Swiss National Science Foundation

Joint Fine-tuning and Conversion of Pretrained Speech and Language Models towards Linear Complexity

Mutian He, Philip N. Garner

arXiv:2410.06846
Code available

idiap RESEARCH INSTITUTE EPFL

Background

- Transformers are so expensive!
  - $O(L^2)$ time complexity
  - $O(L)$ KV cache
- …especially when handing speech
  - few words ≈ 1sec = 16K samples = 50 frames

Background

- The ever-growing arsenal of transformer alternatives
  - Low rank attention: **Linformer**
  - Restricted attention: Longformer, Big Bird, Native Sparse Attention…
  - RNNs (a.k.a. Linear attention): RetNet, RWKV, **Mamba**, DeltaNet …
  - …still increasing!

Motivation

- How to make use of these new models?
  - Pretrained parameters often unavailable, esp. on speech
  - New models emerge rapidly
- Redo the whole pretraining for each new one?
  - Computational costs
  - Access to pretraining data
- Find some way fast & cheap!

Goal

- Convert pretrained transformers into the target model
  - When possible, use only the downstream target task data, avoid re-pretraining
- Retain standard transformer performance

Methods

- Knowledge transfer from original transformer
- Unguided: Parameter transfer
  - Replace attention layers with, e.g. Mamba layers, then fine-tuning
  - Other parameters (e.g. MLPs) are reused
- Guided: Behavior transfer
  - Reproduce the original behavior (hidden states) by layerwise distillation

Method: Cross Architecture Layerwise Distillation

$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{y}^{(s)}, \boldsymbol{y}) = -\sum_i \boldsymbol{y}_i \log(\boldsymbol{y}_i^{(s)})$$

$$\mathcal{L}_{\mathrm{KD}}(\boldsymbol{y}^{(s)}, \boldsymbol{y}^{(t)}) = \sum_i (\frac{\boldsymbol{y}_i^{(t)}}{\beta}) \log\left(\frac{\boldsymbol{y}_i^{(t)}/\beta}{\boldsymbol{y}_i^{(s)}/\beta}\right)$$

$$\mathcal{L}_{\mathrm{LD}}(\boldsymbol{H}^{(s)}, \boldsymbol{H}^{(t)}) = \frac{1}{m}\sum_{i=1}^{m} (\boldsymbol{H}_i^{(s)} - \boldsymbol{H}_i^{(t)})^2$$

$$\mathcal{L} = \alpha_{\mathrm{CE}}\mathcal{L}_{\mathrm{CE}} + \alpha_{\mathrm{KD}}\mathcal{L}_{\mathrm{KD}} + \alpha_{\mathrm{LD}}\mathcal{L}_{\mathrm{LD}}$$

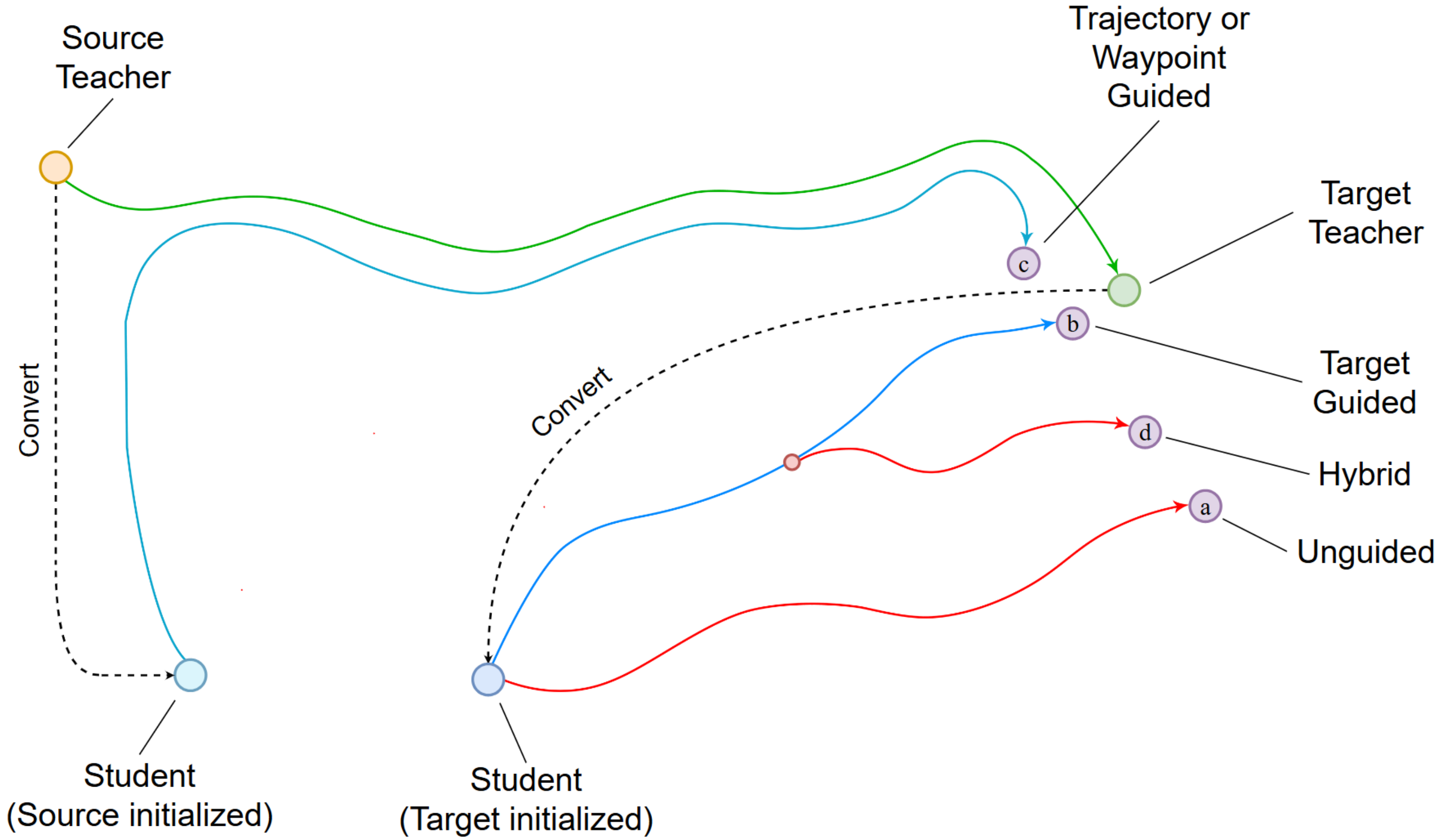
Distillation modes

- Which model should be the teacher?
- Target-guided
  - Using fine-tuned transformer (“target teacher”)
- Trajectory/waypoint guided
  - Original pretrained transformer (“source teacher”) carries important knowledge lost in fine-tuning
  - Can we reproduce the trajectory of transformer fine-tuning?

Distillation modes

- When should we distill?
- Distillation loss terms pose constraint on model training
  - Hybrid: remove distillation loss terms in the late stage of training

Distillation modes



Configuration

- Three sets of experiments considered
  - RoBERTa → Linformer, on NLP tasks: QNLI, QQP, SST2, IMDB
  - Wav2Vec2 → Bidirectional Mamba2, on speech tasks: TEDLIUM (ASR), SLURP (IC), VoxCeleb1 (Speaker ID)
  - Extra: Pythia-1B → Mamba, on zero-shot LM tasks

Empirical results: NLP

Check similar LM and speech results, and trajectory visualization in our paper!

	QNLI	QQP	SST2	IMDB	Average
Pretrained Linformer	91.2%	90.8%	93.1%	94.1%	92.3%
Std. RoBERTa	92.4%	91.8%	95.3%	95.7%	93.8% +1.3
❌ Unguided	53.1%	73.3%	82.6%	82.6%	72.9% -19.4
CALD					
- Target Guided	89.0%	91.8%	93.3%	92.3%	91.6% -0.7
- Src. init.	88.5%	91.7%	93.1%	92.3%	91.4% -0.9
- Trajectory Guided	**91.2%**	**91.9%**	**94.0%**	**93.1%**	**92.5%** +0.2
- Waypoint Guided	89.9%	91.9%	93.7%	92.8%	92.1% -0.2
- Hybrid	86.8%	90.8%	91.4%	90.5%	89.9% -2.4

Performance retained

Better results on speech, see paper for explanation

Takeaway

- Pretrained transformers can be converted to linear-complexity models
  - Guided by distillation only on the target task
- Different modes of distillation may help
  - Guidance from the original transformer fine-tuning trajectory
  - Hybrid of guided and unguided training

arXiv:2410.06846
Code available

THANK YOU