ICCV23 PARIS

IOMatch: Simplifying Open-Set Semi-Supervised Learning with Joint Inliers and Outliers Utilization

Zekun Li[1], Lei Qi[2], Yinghuan Shi[1*], Yang Gao[1]

[1]Nanjing University [2]Southeast University

Background

- **Standard Semi-SL assumptions can be hard to satisfy.**
  - In practice, unlabeled data may contain unseen classes (outliers).
  - Existing Semi-SL methods suffer from open-set unlabeled data.
    - It is impossible to generate correct close-set pseudo-labels for outliers.

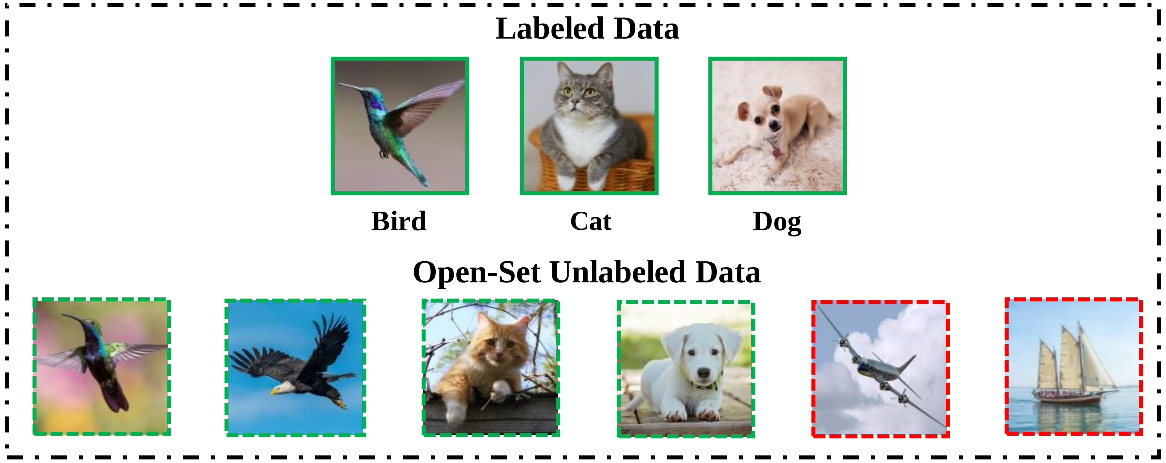
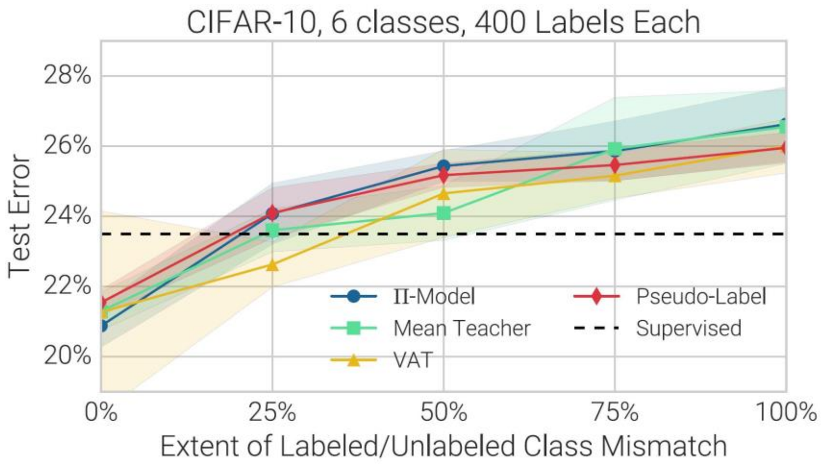


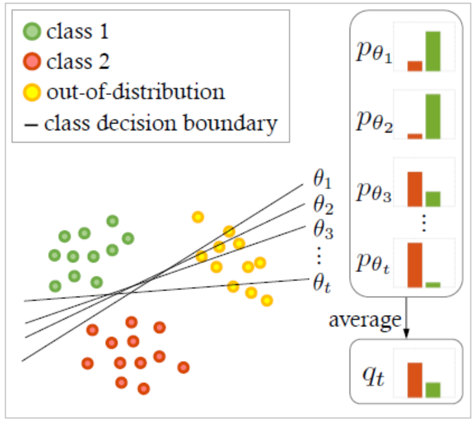
Illustration of Open-Set SSL



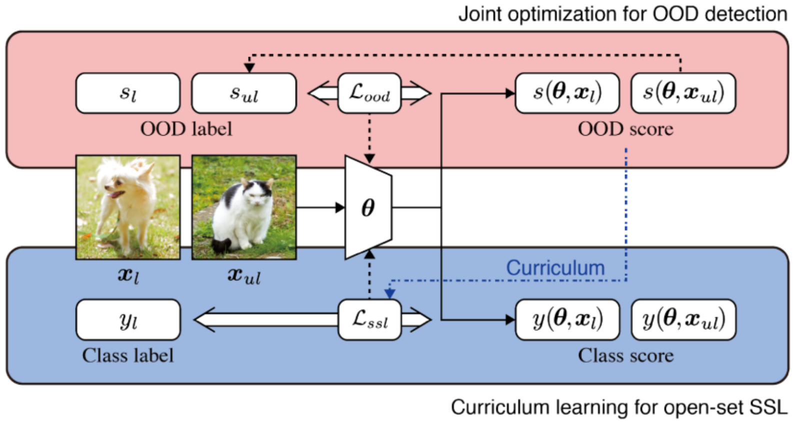
[NeurIPS'18] Oliver *et al.*

Motivation

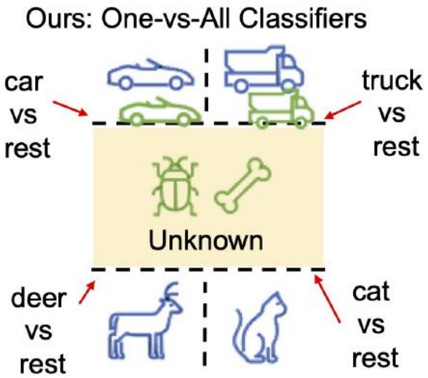
- **Intuition: Outliers are harmful? Remove them first!**
  - It is a common strategy employed in previous works:
    - Detect the outliers first and then filter them out of pseudo-labeling.
    - Detection based on predictions or with additional network modules:



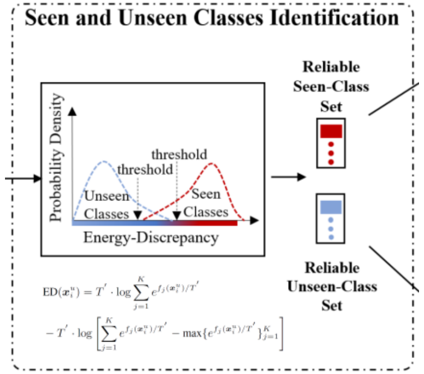
[AAAI'20] Chen *et al.*



[ECCV'20] Yu *et al.*



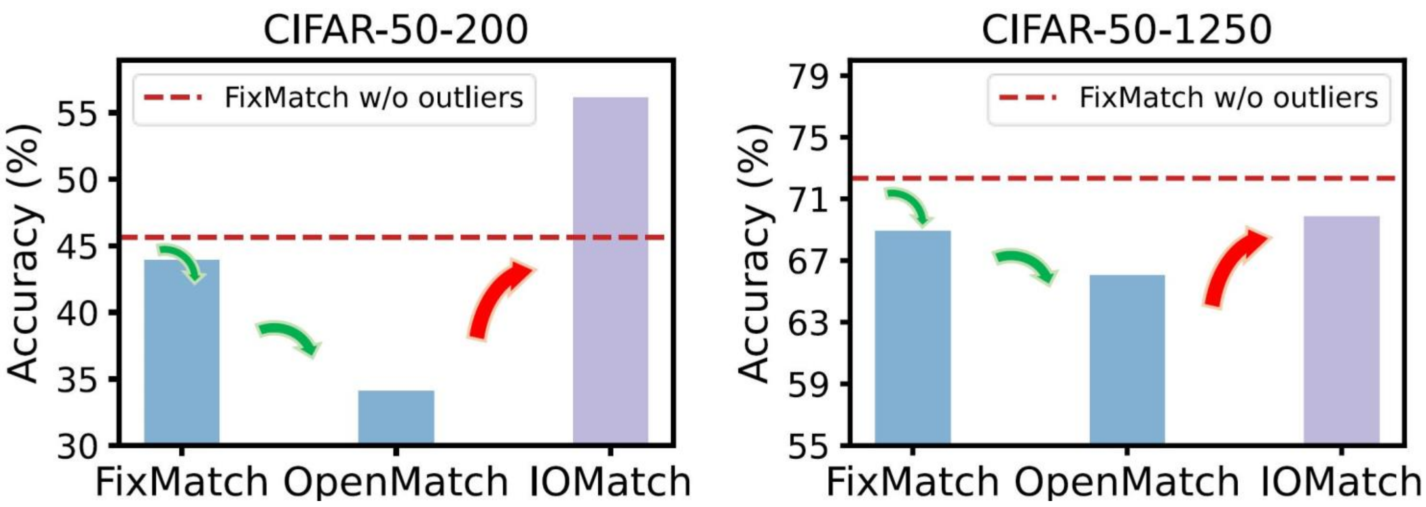
[NeurIPS'21] Saito *et al.*



[CVPR'22] He *et al.*

Motivation

- **The intuitive detect-and-filter strategy can easily fail.**
  - We can hardly obtain a reliable outlier detector at the beginning.
    - Especially when labels are extremely scarce.
  - **An unreliable detector harms more than outliers themselves.**



[1] FixMatch: Sohn *et al.*, NeurIPS'20; [2] OpenMatch: Saito *et al.*, NeurIPS'21

Motivation

- **The intuitive detect-and-filter strategy can easily fail.**
  - We can hardly obtain a reliable outlier detector at the beginning.
    - Especially when labels are extremely scarce.
  - **An unreliable detector harms more than outliers themselves.**
    - Numerous inliers may be wrongly removed.
    - Such errors are difficult to rectify.

Can we utilize open-set unlabeled data
without exactly distinguishing between inliers and outliers?

Approach

- **Key idea: exploit unified open-set targets.**
  - A standard closed-set classifier to predict an unlabeled sample
    - Most likely to belong to which seen class ($c_1/c_2/c_3$)
      - $\boldsymbol{p} = [p_{c_1}, p_{c_2}, p_{c_3}] = [0.7, 0.2, 0.1]$
  - An extra multi-binary classifier to predict
    - Probability of truly belonging to each seen class or not
      - $\boldsymbol{o}_{c_1} = [o_{c_1}, \overline{o_{c_1}}] = [0.4, 0.6]$
      - $\boldsymbol{o}_{c_2} = [o_{c_2}, \overline{o_{c_2}}] = [0.1, 0.9]$
      - $\boldsymbol{o}_{c_3} = [o_{c_3}, \overline{o_{c_3}}] = [0.2, 0.8]$

Labeled Data

Unlabeled Sample

Approach

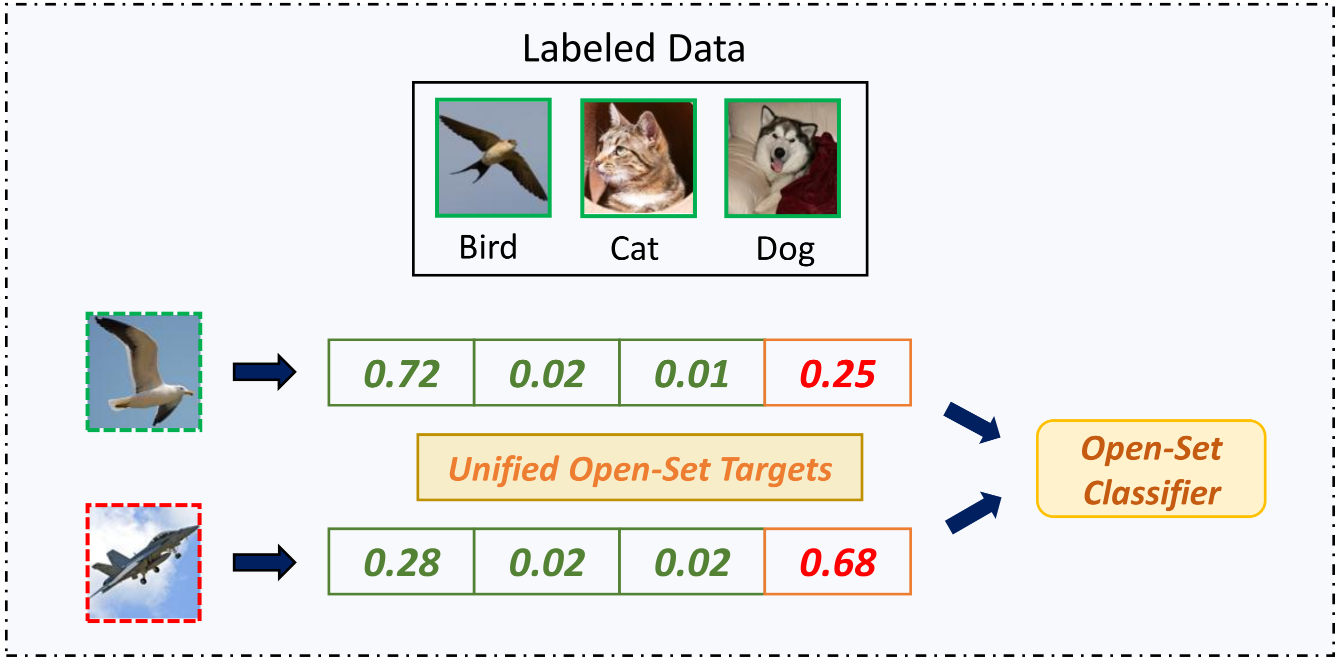
- **Key idea: exploit unified open-set targets.**
  - Fuse these two predictions to estimate the likelihood of a sample
    - Being an inlier of c_1: $p_{c_1} \times o_{c_1} = 0.7 \times 0.4 = 0.28$
    - Being an outlier similar to c_1: $p_{c_1} \times \overline{o_{c_1}} = 0.7 \times 0.6 = 0.42$
    - Same for other seen classes…
      - Being an inlier of $c_1/c_2/c_3$: $[0.28, 0.02, 0.02]$
      - Being an outlier: $0.42 + 0.18 + 0.08 = 0.68$
  - Then we obtain the open-set target:
    - Probability of [Bird, Cat, Dog, Outlier] = $[0.28, 0.02, 0.02, 0.68]$

Labeled Data

Unlabeled Sample

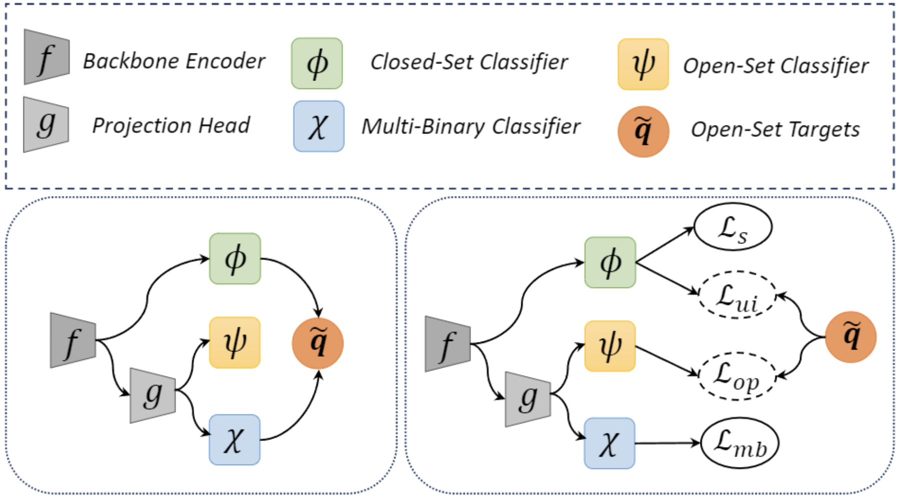
Approach

- **Key idea: exploit unified open-set targets.**
  - Unified open-set targets are produced for both inliers and outliers.
  - Optimize an open-set classifier via pseudo-labeling.



Approach

- **IOMatch demonstrates remarkable simplicity.**
  - All the classifiers in IOMatch are concurrently optimized.
    - No more need for a pre-training (warm-up) stage for an outlier detector.
  - All the learning objectives are cross-entropy losses.
    - Easy for implementation.
    - Easy to tune hyper-parameters.



(a) Produce Open-Set Targets (b) Optimize Whole Framework

Approach

- **IOMatch achieves impressive performance.**
  - Compared with the SOTA standard and open-set Semi-SL methods.
  - For both closed-set and open-set evaluation.

Dataset			CIFAR-10		CIFAR-100					
Class split (Seen / Unseen)			6 / 4		20 / 80		50 / 50		80 / 20	
Number of labels per class			4	25	4	25	4	25	4	25
Standard SSL	MixMatch [3]	NeurIPS'19	43.08 ± 1.79	63.13 ± 0.64	28.13 ± 5.06	51.28 ± 1.45	26.97 ± 0.46	56.93 ± 0.84	28.35 ± 0.83	53.77 ± 0.97
	ReMixMatch [2]	ICLR'20	72.82 ± 1.81	87.08 ± 1.12	36.02 ± 3.56	61.83 ± 0.81	37.57 ± 1.54	65.80 ± 1.33	40.64 ± 2.97	62.90 ± 1.07
	FixMatch [30]	NeurIPS'20	81.58 ± 6.63	92.94 ± 0.80	46.27 ± 0.64	66.45 ± 0.74	48.93 ± 5.05	68.77 ± 0.89	43.06 ± 1.21	64.44 ± 0.51
	CoMatch [20]	ICCV'21	86.08 ± 1.08	92.57 ± 0.47	43.53 ± 3.01	66.82 ± 1.37	43.17 ± 0.55	67.85 ± 1.17	37.89 ± 1.22	62.04 ± 0.08
	FlexMatch [44]	NeurIPS'21	73.34 ± 4.42	86.44 ± 3.72	37.93 ± 4.49	62.68 ± 2.02	44.10 ± 1.88	68.98 ± 0.94	43.44 ± 2.40	64.34 ± 0.64
	SimMatch [47]	CVPR'22	79.84 ± 4.76	90.07 ± 2.44	36.93 ± 5.72	67.23 ± 1.13	51.53 ± 2.02	69.71 ± 1.44	50.32 ± 2.57	**65.68 ± 1.43**
	FreeMatch [37]	ICLR'23	79.26 ± 4.11	92.27 ± 0.15	45.18 ± 8.36	64.62 ± 0.79	50.26 ± 1.92	68.57 ± 0.27	47.34 ± 0.57	64.41 ± 0.55
Open-Set SSL	UASD [7]	AAAI'20	35.25 ± 1.07	56.42 ± 1.34	29.78 ± 4.28	53.78 ± 0.67	29.08 ± 1.44	54.24 ± 1.10	26.41 ± 2.16	50.33 ± 0.62
	DS^{3}L [10]	ICML'20	39.09 ± 1.24	51.83 ± 1.06	19.70 ± 1.98	41.78 ± 1.45	21.62 ± 0.54	47.41 ± 0.61	20.10 ± 0.48	40.51 ± 1.02
	MTCF [42]	ECCV'20	49.15 ± 6.12	74.42 ± 2.95	32.58 ± 3.36	55.93 ± 1.66	35.35 ± 2.39	57.72 ± 0.20	25.40 ± 1.20	54.59 ± 0.49
	T2T [16]	ICCV'21	73.89 ± 1.55	85.69 ± 1.90	44.23 ± 2.27	65.60 ± 0.71	39.31 ± 1.16	68.59 ± 0.92	38.16 ± 0.59	63.86 ± 0.32
	OpenMatch [27]	NeurIPS'21	43.63 ± 3.26	66.27 ± 1.86	37.45 ± 2.67	62.70 ± 1.76	33.74 ± 0.38	66.53 ± 0.54	28.54 ± 1.15	61.23 ± 0.81
	SAFE-STUDENT [14]	CVPR'22	59.28 ± 1.18	77.87 ± 0.14	34.53 ± 0.67	58.07 ± 1.40	35.84 ± 0.86	62.75 ± 0.38	34.17 ± 0.69	57.99 ± 0.34
	IOMatch	**Ours**	**89.68 ± 2.04**	**93.87 ± 0.16**	**53.73 ± 2.12**	**67.28 ± 1.10**	**56.31 ± 2.29**	**69.77 ± 0.58**	**50.83 ± 0.99**	64.75 ± 0.52

Conclusions

- In open-set Semi-SL, it is really challenging, but not mandatory, to exactly identify outliers before pseudo-labeling.
- What truly matters is the idea of joint inliers and outliers utilization.
  - Producing unified open-set targets is just one approach for this.
- We are working towards more realistic Semi-SL!
  - Tackling more practical challenges: imbalanced class distribution, domain shifts, and fine-grained categories…
  - With stronger techniques: self-supervised learning, LLMs, and VLMs…

Looking forward to further discussion!

10:30 am – 12:30 pm

Poster #152 @ Room Nord

Code: https://github.com/nukezil/IOMatch

Paper: https://arxiv.org/abs/2308.13168

Contact me: lizekun@smail.nju.edu.cn