Reading Group : Semi-Supervised Learning for ASR

Slimipl: Language-Model-Free Iterative Pseudo-Labeling, https://arxiv.org/pdf/2010.11524.pdf
Momentum Pseudo-Labeling for Semi-Supervised Speech Recognition, https://arxiv.org/pdf/2106.08922.pdf
Kaizen: Continuously Improving Teacher Using Exponential Moving Average For Semi-Supervised Speech Recognition, Https://Arxiv.Org/Pdf/2106.07759.Pdf

Wav2Vec-S: Semi-Supervised Pre-Training For Speech Recognition, Https://Arxiv.Org/Pdf/2110.04484.Pdf
Joint Masked Cpc And Ctc Training For Asr, Https://Arxiv.Org/Pdf/2011.00093.Pdf
Joint Unsupervised And Supervised Training For Asr, Https://Arxiv.Org/Abs/2111.08137
Don't Stop Pretraining: Adapt Language Models To Domains And Tasks, https://arxiv.org/abs/2004.10964
Should We Be Pre-Training? An Argument For End-Task Aware Training As An Alternative, https://arxiv.org/abs/2109.07437

Dan Berrebbi

April 27th 2022

Introduction and Setup

- Supervised
- Unsupervised (Self-Supervised)
- Semi-Supervised —> Self Training, Pseudo Labeling (PL)

Data :

- D_l : set of labeled data $\{(X_i, Y_i)$ *for i in* …}
- D_u : set of unlabeled data $\{X_j$ *for j in* …}

Self-training employs a base model trained with labeled data which acts as a “teacher” and is used to label unlabeled data (the resulting labels are referred as “pseudo-labels”). A “student” model is then trained with both labeled and pseudo-labeled data to yield a final model.

Why Semi-Supervised Learning?

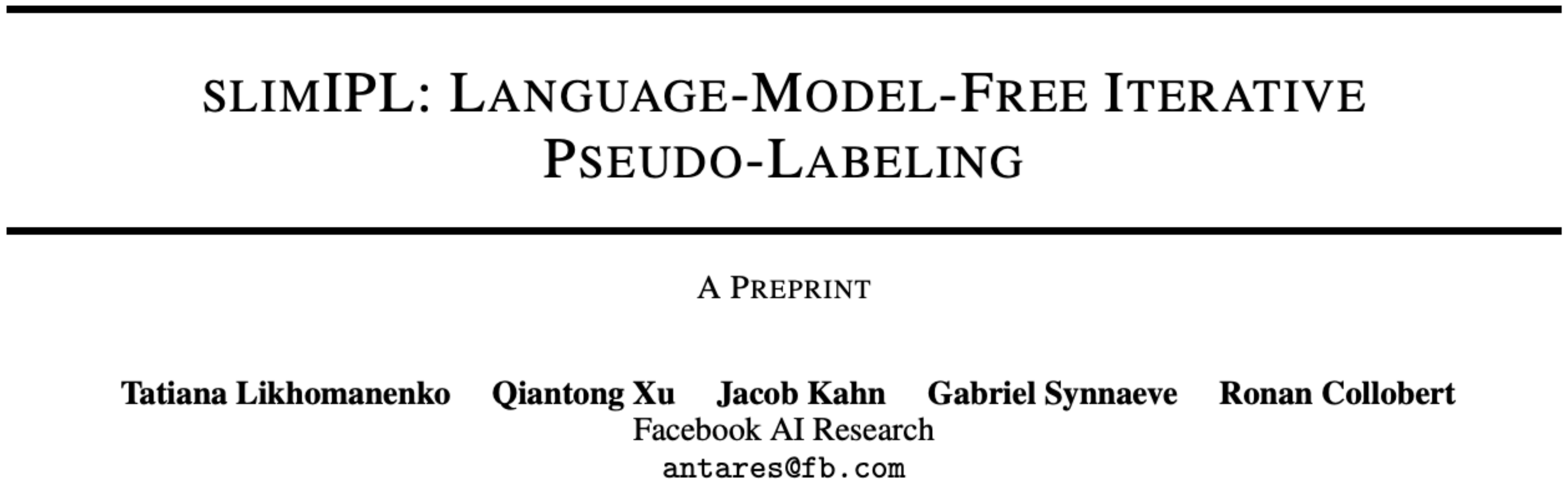
- Supervised : limited amount of labeled data (e2e are hungry)
- Unsupervised (Self-Supervised) : needs lots of data, heuristics …
- Semi-Supervised : can combine both advantages but also limitations

✓We do have labeled data

✓Single stage training

✓Task specific

✓Tuning is easier : early stoping …



SLIMIPL: LANGUAGE-MODEL-FREE ITERATIVE PSEUDO-LABELING

A PREPRINT

Tatiana Likhomanenko **Qiantong Xu** **Jacob Kahn** **Gabriel Synnaeve** **Ronan Collobert**
Facebook AI Research
antares@fb.com

IPL idea :

- Use labeled data to generate pseudo labels for unlabeled data
- Iteratively re-generate new PL as model learns —> to improve the teacher model
- Supervised loss on labeled and pseudo-labeled data

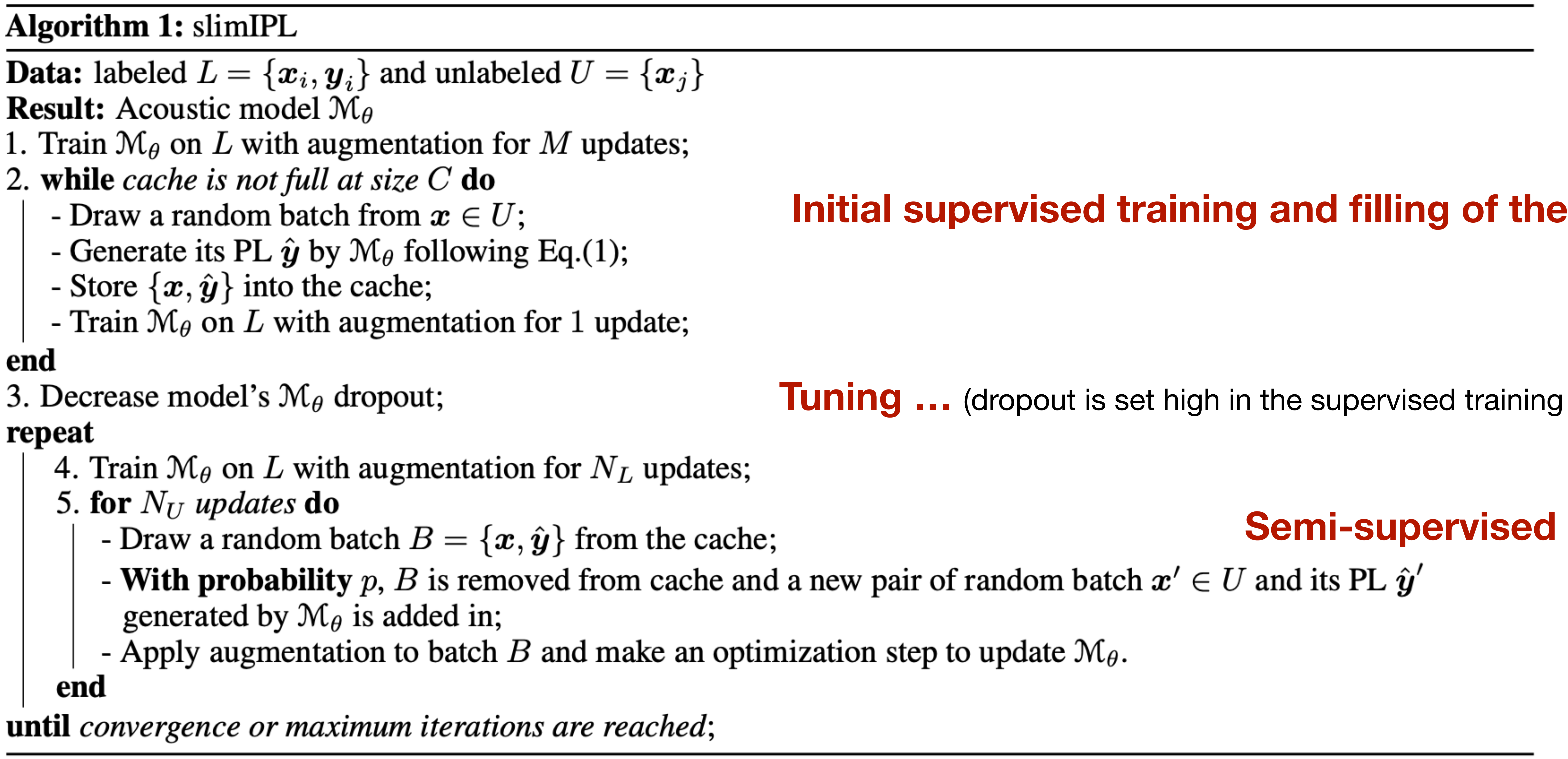
Key differences of this work :

- No beam-search decoding or LM to generate PL (efficiency + overfit to LM)
- Maintain a dynamic cache with PL, not re-generating labels at each iteration (stability)

Pseudo Labeling : CTC loss using argmax : choosing the most likely token at each time step !

Hyperparameters :

- When PL generation begins (…)
- Size of the cache
- Proportion of labeled and unlabeled data
- …



```
Algorithm 1: slimIPL
Data: labeled $L = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ and unlabeled $U = \{\boldsymbol{x}_j\}$
Result: Acoustic model $\mathcal{M}_\theta$
1. Train $\mathcal{M}_\theta$ on $L$ with augmentation for $M$ updates;
2. while cache is not full at size $C$ do
    - Draw a random batch from $\boldsymbol{x} \in U$;
    - Generate its PL $\hat{\boldsymbol{y}}$ by $\mathcal{M}_\theta$ following Eq.(1);
    - Store $\{\boldsymbol{x}, \hat{\boldsymbol{y}}\}$ into the cache;
    - Train $\mathcal{M}_\theta$ on $L$ with augmentation for 1 update;
end
3. Decrease model's $\mathcal{M}_\theta$ dropout;
repeat
    4. Train $\mathcal{M}_\theta$ on $L$ with augmentation for $N_L$ updates;
    5. for $N_U$ updates do
        - Draw a random batch $B = \{\boldsymbol{x}, \hat{\boldsymbol{y}}\}$ from the cache;
        - With probability $p$, $B$ is removed from cache and a new pair of random batch $\boldsymbol{x}' \in U$ and its PL $\hat{\boldsymbol{y}}'$
          generated by $\mathcal{M}_\theta$ is added in;
        - Apply augmentation to batch $B$ and make an optimization step to update $\mathcal{M}_\theta$.
    end
until convergence or maximum iterations are reached;
```

Initial supervised training and filling of the cache

Tuning … (dropout is set high in the supervised training to not overfit to small amount of data)

Semi-supervised Training

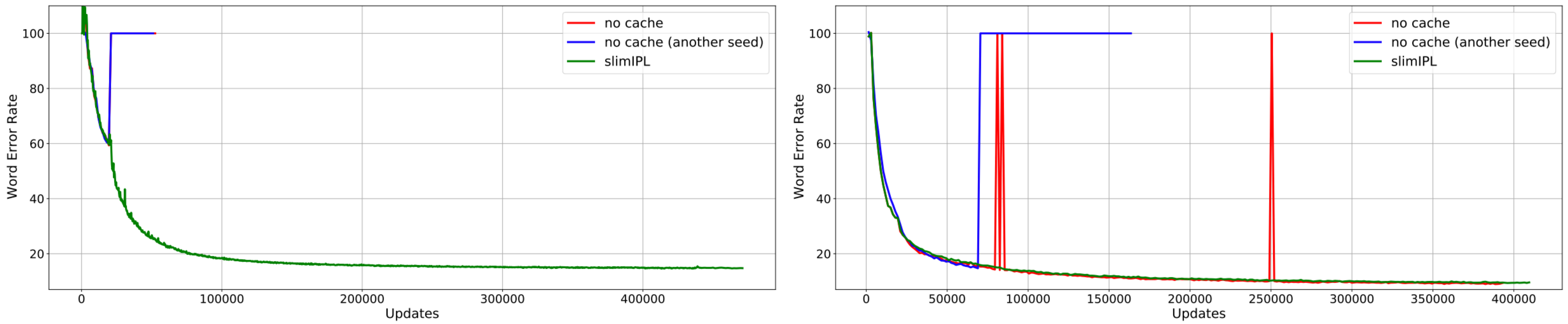
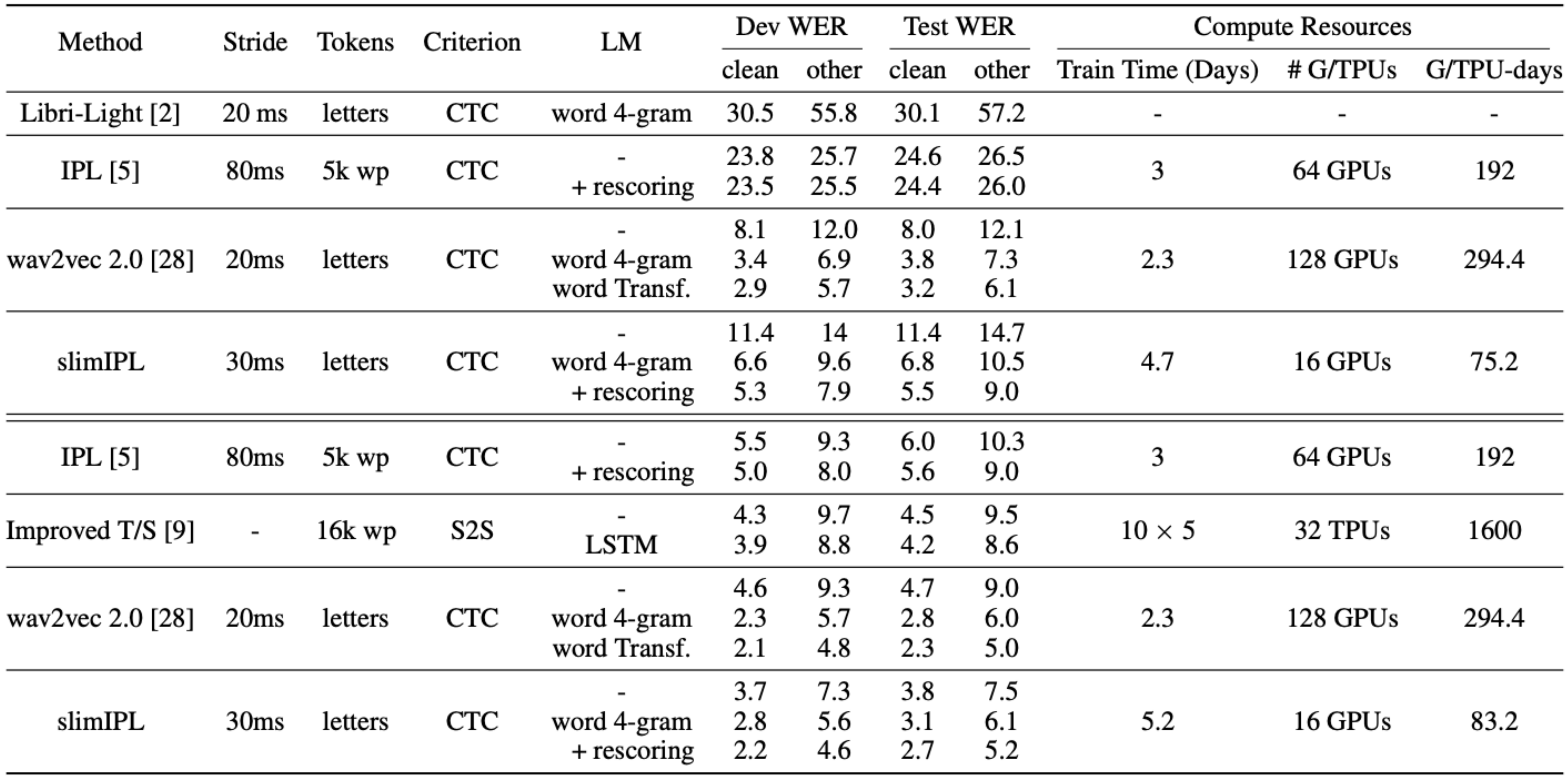


Figure 1: Learning curves on *dev-other* for models trained on LL-10/LS-960 (left) and LS-100/LS-860 (right). slimIPL models refer to baseline models (grey) from Table 3.

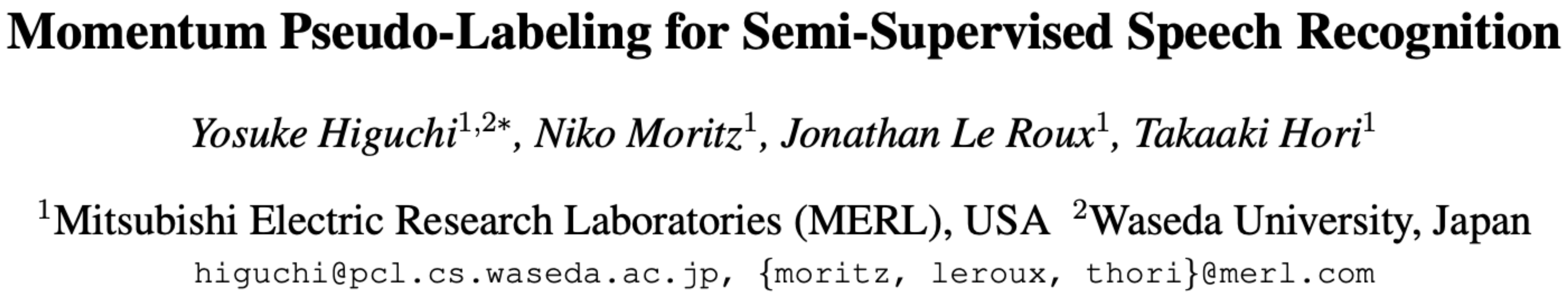
No cache : unstable —> PL become empty sentences

Interesting ablation studies !

Table 2: Comparison with other semi- and unsupervised methods: LL-10/LS-960 (top) and LS-100/LS-860 (bottom).



Method	Stride	Tokens	Criterion	LM	Dev WER		Test WER		Compute Resources		
					clean	other	clean	other	Train Time (Days)	# G/TPUs	G/TPU-days
Libri-Light [2]	20 ms	letters	CTC	word 4-gram	30.5	55.8	30.1	57.2	-	-	-
IPL [5]	80ms	5k wp	CTC	-	23.8	25.7	24.6	26.5	3	64 GPUs	192
				+ rescoring	23.5	25.5	24.4	26.0			
wav2vec 2.0 [28]	20ms	letters	CTC	-	8.1	12.0	8.0	12.1	2.3	128 GPUs	294.4
				word 4-gram	3.4	6.9	3.8	7.3			
				word Transf.	2.9	5.7	3.2	6.1			
slimIPL	30ms	letters	CTC	-	11.4	14	11.4	14.7	4.7	16 GPUs	75.2
				word 4-gram	6.6	9.6	6.8	10.5			
				+ rescoring	5.3	7.9	5.5	9.0			
IPL [5]	80ms	5k wp	CTC	-	5.5	9.3	6.0	10.3	3	64 GPUs	192
				+ rescoring	5.0	8.0	5.6	9.0			
Improved T/S [9]	-	16k wp	S2S	-	4.3	9.7	4.5	9.5	10 × 5	32 TPUs	1600
				LSTM	3.9	8.8	4.2	8.6			
wav2vec 2.0 [28]	20ms	letters	CTC	-	4.6	9.3	4.7	9.0	2.3	128 GPUs	294.4
				word 4-gram	2.3	5.7	2.8	6.0			
				word Transf.	2.1	4.8	2.3	5.0			
slimIPL	30ms	letters	CTC	-	3.7	7.3	3.8	7.5	5.2	16 GPUs	83.2
				word 4-gram	2.8	5.6	3.1	6.1			
				+ rescoring	2.2	4.6	2.7	5.2			



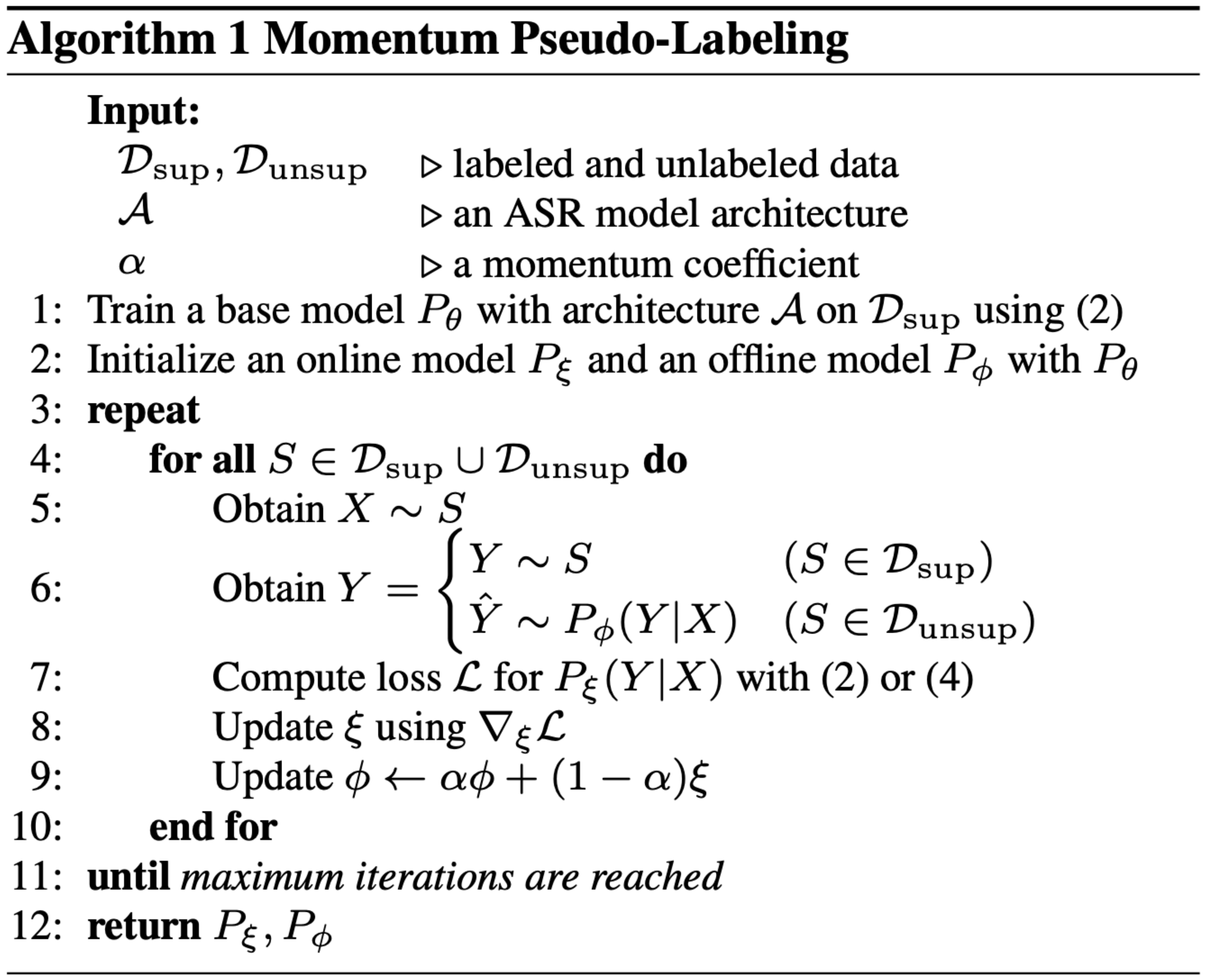
Momentum Pseudo-Labeling for Semi-Supervised Speech Recognition

Yosuke Higuchi[1,2*], *Niko Moritz*[1], *Jonathan Le Roux*[1], *Takaaki Hori*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), USA [2]Waseda University, Japan

higuchi@pcl.cs.waseda.ac.jp, {moritz, leroux, thori}@merl.com

- Train a supervised model
- Initialize 2 models (teacher - student) with the supervised model
- Teacher generates PL, and is updated with momentum
- Student is « classically » trained (data augmentation is used)



Algorithm 1 Momentum Pseudo-Labeling

```
Input:
  $\mathcal{D}_{\mathrm{sup}}, \mathcal{D}_{\mathrm{unsup}}$   ▷ labeled and unlabeled data
  $\mathcal{A}$                                        ▷ an ASR model architecture
  $\alpha$                                             ▷ a momentum coefficient
Train a base model $P_\theta$ with architecture $\mathcal{A}$ on $\mathcal{D}_{\mathrm{sup}}$ using (2)
Initialize an online model $P_\xi$ and an offline model $P_\phi$ with $P_\theta$
repeat
    for all $S \in \mathcal{D}_{\mathrm{sup}} \cup \mathcal{D}_{\mathrm{unsup}}$ do
        Obtain $X \sim S$
        Obtain $Y = \begin{cases} Y \sim S & (S \in \mathcal{D}_{\mathrm{sup}}) \\ \hat{Y} \sim P_\phi(Y|X) & (S \in \mathcal{D}_{\mathrm{unsup}}) \end{cases}$
        Compute loss $\mathcal{L}$ for $P_\xi(Y|X)$ with (2) or (4)
        Update $\xi$ using $\nabla_\xi \mathcal{L}$
        Update $\phi \leftarrow \alpha\phi + (1-\alpha)\xi$
    end for
until maximum iterations are reached
return $P_\xi, P_\phi$
```

$$\phi^{(K)} = \alpha^K \phi^{(0)} + (1-\alpha) \sum_{k=1}^{K} \alpha^{K-k} \xi^{(k)}, \qquad (6)$$

- More sensitive to alpha when domain mismatch (TED)
- No LM or Beam Search
- Alpha = 0 —> IPL, not stable
- Alpha enables to understand and control the model updates

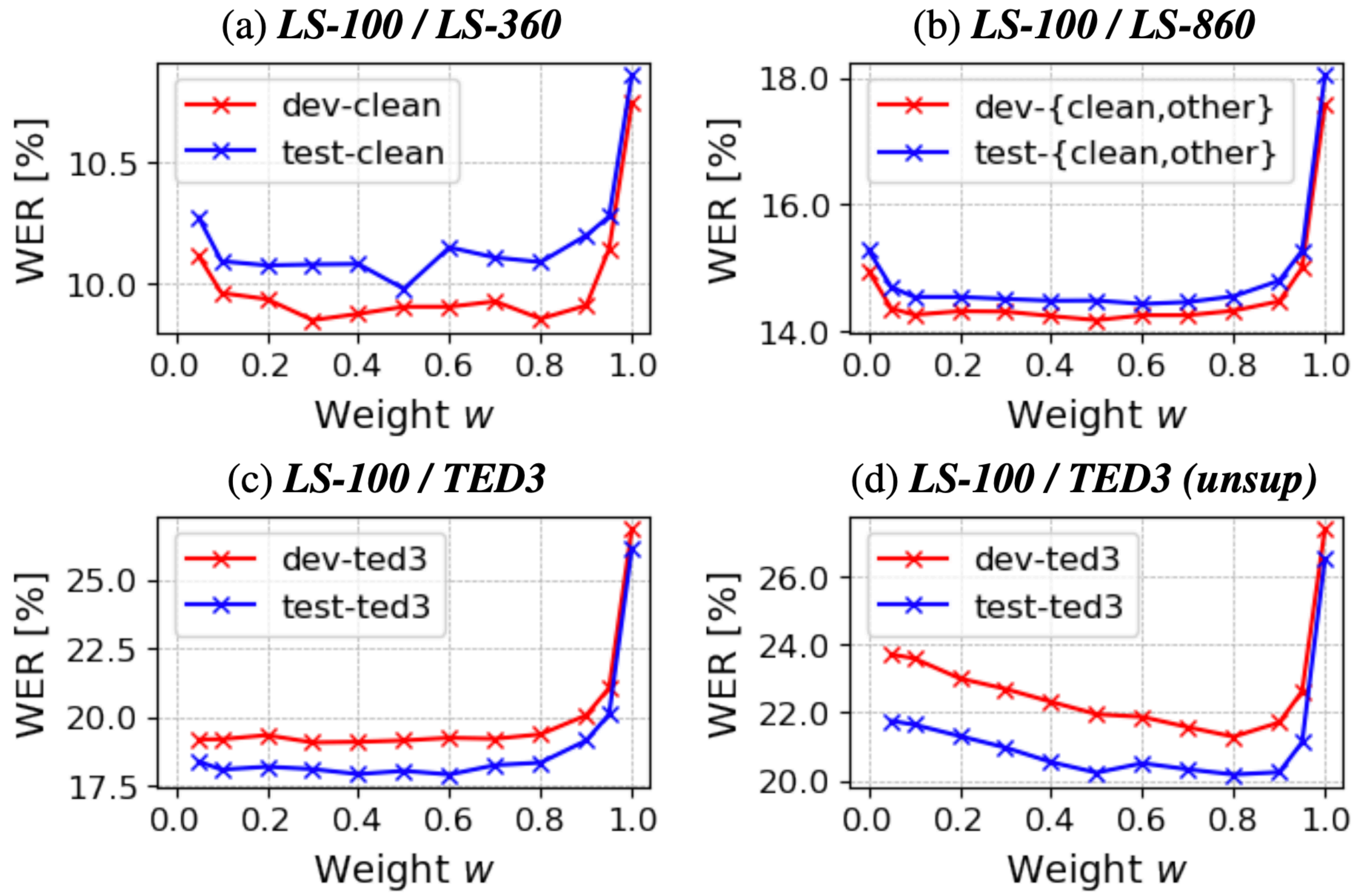
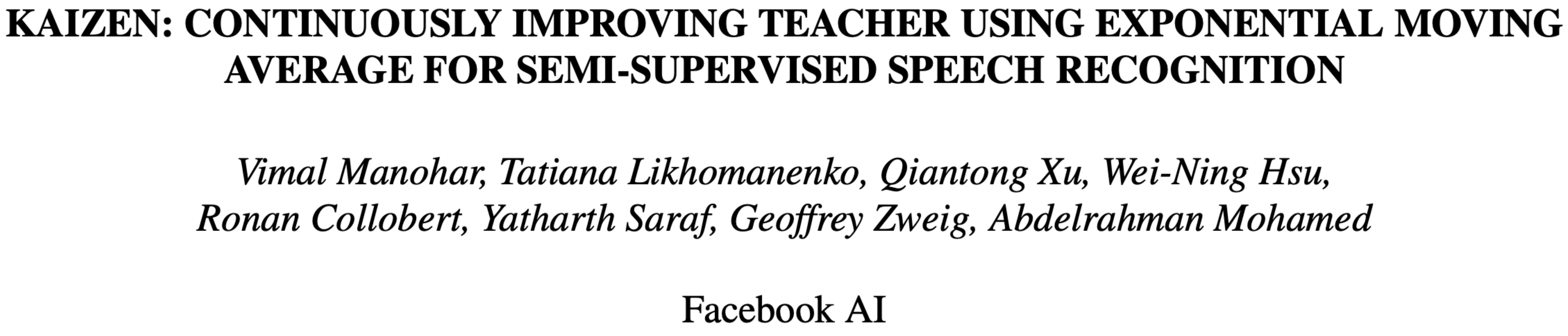


Figure 1: *Influence of momentum update weight w on WER.*

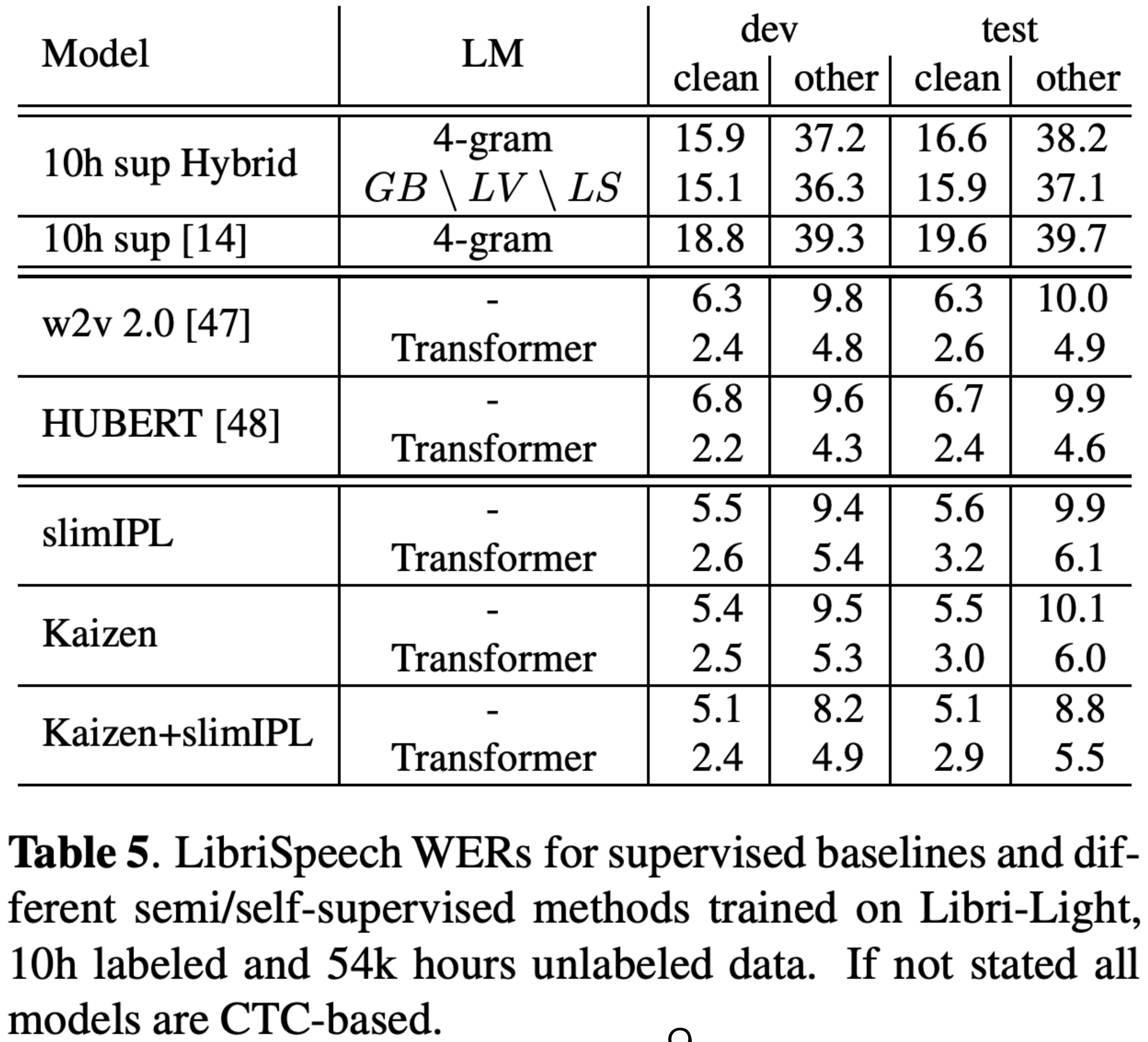


KAIZEN: CONTINUOUSLY IMPROVING TEACHER USING EXPONENTIAL MOVING AVERAGE FOR SEMI-SUPERVISED SPEECH RECOGNITION

Vimal Manohar, Tatiana Likhomanenko, Qiantong Xu, Wei-Ning Hsu,
Ronan Collobert, Yatharth Saraf, Geoffrey Zweig, Abdelrahman Mohamed

Facebook AI

- Tried in a bit more setups (10h supervised data only …)
- « Half-precision floating point training »
- « The Kaizen framework can be seen as a continuous version of the iterative pseudo-labeling approach for semi-supervised training. »



Model	LM	dev		test	
		clean	other	clean	other
10h sup Hybrid	4-gram	15.9	37.2	16.6	38.2
	$GB \setminus LV \setminus LS$	15.1	36.3	15.9	37.1
10h sup [14]	4-gram	18.8	39.3	19.6	39.7
w2v 2.0 [47]	-	6.3	9.8	6.3	10.0
	Transformer	2.4	4.8	2.6	4.9
HUBERT [48]	-	6.8	9.6	6.7	9.9
	Transformer	2.2	4.3	2.4	4.6
slimIPL	-	5.5	9.4	5.6	9.9
	Transformer	2.6	5.4	3.2	6.1
Kaizen	-	5.4	9.5	5.5	10.1
	Transformer	2.5	5.3	3.0	6.0
Kaizen+slimIPL	-	5.1	8.2	5.1	8.8
	Transformer	2.4	4.9	2.9	5.5

Table 5. LibriSpeech WERs for supervised baselines and different semi/self-supervised methods trained on Libri-Light, 10h labeled and 54k hours unlabeled data. If not stated all models are CTC-based.

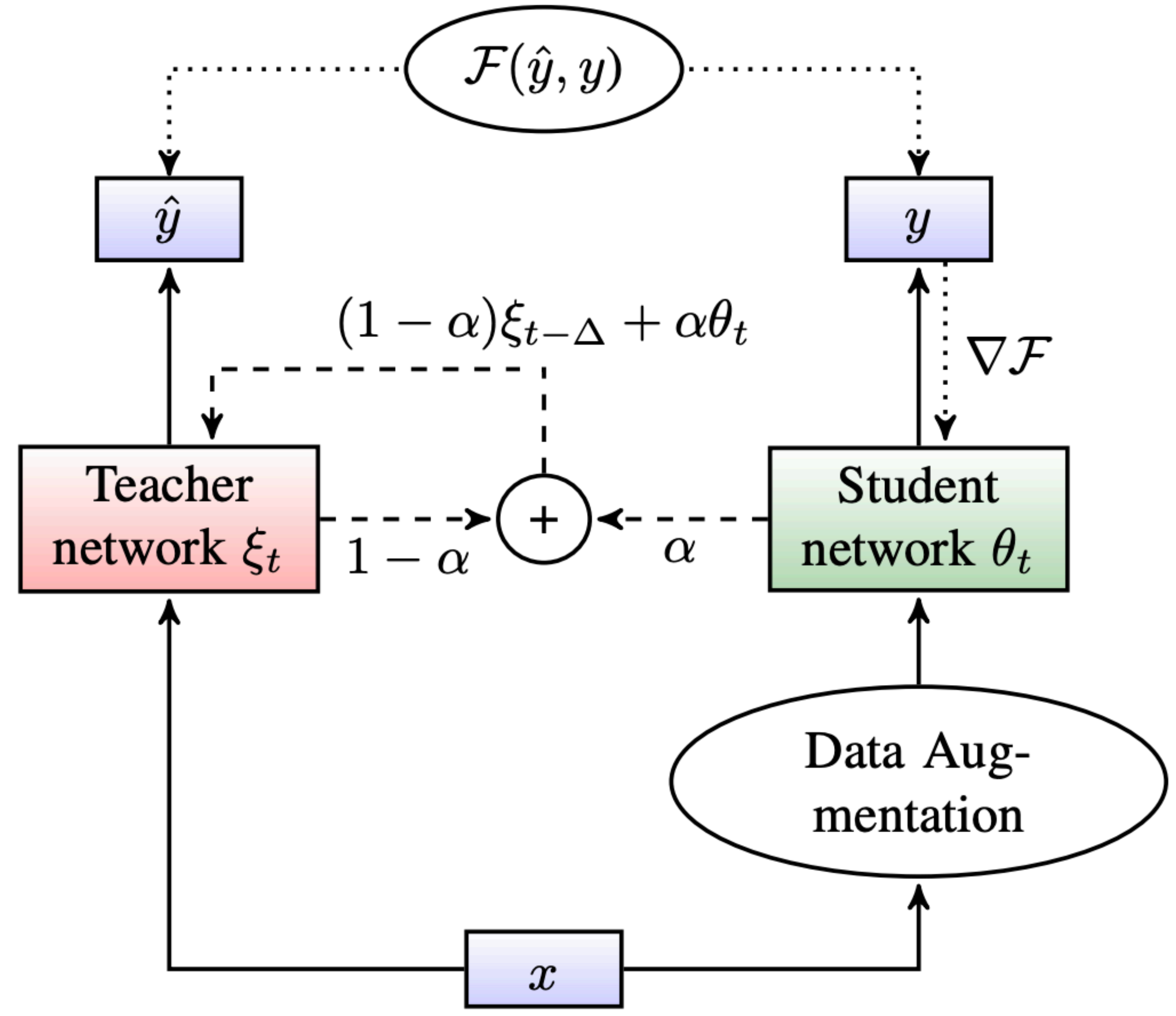
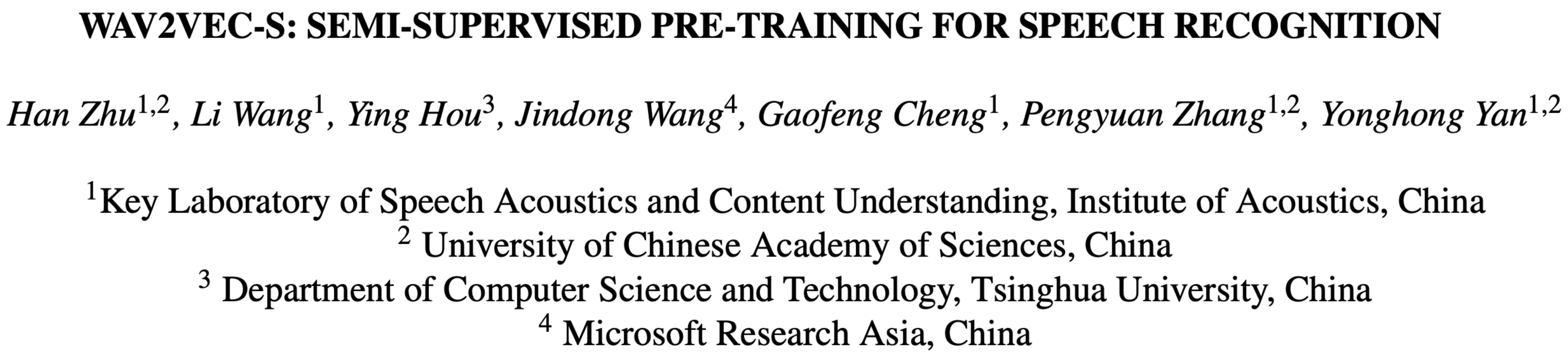

$\mathcal{F}(\hat{y}, y)$
$\hat{y}$
y
$(1-\alpha)\xi_{t-\Delta} + \alpha\theta_t$
$\nabla\mathcal{F}$
Teacher network ξ_t
$1-\alpha$
+
α
Student network θ_t
Data Augmentation
x

Fig. 1. Block diagram of the Kaizen framework.

Mixing Semi-Supervised approaches and SSL



WAV2VEC-S: SEMI-SUPERVISED PRE-TRAINING FOR SPEECH RECOGNITION

Han Zhu[1,2], *Li Wang*[1], *Ying Hou*[3], *Jindong Wang*[4], *Gaofeng Cheng*[1], *Pengyuan Zhang*[1,2], *Yonghong Yan*[1,2]

[1]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, China
[2] University of Chinese Academy of Sciences, China
[3] Department of Computer Science and Technology, Tsinghua University, China
[4] Microsoft Research Asia, China

There is a gap between the task-agnostic pre-training and the task-specific downstream fine-tuning, which may degrade the downstream performance. **—> task-specific semi-supervised pre-training to bridge this gap.**

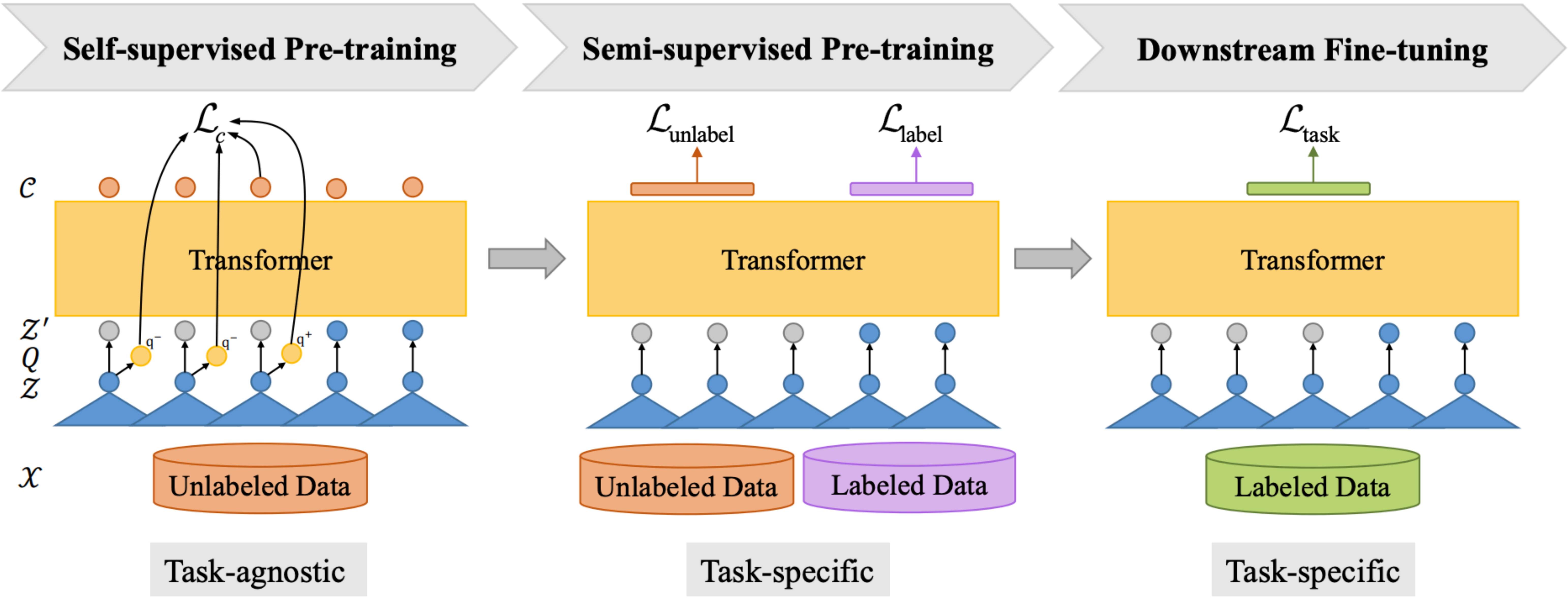


Fig. 1. Illustration of the wav2vec-S procedure.

$$\mathcal{L}_{\text{semi}} = \mathcal{L}_{\text{label}} + \lambda \mathcal{L}_{\text{unlabel}},$$

Loss used : CTC with argmax
(best token at each time step)
They used momentum, or SlimIPL
no precise mention.

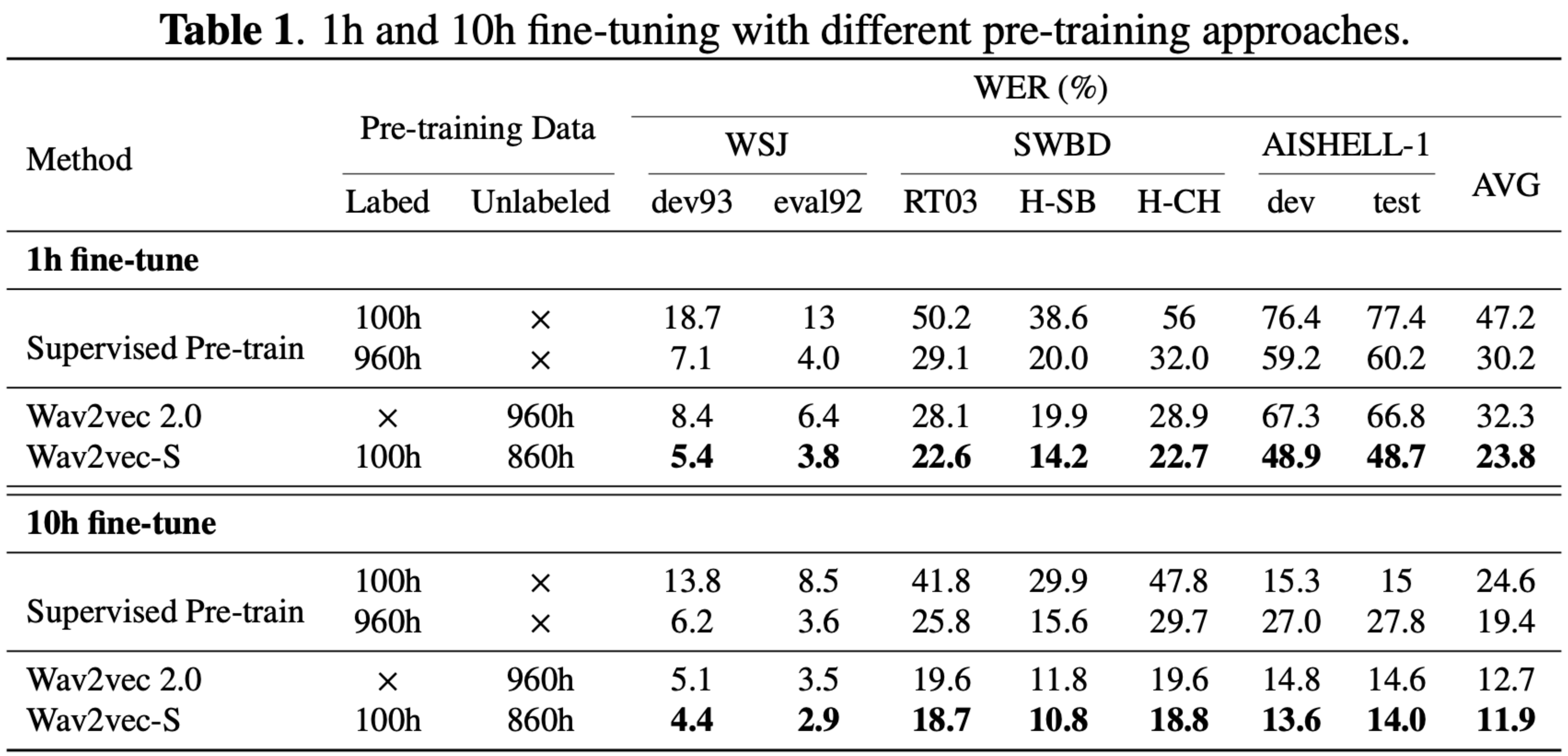


Table 1. 1h and 10h fine-tuning with different pre-training approaches.

Method	Pre-training Data		WER (%)							
			WSJ		SWBD			AISHELL-1		AVG
	Labed	Unlabeled	dev93	eval92	RT03	H-SB	H-CH	dev	test	
1h fine-tune										
Supervised Pre-train	100h	×	18.7	13	50.2	38.6	56	76.4	77.4	47.2
	960h	×	7.1	4.0	29.1	20.0	32.0	59.2	60.2	30.2
Wav2vec 2.0	×	960h	8.4	6.4	28.1	19.9	28.9	67.3	66.8	32.3
Wav2vec-S	100h	860h	**5.4**	**3.8**	**22.6**	**14.2**	**22.7**	**48.9**	**48.7**	**23.8**
10h fine-tune										
Supervised Pre-train	100h	×	13.8	8.5	41.8	29.9	47.8	15.3	15	24.6
	960h	×	6.2	3.6	25.8	15.6	29.7	27.0	27.8	19.4
Wav2vec 2.0	×	960h	5.1	3.5	19.6	11.8	19.6	14.8	14.6	12.7
Wav2vec-S	100h	860h	**4.4**	**2.9**	**18.7**	**10.8**	**18.8**	**13.6**	**14.0**	**11.9**

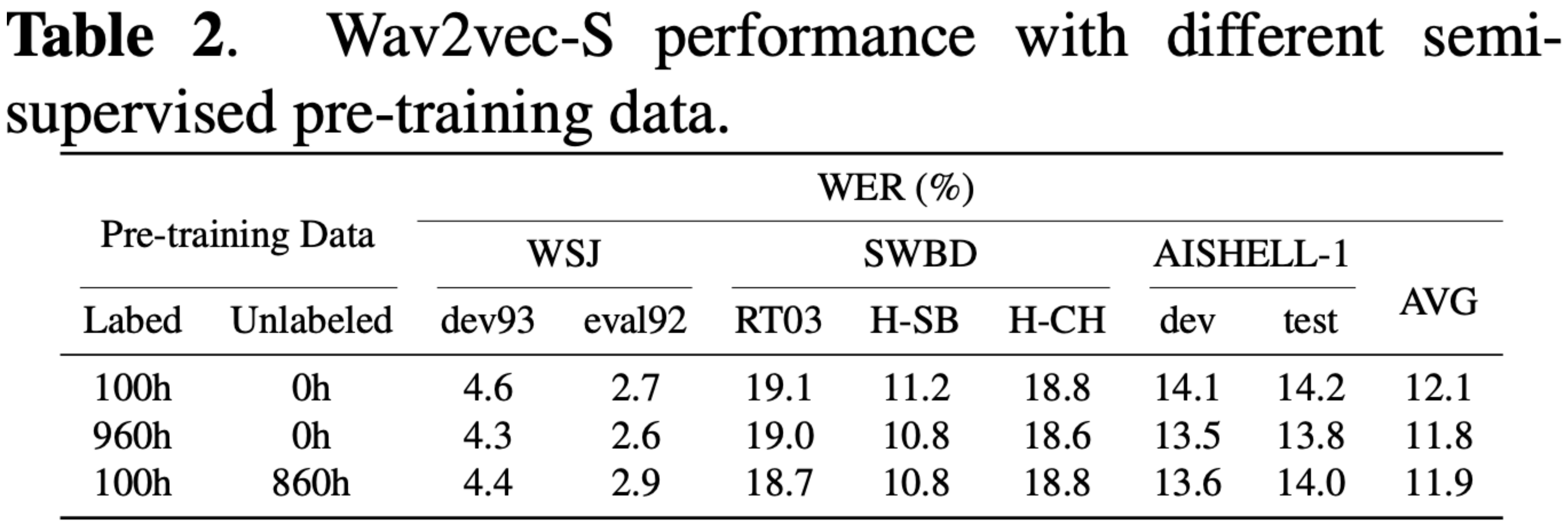


Table 2. Wav2vec-S performance with different semi-supervised pre-training data.

Pre-training Data		WER (%)							
		WSJ		SWBD			AISHELL-1		AVG
Labed	Unlabeled	dev93	eval92	RT03	H-SB	H-CH	dev	test	
100h	0h	4.6	2.7	19.1	11.2	18.8	14.1	14.2	12.1
960h	0h	4.3	2.6	19.0	10.8	18.6	13.5	13.8	11.8
100h	860h	4.4	2.9	18.7	10.8	18.8	13.6	14.0	11.9

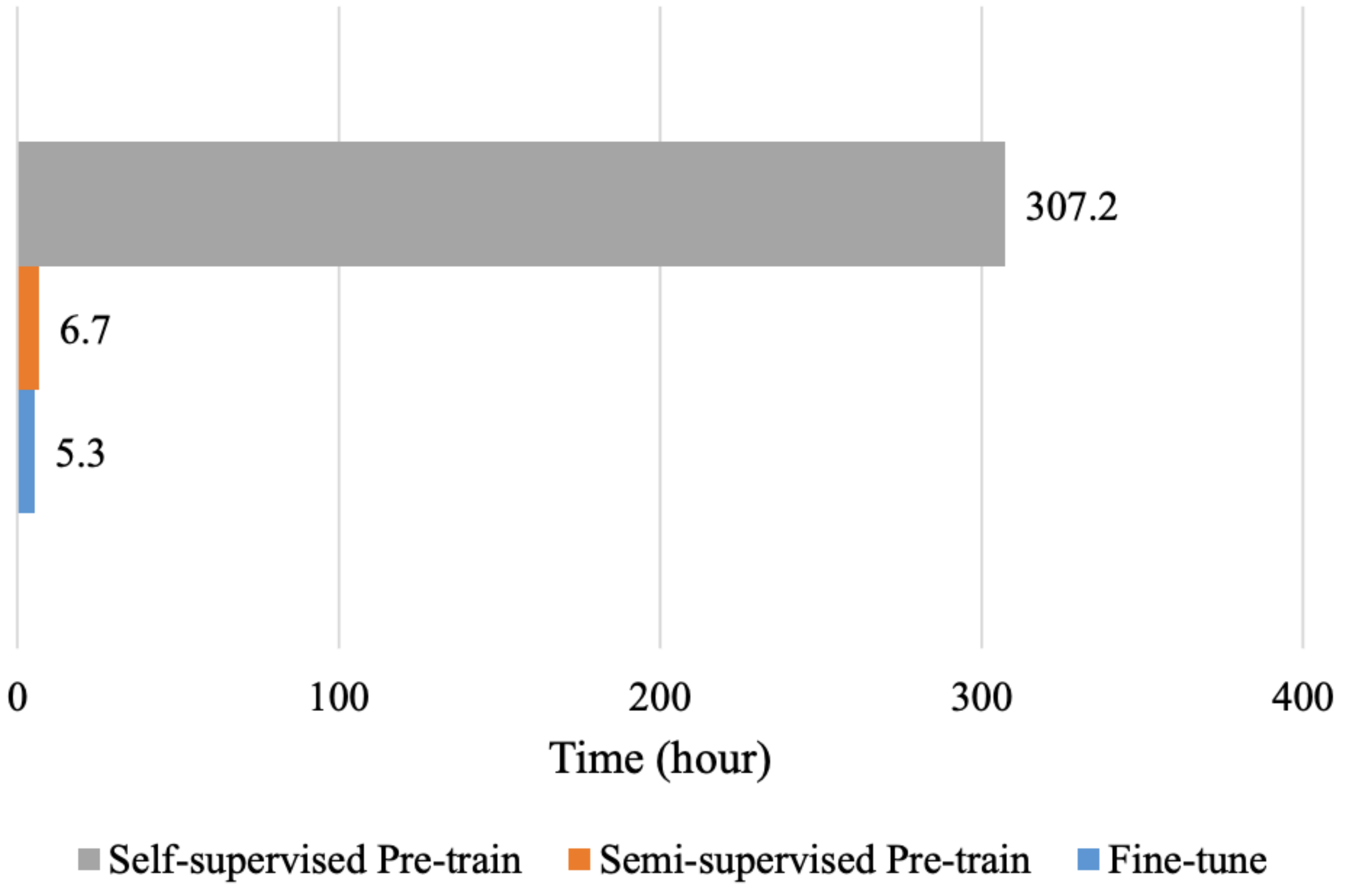


Fig. 2. Comparison of training time.

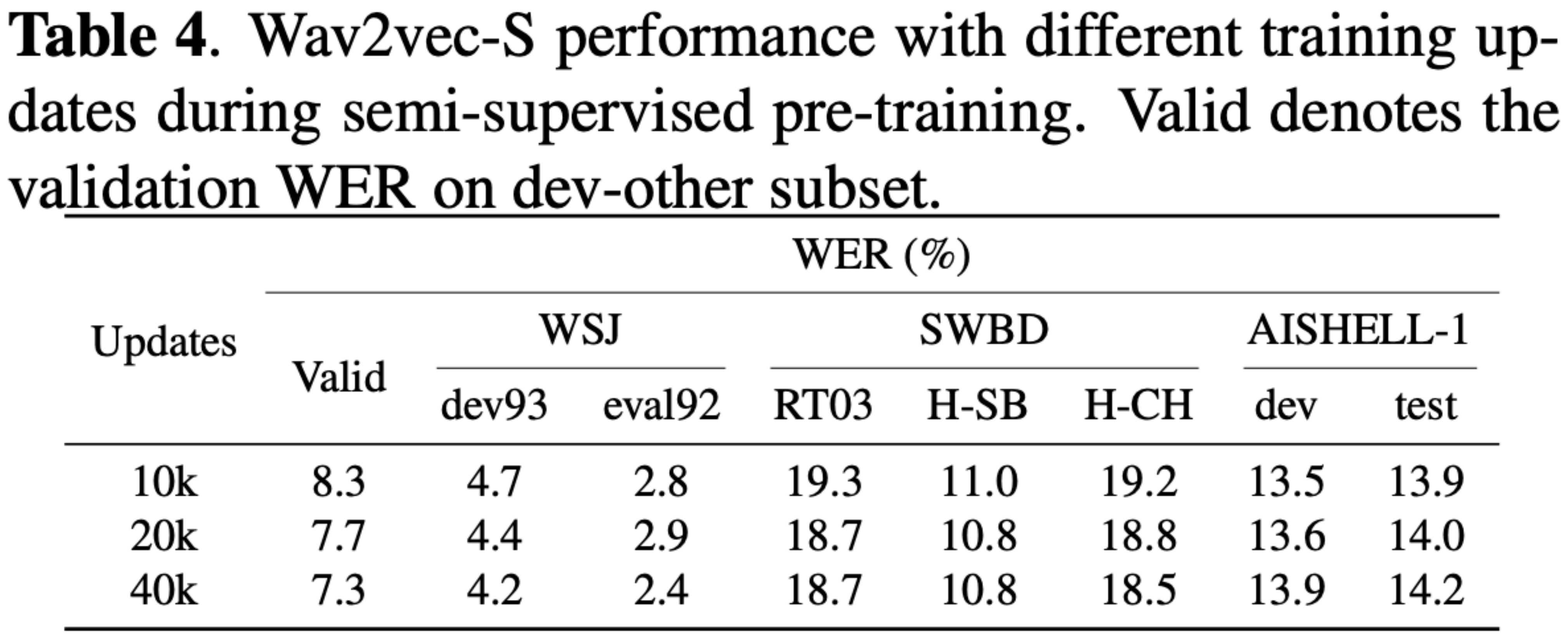
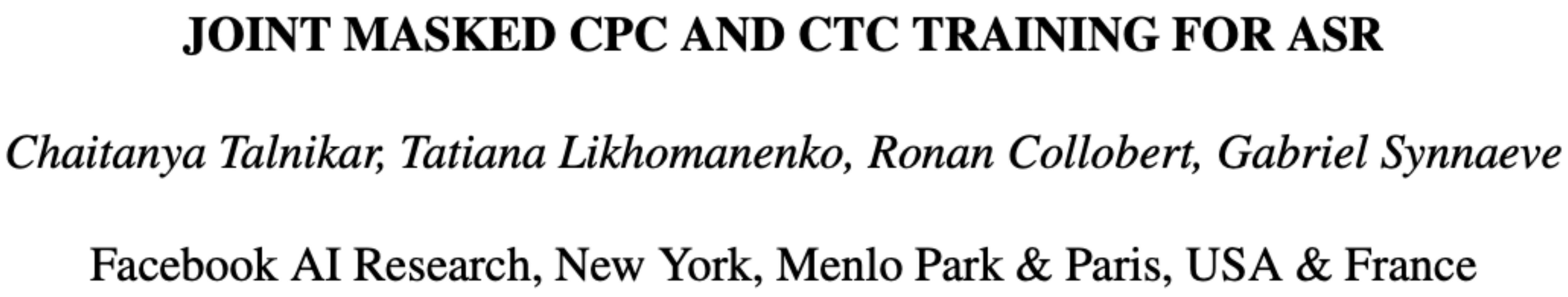


Table 4. Wav2vec-S performance with different training updates during semi-supervised pre-training. Valid denotes the validation WER on dev-other subset.

Updates	Valid	WER (%)						
		WSJ		SWBD			AISHELL-1	
		dev93	eval92	RT03	H-SB	H-CH	dev	test
10k	8.3	4.7	2.8	19.3	11.0	19.2	13.5	13.9
20k	7.7	4.4	2.9	18.7	10.8	18.8	13.6	14.0
40k	7.3	4.2	2.4	18.7	10.8	18.5	13.9	14.2

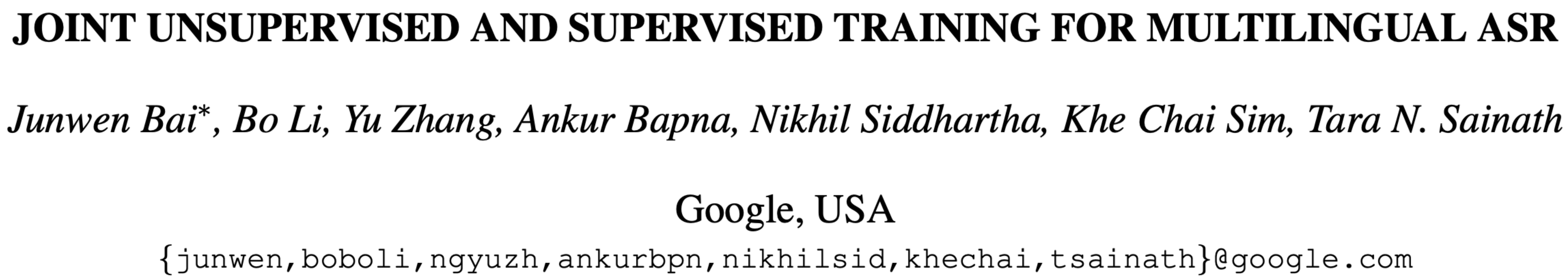
With more training updates, the wav2vec-S model becomes more language-specific and the cross-lingual generalization ability is degraded.



JOINT MASKED CPC AND CTC TRAINING FOR ASR

Chaitanya Talnikar, Tatiana Likhomanenko, Ronan Collobert, Gabriel Synnaeve

Facebook AI Research, New York, Menlo Park & Paris, USA & France



JOINT UNSUPERVISED AND SUPERVISED TRAINING FOR MULTILINGUAL ASR

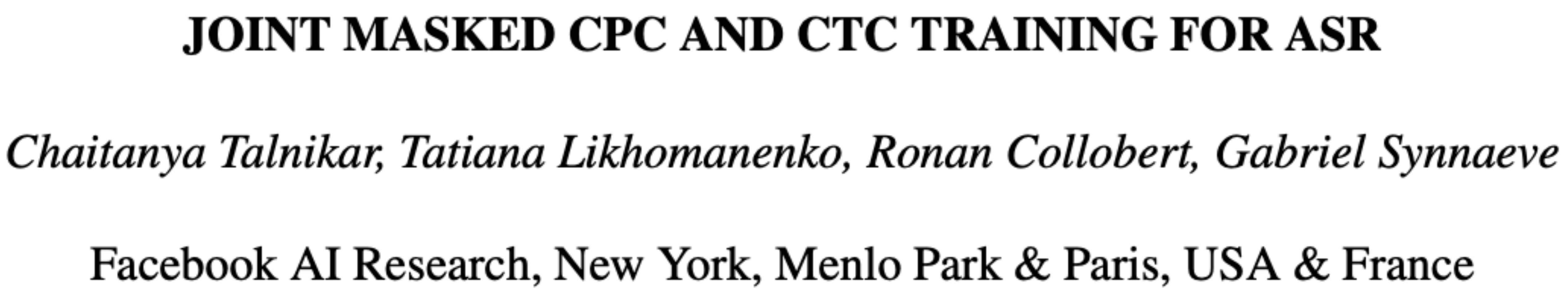
*Junwen Bai**, *Bo Li, Yu Zhang, Ankur Bapna, Nikhil Siddhartha, Khe Chai Sim, Tara N. Sainath*

Google, USA

{junwen,boboli,ngyuzh,ankurbpn,nikhilsid,khechai,tsainath}@google.com

Joint, not Semi-Supervised :

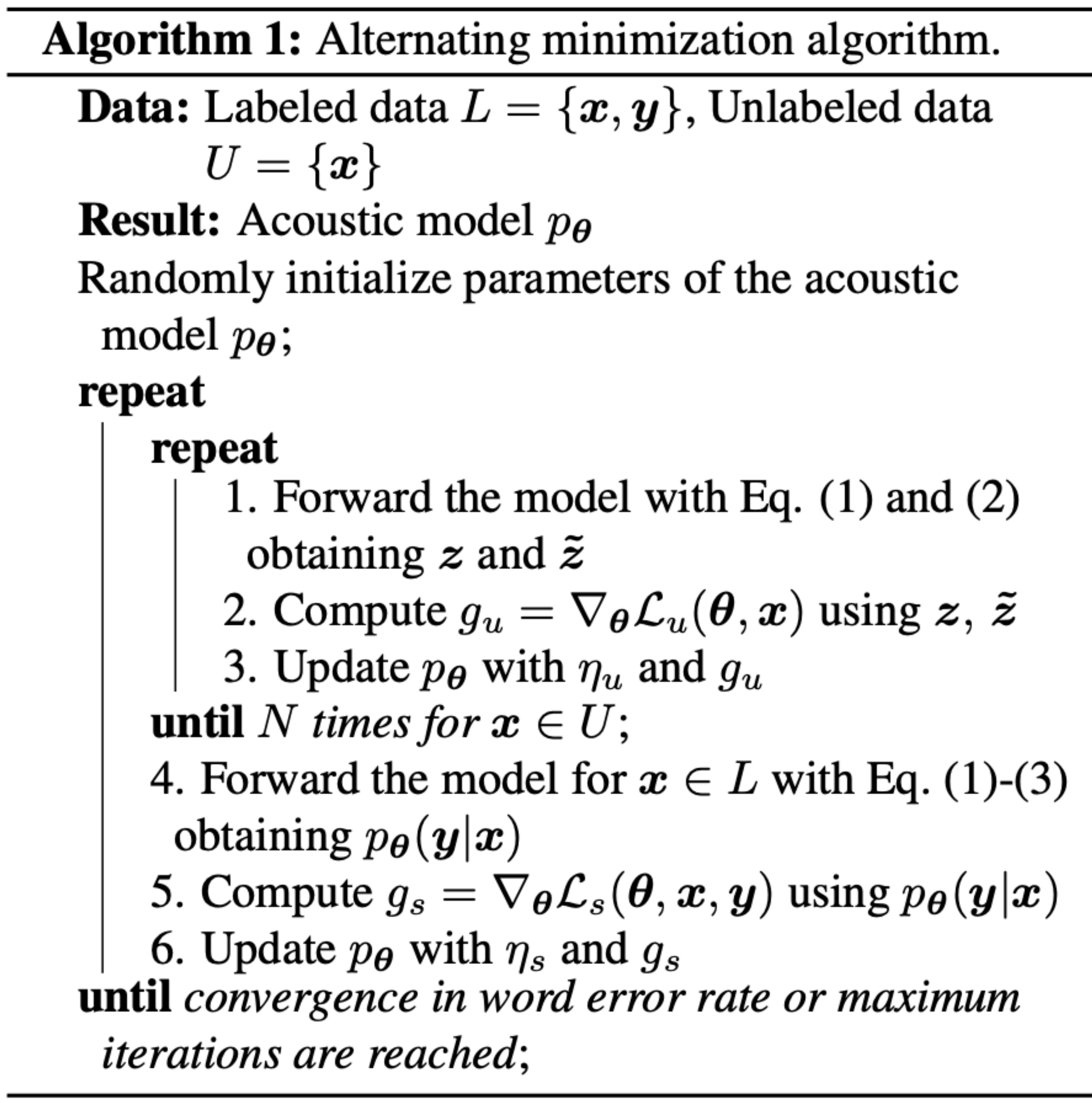
- No pseudo-labels
- Still task specific
- Still single stage training (for ASR) (—> early stop …)



JOINT MASKED CPC AND CTC TRAINING FOR ASR

Chaitanya Talnikar, Tatiana Likhomanenko, Ronan Collobert, Gabriel Synnaeve

Facebook AI Research, New York, Menlo Park & Paris, USA & France



```
Algorithm 1: Alternating minimization algorithm.
Data: Labeled data $L = \{\boldsymbol{x}, \boldsymbol{y}\}$, Unlabeled data
      $U = \{\boldsymbol{x}\}$
Result: Acoustic model $p_{\boldsymbol{\theta}}$
Randomly initialize parameters of the acoustic
 model $p_{\boldsymbol{\theta}}$;
repeat
    repeat
        1. Forward the model with Eq. (1) and (2)
         obtaining $\boldsymbol{z}$ and $\tilde{\boldsymbol{z}}$
        2. Compute $g_u = \nabla_{\boldsymbol{\theta}} \mathcal{L}_u(\boldsymbol{\theta}, \boldsymbol{x})$ using $\boldsymbol{z}$, $\tilde{\boldsymbol{z}}$
        3. Update $p_{\boldsymbol{\theta}}$ with $\eta_u$ and $g_u$
    until $N$ times for $\boldsymbol{x} \in U$;
    4. Forward the model for $\boldsymbol{x} \in L$ with Eq. (1)-(3)
     obtaining $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$
    5. Compute $g_s = \nabla_{\boldsymbol{\theta}} \mathcal{L}_s(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ using $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$
    6. Update $p_{\boldsymbol{\theta}}$ with $\eta_s$ and $g_s$
until convergence in word error rate or maximum
 iterations are reached;
```

Wav2Vec2 contrastive loss

CTC loss

Optimization related tricks :

- One batch of U, one batch of S
- Separate adaptative momentum optimizers with different learning rates —> updates on one loss are not affected by updates on the other loss
- N=1 : equal opportunity for the unsupervised and supervised loss. If N>1 : expensive, if inverse : ~supervised results

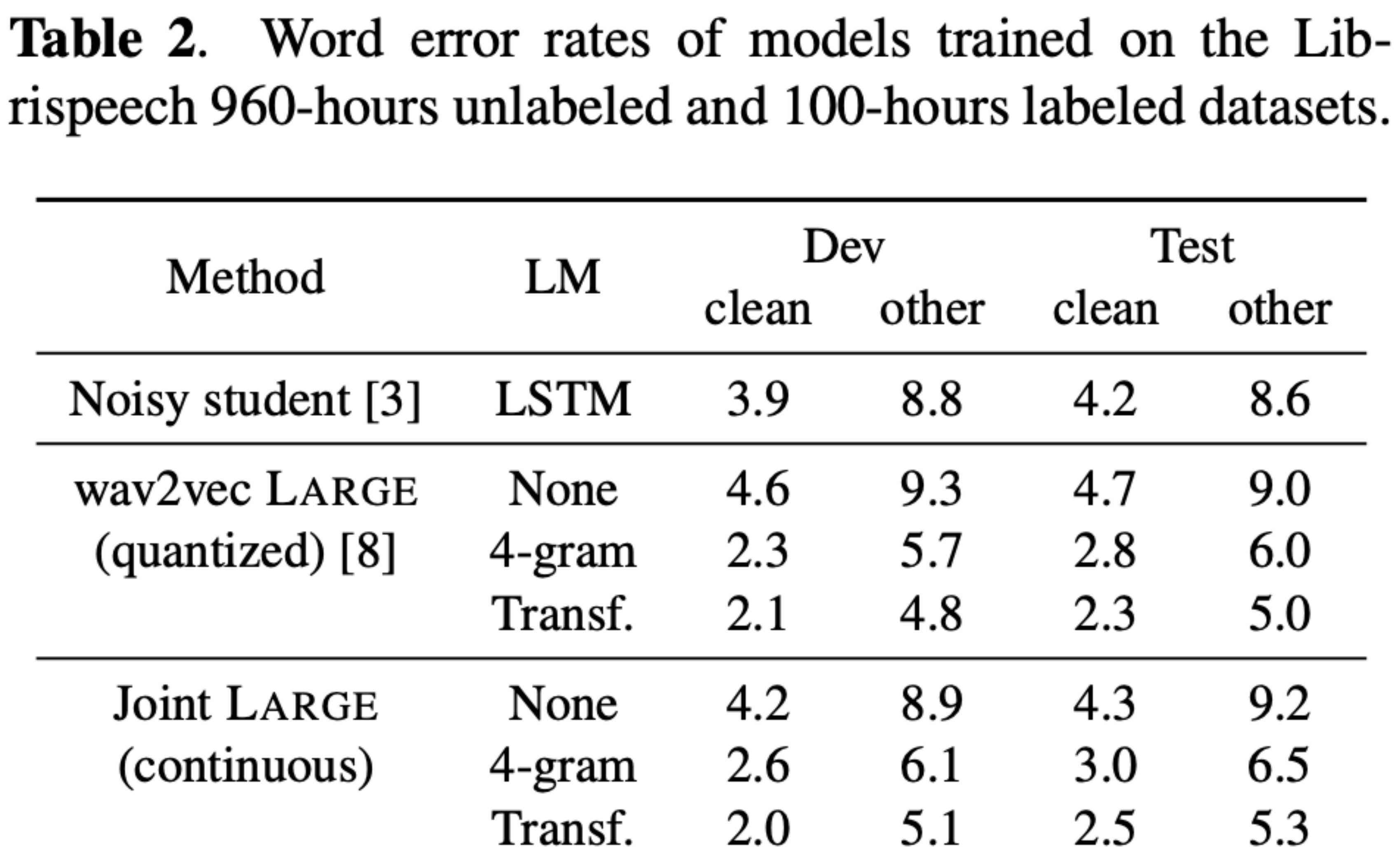


Table 2. Word error rates of models trained on the Librispeech 960-hours unlabeled and 100-hours labeled datasets.

Method	LM	Dev		Test	
		clean	other	clean	other
Noisy student [3]	LSTM	3.9	8.8	4.2	8.6
wav2vec LARGE (quantized) [8]	None	4.6	9.3	4.7	9.0
	4-gram	2.3	5.7	2.8	6.0
	Transf.	2.1	4.8	2.3	5.0
Joint LARGE (continuous)	None	4.2	8.9	4.3	9.2
	4-gram	2.6	6.1	3.0	6.5
	Transf.	2.0	5.1	2.5	5.3

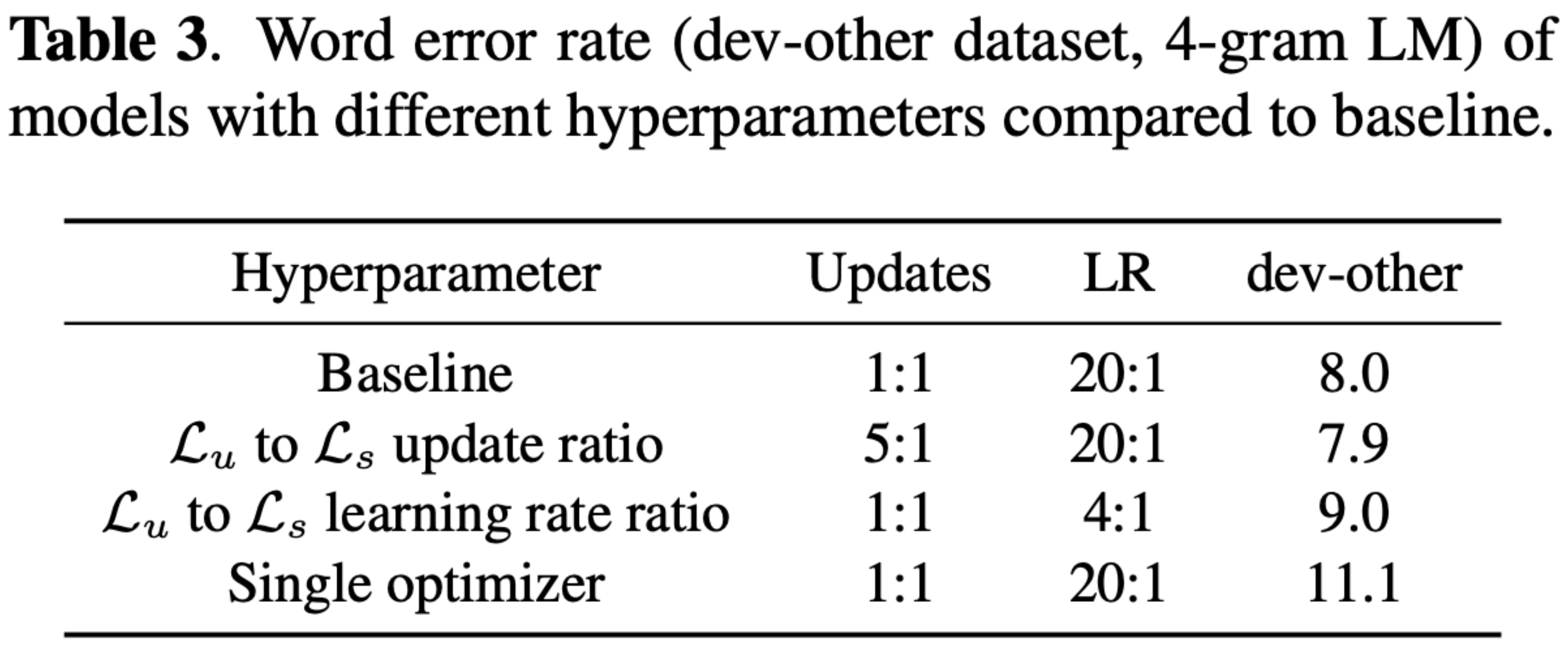


Table 3. Word error rate (dev-other dataset, 4-gram LM) of models with different hyperparameters compared to baseline.

Hyperparameter	Updates	LR	dev-other
Baseline	1:1	20:1	8.0
$\mathcal{L}_u$ to $\mathcal{L}_s$ update ratio	5:1	20:1	7.9
$\mathcal{L}_u$ to $\mathcal{L}_s$ learning rate ratio	1:1	4:1	9.0
Single optimizer	1:1	20:1	11.1

No single optimizer

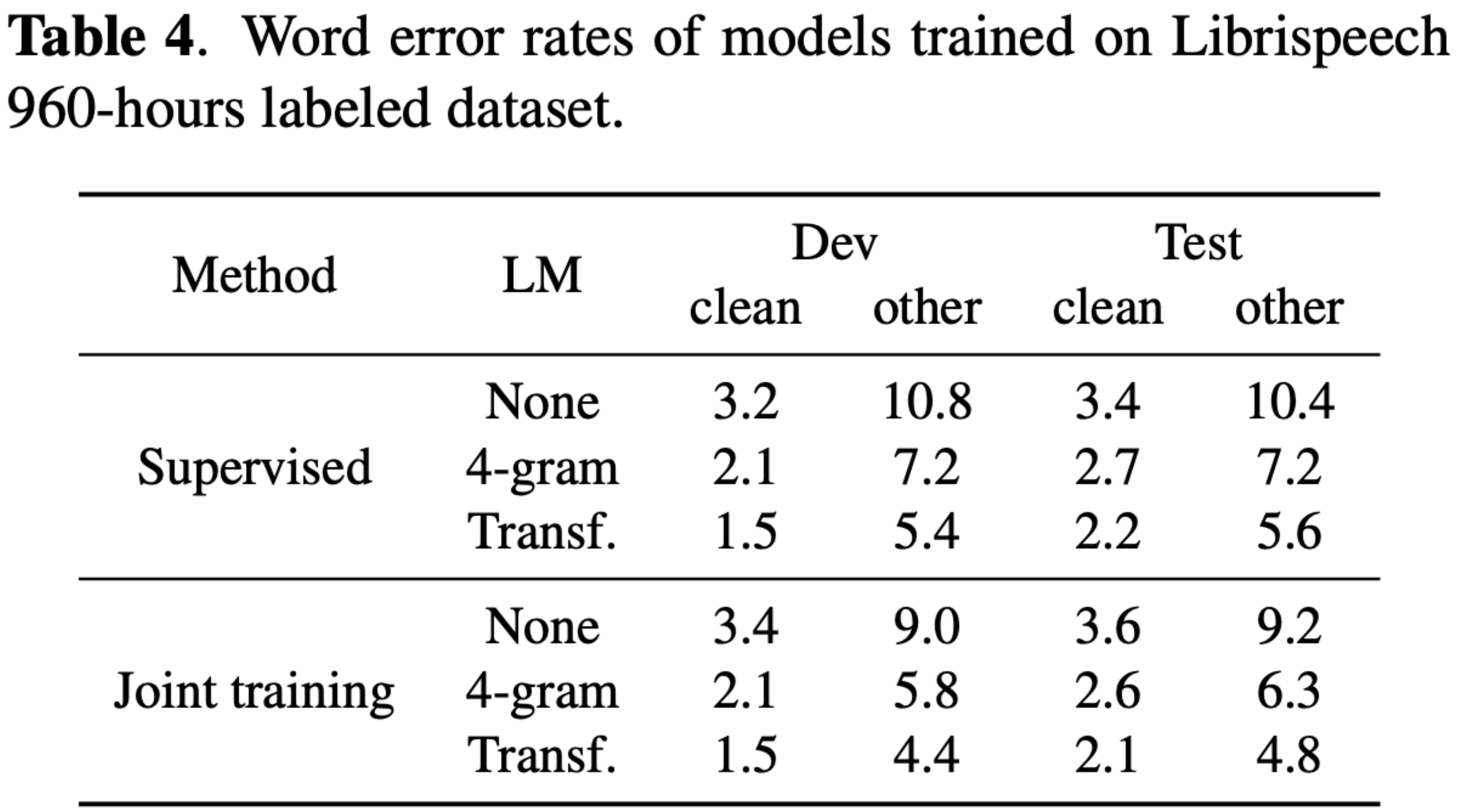


Table 4. Word error rates of models trained on Librispeech 960-hours labeled dataset.

Method	LM	Dev		Test	
		clean	other	clean	other
Supervised	None	3.2	10.8	3.4	10.4
	4-gram	2.1	7.2	2.7	7.2
	Transf.	1.5	5.4	2.2	5.6
Joint training	None	3.4	9.0	3.6	9.2
	4-gram	2.1	5.8	2.6	6.3
	Transf.	1.5	4.4	2.1	4.8

The method provides a regularization to the supervised loss when only using labeled data

JOINT UNSUPERVISED AND SUPERVISED TRAINING FOR MULTILINGUAL ASR

Junwen Bai, Bo Li, Yu Zhang, Ankur Bapna, Nikhil Siddhartha, Khe Chai Sim, Tara N. Sainath*

Google, USA

{junwen,boboli,ngyuzh,ankurbpn,nikhilsid,khechai,tsainath}@google.com

Differences :

- 2 SSL losses instead of one, inspired from w2v-bert
- RNN-T instead of CTC
- Trained on MLS, not only LS

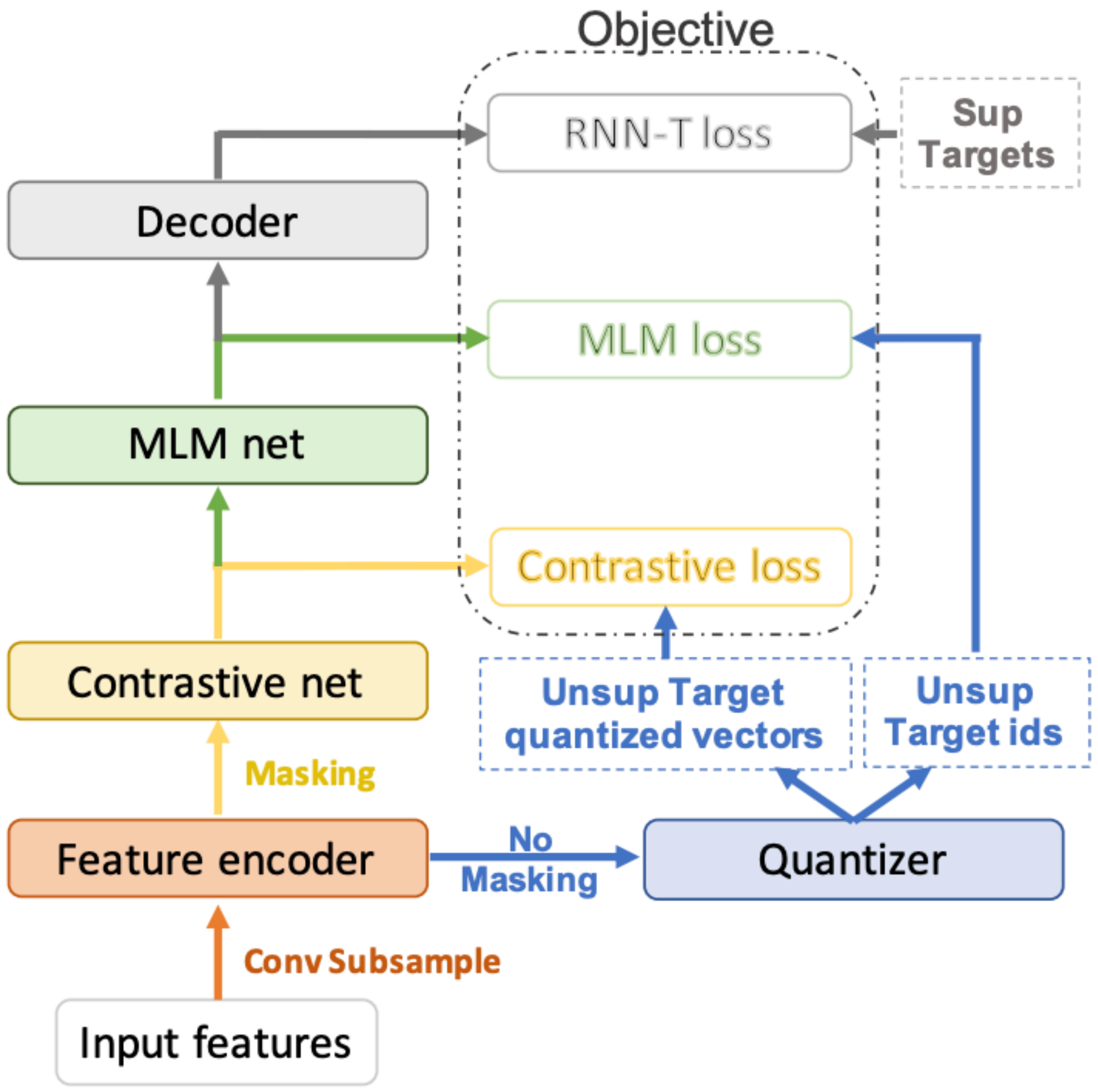
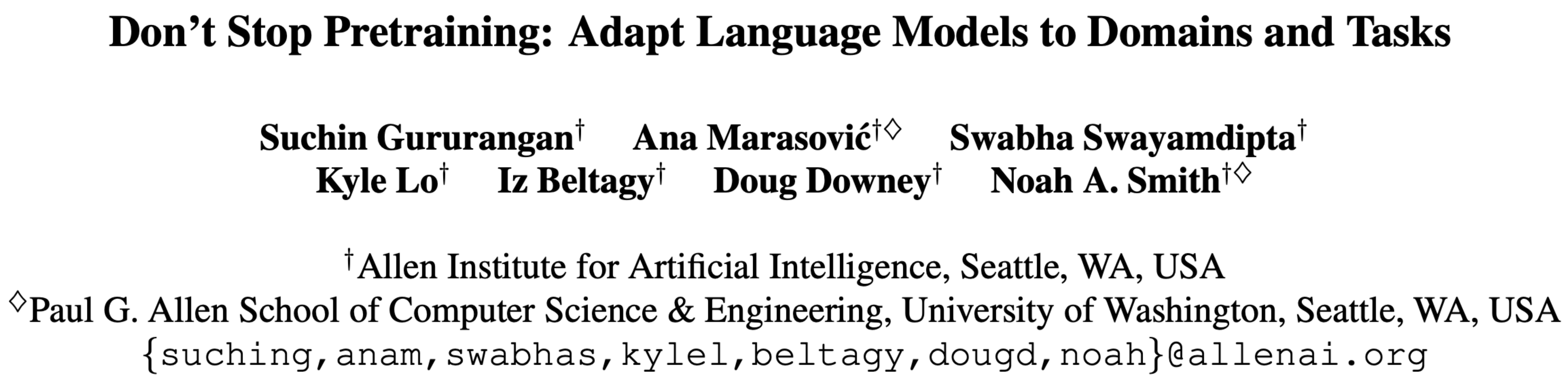


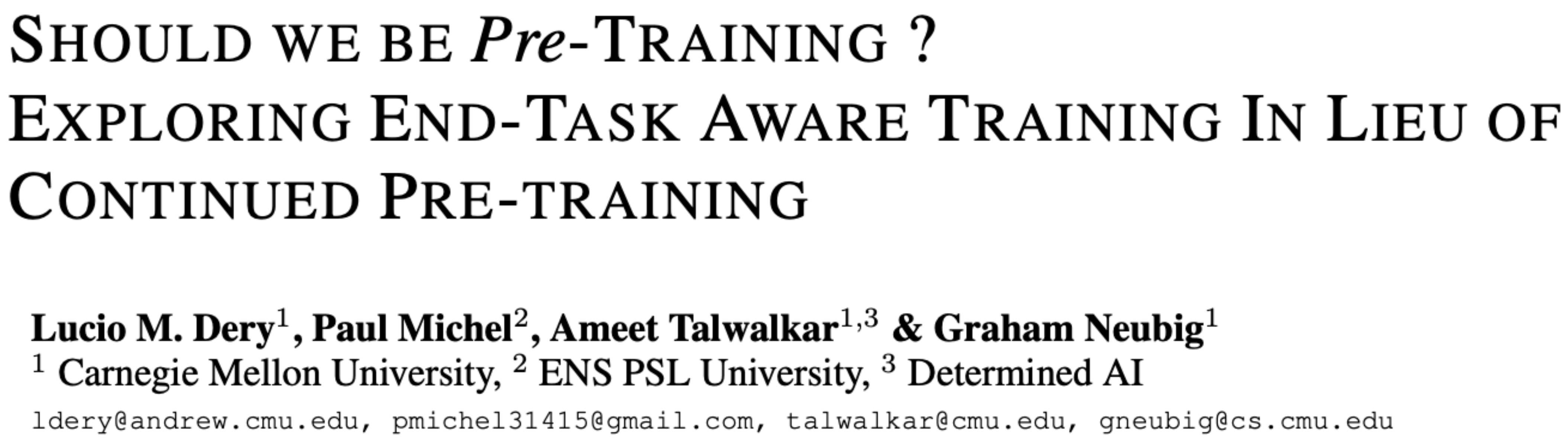
Fig. 1: An overview of our JUST framework. Feature encoder, contrastive net, MLM net and decoder are stacked sequentially. The output of each module constitutes a loss in the objective function. Target vectors and ids in the blue boxes are for unsupervised losses. Supervised targets in the grey box are for RNN-T loss.



Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks

Suchin Gururangan[†] Ana Marasović[†◇] Swabha Swayamdipta[†]
Kyle Lo[†] Iz Beltagy[†] Doug Downey[†] Noah A. Smith[†◇]

[†]Allen Institute for Artificial Intelligence, Seattle, WA, USA
[◇]Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA, USA
{suching,anam,swabhas,kylel,beltagy,dougd,noah}@allenai.org



SHOULD WE BE *PRE*-TRAINING ? EXPLORING END-TASK AWARE TRAINING IN LIEU OF CONTINUED PRE-TRAINING

Lucio M. Dery[1], Paul Michel[2], Ameet Talwalkar[1,3] & Graham Neubig[1]
[1] Carnegie Mellon University, [2] ENS PSL University, [3] Determined AI
ldery@andrew.cmu.edu, pmichel31415@gmail.com, talwalkar@cmu.edu, gneubig@cs.cmu.edu

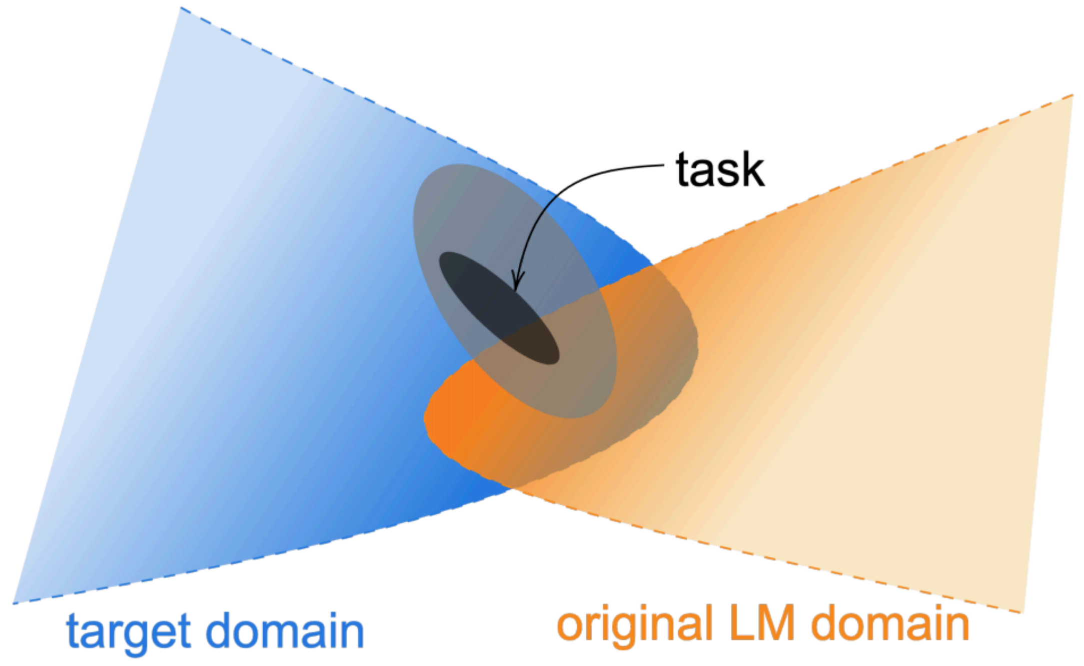


Figure 1: An illustration of data distributions. Task data is comprised of an observable task distribution, usually non-randomly sampled from a wider distribution (light grey ellipsis) within an even larger target domain, which is not necessarily one of the domains included in the original LM pretraining domain – though overlap is possible. We explore the benefits of continued pretraining on data from the task distribution and the domain distribution.

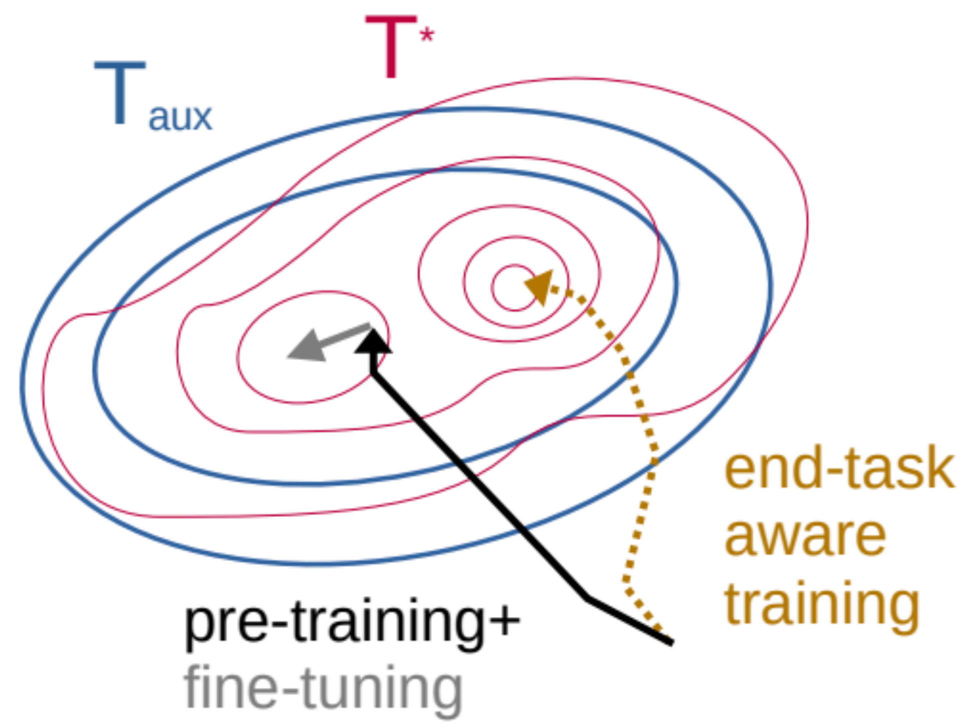


Figure 1: Pre-training trains on auxiliary task T_{aux} before fine-tuning on primary task T^*. End-task aware training optimizes both T_{aux} and T^* simultaneously and can find better minima since optimization is informed by the end-task.

Exploit the fact that we often know the end-task beforehand, and so we can make specific choices about our pre-training regimen to improve end-task performance.

Specific Continue PT approaches
Or Multi-task framework with task specific loss

Conclusion and discussion points?

- Other interesting papers (Unispeech, XSLT …)
- Filtering techniques
- Other loss (MLM, intermediate?)
- Low Resource scenario (10h supervised ?)
- Softer labels ?