Connectivity Patterns are Task Embeddings

Zhiheng Xi[1*], **Rui Zheng**[1*], **Yuansen Zhang**[1], **Xuanjing Huang**[1],
Zhongyu Wei[2], **Minlong Peng**[3], **Mingming Sun**[3], **Qi Zhang**[1†], **Tao Gui**[4†]
[1] School of Computer Science, Fudan University, Shanghai, China
[2] School of Data Science, Fudan University, Shanghai, China
[3] Cognitive Computing Lab Baidu Research, Beijing, China
[4] Institute of Modern Languages and Linguistics, Fudan University, Shanghai, China
{zhxi22,zhangys22}@m.fudan.edu.cn , {pengminlong, sunmingming01}@baidu.com ,
{rzheng20,xjhuang,zywei,qz,tgui}@fudan.edu.cn

Abstract

Task embeddings are task-specific vectors designed to construct a semantic space of tasks, which can be used to predict the most transferable source task for a given target task via the similarity between task embeddings. However, existing methods use optimized parameters and representations as task embeddings, resulting in substantial computational complexity and storage requirements. In this work, we draw inspiration from the operating mechanism of deep neural networks (DNNs) and biological brains, where neuronal activations are sparse and task-specific, and we use the connectivity patterns of neurons as a unique identifier associated with the task. The proposed method learns to assign importance masks for sub-structures of DNNs, and accordingly indicate the task-specific connectivity patterns. In addition to the storage advantages brought by the binary masking mechanism and structured sparsity, the early-bird nature of the sparse optimization process can deliver an efficient computation advantage. Experiments show that our method consistently outperforms other baselines in predicting inter-task transferability across data regimes and transfer settings, while keeping high efficiency in computation and storage.

1 Introduction

With the rapid development and excellent performance of large pre-trained language models (PLMs), the most prevalent paradigm in natural language processing (NLP) has become *pre-training then fine-tuning* (Peters et al., 2018; Devlin et al., 2019a; Brown et al., 2020; Lewis et al., 2020; Raffel et al., 2020). Extending upon the two-step training procedure, previous works show that *intermediate-task transfer*, i.e., fine-tuning the model on an intermediate source task before the target task, can yield further gains (Phang et al., 2018; Wang et al., 2019a). Nevertheless, the improvement by *intermediate-task transfer* heavily relies on the selection of a proper intermediate task because some source tasks lead to performance degradation (Yogatama et al., 2019; Pruksachatkun et al., 2020). One straightforward approach is to enumerate every possible (source, target) task combination, but it is extremely expensive. Therefore, recent works explore methods to predict inter-task transferability accurately with high efficiency.

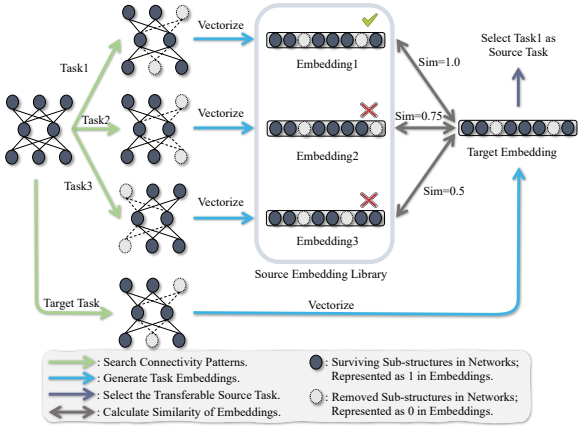


Figure 1: An overview of CoPaTE, including the procedures for searching connectivity patterns, generating task embeddings, and selecting source tasks.

The current state-of-the-art (SOTA) works are established on task embeddings, (i.e., leveraging a single vector to represent a task). They predict inter-task transferability by computing the similarity between task embeddings. Task2Vec (Achille et al., 2019; Vu et al., 2020) develops task embeddings based on the Fisher information matrix while requiring fine-tuning the full model and consuming a large amount of storage (Zhou et al., 2022). Recently, researchers propose that the efficiently tuned parameters like prompts (Li and Liang, 2021; Liu et al., 2021) and LoRA (Hu et al., 2022) encode rich information for a task and thus can serve as task embeddings (Poth et al., 2021; Vu et al., 2022;

[*] Equal contribution.
[†] Corresponding author.

Zhou et al., 2022). However, these tuned parameters are sensitive to model initialization and stochasticity (Li and Liang, 2021; Lester et al., 2021), and optimizing these parameters consumes significantly more computational resources than traditional fine-tuning (Ding et al., 2022).

Different from them, we draw inspiration from the shared working mechanisms of DNNs and biological brains to develop high-quality task embeddings. We start by considering which parts of knowledge within the model are being utilized for a given task. Typically, recent works in sparse optimization and model pruning have shown that sub-structures (e.g., neurons, attention heads, channels, and layers) from different parts of the model exhibit specialization in distinct knowledge and possess varying degrees of importance for a particular task (Dalvi et al., 2020; Liu et al., 2017; Voita et al., 2019a; Glorot et al., 2011; Georgiadis, 2019; Li et al., 2022). These are consistent with the findings in neuroscience that activities of neurons and connectivities in biological brains are sparse (Kerr et al., 2005; Poo and Isaacson, 2009; Barth and Poulet, 2012) and task-specific (Duncan, 2010; Fox et al., 2005; Crinion et al., 2003; Newton et al., 2007). The aforementioned remarkable findings motivate us to use task-specific connectivity patterns in DNNs to represent tasks.

In this work, we propose a novel task embedding, namely **Co**nnectivity **P**atterns **a**s **T**ask **E**mbedding (CoPaTE), and apply it to predict the inter-task transferability, as illustrated in Figure 1. Our key insight is that in over-parameterized DNNs, there exist connectivity patterns (i.e., the structures of subnetworks) that are functional for one certain task, and can capture high-density task-specific information. Concretely, we assign importance masks to attention heads and intermediate neurons of PLMs, jointly train the masks and the model, and extract task embeddings according to the learned masks. Our method has two strengths in efficiency: 1) it is computation-friendly as we extract connectivity patterns early in the training; 2) it is storage-friendly because our embedding granularity is coarse-grained, and CoPaTE can be represented by a binary mask. Experiments show that compared to other approaches, CoPaTE has superior inter-task prediction capability across data regimes and transfer settings. Our codes are available at *Github*[1].

[1] https://github.com/WooooDyy/CoPaTE

Our contributions can be summarized as follows:

- Inspired by the working mechanisms of DNNs and biological brains, we propose CoPaTE, a novel task embedding that represents tasks with sparse connectivity patterns.
- We propose a method to obtain CoPaTE with sparse optimizing techniques, and show the significant positive correlation between embedding similarity and task transferability.
- We conduct thorough experiments on 342 transfer combinations with different settings to show the effectiveness of our method. We further explore an intermediate-curriculum transfer setting to investigate whether there is a beneficial curriculum for a target task.

2 Identifying Sparse, Task-specific Connectivity Patterns

In this section, we demonstrate the framework to identify task-specific connectivity patterns. We represent the task-specific connectivity patterns via the structure of essential subnetworks found by sparse optimizing and pruning techniques (Liu et al., 2017; Chen et al., 2021a; Zheng et al., 2022), including the searching stage (Sec 2.1) and the extracting stage (Sec 2.2).

2.1 Finding Connectivity Patterns

Typically, BERT is constructed by multiple transformer encoder layers that have uniform structure (Vaswani et al., 2017). Each layer has a multi-head self-attention (MHA) block, a feed-forward network (FFN), and residual connections around each block. The MHA is formulated as:

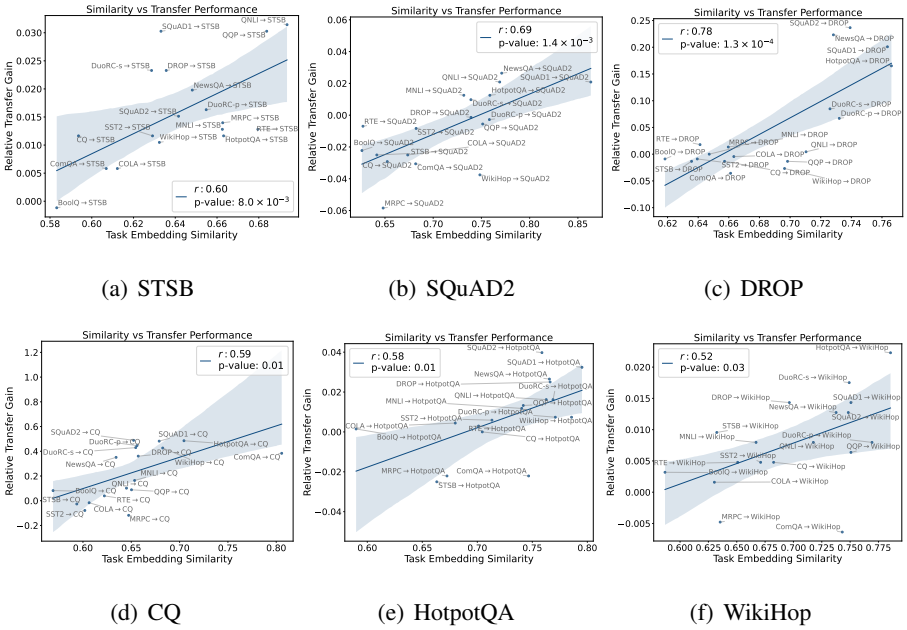
$$\mathrm{MHA}(x) = \sum_{i=1}^{N_h} \mathrm{Att}_{W_K^i, W_Q^i, W_V^i, W_O^i}(x), \quad (1)$$

where x is input, N_h is the number of heads, and the projections $W_K^i, W_Q^i, W_V^i \in \mathbb{R}^{d_h \times d}$, $W_O^i \in \mathbb{R}^{d \times d_h}$ denote the key, query, value and output matrices in the i-th attention head. Here d is the hidden size (e.g., 768), and $d_h = d/N_h$ denotes the output dimension of each head (e.g., 64).

An FFN parameterized by $W_U \in \mathbb{R}^{d \times d_f}$ and $W_D \in \mathbb{R}^{d_f \times d}$ comes next:

$$\mathrm{FFN}(x) = \mathrm{gelu}(XW_U) \cdot W_D, \quad (2)$$

where $d_f = 4d$.



(a) STSB (b) SQuAD2 (c) DROP

(d) CQ (e) HotpotQA (f) WikiHop

Figure 2: Correlation between CoPATE similarity and inter-task transferability. Each point represents a source task to a target task. The x-axis is the similarity between the associated source and target , averaged over three runs, and the y-axis measures the relative transfer gain on the target. We include the Pearson correlation coefficient (r) and p-value. The plots illustrate a significant positive correlation between CoPATE similarity and inter-task transferability. See Appendix B for results on more datasets.

Learnable Importance Masks We adopt a coarse-grained structured pruning strategy to shape connectivity patterns. Specifically, we use the modified network slimming (Liu et al., 2017; Chen et al., 2021a) to find which heads and intermediate neurons are essential for a given task. We first assign learnable importance masks to each head and intermediate neuron:

$$\mathrm{MHA}(x) = \sum_{i=1}^{N_h} m_{\mathrm{H}}^i \cdot \mathrm{Att}_{W_K^i, W_Q^i, W_V^i, W_O^i}(x), \tag{3}$$

$$\mathrm{FFN}(x) = m_{\mathrm{F}} \cdot \mathrm{gelu}(XW_U) \cdot W_D, \tag{4}$$

where m_{H} denotes the masks for heads, i is the index of head, and m_{F} denotes the masks for FFN. Then, we can jointly train BERT with importance masks but with a sparsity-inducing regularizer:

$$\mathcal{R}(m) = \lambda_{\mathrm{H}} \|m_{\mathrm{H}}\|_1 + \lambda_{\mathrm{F}} \|m_{\mathrm{F}}\|_1, \tag{5}$$

where $m = \{m_{\mathrm{H}}, m_{\mathrm{F}}\}$, λ_{H} and λ_{F} denote regularization strength for the two kinds of masks respectively. Hence, the final optimizing objective is:

$$\min_{\theta, m} \mathcal{L}(\theta, m) + \mathcal{R}(m), \tag{6}$$

where $\mathcal{L}$ is the original loss function of fine-tuning.

2.2 Extracting Connectivity Patterns

Early-stopping Strategy Note that the joint training is still as expensive as traditional fine-tuning. Fortunately, (You et al., 2020) and (Chen et al., 2021b) point out that the importance masks converge early in the searching stage. This inspires us to stop the joint training early and dig out early-bird connectivity patterns to generate task embeddings. Nevertheless, it is difficult to determine the exact search termination time as the termination moments of different tasks are different. Moreover, masks of MHA and FFN typically have different convergence rates. Hence, we adopt a termination metric following (Xi et al., 2022) which terminates the searching process when the normalized mask distances between several consecutive miniepochs are all smaller than a threshold γ[2].

Pruning Strategy After the joint training, we can perform pruning to the original models to extract important connectivity patterns that encode task-specific information. Specifically, the self-attention heads and intermediate neurons with the smallest importance masks are believed to contribute the least to the task and the corresponding masks are set to 0, while the masks of the surviving elements

[2]see Appendix C for more details of the termination metric.

Algorithm 1: CoPaTE Generation

Input: model parameters θ, learnable importance masks m, learning rate η, sparsity for self-attention heads p_{H}, and sparsity for intermediate neurons p_{F}.

Procedure Task-specific Connectivity Patterns Searching
Initialize θ to pre-trained weights;
Initialize $m = \{m_{\mathrm{H}}, m_{\mathrm{F}}\}$ to $\mathbf{1}$;
repeat
  $\theta = \theta - \eta \nabla_{\theta}(\mathcal{L}(\theta, c) + \mathcal{R}(c))$;
  $m = m - \eta \nabla_{m}(\mathcal{L}(\theta, m) + \mathcal{R}(m))$;
until *the convergence condition in Sec.2.2 is satisfied, or the fine-tuning is done*;
Procedure Generating CoPaTE with learned Masks
Reset m_{H} and m_{F} to binary form with p_{H} and p_{F} according to mask magnitudes, respectively;
$\mathrm{Emb} = [m_{\mathrm{H}}; m_{\mathrm{F}}]$.

are set to 1. Therefore, we can generate storage-efficient task embeddings with the resulting model structure.

3 CoPaTE: Connectivity Patterns as Task Embedding

In this section, we first show how we generate task embeddings with task-specific connectivity patterns at hand (Sec 3.1). Next we provide empirical evidence for the appropriateness of using the obtained task embeddings to predict inter-task transferability in Sec. 3.2.

3.1 Task Embedding Generating

Typically, the structure of a neural network can be represented as a *mask vector*:

$$\mathrm{m} = [m^1, m^2, ..., m^N], \;\; m^i \in \{0, 1\}, \quad (7)$$

where N denotes the number of elements (i.e., sub-structures) that construct the network and the value of mask m^i indicates whether the i-th element is pruned or not. In our framework, the elements are self-attention heads and intermediate neurons, so the structured subnetworks are represented by:

$$m_{\mathrm{H}} = [m_{\mathrm{H}}^0, m_{\mathrm{H}}^1, ..., m_{\mathrm{H}}^{N_L \times N_h}], \quad (8)$$

$$m_{\mathrm{F}} = [m_{\mathrm{F}}^0, m_{\mathrm{F}}^1, ..., m_{\mathrm{F}}^{N_L \times N_f}], \quad (9)$$

where N_L denotes the number of transformer layers, N_h denotes the number of heads in each layer and N_f is the number of intermediate neurons in each layer. Hence, the resulting task embedding is:

$$\mathrm{Emb} = [m_{\mathrm{H}}; m_{\mathrm{F}}]. \quad (10)$$

We summarize the procedure of generating CoPaTE in Algorithm 1. CoPaTE is quite storage-efficient owing to its binary form. For example, BERT$_{BASE}$ consumes only 4626 bytes to store[3].

3.2 Positive Correlation between CoPaTE Similarity and Task Transferability

We first calculate the similarity between CoPaTEs of different tasks with Hamming Similarity, which is defined as the number of positions at which the corresponding symbols are the same:

$$Sim(V_1, V_2) = \frac{\sum_{i=1}^{n} \sigma(V_1[i], V_2[i])}{n}, \quad (11)$$

where $\sigma(v_1, v_2) = 1$ if $v_1 = v_2$ else 0. Since the numbers of self-attention heads and intermediate neurons differ significantly, we calculate the similarity of the two types of elements separately, and each contributes equally to the final similarity.

We then explore whether the similarity between CoPaTEs is correlated with task transferability. We calculate related transfer gain to measure the impact of transfer learning. Specifically, given a source task s and a target task t, if a baseline PLM that is directly fine-tuned on the target dataset (without any intermediate transferring) achieves a performance of $T(t)$, while a transferred model achieves a performance of $T(s, t)$, the relative transfer gain can be expressed as: $G(s, t) = \frac{T(s, t) - T(t)}{T(t)}$. Figure 2 shows how the relative transfer gain changes as a function of the similarity between the source and target task embeddings. Overall, there is a significant positive correlation between the similarity of task embeddings and task transferability on the majority of the target tasks (16 out of 19). It is possible for the correlation coefficient to attain a high magnitude in many cases, such as on the DROP task, where the correlation coefficient is 0.78 ($p = 0.00013$).

The exciting results suggest that CoPaTE is promising in accurately predicting inter-task transferability. Concretely, for a novel target task, we

[3] BERT$_{BASE}$ has (12×12) heads and (3072×12) intermediate neurons, and requires 37008 bits = 4626 bytes to store.

Data Regime	Method	Classification / Regression (CR)						Question Answering (QA)					
		in-class			*all-class*			*in-class*			*all-class*		
		$R1\downarrow$	$R3\downarrow$	NDCG↑	$R1\downarrow$	$R3\downarrow$	NDCG↑	$R1\downarrow$	$R3\downarrow$	NDCG↑	$R1\downarrow$	$R3\downarrow$	NDCG↑
Full → Full	TextEmb	2.7	1.3	82.6	3.2	2.3	78.3	2.1	0.5	81.1	2.1	**0.5**	81.9
	TaskEmb	2.9	1.3	83.3	2.5	1.6	79.7	3.3	0.9	82.3	3.3	0.8	82.3
	PTuning	2.9	1.4	83.9	3.0	2.3	80.2	2.0	**0.4**	**85.7**	2.0	1.4	82.2
	LoRA	2.5	1.4	83.0	2.5	1.5	79.9	2.8	**0.4**	85.3	6.7	4.4	82.1
	CoPaTE												
	+Early-Emb	2.5	1.3	83.9	2.5	1.6	80.3	**1.1**	**0.4**	84.5	**1.3**	0.9	82.4
	+Lth $_{EP=1}$	2.5	1.4	84.6	**2.2**	**1.2**	80.2	1.2	0.5	83.9	**1.3**	0.9	82.1
	+Lth $_{EP=5}$	**2.3**	**1.2**	**84.9**	2.3	1.3	**81.6**	2.0	**0.4**	84.9	2.2	0.8	**83.0**
Full → Limited	TextEmb	16.4	3.7	60.5	10.7	7.6	52.0	5.8	2.7	68.6	5.5	1.9	73.5
	TaskEmb	15.7	2.9	66.1	**8.9**	**6.5**	52.9	5.7	2.3	73.5	5.7	2.3	75.6
	PTuning	15.2	5.7	66.5	12.4	9.8	52.1	5.6	1.3	**80.9**	4.9	1.2	78.2
	LoRA	14.9	3.5	66.3	9.0	6.6	**53.8**	4.7	**0.7**	79.8	**4.4**	1.1	78.7
	CoPaTE												
	+Early-Emb	15.5	8.0	66.7	14.1	12.2	52.1	7.0	2.7	69.9	10.0	2.7	70.1
	+Lth $_{EP=1}$	**14.2**	2.1	67.3	12.8	10.6	52.2	5.2	2.4	72.7	6.3	2.2	72.1
	+Lth $_{EP=5}$	15.4	**1.1**	**67.7**	13.7	11.1	52.7	**4.2**	**0.7**	80.0	4.7	**0.7**	**79.0**
Limited → Limited	TextEmb	19.4	4.3	61.5	20.2	11.6	46.1	12.8	1.4	65.4	11.2	2.4	69.2
	TaskEmb	**15.9**	5.5	62.6	20.5	10.7	46.8	11.1	1.4	67.3	10.3	1.6	69.5
	PTuning	20.9	10.9	54.5	21.3	19.5	43.6	8.0	1.2	68.3	7.5	1.2	72.4
	LoRA	17.7	3.3	64.4	19.7	10.8	49.4	8.2	1.3	67.5	**7.1**	2.3	70.8
	CoPaTE												
	+Early-Emb	19.3	7.7	63.4	21.6	12.2	46.7	8.3	1.9	69.5	10.1	2.0	69.9
	+Lth $_{EP=1}$	16.0	7.7	63.9	18.5	12.5	47.1	11.0	1.9	72.6	9.9	1.7	72.1
	+Lth $_{EP=5}$	**15.9**	**2.7**	**66.0**	**17.8**	**7.9**	**52.5**	**5.6**	**0.7**	**77.8**	**7.1**	**0.7**	**77.0**

Table 1: Evaluation results of intermediate task selection methods. *In-class* means that the candidate source tasks have the same type as the target task, while *all-class* means the candidate source tasks come from all types of tasks. EP means epochs to search for connectivity patterns. R1 denotes Regret@1 and R3 denotes Regret@3. For NDCG, higher is better; for Regret, lower is better. The best performance in each group is highlighted in **bold**.

rank the candidate source tasks in descending order by the CoPaTE similarity and select the top-ranked task for intermediate fine-tuning.

4 Predicting Task Transferability

In this section, we perform thorough experiments to empirically demonstrate the capability of CoPaTE in predicting inter-task transferability.

4.1 Experimental Setup

Datasets We conduct experiments with 8 tasks of text classification or regression (CR) and 11 tasks of question answering (QA) following previous works (Vu et al., 2020; Zhou et al., 2022). We list the datasets in Appendix A.

Data Regimes For every (source, target) dataset pair, we perform transfer experiments in three data regimes to simulate real-world situations: Full → Full , Full → Limited , and Limited → Limited. The Full regime includes all training data, while in Limited settings, we limit the amount of training data by randomly selecting 1K training examples.

Baselines We compare our method with following strong baselines: (1) **TextEmb** (Vu et al., 2020) averages sentence representations by BERT over the whole dataset. (2) **TaskEmb** (Achille et al., 2019; Vu et al., 2020) embeds tasks based on the Fisher information matrix which captures the curvature of the loss surface. (3) **PTuning** (Vu et al., 2022) interprets the fine-tuned soft prompts in each transformer layer as task embeddings. (4) **LoRA** (Zhou et al., 2022) injects trainable rank decomposition matrics into layers of the model and takes the fine-tuned matrics as task embeddings.

Evaluation Metrics We use the following metrics to evaluate the performance of methods: (1) ***Normalized Discounted Cumulative Gain (NDCG)*** (Järvelin and Kekäläinen, 2002) is a broadly used information retrieval metric aiming to evaluate the quality of a ranking with attached relevances, and it penalizes top-ranked and bottom-ranked mismatches with different weight[4]. (2) ***Regret@k*** (Renggli et al., 2022) measures the relative performance difference between the top k selected

[4]See Appendix D for more details about ***NDCG***.

(a) NDCG on Heads Sparsity (b) NDCG on FFN Sparsity

Figure 3: Impact of sparsity on the performance of CoPATE. The results are from **Full → Full** regime. See more results in Appendix H.

source tasks and the optimal source task[5]. In our experiments, we include $k = 1$ and $k = 3$.

Implementation Details We perform transfer experiments with all (source, target) combinations and use BERT$_{BASE}$ (Devlin et al., 2019b) as the backbone. All the intermediate tuning and target tuning take 3 epochs. For Full → Full regime, we use the results from (Vu et al., 2020). We implement all baseline methods according to their open-source codes and the Transformers library (Wolf et al., 2020). When searching for connectivity patterns in our method, we jointly train the masks and the BERT model for 5 epochs. When extracting early-bird embeddings (i.e., Early-Emb), we set the max searching epoch number to 1. We perform 5 restarts for stable results in Limited regimes. See Appendix F for more details.

4.2 Experimental Results

Table 1 demonstrates the detailed evaluating results. Overall, the proposed CoPATE achieves superior performance across task types, transfer scenarios and data regimes, revealing that it is a robust and accurate predictor of beneficial transfer.

Full → Full In this regime, our method attains impressive performance compared to other baselines. For example, in the setting of *in-class* transfer of Classification tasks, CoPATE exceeds the most competitive baseline by 1.0 in NDCG, and the Regret@3 score achieves 1.2. It is also observed that excessive training steps for identifying task-specific connectivity patterns do not necessarily result in large performance improvement in this regime. The efficient Early-Emb performs slightly worse than Lth $_{EP=5}$, but still performs comparably.

[5]See Appendix E for more results about ***Regret@k***

Full → Limited In this few-shot regime, our method achieves comparable performance to SOTA baselines. However, we find that in QA tasks, the performance of CoPATE degrades sharply as the number of training steps utilized during the search stage decreases. Compared to Lth $_{EP=5}$, Early-Emb's NDCG on in-class and all-class decreased by 10.1 and 8.9, respectively. This trend is also observable in **Limited → Limited** regime. It is not surprising as QA tasks are typically more complex and the connectivity patterns require more training steps to converge better. This suggests a trade-off between performance and efficiency when facing limited examples, and additional training resources should be allocated to the search stage to extract high-quality task embeddings.

Limited → Limited In this regime, CoPATE demonstrates exceptional performance and surpasses other existing baselines by a significant margin. For instance, our method outperforms the strongest baseline by 9.5 in terms of NDCG on in-class transfer of QA tasks, and 4.6 on all-class transfer of QA tasks.

5 Discussion

5.1 Ablation Study

In this section, we perform ablation studies to show the contribution of each component of our method.

Head v.s. FFN Previous experiments utilize both masks of attention heads and intermediate neurons to compute similarity. Here, the contribution of each component is evaluated individually by separately using them to calculate similarity and subsequently assessing the NDCG. Table 2 shows that both components play essential roles in ranking source tasks. We observe that on CR tasks, heads outperform FFN by a large margin, revealing that heads are more important in such tasks.

Impact of Sparsity Figure 3 illustrates the relationship between the level of sparsity and the performance of the obtained embeddings. The performance of the model is significantly impacted by variations in the pruning ratio of heads or FFN when the target tasks are CR, while such variations have a limited effect when the target tasks are QA, revealing that CR tasks are more sensitive to embedding sparsity. After comprehensive consideration, we believe that $1/3$ and 0.4 are reasonable sparsity for heads and FFN, respectively.

Method	CR		QA	
	in-cls	*all-cls*	*in-cls*	*all-cls*
EARLY-EMB	**83.9**	**80.3**	84.5	82.4
w/o Head	78.8	75.2	82.5	**83.8**
w/o FFN	83.3	80.0	**85.0**	81.1

Table 2: Ablation results when heads or intermediate neurons are removed from similarity computing. The results are from **FULL** → **FULL** and others are in Appendix G. In-cls and all-cls are short forms of in-class and all-class, respectively; **w/o** means "without". Both heads and FFN are important for ranking source tasks.

Method	#Time	#Storage
TEXTEMB	0.43×	$3.1K$
TASKEMB	4.22×	$437.9M$
PTUNING	14.43×	$122.9K$
LORA	16.83×	$98.3K$
COPATE		
+EARLY-EMB	0.38×	$4.6K$
+LTH $_{EP=1}$	1.03×	$4.6K$
+LTH $_{EP=5}$	5.12×	$4.6K$

Table 3: Evaluation of time and storage consumptions. We average the results on all datasets. The #Time is quantified as a multiple of the duration of the traditional fine-tuning for a single epoch. We get results from one NVIDIA 3090 GPU for a fair comparison. The #Storage is in bytes and each $float$ number requires 4 bytes.

We include more ablation studies of pruning strategies, early-stopping thresholds, and the sparsity-inducing regularizer in Appendix I.

5.2 Computation and Storage Consumption

Table 3 lists the computational and storage cost of each method. COPATE demonstrates efficiency in both aspects thanks to proper designs (i.e., early-stopping, structured pruning and binary form of embeddings), particularly EARLY-EMB, which exhibits the fastest generation speed and only requires $4.6K$ bytes to store. TASKEMB is also computation-efficient, but it requires much more storage than COPATE. While TEXTEMB is the only method that is comparable to our approach in terms of efficiency, it falls behind EARLY-EMB with an average difference of 1.6 in NDCG.

Further Storage-efficiency with Task-specific Layers Previous studies have established that in BERT, layers are redundant (Dalvi et al., 2020), and that shallower transformer layers contain more general information while deeper layers contain more task-specific information (Voita et al., 2019a; Kim et al., 2020; Sajjad et al., 2020). These insights shed light on further reducing the storage of COPATE by representing tasks using a select number of layers, or even a single layer. Figure 4 illustrates the evaluated performance. We observe that: (1) Using a select number of layers does not result in a significant decrease in performance, and sometimes delivers better performance. (2) Top-down strategy outperforms bottom-up strategy, and consistently exceeds the full model in few-shot settings, showing that deep layers can effectively encode task-specific information, which is in line with previous studies. As a result, if we adopt the last six layers for embedding generation, 50% of the storage can be saved, while little decrease in performance is incurred. We also explore the potential of generating embeddings using a single layer, while sacrificing little performance in Appendix J.

Curriculum Type	Similar-first	Different-first	Recursive-similar
Performance Gain	+2.35	+2.43	+2.56

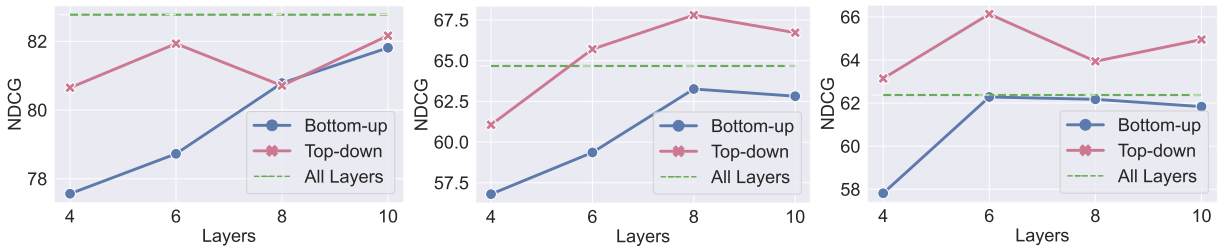
Table 4: Performance gain yielded by each curriculum. The results are an average on all 19 tasks.

5.3 COPATE Captures Task Relationships

The heatmap in Figure 5 illustrates the hierarchical clustering of the similarities between COPATEs. The results indicate that the obtained embeddings effectively capture various intuitive task relationships. We observe that tasks with similar characteristics congregate in clusters, such as QA tasks (WikiHop, SQuAD-1, SQuAD-2, DuoRC-s, DuoRC-p, NewsQA, and HotpotQA), similarity and paraphrasing tasks (STS-B and MRPC), NLI tasks (QNLI and MNLI), and single sentence classification tasks (SST-2 and CoLA). In particular, a closer examination of the clustering reveals that SQuAD-1 and SQuAD-2 are closely grouped together, with the latter being an extension of the former (Rajpurkar et al., 2016, 2018). Furthermore, the tight clustering of DuoRC-p and DuoRC-s is also noteworthy, as they are variations of the same movie plots with different lengths (Saha et al., 2018).

5.4 Intermediate-curriculum Transfer

Here, we extend the boundary of intermediate-task transfer and examine the potential benefits of a specific intermediate task curriculum (i.e., a particular order to arrange several tasks) to a target task using COPATE. Three distinct curriculum strategies are considered: (1) **Similar-first strategy** which selects the three tasks that are most similar to the



(a) FULL → FULL (b) FULL → LIMITED (c) LIMITED → LIMITED

Figure 4: The impact of using different transformer layers for embedding generation. The number of used layers is shown on the x-axis; that number is either selected "bottom-up" or "top-down". More precisely, a bottom-up setting selecting 4 layers means we use the transformer layers $\{0, 1, 2, 3\}$; a top-down setting selecting 4 layers means we mask the transformer layers $\{8, 9, 10, 11\}$. The NDCG is an average of different transfer settings.

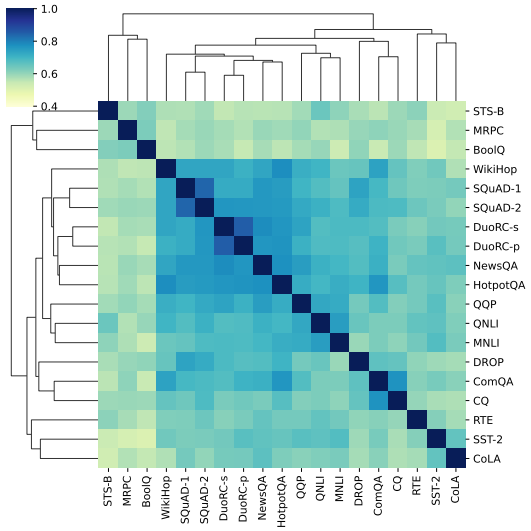


Figure 5: A clustered heatmap of similarities between the task embeddings of 19 NLP tasks. Our COPATE capture task relationships: similar tasks cluster together.

target task and arranges the intermediate tasks in a sequential order of similarity. (2) **Different-first strategy** which also selects the three tasks that are most similar to the target task, but arranges the intermediate tasks in an order of dissimilarity. (3) **Recursive-similar strategy** which starts from the target task, recursively finds the task that is most similar to the current task three times, stacks them, and then sequentially pops these found tasks for intermediate fine-tuning. The results in Table 4 show that: (1) Each curriculum can boost the target task, validating the value of intermediate-task transfer. (2) The recursive-similar strategy yields the most performance gain, suggesting that making each intermediate task learned better can deliver more benefits to target tasks. (3) The different-first strategy performs better than the similar-first, implying that intermediate tasks that are similar to the target task should be assigned later.

6 Related Work

Predicting Beneficial Intermediate Tasks It has been shown that intermediate-task transfer can deliver performance gains for many target tasks (Phang et al., 2018; Wang et al., 2019a; Talmor and Berant, 2019; Liu et al., 2019), but improper intermediate tasks can result in negative transfer results (Yogatama et al., 2019; Pruksachatkun et al., 2020). Hence, researchers try to accurately identify the most beneficial source task based on metadata or extracted representations of tasks (Alonso and Plank, 2017; Vu et al., 2020; Poth et al., 2021). Recent works represent tasks with embeddings that are generated from data representations (Vu et al., 2020), model weight information (Achille et al., 2019; Vu et al., 2020), and efficiently tuned parameters (Poth et al., 2021; Vu et al., 2022; Zhou et al., 2022). Different from them, we start from a model architecture perspective and use connectivity patterns to represent tasks.

Techniques to Obtain Sparse Subnetworks Researchers have explored a variety of techniques to obtain sparse networks by removing sub-structures like weights (Louizos et al., 2018; Frankle and Carbin, 2019; Sanh et al., 2020; Xu et al., 2021), channels (He et al., 2017; Luo et al., 2017; Liu et al., 2017; Molchanov et al., 2019), attention heads (Voita et al., 2019b; Michel et al., 2019; Li et al., 2021) and layers (Fan et al., 2020; Sajjad et al., 2020). These approaches first identify unimportant sub-structures and subsequently remove them. With the increasing size of PLMs, sparse subnetworks have become increasingly important for efficient deployment and inference in NLP, lead-

ing to a proliferation of related research (Prasanna et al., 2020; Hou et al., 2020; Lagunas et al., 2021; Xia et al., 2022). Our proposed method, which uses connectivity patterns as task embeddings, is orthogonal to these existing techniques.

7 Conclusion

In this work, we propose CoPaTE, a novel task embedding that represents tasks with sparse connectivity patterns, and develop a method to get such embeddings. Comprehensive experiments show that the proposed method outperforms other competitive approaches in predicting inter-task transferability while achieving efficiency in both computation and storage. We hope that our work may motivate future work in introducing connectivity patterns as task embeddings to fields like meta learning, multi-task learning, and model interpretability.

Limitations

While the proposed method has demonstrated superior performance and high efficiency, there are several limitations that warrant further investigation: (1) In few-shot settings where the number of training examples is limited, the performance of our method and other baselines drops significantly. Future work should focus on uncovering essential features of the task in few-shot scenarios and generating embeddings of higher quality. (2) The storage consumption has been reduced to a small amount, however, the number of neurons is still relatively large compared to that of heads and therefore becomes a bottleneck for further decreasing storage requirements. As discussed in Sec 5.2, one possible solution is reducing the number of layers used to generate the embedding. Future work could also include assigning intermediate neurons into groups to make the embedding coarser in granularity, thus reducing storage requirements.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No.62076069,62206057,61976056), Shanghai Rising-Star Program (23QA1400200), and Natural Science Foundation of Shanghai (23ZR1403500).

References

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2019. ComQA: A community-sourced dataset for complex factoid question answering with paraphrase clusters. 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 307–317, Minneapolis, Minnesota. Association for Computational Linguistics.

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. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6429–6438. IEEE.

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 44–53. Association for Computational Linguistics.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514, Osaka, Japan. The COLING 2016 Organizing Committee.

Alison L. Barth and James F.A. Poulet. 2012. Experimental evidence for sparse firing in the neocortex. *Trends in Neurosciences*, 35(6):345–355.

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*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang, and Jingjing Liu. 2021a. Early-bert: Efficient BERT training via early-bird lottery tickets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2195–2207. Association for Computational Linguistics.

Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang, and Jingjing Liu. 2021b. Early-bert: Efficient bert training via early-bird lottery tickets. *ArXiv*, abs/2101.00063.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jennifer T. Crinion, Matthew A. Lambon-Ralph, Elizabeth A. Warburton, David Howard, and Richard J. S. Wise. 2003. Temporal lobe regions engaged during normal speech comprehension. *Brain*, 126(5):1193–1201.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4908–4926. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *CoRR*, abs/2203.06904.

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

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

John Duncan. 2010. The multiple-demand (md) system of the primate brain: mental programs for intelligent behaviour. *Trends in Cognitive Sciences*, 14(4):172–179.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Michael D. Fox, Abraham Z. Snyder, Justin L. Vincent, Maurizio Corbetta, David C. Van Essen, and Marcus E. Raichle. 2005. The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *Proceedings of the National Academy of Sciences*, 102(27):9673–9678.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv: Learning*.

Georgios Georgiadis. 2019. Accelerating convolutional neural networks via activation map compression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7085–7095. Computer Vision Foundation / IEEE.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 315–323. JMLR.org.

Yihui He, Xiangyu Zhang, and Jian Sun. 2017. Channel pruning for accelerating very deep neural networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1398–1406. IEEE Computer Society.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic BERT with adaptive width and depth. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs.

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

Jason N. D. Kerr, David Greenberg, and Fritjof Helmchen. 2005. Imaging input and output of neocortical networks in vivo. *Proceedings of the National Academy of Sciences*, 102(39):14063–14068.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. 2020. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M. Rush. 2021. Block pruning for faster transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10619–10629. 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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. 2021. Differentiable subset pruning of transformer heads. *Trans. Assoc. Comput. Linguistics*, 9:1442–1459.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J. Reddi, Ke Ye, Felix X. Chern, Felix X. Yu, Ruiqi Guo, and Sanjiv Kumar. 2022. Large models are parsimonious learners: Activation sparsity in trained transformers. *CoRR*, abs/2210.06313.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1073–1094. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning efficient convolutional networks through network slimming. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2755–2763. IEEE Computer Society.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning sparse neural networks through l_0 regularization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. 2017. Thinet: A filter level pruning method for deep neural network compression. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5068–5076. IEEE Computer Society.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14014–14024.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11264–11272. Computer Vision Foundation / IEEE.

Allen T. Newton, Victoria L. Morgan, and John C. Gore. 2007. Task demand modulation of steady-state functional connectivity to primary motor cortex. *Human Brain Mapping*, 28(7):663–672.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. 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.

Cindy Poo and Jeffry S. Isaacson. 2009. Odor representations in olfactory cortex: "sparse" coding, global inhibition, and oscillations. *Neuron*, 62(6):850–861.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10585–10605. Association for Computational Linguistics.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT plays the lottery, all tickets are winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3208–3229. 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, ACL 2020, Online, July 5-10, 2020*, pages 5231–5247. 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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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.

Cédric Renggli, André Susano Pinto, Luka Rimanic, Joan Puigcerver, Carlos Riquelme, Ce Zhang, and Mario Lucic. 2022. Which model to transfer? finding the needle in the growing haystack. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 9195–9204. IEEE.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. Poor man's BERT: smaller and faster transformer models. *CoRR*, abs/2004.03844.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. 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, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Alon Talmor and Jonathan Berant. 2019. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4911–4921. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer

Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

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

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4395–4405. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019b. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5797–5808. Association for Computational Linguistics.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022. 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5039–5059. 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 2020, Online, November 16-20, 2020*, pages 7882–7926. 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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4465–4476. 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.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Zhiheng Xi, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Efficient adversarial training with robust early-bird tickets. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Abu Dhabi. Association for Computational Linguistics.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1513–1528. Association for Computational Linguistics.

Dongkuan Xu, Ian En-Hsu Yen, Jinxi Zhao, and Zhibin Xiao. 2021. Rethinking network pruning - under the pre-train and fine-tune paradigm. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2376–2382. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for

diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome T. Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *CoRR*, abs/1901.11373.

Haoran You, Chaojian Li, Pengfei Xu, Y. Fu, Yue Wang, Xiaohan Chen, Yingyan Lin, Zhangyang Wang, and Richard Baraniuk. 2020. Drawing early-bird tickets: Towards more efficient training of deep networks. *ArXiv*, abs/1909.11957.

Rui Zheng, Bao Rong, Yuhao Zhou, Di Liang, Sirui Wang, Wei Wu, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Robust lottery tickets for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2211–2224. Association for Computational Linguistics.

Wangchunshu Zhou, Canwen Xu, and Julian J. McAuley. 2022. Efficiently tuned parameters are task embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Abu Dhabi. Association for Computational Linguistics.

Appendices

A List of Datasets

See Table 5 for details of datasets.

Task	|Train|
Text classification / Regression (CR)	
MNLI (Williams et al., 2018)	$393K$
QQP (Iyer et al., 2017)	$364K$
QNLI (Wang et al., 2019b)	$105K$
SST-2 (Socher et al., 2013)	$67K$
CoLA (Warstadt et al., 2019)	$8.5K$
STS-B (Cer et al., 2017)	$7K$
MRPC (Dolan and Brockett, 2005)	$3.7K$
RTE (Dagan et al., 2005)	$2.5K$
Question Answering (QA)	
SQuAD-2 (Rajpurkar et al., 2018)	$162K$
NewsQA (Trischler et al., 2017)	$120K$
HotpotQA (Yang et al., 2018)	$113K$
SQuAD-1 (Rajpurkar et al., 2016)	$108K$
DuoRC-p (Saha et al., 2018)	$100K$
DuoRC-s (Saha et al., 2018)	$86K$
DROP (Dua et al., 2019)	$77K$
WikiHop (Welbl et al., 2018)	$51K$
BoolQ (Clark et al., 2019)	$16K$
ComQA (Abujabal et al., 2019)	$11K$
CQ (Bao et al., 2016)	$2K$

Table 5: The datasets used in our experiments, grouped by task class and sorted by training dataset size.

B More Results of Correlation between CoPaTE Similarity and Inter-task Transferability

See Figure 6 for more results of correlation between CoPaTE similarity and inter-task transferability. There is a significant positive correlation between the similarity of task embeddings and task transferability on most target tasks.

C More Details of Early-stopping Strategy

We use Hamming distance to calculate the normalized normalized mask distance. We stop the searching stage when the normalized mask distances between consecutive 5 miniepochs are all smaller than γ. Each miniepoch consists of 0.05 epochs. We set γ to 0.05 in all settings. This is not the best choice for all transfer scenarios, but we unify the value of hyper-parameters for the sake of generality.

D More Details of NDCG

The NDCG is defined using the *Discounted Cumulative Gain (DCG)*, which is a measure of the relevance score for a list of items, each discounted by its position in the ranking. The DCG of a ranking R at a particular rank position p can be calculated as:

$$\mathrm{DCG}_p(R) = \sum_{i=1}^{p} \frac{2^{\mathrm{rel}_i} - 1}{\log_2(i+1)}$$

In our experiments, R refers to a 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. $\mathrm{rel}_i \in [0, 100]$. We set $p = |\mathcal{S}|$, which is the number of intermediate tasks.

The NDCG finally normalizes the DCG of the ranking predicted by the task selection approach (R_{pred}) by the golden ranking produced by the empirical transfer results (R_{true}). An NDCG of 100% indicates the best ranking.

$$\mathrm{NDCG}_p(R) = \frac{\mathrm{DCG}_p(R_{pred})}{\mathrm{DCG}_p(R_{true})}$$

E More Details of Regret@k

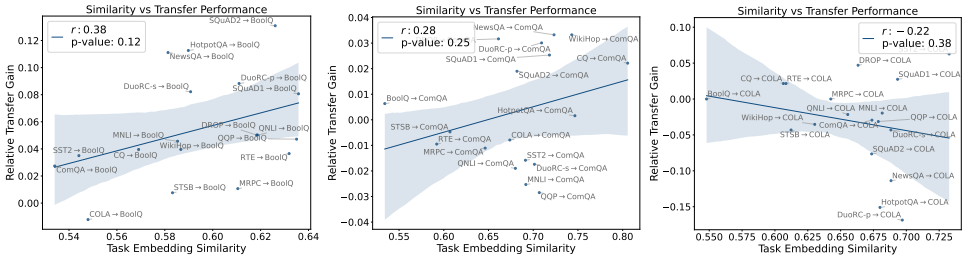
Regret@k is defined as:

$$\mathrm{Regret}_k = \frac{\overbrace{\max_{s \in \mathcal{S}} \mathbf{E}[T(s,t)]}^{O(\mathcal{S},t)} - \overbrace{\max_{\hat{s} \in \mathcal{S}_k} \mathbf{E}[T(\hat{s},t)]}^{M_k(\mathcal{S},t)}}{O(\mathcal{S},t)} \times 100\%$$

where $T(s,t)$ means the performance on target task t when transferring from source task s. $O(\mathcal{S},t)$ is the expected target task performance of the optimal selection. $M_k(\mathcal{S},t)$ denotes the highest performance on t among the k top-ranked source tasks of the evaluated selection method. In our experiments, we include $k = 1$ and $k = 3$.

F More Implementation Details

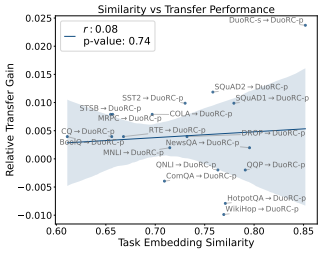
For classification/regression tasks, we set the max sequence length to 128. For question answering tasks, we set the max sequence length to 384. The batch size for all experiments is set to 32. Our experiments are performed on twelve NVIDIA GeForce RTX 3090 GPUs. We perform 3 restarts for our experiments and report the mean. For



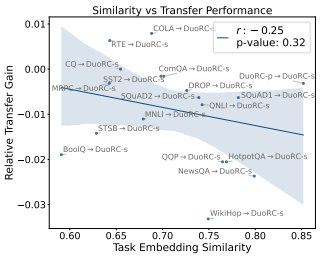
(a) BoolQ

(b) ComQA

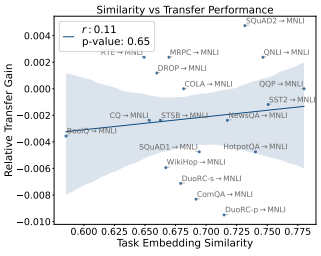
(c) COLA



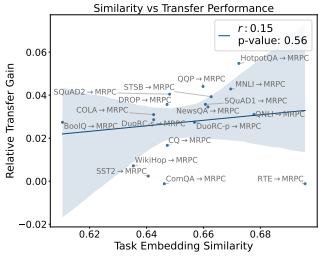
(d) DuoRC-p



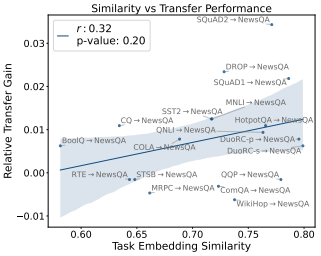
(e) DuoRC-s



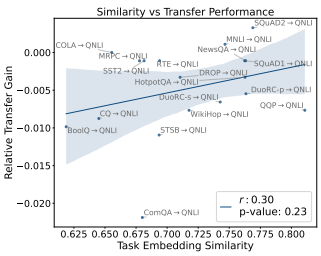
(f) MNLI



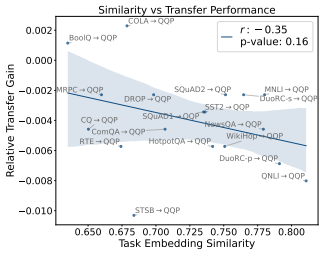
(g) MRPC



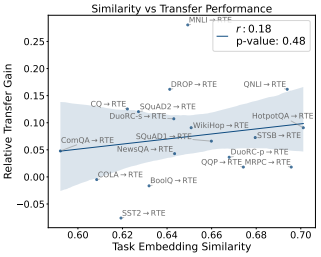
(h) NewsQA



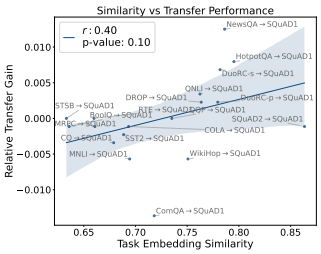
(i) QNLI



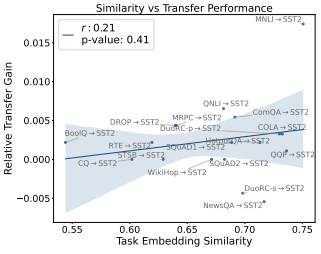
(j) QQP



(k) RTE



(l) SQuAD-1



(m) SST2

Figure 6: More results of correlation between CoPATE similarity and inter-task transferability. Each point represents a source task to a target task.

Method	CR		QA	
	in-cls	*all-cls*	*in-cls*	*all-cls*
EARLY-EMB	**66.7**	52.1	**69.9**	**70.1**
w/o Head	62.2	**55.9**	60.8	59.9
w/o FFN	**66.7**	48.4	64.7	61.5

Table 6: Ablation results when heads or intermediate neurons are removed from similarity computing in FULL → LIMITED regime.

Method	CR		QA	
	in-cls	*all-cls*	*in-cls*	*all-cls*
EARLY-EMB	**63.4**	46.7	**69.5**	**69.9**
w/o Head	57.8	48.2	68.5	67.5
w/o FFN	63.3	**48.3**	65.8	64.2

Table 7: Ablation results when heads or intermediate neurons are removed from similarity computing in LIMITED → LIMITED regime.

PTUNING, we adopt P-Tuning v2 in (Liu et al., 2021), which implements a prompt tuning method by introducing additional attention prefix matrices to each transformer layer. We set the prefix length to 20. For LORA, we set the r to 8 and α to 8. For the searching stage of winning tickets, we set the regularization strength λ_{H} and λ_{F} to $1e-4$.

G More Results of Head v.s. FFN

Table 6 and Table 7 show the results of Head v.s. FFN in FULL → LIMITED and LIMITED → LIMITED, respectively. We can still find that both of them are important for high-quality task embeddings.

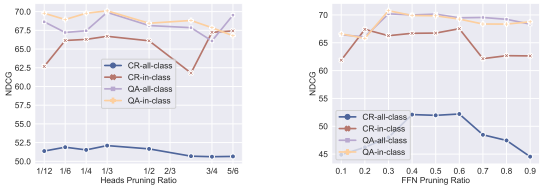
H More Results of Impact of Sparsity

Figure 7 and Figure 8 show the results of impact of sparsity in FULL → LIMITED and LIMITED → LIMITED, respectively. We can still find that $1/3$ and 0.4 are reasonable sparsity for heads and FFN, respectively.

I More Ablation Studies

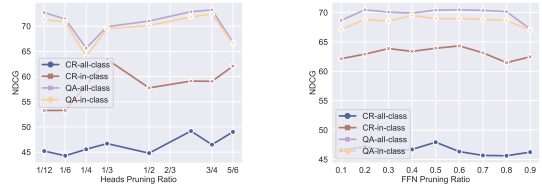
I.1 Impact of Pruning Strategies

In this section, we investigate the impact of different pruning strategies to the embedding performance. Results in Table 8, Table 9 and Table 10 show that layerwise pruning and global pruning are proper strategies for self-attention heads and FFN, respectively.



(a) NDCG on Heads Sparsity (b) NDCG on FFN Sparsity

Figure 7: Impact of sparsity on the performance of COPATE. The results are from **FULL → LIMITED** regime.



(a) NDCG on Heads Sparsity (b) NDCG on FFN Sparsity

Figure 8: Impact of sparsity on the performance of COPATE. The results are from **LIMITED → LIMITED** regime.

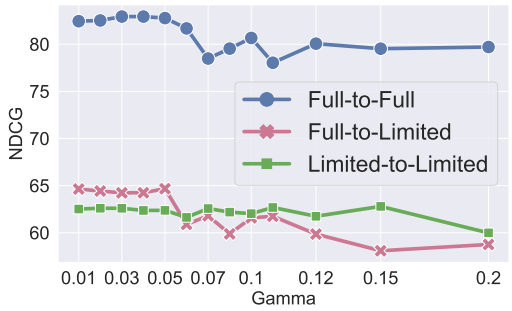


Figure 9: Impact of different early-stopping thresholds γ. Each line in the figure represents a data regime, and we report the mean results of different transfer settings. We can observe that the performance of embeddings converges when γ reduces to near 0.05.

I.2 Impact of Different Early-Stopping Thresholds

In this section, we investigate the impact of different values of the early-stopping threshold γ. Results in Figure 9 show that the performance of COPATE converges when γ reduces to near 0.05.

I.3 Importance of Sparsity-inducing Regularizer

In this section, we investigate the importance of the sparsity-inducing regularizer during the connectivity pattern searching stage. Results in Table 11 show that the regularizer is indispensable for

Strategy	FFN-Global	FFN-Layerwise
Head-Global	76.2	78.7
Head-Layerwise	**82.8**	81.8

Table 8: Impact of pruning strategies in FULL → FULL regime. The results are NDCG scores averaged on different transfer settings.

Strategy	FFN-Global	FFN-Layerwise
Head-Global	55.4	58.7
Head-Layerwise	**64.7**	63.8

Table 9: Impact of pruning strategies in FULL → LIMITED regime. The results are NDCG scores averaged on different transfer settings.

Strategy	FFN-Global	FFN-Layerwise
Head-Global	61.1	61.3
Head-Layerwise	**62.4**	61.9

Table 10: Impact of pruning strategies in LIMITED → LIMITED regime. The results are NDCG scores averaged on different transfer settings.

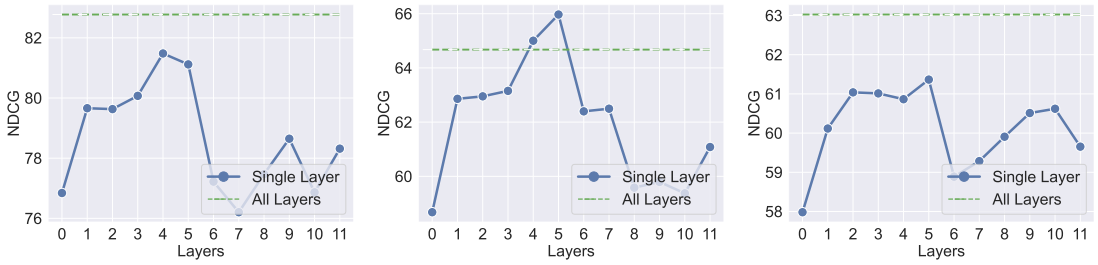
Method	FULL → FULL	FULL → LIMITED	LIMITED → LIMITED
EARLY-EMB	**82.8**	**64.7**	**62.4**
w/o Regularizer	72.3	58.2	52.5

Table 11: Ablation results if we remove the sparsity-inducing regularizer during connectivity pattern searching. We report the average results of different settings.

generating high-quality task embeddings.

J Further Storage-efficiency with Single Layer

In this study, we examine the performance of COPATE when utilizing a single layer to generate task embeddings. The results, as illustrated in Figure 10, demonstrate the performance of each layer. The findings indicate that a single layer can yield performance comparable to that of the full model. Specifically, when the fifth layer is used to generate the embedding, there is a significant reduction of 91.7% in the storage space required for the embedding, while the final NDCG score is only slightly lower, at 0.67 on average, as compared to the full model.



(a) FULL → FULL (b) FULL → LIMITED (c) LIMITED → LIMITED

Figure 10: The impact of using one single transformer layer for embedding generation. The NDCG is an average of different transfer settings.