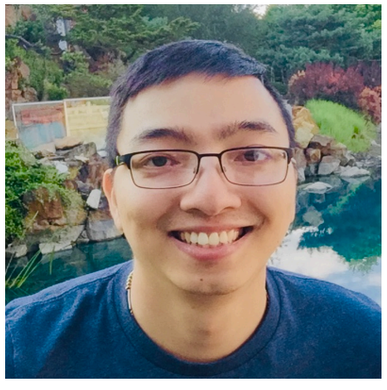
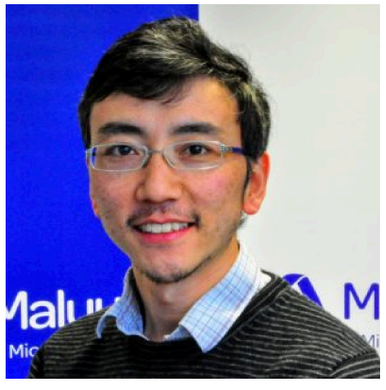
Exploring and Predicting Transferability across NLP Tasks

Library & code available at **http://github.com/tuvuumass/task-transferability**



Tu Vu[1]

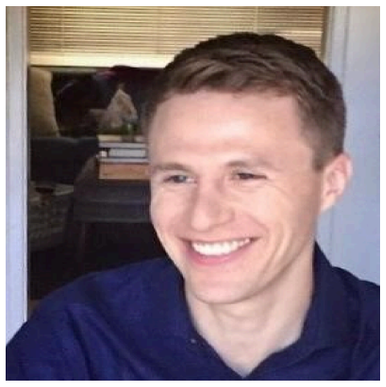


Tong Wang[2]

Tsendsuren Munkhdalai[2]

Alessandro Sordoni[2]

Adam Trischler[2]



Andrew Mattarella-Micke[3]

Subhransu Maji[1]

Mohit Iyyer[1]

UMass Amherst [1]

Microsoft Research [2]

INTUIT [3]

Overview

Task and task transferability

Motivation

Exploring task transferability

Predicting task transferability

Overview

Task and task transferability

Motivation

Exploring task transferability

Predicting task transferability

What *is a task?*

A (**dataset**, **learning objective**) pair:

- dataset $D = \{(x^i, y^i)\}_{i=1}^n$
- learning objective

  e.g., natural language inference

  description (Wang et al., 2019)

MNLI The Multi-Genre Natural Language Inference Corpus (Williams et al., 2018) is a crowd-sourced collection of sentence pairs with textual entailment annotations. Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis (*entailment*), contradicts the hypothesis (*contradiction*), or neither (*neutral*). The premise sentences are gathered from ten different sources, including transcribed speech, fiction, and government reports.

What is task transferability?

Transfer learning:
knowledge learned from a ***source*** task is transferred to facilitate learning on a ***target*** task

Task transferability:
the change in performance on the ***target*** task between learning with and without transfer from the ***source*** task

Overview

Task and task transferability

Motivation

Exploring task transferability

Predicting task transferability

Tasks can help each other!

Text classification: supplementing language model pretraining with further training on data-rich intermediate supervised tasks leads to improvements and reduced variance (Phang et al., 2018)

Question answering: pretraining on multiple related datasets leads to robust generalization and transfer (Talmor and Berant, 2019)

Sequence labeling: pretraining on a closely related task yields better performance than language model pretraining when the pretraining dataset is fixed (Liu et al., 2019)

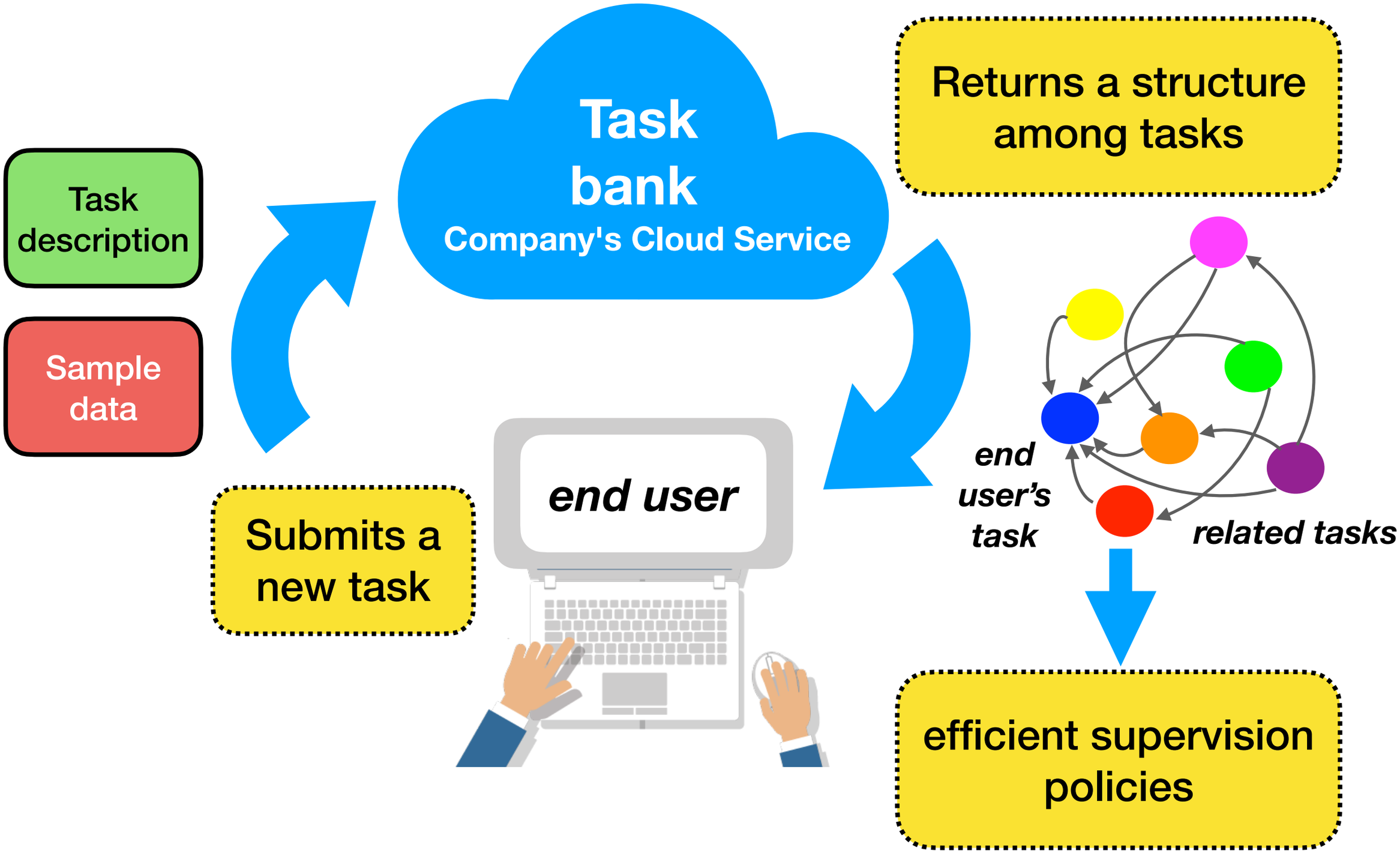
Understanding the relationships between tasks is essential!

Goal: identify similarities and beneficial relationships among tasks

Practical application:

- **efficient supervision:** reduce the need for supervision among related tasks
- **transfer learning:** select source tasks for a given target task
- **multi-task learning:** solve many related tasks in one system

A real-world scenario



The conditions for successful transfer remain opaque!

Research question 1:

Which combinations of tasks can perform well in the transfer learning setting?

- An arbitrary combination of tasks often adversely impacts target task performance (Wang et al., 2019b)

Research question 2:

Is there a principled way to predict the most transferable source tasks for a given target task?

This work

Research question 1:
Which combinations of tasks can perform well in the transfer learning setting?

➡ a ***large-scale empirical study*** across **33** different datasets to shed light on the transferability between NLP tasks

Research question 2:
Is there a principled way to predict which source tasks to use for a given target task?

➡ create ***task embeddings*** to predict the most transferable source tasks

Overview

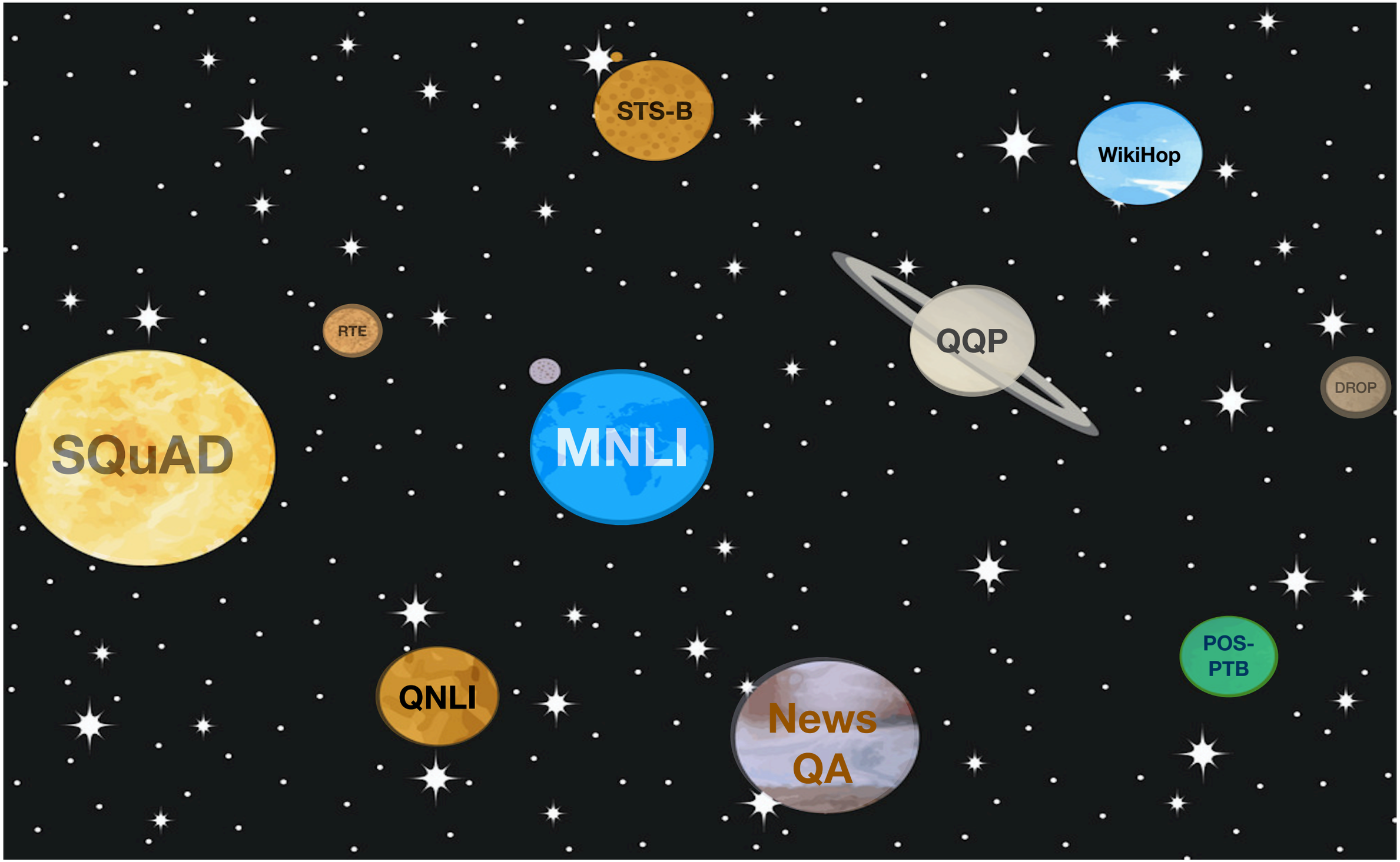
Task and task transferability

Motivation

Exploring task transferability

Predicting task transferability

Let's discover!



A large-scale empirical study

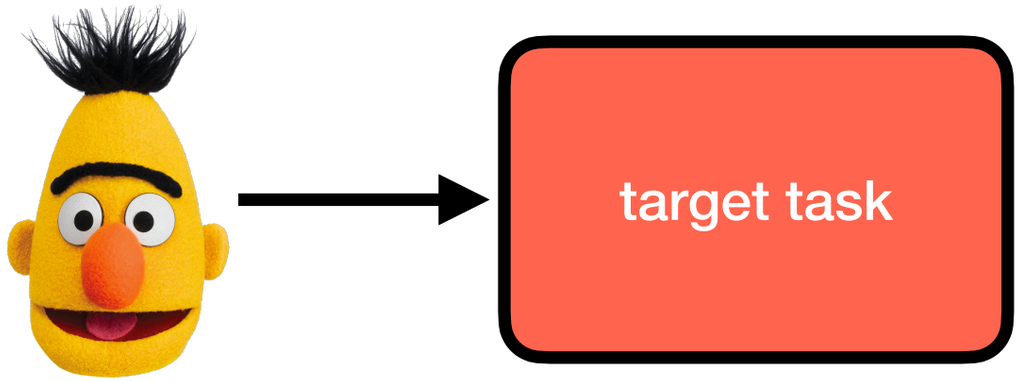
considerably more comprehensive than prior work

- **33 NLP tasks**
- **three broad classes of problems:**
  text classification/regression (CR),
  question answering (QA), and
  sequence labeling (SL)
- **three data regimes** controlled for source and target data size

➡ over **3,000 combinations** of tasks and data regimes

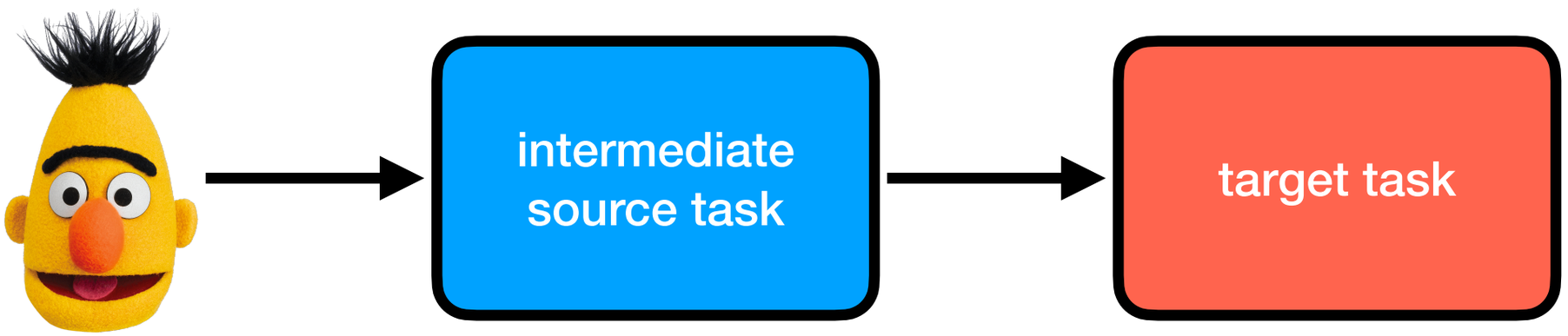
Intermediate-task transfer

Pretrained language model fine-tuning:



(Devlin et al., 2018)

Intermediate-task transfer:



(Phang et al., 2018)

Tasks

Task	|Train|
text classification/regression (CR)	
SNLI (Bowman et al., 2015)	570K
MNLI (Williams et al., 2018)	393K
QQP (Iyer et al., 2017)	364K
QNLI (Wang et al., 2019b)	105K
SST-2 (Socher et al., 2013)	67K
SciTail (Khot et al., 2018)	27K
CoLA (Warstadt et al., 2019)	8.5K
STS-B (Cer et al., 2017)	7K
MRPC (Dolan and Brockett, 2005)	3.7K
RTE (Dagan et al., 2005, et seq.)	2.5K
WNLI (Levesque, 2011)	634
sequence labeling (SL)	
ST (Bjerva et al., 2016)	43K
CCG (Hockenmaier and Steedman, 2007)	40K
Parent (Liu et al., 2019a)	40K
GParent (Liu et al., 2019a)	40K
GGParent (Liu et al., 2019a)	40K
POS-PTB (Marcus et al., 1993)	38K
GED (Yannakoudakis et al., 2011)	29K
NER (Tjong Kim Sang and De Meulder, 2003)	14K
POS-EWT (Silveira et al., 2014)	13K
Conj (Ficler and Goldberg, 2016)	13K
Chunk (Tjong Kim Sang and Buchholz, 2000)	9K

Task	|Train|
question answering (QA)	
SQuAD-2 (Rajpurkar et al., 2018)	162K
NewsQA (Trischler et al., 2017)	120K
HotpotQA (Yang et al., 2018)	113K
SQuAD-1 (Rajpurkar et al., 2016)	108K
DuoRC-p (Saha et al., 2018)	100K
DuoRC-s (Saha et al., 2018)	86K
DROP (Dua et al., 2019)	77K
WikiHop (Welbl et al., 2018)	51K
BoolQ (Clark et al., 2019)	16K
ComQA (Abujabal et al., 2019)	11K
CQ (Bao et al., 2016)	2K

Table 1: Datasets used in our experiments, grouped by task class and sorted by training dataset size.

Data regimes

Three data regimes to examine the impact of data size on **SOURCE → TARGET** transfer:

- **FULL → FULL**
- **FULL → LIMITED**
- **LIMITED → LIMITED**

FULL: all training data is used
LIMITED: randomly select **1K training examples**

Positive transfer can occur in a more diverse array of settings than previously thought

FULL → FULL

↓src,tgt→	CR	QA	SL
CR	6.3 (11)	3.4 (10)	0.3 (10)
QA	3.2 (10)	9.5 (11)	0.3 (9)
SL	5.3 (8)	2.5 (10)	0.5 (11)

FULL → LIMITED

	CR	QA	SL
CR	56.9 (11)	36.8 (10)	2.0 (10)
QA	44.3 (11)	63.3 (11)	5.3 (11)
SL	45.6 (11)	39.2 (6)	20.9 (11)

LIMITED → LIMITED

	CR	QA	SL
CR	23.7 (11)	7.3 (11)	1.1 (11)
QA	37.3 (11)	49.3 (11)	4.2 (11)
SL	29.3 (10)	30.0 (8)	10.2 (11)

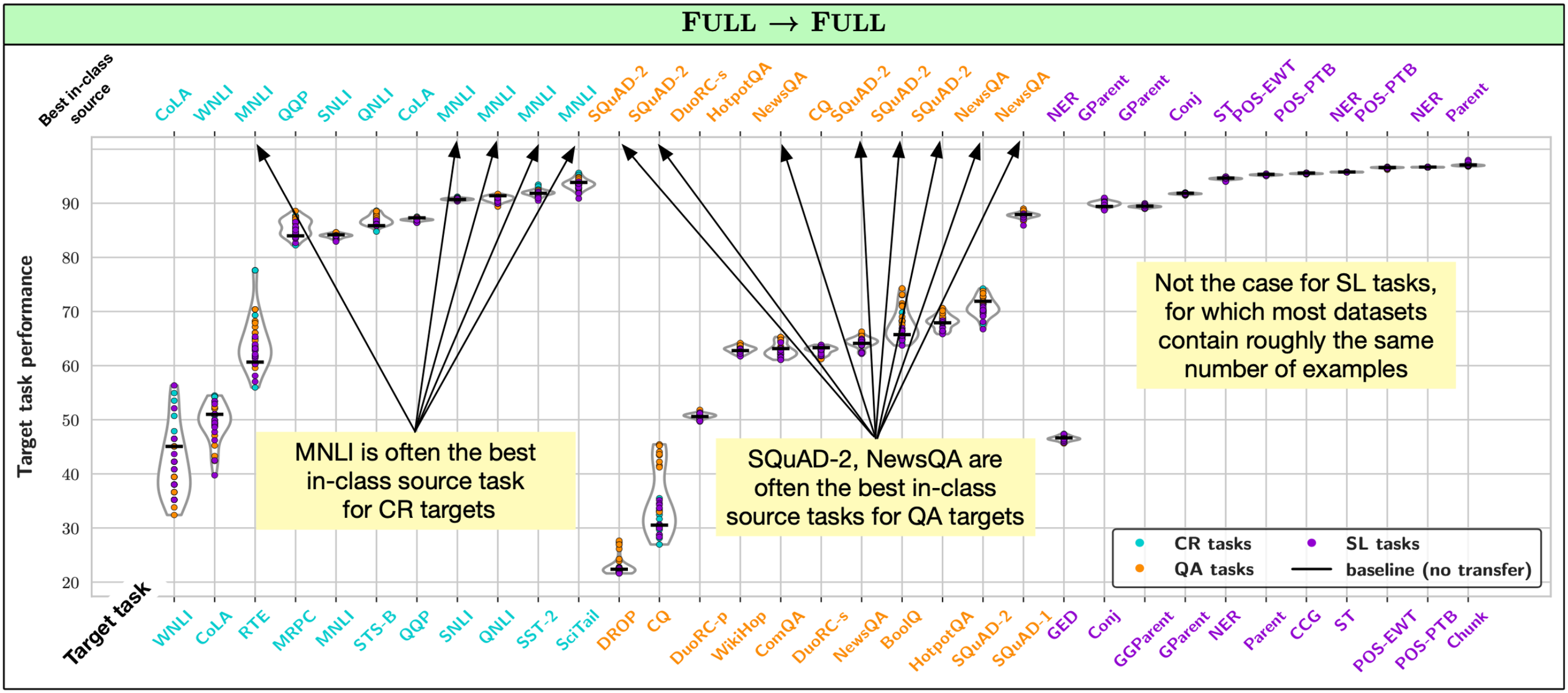
Table 2: A summary of our transfer results for each combination of the three task classes in the three data regimes. Each cell represents the relative gain of the *best* source task in the source class (row) for a given target task, averaged across all of target tasks in the target class (column). In parentheses, we additionally report the number of target tasks (out of 11) for which at least one source task results in a positive transfer gain. The diagonal cells indicate in-class transfer.

Transfer results

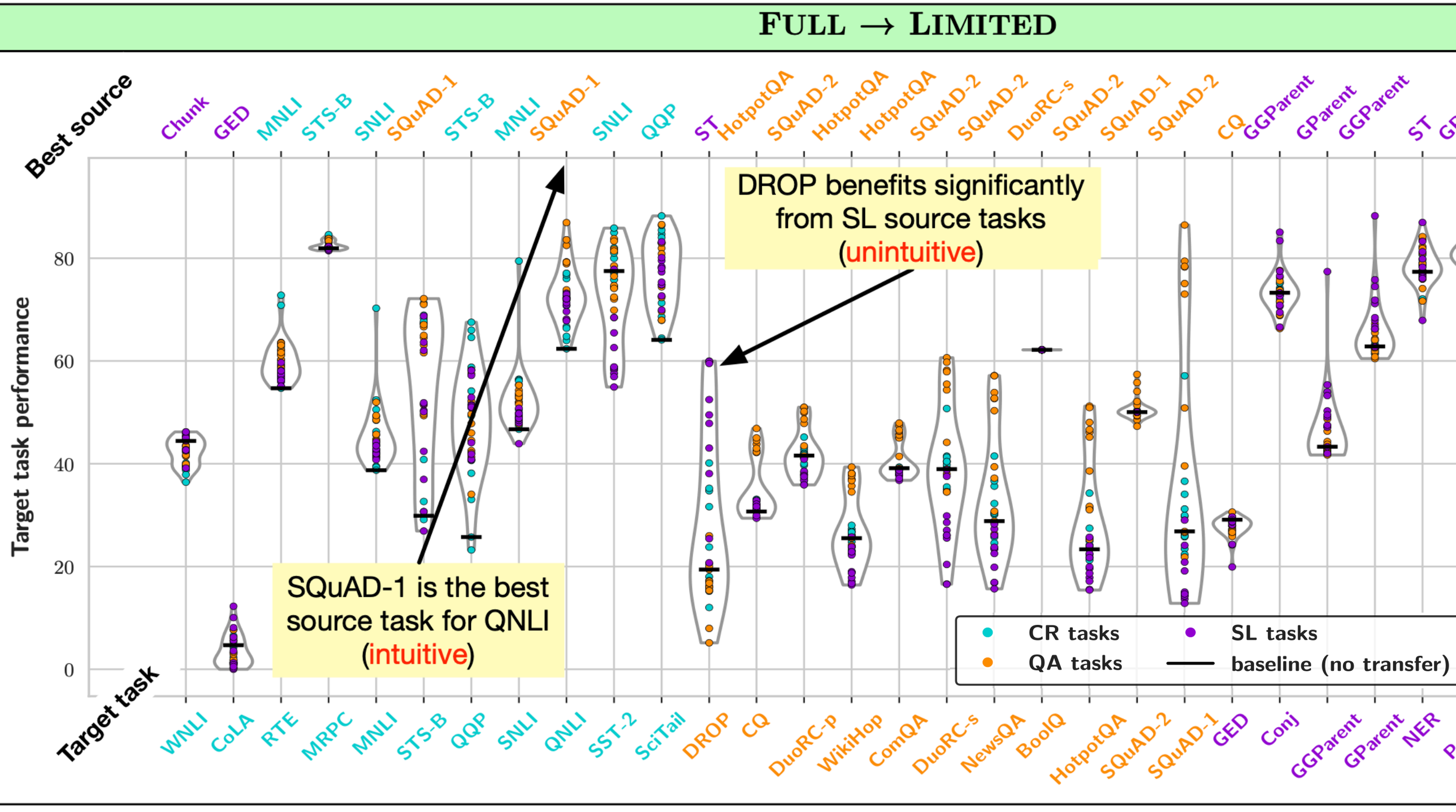
Main findings:

- Contrary to prior belief, **transfer gains are possible even when the source dataset is small**
- **Out-of-class transfer succeeds in many cases**, some of which are unintuitive
- Factors other than source dataset size, such as **the similarity between source and target tasks, matter more in low-data regimes**

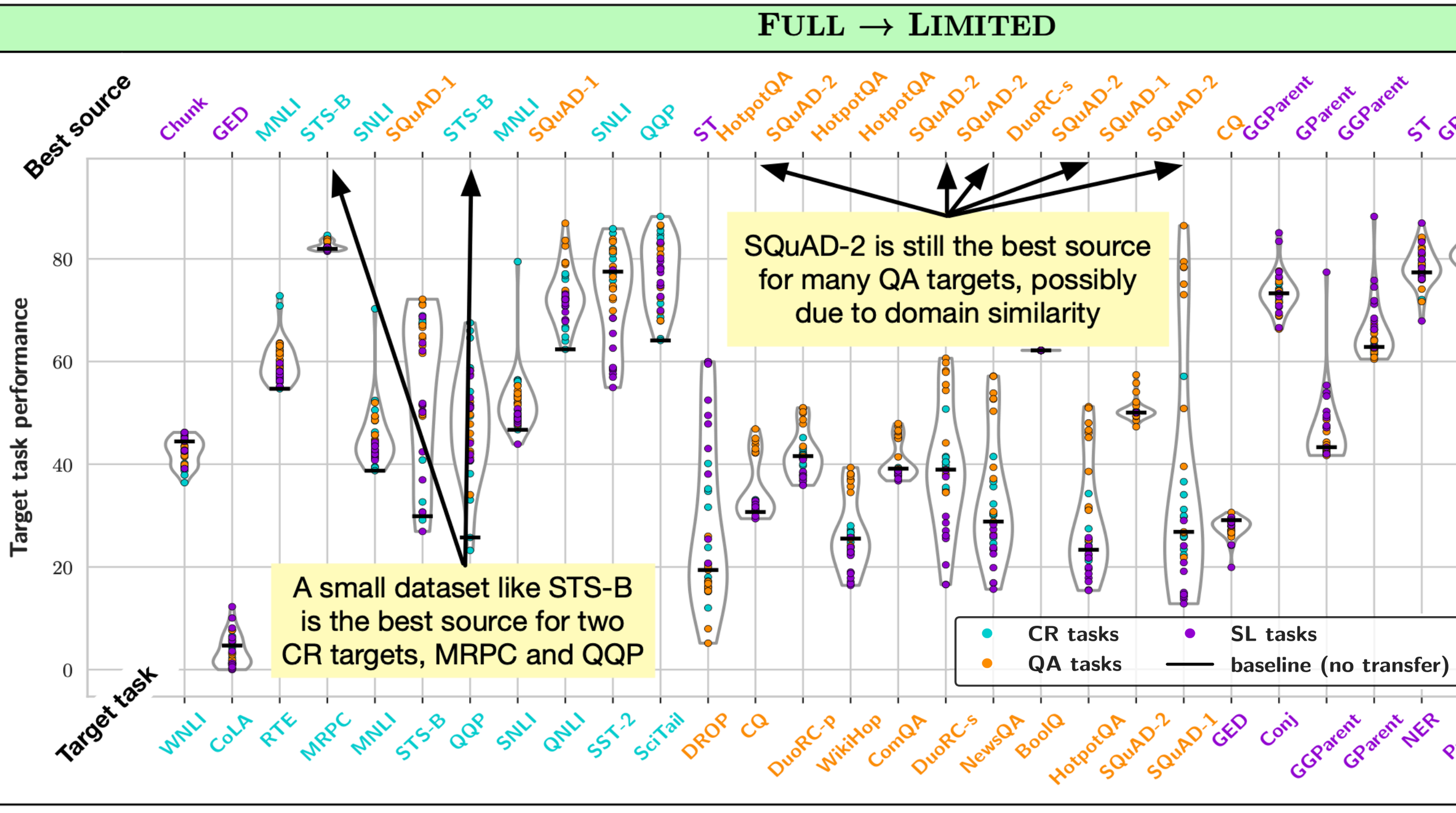
The best source tasks in FULL → FULL *tend to be data-rich tasks*



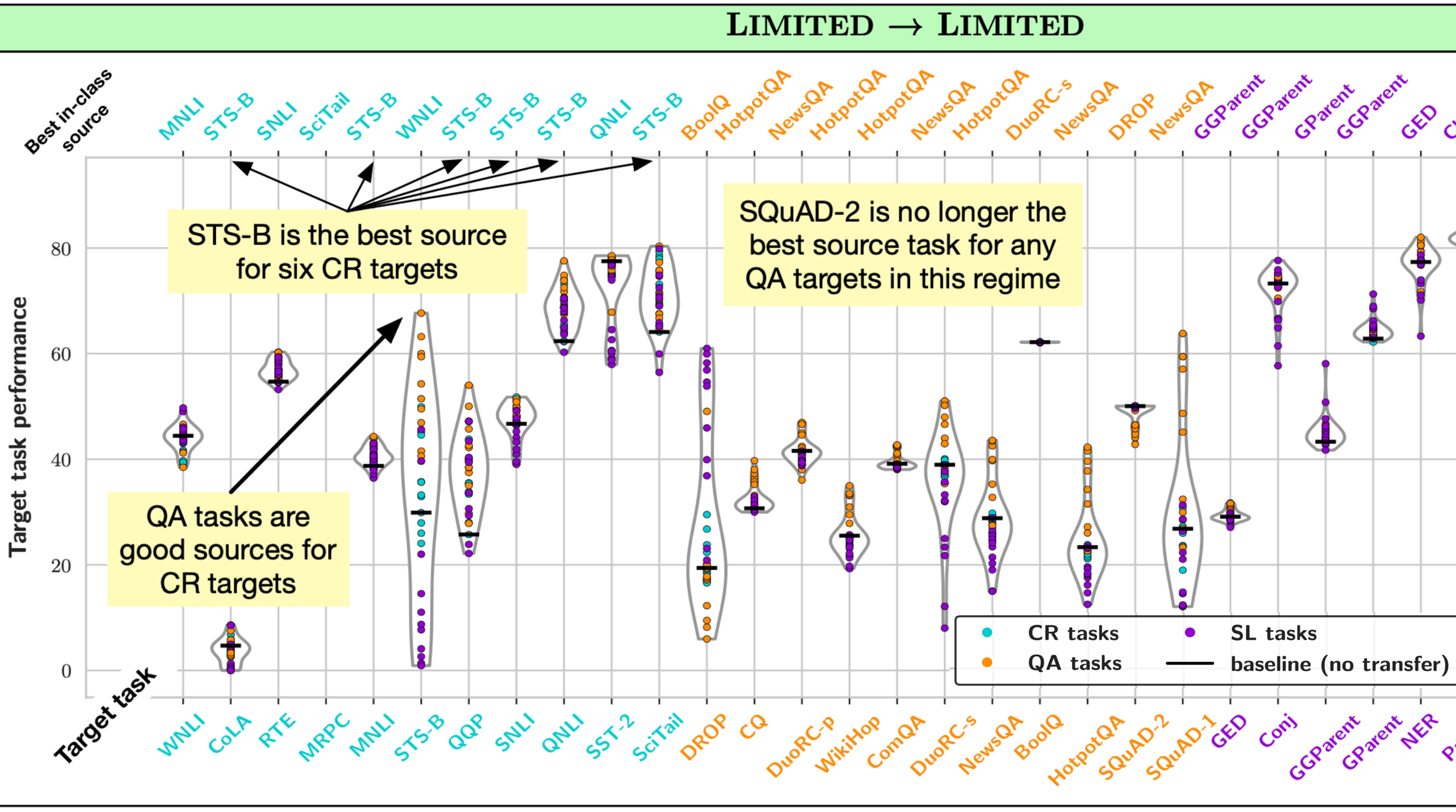
Out-of-class transfer succeeds *in many cases, some of which are unintuitive!*



Large source datasets are not always best for data-constrained target tasks



When does transfer work with data-constrained sources?



Overview

Task and task transferability

Motivation

Exploring task transferability

Predicting task transferability

Task embeddings

What are task embeddings?

- fixed-length dense vector representations of tasks

What might they tell us?

- properties of individual tasks
- similarities and beneficial relationships among tasks

How can we use them to identify similarities and beneficial relationships among tasks?

- the vector space can tell us how closely related two tasks are (i.e., via cosine distance)

How to create task embeddings?

Approach: following the methodology of TASK2VEC (Achille et al., 2019)

- use a **single base network** to represent tasks in a topological space
- pass the given dataset forward through the network
- use the base network's outputs or the gradients of its parameters/ outputs w.r.t. to a task-specific loss

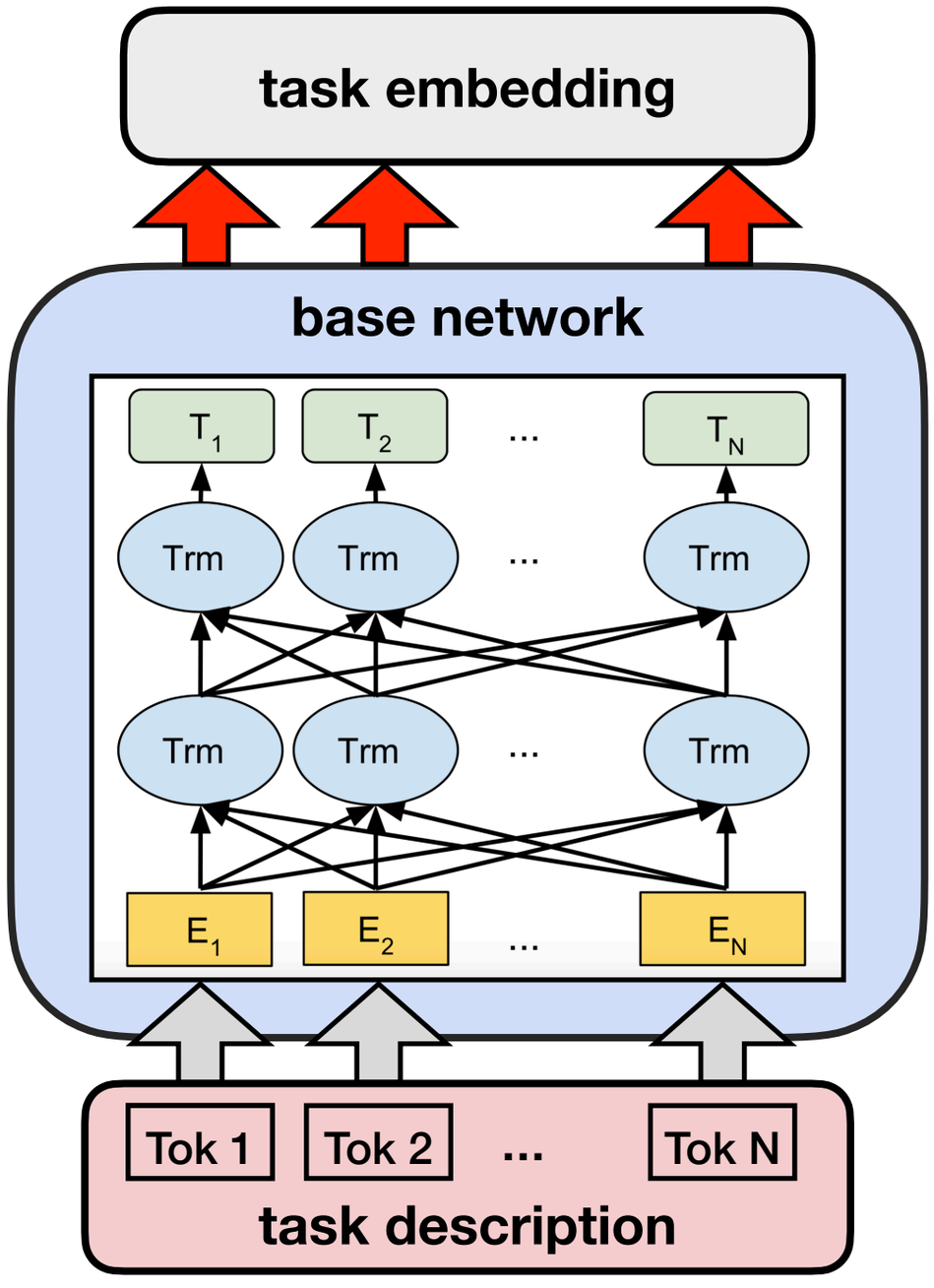
What is the base network?

- a pretrained language model, e.g., BERT, RoBERTa, XLNet, T5

A simple task embedding method

Method:

- use the **task description** only (i.e., a paragraph describing the task)



Limitations:

- requires a clear description for each task

- ignores a lot of interesting information about the task

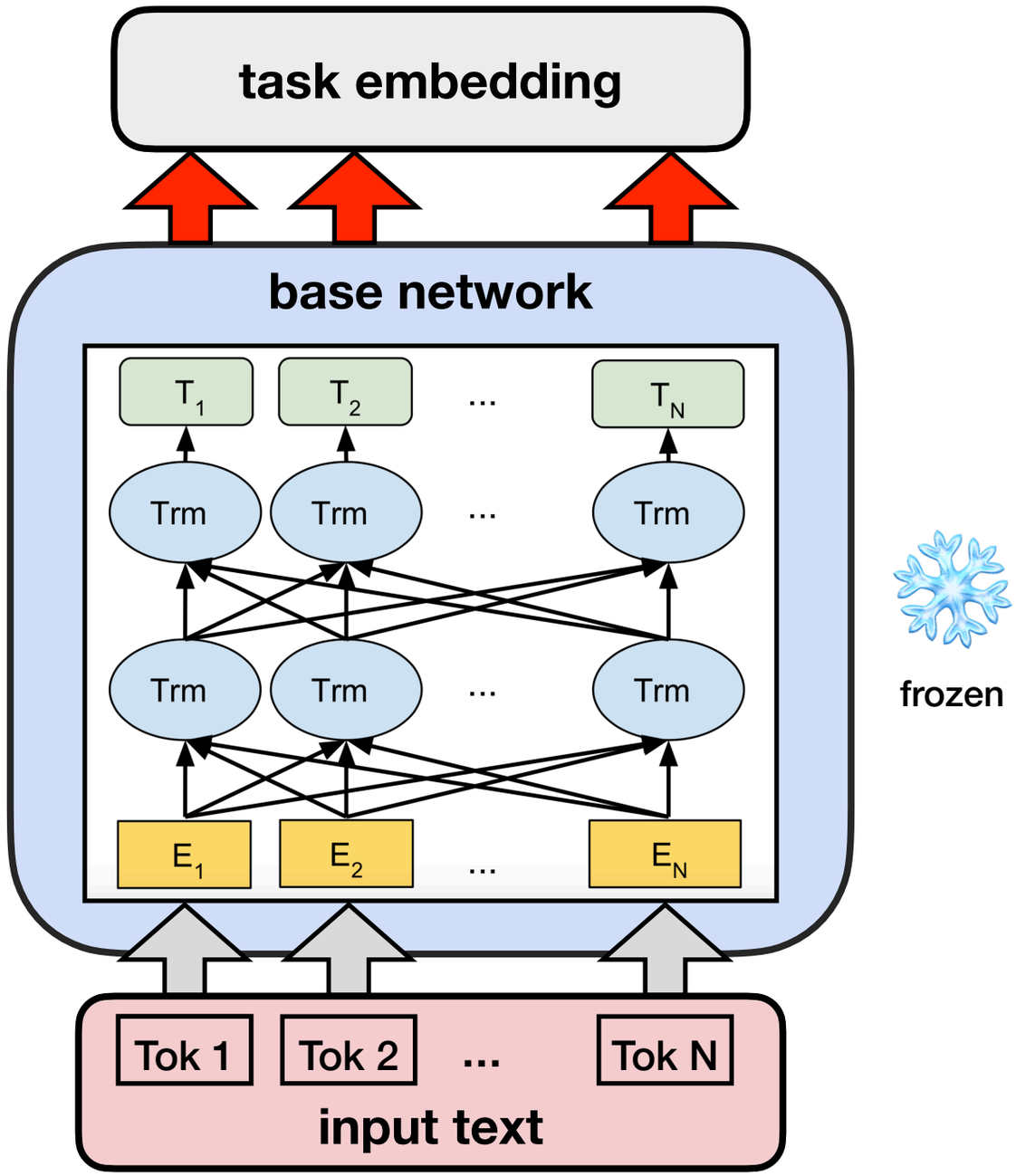
TEXTEMB

Method:

- process each input text through the model without any fine-tuning
- compute **the average of final layer token-level representations**
- average of these pooled vectors over the entire dataset

Motivation:

- capture **properties of the text and domain**



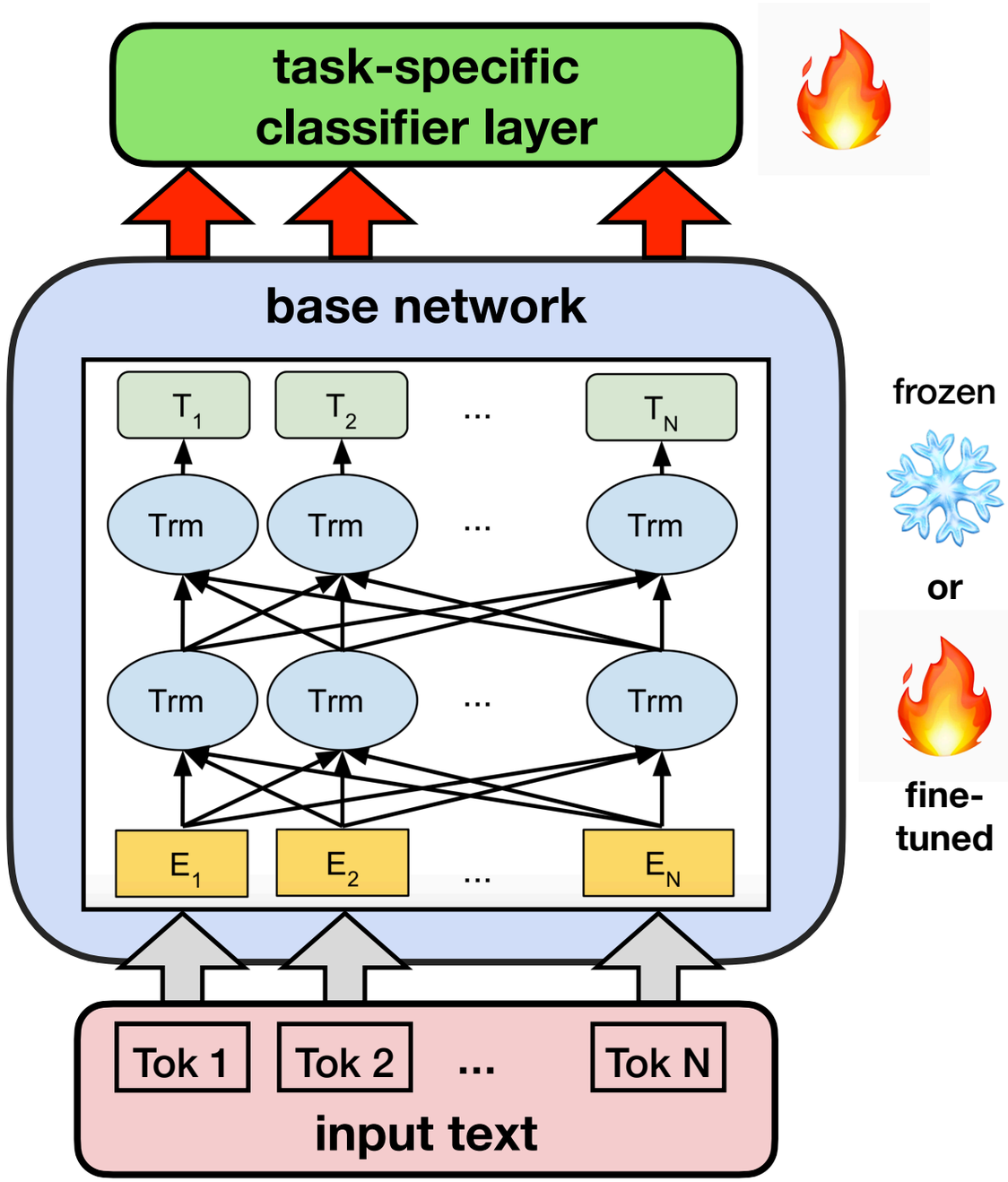
TASKEMB

Method:

- add a task-specific layer for each given task
- fine-tune the model on the training dataset of the task
- compute the task embedding based on **the Fisher information matrix of the base network's parameters and outputs**

Motivation:

- encode information about **the type of knowledge and reasoning required to solve the task**



Fisher information matrix

Formula:

- **theoretical Fisher:**
  the expected covariance of the gradients of the log-likelihood with respect to the model parameters

$$F_\theta = \mathop{\mathbb{E}}_{x,y \sim P_\theta(x,y)} \nabla_\theta \log P_\theta(y|x) \nabla_\theta \log P_\theta(y|x)^T$$

- **empirical Fisher:**
  using the training labels instead of sampling from the model's predictive distribution

$$F_\theta = \frac{1}{n} \sum_{i=1}^{n} \left[\nabla_\theta \log P_\theta(y^i|x^i) \nabla_\theta \log P_\theta(y^i|x^i)^T \right]$$

Fisher information matrix (cont.)

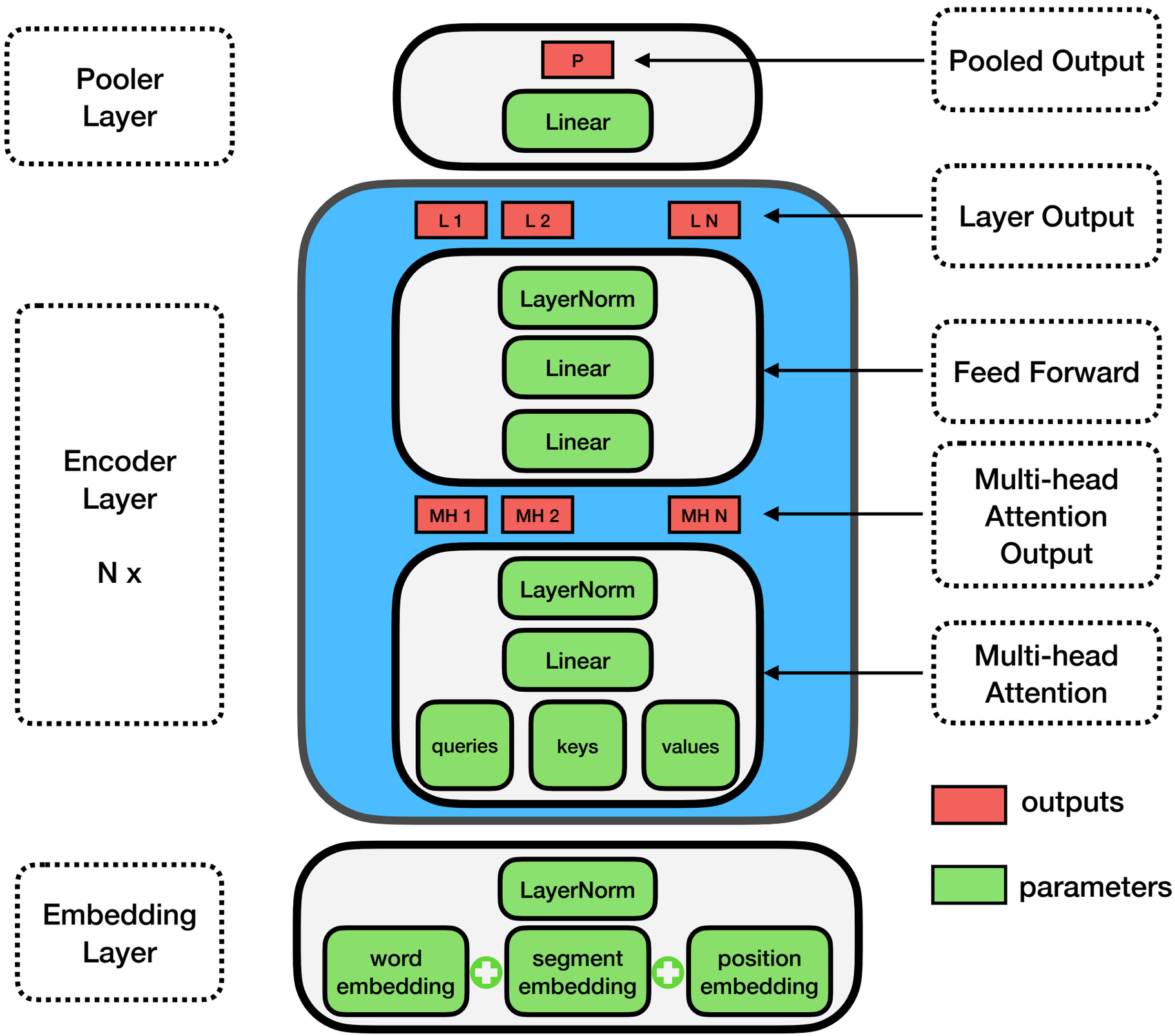
Intuition:

- captures the curvature of the loss surface (the sensitivity of the loss to small perturbations of the model parameters), which intuitively tells us **which of the model parameters are most useful for the task** and thus provides a rich source of knowledge about the task itself

Simplification:

- only consider the diagonal entries of the Fisher or the square of the gradients

Aggregating information from multiple spaces



Evaluation

How do we evaluate the task embeddings?

- Our evaluation centers around the meta-task of selecting the best source task for a given target task

What are the baseline methods?

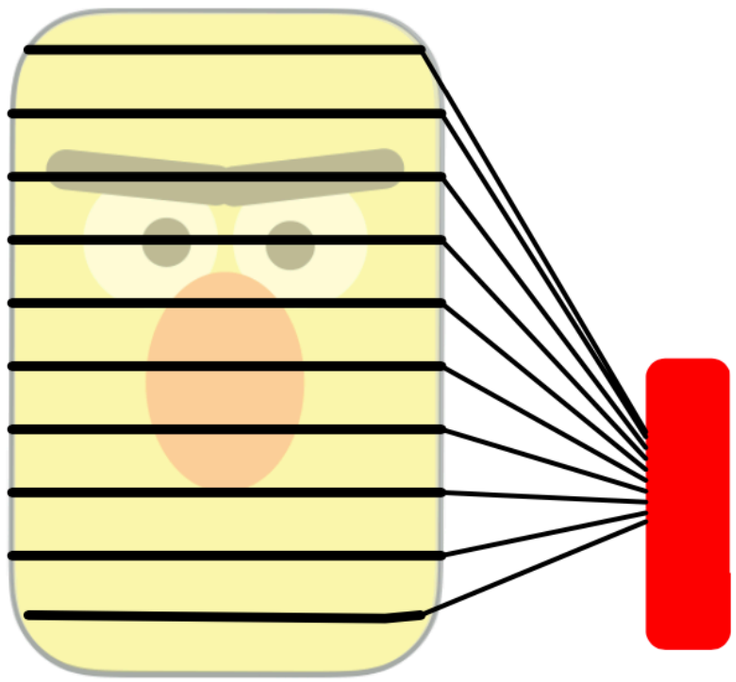
- **DataSize**: ranks all source tasks by the number of training examples
- **CurveGrad**: uses the gradients of the loss curve for each task (Bingel and Søgaard, 2017)

What are the evaluation metrics?

- ***ρ***: the average rank of the source task with the highest absolute transfer gain from our transfer experiments
- **NDCG**: a common information retrieval measure that evaluates the quality of the entire ranking, not just the rank of the best source task

Our task embedding pipeline

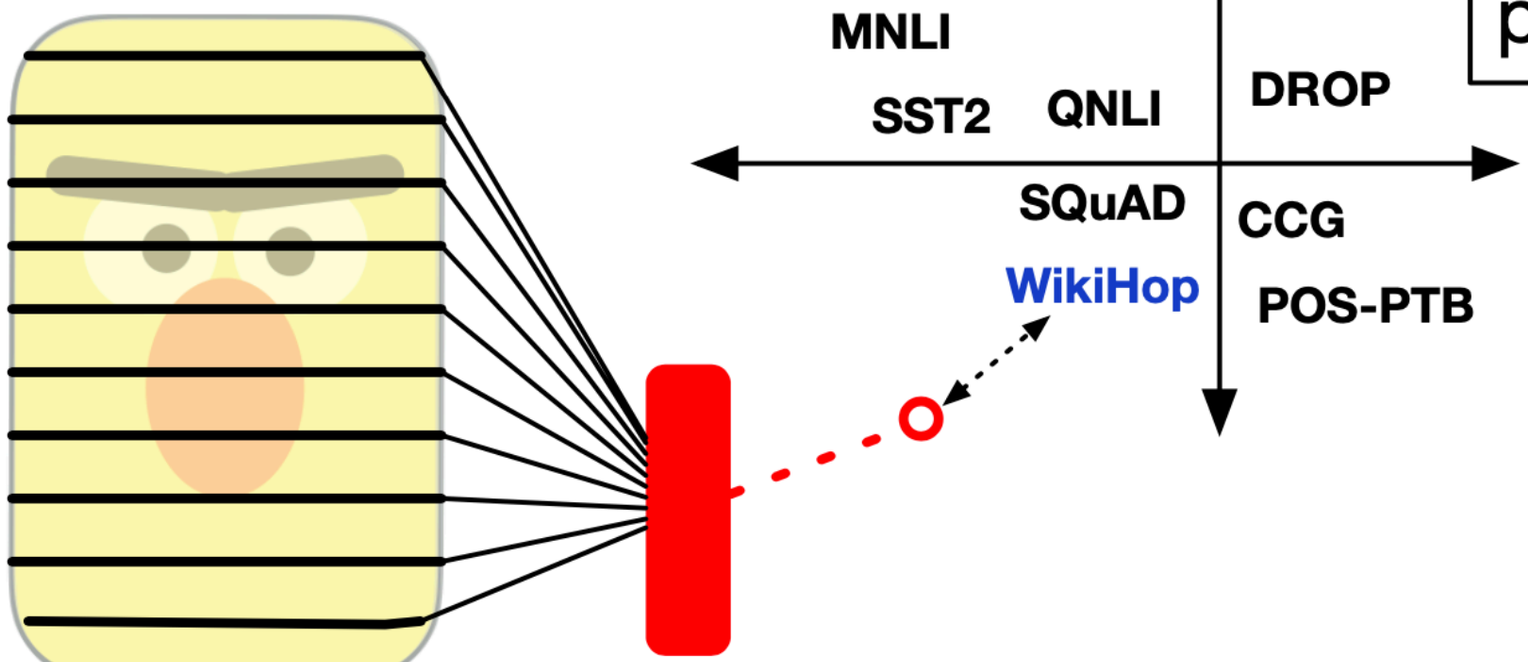
1. given a target task of interest, compute a *task embedding* from BERT's layer-wise gradients



Our task embedding pipeline

1. given a target task of interest, compute a *task embedding* from BERT's layer-wise gradients

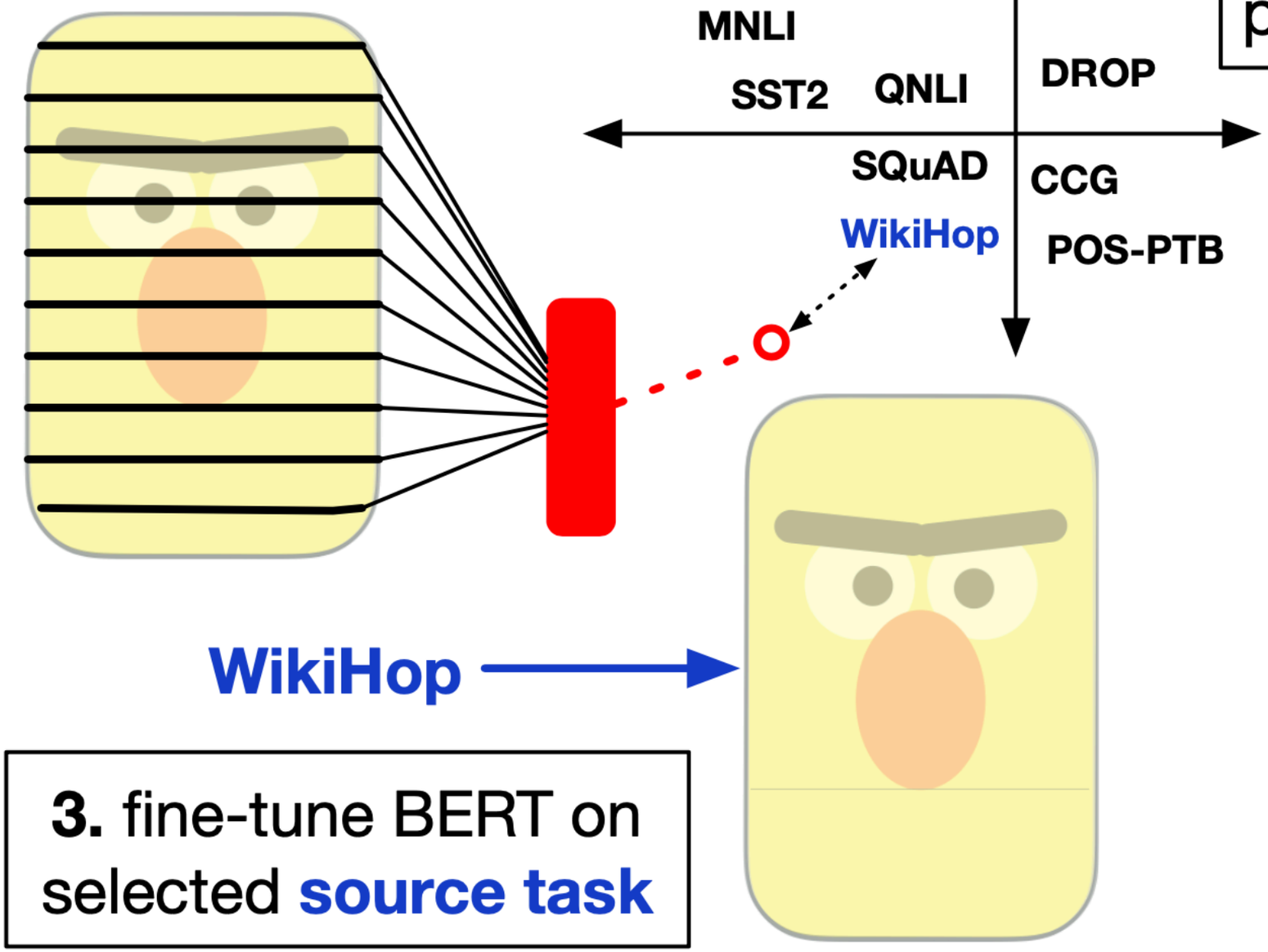
2. identify the most similar **source task** embedding from a precomputed library



Our task embedding pipeline

1. given a target task of interest, compute a *task embedding* from BERT's layer-wise gradients

2. identify the most similar **source task** embedding from a precomputed library

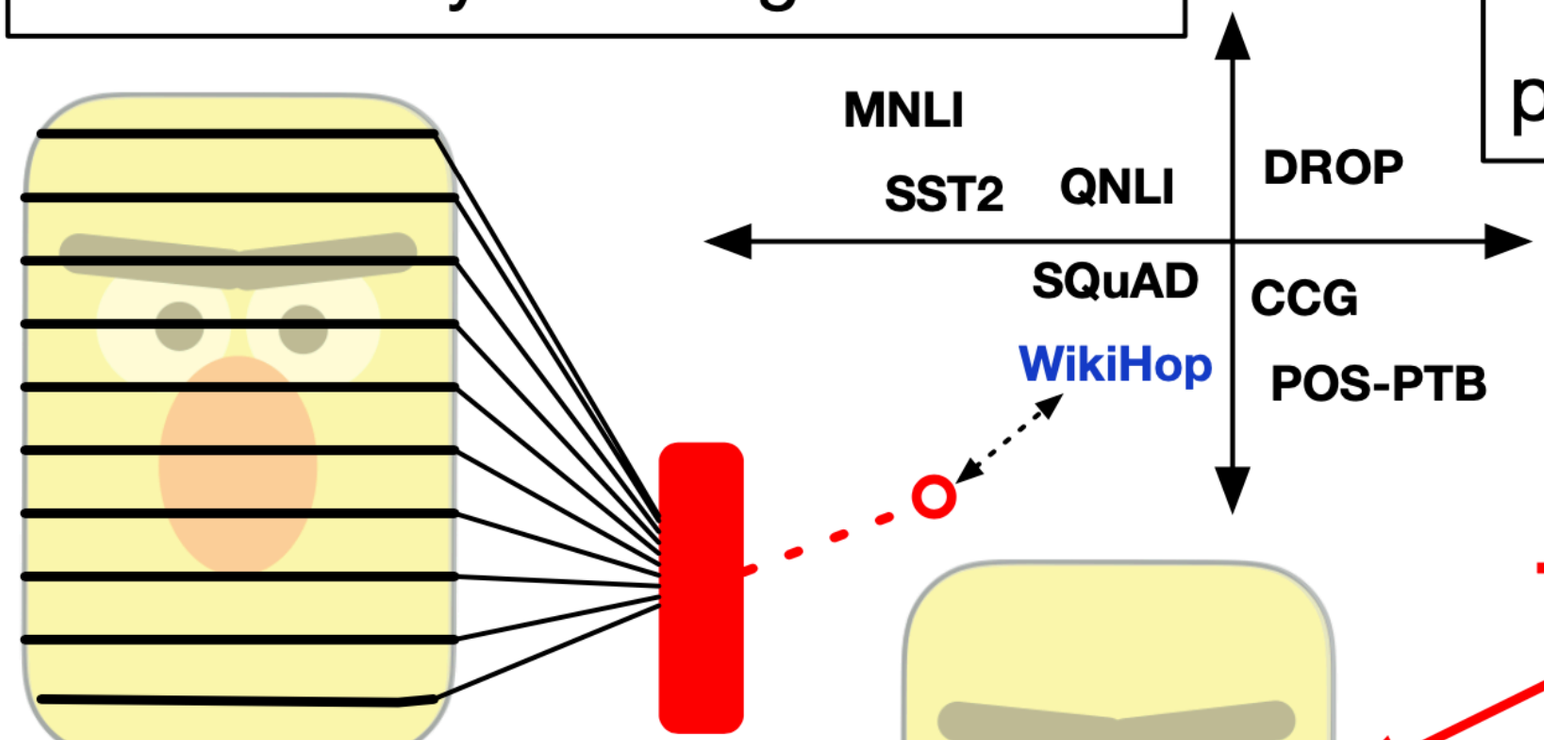


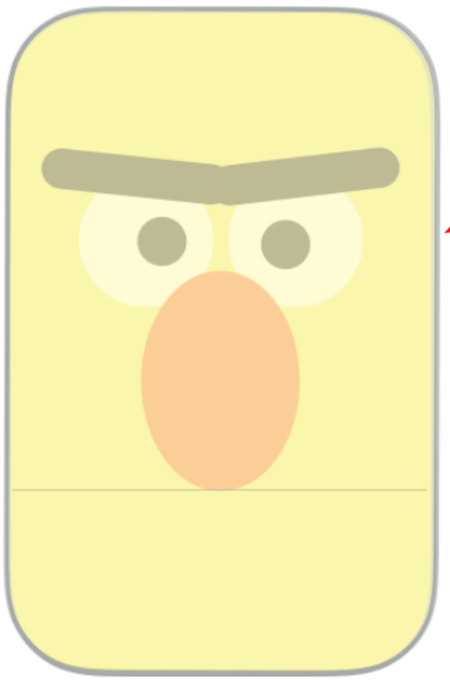
3. fine-tune BERT on selected **source task**

Our task embedding pipeline

1. given a target task of interest, compute a *task embedding* from BERT's layer-wise gradients

2. identify the most similar **source task** embedding from a precomputed library





Target task

WikiHop

3. fine-tune BERT on selected **source task**

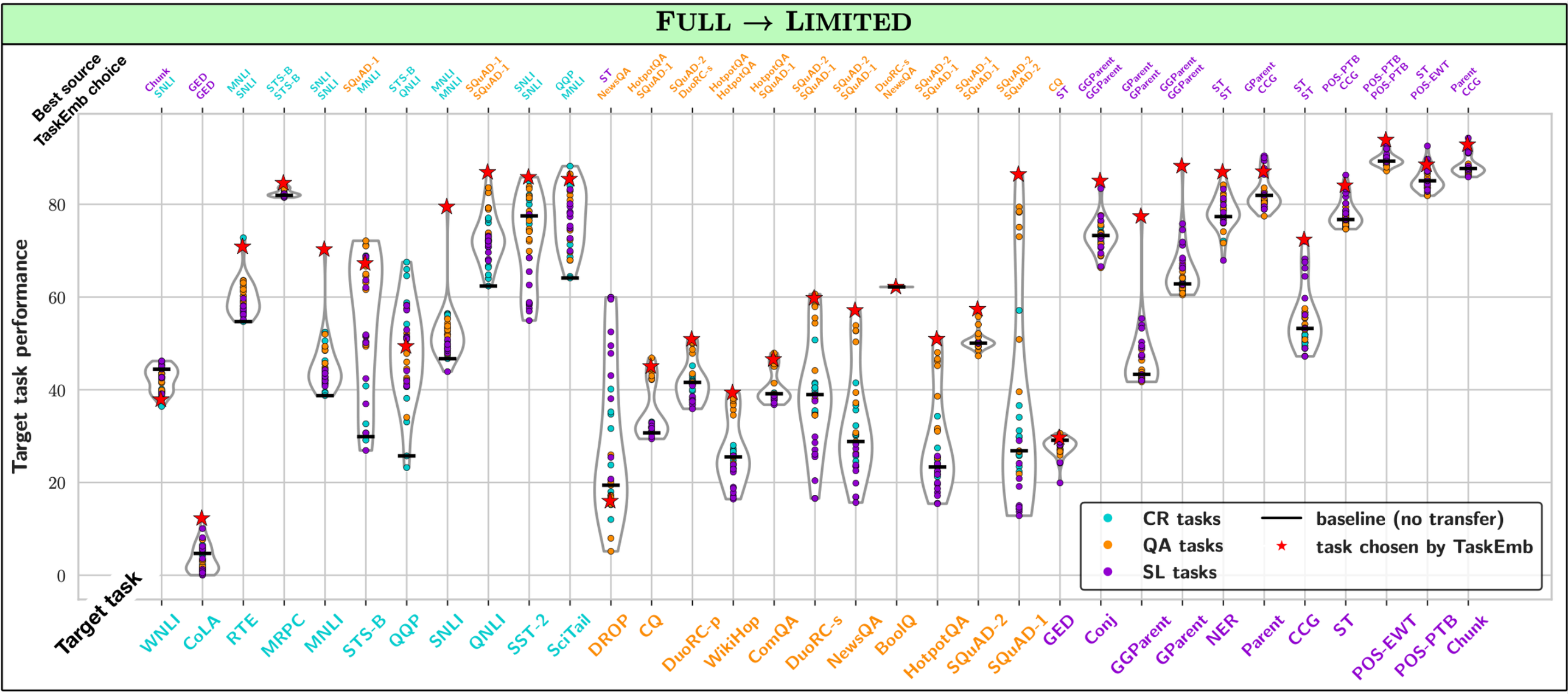
4. fine-tune the resulting model on **target task**

Our approach generally outperforms baseline methods across all settings

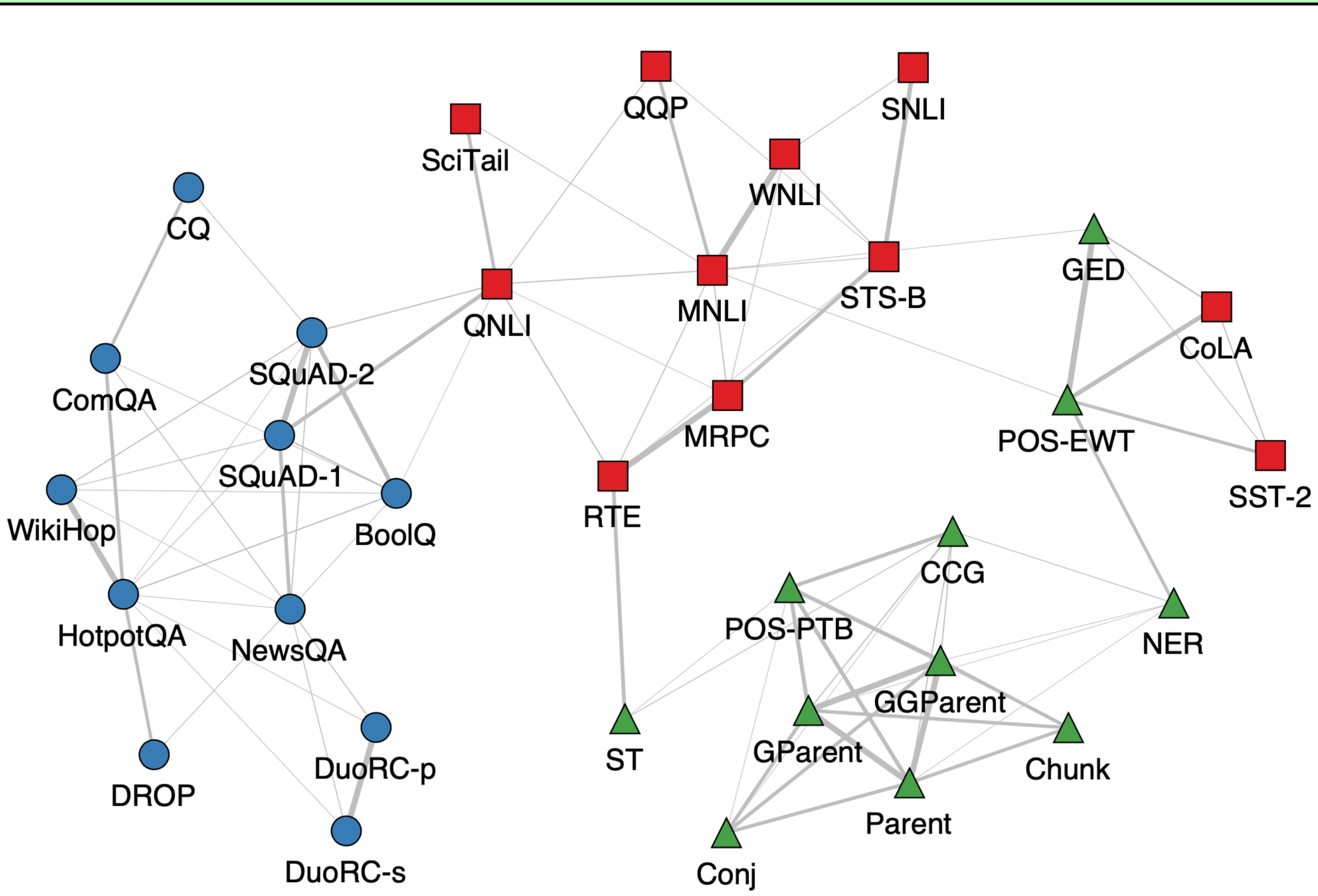
	FULL → FULL				FULL → LIMITED				LIMITED → LIMITED			
	in-class (10)		*all-class (32)*		*in-class (10)*		*all-class (32)*		*in-class (10)*		*all-class (32)*	
Method	ρ	NDCG	ρ	NDCG	ρ	NDCG	ρ	NDCG	ρ	NDCG	ρ	NDCG
classification / regression												
DATASIZE	3.6	80.4	8.5	74.7	3.8	62.9	9.8	54.6	-	-	-	-
CURVEGRAD	5.5	68.6	17.8	64.9	6.4	45.2	18.8	35.0	5.9	50.8	13.3	42.4
TEXTEMB	5.2	76.4	13.1	71.3	3.5	60.3	8.6	52.4	4.8	61.4	13.2	43.9
TASKEMB	2.8	82.3	6.2	76.7	3.4	68.2	**8.2**	60.9	**4.2**	62.6	11.6	**44.8**
TEXT+TASK	**2.6**	**83.3**	**5.6**	**78.0**	**3.3**	**69.5**	**8.2**	**62.0**	**4.2**	**62.7**	**11.4**	**44.8**
question answering												
DATASIZE	**3.2**	84.4	13.8	63.5	2.3	77.0	13.6	40.2	-	-	-	-
CURVEGRAD	8.3	64.8	15.7	55.0	8.2	49.1	16.7	32.8	6.8	53.4	15.3	40.1
TEXTEMB	4.1	81.1	6.8	79.7	2.7	77.6	4.1	77.0	4.1	65.6	7.6	66.5
TASKEMB	**3.2**	84.5	6.5	81.6	2.5	78.0	4.0	79.0	**3.6**	**67.1**	7.5	68.5
TEXT+TASK	**3.2**	**85.9**	**5.4**	**82.5**	**2.2**	**81.2**	**3.6**	**82.0**	**3.6**	66.5	**7.0**	**69.6**
sequence labeling												
DATASIZE	7.9	90.5	19.2	91.6	4.3	63.2	20.3	34.0	-	-	-	-
CURVEGRAD	5.6	92.6	14.6	92.8	8.0	40.7	17.9	30.8	7.0	53.2	18.6	40.8
TEXTEMB	3.7	95.0	10.4	**95.3**	3.9	65.1	8.5	61.1	5.0	67.2	10.1	63.8
TASKEMB	3.4	95.7	**9.6**	95.2	**2.7**	**80.5**	4.4	76.3	**2.5**	82.1	5.5	**76.9**
TEXT+TASK	**3.3**	**96.0**	**9.6**	95.2	**2.7**	80.3	**4.2**	**78.4**	**2.5**	**82.5**	**5.3**	**76.9**

Table 3: To evaluate our embedding methods, we measure the average rank (ρ) that they assign to the best source task (i.e., the one that results in the largest transfer gain) across target tasks, as well as the average NDCG measure of the overall ranking's quality. In parentheses, we show the number of source tasks in each setting. Combining the complementary signals in TASKEMB and TEXTEMB consistently decreases ρ (lower is better) and increases NDCG across all settings, and both methods in isolation generally perform better than the baseline methods.

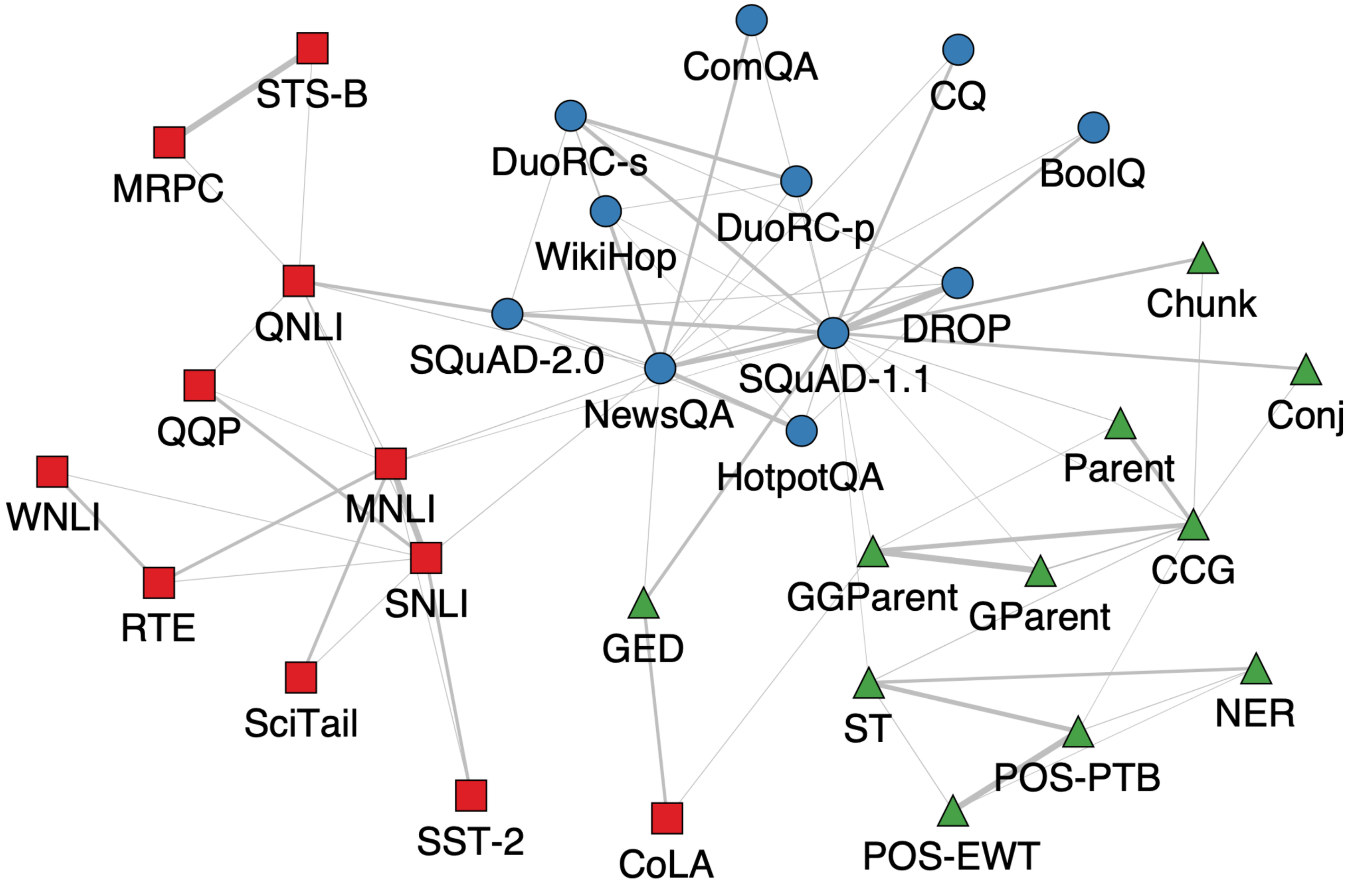
Our approach generally selects source tasks that yield positive transfer, and often selects the best source task



TextEmb Space



TaskEmb Space



Conclusions

What are the main contributions of this paper?

- We perform a large-scale empirical study across 33 different datasets to shed light on the transferability between NLP tasks. We show that the benefits of transfer learning are more pronounced than previously thought, especially when target training data is limited

- We develop methods that learn vector representations of tasks that can be used to reason about the relationships between them

Take-away messages

- positive transfer can occur in a more diverse array of settings than previously thought (e.g., with *low-data source tasks* or *out-of-class transfer*)

- factors such as *data size*, *task and domain similarity*, and *task complexity* all play a role in determining transferability

- we should develop *more principled ways to encode tasks* that can be used to reason about characteristics of individual tasks and the relationships between them

Thank you!

Library & code available at

http://github.com/tuvuumass/task-transferability