

# DEPT: Decoupled Embedding for Pre-Training LMs

**Oral Presentation at ICLR** 

Alex Iacob\*, Lorenzo Sani\*, Meghdad Kurmanji, William F. Shen, Xinchi Qiu, Dongqi Cai, Yan Gao, Nicholas Donald Lane

# **DEPT: Decoupled Embedding Pre-Training**

- Issue: Shared vocabularies result in suboptimal tokenization and embeddings
- Solution: Separate embeddings from the transformer blocks



UNIVERSITY OF CAMBRIDGE

**ICLR** 

#### CAMBRIDGE 🙆 🧑 🎆 ICLR

# From strings to tokens

#### The set of words is unlimited

- Subword tokenization (byte-pair encoding) balances encoding every word and char-level models
- Effectiveness depends on the pre-training corpus

Input:	The quick brown fox jumps over the lazy dog!
Tokens:	<pre>"<sos>", "The", "_quick", "_brown", "_fox", "_jumps", "_over", "_the", "_lazy", "_dog", "!", "<eos>"</eos></sos></pre>
Token IDs:	[976, 4853, 19705, 68347, 65613, 1072, 290, 29082, 6446, 0]

```
def merge vocab(pair, v in):
  v \text{ out} = \{\}
 bigram = re.escape(' '.join(pair))
 p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
 for word in v in:
   w out = p.sub(''.join(pair), word)
   v_out[w_out] = v_in[word]
  return v out
vocab = {'low </w>': 5, 'low er </w>': 2,
         'newest </w>':6, 'widest </w>':3}
num merges = 10
for i in range(num merges):
 pairs = get stats(vocab)
 best = max(pairs, key=pairs.get)
  vocab = merge vocab(best, vocab)
 print(best)
```

```
\begin{array}{cccc} r \cdot & \to & r \cdot \\ l \ o & \to & l o \\ l o \ w & \to & l o w \\ e \ r \cdot & \to & e r \cdot \end{array}
```

Figure 1: BPE merge operations learned from dictionary {'low', 'lowest', 'newer', 'wider'}.

Sennrich, et.al., "Neural Machine Translation of Rare Words with Subword Units"

# Data Heterogeneity

- Languages, mathematics, code vary in vocabulary / syntax / semantics
- The differences cause
  - The curse of multilinguality
  - Negative interference
- Adding more data sources can cause vocabulary dilution + capacity contention



# **DEPT: Decoupled Embedding Pre-Training**

- Issue: Shared vocabularies result in suboptimal tokenization and embeddings
- Solution: Separate embeddings from the transformer blocks



UNIVERSITY OF CAMBRIDGE

**ICLR** 



# DEPT Can...

### **#1 Enable vocabulary-agnostic training**

- 1. Allows each data source to have its own optimized vocabulary
- 2. Avoids vocabulary dilution and capacity contention in the embeddings

### **#2 Reduce comms and memory**

- 1. Shrinks vocabulary size by manipulating the embedding matrices
- 2. Avoids training and communicating tokens which are not relevant to a data source

### **#3 Improve transformer bodies**

- 1. DEPT-trained transformer bodies show improved generalization to downstream tasks
- 2. They also show greater plasticity when adapting to new data distributions

# CAMBRIDGE 🧿 🦓 🎆 ICLR

# Federated Learning (FL) for Pre-Training

- Standard centralized learning algorithms like SGD assume data is independent and identically distributed (IID)
- In FL this assumption often breaks due to the private nature of data
- For LLMs this may be modeled by splitting languages / domain



Sani, et.al., "Photon: Federated LLM Pre-Training"

















# **DEPT Results**

Method	Embodding Layer Parameters	Total Trainable Parameters	Per-stop Comme Cost (4)
ano	192M	278M [1x]	228M (1×)
61.08	19214	278M (1x)	0.56M (0.002 x )
TRIN	166M	252M (0.92 x)	0.SM (0.002;x)
SPRC	166M	252M (0.92 x)	0.17M (0.0006 x)
SPEC-OFT	38.6M	125M (0.45 ×)	0.17M (0.0006×)
STD SPRC-OPT	512.2 M 102.9 M	1718 (1×) 138 (0.76×)	1318 [1×] 2.4M (0.001×)

### CAMBRIDGE

### **DEPT Improves Comms**

Method	Embedding Layer	Total Trainable	Per-stop
	Parameters	Parameters	Comme Cost (↓)
STD	19254	238M (1×)	228M (1 x)
CLOB	19254	238M (1×)	0.56M (0.002 x)
TRIN	16654	253M (092×)	0.5M (0.0002 x)
SPEC	16654	253M (092×)	0.37M (0.0006 x)
SPEC-OPT	38.654	125M (0.45×)	0.17M (0.0006 x)
STD	512-254	1718 (1×)	1318 (1×)
SPEC-OPT	102-964	138 (0.76×)	2.4M (0.001×)

# 500x Reduction in Comms (all scales)

#### CAMBRIDGE 🙆 🥐 🎆 ICLR

# **DEPT Improves Memory**

Method	Embedding Layer	Total Trainable	Per-stop
	Parameters	Parameters	Comme Cost (4)
STD	19254	238M (1×)	238M (1)x)
CLOB	19254	238M (1×)	0.56M (0.002 x)
TRIN	16654	252M (092×)	0.5M (0.002 x)
SPRC	166M	252M (0.92 ×)	0.17M (0.0006×)
SPRC-OPT	38.6M	125M (0.45 ×)	0.17M (0.0006×)
STD	512-314	1718 (1×)	1718 (1×)
SPEC-OPT	102-944	138 (0.76×)	2.4M (0.001×)

# 80%

**Reduction in Embedding Parameters (at >1B scale)** 



# **DEPT SPEC Improves Comms**

Method	Embodiding Layor	Total Trainable	Per-stop
	Parameters	Parameters	Comme Cost (4)
STD	19254	278M (1 ×)	238M (1×)
GLOB	19254	278M (1 ×)	0.56M (0.002×)
TRIN	16654	252M (0.92 ×)	0.5M (0.002×)
SPRC	16654	252M (0.92 ×)	0.37M (0.0006×)
SPRC-OPT	38.6M	125M (0.45 ×)	0.17H (0.0006×3)
	512.2M	1718 (1×)	1.71B (1×3)
	102.9M	1.39 (0.76 ×)	2.4M (0.001×3)

714x

**Reduction in Communicated Parameters (>1B scale)** 



## **DEPT Improves Downstream Performance**

	Random Init			
Name	RACE (ACC)	MNLI (ACC)	STSB (PC)	SST2 (ACC)
STD $(\tau = 0)$	0.50	0.60	0.66	0.79
STD $(\tau = 1)$	0.46	0.68	0.73	0.81
ACT	0.45	0.66	0.73	0.80
GLOB	0.51	0.72	0.78	0.83
TRIM	0.53	0.71	0.78	0.83
SPEC	0.52	0.71	0.79	0.81
SPEC-OPT	0.51	0.69	0.77	0.85
Min Imp (%)	2.9%	4.6%	5.9%	-0.7%
Max Imp (%)	5.8%	6.1%	7.5%	4.1%

# 4.1 – 7.5%

Improved downstream task performance



### **DEPT Improves Plasticity**





# A New Pre-training Paradigm

# Flexibility

Train on diverse—and even private—data sources without managing one global vocabulary

# Efficiency

Slash communication and memory costs, enabling large-scale, low-bandwidth pre-training

### Generality

Produce versatile foundation models that excel across tasks and adapt to new domains



# **Questions?**



#### CAMBRIDGE 🙆 🖗 🦄 ICLR

Table 1: Memory and communication costs of DEPT, where:  $\mathcal{M}$  is the number of model parameters;  $|\mathcal{V}|$  is the global vocabulary size;  $\overline{|\mathcal{V}_k|}$  is the mean data source vocabulary size;  $d_{\text{model}}$  is the embedding dimension;  $N_{\text{local}} = N/T$  is the number of local steps done per iteration for a total number steps N;  $\mathcal{L}$  is the sequence length. GLOB reduces comms by only communicating every  $N_{\text{local}}$ steps while TRIM also reduces embedding size. SPEC brings further reductions over TRIM by not sharing token or position embeddings. The standard baseline is assumed to be distributed training with per-step synchronization. Concrete numbers for our models (see Table 8) are shown in Table 2.

Method	<b>Memory Cost</b>	Per-step Comms Cost	Vocab Agnostic
STD	$\mathcal{O}(\mathcal{M})$	$\mathcal{O}(\mathcal{M})$	×
GLOB	$\mathcal{O}(\mathcal{M})$	$\mathcal{O}ig(rac{\mathcal{M}}{N_{ ext{local}}}ig)$	×
TRIM	$\mathcal{O}(\mathcal{M} - ( \mathcal{V}  - \overline{ \mathcal{V}_k }) d_{ ext{model}})$	$\mathcal{O}ig(rac{\mathcal{M}-( \mathcal{V} -\overline{ \mathcal{V}_k })d_{ ext{model}}}{N_{ ext{local}}}ig)$	×
SPEC	$\mathcal{O}(\mathcal{M} - ( \mathcal{V}  - \overline{ \mathcal{V}_k })d_{ ext{model}})$	$\mathcal{O}ig(rac{\mathcal{M}-( \mathcal{V} +\mathcal{L})d_{ ext{model}}}{N_{ ext{local}}}ig)$	$\checkmark$



# Full Diagram



### CAMBRIDGE 🙆 🧑 🎆 ICLR

# **Full Algorithm**

Algorithm 1 Decoupled Embedding for Pre-Training (DEPT) variants: GLOB TRIM SPEC

**Require:** S: set of K data sources, T: number of rounds **Require:**  $\theta_0$ : initial transformer blocks,  $\phi_0, \psi_0$ : optional token/positional embeddings **Require:**  $\{\mathcal{D}_k\}_{k=1}^K$ : source-specific datasets,  $\{\mathcal{V}_k\}_{k=1}^K$ : source-specific vocabularies Require: InnerOPT: inner optimizer, OuterOPT: outer optimizer, e.g., AdamW and FedAvg 1: for each update round  $t = 1, 2, \ldots, T$  do 2: Randomly select a subset  $S_t \subseteq S$  of data sources for round t3: for each data source  $k \in S_t$  in parallel do  $\theta_t^k, \phi_t^k, \psi_t^k \leftarrow \texttt{InnerOPT}(\theta_{t-1}, \phi_{t-1}, \psi_{t-1}, \mathcal{D}_k)$ 4: ▷ GLOB: Global embeddings 5:  $\phi_{t-1}|_{\mathcal{V}_k} = \operatorname{Trim}(\phi_{t-1}, \mathcal{V}_k)$ ▷ TRIM: Trim global token embeddings  $\theta_t^k, \phi_t|_{\mathcal{V}_k}, \psi_t^k \leftarrow \text{InnerOPT}(\theta_{t-1}, \phi_{t-1}|_{\mathcal{V}_k}, \psi_{t-1}, \mathcal{D}_k)$ 6: ▷ TRIM  $\theta_t^k, \phi_t^k, \psi_t^k \leftarrow \texttt{InnerOPT}(\theta_{t-1}, \phi_{t-1}^k, \psi_{t-1}^k, \mathcal{D}_k)$ 7: ▷ SPEC: specialized embeddings  $\Delta \theta_t^k \leftarrow \theta_t^k - \theta_{t-1}$ 8: ▷ Compute parameter update  $\Delta \phi_t^k \leftarrow \phi_t^k - \phi_{t-1}$ 9: ▷ GLOB: Compute global token embedding update  $\Delta \phi_t |_{\mathcal{V}_k} \leftarrow \phi_t |_{\mathcal{V}_k} - \phi_{t-1} |_{\mathcal{V}_k}$ 10: ▷ TRIM: Compute Trimmed embeddings update 11:  $\Delta \psi_t^k \leftarrow \psi_t^k - \psi_{t-1}$ ▷ GLOB + TRIM: global positional embedding update  $\theta_t \leftarrow \texttt{OuterOPT}(\theta_{t-1}, \{\Delta \theta_t^k\}_{k \in S_t})$ 12: ▷ Apply the updates for the transformer body  $\phi_t \leftarrow \texttt{OuterOPT}(\phi_{t-1}, \{\Delta \phi_t^k\}_{k \in S_t})$ 13: ▷ GLOB: Apply token updates 14:  $\phi_t \leftarrow \texttt{OuterOPT}(\phi_{t-1}, \{\Delta \phi_t |_{\mathcal{V}_k}\}_{k \in S_t})$ ▷ TRIM: Apply token updates  $\psi_t \leftarrow \texttt{OuterOPT}(\psi_{t-1}, \{\Delta \psi_t^k\}_{k \in S_t})$ 15:  $\triangleright$  GLOB + TRIM: Apply position updates 16: return  $\theta_T, \phi_T, \psi_T$ 

### CAMBRIDGE 🙆 🧑 🎆 ICLR



(a) The Pile pre-train, activation norms, 24-block

(b) The Pile pre-train, parameter norms, 24-block

Figure 3: Activations and model norms of STANDARD (STD) training versus DEPT (avg  $\pm$  min/max) for a 350M model trained with identical local hyperparameters—prior to adjusting STD ( $\tau = 0$ ) and STD ( $\tau = 1$ ) (uniform and proportional sampling) to a lower learning rate. The OuterOpt of DEPT introduces regularization effects due to noise-injection (Lin et al., 2020), meta-learning (Nichol et al., 2018) characteristics, which constrain these sources (Zhang et al., 2022) of model divergence.