Continuous Pseudo-Labeling from the Start

Dan Berrebbi*, Ronan Collobert†, Samy Bengio†, Navdeep Jaitly†, Tatiana Likhomanenko†

ICLR 2023 - Kigali, Rwanda

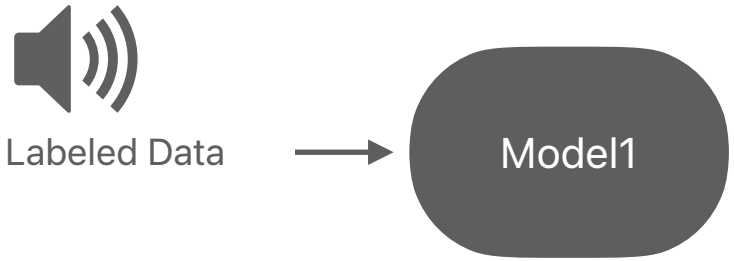
Carnegie Mellon University
Language Technologies Institute

Pseudo-Labeling (PL) to Leverage Unlabeled Data

Easy to implement + uses less compute than SSL

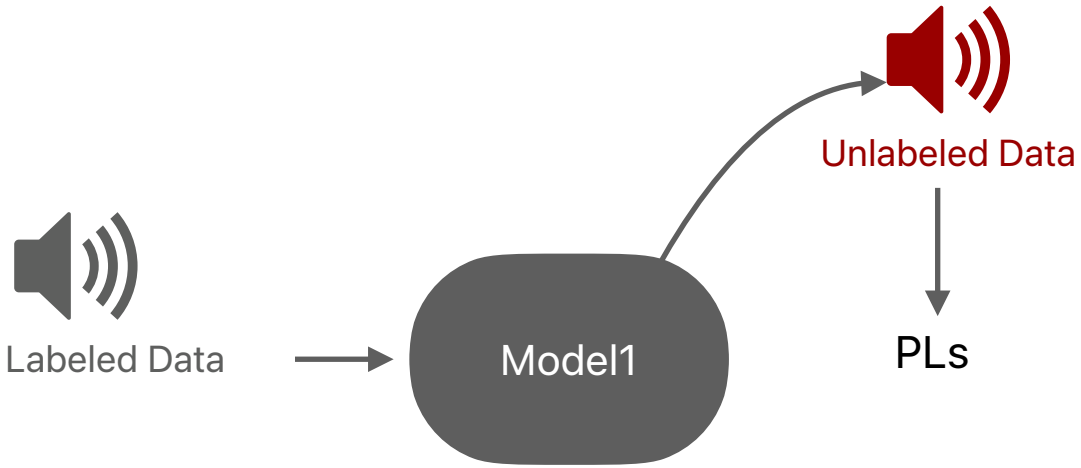
Pseudo-Labeling (PL) to Leverage Unlabeled Data

Easy to implement + uses less compute than SSL



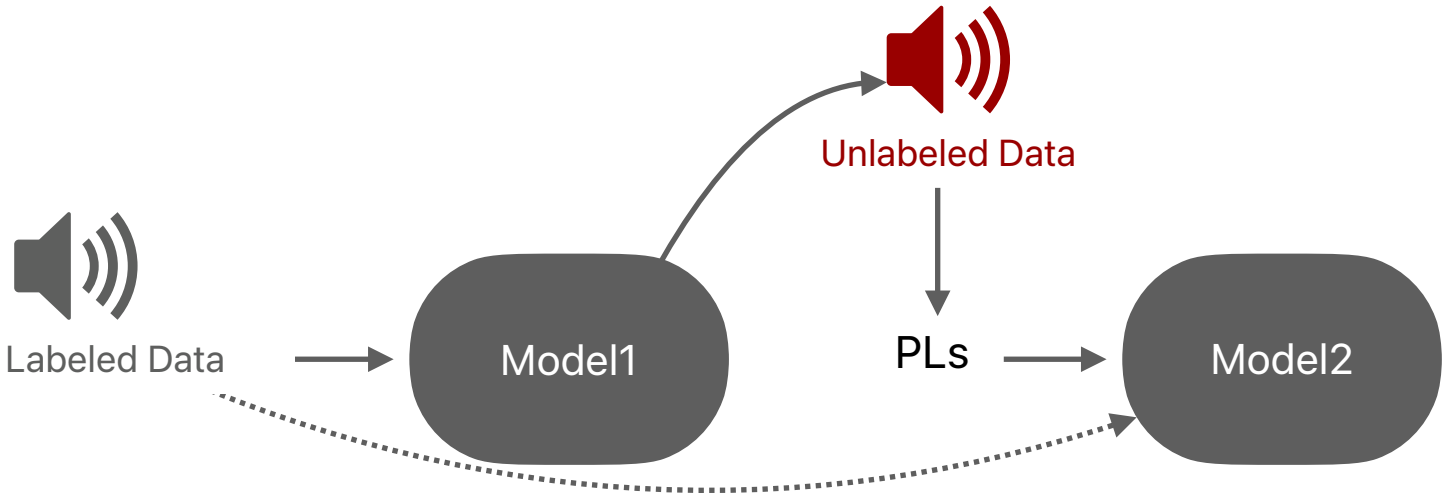
Pseudo-Labeling (PL) to Leverage Unlabeled Data

Easy to implement + uses less compute than SSL



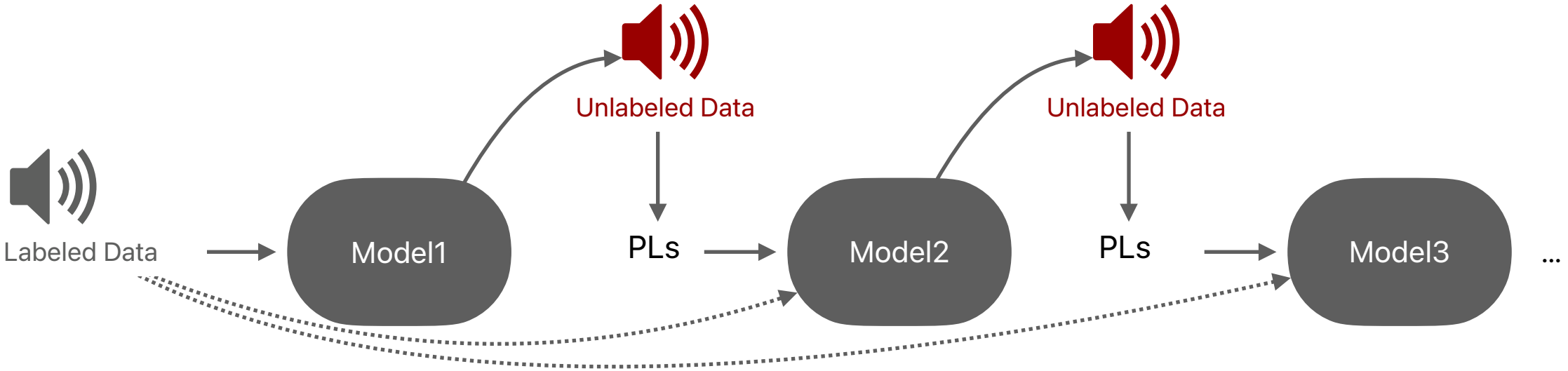
Pseudo-Labeling (PL) to Leverage Unlabeled Data

Easy to implement + uses less compute than SSL



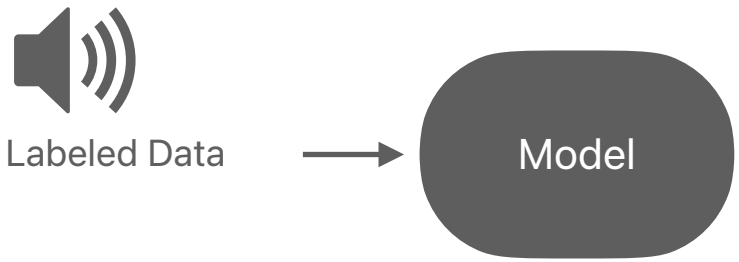
Pseudo-Labeling (PL) to Leverage Unlabeled Data

Easy to implement + uses less compute than SSL

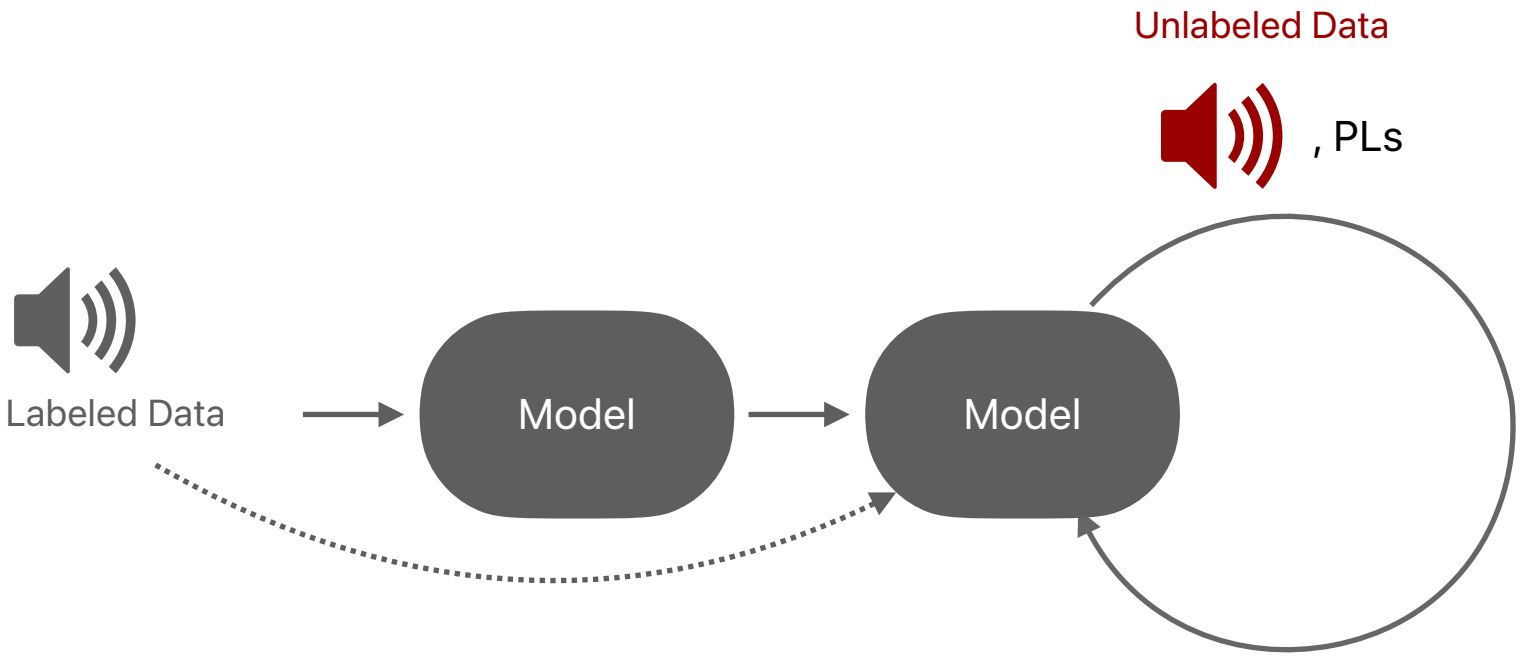


Continuous Pseudo-Labeling: Simple and Efficient

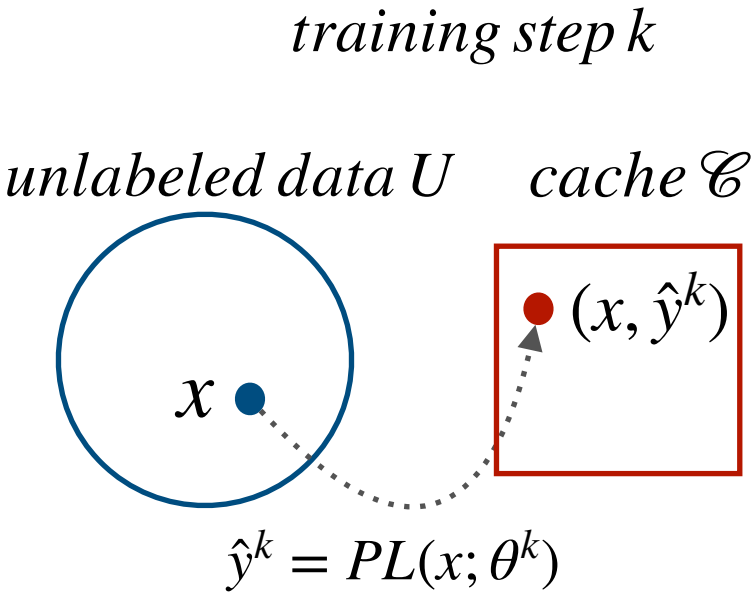
Continuous Pseudo-Labeling: Simple and Efficient



Continuous Pseudo-Labeling: Simple and Efficient



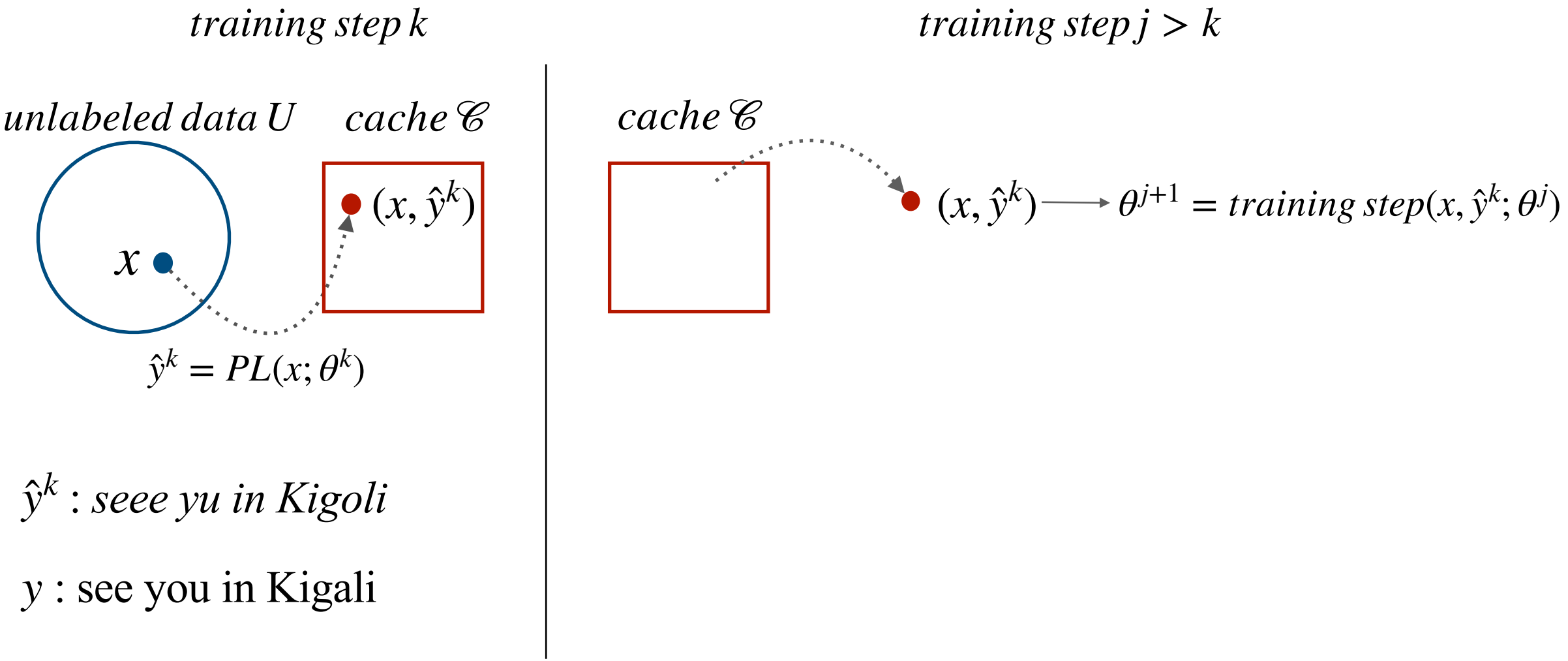
SlimIPL: Cache for Stable Continuous PL



$\hat{y}^k$: *seee yu in Kigoli*

y : see you in Kigali

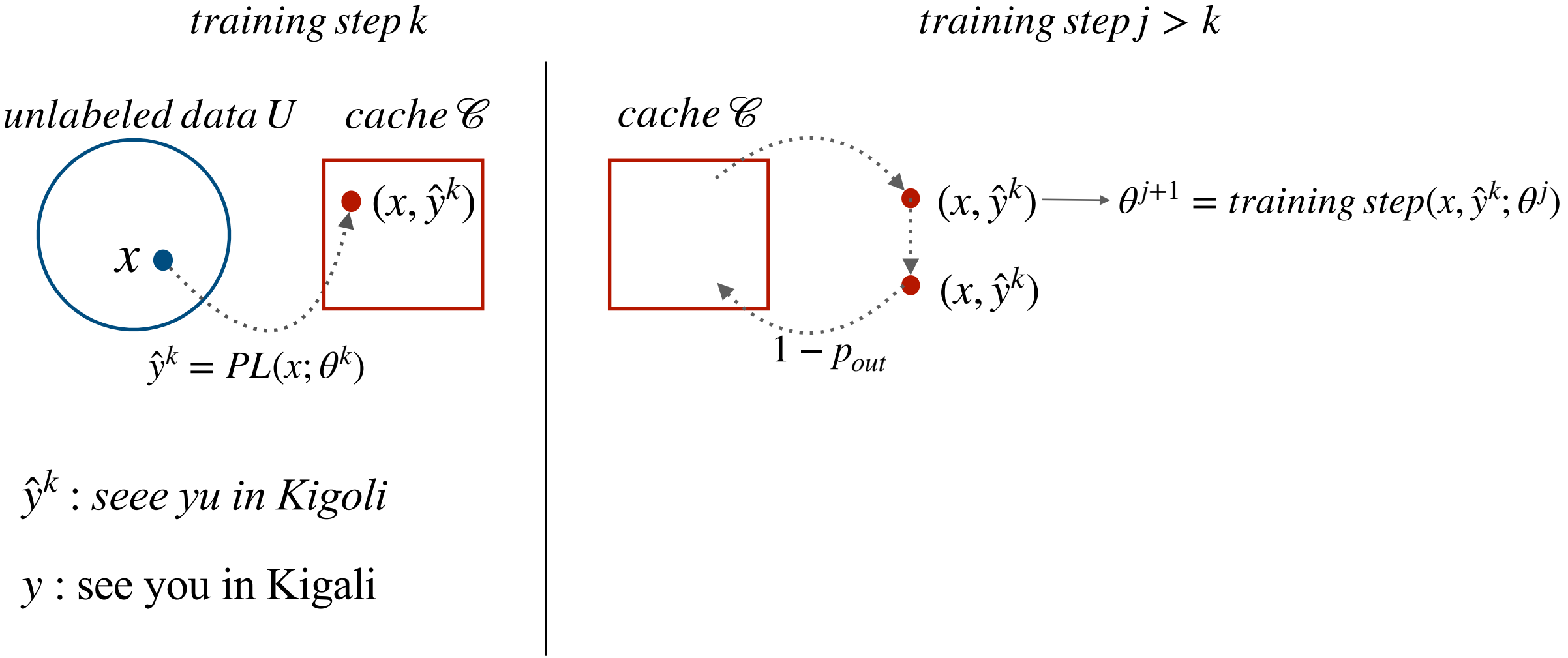
SlimIPL: Cache for Stable Continuous PL



$\hat{y}^k$: *seee yu in Kigoli*

y : see you in Kigali

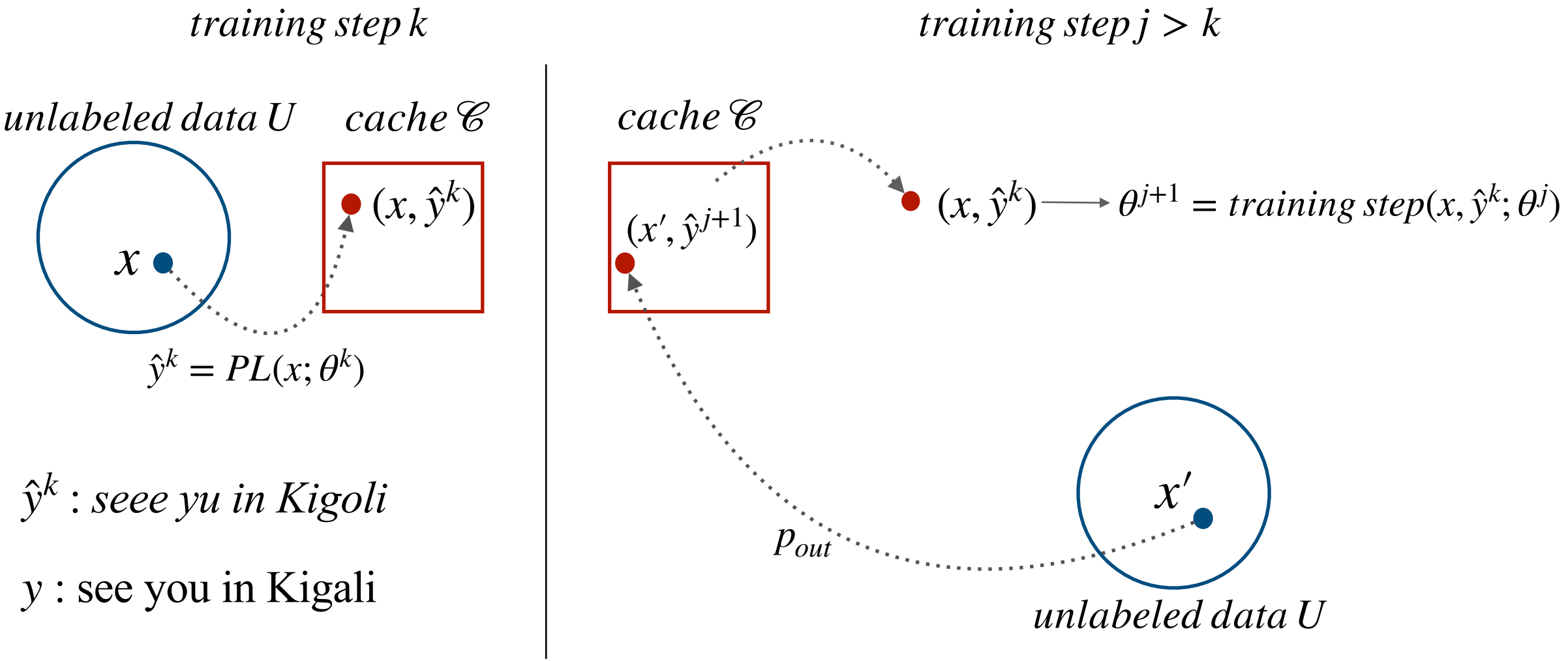
SlimIPL: Cache for Stable Continuous PL



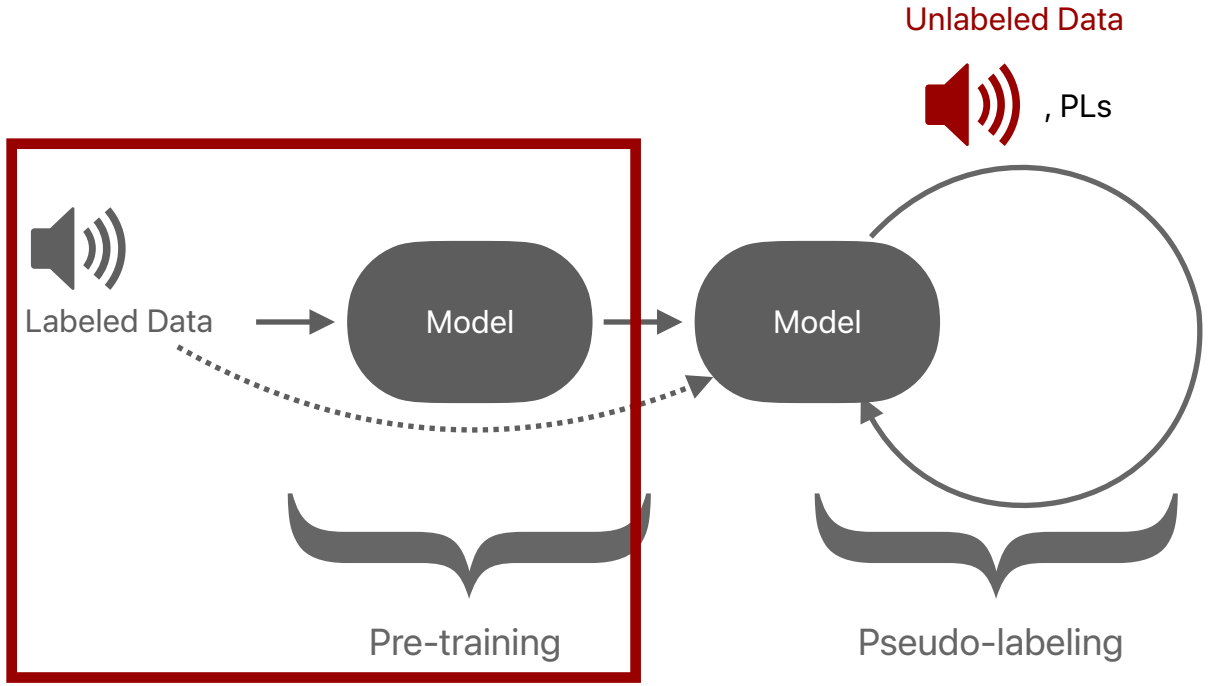
$\hat{y}^k$: *seee yu in Kigoli*

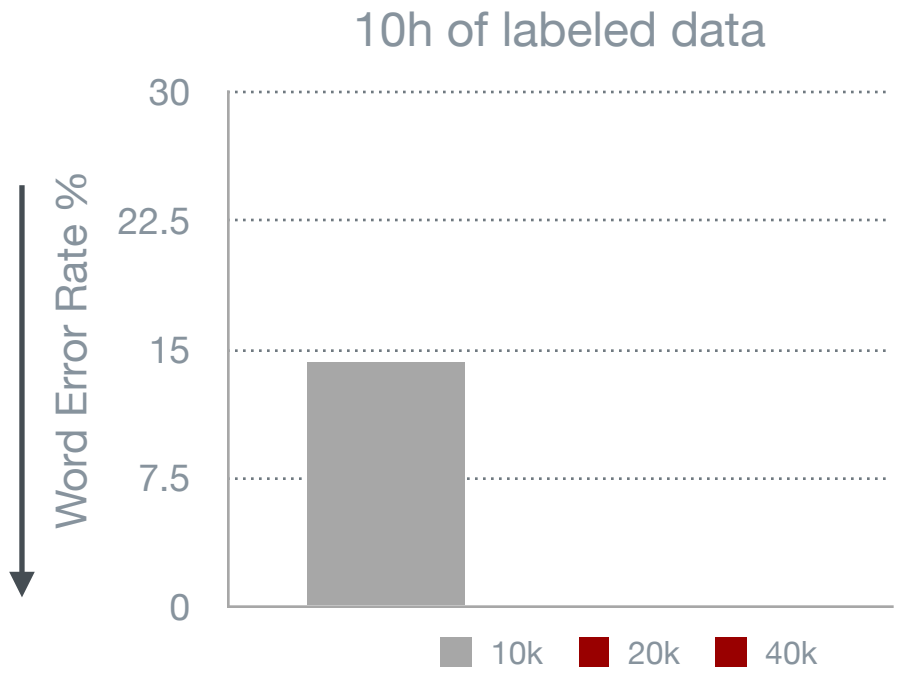
y : see you in Kigali

SlimIPL: Cache for Stable Continuous PL

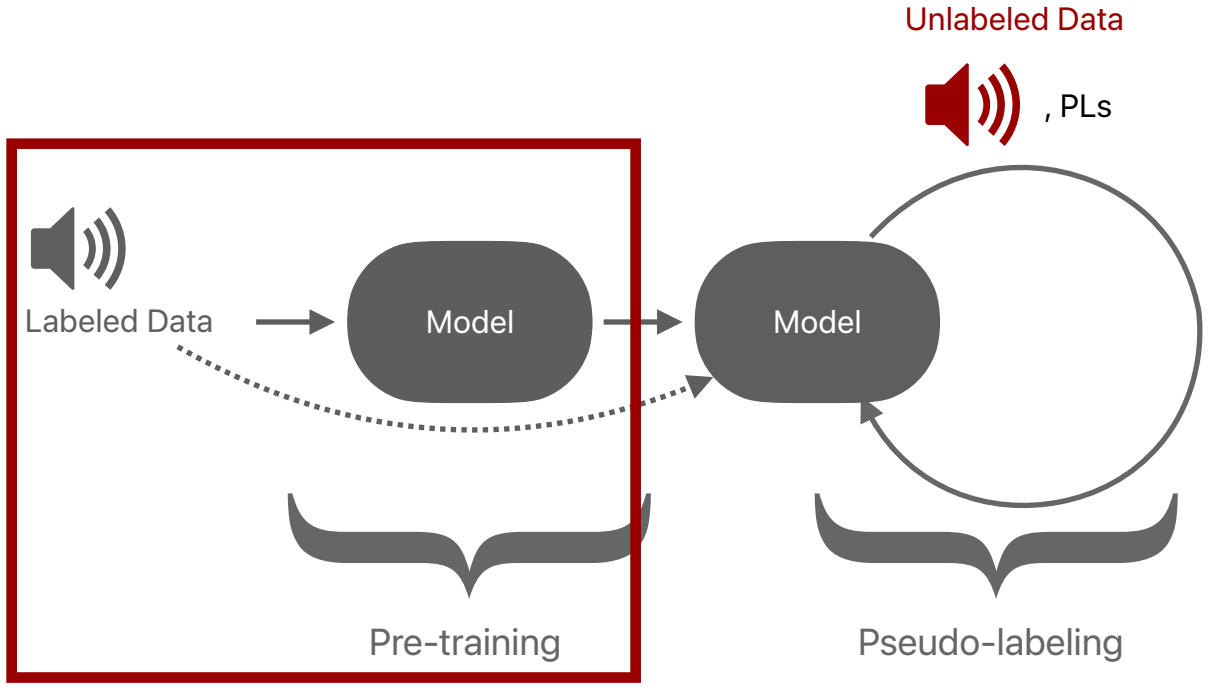


Dependence on Pre-Training Steps

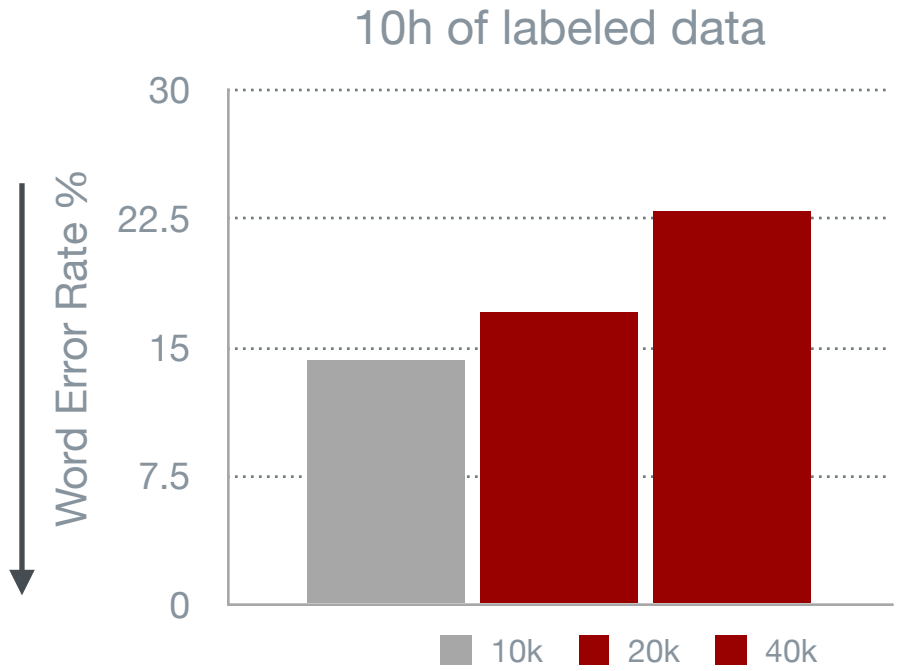




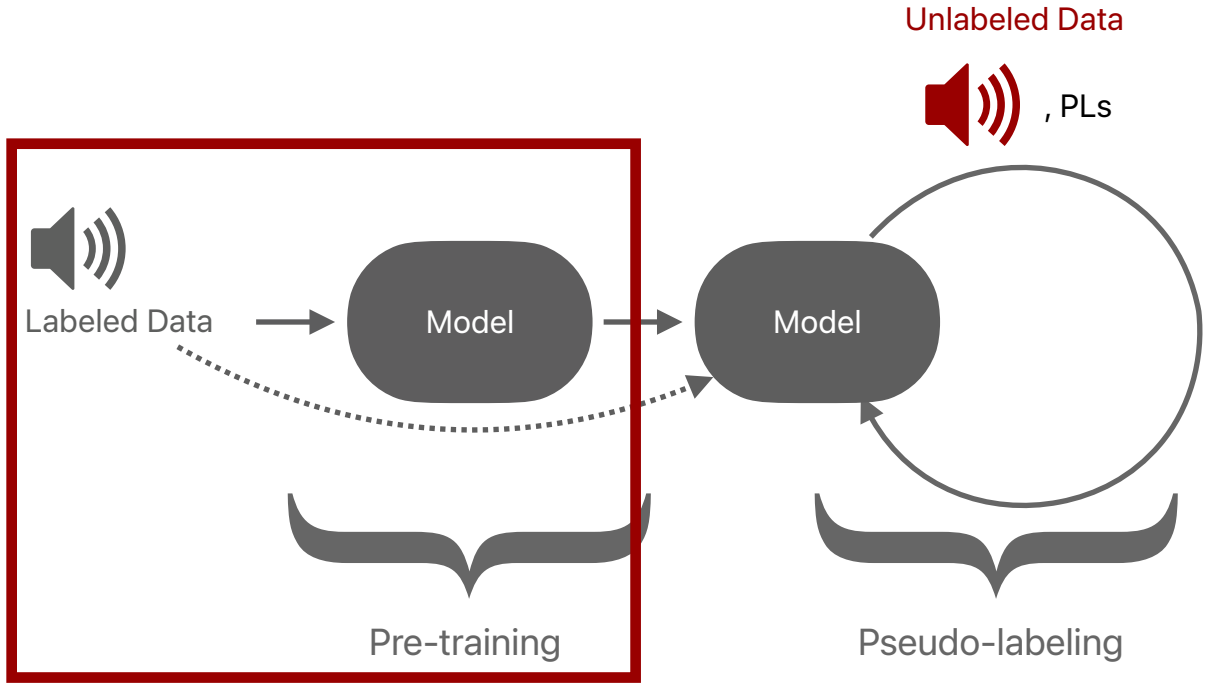
Dependence on Pre-Training Steps

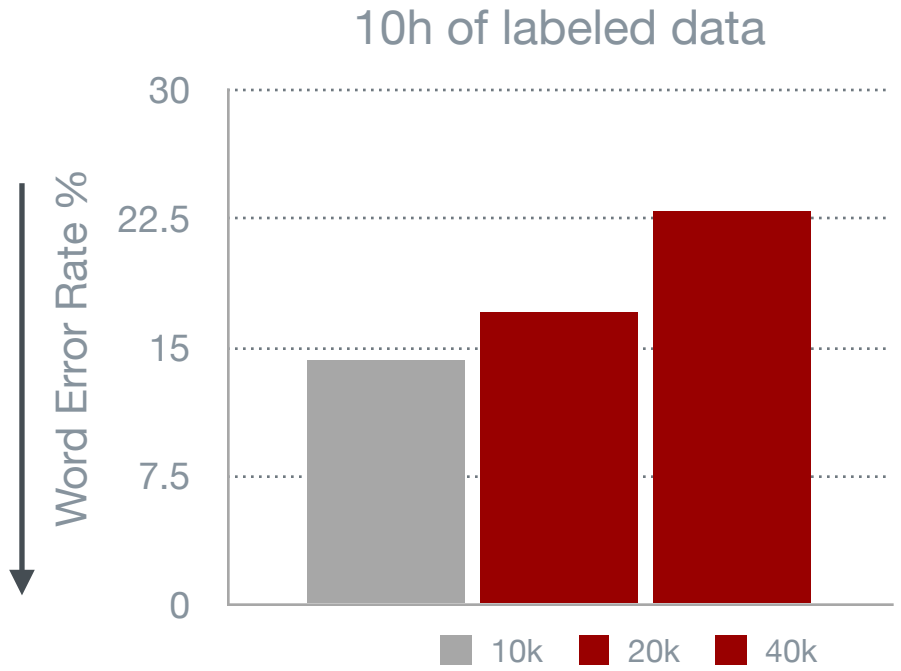


Long pre-training → worse performance (overfit)



Dependence on Pre-Training Steps

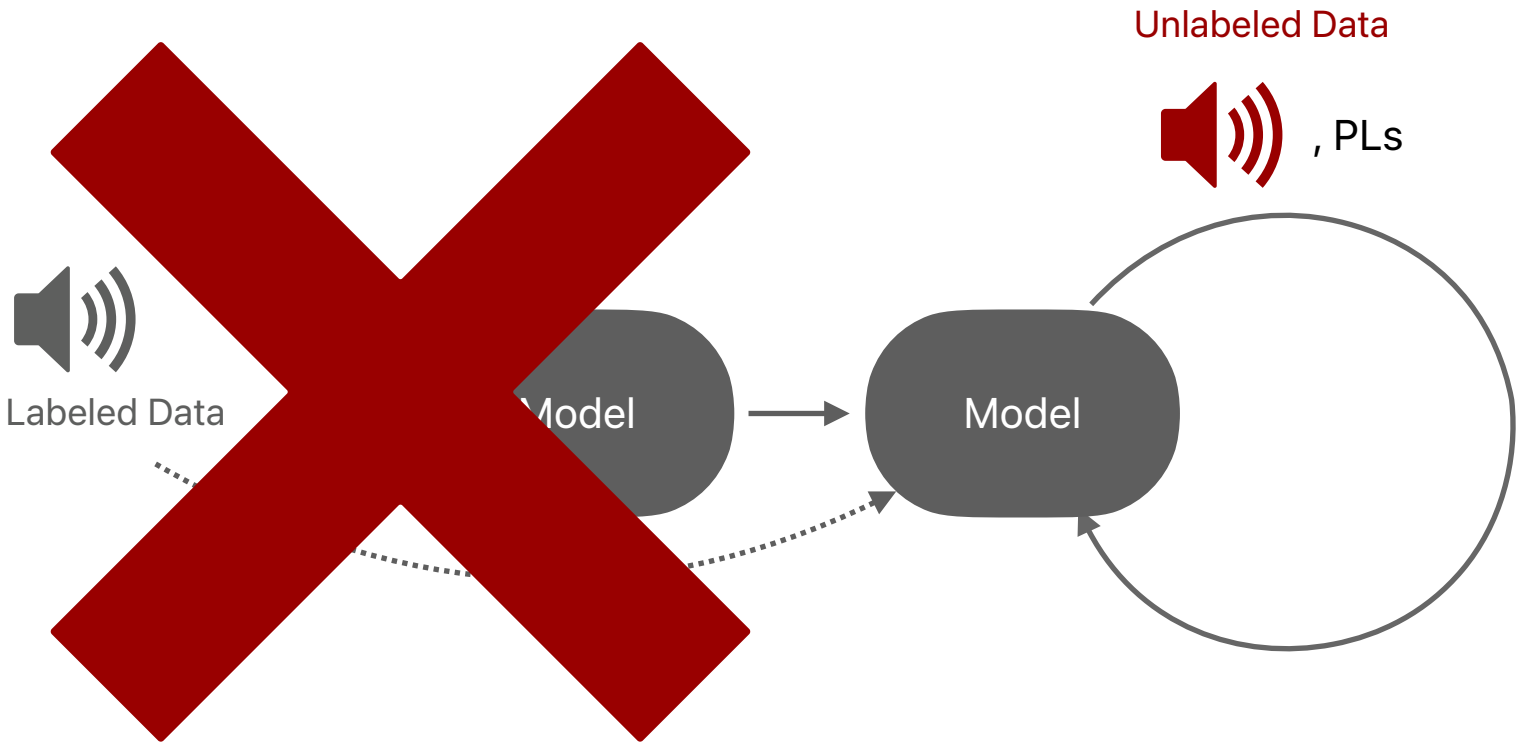




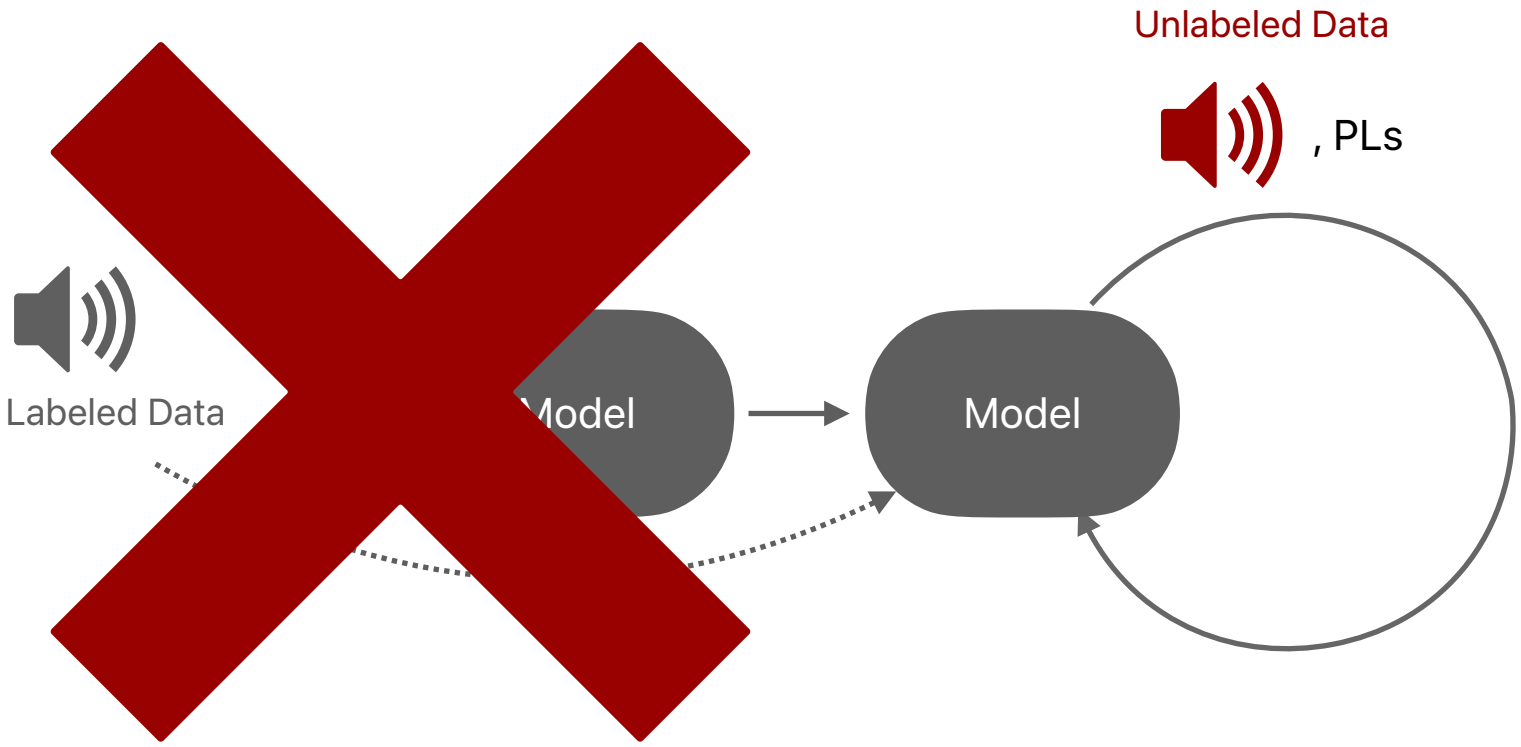
Long pre-training → worse performance (overfit)

Short pre-training → divergence (unstable model)

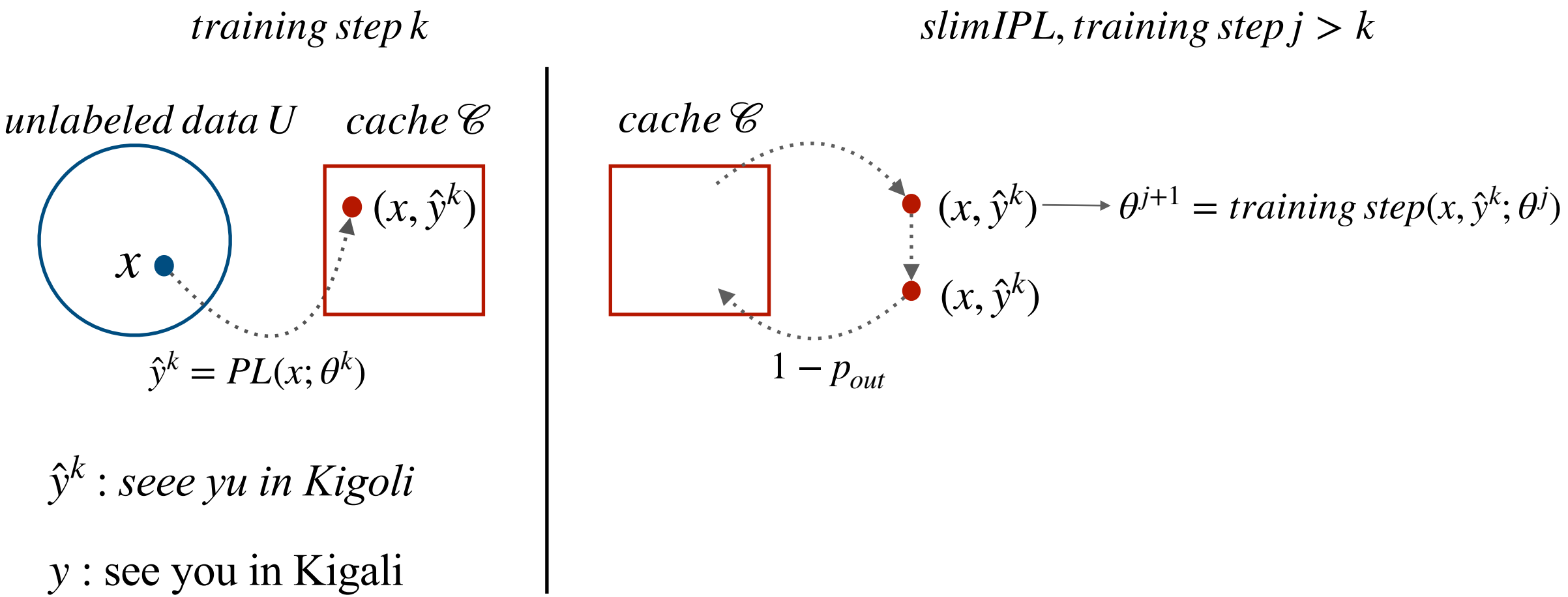
PL without Pre-Training



PL without Pre-Training



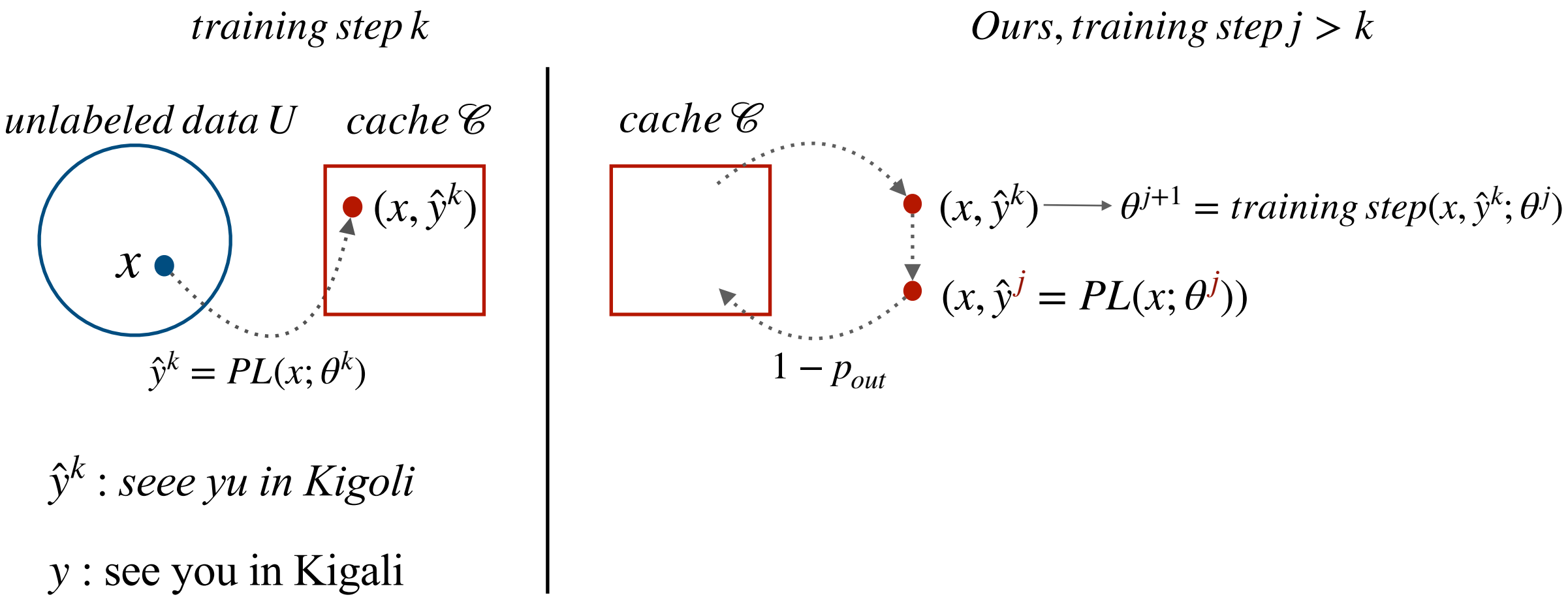
Controlling Cache by PL Evolution



$\hat{y}^k$: *seee yu in Kigoli*

y : see you in Kigali

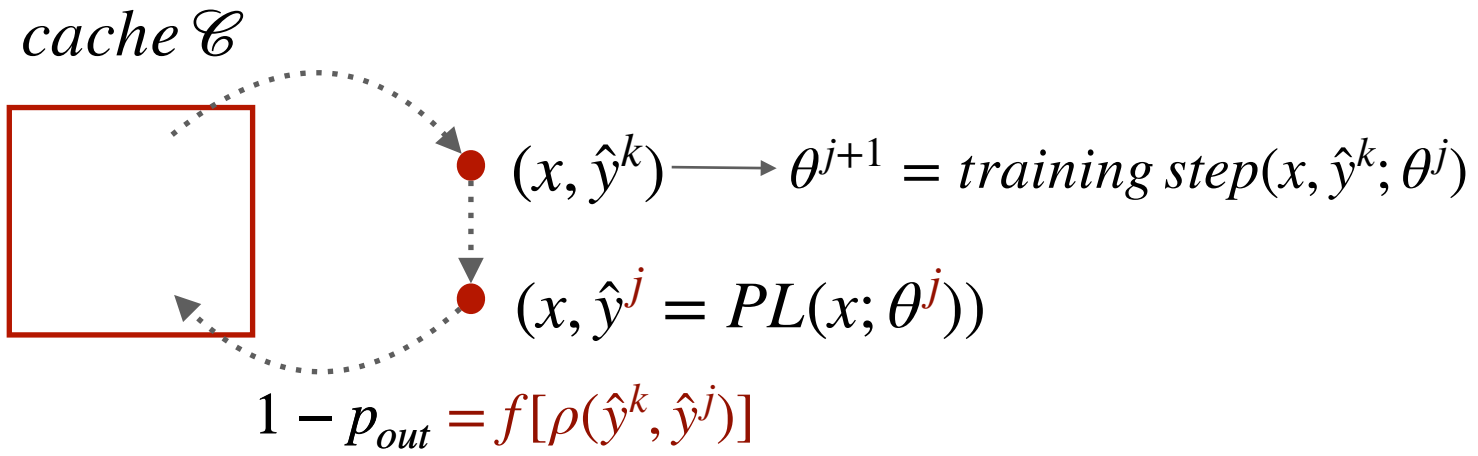
Controlling Cache by PL Evolution



Controlling Cache by PL Evolution

p_{out} depends on edit-distance ρ, e.g.

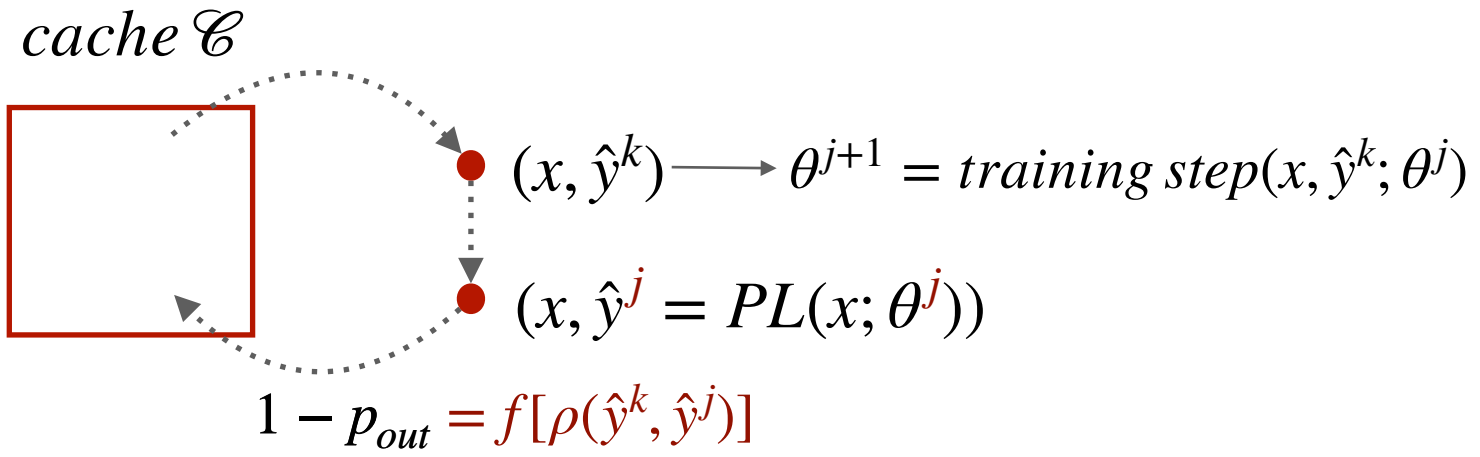
$p_{out} = \rho(\hat{y}^k, \hat{y}^j)$ or
$p_{out} = 1 - \rho(\hat{y}^k, \hat{y}^j)$



Controlling Cache by PL Evolution

p_{out} depends on edit-distance ρ, e.g.

$p_{out} = \rho(\hat{y}^k, \hat{y}^j)$ or
$p_{out} = 1 - \rho(\hat{y}^k, \hat{y}^j)$

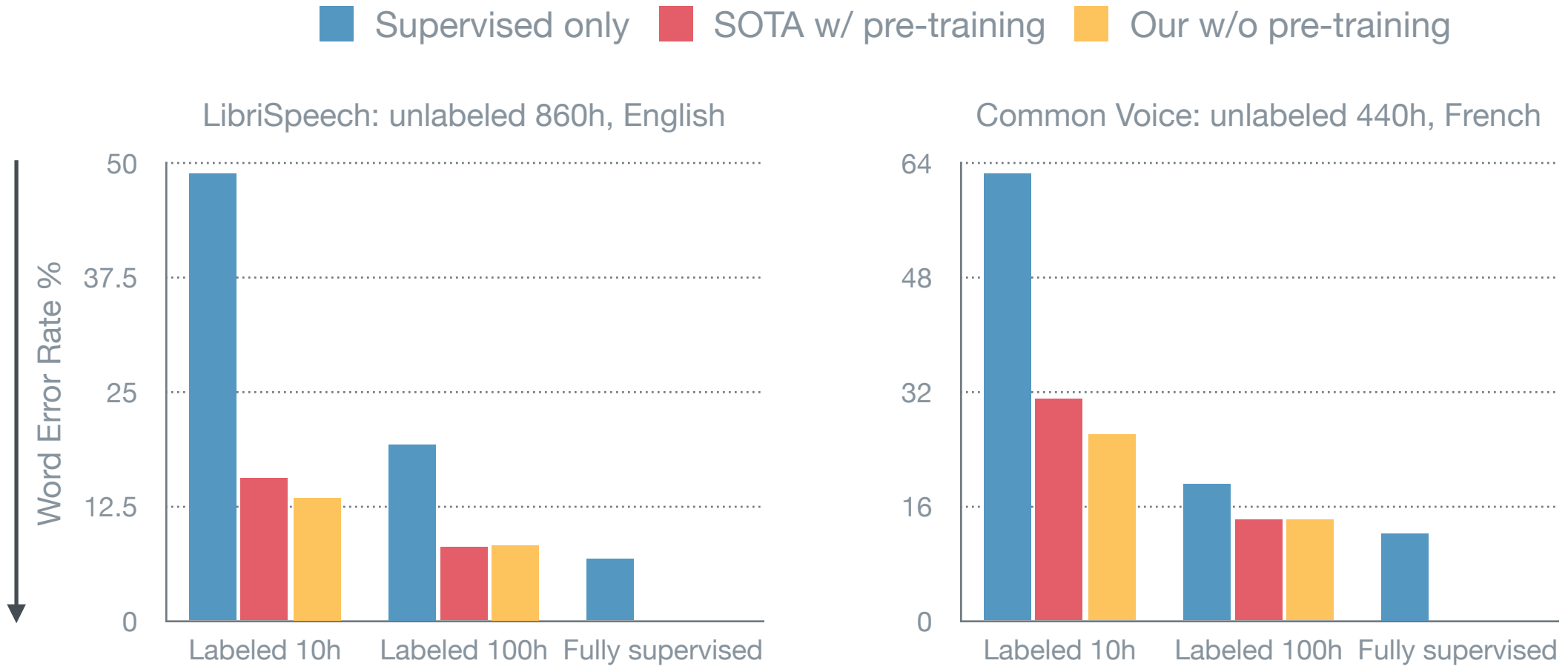


$\hat{y}^k$: *seee yu in Kigoli*

$\hat{y}^j$: *see yu in Kigali*

$$\rho(\hat{y}^k, \hat{y}^j) = \frac{2}{13} \approx 0.15 \implies p_{out} = 0.15 \quad \text{or} \quad p_{out} = 0.85$$

PL from Start is Stable and Matches Pre-Training



Conclusions

- Pseudo-labeling is a simple and effective technique
- Pre-training on labeled data is not necessary for continuous pseudo-labeling
- Ingredients to stabilize training
    - Controlling dynamically when and how we update pseudo-labels
    - Sampling transcriptions with an evolving temperature parameter
    - ...
- Training from the start improves results in low supervision setting while matches in standard setting

Carnegie Mellon University

Work done during Dan Berrebbi internship at Apple MLR team

advised by Tatiana Likhomanenko, Navdeep Jaitly, Ronan Collobert and Samy Bengio

dberrebb@andrew.cmu.edu antares@apple.com