Learning to Self-Train for Semi-Supervised Few-Shot Classification

Xinzhe Li, Qianru Sun, Yaoyao Liu, Shibao Zheng
Qin Zhou, Tat-Seng Chua, Bernt Schiele

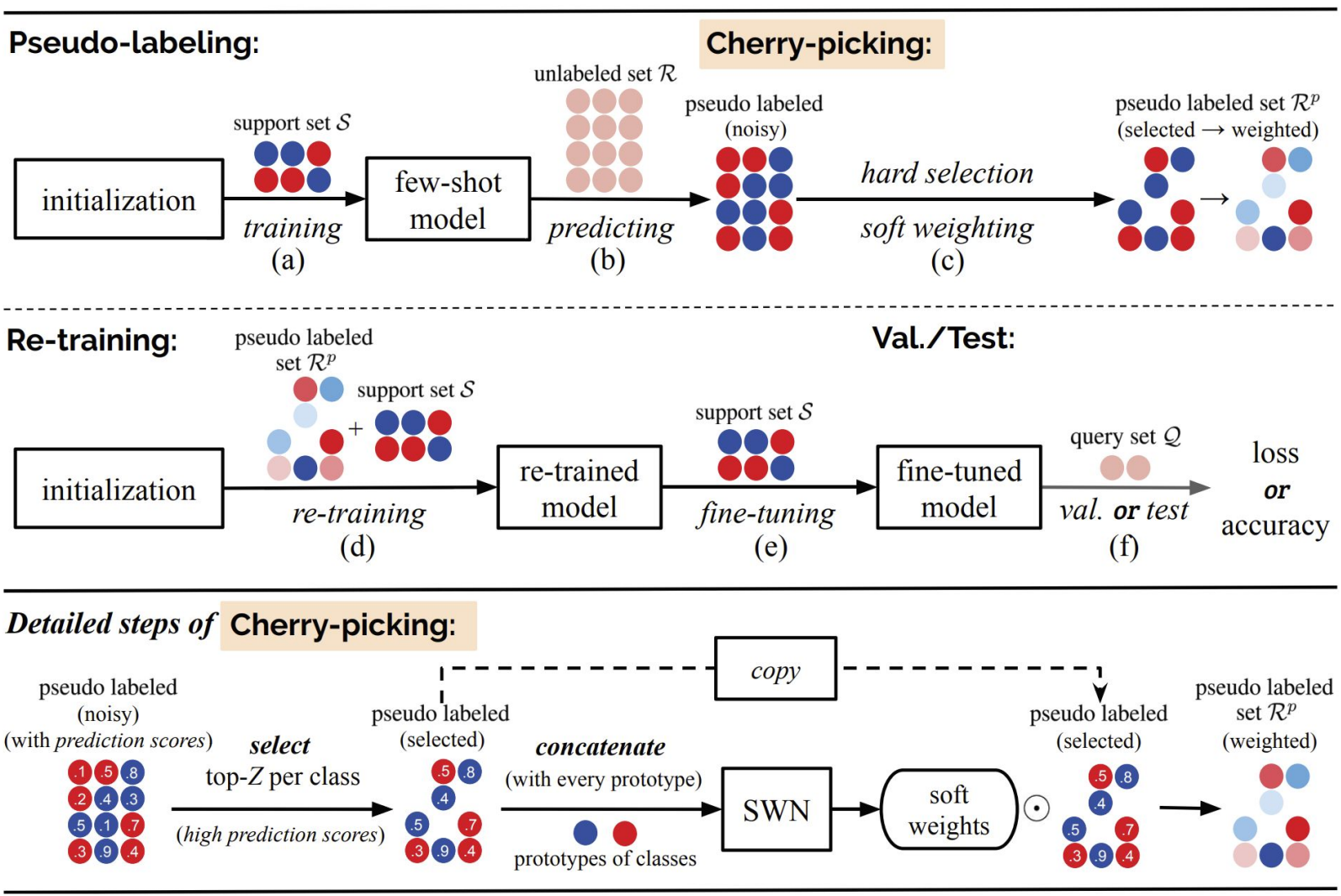
NeurIPS 2019

Motivation

Leverage unlabeled data for FSL

Meta-learn to cherry pick the data

Learning to minimize the noise from unlabeled data


Pseudo-labeling:
Cherry-picking:
initialization
support set $\mathcal{S}$
training
(a)
few-shot model
unlabeled set $\mathcal{R}$
predicting
(b)
pseudo labeled (noisy)
hard selection
soft weighting
(c)
pseudo labeled set $\mathcal{R}^p$ (selected → weighted)
Re-training:
pseudo labeled set $\mathcal{R}^p$
support set $\mathcal{S}$
initialization
re-training
(d)
re-trained model
support set $\mathcal{S}$
fine-tuning
(e)
fine-tuned model
Val./Test:
query set $\mathcal{Q}$
val. or test
(f)
loss or accuracy
Detailed steps of Cherry-picking:
pseudo labeled (noisy) (with prediction scores)
select
top-Z per class
(high prediction scores)
pseudo labeled (selected)
copy
concatenate
(with every prototype)
prototypes of classes
SWN
soft weights
pseudo labeled (selected)
pseudo labeled set $\mathcal{R}^p$ (weighted)

Pseudo-Labeling

Train a fast learner using support set

$$\theta_t \leftarrow \theta_{t-1} - \alpha \nabla_{\theta_{t-1}} L\big(\mathcal{S}; [\Phi_{ss}, \theta_{t-1}]\big)$$

Label the unsupervised dataset

$$Y^{\mathcal{R}} = f_{[\Phi_{ss}, \theta_T]}(\mathcal{R}),$$

Cherry picking

Soft-weight the semi-supervised samples

$$w_{i,c} = f_{\Phi_{swn}}\left(\left[f_{\Phi_{ss}}(x_i); \frac{\sum_k f_{\Phi_{ss}}(x_{c,k})}{K}\right]\right),$$

Self-train

Merge support set and weakly-labeled data and retrain

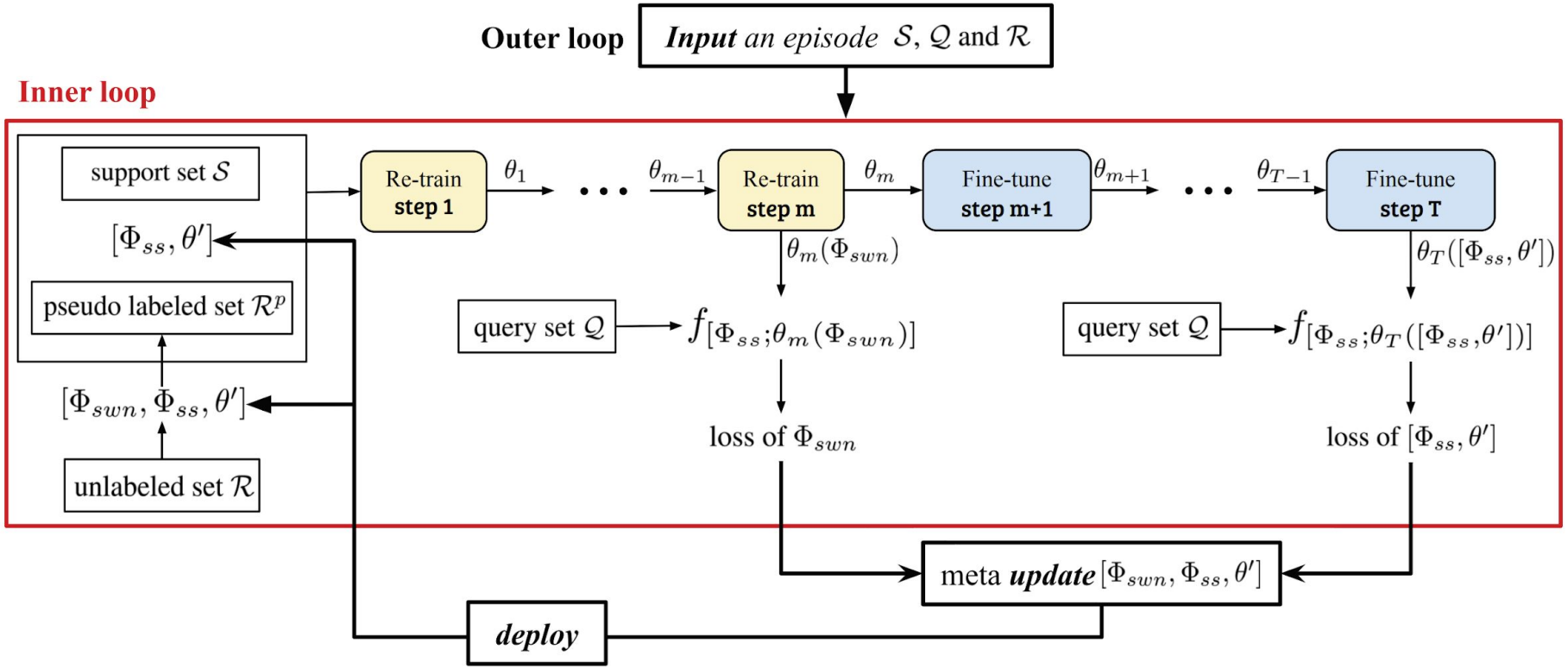
$$\theta_t \leftarrow \theta_{t-1} - \alpha \bigtriangledown_{\theta_{t-1}} L\big(\mathcal{S} \cup \mathcal{R}^p; [\Phi_{swn}, \Phi_{ss}, \theta_{t-1}]\big),$$

$$L\big(\mathcal{S} \cup \mathcal{R}^p; [\Phi_{swn}, \Phi_{ss}, \theta_t]\big) = \begin{cases} L_{ce}\big(f_{[\Phi_{swn}, \Phi_{ss}, \theta_t]}(x_i), y_i\big), \text{ if } (x_i, y_i) \in \mathcal{S}, \\ L_{ce}\big(\mathbf{w}_i \odot f_{[\Phi_{swn}, \Phi_{ss}, \theta_t]}(x_i), y_i\big), \text{ if } (x_i, y_i) \in \mathcal{R}^p, \end{cases}$$

Update params at different time step

$$\begin{aligned} \Phi_{swn} &=: \Phi_{swn} - \beta_1 \bigtriangledown_{\Phi_{swn}} L(\mathcal{Q}; [\Phi_{swn}, \Phi_{ss}, \theta_m]), \\ [\Phi_{ss}, \theta'] &=: [\Phi_{ss}, \theta'] - \beta_2 \bigtriangledown_{[\Phi_{ss}, \theta']} L(\mathcal{Q}; [\Phi_{swn}, \Phi_{ss}, \theta_T]), \end{aligned}$$

Inner loop of selft-train



Result

Few-shot Learning Method		Backbone	miniImageNet (test) 1-shot	5-shot
Data augmentation	Adv. ResNet, [15]	WRN-40 (pre)	55.2	69.6
	Delta-encoder, [29]	VGG-16 (pre)	58.7	73.6
Gradient descent	MAML, [3]	4 CONV	48.70 ± 1.75	63.11 ± 0.92
	Bilevel Programming, [5]	ResNet-12$^{\diamond}$	50.54 ± 0.85	64.53 ± 0.68
	MetaGAN, [42]	ResNet-12	52.71 ± 0.64	68.63 ± 0.67
	adaResNet, [19]	ResNet-12‡	56.88 ± 0.62	71.94 ± 0.57
	LEO, [27]	WRN-28-10 (pre)	61.76 ± 0.08	77.59 ± 0.12
	MTL, [32]	ResNet-12 (pre)	61.2 ± 1.8	75.5 ± 0.9
	MetaOpt-SVM, [10]†	ResNet-12	62.64 ± 0.61	78.63 ± 0.46
LST (Ours)	*recursive*, *hard*, *soft*	ResNet-12 (pre)	**70.1** ± 1.9	**78.7** ± 0.8

Few-shot Learning Method		Backbone	tieredImageNet (test) 1-shot	5-shot
Gradient descent	MAML, [3] (by [13])	ResNet-12	51.67 ± 1.81	70.30 ± 0.08
	LEO, [27]	WRN-28-10 (pre)	66.33 ± 0.05	81.44 ± 0.09
	MTL, [32] (by us)	ResNet-12 (pre)	65.6 ± 1.8	78.6 ± 0.9
	MetaOpt-SVM, [10]†	ResNet-12	65.99 ± 0.72	81.56 ± 0.53
LST (Ours)	*recursive*, *hard*, *soft*	ResNet-12 (pre)	**77.7** ± 1.6	**85.2** ± 0.8