On Transferability of Prompt Tuning for Natural Language Processing

Anonymous ACL submission

Abstract

Prompt tuning (PT) is a promising parameter-efficient method to utilize extremely large pre-trained language models (PLMs), which can achieve comparable performance to full-parameter fine-tuning by only tuning a few soft prompts. However, PT requires much more training time than fine-tuning. Intuitively, knowledge transfer can help to improve the efficiency. To explore whether we can improve PT via prompt transfer, we empirically investigate the transferability of soft prompts across different downstream tasks and PLMs in this work. We find that (1) in zero-shot setting, trained soft prompts can effectively transfer to similar tasks on the same PLM and also to other PLMs with a cross-model projector trained on similar tasks; (2) when used as initialization, trained soft prompts of similar tasks and projected prompts of other PLMs can significantly accelerate training and also improve the performance of PT. Moreover, to explore what decides prompt transferability, we investigate various transferability indicators and find that the overlapping rate of activated neurons strongly reflects the transferability, which suggests how the prompts *stimulate* PLMs is essential. Our findings show that prompt transfer is promising for improving PT, and further research shall focus more on prompts' stimulation to PLMs. The source code will be publicly released.

1 Introduction

Pre-trained language models (PLMs), such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018) have achieved great performance on various natural language processing (NLP) tasks (Han et al., 2021). Recently, after the success of GPT-3 (Brown et al., 2020), people have found that extremely large PLMs can achieve remarkable improvements, and various large PLMs are continually developed (Raffel et al., 2020; Zhang et al., 2021; Zeng et al., 2021; Wei et al., 2021; Sun et al., 2021), which contain up to hundreds of billions of parameters.

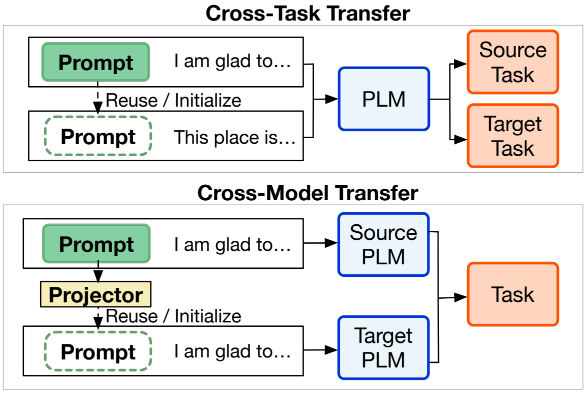


Figure 1: We explore prompt transferring across different tasks (cross-task) and PLMs (cross-model) with directly reusing prompts and initializing prompt tuning.

Considering the extremely large scale of these state-of-the-art PLMs, conventional full-parameter fine-tuning methods become extremely expensive. Hence, various parameter-efficient tuning methods (Houlsby et al., 2019; Ben Zaken et al., 2021; Lester et al., 2021; Li and Liang, 2021; Liu et al., 2021) are explored, among which prompt tuning (PT) has attracted broad research attention. PT prepends some *soft prompts*, which are essentially learnable virtual tokens, into the input sequences and only train them while keeping all the PLM's parameters fixed. The training objective is to generate desired outputs in the same way as the pre-training tasks. PT can match the downstream task performance of fine-tuning with only thousands of tunable parameters (Lester et al., 2021) when the PLM has billions of parameters.

Although PT is an effective approach to utilize extremely large PLMs, it requires much more training time than fine-tuning to reach the convergence as shown in Figure 2; hence, it is worthwhile to explore how to improve the efficiency of PT. In this work, we attempt to improve PT via **prompt transfer** across different tasks and models. Knowledge

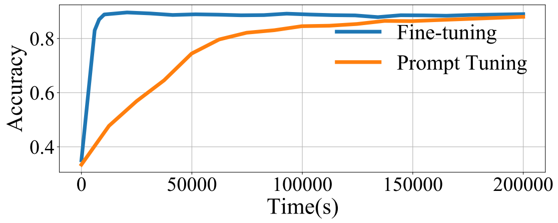


Figure 2: Validation accuracies against training time of fine-tuning and PT for RoBERTa$_{\text{LARGE}}$ on MNLI. PT takes much more training time.

transfer across tasks (Vu et al., 2020) and models (Qin et al., 2021) have been widely used to improve the efficiency and effectiveness of NLP systems. Intuitively, soft prompts are the only tuned parameters in PT and thus shall concentrate the knowledge required to solve tasks conditioned on PLMs. Hence only transferring the trained prompts is promising to accelerate PT.

As shown in Figure 1, we empirically analyze the transferability of prompts across different tasks (*cross-task transfer* setting) and PLMs (*cross-model transfer* setting) in this paper. The empirical analysis is conducted on 17 NLP tasks of 6 types and two representative PLM series: RoBERTa (Liu et al., 2019b) and T5 (Raffel et al., 2020). In cross-task transfer, the prompt transfer can be done by directly reusing the trained prompts of the source task on the target task. However, in cross-model transfer, directly reusing prompts is intractable since the semantic spaces of different PLMs are inconsistent; hence, we develop various **prompt projectors** to project the soft prompts trained on the source PLM to the semantic space of the target PLM. We conduct two lines of experiments: (1) We investigate the **zero-shot transfer performance** and find that the transferability of prompts is influenced by task types. In cross-task transfer, the soft prompts can directly transfer to same-type tasks and achieve non-trivial performance, but poorly transfer to different-type tasks requiring different language skills. In cross-model transfer, we can successfully train a prompt projector with PT on a task, but the trained projector also only well generalizes to the same-type tasks of the projector-training task. (2) To accelerate PT, we propose to **transfer prompts with initialization**. In cross-task transfer, we start PT with the trained soft prompts of similar tasks as initialization. While in cross-model transfer, the initialization is the projected prompts of the same task trained on the source PLM. The two methods are dubbed as TPT$_{\text{TASK}}$ and TPT$_{\text{MODEL}}$, respectively. Experiments show that they can both significantly accelerate PT and also achieve a certain performance improvement.

Furthermore, we explore why can the prompts transfer and what decides their transferability. To this end, we design various prompt similarity metrics from different perspectives and examine how well they can serve as **transferability indicators**, i.e., how well they correlate with prompt transfer performance. Experiments find that the embedding distances of prompts do not well indicate prompt transferability but the overlapping rate of the prompts' activated neurons in the feed-forward layers can better reflect prompt transferability. This suggests the prompts are essentially stimulating PLM's inner ability distributing among neurons to do specific NLP tasks, and future prompt transfer works should focus more on how the PLMs respond to different prompts' stimulation rather than the prompts' embedding properties.

To summarize, our contributions are three-fold: (1) We thoroughly analyze the transferability of prompts across different tasks and models, and show that improving PT with prompt transfer is possible and promising. (2) We propose to transfer prompts with initialization, which enhances both PT's efficiency and effectiveness. (3) We explore the effectiveness of various prompt similarity metrics serving as transferability indicators and demonstrate how the prompts stimulate PLMs to decide the transferability, which may facilitate further transferrable PT research.

2 Related Work

Prompt Tuning GPT-3 (Brown et al., 2020) demonstrates remarkable few-shot performance by prepending textual prompts before the inputs and thus help the PLM to generate desired outputs of NLP tasks directly. Motivated by this, many works have tried to improve various NLP tasks by creating manually-crafted (Schick and Schütze, 2021a,b; Mishra et al., 2021) or automatically-searched (Jiang et al., 2020; Shin et al., 2020; Gao et al., 2021) *hard prompts*, which are discrete tokens but not necessarily human-readable. Furthermore, *soft prompts* (Li and Liang, 2021; Hambardzumyan et al., 2021; Zhong et al., 2021; Liu et al., 2021) are proposed, which are tuneable embeddings rather than tokens in the vocabularies and can be directly trained with task-specific supervision. Lester et al. (2021) demonstrate that prompt tuning (PT) method can match the performance of full-parameter fine-tuning when the PLM has billions of parameters. This suggests that PT is promising to utilize extremely large PLMs. However, the much more training time needed to reach the convergence makes PT inefficient. In this work, we show that prompt transfer can remedy, improve the effectiveness to some extent with knowledge transfer, and empirically analyze the transferability of prompts across tasks and PLMs.

Knowledge Transfer Cross-task knowledge transfer (Ruder, 2017) has been a long-standing way to improve the effectiveness and efficiency of NLP systems. In the PLM era, some works propose to tune the PLMs on intermediate tasks (Phang et al., 2018; Pruksachatkun et al., 2020; Gururangan et al., 2020; Wang et al., 2019a; Vu et al., 2020; Poth et al., 2021) before fine-tuning on specific target tasks to achieve certain benefits. Vu et al. (2020) empirically analyze the transferability between tasks in this setting.

These explorations are all for fine-tuning. Considering the potential of PT, we believe the transferability and knowledge transfer methods for PT are worth exploring. As a prior attempt, Lester et al. (2021) demonstrate that PT's cross-domain transferability is stronger than fine-tuning. Similar to our work, concurrent work (Vu et al., 2021) explores the cross-task transferability of PT and improves performance with transfer initialization. Differently, we attempt to improve the efficiency of PT and further analyze what decides the prompt transferability by exploring various transferability indicators. Additionally, we also attempt cross-model transfer, which is inspired by previous cross-model knowledge transfer works such as Net2Net (Chen et al., 2016), knowledge distillation (Hinton et al., 2015) and knowledge inheritance (Qin et al., 2021).

3 Preliminary

Here we introduce the basic knowledge about PT (§ 3.1) as well as the downstream tasks (§ 3.2) and models (§ 3.3) investigated in experiments.

3.1 Prompt Tuning

In this work, we study the PT method that is capable of tuning large PLMs (Li and Liang, 2021; Lester et al., 2021; Liu et al., 2021), i.e., we only explore the PT method freezing PLM parameters. PT prepends some virtual tokens, i.e., the *soft prompts*, into the inputs of the PLM to provide knowledge about downstream tasks. The soft prompts are essentially tunable embedding vectors, which are trained with the objective enforcing the PLM to generate desired outputs of the downstream task in the same way of the pre-training objective.

Formally, given an input sequence with n tokens $X = \{x_1, x_2, \ldots, x_n\}$, we first prepend l randomly initialized soft prompts $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_l\}$ before them, where $\mathbf{p}_i \in \mathbb{R}^d$ is an embedding vector, and d is the input dimension of the PLM. The training objective is to maximize the likelihood of decoding the desired output y:

$$\mathcal{L} = p(y|P, x_1, \ldots, x_n), \qquad (1)$$

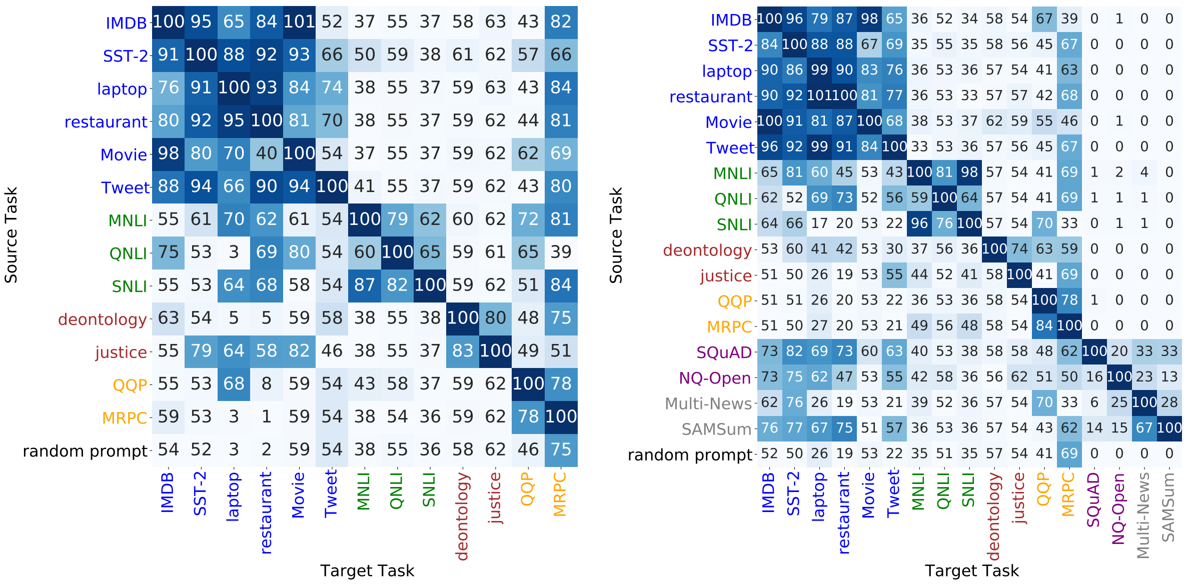
where only P is learnable. For the language understanding tasks, y is the label token corresponding to the label of X. For the conditional generation tasks, y is a sequence. Especially, for the models pre-trained with the masked language modeling objective like RoBERTa, we additionally prepend a special `[MASK]` token before the prompts and train the prompts to let the PLM fill y into it.

3.2 Investigated NLP Tasks

To comprehensively study the prompt transferability across various NLP tasks, we involve 17 diverse tasks, which can be divided into 6 types: (1) **Sentiment Analysis (SA)**, including `IMDB` (Maas et al., 2011), `SST-2` (Socher et al., 2013), `laptop` (Pontiki et al., 2014), `restaurant` (Pontiki et al., 2014), Movie Rationales (`Movie`) (Zaidan et al., 2008) and TweetEval (`Tweet`) (Barbieri et al., 2020); (2) **Natural Language Inference (NLI)**, including `MNLI` (Williams et al., 2018), `QNLI` (Wang et al., 2019b) and `SNLI` (Bowman et al., 2015); (3) **Ethical Judgement (EJ)**, including `deontology` (Hendrycks et al., 2021) and `justice` (Hendrycks et al., 2021); (4) **Paraphrase Identification (PI)**, including `QQP` (Sharma et al., 2019) and `MRPC` (Dolan and Brockett, 2005); (5) **Question Answering (QA)**, including `SQuAD` (Rajpurkar et al., 2016) and `NQ-Open` (Lee et al., 2019); (6) **Summarization (SUM)**, including `Multi-News` (Fabbri et al., 2019) and `SAMSum` (Gliwa et al., 2019). Details for these tasks, evaluation metrics, label tokens, implementations are in appendix A.

3.3 Investigated Models

We investigate prompt transferability for two series of PLMs: RoBERTa (Liu et al., 2019b) and T5 (Raf-



Source Task	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC
IMDB	100	95	65	84	101	52	37	55	37	58	63	43	82
SST-2	91	100	88	92	93	66	50	59	38	61	62	57	66
laptop	76	91	100	93	84	74	38	55	37	59	63	43	84
restaurant	80	92	95	100	81	70	38	55	37	59	62	44	81
Movie	98	80	70	40	100	54	37	55	37	59	62	62	69
Tweet	88	94	66	90	94	100	41	55	37	59	62	43	80
MNLI	55	61	70	62	61	54	100	79	62	60	62	72	81
QNLI	75	53	3	69	80	54	60	100	65	59	61	65	39
SNLI	55	53	64	68	58	54	87	82	100	59	62	51	84
deontology	63	54	5	5	59	58	38	55	38	100	80	48	75
justice	55	79	64	58	82	46	38	55	37	83	100	49	51
QQP	55	53	68	8	59	54	43	58	37	59	62	100	78
MRPC	59	53	3	1	59	54	38	54	36	59	62	78	100
random prompt	54	52	3	2	59	54	38	55	36	58	62	46	75

(a) RoBERTa$_{\text{LARGE}}$

Source Task	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC	SQuAD	NQ-Open	Multi-News	SAMSum
IMDB	100	96	79	87	98	65	36	52	34	58	54	67	39	0	1	0	0
SST-2	84	100	88	88	67	69	35	55	35	58	56	45	67	0	0	0	0
laptop	90	86	99	90	83	76	36	53	36	57	54	41	63	0	0	0	0
restaurant	90	92	101	100	81	77	36	53	33	57	57	42	68	0	0	0	0
Movie	100	91	81	87	100	68	38	53	37	62	59	55	46	0	1	0	0
Tweet	96	92	99	91	84	100	33	53	36	57	56	45	67	0	0	0	0
MNLI	65	81	60	45	53	43	100	81	98	57	54	41	69	1	2	4	0
QNLI	62	52	69	73	52	56	59	100	64	57	54	41	69	1	1	1	0
SNLI	64	66	17	20	53	22	96	76	100	57	54	70	33	0	1	1	0
deontology	53	60	41	42	53	30	37	56	36	100	74	63	59	0	0	0	0
justice	51	50	26	19	53	55	44	52	41	58	100	41	69	0	0	0	0
QQP	51	51	26	20	53	22	36	53	36	58	54	100	78	1	0	0	0
MRPC	51	50	27	20	53	21	49	56	48	58	54	84	100	0	0	0	0
SQuAD	73	82	69	73	60	63	40	53	38	58	58	48	62	100	20	33	33
NQ-Open	73	75	62	47	53	55	42	58	36	56	62	51	50	16	100	23	13
Multi-News	62	76	26	19	53	21	39	52	36	57	54	70	33	6	25	100	28
SAMSum	76	77	67	75	51	57	36	53	36	57	54	43	62	14	15	67	100
random prompt	52	50	26	19	53	22	35	51	35	57	54	41	69	0	0	0	0

(b) T5$_{\text{XXL}}$

Figure 3: Relative zero-shot transfer performance (zero-shot transfer performance / original PT performance) (%) on the target tasks (columns) of the soft prompts trained on the source tasks (rows) for RoBERTa$_{\text{LARGE}}$ and T5$_{\text{XXL}}$. Colors of the task names indicate task types. Blue: SA. Green: NLI. Brown: EJ. Orange: PI. Purple: QA. Gray: SUM. *Random Prompt* of the last row means the soft prompts are randomly generated without any training.

fel et al., 2020), which represent two mainstream pre-training types: masked language modeling and sequence-to-sequence pre-training. Considering RoBERTa can only predict a single token (or a fixed length of tokens), for the conditional generation tasks (QA and SUM) that output multiple tokens, we only investigate T5. We mainly report results for the two largest versions of PLMs, i.e., RoBERTa$_{\text{LARGE}}$ and T5$_{\text{XXL}}$. The more detailed results for the other sizes are attached in appendix.

4 Cross-Task Transfer

We empirically study the cross-task transferability of soft prompts (§ 4.1) and try to improve the effectiveness and efficiency of PT with transfer (§ 4.2).

4.1 Zero-shot Transfer Performance

To study the cross-task transferability, we first examine PT's zero-shot transfer performance, i.e., we conduct PT on a source task, then directly reuse the trained prompts on other target tasks and evaluate their performance. The results are shown in Figure 3[1], from which we can observe that: (1) For the tasks within the same type, transferring soft prompts between them can generally perform well and may even outperform vanilla PT on the target task, especially when the source task has more data (the case of transferring from `IMDB` to `Movie` in Figure 3 (a) and transferring from `restaurant` to `laptop` in Figure 3 (b)), which demonstrates that it is promising to improve PT's effectiveness and efficiency with knowledge transfer from similar tasks. (2) For the tasks of different types, the transferability of soft prompts among them is generally poor, and transferring soft prompts often achieve similar performance to randomly initialized prompts. (3) However, some tasks can transfer to different-type tasks to some extent, such as the QA and SUM tasks to SA tasks in Figure 3 (b). To understand this, it is worthwhile to explore what controls the transferability between prompts, and we do some preliminary study in § 6.

4.2 Transfer with Initialization

To improve the effectiveness and efficiency of PT with cross-task transfer, we explore a cross-task transferable prompt tuning (TPT$_{\text{TASK}}$) method, which initializes soft prompts with well-trained prompts of the most similar task and then starts PT.

For a target task, we start TPT$_{\text{TASK}}$ with trained prompts of the source task achieving the best zero-shot transfer performance in Figure 3. From the results of the performance and training time com-

[1] More results on other PLMs are left in appendix B.1.

Task Type	**SA**						**NLI**			**EJ**		**PI**		**QA**		**SUM**	
Task	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC	SQuAD	NQ-Open	Multi-News	SAMSum
Metric	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	F1	F1	ROUGE-L	ROUGE-L
RoBERTa$_{\text{LARGE}}$																	
Performance (PT) (%)	92.2	96.1	76.4	83.7	84.9	**76.1**	87.3	92.4	**91.9**	**85.6**	**81.0**	**88.9**	**81.2**	N/A	N/A	N/A	N/A
Performance (TPT$_{\text{TASK}}$) (%)	**92.4**	**96.3**	**79.1**	**85.8**	**85.1**	**76.1**	**87.9**	**93.1**	**91.9**	**85.6**	78.2	86.1	79.2	N/A	N/A	N/A	N/A
Convergence Speedup	1.7	1.1	1.0	1.9	1.2	0.9	1.2	1.2	1.3	0.9	0.7	0.8	0.9	N/A	N/A	N/A	N/A
Comparable-result Speedup	2.5	2.4	1.0	3.8	1.5	1.3	1.1	2.3	1.0	0.9	N/A	N/A	N/A	N/A	N/A	N/A	N/A
T5$_{\text{XXL}}$																	
Performance (PT) (%)	96.5	97.4	76.6	**90.1**	**97.9**	76.2	90.5	95.2	93.4	87.0	**92.5**	90.0	86.3	**86.3**	20.8	29.2	45.8
Performance (TPT$_{\text{TASK}}$) (%)	**96.6**	**97.8**	**84.2**	88.6	97.5	**77.0**	**92.0**	**96.2**	**94.0**	**95.3**	90.7	**90.9**	**89.0**	85.9	**21.3**	**29.3**	**46.8**
Convergence Speedup	1.2	49.7	2.2	1.1	3.9	1.4	12.5	24.9	49.9	29.8	1.5	1.0	3.3	1.1	1.0	2.0	2.0
Comparable-result Speedup	1.2	48.9	219.8	N/A	N/A	1.5	12.5	29.9	49.9	29.9	N/A	1.0	5.0	N/A	1.0	2.0	2.5

Table 1: Performance on 17 NLP tasks of vanilla prompt tuning ($\mathbf{PT}$) and prompt tuning with transferring inital-ization (TPT$_{\text{TASK}}$) as well as the convergence speedup (the quotient of the training steps of $\mathbf{PT}$ by the training time of TPT$_{\text{TASK}}$ reaching convergence) and comparable-result speedup (the quotient of the training time of $\mathbf{PT}$ by the training time of TPT$_{\text{TASK}}$ achieving comparable performance to $\mathbf{PT}$). N/A represents the tasks that RoBERTa$_{\text{LARGE}}$ cannot conduct, or we fail to speed up training with TPT$_{\text{TASK}}$.

parisons[2] in Table 1, we can see TPT$_{\text{TASK}}$ can mostly achieve better or comparable performance to vanilla PT starting from random initialization, and TPT$_{\text{TASK}}$ generally takes less training time.

5 Cross-Model Transfer

We further study the cross-model transferability of soft prompts. We investigate the feasibility of cross-model transfer on transferring from a source PLM (RoBERTa$_{\text{LARGE}}$) to a larger and heterogeneous target PLM (T5$_{\text{XXL}}$), which shall be the most difficult setting. Appendix C shows the experimental results of other settings. Directly reusing trained soft prompts between different PLMs is infeasible since their embedding spaces are different. Hence, we investigate how to do cross-model prompt projection (§ 5.1) and see the transfer performance (§ 5.2). Furthermore, we explore to improve PT with cross-model transfer initialization (§ 5.3).

5.1 Cross-Model Prompt Projection

To project the trained soft prompts of a PLM to the semantic space of a different PLM, we train projectors with various objectives and examine their effectiveness. A good way to train the cross-model projectors may need some task-specific supervisions, but the trained projector shall generalize to different tasks so that the efficiency for learning the new tasks on the target model could be improved.

Formally, given the prompt of the source PLM $P^s = \{\mathbf{p}_1^s, \ldots, \mathbf{p}_l^s\}$, we concatenate the l virtual tokens into a unified vector $\mathbf{P}^s \in \mathbb{R}^{ld_s}$. The projector $\mathbf{Proj}(\cdot)$ is to project it to $\tilde{\mathbf{P}}^s \in \mathbb{R}^{ld_t}$ in the semantic space of the target PLM, where d_s and d_t are the input embedding dimensions of the source and target PLM, respectively. We parameterize the projector with a two-layer perceptron as follows:

$$\tilde{\mathbf{P}}^s = \mathbf{Proj}(\mathbf{P}^s) = \mathbf{W}_2(\sigma(\mathbf{P}^s\mathbf{W}_1 + \mathbf{b}_1)) + \mathbf{b}_2, \quad (2)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_h \times ld_s}, \mathbf{W}_2 \in \mathbb{R}^{ld_t \times d_h}$ are trainable matrices, $\mathbf{b}_1 \in \mathbb{R}^{d_h}, \mathbf{b}_2 \in \mathbb{R}^{ld_t}$ are biases, σ is a non-linear activation function. We investigate two learning objectives to train the projector[3]:

Distance Minimizing We firstly try to learn cross-model projections by minimizing the distance between the projected prompt and the parallel prompt $\mathbf{P}^t$ originally trained on the target PLM with the same task, i.e., the training objective is to minimize their L_2-distance $\|\mathbf{Proj}(\mathbf{P}^s) - \mathbf{P}^t\|_2$.

Task Tuning We then try to train the cross-model projector with task-specific supervision signals on the target PLM. Specifically, we directly tune the projected prompts on some tasks and back propagate the supervision signals to train the projector.

These methods rely on some tasks (parallel trained soft prompts or training data) to train the projector. In the experiments, we select `laptop` and `MNLI` for the projector learning.

5.2 Zero-shot Transfer Performance

The zero-shot transfer performance of various projector-learning methods are shown in Table 2[4] (a). We can observe that: (1) **Distance Minimizing** works well to transfer the prompts of the projector-training task, but falls back to random performance on the other unseen tasks, which is not practically

[2] Training time comparisons are left in appendix B.3.

[3] More projector-training details are left in appendix C.1.

[4] More results on other PLMs are left in appendix C.2.

Method		SA						NLI			EJ		PI	
		IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC
PT on T5$_{\text{XXL}}$		96.5	97.4	76.6	88.1	97.9	72.5	90.5	95.2	93.4	87.0	92.5	90.0	86.3
Random Prompt		49.7	49.0	19.8	17.0	51.6	15.5	31.8	49.3	31.9	51.3	50.0	36.4	67.0
(a) Zero-shot Transfer Performance (%)														
`laptop`	Distance Minimizing	49.6	49.0	76.6	17.5	51.5	14.4	31.8	48.1	32.8	53.3	49.9	36.8	66.6
	Task Tuning	**82.9**	**89.3**	**80.3**	**85.7**	**78.6**	**58.4**	32.4	50.7	33.6	54.9	51.6	33.9	63.7
`MNLI`	Distance Minimizing	49.6	50.1	19.8	18.3	51.2	15.0	**90.5**	49.0	32.9	50.3	49.0	36.8	65.6
	Task Tuning	49.7	48.8	19.8	17.0	51.6	16.0	89.8	**82.7**	**88.2**	49.7	50.0	36.8	67.7
(b) Transfer with Initialization (TPT$_{\text{MODEL}}$)														
`laptop`	Performance (%)	96.5	97.4	82.9	90.3	97.4	74.4	91.0	95.4	93.4	92.5	92.5	90.0	87.9
	Convergence Speedup	1.1	1.7	1.9	1.3	0.6	1.3	0.9	0.9	1.0	1.0	0.7	1.1	1.1
	Comparable-result Speedup	1.0	19.0	16.0	6.0	N/A	2.2	3.6	1.1	6.0	6.0	0.9	1.8	3.4
`MNLI`	Performance (%)	96.5	97.4	82.7	88.5	95.8	74.7	91.2	95.9	93.5	94.6	92.5	90.0	87.7
	Convergence Speedup	1.0	1.6	1.8	0.9	0.4	1.3	1.0	1.1	1.4	2.0	1.7	0.9	0.9
	Comparable-result Speedup	1.0	18.0	15.0	1.6	N/A	1.5	18.0	20.0	30.0	7.5	5.0	1.5	1.9

Table 2: Cross-model prompt transfer (RoBERTa$_{\text{LARGE}}$ to T5$_{\text{XXL}}$) results, including non-transfer baselines (vanilla PT and randomly generated prompts), zero-shot transfer performance of various projectors, and TPT$_{\text{MODEL}}$ results (performance, convergence speedup, and comparable-result speedup similar to Table 1).

usable. This is consistent with our findings in § 6 that the embedding distances do not strongly correlate to prompt transferability. (2) **Task Tuning** performs better and successfully generalizes to same-type unseen tasks of the projector-training tasks (e.g. NLI tasks for the projectors trained with `MNLI`), which proves the feasibility of practical cross-model prompt transfer. (3) The projectors trained with **Task Tuning** still cannot work for different-type tasks, which may be limited by the cross-task prompt transferability investigated in § 4.1. This urges further attention on developing universal cross-model projections.

5.3 Transfer with Initialization

Similar to § 4.2, we further study whether the projected soft prompts can initialize PT on the target PLM and accelerate training as well as improve performance. We propose cross-model transferable prompt tuning, TPT$_{\text{MODEL}}$, which adopts the **Task Tuning** projectors to project the soft prompts trained on the source PLM into the target PLM and initialize PT with the projected prompts.

The performance and speedup are shown in Table 2 (b). We can see that, for the tasks within the same type of the projector-training task, compared to vanilla PT, TPT$_{\text{MODEL}}$ can mostly achieve comparable or better performance with much less training time, which demonstrates that practical cross-model prompt transfer is promising for improving the efficiency and effectiveness of PT.

6 Exploring Transferability Indicator

Based on the positive results in cross-task and cross-model transfer, we explore why the soft prompts can transfer across tasks and what decides the transferability between them, which may shed light on the mechanisms behind PT and help to design transferable PT methods. We explore various **prompt similarity metrics** and examine how well do they align with the zero-shot transfer performance. If a similarity metric can well indicate transferability, it suggests the factors considered in designing this metric decide the prompt transferability. Moreover, the prompt similarity metrics can qualify task similarities using the trained soft prompts as task embeddings and may help in developing cross-task transfer methods. As a straightforward example, if we build a *prompt warehouse* containing prompts of diverse tasks, we can retrieve prompts of similar tasks for a new task with a certain similarity metric and better improve PT with TPT$_{\text{TASK}}$.

6.1 Prompt Similarity Metric

We explore the following two kinds of metrics:

Embedding Similarity We firstly regard the trained soft prompts as only embeddings in the vector space and calculate their *Euclidean similarity* and *cosine similarity*.

Given two groups of trained prompts containing l virtual tokens: $P^{t_1} = \{\mathbf{p}_1^{t_1}, \ldots, \mathbf{p}_l^{t_1}\}$ and $P^{t_2} = \{\mathbf{p}_1^{t_2}, \ldots, \mathbf{p}_l^{t_2}\}$, which correspond to tasks t_1 and t_2. Firstly, we concatenate the l virtual tokens for each group and get two concatenation embeddings $\mathbf{P}^{t_1}, \mathbf{P}^{t_2} \in \mathbb{R}^{ld}$, then we compute Euclidean similarity and cosine similarity of them:

$$\begin{aligned} \mathrm{E}_{\mathrm{concat}}(P^{t_1}, P^{t_2}) &= \frac{1}{1 + \|\mathbf{P}^{t_1} - \mathbf{P}^{t_2}\|}, \\ \mathrm{C}_{\mathrm{concat}}(P^{t_1}, P^{t_2}) &= \frac{\mathbf{P}^{t_1} \cdot \mathbf{P}^{t_2}}{\|\mathbf{P}^{t_1}\|\|\mathbf{P}^{t_2}\|}. \end{aligned} \tag{3}$$

We further explore a simple way to make the metrics invariant to token positions. We compute Euclidean distances and cosine similarities for every virtual token pairs in the two groups and use the averaged results in the final similarity metrics:

$$\begin{aligned} \mathrm{E}_{\mathrm{average}}(P^{t_1}, P^{t_2}) &= \frac{1}{1 + \frac{1}{l^2}\sum_{i=1}^{l}\sum_{j=1}^{l}\|\mathbf{p}_i^{t_1} - \mathbf{p}_j^{t_2}\|}, \\ \mathrm{C}_{\mathrm{average}}(P^{t_1}, P^{t_2}) &= \frac{1}{l^2}\sum_{i=1}^{l}\sum_{j=1}^{l}\frac{\mathbf{p}_i^{t_1} \cdot \mathbf{p}_j^{t_2}}{\|\mathbf{p}_i^{t_1}\|\|\mathbf{p}_j^{t_2}\|}. \end{aligned} \tag{4}$$

Model Stimulation Similarity In the second way, we depict their similarities based on how they *stimulate the PLMs*, i.e., we examine the similarities between the responses of PLMs to the two soft prompts. Motivated by Geva et al. (2021) and Dai et al. (2021), which both find that the activation of the neurons in the feed-forward layers of Transformers (Vaswani et al., 2017) corresponds to specific model behaviors, we propose to use the *overlapping rate of activated neurons* as a similarity metric of prompts. Specifically, the feed-forward network $\mathrm{FFN}(\cdot)$ in a Transformer layer is:

$$\mathrm{FFN}(\mathbf{x}) = \max(\mathbf{x}\mathbf{W}_1^\top + \mathbf{b}_1, \mathbf{0})\mathbf{W}_2 + \mathbf{b}_2, \tag{5}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the input embedding, $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d_m \times d}$ are trainable matrices, and $\mathbf{b}_1, \mathbf{b}_2$ are bias vectors. The $\max(\mathbf{x}\mathbf{W}_1^\top + \mathbf{b}_1, \mathbf{0})$ can be regarded as the non-negative activation values for d_m hidden neurons (Geva et al., 2021). We then change all the positive elements of $\max(\mathbf{x}\mathbf{W}_1^\top + \mathbf{b}_1, \mathbf{0})$ to 1 and get the one-hot activation state vector $\mathbf{s}$.

We feed an input sequence $\{P$,<s>$\}$ into the PLMs, where <s> is the special token indicating the start of a sentence. For RoBERTa, a `[MASK]` is additional prepended. This sequence is in the format of PT inputs but without specific input sentences. We use the activation states of the positions used to decode outputs, which shall be more task-specific. Specifically, for T5, we use the decoder module's activation states at the first position. For RoBERTa, we use the activation states of `[MASK]`. Finally, we concatenate the activation states of PLM's L layers to get the overall activation states:

Model	Metric	Same Task	Different Tasks
RoBERTa$_{\mathrm{LARGE}}$	E$_{\mathrm{concat}}$	9.4	6.8
	E$_{\mathrm{average}}$	41.6	37.6
	C$_{\mathrm{concat}}$	47.6	31.7
	C$_{\mathrm{average}}$	1.7	1.1
	ON	39.4	21.4
T5$_{\mathrm{XXL}}$	E$_{\mathrm{concat}}$	0.5	0.3
	E$_{\mathrm{average}}$	4.0	3.4
	C$_{\mathrm{concat}}$	29.4	3.4
	C$_{\mathrm{average}}$	4.0	2.1
	ON	62.0	46.1

Table 3: The average values (%) of the 5 similarity metrics for prompt pairs of the same task (trained with 3 different random seeds) and different tasks.

Metric	RoBERTa$_{\mathrm{LARGE}}$	T5$_{\mathrm{XXL}}$
E$_{\mathrm{concat}}$	22.6	12.9
E$_{\mathrm{average}}$	2.8	−2.5
C$_{\mathrm{concat}}$	24.8	31.6
C$_{\mathrm{average}}$	44.7	33.5
ON	**49.7**	**36.9**

Table 4: The Spearman's rank correlation scores (%) between various similarity metrics and cross-task zero-shot transfer performance of soft prompts.

$$\mathrm{AS}(P) = [\mathbf{s}_1; \mathbf{s}_2; ...; \mathbf{s}_L]. \tag{6}$$

We can only retrieve the activation states of a part of layers in the similarity computation. In experiments, we find that the higher layers tend to be more task-specific, which is consistent with the probing results (Liu et al., 2019a). Hence we use the activation states of the top 3 layers[5] in experiments below. We calculate the overlapping rate of activated neurons $\mathrm{ON}(P^{t_1}, P^{t_2})$ between the trained soft prompts of task t_1 and t_2 with the cosine similarity:

$$\mathrm{ON}(P^{t_1}, P^{t_2}) = \frac{\mathrm{AS}(P^{t_1}) \cdot \mathrm{AS}(P^{t_2})}{\|\mathrm{AS}(P^{t_1})\|\|\mathrm{AS}(P^{t_2})\|}. \tag{7}$$

6.2 Experimental Results

To evaluate the effectiveness of the above similarity metrics of soft prompts, we (i) test whether the similarity metrics can distinguish the trained prompts of the same tasks and different tasks, and (ii) examine whether these metrics align with the zero-shot transfer performance.

[5] More results about the different layers's performance are left in appendix D.4.

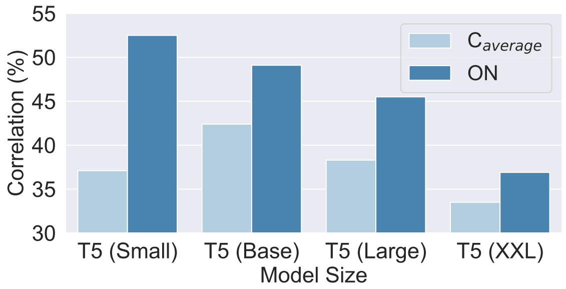


Figure 4: Spearman's correlation scores of ON and $\mathrm{C}_{\mathrm{average}}$ with cross-task zero-shot transfer performance change along with the parameter size of T5.

Projector	Task	$\mathrm{C}_{\mathrm{average}}$	ON
Task Tuning (`laptop`)	`laptop`	3.8	52.4
	Same-Type Tasks	4.1	51.0
	Different-Type Tasks	3.4	46.0
Task Tuning (`MNLI`)	`MNLI`	2.7	70.7
	Same-Type Tasks	2.7	56.7
	Different-Type Tasks	4.1	53.4

Table 5: Similarities (%) between the prompts projected with **Task Tuning** projector and the original prompts trained on $\mathrm{T5}_{\mathrm{XXL}}$.

Regarding (i), we compare the similarities of the investigated metrics for two trained prompts within the same task (trained with different random seeds) and between different tasks in Table 3. From the results, we can observe that all the metrics work well to distinguish the prompts of the same task and different tasks. This suggests that the trained soft prompts of different tasks form distinguishable clusters in the embedding space and also stimulate different abilities within the PLM.

Moreover, to evaluate (ii), how well the similarity metrics align with the cross-task transfer performance, we quantify the correlations between the similarities and zero-shot transfer performance in Figure 3. Specifically, for each target task's prompt, we rank various source tasks' prompts with similarity scores and zero-shot transfer performance and then compute the Spearman's rank correlation (Spearman, 1987) between the two ranks generated by these two ways. The overall results are shown in Table 4[6]. We can see that: (1) The *overlapping rate of activated neurons* (ON) metric works better than all the embedding similarities, which suggests that model stimulation is more important for prompt transferability than embedding distances. (2) ON works much worse on $\mathrm{T5}_{\mathrm{XXL}}$ (11B parameters) than on $\mathrm{RoBERTa}_{\mathrm{LARGE}}$ (330M parameters). We guess this is because larger PLMs have higher redundancy (Aghajanyan et al., 2021), which means prompts can activate different redundant neurons to do similar jobs and thus influence the sensitivity of ON metric. This is supported by the experiments showing that the Spearman's correlation scores of ON drop with the increase of PLM scales (Figure 4). We encourage future work to explore how to overcome the PLM redundancy for better transferrable PT. As a preliminary trial, we find that by taking the intersection of activation states of 3 prompts trained with different random seeds, ON's correlation score on $\mathrm{T5}_{\mathrm{XXL}}$ raises from 36.9% to 46.3%.

We further explore whether the prompt similarity metrics also work in the cross-model transfer setting by testing whether they work between the projected prompts and original prompts of the same task. In Table 5, we show the similarities of prompts projected with **Task Tuning** projectors by the two best metrics $\mathrm{C}_{\mathrm{average}}$ and ON. We can see: (1) ON metric shows that the projected prompts are highly similar to the original prompts within the same type of projector-training tasks but are not so similar to different-type tasks, which is quite consistent with the cross-model zero-shot transfer performance in Table 2. (2) However, $\mathrm{C}_{\mathrm{average}}$ cannot reflect this phenomena, which shows that the perspective of model stimulation is more promising for understanding transferability again.

7 Conclusion

We empirically investigate the transferability of prompts in this paper. In the cross-task setting, we find that soft prompts can transfer to similar tasks without training. In the cross-model setting, we successfully project prompts into the space of other PLMs. Further, we utilize trained prompts of other tasks or other PLMs as initialization to significantly accelerate training and improve effectiveness. Moreover, we explore various prompt transferability indicators and show that how the prompts stimulate PLMs are important to transferability. We hope the empirical analyses and the *model stimulation* idea can facilitate further research on transferable and efficient PT.

[6] The detailed results by task types are left in appendix D.2.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. 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 7319–7328, Online. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *ArXiv preprint*, abs/1607.06450.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *ArXiv preprint*, abs/2106.10199.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tianqi Chen, Ian J. Goodfellow, and Jonathon Shlens. 2016. Net2net: Accelerating learning via knowledge transfer. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *ArXiv preprint*, abs/2104.08696.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. 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 4921–4933, Online. Association for Computational Linguistics.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, et al. 2021. Pre-trained models: Past, present and future. *Proceedings of AI Open*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021. Aligning ai with shared human values. In *Proceedings of ICLR*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv preprint*, abs/1503.02531.

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, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

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.

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.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *ArXiv preprint*, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized bert pretraining approach. In *Proceedings of ICLR*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021. Reframing instructional prompts to gptk's language. *ArXiv preprint*, abs/2109.07830.

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

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

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.

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.

Yujia Qin, Yankai Lin, Jing Yi, Jiajie Zhang, Xu Han, Zhengyan Zhang, Yusheng Su, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. Knowledge inheritance for pre-trained language models. *ArXiv preprint*, abs/2105.13880.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *arXiv e-prints*.

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*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *ArXiv preprint*, abs/1706.05098.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. 2019. Natural language understanding with the quora question pairs dataset. *ArXiv preprint*, abs/1907.01041.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Charles Spearman. 1987. The proof and measurement of association between two things. In *Proceedings of The American journal of psychology*, volume 100, pages 441–471.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *ArXiv preprint*, abs/2107.02137.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better frozen model adaptation through soft prompt transfer. *ArXiv preprint*, abs/2110.07904.

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.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019a. Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4465–4476, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *ArXiv preprint*, abs/2109.01652.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *ArXiv preprint*, abs/1505.00853.

Omar F. Zaidan, Jason Eisner, and Christine Piatko. 2008. Machine learning with annotator rationales to reduce annotation cost. In *Proceedings of NeurIPS (Workshop)*.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, et al. 2021. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *ArXiv preprint*, abs/2104.12369.

Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun, Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, et al. 2021. Cpm-2: Large-scale cost-effective pre-trained language models. *ArXiv preprint*, abs/2106.10715.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

A Basic Setup for Various Tasks

A.1 Dataset and Task

Sentiment Analysis (SA) Given a sentence, a PLM will identify the opinions in this sentence. We choose IMDB (Maas et al., 2011), SST-2 (Socher et al., 2013), SemEval/laptop (Pontiki et al., 2014), SemEval/restaurant (Pontiki et al., 2014), Movie Rationales (Movie) (Zaidan et al., 2008), and TweetEval (Tweet) (Barbieri et al., 2020) to analyze.

Natural Language Inference (NLI) Given a premise and hypothesis pair, a PLM determines whether the hypothesis is entailed, contradict, or undetermined by the premise. We choose MNLI (Williams et al., 2018), QNLI (Wang et al., 2019b), and SNLI (Bowman et al., 2015) to analyze.

Ethical Judgement (EJ) Given a sentence, a PLM judges whether it is ethically acceptable. We choose Ethics/deontology (Hendrycks et al., 2021) and Ethics/justice (Hendrycks et al., 2021) to analyze.

Paraphrase Identification (PI) Given a pair of sentences, a PLM judges whether they are semantically identical. We choose QQP (Sharma et al., 2019) and MRPC (Dolan and Brockett, 2005) to analyze.

Question Answering (QA) Given a question, a PLM answer the question. We choose SQuAD (Rajpurkar et al., 2016) and NQ-Open (Lee et al., 2019) to analyze. For SQuAD, a PLM finds the answer from the content. As for NQ-Open, a PLM directly generates the answer without the content.

Summarization (SUM) Given an article, a PLM summarizes it. We choose Multi-News (Fabbri et al., 2019), and SAMSum (Gliwa et al., 2019) to analyze.

A.2 Evaluation Metrics

For SA, NLI, EJ, and PI tasks, we choose accuracy (Acc.) as their evaluation metric in the experiments. For QA and SUM tasks, we utilize F1 and ROUGE-L (Lin, 2004), respectively.

A.3 Prompt Tuning Setting

In the experiments, for all the investigated tasks, we use AdamW (Loshchilov and Hutter, 2019) as the optimizer and set the learning rate as 0.001. We set the length of soft prompts l as 100. All the soft prompts are randomly initialized and optimized with Equation 1. In the inference stage, RoBERTa predicts the label tokens at the [MASK] position and T5 directly uses its decoder to do generation.

A.4 Label Tokens

The used label tokens for the classification tasks (SA, NLI, EJ, PI) are shown in Table 6. For generation tasks (QA, SUM), the desired output is just the annotated answers.

Task	Label Tokens
Sentiment Analysis (SA)	
IMDB	positive, negative
SST-2	positive, negative
laptop	positive, moderate, negative
restaurant	positive, moderate, negative
Movie	positive, negative
Tweet	positive, moderate, negative
Natural Language Inference (NLI)	
MNLI	yes, neutral, no
QNLI	yes, no
SNLI	yes, neutral, no
Ethical Judgement (EJ)	
deontology	acceptable, un
justice	acceptable, un
Paraphrase Identification (PI)	
QQP	true, false
MRPC	true, false

Table 6: Label tokens of classification tasks.

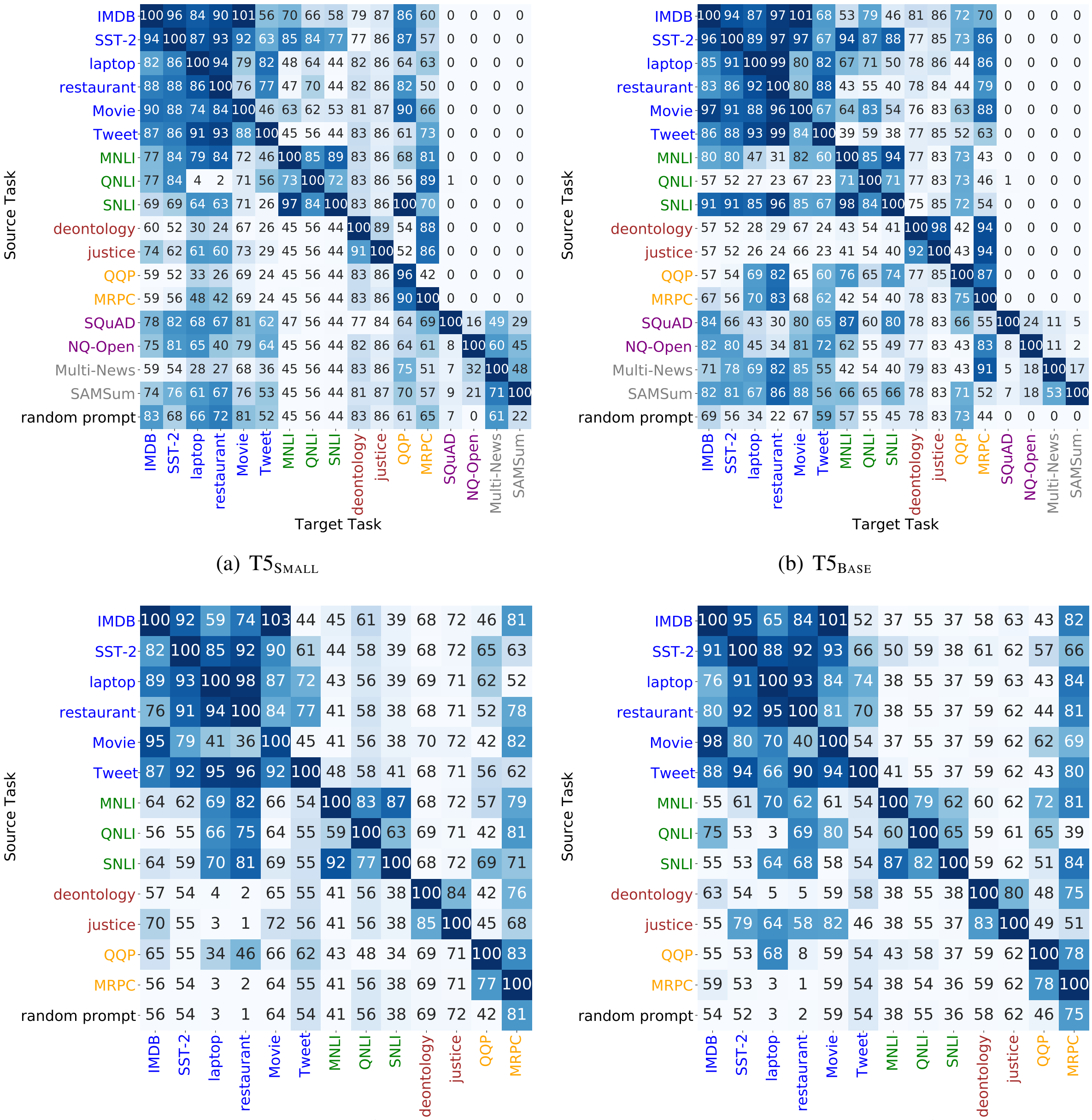
B Cross-Task Transfer

B.1 More Zero-shot transfer performance

In § 4.1, we report the zero-shot transfer performance (relative performance) on RoBERTa$_{\text{LARGE}}$ and T5$_{\text{XXL}}$. Here, we investigate the zero-shot transfer performance on other sizes of RoBERTa and T5, which are shown in Figure 5. According to these results, we can find that the transferability of soft prompts between the tasks of different types is generally poor, which is consistent with the conclusion in § 4.1.

B.2 Unifying Label Tokens

We hypothesize that the poor transferability between different task types may result from the fact that different-type tasks usually use different label tokens, e.g., yes and no are for NLI tasks while positive and negative are for SA tasks. To verify whether this factor influences the transferability, we unify the label tokens of different tasks into the same set of numbers ($1, 2, \ldots$) and choose



	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC	SQuAD	NQ-Open	Multi-News	SAMSum
IMDB	100	96	84	90	101	56	70	66	58	79	87	86	60	0	0	0	0
SST-2	94	100	87	93	92	63	85	84	77	77	86	87	57	0	0	0	0
laptop	82	86	100	94	79	82	48	64	44	82	86	64	63	0	0	0	0
restaurant	88	88	86	100	76	77	47	70	44	82	86	82	50	0	0	0	0
Movie	90	88	74	84	100	46	63	62	53	81	87	90	66	0	0	0	0
Tweet	87	86	91	93	88	100	45	56	44	83	86	61	73	0	0	0	0
MNLI	77	84	79	84	72	46	100	85	89	83	86	68	81	0	0	0	0
QNLI	77	84	4	2	71	56	73	100	72	83	86	56	89	1	0	0	0
SNLI	69	69	64	63	71	26	97	84	100	83	86	100	70	0	0	0	0
deontology	60	52	30	24	67	26	45	56	44	100	89	54	88	0	0	0	0
justice	74	62	61	60	73	29	45	56	44	91	100	52	86	0	0	0	0
QQP	59	52	33	26	69	24	45	56	44	83	86	96	42	0	0	0	0
MRPC	59	56	48	42	69	24	45	56	44	83	86	90	100	0	0	0	0
SQuAD	78	82	68	67	81	62	47	56	44	77	84	64	69	100	16	49	29
NQ-Open	75	81	65	40	79	64	45	56	44	82	86	64	61	8	100	60	45
Multi-News	59	54	28	27	68	36	45	56	44	83	86	75	51	7	32	100	48
SAMSum	74	76	61	67	76	53	45	56	44	81	87	70	57	9	21	71	100
random prompt	83	68	66	72	81	52	45	56	44	83	86	61	65	7	0	61	22

(a) T5$_{\text{SMALL}}$

	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC	SQuAD	NQ-Open	Multi-News	SAMSum
IMDB	100	94	87	97	101	68	53	79	46	81	86	72	70	0	0	0	0
SST-2	96	100	89	97	97	67	94	87	88	77	85	73	86	0	0	0	0
laptop	85	91	100	99	80	82	67	71	50	78	86	44	86	0	0	0	0
restaurant	83	86	92	100	80	88	43	55	40	78	84	44	79	0	0	0	0
Movie	97	91	88	96	100	67	64	83	54	76	83	63	88	0	0	0	0
Tweet	86	88	93	99	84	100	39	59	38	77	85	52	63	0	0	0	0
MNLI	80	80	47	31	82	60	100	85	94	77	83	73	43	0	0	0	0
QNLI	57	52	27	23	67	23	71	100	71	77	83	73	46	1	0	0	0
SNLI	91	91	85	96	85	67	98	84	100	75	85	72	54	0	0	0	0
deontology	57	52	28	29	67	24	43	54	41	100	98	42	94	0	0	0	0
justice	57	52	26	24	66	23	41	54	40	92	100	43	94	0	0	0	0
QQP	57	54	69	82	65	60	76	65	74	77	85	100	87	0	0	0	0
MRPC	67	56	70	83	68	62	42	54	40	78	83	75	100	0	0	0	0
SQuAD	84	66	43	30	80	65	87	60	80	78	83	66	55	100	24	11	5
NQ-Open	82	80	45	34	81	72	62	55	49	77	83	43	83	8	100	11	2
Multi-News	71	78	69	82	85	55	42	54	40	79	83	43	91	5	18	100	17
SAMSum	82	81	67	86	88	56	66	65	66	78	83	71	52	7	18	53	100
random prompt	69	56	34	22	67	59	57	55	45	78	83	73	44	0	0	0	0

(b) T5$_{\text{BASE}}$

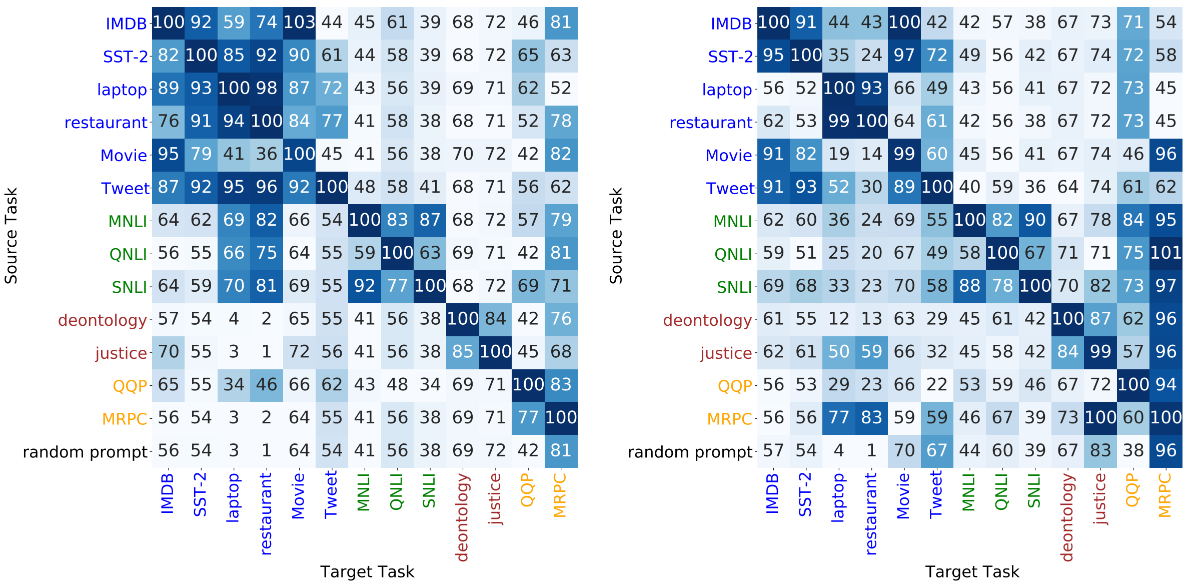
	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC
IMDB	100	92	59	74	103	44	45	61	39	68	72	46	81
SST-2	82	100	85	92	90	61	44	58	39	68	72	65	63
laptop	89	93	100	98	87	72	43	56	39	69	71	62	52
restaurant	76	91	94	100	84	77	41	58	38	68	71	52	78
Movie	95	79	41	36	100	45	41	56	38	70	72	42	82
Tweet	87	92	95	96	92	100	48	58	41	68	71	56	62
MNLI	64	62	69	82	66	54	100	83	87	68	72	57	79
QNLI	56	55	66	75	64	55	59	100	63	69	71	42	81
SNLI	64	59	70	81	69	55	92	77	100	68	72	69	71
deontology	57	54	4	2	65	55	41	56	38	100	84	42	76
justice	70	55	3	1	72	56	41	56	38	85	100	45	68
QQP	65	55	34	46	66	62	43	48	34	69	71	100	83
MRPC	56	54	3	2	64	55	41	56	38	69	71	77	100
random prompt	56	54	3	1	64	54	41	56	38	69	72	42	81

(c) RoBERTa$_{\text{BASE}}$

	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC
IMDB	100	95	65	84	101	52	37	55	37	58	63	43	82
SST-2	91	100	88	92	93	66	50	59	38	61	62	57	66
laptop	76	91	100	93	84	74	38	55	37	59	63	43	84
restaurant	80	92	95	100	81	70	38	55	37	59	62	44	81
Movie	98	80	70	40	100	54	37	55	37	59	62	62	69
Tweet	88	94	66	90	94	100	41	55	37	59	62	43	80
MNLI	55	61	70	62	61	54	100	79	62	60	62	72	81
QNLI	75	53	3	69	80	54	60	100	65	59	61	65	39
SNLI	55	53	64	68	58	54	87	82	100	59	62	51	84
deontology	63	54	5	5	59	58	38	55	38	100	80	48	75
justice	55	79	64	58	82	46	38	55	37	83	100	49	51
QQP	55	53	68	8	59	54	43	58	37	59	62	100	78
MRPC	59	53	3	1	59	54	38	54	36	59	62	78	100
random prompt	54	52	3	2	59	54	38	55	36	58	62	46	75

(d) RoBERTa$_{\text{LARGE}}$

Figure 5: Relative performance (transferring zero-shot performance / original PT performance) (%) on the target tasks (columns) of the soft prompts trained on the source tasks (rows), both of which demonstrate the relative performance for zero-shot transfer of prompts of RoBERTa and T5. Colors of the tasks names indicate the task types. **Blue**: sentiment analysis (SA). Green: natural language inference (NLI). Brown: ethical judgement (EJ). Orange: paraphrase identification (PI). Purple: question answering (QA). Gray: summarization (SUM). *Random Prompt* of the last row means the soft prompts are randomly generated without any training.



(a) Directly transferring (RoBERTa$_{\text{BASE}}$)

(b) Unifying the label tokens (RoBERTa$_{\text{BASE}}$)

Figure 6: To exclude The poor transferability, which may result from the fact that different-type tasks use different label tokens, we unify the label tokens of different tasks into the same set of numbers (`1`, `2`, . . .) and choose RoBERTa$_{\text{BASE}}$ for the experiments. From the Figure (a) and (b), we observe that the transferability between different-type tasks are still generally not improved in this way. This indicates that different-type tasks surely require distinct abilities.

RoBERTa$_{\text{BASE}}$ for the experiments. In Figure 6, we can observe that the transferability between different-type tasks are generally not improved in this way. This indicates that different-type tasks surely require distinct abilities, which prohibits reusing prompts between them.

B.3 Speedup Calculation

In this paper, we compute convergence speedup and comparable-result speedup as follows:

$$\text{Convergence Speedup(x)} = \frac{\text{PT convergence time}}{\text{TPT convergence time}},$$

$$\text{Comparable-result Speedup(x)} = \frac{\text{PT convergence time}}{\text{time of TPT achieving comparable result to PT}}. \tag{8}$$

We calculate the training loss and the evaluation score per 100 steps during the training. When the training loss stops dropping and the evaluation score stops increasing for 300 steps, we set the point as the convergence point. For the convergence speedup in Equation 8, the PT convergence time is divided by the TPT convergence time. As for the comparable-result speedup in Equation 8, the PT convergence time are divided by the time of TPT achieving comparable performance to PT.

C Cross-Model Transfer

C.1 Implementation Details of Projector

As mentioned in § 5.1, we give the prompt of the source PLM, $P^s = \{\mathbf{p}_1^s, \ldots, \mathbf{p}_l^s\}$, and concatenate its l virtual tokens into a unified vector $\mathbf{P}^s \in \mathbb{R}^{ld_s}$, where d_s is the hidden size of the source PLM. To transfer $\mathbf{P}^s$ to the target PLM whose hidden size is d_t, we design a projection function $\mathbf{Proj}(\cdot)$ parameterized by a two-layer perceptron as follows:

$$\tilde{\mathbf{P}}^s = \mathbf{Proj}(\mathbf{P}^s) = \mathbf{W}_2(\sigma(\mathbf{P}^s\mathbf{W}_1 + \mathbf{b}_1)) + \mathbf{b}_2, \quad (9)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_h \times ld_s}, \mathbf{W}_2 \in \mathbb{R}^{ld_t \times d_h}$ are trainable matrices, $\mathbf{b}_1 \in \mathbb{R}^{d_h}, \mathbf{b}_2 \in \mathbb{R}^{ld_t}$ are biases, σ is a non-linear activation function. We set the inner hidden size d_h to 768. In this paper, we investigate cross-model transfer among various PLMs including BERT$_{\text{BASE}}$, RoBERTa$_{\text{BASE}}$, RoBERTa$_{\text{LARGE}}$, T5$_{\text{SMALL}}$, T5$_{\text{BASE}}$, and T5$_{\text{XXL}}$, whose hidden sizes are 768, 768, 1024, 512, 768, and 1024, respectively. Besides, for non-linear activation functions, we have tried `tanh` and `LeakyReLU` (Xu et al., 2015), and find their performance on various PLMs are similar. The reported results are based on the `LeakyReLU` activation.

Method		SA						NLI			EJ		PI	
		IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC
From BERT$_{\text{BASE}}$ to RoBERTa$_{\text{BASE}}$														
PT on RoBERTa$_{\text{BASE}}$		89.9	93.8	77.3	80.7	79.2	74.5	80.6	90.5	88.5	72.9	70.0	86.9	83.9
Random Prompt		50.6	50.8	2.3	1.2	50.5	40.5	32.8	50.5	33.3	50.4	50.2	36.8	68.0
IMDB, laptop	Distance Minimizing	**89.7**	53.1	**75.6**	18.3	54.2	24.0	31.2	50.0	33.3	50.6	50.0	36.8	67.2
	Task Tuning	**88.2**	**82.2**	**76.3**	**77.9**	**73.4**	43.6	32.0	47.9	32.8	49.8	49.4	50.2	47.7
MNLI	Distance Minimizing	55.6	51.0	2.5	1.4	53.1	41.1	**80.0**	50.6	33.3	50.6	50.0	48.3	68.0
	Task Tuning	50.9	52.0	11.9	13.1	45.8	18.2	**80.0**	**74.9**	**80.0**	50.4	49.9	36.8	68.1
From RoBERTa$_{\text{BASE}}$ to RoBERTa$_{\text{LARGE}}$														
PT on RoBERTa$_{\text{LARGE}}$		91.8	96.0	78.1	81.7	81.7	76.6	88.5	93.4	90.7	85.6	81.1	89.0	82.7
Random Prompt		50.1	50.2	2.0	2.0	49.5	40.5	32.7	51.0	33.3	50.3	49.9	40.6	61.2
IMDB,laptop	Distance Minimizing	**92.1**	50.1	**77.0**	1.4	51.0	37.6	33.1	50.2	32.8	50.4	50.0	62.3	38.3
	Task Tuning	**90.4**	**76.2**	**64.2**	**69.5**	**79.7**	45.0	33.3	50.5	33.1	50.3	50.0	38.5	79.7
MNLI	Distance Minimizing	50.3	51.2	5.2	5.9	51.0	40.6	**88.5**	49.1	33.2	50.3	50.0	45.1	66.4
	Task Tuning	67.7	76.1	28.9	43.7	60.4	49.1	**87.1**	**79.4**	**84.5**	49.7	50.0	36.8	68.5
From T5$_{\text{BASE}}$ to T5$_{\text{XXL}}$														
PT on T5$_{\text{XXL}}$		96.5	97.4	76.6	88.1	97.9	72.5	90.5	95.2	93.4	87.0	92.5	90.0	86.3
Random Prompt		49.7	49.0	19.8	17.0	51.6	15.5	31.8	49.3	31.9	51.3	50.0	36.4	67.0
laptop	Distance Minimizing	49.0	49.7	**76.6**	17.0	52.3	16.3	31.8	48.7	33.3	54.1	49.0	36.7	67.7
	Task Tuning	**77.2**	**86.2**	**80.3**	**83.5**	**64.6**	**55.2**	31.9	49.9	32.9	48.7	52.8	50.7	53.1
MNLI	Distance Minimizing	49.7	49.0	19.8	17.1	51.6	15.5	**90.5**	49.3	34.8	52.3	50.0	36.8	67.7
	Task Tuning	54.9	70.0	60.8	74.1	3.6	41.4	**89.7**	**84.8**	**90.8**	49.7	50.0	37.2	66.4

Table 7: We conduct experiments between various PLMs in different scales and heterogeneous frameworks: from BERT$_{\text{BASE}}$ to RoBERTa$_{\text{BASE}}$, from RoBERTa$_{\text{BASE}}$ to RoBERTa$_{\text{LARGE}}$, and from T5$_{\text{BASE}}$ to T5$_{\text{XXL}}$. Besides, we color the non-trivial zero-shot performance (%) of the cross-model setting with **bold**.

C.2 More Zero-shot Transfer Performance

In § 5.2, we have introduced the zero-shot transfer performance of various projector-learning methods in the setting of transferring from RoBERTa$_{\text{LARGE}}$ to T5$_{\text{XXL}}$. We explore more cross-model transfer settings here, which are transferring between various PLMs in different scales and heterogeneous frameworks, including from BERT$_{\text{BASE}}$ to RoBERTa$_{\text{BASE}}$, from RoBERTa$_{\text{BASE}}$ to RoBERTa$_{\text{LARGE}}$, and from T5$_{\text{BASE}}$ to T5$_{\text{XXL}}$.

Table 7 shows the experimental results. We can see the phenomena and conclusions are all consistent with § 5.2.

C.3 Technical Details of TPT$_{\text{MODEL}}$ (Transfer with Initialization)

In § 5.3, we demonstrate cross-model transferrable prompt tuning (TPT$_{\text{MODEL}}$) can well improve performance and reduce training time.

However, when we apply TPT$_{\text{MODEL}}$ to more PLMs, we find that the projected prompts may have quite different L_2 norm values with the original prompts, especially for the small-scale PLMs (e.g., from BERT$_{\text{BASE}}$ to RoBERTa$_{\text{BASE}}$). Specifically, we obtain the projected prompts with the trained Task Tuning projector, and find that the projected prompts are hard to optimize in some tasks as shown in Figure 7 `[Without LayerNorm]`. Thus, we attempt to add the layer normalization operation (Ba et al., 2016) `LayerNorm` into the projectors to regularize the norm of the projected prompt as follows:

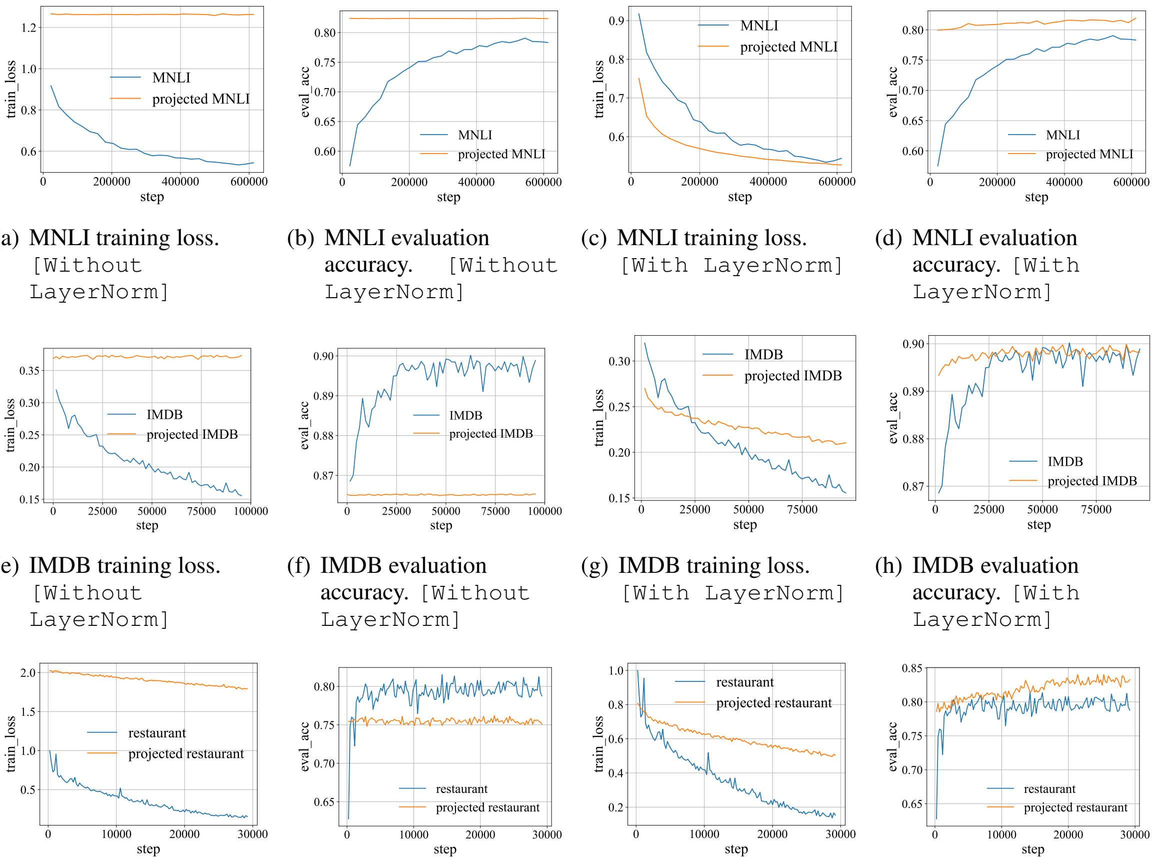
$$\tilde{\mathbf{P}}^s = \texttt{LayerNorm}(\mathbf{Proj}(\mathbf{P}^s)). \quad (10)$$

By the `LayerNorm`, the projected prompts can work well on TPT$_{\text{MODEL}}$ and achieve better performance and speedup as shown in Figure 7 `[With LayerNorm]`. Interestingly, although prompts projected by the projectors `[Without LayerNorm]` are hard to be trained in TPT$_{\text{MODEL}}$, they can achieve similar zero-shot transfer performance with the prompts projected by the projectors `[With LayerNorm]` in Table 8.

D Transferability Indicator

D.1 Effectiveness of Similarity Metrics

We categorize all prompts into three groups: same tasks (prompts trained with different seeds on the same dataset), same-type tasks, and different-type tasks. Table 9 shows that all the similarity metrics successfully distinguish task types.



(a) MNLI training loss. [Without LayerNorm]

(b) MNLI evaluation accuracy. [Without LayerNorm]

(c) MNLI training loss. [With LayerNorm]

(d) MNLI evaluation accuracy. [With LayerNorm]

(e) IMDB training loss. [Without LayerNorm]

(f) IMDB evaluation accuracy. [Without LayerNorm]

(g) IMDB training loss. [With LayerNorm]

(h) IMDB evaluation accuracy. [With LayerNorm]

(i) restaurant training loss. [Without LayerNorm]

(j) restaurant evaluation accuracy. [Without LayerNorm]

(k) restaurant training loss. [With LayerNorm]

(l) restaurant evaluation accuracy. [With LayerNorm]

Figure 7: Transfer prompts of BERT$_{\text{BASE}}$ to RoBERTa$_{\text{BASE}}$. The (—) represents vanilla PT, and (—) is TPT$_{\text{MODEL}}$ that utilizes projected prompts as initizations to conduct PT. The projected prompts respectively come from two different Task Tuning projectors ([Without LayerNorm] and [With LayerNorm]).

Method	SA						NLI			EJ		PI	
	IMDB	SST-2	laptop	restaurant	Movie	Tweet	MNLI	QNLI	SNLI	deontology	justice	QQP	MRPC
PT on RoBERTa$_{\text{BASE}}$	89.9	93.8	77.3	80.7	79.2	74.5	80.6	90.5	88.5	72.9	70.0	86.9	83.9
[Without LayerNorm]													
Task Tuning (IMDB, laptop)	**86.5**	**84.9**	**73.4**	**75.3**	**76.6**	47.7	31.8	52.0	32.9	50.3	50.0	37.6	67.5
Task Tuning (MNLI)	66.6	70.4	53.0	43.8	57.8	47.9	**82.4**	**74.9**	**78.1**	50.4	49.9	45.3	70.1
[With LayerNorm]													
Task Tuning (IMDB, laptop)	**88.2**	**82.2**	**76.3**	**77.9**	**73.4**	43.6	32.0	47.9	32.8	49.8	49.4	50.2	47.7
Task Tuning (MNLI)	50.9	52.0	11.9	13.1	45.8	18.2	**80.0**	**74.9**	**80.0**	50.4	49.9	36.8	68.1

Table 8: We compare the zero-shot performances of prompts projected by Task Tuning projectors ([With LayerNorm] and [Without LayerNorm]) and find that their accuracies are close. **Bold** represents non-trivial performance.

D.2 Correlation Between Prompt Transferability and Prompt Similarity

In § 6, we provide the overall averaged Spearman's rank correlation scores (%) between various similarity metrics and zero-shot transfer performance of soft prompts for RoBERTa$_{\text{LARGE}}$ and T5$_{\text{XXL}}$.

Here, we further show Spearman's rank correlation scores grouped by the task types on more PLMs. The results are shown in Table 10 and Table 11.

Metric	Same Tasks	Same-type Tasks	Different-type Tasks
RoBERTa$_{\text{LARGE}}$			
E_{concat}	9.4	9.4	6.8
$\text{E}_{\text{average}}$	41.6	41.4	37.6
C_{concat}	47.6	45.3	31.7
$\text{C}_{\text{average}}$	1.7	1.3	1.1
ON (Bottom 3)	42.8	43.3	39.1
ON (Top 3)	39.4	28.2	21.4
ON (All 24)	40.0	35.8	29.6
T5$_{\text{XXL}}$ (Decoder Module)			
E_{concat}	0.5	0.5	0.3
$\text{E}_{\text{average}}$	4.0	5.1	3.4
C_{concat}	29.4	2.8	2.4
$\text{C}_{\text{average}}$	4.0	2.6	2.1
ON (Bottom 3)	80.3	75.4	76.3
ON (Top 3)	62.0	52.7	46.1
ON (All 24)	60.8	54.0	49.2

Table 9: The average values (%) of the 5 similarity metrics for prompt pairs within the same task (trained with 3 different random seeds) and between different tasks (of the same type and different types) on RoBERTa$_{\text{LARGE}}$ and T5$_{\text{XXL}}$.

Metric	SA	NLI	EJ	PI	QA	SUM	All
T5$_{\text{SMALL}}$ (Decoder Module)							
E_{concat}	10.1	19.6	31.3	5.3	27.3	38.0	21.9
$\text{E}_{\text{average}}$	-6.8	-28.0	18.7	-2.6	29.1	42.9	8.9
C_{concat}	34.6	63.6	26.6	**19.3**	-2.1	12.5	25.7
$\text{C}_{\text{average}}$	**64.3**	65.1	30.7	15.7	27.7	19.2	37.1
ON (Bottom 3)	32.9	72.6	41.8	14.2	45.5	52.8	43.3
ON (Top 3)	50.6	74.8	**51.4**	2.6	**60.3**	**78.8**	**52.5**
ON (All 24)	44.8	**79.7**	44.5	6.3	59.7	67.9	50.5
T5$_{\text{BASE}}$ (Decoder Module)							
E_{concat}	55.2	-17.0	10.2	21.5	5.9	-1.1	20.8
$\text{E}_{\text{average}}$	53.4	-42.3	-10.7	7.5	-27.7	-10.8	9.0
C_{concat}	**57.2**	25.2	35.1	37.0	30.2	-20.5	28.4
$\text{C}_{\text{average}}$	47.6	**70.0**	30.4	**48.0**	34.9	16.8	42.4
ON (Bottom 3)	34.7	29.8	40.8	16.9	24.2	72.2	36.0
ON (Top 3)	53.8	24.3	**50.6**	46.1	54.7	**79.1**	**49.1**
ON (All 24)	46.1	25.0	42.6	39.7	**56.7**	72.3	43.4
T5$_{\text{XXL}}$ (Decoder Module)							
E_{concat}	40.8	-13.4	19.3	11.4	-4.3	-19.5	12.9
$\text{E}_{\text{average}}$	32.2	-42.6	9.7	-2.0	-27.7	-34.0	-2.5
C_{concat}	21.4	40.9	**42.6**	24.6	30.2	45.6	31.6
$\text{C}_{\text{average}}$	23.3	**44.8**	33.3	29.3	34.9	**49.9**	33.5
ON (Bottom 3)	9.1	20.7	14.8	18.3	24.2	-9.9	12.4
ON (Top 3)	**42.7**	33.6	39.1	30.3	54.7	11.1	**36.9**
ON (All 24)	31.0	23.6	37.7	**34.2**	**56.7**	15.4	32.0
$\text{ON}_{\mathbf{I}}$ (Bottom 3)	- -	- -	- -	- -	- -	- -	**25.3**
$\text{ON}_{\mathbf{I}}$ (Top 3)	- -	- -	- -	- -	- -	- -	**46.3**
$\text{ON}_{\mathbf{I}}$ (All 24)	- -	- -	- -	- -	- -	- -	**40.0**

Table 10: Spearman's rank correlation scores (%) between various similarity metrics and zero-shot transfer performance of soft prompts for various scales of T5 and $\text{ON}_{\mathbf{I}}$ as introduced in appendix D.3.

Metric	SA	NLI	EJ	PI	All
RoBERTa$_{\text{BASE}}$					
E_{concat}	31.1	-5.9	30.5	16.2	20.2
$\text{E}_{\text{average}}$	17.2	-52.4	12.1	-13.5	-4.4
C_{concat}	51.6	8.8	38.5	29.7	36.3
$\text{C}_{\text{average}}$	65.8	55.9	26.1	28.9	51.7
ON (Bottom 3)	56.2	64.3	17.9	21.2	46.8
ON (Top 3)	**77.9**	**74.2**	**43.4**	**32.7**	**64.8**
ON (All 24)	71.2	70.5	33.6	25.0	58.1
RoBERTa$_{\text{LARGE}}$					
E_{concat}	42.5	-16.3	21.4	22.8	22.6
$\text{E}_{\text{average}}$	34.5	-55.1	-5.8	3.6	2.8
C_{concat}	44.5	-11.7	23.6	22.0	24.8
$\text{C}_{\text{average}}$	38.2	77.1	12.4	**47.8**	44.7
ON (Bottom 3)	32.0	34.8	**44.5**	30.3	34.3
ON (Top 3)	**70.9**	**45.6**	13.5	28.9	**49.7**
ON (All 24)	62.7	40.6	16.0	31.1	45.6

Table 11: Spearman's rank correlation scores (%) between various similarity metrics and zero-shot transfer performance of soft prompts for various scales of RoBERTa.

D.3 PLMs' Redundancy Influence Indicators

From Table 10, we find that the correlation between prompt transferability and prompt similarity will drop with the increase of PLM size.

We guess that this phenomena may result from PLMs' high redundancy (Aghajanyan et al., 2021). To try to overcome this, we simultaneously utilize the prompts trained with three random seeds on the same dataset and take their intersection of activation states as the activated neurons into the similarity (ON) computation. This similarity is called $\text{ON}_{\mathbf{I}}$. By using it, the correlation score of ON can significantly raise as shown in Table 10.

D.4 Overlapping Rate of Activated Neurons in Different Layers

To further understand model stimulation in PLMs, we investigate ON in different layers of PLMs. Specifically, on RoBERTa$_{\text{BASE}}$, we measure the similarity between different prompts with activation states of from 1 to 3 layers (Figure 8), from 4 to 6 layers (Figure 9), from 7 to 9 layers (Figure 10), from 10 to 12 layers (Figure 11), and all 12 layers (Figure 12), respectively.

We find that the activated neurons are common in the bottom layers but tend to be more task-specific in top layers, which is consistent with the findings of previous works (Liu et al., 2019a).

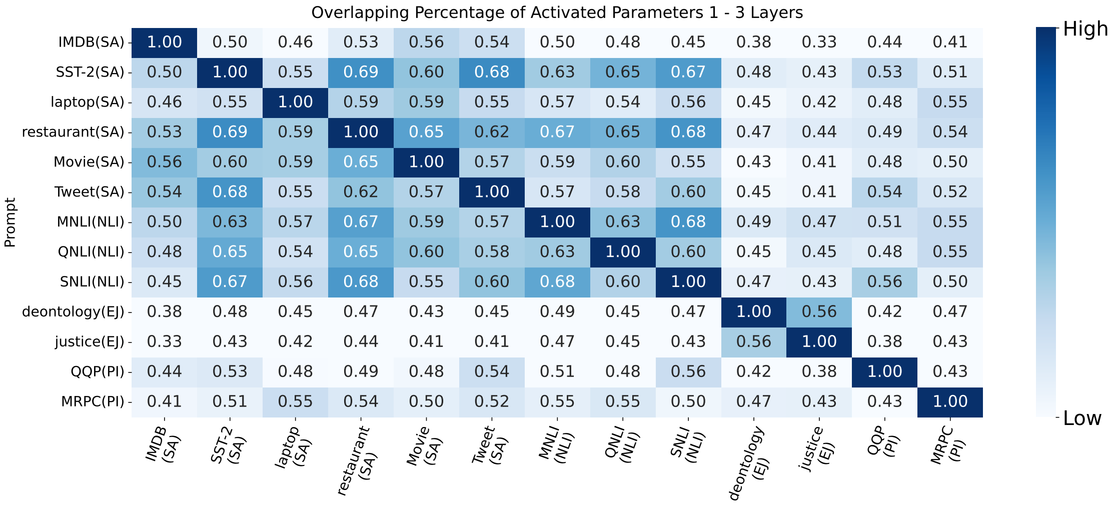


Figure 8: ON in 1 - 3 layers of RoBERTa$_{\text{BASE}}$.

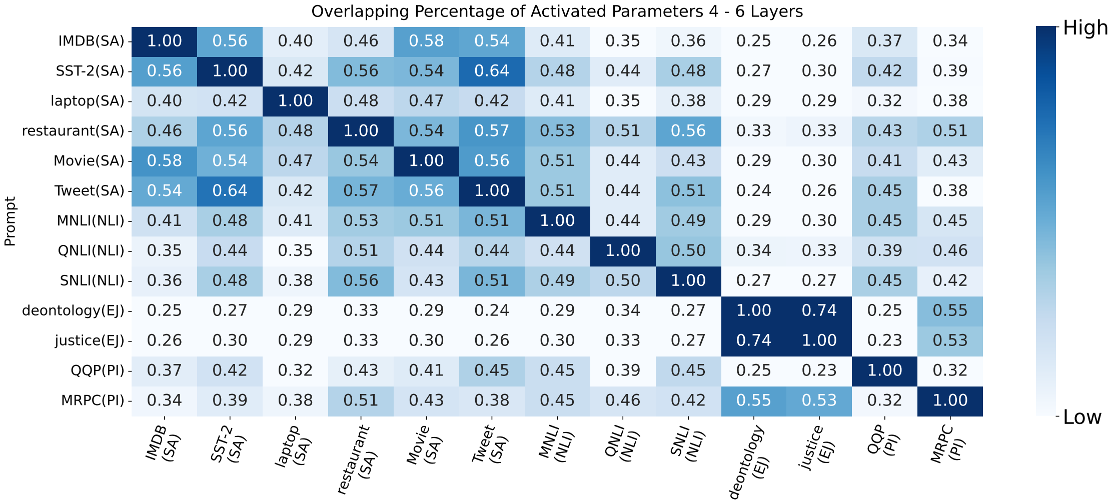


Figure 9: ON in 4 - 6 layers of RoBERTa$_{\text{BASE}}$.

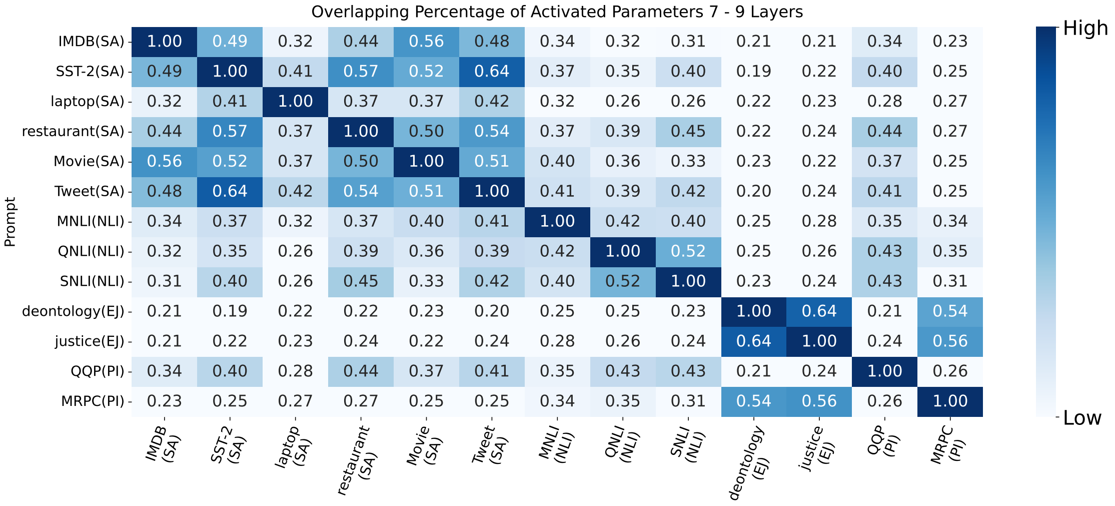
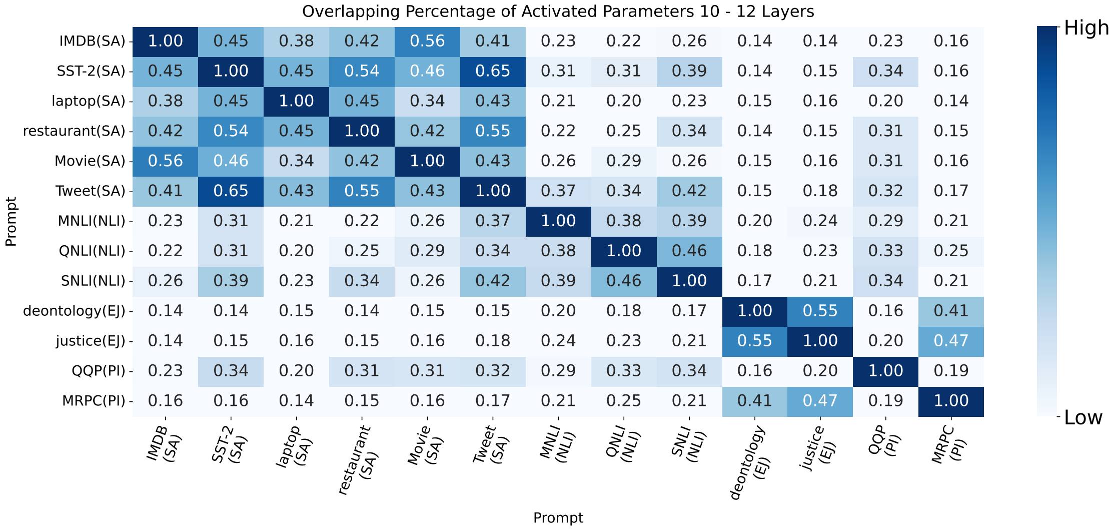
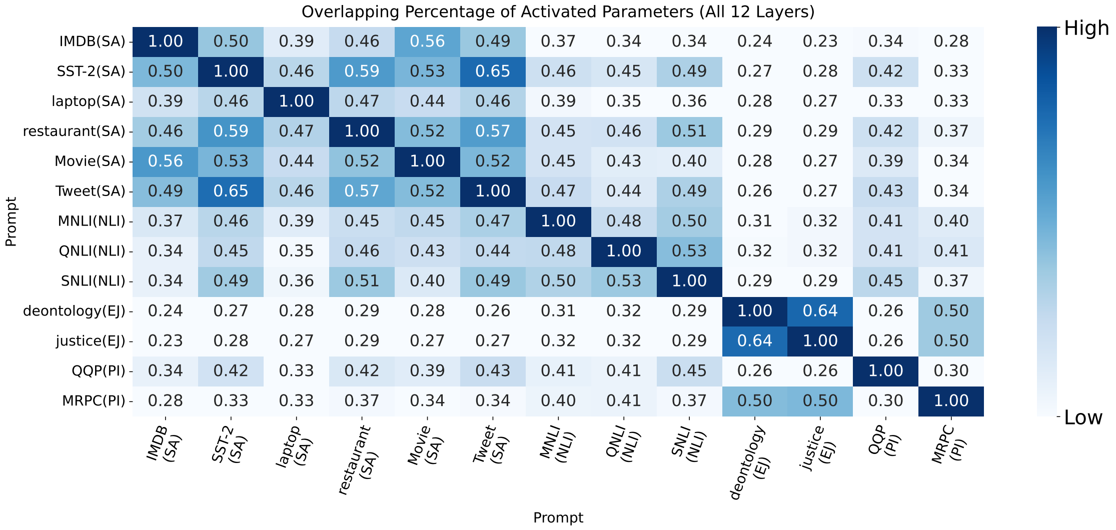


Figure 10: ON in 7 - 9 layers of RoBERTa$_{\text{BASE}}$.



	IMDB (SA)	SST-2 (SA)	laptop (SA)	restaurant (SA)	Movie (SA)	Tweet (SA)	MNLI (NLI)	QNLI (NLI)	SNLI (NLI)	deontology (EJ)	justice (EJ)	QQP (PI)	MRPC (PI)
IMDB(SA)	1.00	0.45	0.38	0.42	0.56	0.41	0.23	0.22	0.26	0.14	0.14	0.23	0.16
SST-2(SA)	0.45	1.00	0.45	0.54	0.46	0.65	0.31	0.31	0.39	0.14	0.15	0.34	0.16
laptop(SA)	0.38	0.45	1.00	0.45	0.34	0.43	0.21	0.20	0.23	0.15	0.16	0.20	0.14
restaurant(SA)	0.42	0.54	0.45	1.00	0.42	0.55	0.22	0.25	0.34	0.14	0.15	0.31	0.15
Movie(SA)	0.56	0.46	0.34	0.42	1.00	0.43	0.26	0.29	0.26	0.15	0.16	0.31	0.16
Tweet(SA)	0.41	0.65	0.43	0.55	0.43	1.00	0.37	0.34	0.42	0.15	0.18	0.32	0.17
MNLI(NLI)	0.23	0.31	0.21	0.22	0.26	0.37	1.00	0.38	0.39	0.20	0.24	0.29	0.21
QNLI(NLI)	0.22	0.31	0.20	0.25	0.29	0.34	0.38	1.00	0.46	0.18	0.23	0.33	0.25
SNLI(NLI)	0.26	0.39	0.23	0.34	0.26	0.42	0.39	0.46	1.00	0.17	0.21	0.34	0.21
deontology(EJ)	0.14	0.14	0.15	0.14	0.15	0.15	0.20	0.18	0.17	1.00	0.55	0.16	0.41
justice(EJ)	0.14	0.15	0.16	0.15	0.16	0.18	0.24	0.23	0.21	0.55	1.00	0.20	0.47
QQP(PI)	0.23	0.34	0.20	0.31	0.31	0.32	0.29	0.33	0.34	0.16	0.20	1.00	0.19
MRPC(PI)	0.16	0.16	0.14	0.15	0.16	0.17	0.21	0.25	0.21	0.41	0.47	0.19	1.00

Figure 11: ON in 10 - 12 layers of RoBERTa$_{\text{BASE}}$.



	IMDB (SA)	SST-2 (SA)	laptop (SA)	restaurant (SA)	Movie (SA)	Tweet (SA)	MNLI (NLI)	QNLI (NLI)	SNLI (NLI)	deontology (EJ)	justice (EJ)	QQP (PI)	MRPC (PI)
IMDB(SA)	1.00	0.50	0.39	0.46	0.56	0.49	0.37	0.34	0.34	0.24	0.23	0.34	0.28
SST-2(SA)	0.50	1.00	0.46	0.59	0.53	0.65	0.46	0.45	0.49	0.27	0.28	0.42	0.33
laptop(SA)	0.39	0.46	1.00	0.47	0.44	0.46	0.39	0.35	0.36	0.28	0.27	0.33	0.33
restaurant(SA)	0.46	0.59	0.47	1.00	0.52	0.57	0.45	0.46	0.51	0.29	0.29	0.42	0.37
Movie(SA)	0.56	0.53	0.44	0.52	1.00	0.52	0.45	0.43	0.40	0.28	0.27	0.39	0.34
Tweet(SA)	0.49	0.65	0.46	0.57	0.52	1.00	0.47	0.44	0.49	0.26	0.27	0.43	0.34
MNLI(NLI)	0.37	0.46	0.39	0.45	0.45	0.47	1.00	0.48	0.50	0.31	0.32	0.41	0.40
QNLI(NLI)	0.34	0.45	0.35	0.46	0.43	0.44	0.48	1.00	0.53	0.32	0.32	0.41	0.41
SNLI(NLI)	0.34	0.49	0.36	0.51	0.40	0.49	0.50	0.53	1.00	0.29	0.29	0.45	0.37
deontology(EJ)	0.24	0.27	0.28	0.29	0.28	0.26	0.31	0.32	0.29	1.00	0.64	0.26	0.50
justice(EJ)	0.23	0.28	0.27	0.29	0.27	0.27	0.32	0.32	0.29	0.64	1.00	0.26	0.50
QQP(PI)	0.34	0.42	0.33	0.42	0.39	0.43	0.41	0.41	0.45	0.26	0.26	1.00	0.30
MRPC(PI)	0.28	0.33	0.33	0.37	0.34	0.34	0.40	0.41	0.37	0.50	0.50	0.30	1.00

Figure 12: ON in all 12 layers of RoBERTa$_{\text{BASE}}$.