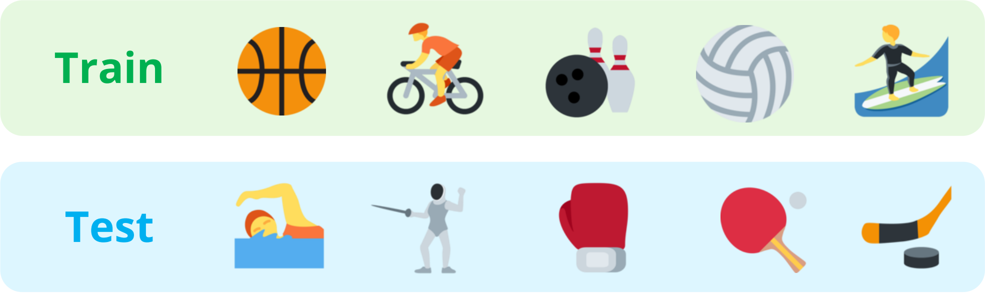
Muppet
(Aghajanyan et al., 2021)

Test tasks are typically seen during training.
Investigating implementation (parallel training and loss scaling)

In this work

Train tasks and test tasks are non-overlapping.
We are also interested in how different task partitions influence the results.





Defining "Tasks"

- The meaning of "task" is overloaded. "Tasks" can be categorized at different granularity.
  - Classification vs. QA
  - Yes/No QA vs. machine reading comprehension
  - QA in science domain vs. QA in news domain
- We take a general formulation by defining a "task" with its training and testing examples.
  - i.e., **A task T is a tuple of $(D_{train}, D_{dev}, D_{test})$**

Task D_{train} D_{dev} D_{test}

Defining “Tasks”

- We’re interested in cross-task generalization -- generalization to novel tasks.
- We need to partition all tasks into seen tasks and unseen tasks.

Seen

Train Tasks T_{train}

- Task 1 D_{train} D_{dev} D_{test}
- Task 2 D_{train} D_{dev} D_{test}
- Task 3 D_{train} D_{dev} D_{test}
- Task 4 D_{train} D_{dev} D_{test}
- Task 5 D_{train} D_{dev} D_{test}
- Task 6 D_{train} D_{dev} D_{test}

Development

Dev Tasks T_{dev}

- Task 1 D_{train} D_{dev} D_{test}
- Task 2 D_{train} D_{dev} D_{test}
- Task 3 D_{train} D_{dev} D_{test}

Unseen

Test Tasks T_{test}

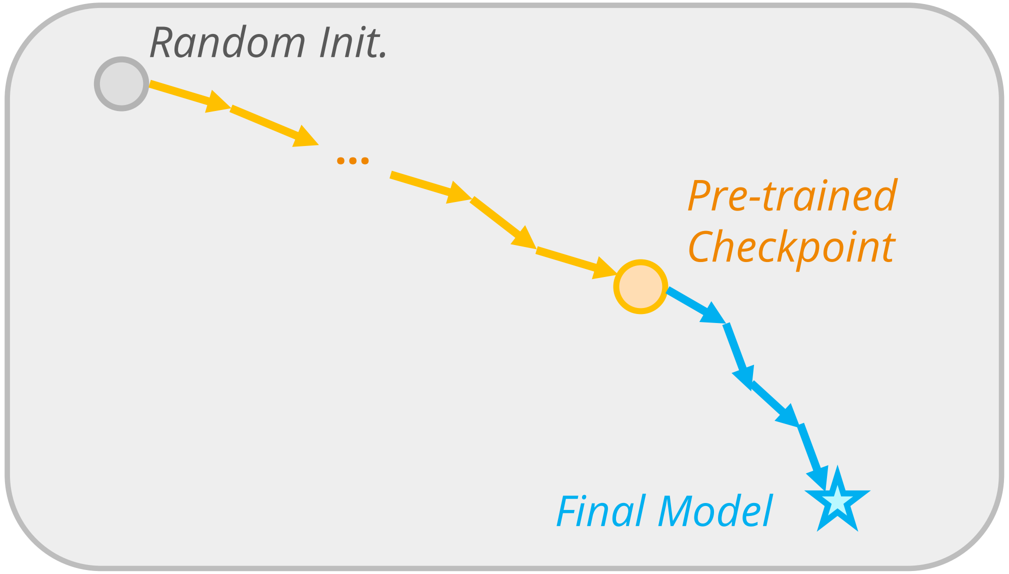
- Task 1 D_{train} D_{dev} D_{test}
- Task 2 D_{train} D_{dev} D_{test}
- Task 3 D_{train} D_{dev} D_{test}

Problem Setting

Prevalent Pipeline

Large-scale Pre-training

+ Downstream Fine-tuning



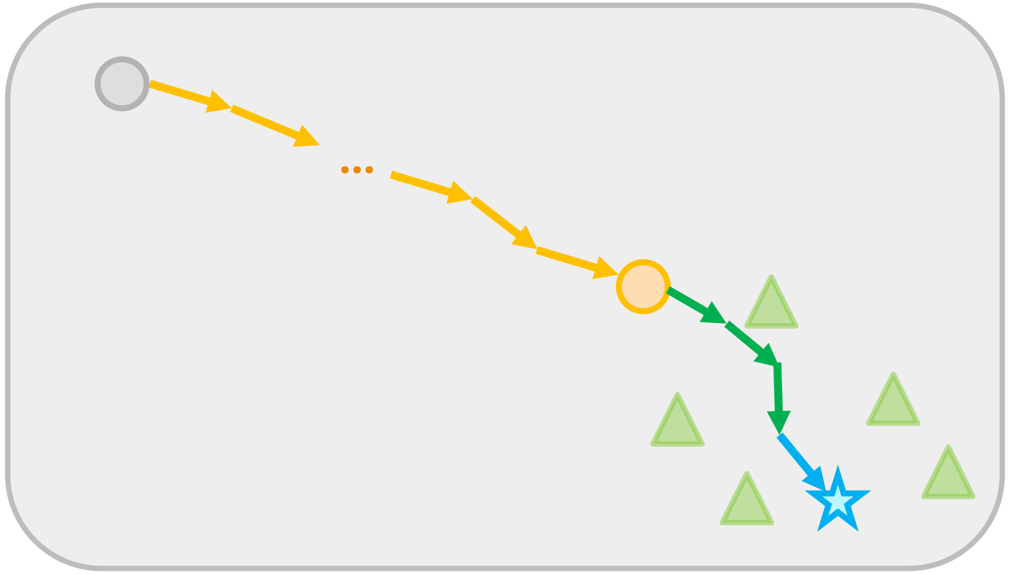
Parameter Space

In our CrossFit Setting

Large-scale Pre-training

+ Upstream Learning on a set of seen tasks

+ Downstream Fine-tuning on an unseen target task



Parameter Space

- **Evaluation Metric**
  - We define ***Average Relative Gain*** (ARG), to measure the overall performance gain on all unseen tasks.
  - ARG is the relative performance changes before and after the upstream learning stage for each test task, and averaged across all test tasks.
  - ***This is not a perfect metric***, but it helps us to get a general sense. We still plot and report relative gain for individual tasks.

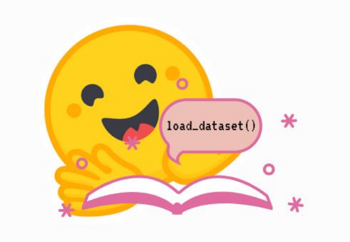
Example

(40%-25%)/2=7.5%

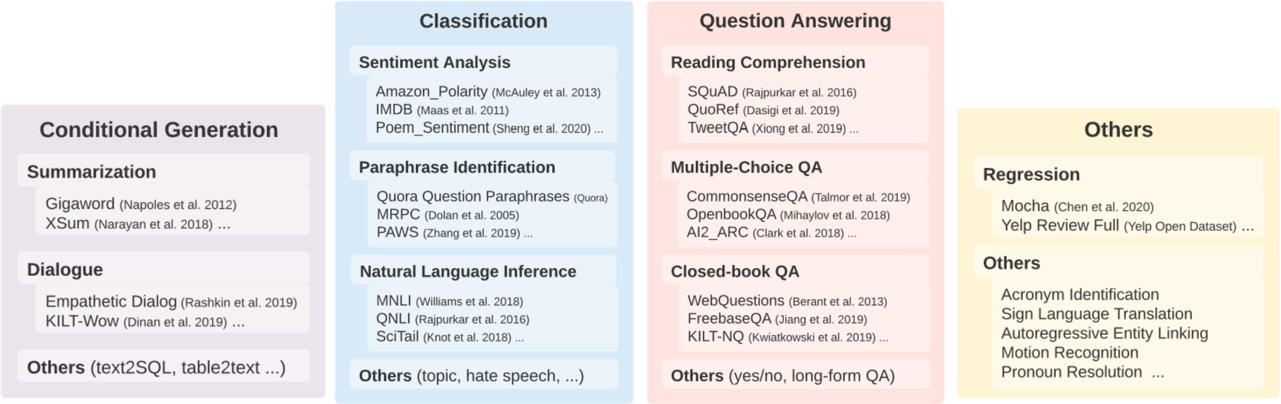
	Direct FT	*Upstream + FT*	*Rel. Gain*	*ARG*
Task A	50% F1	70% F1	40%	7.5%
Task B	40% Acc.	30% Acc.	-25%	

Tasks and Partitions

- To instantiate different settings in **CrossFit** and facilitate in-depth analysis ...
- We present **NLP Few-shot Gym**, a repository of **160 diverse few-shot NLP tasks**.
  - Gathered from open-source datasets on **Hugging Face Datasets**
  - Converted to a **unified text-to-text format**
  - 16 examples per class for classification tasks; 32 examples for other tasks
  - **Reproducible** with our released code (**https://github.com/INK-USC/CrossFit**)



- We manually create **a task ontology** with categories and sub-categories

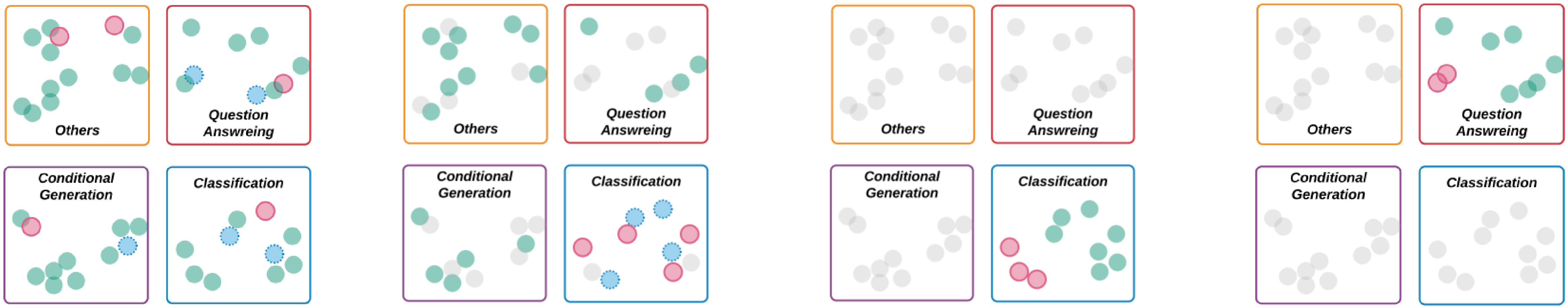


Tasks and Partitions

- **Partitions of train/dev/test tasks**

Training Task   Dev Task   Test Task   Unused Task

The locations and distances in these figures are hypothetical and for illustrative purposes only.



Partition 1: Random
Randomly split 160 tasks into 120/20/20 for train/dev/test tasks.

Partition 2.1: 45non-class
Train: 45 non-classification tasks
Dev/Test: 10 classification tasks

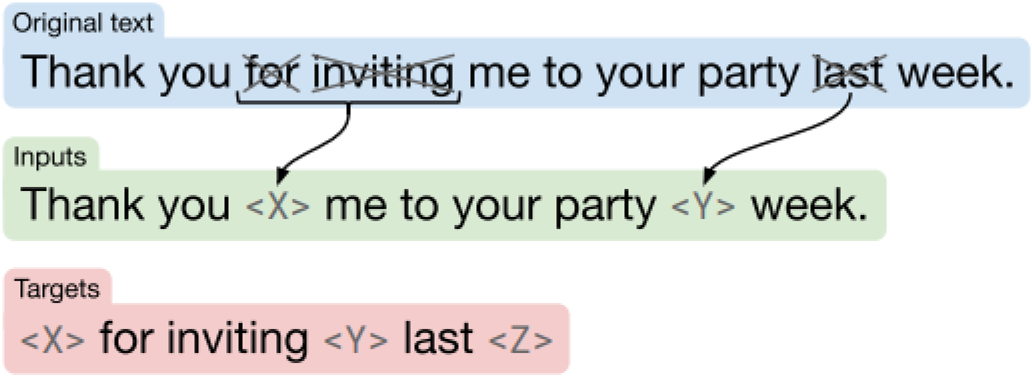
Partition 3.1: Held-out-NLI
Train: 57 non-NLI classification tasks
Test: 8 NLI tasks

Partition 4.1: Held-out-MRC
Train: 42 non-MRC QA Tasks
Test: 9 MRC QA tasks

Here we present 4 partitions. We have 8 in total in the paper.

Experiments

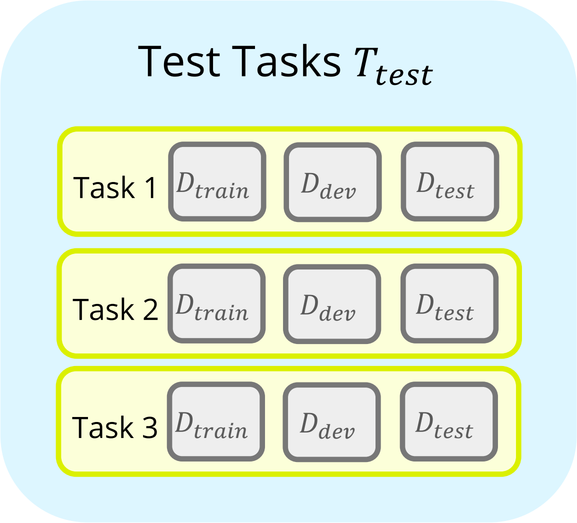
- We mainly use **BART-Base** (Lewis et al., 2020) as the main model for our analysis.
  - Also we verify some of our findings with **BART-Large** and **T5-v1.1-Base** (Raffel et al., 2019)
- These are off-the-shelf transformer models, pre-trained on large corpus with masked language modeling or similar objectives.



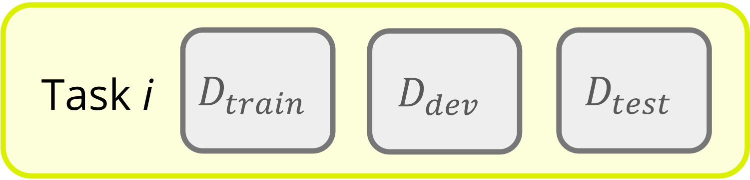
Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Raffel et al., 2019

Experiments

- We mainly use **BART-Base** (Lewis et al., 2020) as the main model for our analysis.
  - Also we verify some of our findings with **BART-Large** and **T5-v1.1-Base** (Raffel et al., 2019)
- Methods
  - **Downstream Fine-tuning** (also used as the baseline for computing ARG)



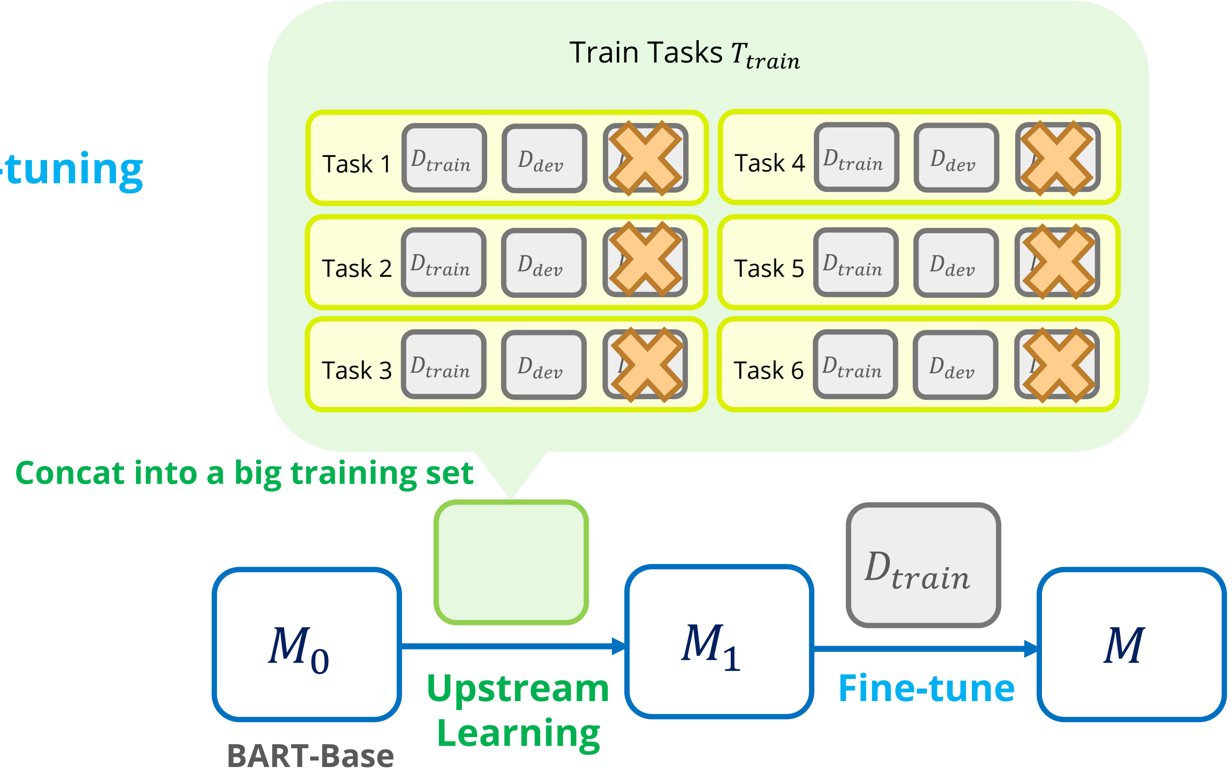
For each task in T_{test}



Fine-tune on D_{train}
Validate on D_{dev}
Report performance on D_{test}

Experiments

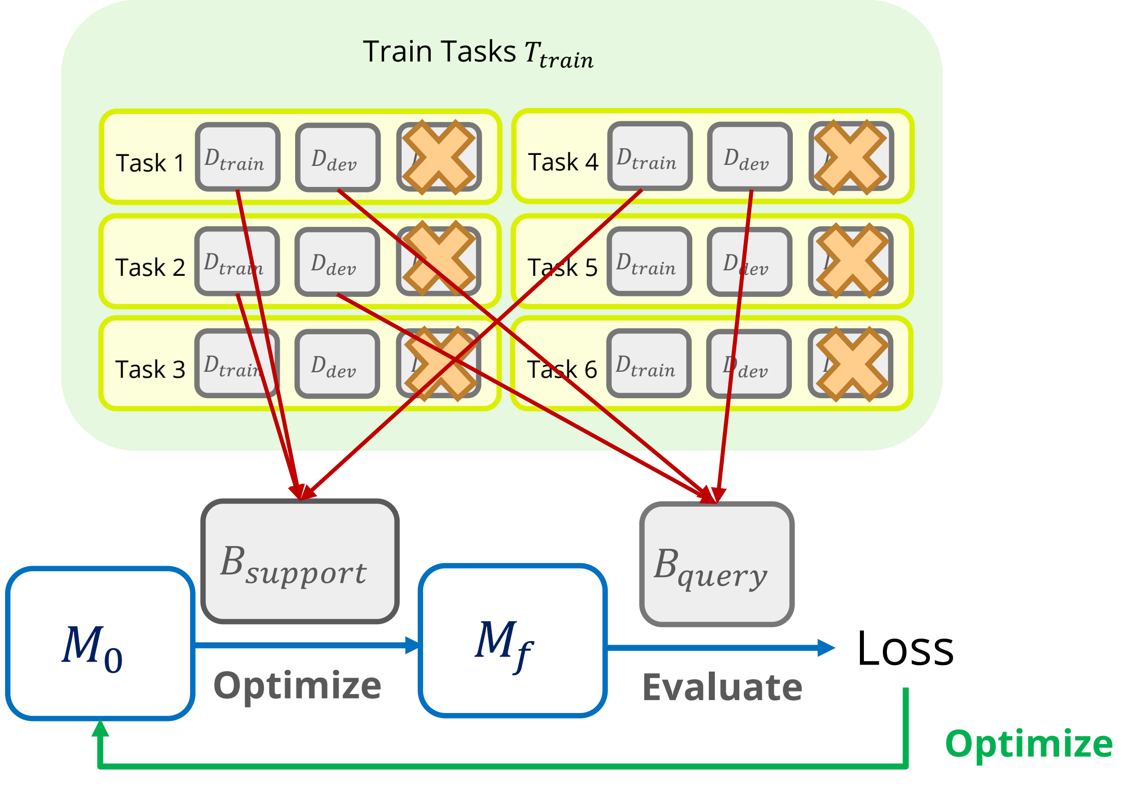
- We mainly use **BART-Base** (Lewis et al., 2020) as the main model for our analysis.
  - Also we verify some of our findings with **BART-Large** and **T5-v1.1-Base** (Raffel et al., 2019)
- Methods
  - **Downstream Fine-tuning**
  - **Upstream Learning then Downstream Fine-tuning**
    - Multi-task Learning



Experiments

- We mainly use **BART-Base** (Lewis et al., 2020) as the main model for our analysis.
  - Also we verify some of our findings with **BART-Large** and **T5-v1.1-Base** (Raffel et al., 2019)
- Methods
  - **Downstream Fine-tuning**
  - **Upstream Learning then Downstream Fine-tuning**
    - Multi-task Learning
    - Model Agnostic Meta-learning (Finn et al., 2017)

One update in upstream learning with MAML



Experiments

- We mainly use **BART-Base** (Lewis et al., 2020) as the main model for our analysis.
  - Also we verify some of our findings with **BART-Large** and **T5-v1.1-Base** (Raffel et al., 2019)
- Methods
  - **Downstream Fine-tuning**
  - **Upstream Learning then Downstream Fine-tuning**
    - Multi-task Learning
    - Model Agnostic Meta-learning (Finn et al., 2017)
    - First-order MAML
    - Reptile (Nichol et al., 2017)

Variants of MAML

Quick Summary

NLP Few-shot Gym

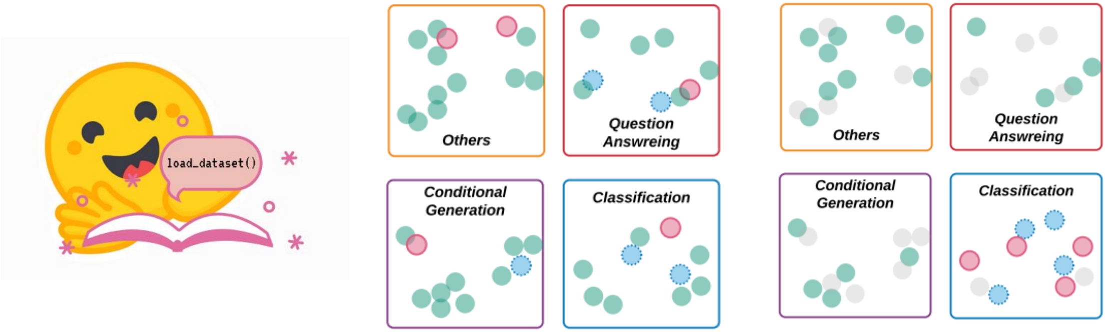
- Gather **160 diverse few-shot tasks** in text-to-text format
- Manually classify the tasks into categories and sub-categories.
- Design **8 partitions** of the tasks to test cross-task generalization in different scenarios

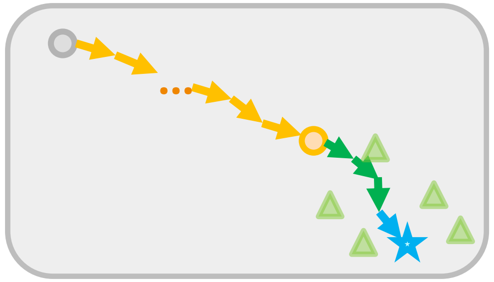
CrossFit Setting

Large-scale Pre-training

\+ Upstream Learning on a set of seen tasks (T_{train})

\+ Downstream Fine-tuning on an unseen target task (T_{test})





Parameter Space

Using **multi-task learning** and **meta-learning** methods (e.g., MAML, Reptile)

Key Findings

- **Q1. Can we teach pre-trained LMs to generalize across tasks with an upstream learning stage?**

Evidence 1

ARG (defined earlier) is ***positive*** for all 8 partitions and all 4 upstream learning methods

No.	Shorthand	ARG(Multi)	ARG(MAML)	ARG(FoMAML)	ARG(Rept.)
1	Random	35.06%	28.50%	22.69%	25.90%
2.1	45cls	11.68%	9.37%	10.28%	13.36%
2.2	23cls+22non-cls	11.82%	9.69%	13.75%	14.34%
2.3	45non-cls	11.91%	9.33%	11.20%	14.14%
3.1	Held-out-NLI	16.94%	12.30%	12.33%	14.46%
3.2	Held-out-Para	18.21%	17.90%	21.57%	19.72%
4.1	Held-out-MRC	32.81%	27.28%	28.85%	28.85%
4.2	Held-out-MCQA	12.20%	4.69%	6.73%	7.67%

Evidence 2

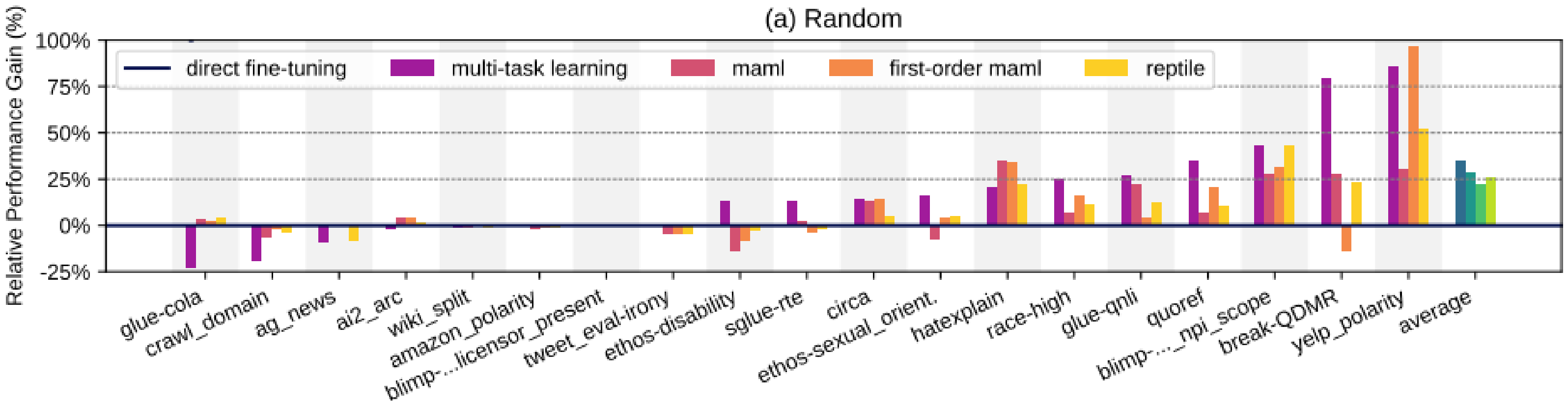
When we aggregate test tasks performance gain from all upstream learning methods and partitions…

⬆	***>5% relative gain***	***51.47%***
➡	***within ±5%***	***35.93%***
⬇	***<-5% relative gain***	***12.60%***

Yes! Upstream learning methods do help pre-trained LMs to acquired cross-task generalization!

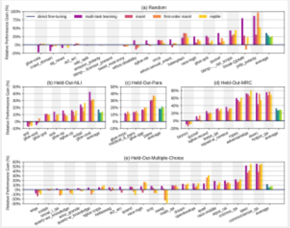
Key Findings

- **Q1. Can we teach pre-trained LMs to generalize across tasks with an upstream learning stage?**



height: relative performance gain with vs. without upstream learning

Full results in the paper



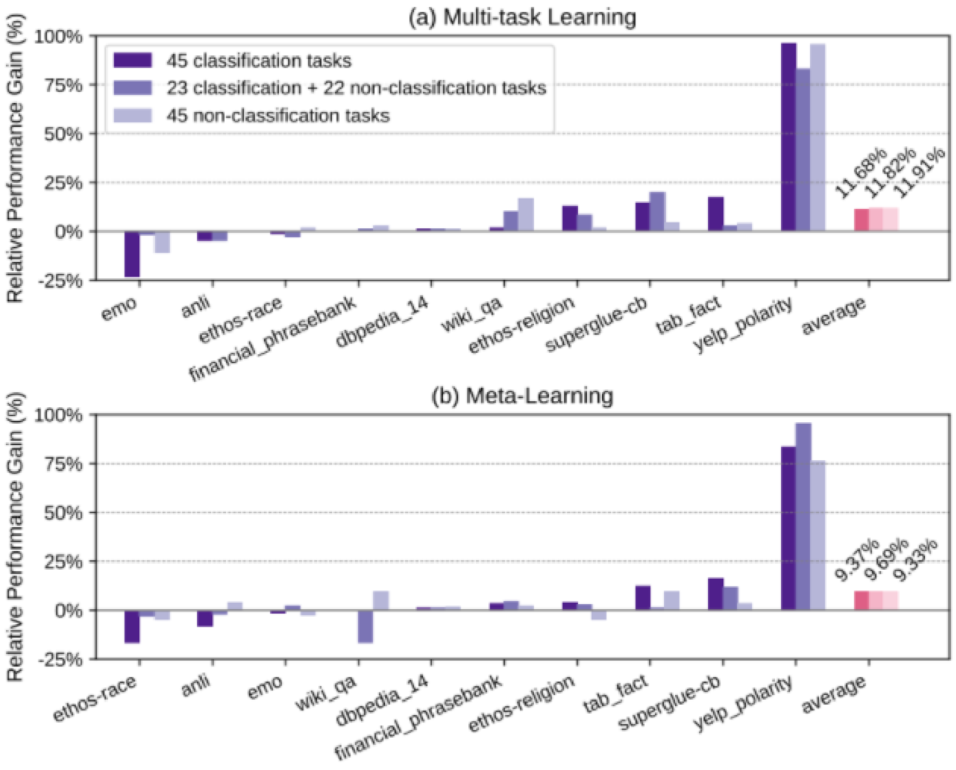
- Correlated Performance Gains
  - Tasks that benefit from one upstream method are likely to also benefit from another upstream learning method.
- Multi-task learning is a strong baseline
  - Outperforms in meta-learning algorithms in most settings. We suspect complex optimization for transformer models is too challenging.
- Forgetting Pre-Trained Knowledge
  - Tasks that resemble the pre-training objective (masked language modeling) is likely to get negative performance gain after upstream learning.

Key Findings

- **Q2. “Well-rounded” or “specialized”? How to select tasks during upstream learning?**

- We conduct ***controlled experiments*** by fixing the test tasks to be 10 classification tasks.
- The upstream tasks are
  - **100% classification tasks**
  - **50% classification + 50% non-classification tasks**
  - **100% non-classification tasks**

- Classification tasks and non-classification tasks seem to be equivalently helpful.
- **Our understanding of tasks may not align with how models learn transferable skills**.

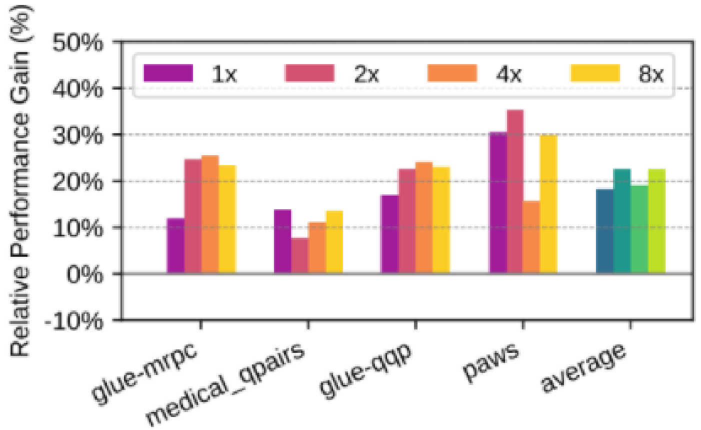


Key Findings

- **Q3. Does it help if we have more labelled data for *upstream* tasks?**

- In previous experiments, we limit the number of examples in each upstream task
  - Classification tasks: 16 examples per class
  - Non-classification tasks: 32 examples
- We experiment with using 2x, 4x, 8x data in *upstream* task …

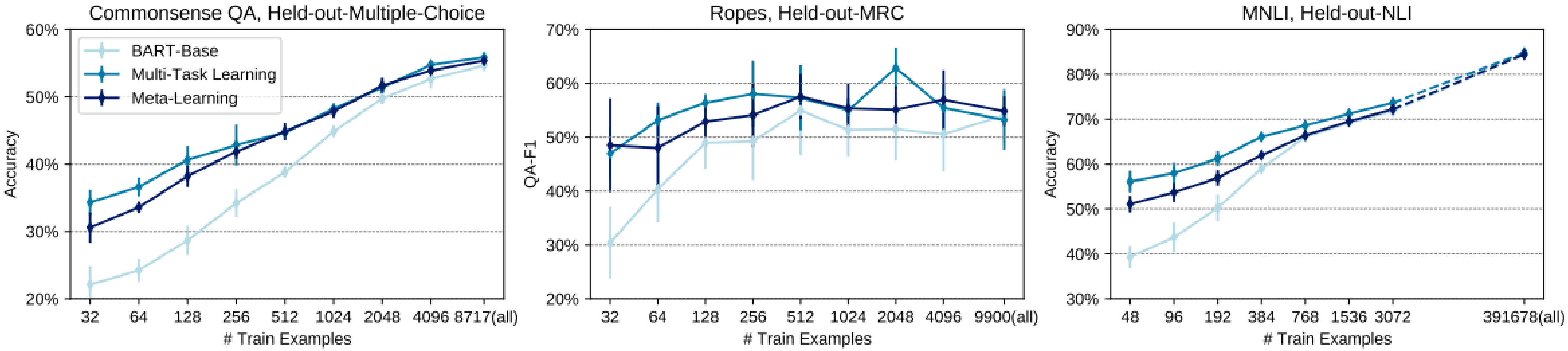
- We find that the effect from using more upstream data is inconsistent on different target tasks.
- **More examples in each upstream task does not necessarily lead to better cross-task generalization.**



On a side note, in settings closely related to ours (Mishra et al., 2021; Wei et al., 2021), it is shown that the <u>number of tasks</u> is critical.

More Findings

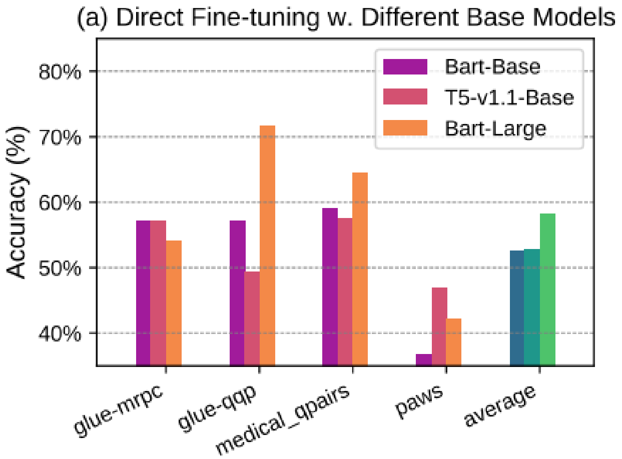
- **Q4. From Few-shot to More-shot: Does the improved cross-task generalization ability go beyond few-shot settings?**

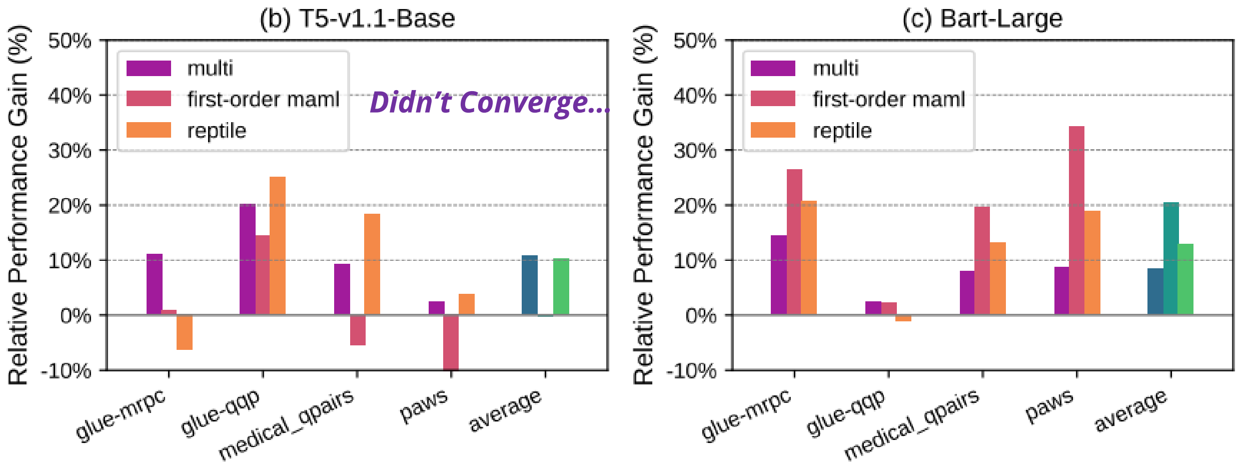


- Cross-task generalization helps *most* on CommonsenseQA, ROPES and MNLI.
- On these three datasets, the **benefits** brought by upstream learning methods **extend into medium resource cases** with up to 2048 training examples.

More Findings

- **Q5. Can we further improve few-shot performance by using different/larger pre-trained models?**





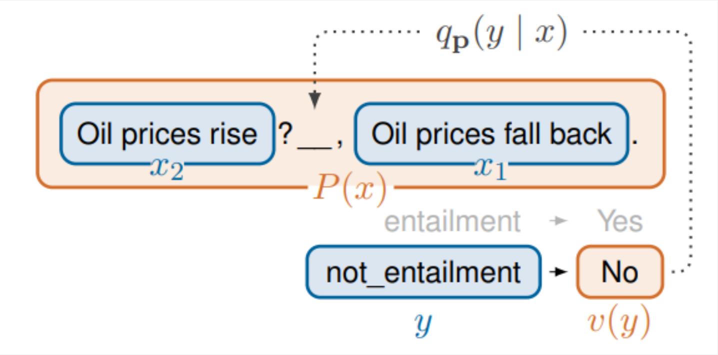
Larger pre-trained LMs are better few-shot learners by themselves.

They still benefit from acquiring cross-task generalization via upstream learning

More Findings

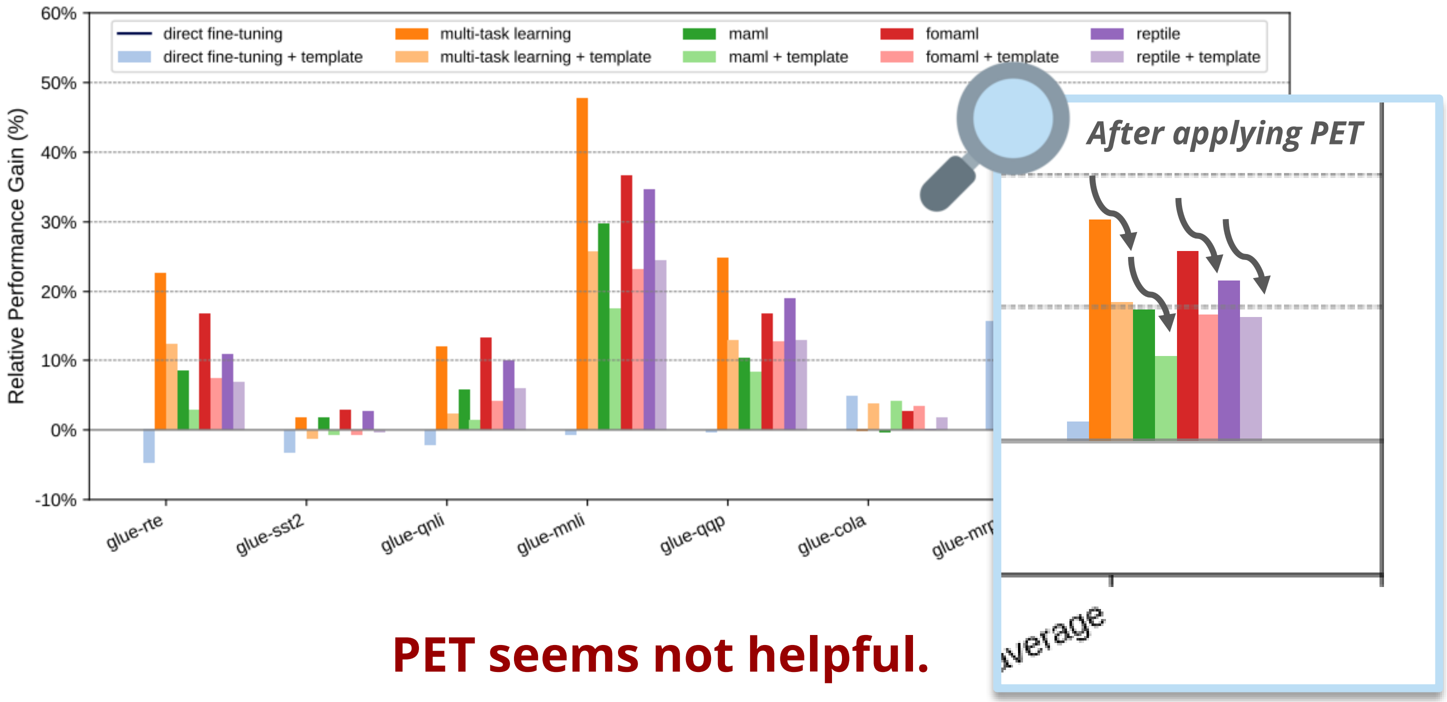
- **Q6. Can we use pattern-exploiting training (PET) to replace direct fine-tuning and achieve even better performance?**

Recall...



Small Language Models Are Also Few-Shot Learners
Schick and Schütze, 2020

Pattern-exploiting Training (PET)



PET seems not helpful.

Perhaps PET is not directly applicable to auto-regressive models?
Perhaps there is a mis-match in format? During upstream learning tasks are not in cloze-style.

Conclusions

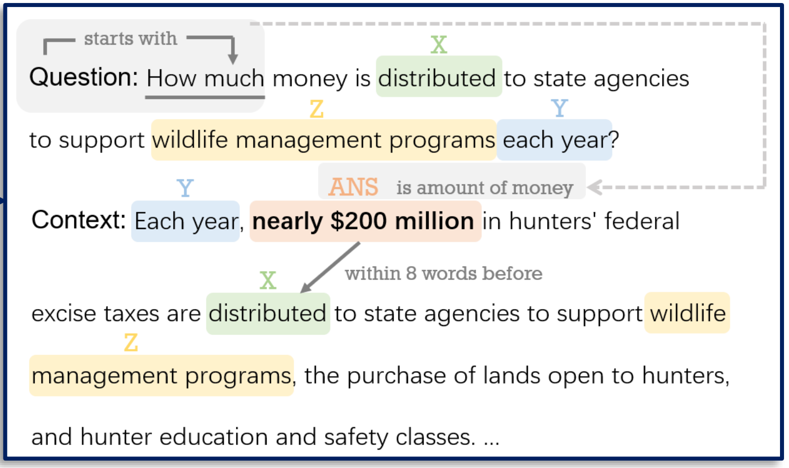
- We introduced …
    - **CrossFit** 🏋, a task setup which aims at building few-shot learners that generalize across diverse NLP tasks.
    - **NLP Few-shot Gym**, a repository of 160 NLP tasks gathered from existing open-access datasets.
- We found that …
    - **Upstream learning methods** such as multi-task learning and meta-learning help pre-trained LMs to **acquired cross-task generalization**.
    - Task similarity in terms of task format ***does not*** align with how models learn transferable skills.
    - More labelled data for upstream tasks ***does not*** necessarily lead to better cross-task generalization ability.

Future Work

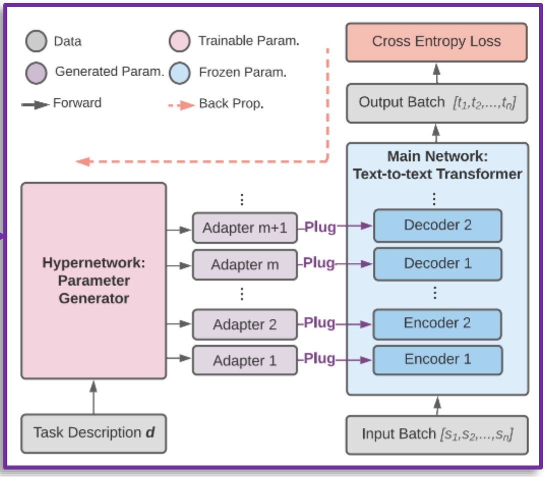
- We envision the **CrossFit** 🏋️ Challenge and the **NLP Few-shot Gym** to serve as the **testbed** for many interesting **"meta-problems"**
  - Generating Prompts? (Shin et al., 2020; Gao et al., 2020)
  - Select appropriate upstream tasks? (Zamir et al., 2018; Standley et al., 2020; Vu et al., 2020)
  - Apply task augmentation? (Murty et al., 2021)
  - Continual Learning? (Jin et al., 2021)
  - Task decomposition? (Andreas et al., 2016; Khot et al., 2021)

My PhD Progress

Reducing human annotation efforts in NLP



Teaching Machine Reading Comprehension (Findings of EMNLP 2020)

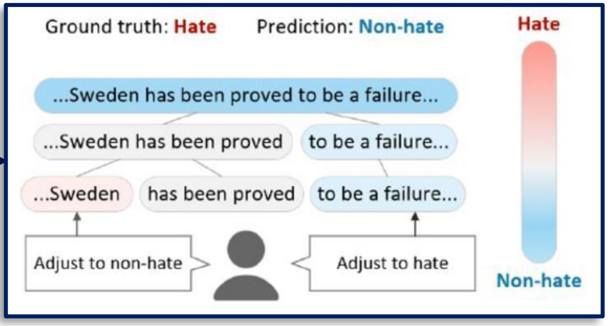


Generate Adapters from Task Instructions (ACL 2021)

Learning from Explanations

Acquiring Task-level Generalization

Refining Language Models (NeurIPS 2021; with Huihan and Ying)



Generalize from Previously Seen Tasks (This Presentation, EMNLP 2021)