DEPT: Decoupled Embedding for Pre-Training LMs

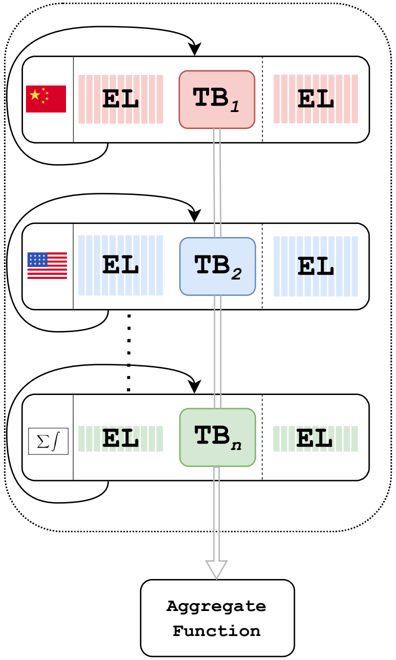
Oral Presentation at ICLR

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

DEPT: Decoupled Embedding Pre-Training

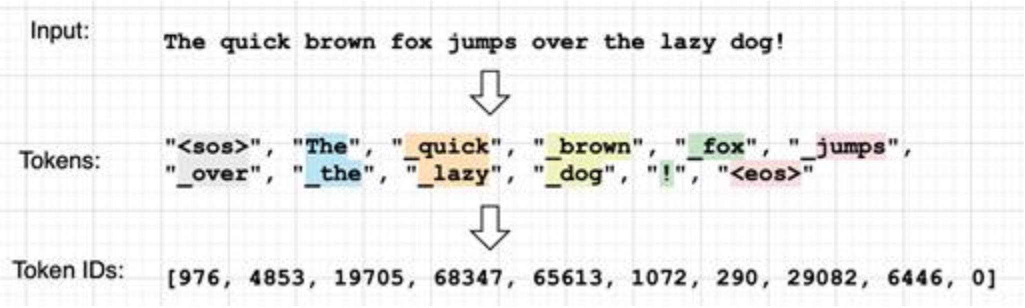
- **Issue:** Shared vocabularies result in **sub-optimal** tokenization and embeddings

- **Solution**: Separate embeddings from the transformer blocks



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




```
def merge_vocab(pair, v_in):
  v_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 = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </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)
```

r ·	→	r·
l o	→	lo
lo w	→	low
e r·	→	er·

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

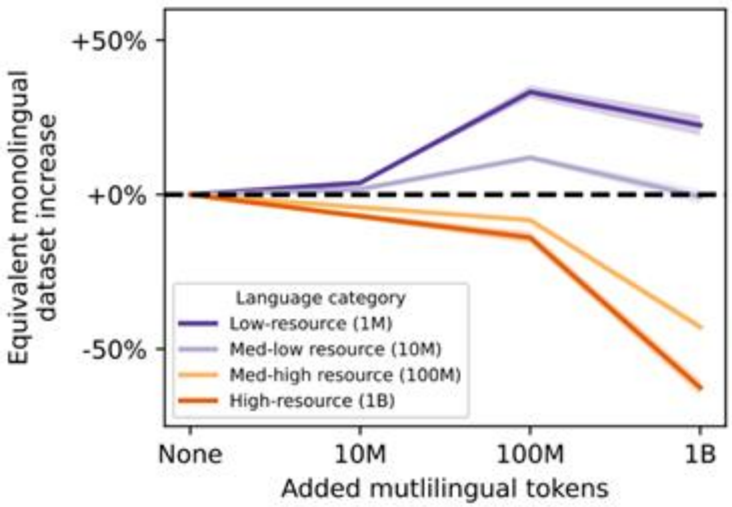
Sennrich, et.al., "Neural Machine Translation of Rare Words with Subword Units"

Data Heterogeneity

- ➢ Languages, mathematics, code vary in vocabulary / syntax / semantics

- ➢ The differences cause
  - ➢ The curse of multilinguality
  - ➢ Negative interference

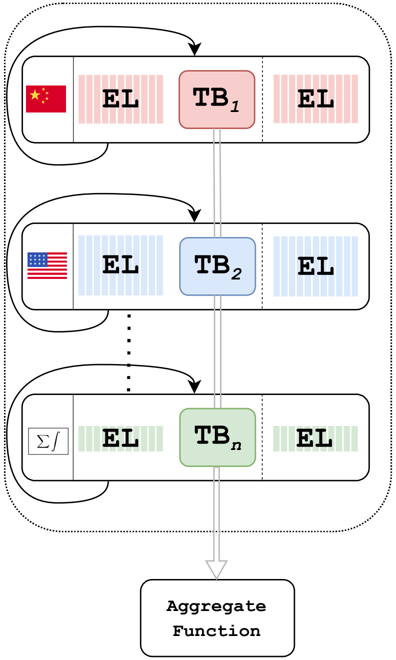
- ➢ Adding more data sources can cause vocabulary dilution + capacity contention



Chang, et.al., "When is multilinguality a curse?"

DEPT: Decoupled Embedding Pre-Training

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



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

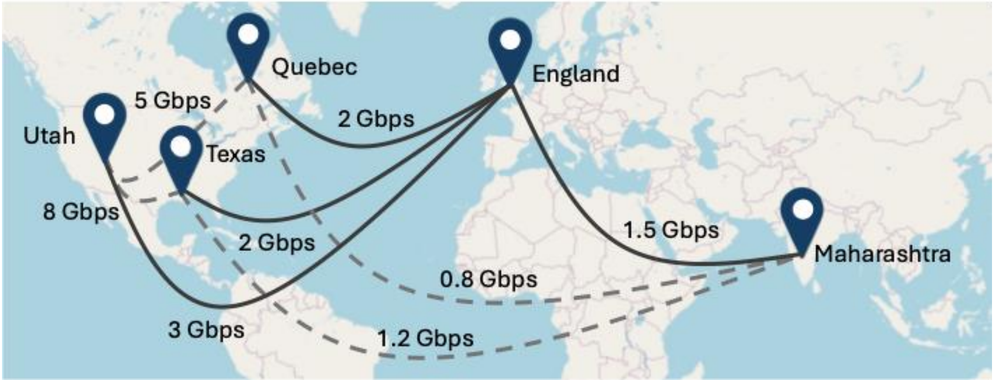
1. Shrinks vocabulary size by manipulating the embedding matrices
2. Avoids training and communicating tokens which are not relevant to a data source

#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

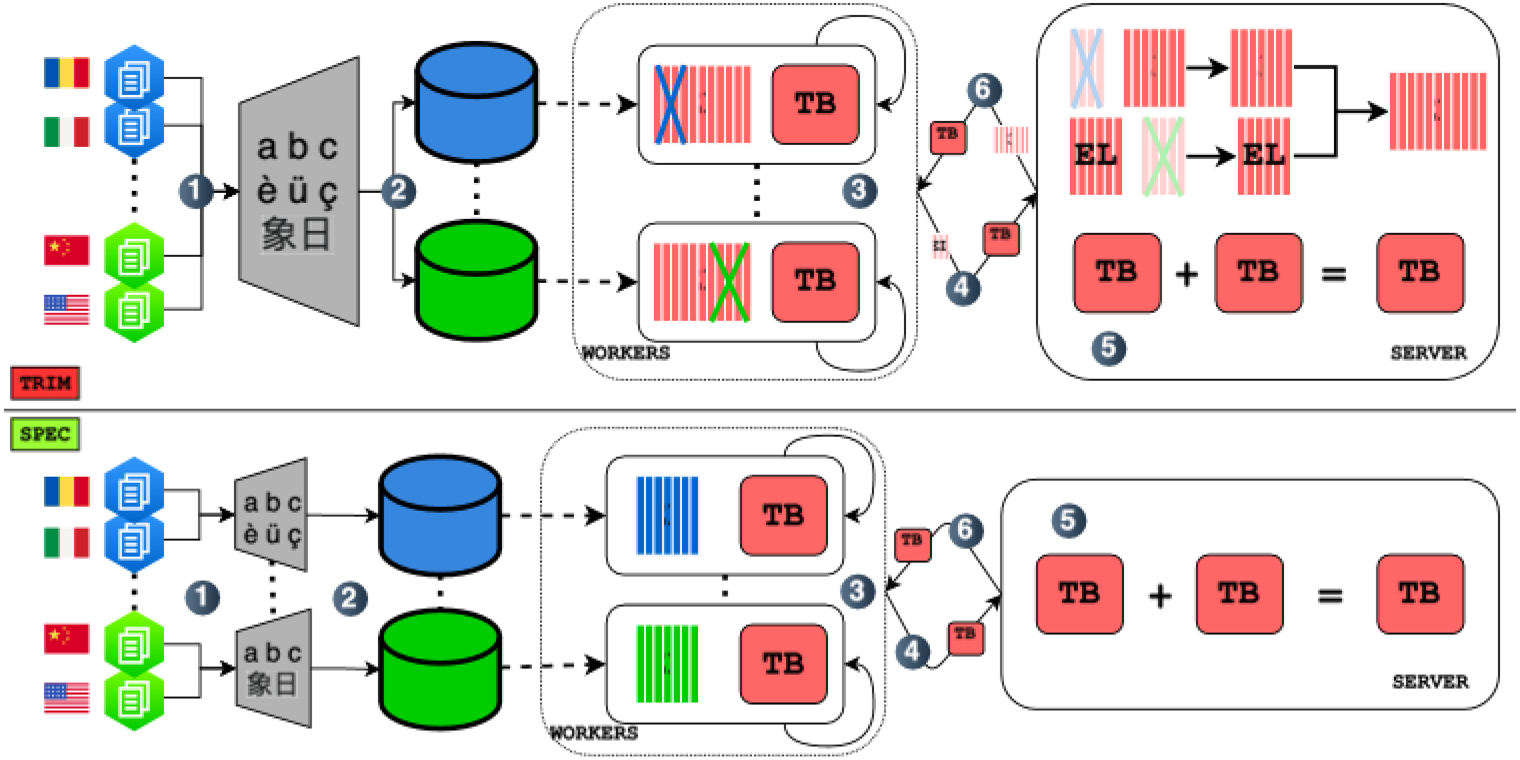
Federated Learning (FL) for Pre-Training

- 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

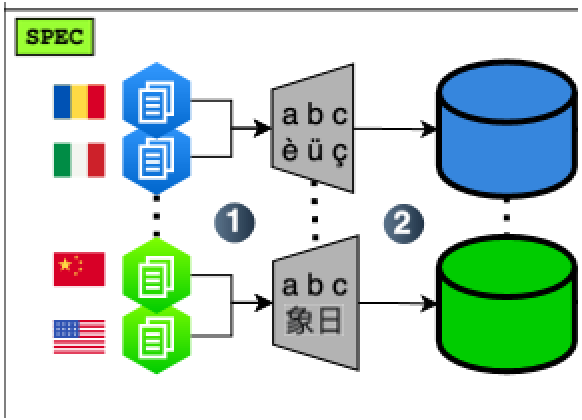


Sani, et.al., "Photon: Federated LLM Pre-Training"

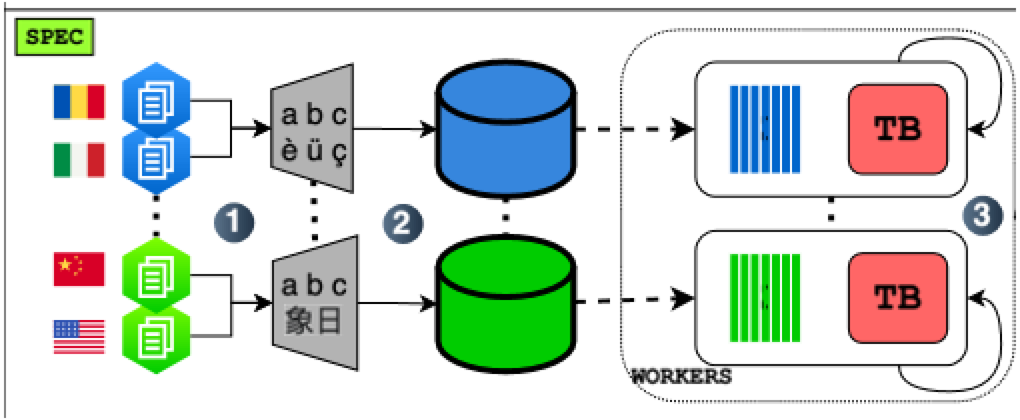
Optimizing the Embedding Layer



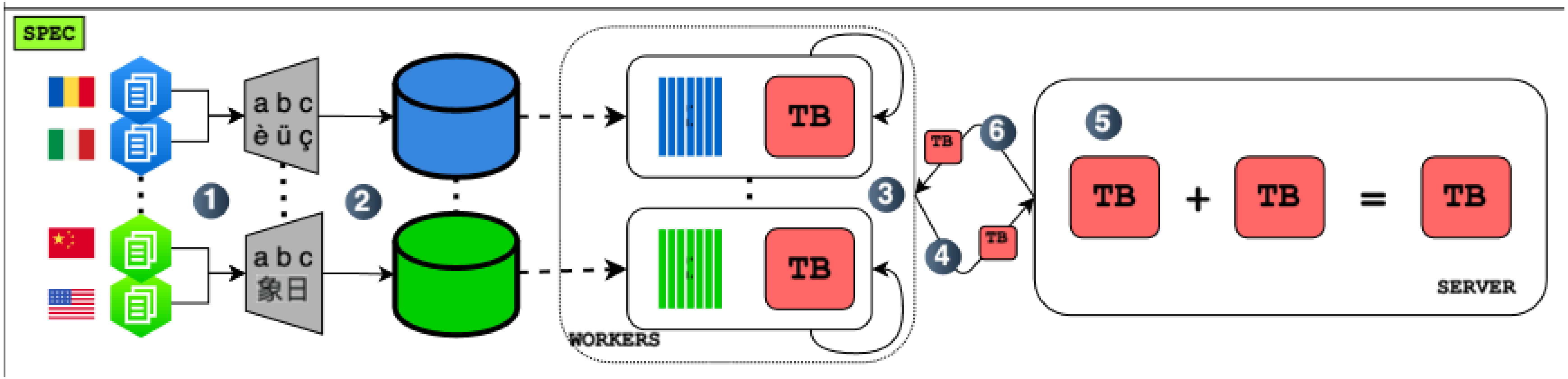
Optimizing the Embedding Layer



Optimizing the Embedding Layer



Optimizing the Embedding Layer



DEPT Results

Method	Embedding Layer Parameters	Total Trainable Parameters	Per-step Comms Cost (↓)
STD	192M	278M (1×)	278M (1×)
GLOB	192M	278M (1×)	0.56M (0.002×)
TRIM	166M	252M (0.91×)	0.5M (0.002×)
SPEC	166M	252M (0.91×)	**0.17M (0.0006×)**
SPEC-OPT	38.6M	**125M (0.45×)**	**0.17M (0.0006×)**
STD	513.2M	1.7B (1×)	1.7B (1×)
SPEC-OPT	102.9M	**1.3B (0.76×)**	**2.4M (0.001×)**

DEPT Improves Comms

Method	Embedding Layer Parameters	Total Trainable Parameters	Per-step Comms Cost (↓)
STD	190M	278M (1×)	278M (1×)
GLOB	190M	278M (1×)	0.56M (0.002×)
TRIB	166M	253M (0.92×)	0.5M (0.002×)
SPEC	166M	253M (0.92×)	0.17M (0.0006×)
SPEC-OPT	38.6M	125M (0.45×)	0.17M (0.0006×)
STD	513.2M	1.71B (1×)	1.71B (1×)
SPEC-OPT	102.9M	1.3B (0.76×)	2.4M (0.001×)

500x

Reduction in Comms (all scales)

DEPT Improves Memory

Method	Embedding Layer Parameters	Total Trainable Parameters	Per-step Comms Cost (↓)
STD	190M	278M (1×)	278M (1×)
GLOB	190M	278M (1×)	0.56M (0.002×)
TRIM	166M	253M (0.92×)	0.5M (0.002×)
SPEC	166M	253M (0.92×)	**0.17M (0.0006×)**
SPEC-OPT	38.6M	**125M (0.45×)**	**0.17M (0.0006×)**
STD	513.3M	1.71B (1×)	1.71B (1×)
SPEC-OPT	102.9M	**1.3B (0.76×)**	**2.4M (0.001×)**

80%

Reduction in Embedding Parameters (at >1B scale)

DEPT SPEC Improves Comms

Method	Embedding Layer Parameters	Total Trainable Parameters	Per-step Comms Cost (↓)
STD	190M	278M (1×)	278M (1×)
GLOB	190M	278M (1×)	0.56M (0.002×)
TRIM	166M	252M (0.92×)	0.5M (0.002×)
SPEC	166M	252M (0.92×)	**0.17M (0.0006×)**
SPEC-OPT	38.6M	**125M (0.45×)**	**0.17M (0.0006×)**
STD	513.2M	1.71B (1×)	1.71B (1×)
SPEC-OPT	102.9M	**1.3B (0.76×)**	**2.4M (0.001×)**

714x

Reduction in Communicated Parameters (>1B scale)

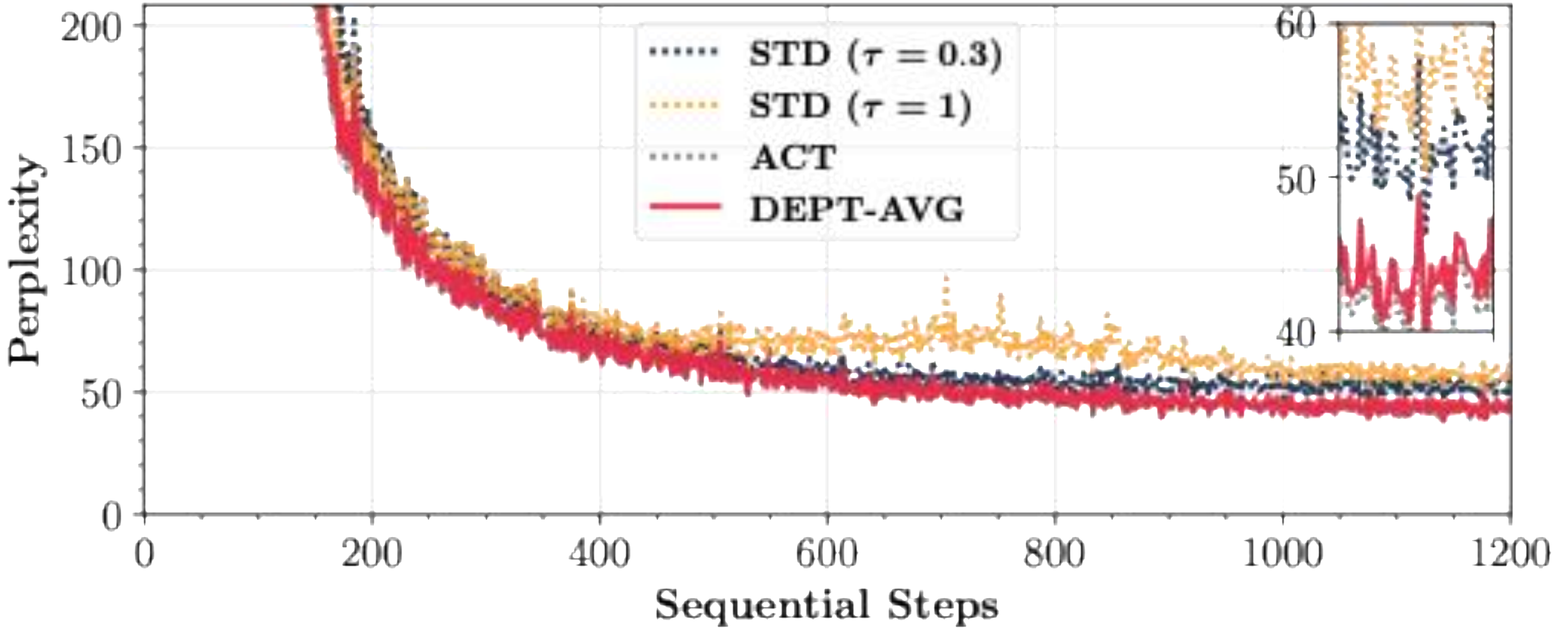
DEPT Improves Downstream Performance

Name	Random Init 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

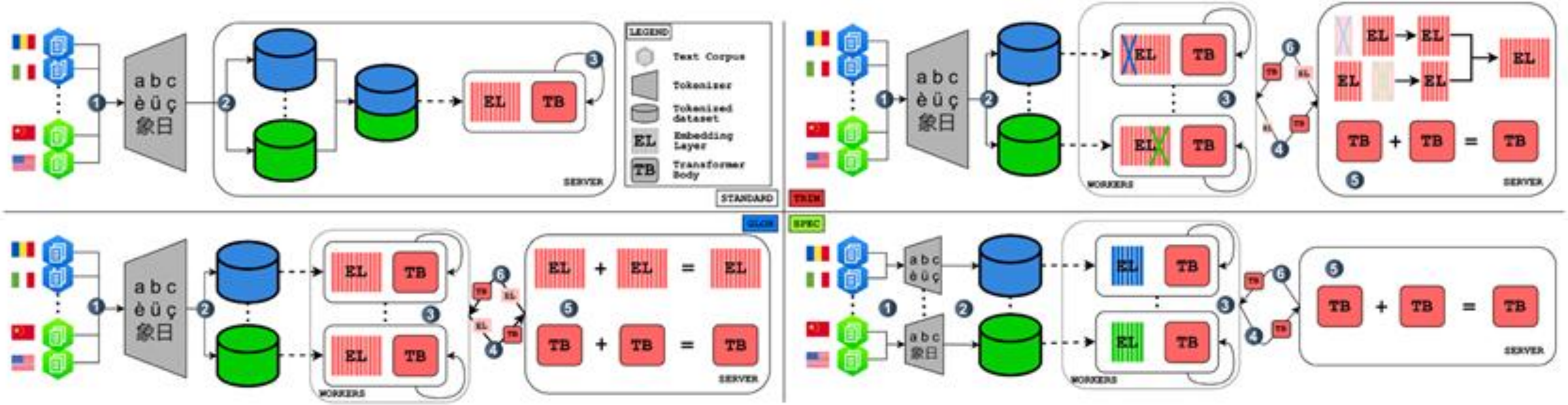
- Produce versatile foundation models that excel across tasks and adapt to new domains

Questions?

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_{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_{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	Memory Cost	Per-step Comms Cost	Vocab Agnostic
STD	$\mathcal{O}(\mathcal{M})$	$\mathcal{O}(\mathcal{M})$	$\times$
GLOB	$\mathcal{O}(\mathcal{M})$	$\mathcal{O}(\frac{\mathcal{M}}{N_{\text{local}}})$	$\times$
TRIM	$\mathcal{O}(\mathcal{M} - (\lvert\mathcal{V}\rvert - \overline{\lvert\mathcal{V}_k\rvert})d_{\text{model}})$	$\mathcal{O}(\frac{\mathcal{M} - (\lvert\mathcal{V}\rvert - \overline{\lvert\mathcal{V}_k\rvert})d_{\text{model}}}{N_{\text{local}}})$	$\times$
SPEC	$\mathcal{O}(\mathcal{M} - (\lvert\mathcal{V}\rvert - \overline{\lvert\mathcal{V}_k\rvert})d_{\text{model}})$	$\mathcal{O}(\frac{\mathcal{M} - (\lvert\mathcal{V}\rvert + \mathcal{L})d_{\text{model}}}{N_{\text{local}}})$	$\checkmark$

Full Diagram

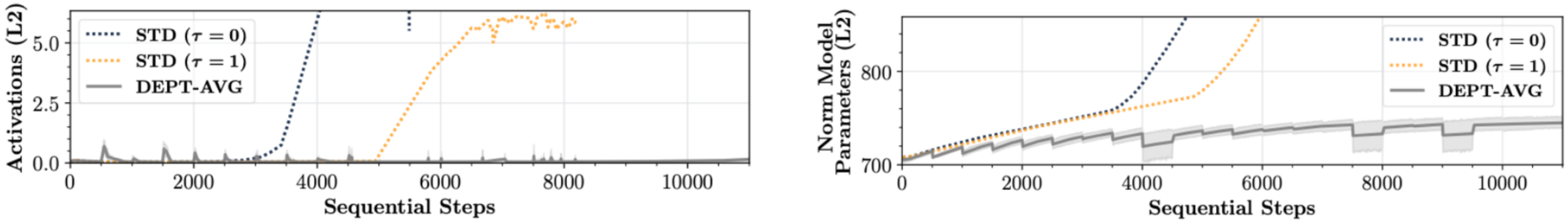


Full Algorithm

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

Require: S: set of K data sources, T: number of rounds
Require: θ_0: initial transformer blocks, ϕ_0, ψ_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

for each update round $t = 1, 2, \ldots, T$ **do**
  Randomly select a subset $S_t \subseteq S$ of data sources for round t
  for each data source $k \in S_t$ **in parallel do**
    $\theta_t^k, \phi_t^k, \psi_t^k \leftarrow \texttt{InnerOPT}(\theta_{t-1}, \phi_{t-1}, \psi_{t-1}, \mathcal{D}_k)$ ▷ `GLOB:` Global embeddings
    $\phi_{t-1}|_{\mathcal{V}_k} = \texttt{Trim}(\phi_{t-1}, \mathcal{V}_k)$ ▷ `TRIM:` Trim global token embeddings
    $\theta_t^k, \phi_t|_{\mathcal{V}_k}, \psi_t^k \leftarrow \texttt{InnerOPT}(\theta_{t-1}, \phi_{t-1}|_{\mathcal{V}_k}, \psi_{t-1}, \mathcal{D}_k)$ ▷ `TRIM`
    $\theta_t^k, \phi_t^k, \psi_t^k \leftarrow \texttt{InnerOPT}(\theta_{t-1}, \phi_{t-1}^k, \psi_{t-1}^k, \mathcal{D}_k)$ ▷ `SPEC:` specialized embeddings
    $\Delta\theta_t^k \leftarrow \theta_t^k - \theta_{t-1}$ ▷ Compute parameter update
    $\Delta\phi_t^k \leftarrow \phi_t^k - \phi_{t-1}$ ▷ `GLOB:` Compute global token embedding update
    $\Delta\phi_t|_{\mathcal{V}_k} \leftarrow \phi_t|_{\mathcal{V}_k} - \phi_{t-1}|_{\mathcal{V}_k}$ ▷ `TRIM:` Compute Trimmed embeddings update
    $\Delta\psi_t^k \leftarrow \psi_t^k - \psi_{t-1}$ ▷ `GLOB`+`TRIM:` global positional embedding update
  $\theta_t \leftarrow \texttt{OuterOPT}(\theta_{t-1}, \{\Delta\theta_t^k\}_{k\in S_t})$ ▷ Apply the updates for the transformer body
  $\phi_t \leftarrow \texttt{OuterOPT}(\phi_{t-1}, \{\Delta\phi_t^k\}_{k\in S_t})$ ▷ `GLOB:` Apply token updates
  $\phi_t \leftarrow \texttt{OuterOPT}(\phi_{t-1}, \{\Delta\phi_t|_{\mathcal{V}_k}\}_{k\in S_t})$ ▷ `TRIM:` Apply token updates
  $\psi_t \leftarrow \texttt{OuterOPT}(\psi_{t-1}, \{\Delta\psi_t^k\}_{k\in S_t})$ ▷ `GLOB`+`TRIM:` Apply position updates
return θ_T, ϕ_T, ψ_T



(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.