Exploring the Effectiveness and Consistency of Task Selection in Intermediate-Task Transfer Learning

Pin-Jie Lin[1] **Miaoran Zhang**[2] **Marius Mosbach**[3,4] **Dietrich Klakow**[2]
[1]Virginia Tech [2]Saarland University, Saarland Informatic Campus
[3]Mila Quebec AI Institute [4]McGill University
pinjie@vt.edu

Abstract

Identifying beneficial tasks to transfer from is a critical step toward successful intermediate-task transfer learning. In this work, we experiment with 130 source-target task combinations and demonstrate that the transfer performance exhibits severe variance across different source tasks and training seeds, highlighting the crucial role of intermediate-task selection in a broader context. We compare four representative task selection methods in a unified setup, focusing on their effectiveness and consistency. Compared to embedding-free methods and text embeddings, task embeddings constructed from fine-tuned weights can better estimate task transferability by improving task prediction scores from 2.59% to 3.96%. Despite their strong performance, we observe that the task embeddings do not consistently demonstrate superiority for tasks requiring reasoning abilities. Furthermore, we introduce a novel method that measures pairwise token similarity using maximum inner product search, leading to the highest performance in task prediction. Our findings suggest that token-wise similarity is better predictive for predicting transferability compared to averaging weights.[1]

1 Introduction

Pre-trained language models (PLMs) have become foundational in the transfer learning paradigm of natural language processing (NLP) (Devlin et al., 2019; Brown et al., 2020; Chowdhery et al., 2023). Intermediate-task transfer learning aims to improve model performance further by introducing an intermediate stage of supervised training on data-rich tasks before fine-tuning the target downstream task (Phang et al., 2018; Pruksachatkun et al., 2020; Vu et al., 2020). The paradigm has shown to be particularly useful for improving performance in resource-constrained scenarios where annotated training data is often limited (Prasad et al., 2021; Vu et al., 2022b).

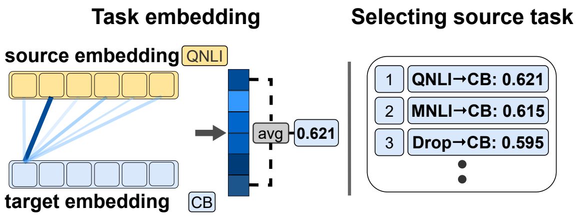


Figure 1: Our proposed method, *maximum inner product search*, is based on pairwise token similarity. Left: Given a target task (e.g., CB), we obtain the maximum token-wise similarity scores between the target and the source tasks for each embedding position. Right: We select the source task with the highest mean of maximum similarity scores.

A crucial aspect of intermediate-task transfer learning is to select beneficial tasks to transfer from. However, the costs of searching for the optimal intermediate-task, especially with the growing array of available NLP tasks and the exhaustive process of model fine-tuning (Pruksachatkun et al., 2020; Vu et al., 2020), are prohibitive. Research on intermediate-task selection mainly predicts task transferability using task-specific embeddings, which condense the task information of a given target task into a single vector representation. For example, some works construct task embedding from fine-tuned weights (Vu et al., 2022b; Zhou et al., 2022) or leverage text embedding (Poth et al., 2021). More specifically, Poth et al. (2021) use sentence transformers to encode dataset examples as text embeddings. The more recent approach by Vu et al. (2022b) constructs task embeddings from the weights of soft prompts, which have been effectively applied in large-scale studies.

Despite their promising results, a systematic study of the consistency of these task selection methods is still missing. Specifically, it remains unclear how consistent these approaches are at predicting the best source task to transfer from.

[1] We release the code publicly at https://github.com/uds-lsv/intermediate-task-selection.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 188–203
August 11-16, 2024 ©2024 Association for Computational Linguistics

To address this gap, we perform a comprehensive evaluation of existing task selection methods in intermediate-task transfer learning. Our research questions are: (1) *Do intermediate-task selection approaches exhibit consistent performance across downstream tasks?* (2) *What are the key ingredients that result in accurate transferability predictions?*

To answer these questions, we perform experiments across 130 intermediate and downstream task combinations derived from 13 source and 10 target tasks. Our results show that intermediate-task transfer exhibits significant performance variance across tasks. Comparing four representative task selection methods, we find that task embeddings based on fine-tuned weights (Vu et al., 2022b) generally outperform embedding-free and text embedding methods (Poth et al., 2021). However, we also observe that such task embeddings do not consistently perform well on tasks requiring high-level reasoning abilities. Exploring this further, we revisit the task embedding design and propose a new construction method based on pairwise token similarity (see Figure 1), which yields the highest average task prediction performance of 82.5%. Our main contributions are as follows:

1. We systematically investigate intermediate-task transfer learning across 130 intermediate and downstream task combinations.

2. We examine four representative task selection methods in a unified setup, including both embedding-free and embedding-based methods.

3. We introduce a novel task embedding construction approach based on pairwise token similarity, which achieves the highest task prediction performance of 82.5% in nDCG score.

4. We provide an in-depth analysis of the impact of task type and training seed, along with an exploration into embedding distributions.

2 Related Work

Identifying a beneficial task from a broader set of source tasks is a crucial step in intermediate-task transfer learning. Various studies have proposed methods to estimate task transferability based on task embeddings.

A foundational approach is Task2Vec (Achille et al., 2019; Vu et al., 2020), which involves computing the Fisher information matrix and enables to measure semantic and taxonomic relationships between tasks. In contrast, Poth et al. (2021) demonstrate the effectiveness of text embeddings based on sentence encoders. The landscape of task selection approaches has further evolved with the introduction of parameter-efficient fine-tuning (PEFT) techniques. For instance, Vu et al. (2022b) use soft prompts to generate task embeddings, demonstrating the effectiveness of prompt-based embeddings. Expanding on this, Zhou et al. (2022) investigate other PEFT methods, including P-tuning (Liu et al., 2022a,b), fine-tuning only bias terms (Ben Zaken et al., 2022), and LoRA (Hu et al., 2022). They construct task embeddings based on the fine-tuned weights.

Task selection based on neuron activations provides another perspective by focusing on the patterns of activations within models. Su et al. (2022) propose model stimulation similarity to identify beneficial source tasks through the overlap rate of activations. More recently, Xi et al. (2023) introduce connectivity patterns as task embeddings, identifying task-specific patterns in deep neural networks that best represent the tasks.

Our work differs from previous studies by contributing a comparison of existing task selection methods in a unified setup, specifically focusing on the effectiveness and consistency of these approaches.

3 Background

In the following, we introduce the intermediate-task transfer learning paradigm and motivate our focus on parameter-efficient fine-tuning.

3.1 Intermediate-Task Transfer Learning

As depicted in Figure 2, intermediate-task training involves sequentially fine-tuning on a source task followed by fine-tuning on a target task. By incorporating an intermediate stage of supervision (typically on data-rich tasks), intermediate-task transfer learning enables knowledge transfer across tasks, thereby enhancing performance on low-resource target tasks (Vu et al., 2022b).

More formally, the intermediate-task transfer learning paradigm can be divided into two stages: (1) training a PLM f_θ on a given source task $\mathcal{T}_s$ to obtain the intermediate model $f_{\theta'}$; (2) training the intermediate model $f_{\theta'}$ on the target task $\mathcal{T}_t$. The objective function with a cross-entropy loss $\mathcal{L}$ of

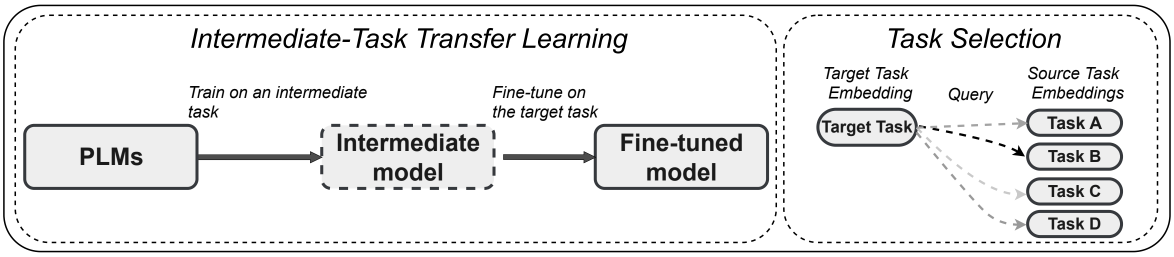


Figure 2: Left: **Intermediate-task transfer learning** performs sequentially learning on the source task followed by fine-tuning on the target task. Right: **Task selection** is a process where given a target task, the goal is to identify the most beneficial task for transfer by searching over a set of source tasks through its task embedding. The selection process relies on a similarity metric to measure the transferability of tasks or datasets.

the first stage is defined as follows:

$$\boldsymbol{\theta}^{'} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{T}_s}(f_{\theta}). \tag{1}$$

Here, the source task $\mathcal{T}_s$ is selected based on a selection criterion using metadata of datasets, domain similarity, or task similarity. Subsequently, the intermediate model is trained on the target task:

$$\boldsymbol{\theta}^{*} = \arg\min_{\boldsymbol{\theta}^{'}} \mathcal{L}_{\mathcal{T}_t}(f_{\theta'}) \tag{2}$$

Note that in Equation 2 the intermediate model f is parameterized with $\boldsymbol{\theta}^{'}$, representing the parameters of the model trained on source task $\mathcal{T}_s$.

3.2 Parameter-Efficient Fine-Tuning via Soft Prompts

Modern language models often contain billions of parameters, making sequential fine-tuning and experimenting with a large number of source and target task combinations impractical. Recent studies have explored parameter-efficient fine-tuning approach through prompt tuning, which involves learning task-specific soft prompts that allow a frozen language model to efficiently perform specific downstream tasks (Lester et al., 2021; Li and Liang, 2021; Liu et al., 2022a). Unlike discrete prompts, soft prompts consist of a set of learnable prompt tokens that are learned through backpropagation and can be applied to various downstream tasks. This approach has been successfully used to efficiently adapt large language models in various scenarios (Qin and Eisner, 2021; Vu et al., 2022a; Asai et al., 2022).

More recently, researchers have focused on intermediate-task transfer learning using prompt tuning, specifically Soft Prompt Transfer (SPoT) (Vu et al., 2022b). SPoT employs a series of soft prompt tokens to adapt frozen models to specific downstream tasks, making it highly parameter-efficient for intermediate-task transfer learning. In this transfer learning procedure, a pre-trained model is adapted to each task by conditioning on a set of learnable prompt tokens. Moreover, the resulting prompts can directly serve as task embeddings to assess task transferability.

Method	**DATASET** D	**MODEL** f	**OUTPUT**
EMBEDDING-FREE			
RANDOM	✗	✗	-
METADATA SIZE	✓	✗	$\mathbb{R}$
EMBEDDING-BASED			
TEXT EMBEDDING SEMB	✓	✓	$\mathbb{R}^d$
TASK EMBEDDING FEATURE	✓	✓	$\mathbb{R}^d$

Table 1: An overview of task selection methods. These task selection methods differ in whether the dataset D and a model f is used for selection and their output format. Note that **SEMB** relies on sentence encoder models, while **FEATURE** requires intermediate models to construct task embeddings.

4 Intermediate-Task Selection Methods

Intermediate-task transfer can improve the performance of the target downstream task, but it is computationally infeasible to try out all possible task combinations, making choosing a beneficial source task an important problem.

Intermediate-task selection aims to predict task transferability and retrieve the most beneficial task from a broad set of available source tasks. This eliminates the need for exhaustive training and is more feasible in resource-constrained scenarios. Here, we compare existing intermediate-task selection methods which can be categorized into two groups: *embedding-free* and *embedding-based* methods (see Table 1).

4.1 Embedding-Free Methods

The first group of methods operates without accessing any model. They estimate task transferability based on certain criteria, such as data size, or simply perform random selection. These methods serve as baseline approaches in Poth et al. (2021).

Random selection (Random) This method selects the intermediate-tasks randomly without using any specific information for the tasks and models.

Data size (Size) This method predicts the task transferability based on the data size, assuming that larger datasets indicate higher transferability to model performance.

4.2 Embedding Methods

The second group of methods constructs embeddings either using a pre-trained sentence encoder model or an intermediate model $f_{\theta'}$. We consider two such methods:

Sentence embeddings (SEmb) It represents the text embedding obtained by averaging all sentence representations on the whole dataset (Poth et al., 2021). Each sentence representation, denoted as h_{x_i}, is encoded by the encoder model for a given example x_i. These sentence representations are averaged over the entire dataset: $\sum_{x_i \sim \mathcal{D}} \frac{h_{x_i}}{|\mathcal{D}|}$. This method captures linguistic properties of the input text x for both the source and target tasks, independent of the intermediate-task training algorithm.

Prompt similarity (Feature) It measures task similarity based on the similarity between their task-specific prompts and employs solely fine-tuned weights to create task embeddings (Vu et al., 2022b). Let the prompt weights be denoted as $[e_1, e_2, ...e_N] \in \mathbb{R}^{N \times d}$, consisting of N soft prompt tokens with d feature dimensions. The prompt similarity score between two tasks, t^1 and t^2, is defined as the cosine similarity of the average representations of prompt tokens:

$$\mathrm{sim}(t^1, t^2) = \cos(\frac{1}{N}\sum_{i=1}^{N} e_i^1, \frac{1}{N}\sum_{j=1}^{N} e_j^2) \quad (3)$$

where e_i^1 and e_j^2 represent the prompt token representations of the tasks t^1 and t^2, and $\cos$ denotes the cosine similarity. This method computes the task embedding, represented as a vector in $\mathbb{R}^d$, by averaging the feature values across all prompt tokens. We refer to this method as Feature to emphasize its focus on capturing task-specific features.

Name	Task	|Train|
source tasks		
MNLI	NLI	393K
QQP	paragraph detection	364K
QNLI	NLI	105K
ReCoRD	QA	101K
CXC	semantic similarity	88K
SQuAD	QA	88K
DROP	QA	77K
SST-2	sentiment analysis	67K
WinoGrande	commonsense reasoning	40K
HellaSWAG	commonsense reasoning	40K
MultiRC	QA	27K
CosmosQA	commonsense reasoning	25K
RACE	QA	25K
target tasks		
BoolQ	QA	9K
CoLA	grammatical acceptability	9K
STS-B	semantic similarity	6K
WiC	word sense disambiguation	5K
CR	sentiment analysis	4K
MRPC	paraphrase detection	4K
RTE	NLI	2K
WSC	coreference resolution	554
COPA	QA	400
CB	NLI	250

Table 2: Overview of source and target tasks. For intermediate-task transfer, we first train on one of the source tasks and then continually fine-tune on the target task.

5 Systematic Evaluation of Task Selection Methods

5.1 Experimental Setup

Datasets. We consider 13 source tasks of various types, including question answering (QA), natural language inference (NLI), and sentiment analysis, among others. We evaluate the transfer performance on 10 target tasks, following the setting in Vu et al. (2022b), as presented in Table 2. More details on the datasets are provided in Appendix A.1.

Models. For all experiments, we adopt T5 Base (Raffel et al., 2020) as our PLM. The pre-trained weights remain frozen, and only the weights of the soft prompt tokens are updated. After training, these fine-tuned weights are then used to construct task embeddings and perform soft prompt transfer.

Implementation details. We closely follow the training configurations outlined in Lester et al. (2021). We train soft prompts for 30K steps, using three random seeds (42, 150, 386). We use $N = 100$ prompt tokens and initialize the weights of the prompt tokens from the embeddings of the top 5K most frequent tokens in the pre-training data. We use the AdaFactor optimizer (Shazeer and Stern, 2018) with a linear scheduler. After conducting prompt tuning, we select the best-performing checkpoint for prompt transfer. The prompt transfer experiment is conducted with another set of training seeds (112, 28, 52).

We evaluate the effectiveness of prompt transfer using a relative transfer performance metric, calculated as follows: $\frac{M_{s\to t}-M_t}{M_t}$. Here, the M_t indicates the model performance with no-transfer prompt tuning, and $M_{s\to t}$ represents the transfer performance. The evaluation metric for the model performance varies according to individual tasks.

5.2 Task Selection Methods and Evaluation

Embedding-based methods. For text embeddings, we follow the model choice in Poth et al. (2021). We use the off-the-shelf encoder models to derive sentence representations for both source and target tasks. Specifically, we adopt Sentence-BERT and Sentence-RoBERTa (Reimers and Gurevych, 2019) as encoders for **SEmb-B** and **SEmb-R**, respectively.

Selection criterion. We rank the order of beneficial tasks based on quantitative values from embedding-free methods. For embedding-based methods on tasks t^1 and t^2, we employ cosine similarity using the mapping function $h(\cdot)$ to construct the task embedding or text embedding for a given intermediate task. To get the ranking order, we sort the source tasks based on the score $\mathrm{sim}(t^1, t^2) = \cos(h(t^1), h(t^2))$ between the source and target tasks. The ground-truth ranking is obtained by transferring source tasks to the downstream task and sorting them based on transfer performance.

Evaluation. We use two metrics[2] to evaluate the effectiveness of task selection methods: (1) Normalized Discounted Cumulative Gain (nDCG) (Järvelin and Kekäläinen, 2002), a widely accepted information retrieval measure that evaluates the overall quality of a ranking, emphasizing the entire order rather than merely focusing on the rank of the best source task. The nDCG score ranges from 0 to 1, where 1 presents the exact match with the ideal order and lower values indicate a lower quality of ranking. (2) Regret@k (Renggli et al., 2022), a metric for computational regret, quantifying the relative performance between the expected performance of the top-k selected intermediate-tasks and the optimal intermediate-task. Lower regret signifies a more effective selection strategy among the k intermediate models. For each target task, we evaluate the overall ranking prediction of the 13 source tasks against the ground-truth ranking using nDCG score. We evaluate the efficacy of the top-k selected source tasks compared to the ground-truth selection using Regret@k.

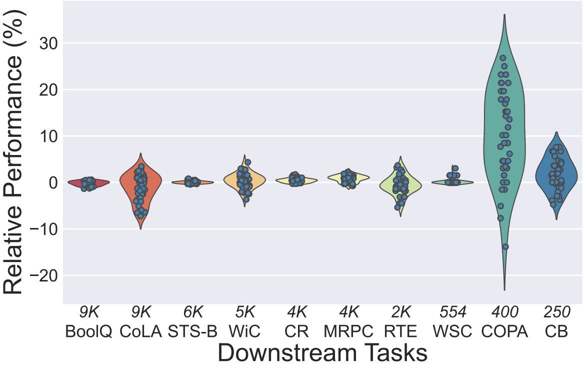


Figure 3: Relative transfer performance across ten downstream tasks with 390 intermediate-task trained models (13 source × 10 target tasks × 3 seeds). Each violin plot illustrates the distribution of performance on the x-axis, with each dot denoting the relative improvement or deterioration compared to the no-transfer baseline on the y-axis. Tasks are arranged in descending order of the training sample sizes.

5.3 Results

Intermediate-task transfer exhibits high-performance variance across tasks. Figure 3 illustrates the relative transfer performance across 10 target tasks, sorted by their training data sizes [3]. We find that relative transfer performance through intermediate-task training exhibits significant variance across tasks, especially for the downstream tasks CoLA, RTE, COPA, and CB. This observation aligns with previous studies showing significant performance variation across source tasks (Pruksachatkun et al., 2020; Jiang et al.,

[2] See formal definitions in Appendix A.2.

[3] The detailed transfer performances are presented in Appendix C.

	Classification		M. Choice		QA		All	
	R@1↓	nDCG↑	R@1↓	nDCG↑	R@1↓	nDCG↑	R@1↓	nDCG↑
Random	2.18	81.53	2.20	84.52	1.45	86.43	2.89	77.89
Size	2.10	83.73	**1.44**	86.01	**0.88**	90.06	2.78	78.00
SEmb-B	1.92	85.21	1.91	86.12	1.21	90.11	2.75	78.23
SEmb-R	1.82	86.51	1.74	86.31	1.12	90.23	2.32	79.26
Feature	**1.28**	**87.31**	1.67	**86.40**	1.02	**90.70**	**2.04**	**81.85**

Table 3: Comparison of task selection methods on 10 downstream tasks. The nDCG and Regret@1 (R@1) scores are grouped by the target task category and we report the mean scores for each group. The best scores in each group are boldfaced.

2023). Additionally, we find that this phenomenon is particularly pronounced in downstream tasks with extremely limited labeled data, such as COPA and CB. In contrast, the relative transfer performance is more consistent for downstream tasks that have sufficient training data, like BoolQ and STS-B. In Appendix B, we show that there exists a correlation between transfer gains and training data sizes. These results highlight the importance of carefully selecting beneficial tasks to enhance transfer gains, especially in low-resource scenarios.

Embedding-based selection methods outperform embedding-free methods, but the transfer gains are limited. Table 3 presents results for the four task selection methods. Embedding-based approaches show higher task prediction performance over embedding-free methods, indicating richer information is obtained from encoded representations for predicting task transferability. Specifically, Feature outperforms all other task selection methods on average. Despite its strong performance, Feature falls short of the simple Size approach in Regret@1 for multiple choice (M. Choice) and question answering (QA) tasks. This highlights the need to further improve task embeddings, especially for tasks that require reasoning abilities.

In Table 4, we show the effectiveness of task selection methods on prompt transfer performance. Random and Size select the source task with the highest task transferability score. SEmb-R and Feature select top-k tasks that exhibit the largest value of the transferability scores. Compared to the no-transfer baseline, these task selection methods show average absolute performance improvements ranging from 0.38% to 0.91%. With an increase of the selection pool (k=1 to k=3), the improvements by Semb-R and Feature further increase to 0.78% and 1.03%, respectively. However, the overall transfer gains remain marginal, indicating that the effectiveness of intermediate-task selection is still limited across diverse tasks.

	Transfer Gain		Avg. score
	Abs.	Rel.	
No Transfer	-	-	77.2
Random	0.38	0.49	77.58
Size	0.52	0.67	77.72
SEmb-R			
Best of Top-k			
k=1	0.72	0.93	77.92
k=3	0.78	1.01	77.98
Feature			
Best of Top-k			
k=1	0.91	1.17	78.11
k=3	**1.03**	**1.33**	**78.23**

Table 4: Comparison of task selection methods on model performance. Abs and Rel represent absolute and relative improvements compared to no-transfer baseline. Avg. score is calculated across 10 downstream tasks with three runs. Best of Top-k is the best performance across the top-k selected source tasks.

5.4 Effect of Task Type and Training Seed

To dissect the impact of task type and training seed, Table 5 presents the top-3 beneficial intermediate-tasks for COPA and CB. Results for all other tasks are shown in Appendix D.

Task type is not a reliable transferability predictor. While it is intuitive to assume that similar tasks should transfer well to the downstream task, our results reveal that the top-performing source tasks for a given target task can vary widely in task type. We find that task types are generally uncorrelated with transfer performances. For example,

TARGET		*seed 112*			28			52	
	SOURCE	TASK TYPE	REL. (%)	SOURCE	TASK TYPE	REL. (%)	SOURCE	TASK TYPE	REL. (%)
Top-3 transfer									
COPA (QA)	MultiRC*	QA	7.69	CxC	semantic sim.	16.94	QQP	paraphrase	26.78
	DROP*	QA	6.15	MultiRC*/RACE*	QA/QA	15.25	ReCORD*	QA	24.99
	RACE*	QA	4.61	QQP	paraphrase	13.55	WinoGr./MultiRC*	reasoning/QA	23.21
Top-3 transfer									
CB (NLI)	QNLI*	NLI	4.11	RACE	QA	4.04	CxC/RACE	semantic sim./QA	7.60
	MNLI/WinoGr.	NLI/reasoning	3.61	ReCORD	QA	3.53	ReCORD	QA	7.57
	SQuAD	QA	2.70	SQuAD	QA	2.73	QNLI/HellaSWAG	NLI/reasoning	7.72

Table 5: Top-3 intermediate-task transfer on COPA and CB. **REL.** is the relative performance improvement (%) calculated based on the corresponding no-transfer prompt tuning. * indicates that the source task type is identical to the downstream task type.

the most performant source tasks for COPA and CB often come from different task types when various training seeds are used. Based on three separate runs, the most beneficial source tasks for COPA (QA) are from other task types, such as CXC (semantic similarity) and QQP (paraphrase detection). Similarly, many of the beneficial tasks for CB (NLI) originated from non-NLI tasks.

Random seed significantly impacts the transfer performance. For COPA, using different training seeds leads to 7.69% to 26.78% relative performance improvements. Similarly, the relative improvements for CB range from 4.11% to 7.60%. This emphasizes the crucial role of seed choice in determining transfer performance. We observe similar variations across seeds in other downstream tasks as well, such as CoLA, WIC, and RTE. This can be attributed to the instability in fine-tuning introduced by different random seeds during prompt transfer (Mosbach et al., 2021; Chen et al., 2022), which can largely affect the robustness of intermediate-task selection.

6 Revisiting the Construction of Task Embeddings

Despite task embeddings from fine-tuned weights demonstrating superior performance in task prediction compared to other selection methods, the effectiveness of various task embedding constructions remains underexplored. In this section, we investigate different construction methods of task embeddings. In addition to **FEATURE**, we explore two more types of task embeddings as follows.

6.1 Construction Methods

Token-wise mean (UNIGRAM) In **FEATURE**, we compute the mean of token representations to obtain a task embedding in $\mathbb{R}^d$. To explore an alternative approach, we compute the task embeddings from another axis, resulting in a task embedding in $\mathbb{R}^N$. Specifically, the task embedding for a task t denotes as $h_t = \frac{1}{d}[\sum_d e_1, \sum_d e_2, ..., \sum_d e_N]$. The similarity between tasks t^1 and t^2 is defined as: $\mathrm{sim}(t^1, t^2) = \cos(h_{t^1}, h_{t^2})$. We refer to this method as **UNIGRAM** to emphasize that task-specific information is aggregated from the token-wise dimension.

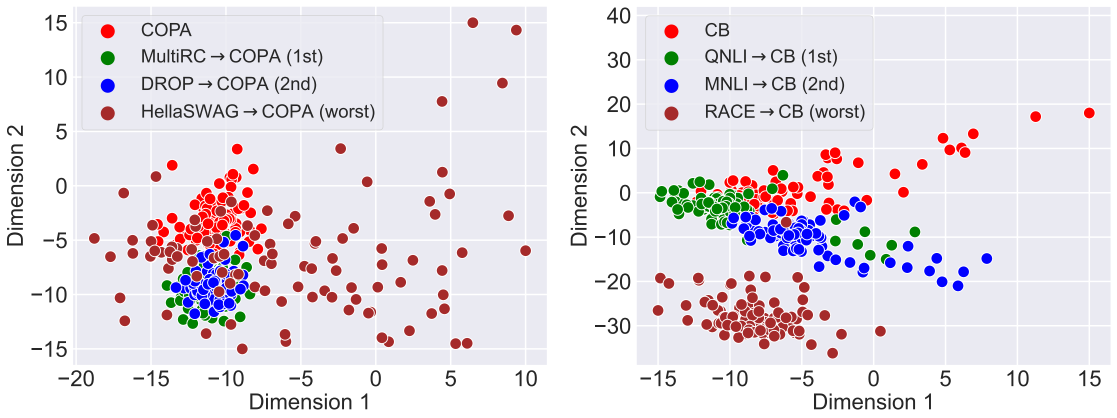
Maximum inner product search (MAX) We propose a novel task embedding method, referred to as **MAX**, based on the maximum token-to-token similarity scores. Given the source task t^1 and the target task t^2, for each prompt token in t^2, we obtain the highest token representation similarity score across all tokens in t^1. The task similarity is then defined as the mean of these maximum similarity scores:

$$\mathrm{sim}(t^1, t^2) = \frac{1}{N} \sum_{j=1}^{N} \max_i \cos(e_i^1, e_j^2) \qquad (4)$$

6.2 Results and Analysis

MAX achieves the highest task transferability prediction. Figure 5 presents three types of task embeddings, each derived from prompt checkpoints trained for different numbers of steps. All three methods show improved performance with longer training steps, suggesting that longer training improves task transferability predictions. Notably, **MAX** achieves the highest nDCG score of 82.5% at the 20K step, indicating that token-wise similarity captures richer task information than **FEATURE** and **UNIGRAM**, leading to more accurate task predictions.

Prompt tokens from beneficial tasks are distributed closer to the target prompt tokens. To better understand the prompt token distribution and different levels of transfer performance, we project prompt tokens of the best, 2nd-best,



(a) COPA

(b) CB

Figure 4: Projecting prompt tokens of the best, 2nd-best, and worst-performing intermediate-tasks for (a) COPA and (b) CB using t-SNE. We observe that prompt tokens from beneficial tasks are distributed more closely to the tokens of no-transfer prompt tuning.

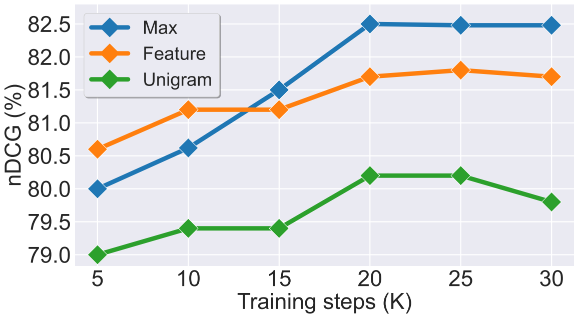


Figure 5: Task prediction performances (average nDCG scores) of three types of task embeddings.

and worst-performing intermediate-tasks onto low-dimensional spaces using t-SNE (van der Maaten and Hinton, 2008). Figure 4 illustrates that the prompt tokens from no-transfer prompt tuning (red), are close to the tokens from their beneficial intermediate-tasks (green, blue). Furthermore, we observe a considerable overlap in these beneficial source tasks, such as MULTIRC and DROP, for downstream task COPA. This suggests that beneficial tasks tend to be distributed closer to the target prompt tokens and share similar characteristics in low dimensions. For COPA and CB, the worst-performing intermediate-task (brown) deviates from the no-transfer prompt tokens. Future research can further explore a clearer correlation between intermediate-task token distribution and transfer performance.

7 Conclusion

In this work, we conduct a systematic study on intermediate-task selection across a wide range of tasks. Our results show that task embeddings based on fine-tuned weights outperform random selection, data size, and text embeddings with improvements of +3.96%, +3.85%, and +2.59% in nDCG scores, underscoring the importance of a task-specific approach. Nevertheless, we find that task embeddings do not excel in all scenarios, particularly in multiple choice and QA tasks. By revisiting the task embedding construction, we propose a novel method based on pairwise token similarity, which achieves the highest performance of 82.5% in task transferability prediction, suggesting that token-wise similarity is better predictive in task transferability prediction.

Limitation

Despite our proposed method being effective in many scenarios, we observe that it falls short in predicting task transferability for tasks requiring reasoning abilities, which needs to be further explored. We also face a challenge in precisely evaluating how the parameter configurations of soft prompt tuning impact transfer performance, as prompt tuning is highly sensitive to hyperparameter selection. Moreover, our evaluation of task selection is limited to one specific model architecture and focused on soft prompt tuning. Evaluating on different model architectures, model scales, and fine-tuning methods would provide a more comprehensive understanding of the robustness of intermediate-task selection.

Acknowledgements

We would like to thank anonymous reviewers for their constructive feedback. This work was funded by the Deutsche Forschungsgemeinschaft

(DFG, German Research Foundation) – project-id 232722074 – SFB 1102.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. *CoRR*, abs/1902.03545.

Akari Asai, Mohammadreza Salehi, Matthew Peters, and Hannaneh Hajishirzi. 2022. ATTEMPT: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6655–6672, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

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.

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

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*.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Junguang Jiang, Baixu Chen, Junwei Pan, Ximei Wang, Dapeng Liu, jie jiang, and Mingsheng Long. 2023. Forkmerge: Mitigating negative transfer in auxiliary-task learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Joongwon Kim, Akari Asai, Gabriel Ilharco, and Hannaneh Hajishirzi. 2023. TaskWeb: Selecting better source tasks for multi-task NLP. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11032–11052, Singapore. Association for Computational Linguistics.

Changho Lee, Janghoon Han, Seonghyeon Ye, Stanley Jungkyu Choi, Honglak Lee, and Kyunghoon Bae. 2024. Instruction matters, a simple yet effective task selection approach in instruction tuning for specific tasks. *Preprint*, arXiv:2404.16418.

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.

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

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022a. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *Preprint*, arXiv:2110.07602.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *CoRR*, abs/1811.01088.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? Efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10585–10605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Archiki Prasad, Mohammad Ali Rehan, Shreya Pathak, and Preethi Jyothi. 2021. The effectiveness of intermediate-task training for code-switched natural language understanding. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 176–190, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. 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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Cedric Renggli, André Susano Pinto, Luka Rimanic, Joan Puigcerver, Carlos Riquelme, Ce Zhang, and Mario Lučić. 2022. Which model to transfer? finding the needle in the growing haystack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9205–9214.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan

Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. 2022. On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3949–3969, Seattle, United States. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022a. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9279–9300, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022b. SPoT: Better frozen model adaptation through soft prompt transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Zhiheng Xi, Rui Zheng, Yuansen Zhang, Xuanjing Huang, Zhongyu Wei, Minlong Peng, Mingming Sun, Qi Zhang, and Tao Gui. 2023. Connectivity patterns are task embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11993–12013, Toronto, Canada. Association for Computational Linguistics.

Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wangchunshu Zhou, Canwen Xu, and Julian McAuley. 2022. Efficiently tuned parameters are task embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5007–5014, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A More Details to Datasets and Evaluation Metrics

A.1 Datasets

We select the datasets drawn from different NLP benchmarks and families of tasks, including natural language inference (NLI), paraphrase detection, semantic similarity, sentiment analysis, question answering (QA), commonsense reasoning, and grammatical acceptability. In total, we consider 13 source and 10 target tasks. The distinguishing between high-resource and low-resource tasks follows conventional notions respect with to the training split size. Table 6 summarizes the statistics of 23 tasks and the evaluation metrics. All data was sourced from HuggingFace Datasets (Lhoest et al., 2021).

A.2 Evaluation Metircs

nDCG This metric is built on the concept of Discounted Cumulative Gain (DCG), a measure of the relevance score for a list of items, each discounted by its position in the ranking.

$$DCG(R) = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)} \qquad (5)$$

where R represents the ranking of source tasks, where the relevance rel_i of the source task with rank i is set to the averaged target performance, i.e., $rel_i \in [0, 100]$. The ranking position ρ corresponds to the size of the selection budget.

The nDCG is computed as follows:

$$nDCG(R_{pred}, R_{true}) = \frac{DCG(R_{pred})}{DCG(R_{true})} \qquad (6)$$

While nDCG generally considers the overall ranking and the difference between predicted transfer performance and actual performance, realistic applications often prioritize the top-1 transfer performance. In this study, our focus is on metrics that accurately quantify the accuracy of top-1 predictions.

Regret@k The Regret@k metric is crucial for evaluating how well the task embeddings retrieve the beneficial task for top-1 prompt transfer performance. Its formula is as follows:

$$\text{Regret@k} = \frac{\max_{s\in S} \mathbb{E}[T(s,t)] - \max_{\tilde{s}\in S_k} \mathbb{E}[T(\tilde{s},t)]}{O(S)} \qquad (7)$$

Now, let's simplify the equation by understanding each term: $T(s,t)$ represents the performance achieved on the target task t when knowledge is transferred from the source task s. In simpler terms, it measures how effective insights from task s are in improving performance on task t. Moving on to $O(S,t)$, this term signifies the expected performance on the target task t under the optimal selection strategy. It establishes a performance benchmark achievable with the most advantageous source task selection. Finally, consider $M_k(S,t)$, which takes into account the highest performance observed on task t among the k top-ranked source tasks. This aspect evaluates the potential of the selected set of source tasks in contributing to superior performance on the target task t.

B Transfer Gains with Varying Training Data Sizes

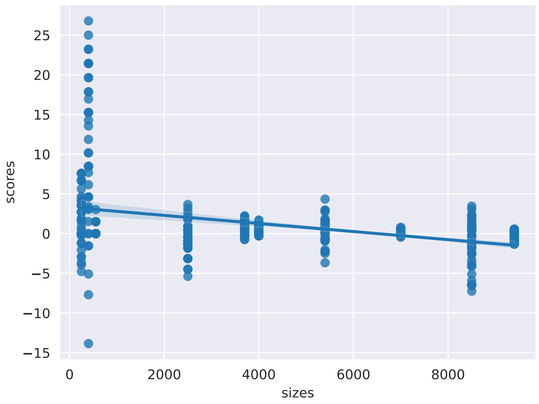


Figure 6: Transfer gains with soft prompt transfer. The dot on y-axis indicates the number of improved transfer performances compared to prompt tuning, while the x-axis enumerates the training set sizes on 10 downstream tasks.

We further explore how the training data size influences the relative performance. Figure 6 illustrates the correlation between the training split size and the level of transfer gains and losses. The plot shows 39 runs for each target task. Remarkably, tasks with extremely low resources (fewer than 1K training samples) exhibit a broad range of transfer gains and losses. Specifically, Tasks like COPA and CB with minimal training samples (400 and 250, respectively) show transfer gains varying from +25% to -15% in relative performance.

On the other hand, tasks with smaller variance in transfer gains, such as WSC and RTE, tend to have

Name	Task	Task category	Domain	|Train|	|Dev|	Metric
13 source tasks						
MNLI	NLI	Classification	Misc.	393K	9.8K	Acc.
QQP	Paraphrase detection	Classification	Social QA	364K	40.4K	F1/Acc.
QNLI	NLI	Classification	Wikipedia	105K	5.4K	Acc.
ReCoRD	QA	Multiple Choice	News articles	101K	10K	F1/EM
CxC	Semantic similarity	Classification	Misc.	88K	1K	Acc.
SQuAD	QA	QA	Wikipedia, crowd.	88K	10.6K	F1/EM
DROP	QA	QA	Wikipedia, crowd.	77K	9.5K	F1/EM
SST-2	Sentiment analysis	Classification	Movie reviews	67K	872	Acc.
WinoGrande	Commonsense reasoning	Multiple Choice	Crowdsourced	40K	1.2K	Acc.
HellaSWAG	Commonsense reasoning	Multiple Choice	Misc.	40K	10K	Acc.
MultiRC	QA	Classification	Misc.	27K	4.8K	F1$_\alpha$/EM
CosmosQA	Commonsense reasoning	Multiple Choice	Crowdsourced	25K	2.9K	Acc.
RACE	QA	Multiple Choice	English exams	25K	4.8K	Acc.
10 target tasks						
BoolQ	QA	Classification	Wikipedia, web queries	9K	3.2K	Acc.
CoLA	Grammatical acceptability	Classification	Books, journals	9K	1K	Matthews cor.
STS-B	Semantic similarity	Classification	Misc.	6K	1.5K	Pear./spear.
WiC	Word sense disambiguation	Classification	Misc.	5K	638	Acc.
CR	Sentiment analysis	Classification	Custom review	4K	753	Acc.
MRPC	Paraphrase detection	Classification	News	4K	408	F1/Acc.
RTE	NLI	Classification	Wikipedia, news	2K	277	Acc.
WSC	Coreference resolution	Classification	Fiction books	554	104	Acc.
COPA	QA	Multple Choice	Blog, encyclopedia	400	100	Acc.
CB	NLI	Classification	Misc.	250	56	F1/Acc.

Table 6: Statistics of source and target tasks. We categorize task types into three types: classification, QA, and multiple choice. We distinguish multiple choice tasks from QA tasks based on whether options are provided in the input.

fewer instances of positive transfer. This is influenced by a substantial number of runs achieving similar performance to baselines, leading to fewer positive transfers. Additionally, our prompt tuning settings, optimized for near-optimal performance, result in less pronounced benefits from prompt training.

The mean slope emphasizes trends, highlighting a strong correlation between the number of positive gains and the training sample sizes across most downstream tasks. Notably, the extent of performance improvement is more significant for tasks with smaller training sample sizes. However, despite high variance in relative performance, transfer gains tend to converge to zero when the dataset size reaches around 5K.

Prompt transfer's success is intricately tied to the data size of downstream tasks. Smaller training examples are more likely to exhibit positive transfer. While prompt transfer brings benefits, the presence of negative transfer underscores associated risks.

C Prompt Transfer Performance

Table 7 presents the mean performance across three runs on low-resource tasks, utilizing the best-performing soft prompt as the initialization point. As seen in previous studies, the prompt transfer results indicate improvements over the no-transfer baselines.

In particular, our most successful transfer results exhibit significant enhancements, surpassing the no-transfer outcomes on tasks such as COPA and CB by considerable margins, with improvements of +8% and +3.46%, respectively. However, it's noteworthy that the mean performance improvements for other tasks are relatively minor. This can be attributed to the extensive hyperparameter search conducted for the strong baseline (Prompt-Text), contrasting with the suboptimal nature of the weak baseline (Prompt-Abstract). This underlines the significance of optimization in the prompt tuning process.

Our exploration of prompt transfer performance sheds light on the nuanced dynamics at play, emphasizing the need for strategic optimization strategies in achieving robust and notable improvements,

	BoolQ	CoLA	STS-B	WiC	CR	MRPC	RTE	WSC	COPA	CB
Prompt-Abstract	$73.0_{1.2}$	$52.9_{1.2}$	$88.1_{0.6}$	$63.6_{1.6}$	$93.5_{0.2}$	$86.1_{0.7}$	$68.7_{1.2}$	$71.5_{1.7}$	$56.7_{1.7}$	$92.7_{1.9}$
Prompt-Text	$78.69_{0.18}$	$62.47_{1.51}$	$90.14_{0.20}$	$69.07_{0.45}$	$92.96_{0.29}$	$89.95_{0.52}$	$79.66_{0.74}$	$63.46_{0.00}$	$60.0_{3.74}$	$85.64_{2.21}$
MNLI	$78.36_{0.20}$	$61.55_{0.70}$	$90.22_{0.16}$	$69.07_{0.32}$	$93.18_{0.31}$	$90.93_{0.16}$	$78.45_{0.45}$	$63.46_{0.00}$	$63.00_{5.09}$	$87.62_{2.79}$
QQP	$78.66_{0.09}$	$61.68_{0.86}$	$90.29_{0.15}$	$68.44_{0.29}$	$92.96_{0.21}$	$90.69_{0.15}$	$80.14_{0.88}$	$\mathbf{64.42}_{0.78}$	$67.33_{2.86}$	$84.72_{1.02}$
QNLI	$\mathbf{78.80}_{0.15}$	$61.97_{0.79}$	$90.04_{0.13}$	$68.39_{0.14}$	$\mathbf{93.80}_{0.16}$	$90.49_{0.37}$	$77.61_{0.77}$	$63.46_{0.00}$	$61.33_{3.77}$	$88.67_{1.50}$
ReCoRD	$78.27_{0.18}$	$60.31_{0.23}$	$90.36_{0.10}$	$69.64_{0.63}$	$93.05_{0.06}$	$90.65_{0.47}$	$79.18_{0.61}$	$63.78_{0.45}$	$67.67_{1.70}$	$\mathbf{89.29}_{0.73}$
CxC	$78.71_{0.25}$	$62.49_{0.82}$	$90.12_{0.11}$	$69.59_{1.22}$	$93.45_{0.35}$	$90.62_{0.21}$	$79.30_{1.12}$	$63.46_{0.00}$	$68.00_{0.82}$	$86.60_{2.06}$
SQuAD	$78.80_{0.28}$	$61.43_{1.43}$	$90.17_{0.08}$	$69.49_{0.77}$	$93.63_{0.38}$	$90.41_{0.28}$	$77.74_{1.33}$	$63.78_{0.45}$	$65.67_{1.25}$	$87.15_{3.44}$
DROP	$78.37_{0.46}$	$61.01_{0.17}$	$90.23_{0.10}$	$69.12_{0.80}$	$93.71_{0.23}$	$\mathbf{91.22}_{0.47}$	$80.39_{0.45}$	$63.46_{0.00}$	$67.00_{2.16}$	$86.37_{2.37}$
SST-2	$78.56_{0.33}$	$61.36_{0.73}$	$89.91_{0.14}$	$69.64_{0.60}$	$93.54_{0.41}$	$90.35_{0.05}$	$78.46_{1.12}$	$63.78_{0.45}$	$61.67_{1.70}$	$86.93_{0.39}$
WinoGrande	$78.42_{0.13}$	$62.72_{1.02}$	$90.19_{0.11}$	$69.70_{1.04}$	$92.87_{0.17}$	$90.98_{0.44}$	$79.18_{1.23}$	$63.46_{0.00}$	$67.67_{1.25}$	$87.05_{2.40}$
HellaSWAG	$78.42_{0.30}$	$\mathbf{63.04}_{1.32}$	$\mathbf{90.46}_{0.10}$	$69.38_{0.77}$	$93.23_{0.11}$	$90.59_{0.25}$	$78.70_{0.59}$	$63.78_{0.45}$	$63.33_{5.25}$	$85.75_{2.05}$
MultiRC	$78.69_{0.02}$	$62.26_{0.46}$	$90.13_{0.15}$	$69.59_{0.22}$	$93.14_{0.27}$	$90.37_{0.12}$	$79.06_{1.53}$	$63.78_{0.45}$	$68.00_{2.16}$	$87.63_{0.31}$
CosmosQA	$78.47_{0.24}$	$61.40_{0.52}$	$90.10_{0.06}$	$\mathbf{70.22}_{1.02}$	$93.63_{0.11}$	$90.96_{0.20}$	$\mathbf{80.63}_{1.04}$	$63.46_{0.00}$	$66.67_{1.25}$	$87.46_{0.38}$
RACE	$78.24_{0.43}$	$61.05_{1.42}$	$90.16_{0.11}$	$68.70_{1.93}$	$93.67_{0.13}$	$90.67_{0.33}$	$80.39_{0.90}$	$63.46_{0.00}$	$\mathbf{68.00}_{0.00}$	$88.07_{2.56}$

Table 7: Results of prompt transfer. Downstream task performances involve soft prompt transfer between intermediate tasks (rows) and target tasks (columns) using the T5 base model. The first two rows represent the baseline performances with prompt tuning, without any pre-trained prompt weights. Prompt-Abstract refers to prompt tuning with the abstract symbol as a class label, and Prompt-Text refers to prompt tuning using the text span. Subsequent rows provide insights into prompt transfer performances, where the best-performing prompts from each task are transferred to ten different downstream tasks. All reported scores are mean values obtained from three random restarts.

especially in the context of low-resource tasks.

D More Results on the Effect of Task Type and Training Seed

Table 8 presents the top three prompt transfer results on eight downstream target tasks, along with their respective task types. These results reflect the most significant improvements in prompt transfer across three random seeds. On tasks with limited annotations, such as COPA and CB, different random seeds lead to substantial variance in transfer performance. Similarly, tasks like CoLA, WIC, and RTE also exhibit high variance. For WSC [†], we observed that most prompt transfer performances either present identical transfer gain or show no improvement in performance. This phenomenon is likely attributed to the unique task type of WSC compared to other downstream tasks. Specifically, the knowledge of source tasks has limited influence on performing the tasks.

E More Results on the Construction of Task Embeddings

Figure 7 analyzes how training steps for prompt tuning affect ranking prediction across various task embedding constructions, MAX, Feature and Unigram. We examined the prompt weights trained at intervals of 5K, up to 30K, using nDCG for ranking prediction. Three construction methods of task embeddings were compared across ten downstream tasks, indexed alphabetically from BoolQ (a) to CB (j).

Tasks with very limited data exhibit low nDCG scores. We found that the three methods performed well on five tasks, showing high nDCG scores. For instance, in BoolQ, STS-B, CR, MRPC, and WSC, all three methods demonstrated similar performance with relatively flat performance curves.

We further observed the significant variability in task prediction performance across four tasks: CoLA, RTE, COPA, and CB. Notably, COPA and CB presented considerable challenges due to their limited availability of labeled data. As a result, the computed nDCG scores for these tasks were notably lower compared to other downstream tasks, underscoring the difficulty in identifying effective intermediate tasks.

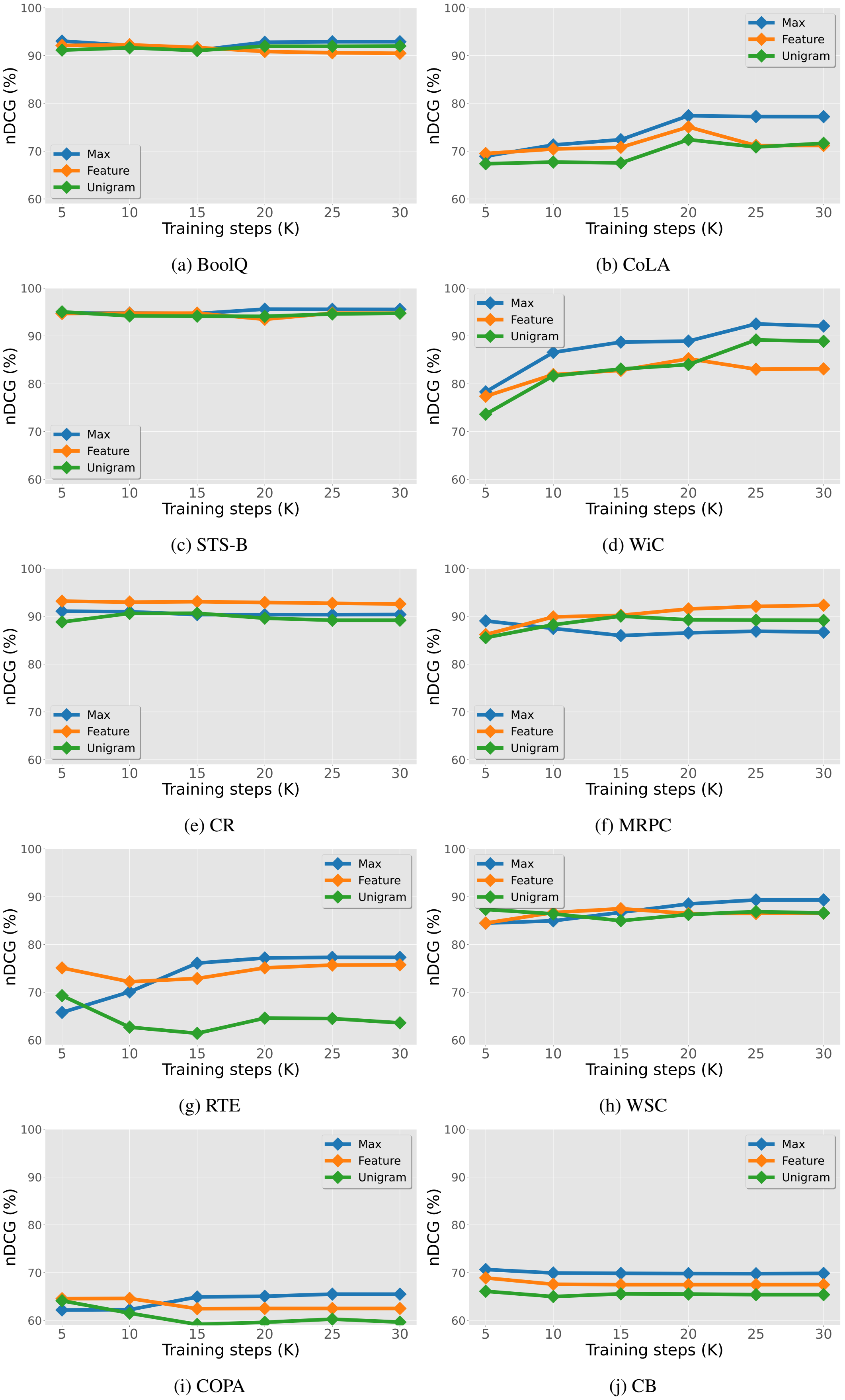
MAX yields superior performances in task prediction. Across 10 downstream tasks, we observed that MAX generally yields superior nDCG scores. On CoLA, RTE, and COPA, nDCG surpasses Feature after 15K training steps. For CB, MAX excels in capturing the essence between intermediate tasks during continual prompt tuning on challenging low-resource tasks. This highlights the importance of measuring token-wise similarity between source and target prompts for improved performance. Our analysis suggests that MAX method tends to perform better in certain scenarios, em-

Target	seed 112			28			52		
	Source	Task Type	Rel. (%)	Source	Task Type	Rel. (%)	Source	Task Type	Rel. (%)
Top-3 transfer									
BoolQ (QA)	DROP*	QA	0.58	SQuAD*	QA	0.31	CxC	senti. similarity	0.31
	SST-2	sentiment	0.55	QQP	paragraph	-0.12	QNLI	NLI	0.27
	HellaSWAG	commonsense	0.50	QNLI	NLI	-0.15	SQuAD*	QA	0.11
CoLA (grammatical acceptability)	WinoGrande	commonsense	3.47	WinoGrande	commonsense	3.10	HellaSWAG	commonsense	0.44
	RACE	QA	2.47	CxC	senti. similarity	2.10	CxC	senti. similarity	-1.79
	MultiRC	QA	2.24	QQP	paragraph	1.65	QNLI	NLI	-2.35
STS-B (sentiment similarity)	ReCoRD	QA	0.16	ReCoRD	QA	0.18	HellaSWAG	commonsense	0.81
	HellaSWAG	commonsense	0.08	HellaSWAG	commonsense	0.16	QQP	paragraph	0.68
	DROP	QA	0.07	WinoGrande	commonsense	0.08	MultiRC	QA	0.52
WiC (word sense disambiguation)	WinoGrande	commonsense	2.95	ReCoRD	QA	1.35	CosmosQA	commonsense	4.35
	CxC	senti. similarity	1.81	SQuAD	QA	1.13	CxC	senti. similarity	2.98
	CosmosQA	commonsense	1.59	SST-2	sentiment	0.90	HellaSWAG	commonsense	2.75
CR (sentiment)	SST-2*	sentiment	0.71	SQuAD	QA	1.72	DROP	QA	1.00
	CosmosQA/RACE	commonsense/QA	0.57	QNLI	NLI	1.58	QNLI/SST-2*	NLI/sentiment	0.71
	MNLI/QNLI	NLI/NLI	0.43	CxC	senti. similarity	1.29	CxC/CosmosQA	senti. similarity/commonsense	0.57
MRPC (paraphrase)	DROP	QA	2.24	WinoGrande	commonsense	2.19	DROP	QA	0.95
	CosmosQA	commonsense	1.85	RACE	QA	1.68	MNLI	NLI	0.48
	QQP*	paragraph	1.75	ReCoRD	QA	1.66	CosmosQA	commonsense	0.27
RTE (NLI)	MultiRC	QA	1.81	RACE	QA	3.67	CosmosQA	commonsense	1.79
	QQP/RACE	paragraph/QA	0.45	QQP	paragraph	3.21	CxC/WinoGrande	senti. similarity/commonsense	0.45
	DROP	QA	0.00	DROP	QA	2.75	DROP	QA	0.00
WSC† (coreference resolution)	QQP/SQuAD/SST-2	paragraph/QA/sentiment	1.52	ReCoRD/MultiRC	QA/QA	1.52	QQP	paragraph	3.03
	MNLI/QNLI	NLI/NLI	0.00	MNLI/QQP/QNLI	NLI/paraphrase/NLP	0.00	MNLI/QNLI	NLI/NLI	0.00
	-	-	-	-	-	-	-	-	-

Table 8: Top-3 prompt transfer on eight downstream target tasks and their task types. The three most significant improvements in prompt transfer across three random seeds, 112, 28, and 52. The relative performance is reported as a percentage (%) and calculated based on the corresponding no-transfer prompt-tuning. * indicates that the source task type is identical to the task type of the downstream task.

phasizing its effectiveness in ranking prediction compared to other methods.

Longer training leads to better performance. Furthermore, **MAX** achieves higher task prediction performance with longer training steps. Furthermore, **MAX** achieves higher task prediction performance with longer training steps. For example, in tasks such as CoLA, WiC, and RTE, **MAX** shows marked improvements in the ranking prediction with extended training durations.



(a) BoolQ

(b) CoLA

(c) STS-B

(d) WiC

(e) CR

(f) MRPC

(g) RTE

(h) WSC

(i) COPA

(j) CB

Figure 7: Comparison of task embedding construction methods on various training steps, with intervals of 5K. The x-axis denotes the training steps of prompt-tuning.