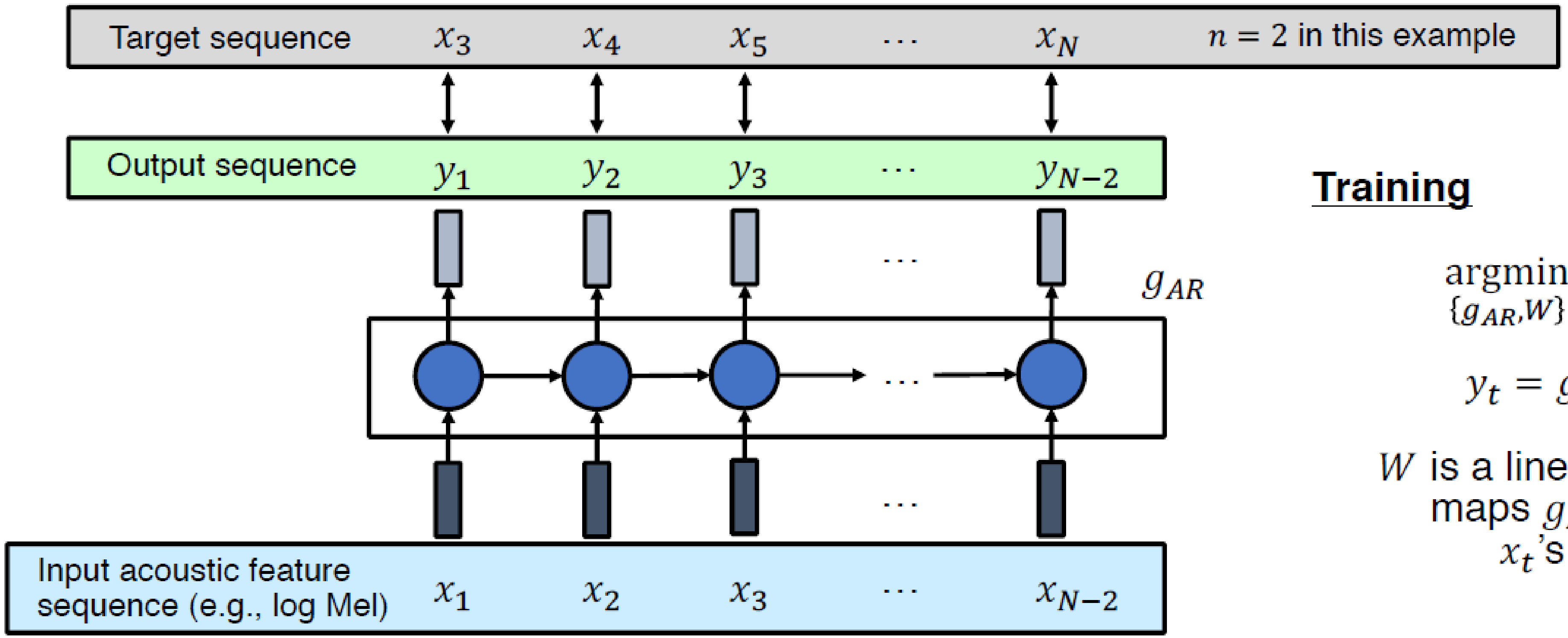
Pretext tasks
4. CPC
|| AR / LPC / VAR / RNN / APC / CPC ||

Autoregressive Predictive Coding (APC)

- Given a previous context $(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_t})$, APC tries to predict a future audio feature $\boldsymbol{x_{t+n}}$ that is $\boldsymbol{n}$ steps ahead of $\boldsymbol{x_t}$
  - Uses an autoregressive model g_{AR} to summarize history and produce output
  - $n \geq 1$ encourages g_{AR} to infer more global underlying structures of the data rather than simply exploiting local smoothness of speech signals



<u>Training</u>

$$\underset{\{g_{AR}, W\}}{\operatorname{argmin}} \ \sum_{t=1}^{N-n} |x_{t+n} - y_t|,$$

$$y_t = g_{AR}(x_1, \ldots, x_t) \cdot W$$

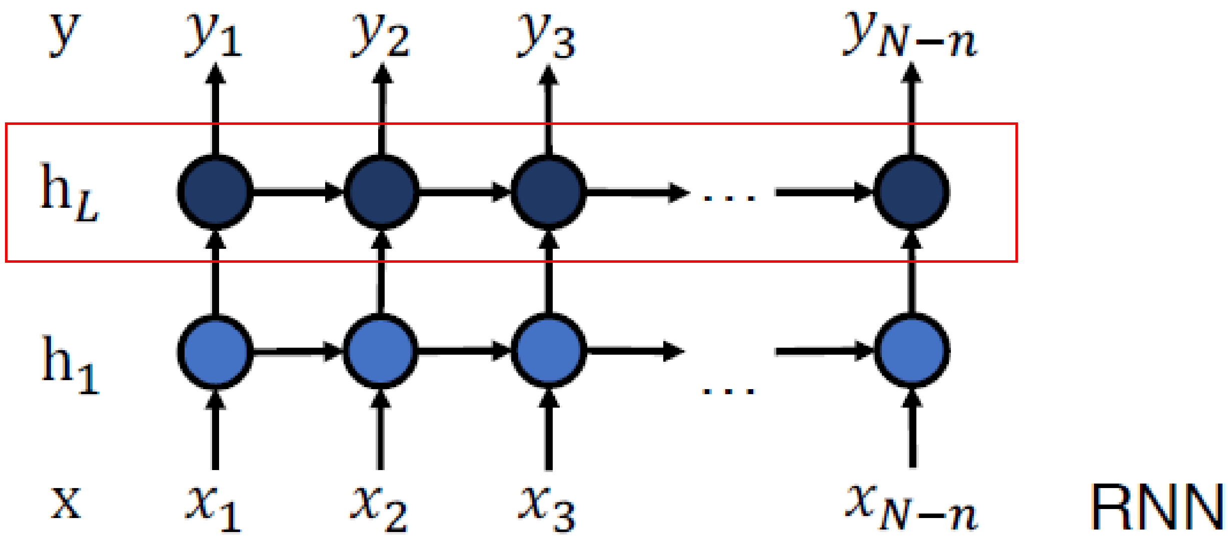
W is a linear transformation that maps g_{AR}'s output back to x_t's dimensionality

Types of autoregressive model g_{AR}

- g_{AR}
  - Input: $\mathbf{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N})$
  - Output: $\mathbf{y} = (\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_N})$

- L-layer Unidirectional RNN:

$$\mathrm{h}_0 = \mathrm{x}$$
$$\mathrm{h}_l = \mathrm{RNN}^{(l)}(h_{l-1}), \forall l \in [1, L]$$
$$\mathrm{y} = \mathrm{h}_L \cdot W$$



- Feature extraction: $\mathbf{h}_L$

Table 1: *Comparing APCs with a series of CPC models on phone classification. PERs are reported.*

Method	#(step)			
	2	5	10	20
`cpc-n9all`	51.3	48.8	50.8	54.6
`cpc-n9same`	47.5	48.2	50.0	53.0
`cpc-ctx-n9same`	42.1	46.1	48.8	53.8
`cpc-ctx-exhaust`	42.9	43.1	45.6	49.1
apc (proposed)	36.5	35.6	35.4	37.7

Table 2: *PERs on phone classification. All features are fed to a linear classifier unless otherwise stated. The number of steps to the target #(steps) is not relevant in the first four rows.*

Method	#(step)					
	1	2	3	5	10	20
Mel	50.0					
Mel + MLP-1	43.4					
Mel + MLP-3	41.3					
cpc best	42.1					
apc 1-layer	39.4	36.5	35.4	35.6	35.4	37.7
apc 2-layer	38.5	34.6	35.9	35.7	34.6	38.8
apc 3-layer	37.2	36.7	33.5	36.1	37.1	38.8
apc 4-layer	36.2	34.4	34.5	35.3	36.9	39.6

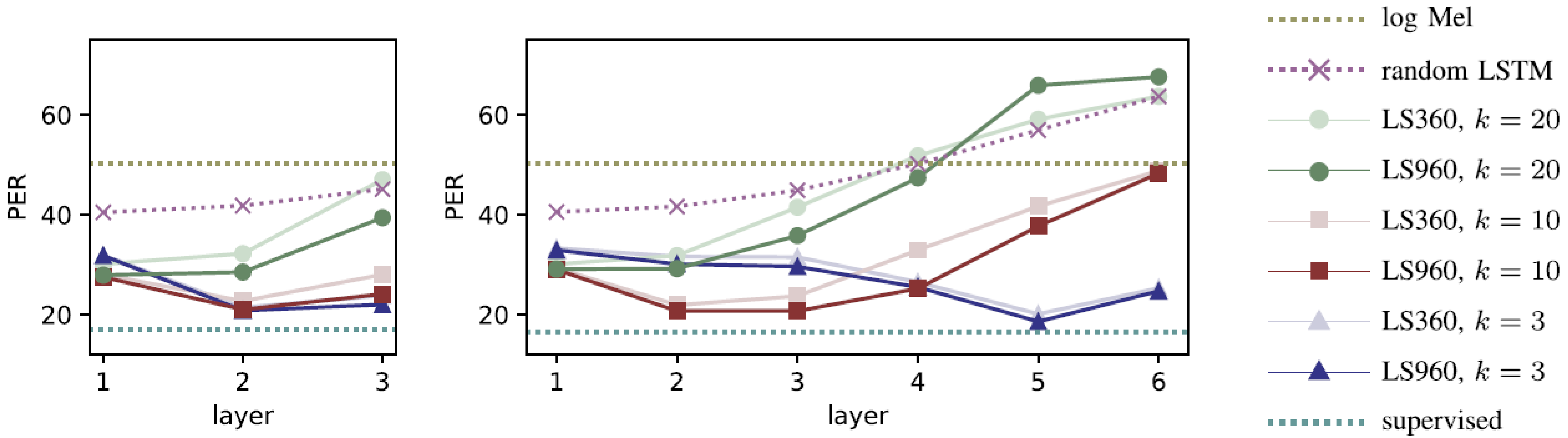


Fig. 1. Phone error rates (PERs) of frame classification on `dev93` with representations produced by 3-layer LSTMs (*left*) and 6-layer LSTMs (*right*). We use LS360 and LS960 to denote the LSTMs trained on the 360-hour subset and the 960 hours combined of LibriSpeech, respectively. We use k to denote the number of time steps into the future in the APC objective.

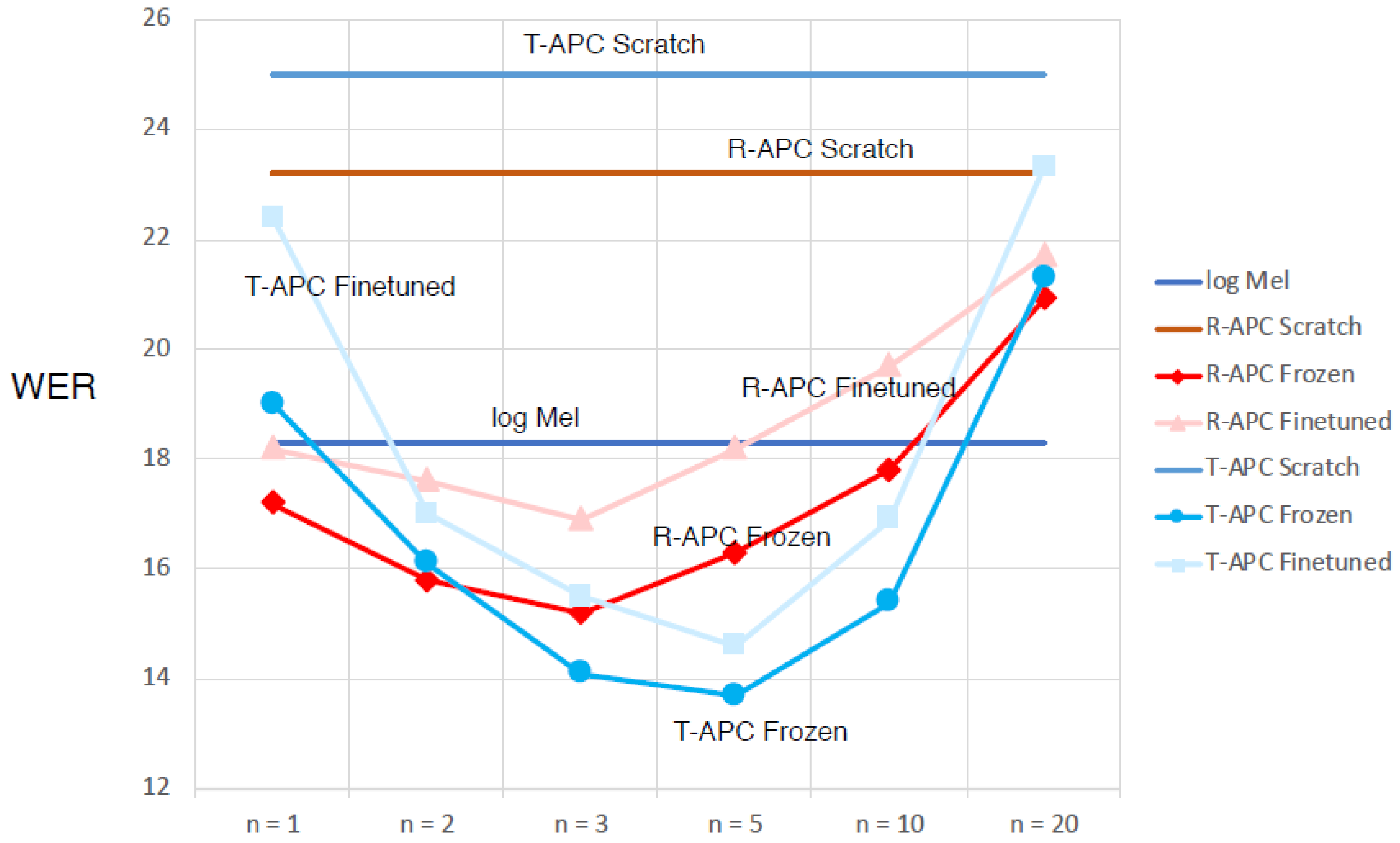
Transfer learning experiments

- Setup: pre-training + fine-tuning
- Pre-training data
  - Speech portion of the LibriSpeech 360 hours subset
  - 921 speakers
  - 80-dimensional log Mel spectrograms as input acoustic features (i.e., $x_t \in \mathbb{R}^{80}$)
  - Use extracted features to replace log Mel as new inputs to downstream models
- Considered downstream tasks
  - Speech recognition
  - Speech translation
  - Speaker identification (skipped in this talk, see paper!)
- Comparing methods
  - Contrastive predictive coding (CPC)
  - Problem-agnostic speech encoder (PASE)

Speech Recognition

- Considered dataset: Wall Street Journal
  - Training: 90% of si284 (~ 72 hours of audio)
  - Validation: 10% of si284
  - Test: dev93
- APC g_{AR}
  - RNNs: 4-layer, 512-dim GRUs
  - Transformers: 4-layer, 512-dim Transformer decoder blocks
- Downstream ASR model
  - Seq2seq with attention [Chorowski et al., 2015]
  - Beam search with beam size = 5
  - No language model rescoring

Choice of $\boldsymbol{n}$, and whether to fine-tune $\boldsymbol{g_{AR}}$



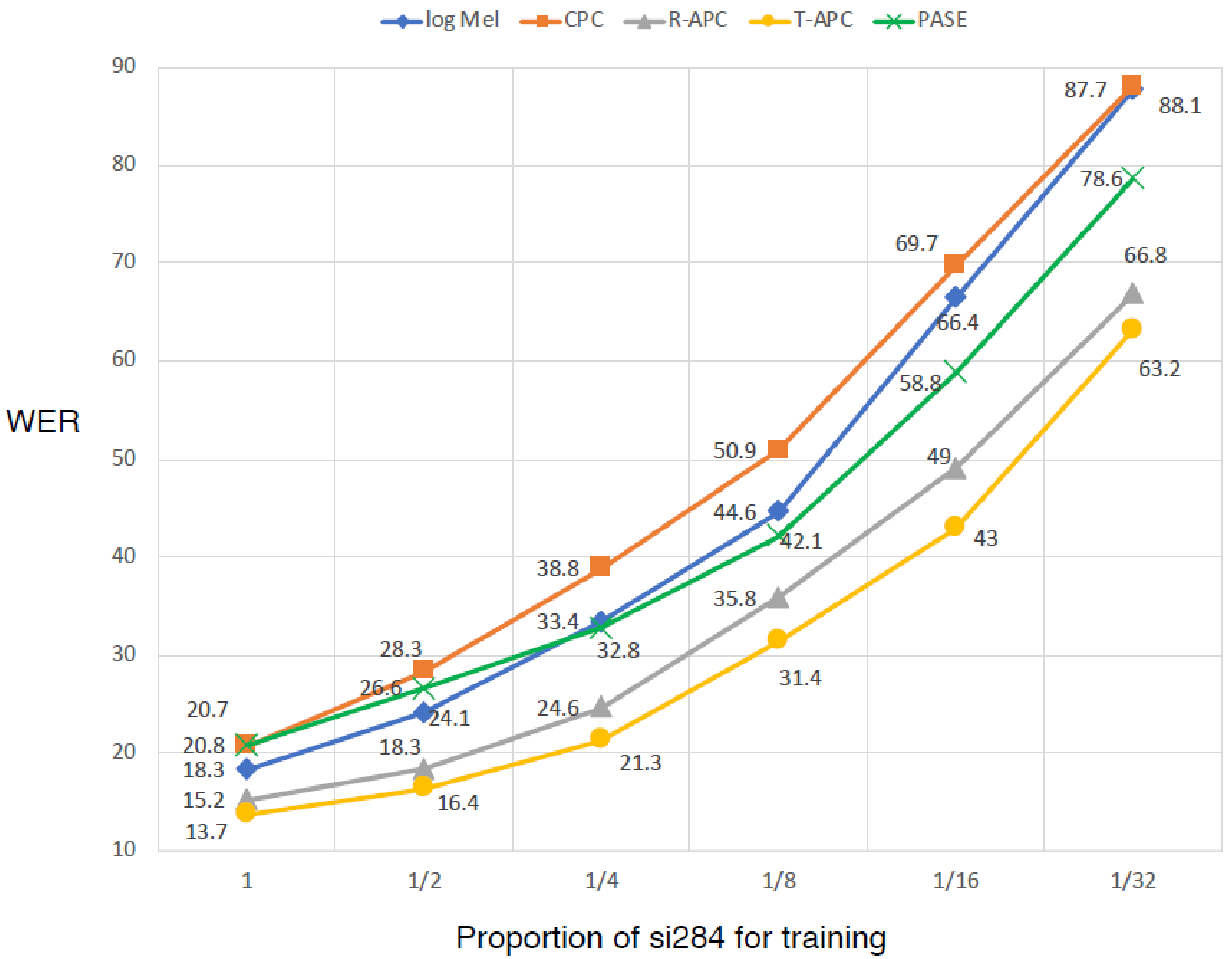
Notations

- R stands for RNN
- T stands for Transformer
- **Scratch**: g_{AR} randomly initialized and concatenate with ASR model
- **Frozen**: keep g_{AR} frozen when training ASR model
- **Finetuned**: fine-tune g_{AR} along with ASR model

Findings

- Sweet spot exists for both Frozen and Finetuned when varying n
- Scratch performance is poor, even worse than log Mel baseline
- APC outperforms log Mel most of the time
- For both R and T, Frozen outperforms Finetuned
- Will use R-APC Frozen with $n = 3$ and T-APC Frozen with $n = 5$ for the rest

APC for reducing the amount of labeled training data

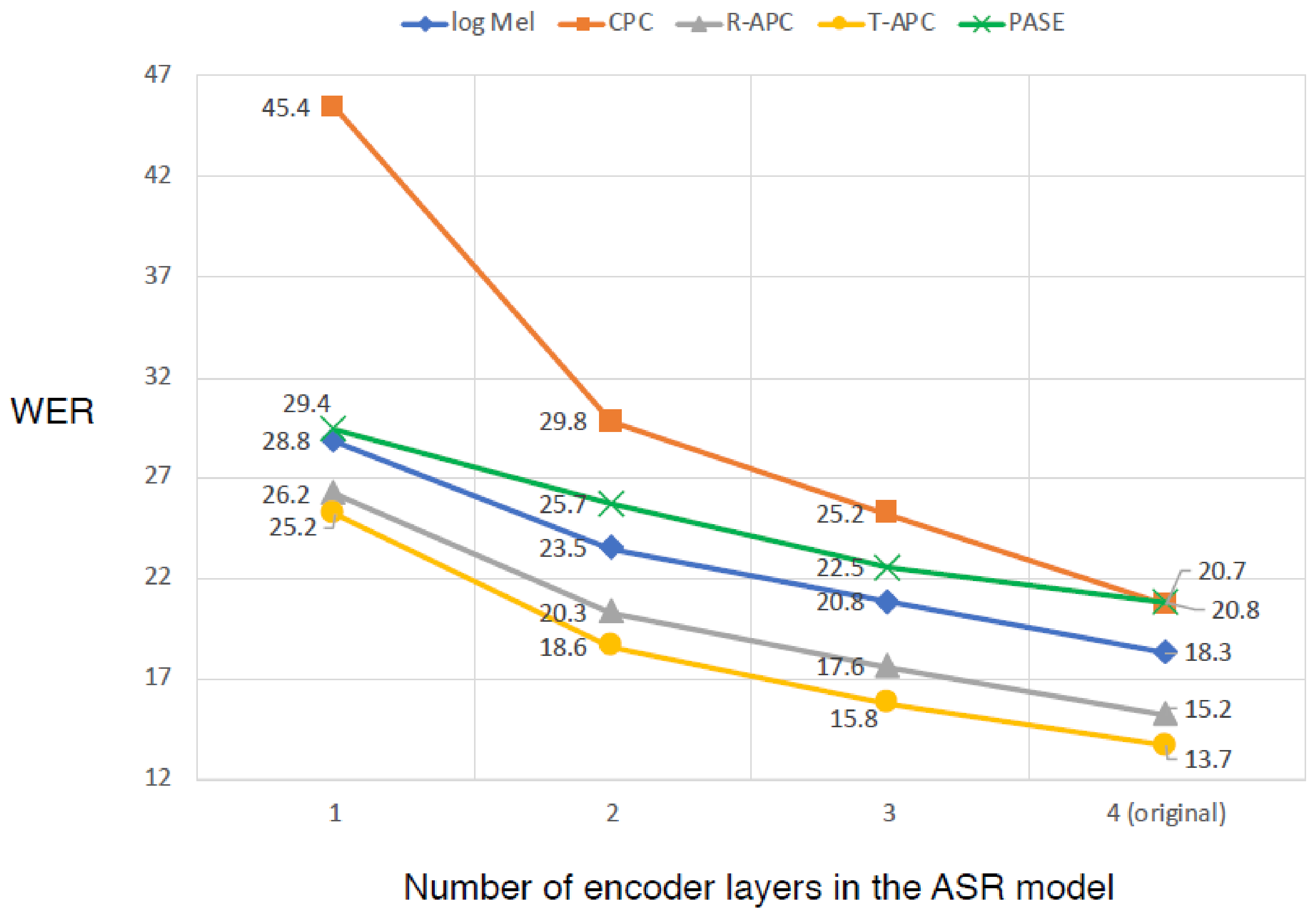


Recap: all feature extractors were pre-trained with 360 hours of LibriSpeech data; we did not fine-tune any feature extractor with the ASR model

Findings

- Full set:
  - 25% and 17% relative improvement for T-APC (13.7) and R-APC (15.2) over log Mel baseline (18.3), respectively
- As we decrease the amount of training data:
  - T-APC (yellow) and R-APC (gray) always outperform other methods
  - Gap between T-APC / R-APC and log Mel (blue) becomes larger
  - Using just half of si284, T-APC (16.4) already outperforms log Mel trained on full set (18.3)
- In the paper we also have the figure where all feature extractors were pre-trained on only 10 hrs of LibriSpeech data. **TLDR**: pre-training still helps even with just 10 hrs of pre-training data

APC for reducing downstream model size



Note: all models trained on full si284

Findings

- T-APC (yellow) and R-APC (gray) always outperform other methods
- T-APC with just 2 layers (18.6) performs similar to log Mel with 4 layers (18.3)

Thank you !!