Semi-supervised Learning from Uncurated Echocardiogram Images with Fix-A-Step

Zhe Huang[1], **Mary-Joy Sidhom**[1], **Benjamin S. Wessler**[2], and **Michael C. Hughes**[1]

[1]Dept. of Computer Science, Tufts University, Medford, MA, USA
[2]Division of Cardiology, Tufts Medical Center, Boston, MA, USA

Abstract

Semi-supervised learning (SSL) promises gains in accuracy compared to training classifiers on small labeled datasets by also training on many unlabeled images. In real medical applications, unlabeled images are often *uncurated* and thus possibly different from the labeled set in represented classes. Unfortunately, modern deep SSL often makes accuracy worse when given uncurated unlabeled data. Recent remedies suggest filtering approaches that detect out-of-distribution (OOD) unlabeled examples and then discard or downweight them. Instead, we view all unlabeled examples as potentially helpful. We introduce a procedure called Fix-A-Step that can improve heldout accuracy of common deep SSL methods despite lack of curation. Our first insight is that unlabeled data, *even OOD*, can usefully inform augmentations of labeled data. Our second innovation is to modify gradient descent updates to prevent following the multi-task SSL loss from hurting labeled-set accuracy. Though our method is simpler than alternatives, we show consistent accuracy gains on a common CIFAR-10 benchmark across all levels of contamination. We further create a new medical benchmark for robust SSL called Heart2Heart[1], where the task is recognizing the view type of ultrasound images of the heart. On Heart2Heart, Fix-A-Step can learn from 353,500 truly uncurated unlabeled images to deliver gains that generalize across hospitals.

1 Introduction

Semi-supervised learning (SSL) [30, 26] is a promising approach to medical imaging problems where labeled images are expensive to acquire while unlabeled images are more affordable. SSL methods can train a classifier from the union of a small labeled dataset and a large unlabeled dataset, often claiming to deliver accuracies on par with a conventionally-trained classifier given many more labels.

Recent SSL developments have achieved excellent results on standard benchmarks such as SVHN [20] or CIFAR-10 [13]. Unfortunately, these results are too optimistic, since the unlabeled set are carefully curated by dropping known labels. In real tasks, the unlabeled set is often *uncurated* due to limited annotation time, and thus might differ from the labeled set in terms of represented classes or class frequencies, among other differences. Recent work [21] shows that off-the-shelf SSL performance deteriorates when unlabeled contents differ in label composition from the labeled set.

Many recent SSL methods try to be robust to uncurated unlabeled data [12, 7, 28, 5, 23]. This line of work is sometimes called "safe" or "open-set" SSL. Broadly, most of these methods follow the same intuitive direction: learn to identify examples in the unlabeled set that are *out-of-distribution* (OOD), remove or downweight these, and train on the remainder. However, we find this OOD-removal paradigm *neglects the potential value of OOD samples*, and thus might limit prediction quality.

[1]Code for Fix-A-Step SSL and our Heart2Heart benchmark: https://github.com/tufts-ml/fix-a-step

Accepted paper at the Medical Imaging Meets NeurIPS Workshop, 2022

This study makes 3 contributions toward more robust SSL methods. First, we challenge the dominant paradigm that handles uncurated unlabeled sets by filtering out OOD examples. Our experiments suggest that even perfect OOD filtering (which is unrealistic in practice) does not perform well. Instead, we argue for a **new paradigm: OOD images from uncurated unlabeled sets are potentially helpful**. Second, following this paradigm we introduce **a new SSL training procedure called *Fix-A-Step*** designed for robust SSL classification even when given uncurated unlabeled sets. Our method improves predictions on better than alternative methods while being substantially simpler. Finally, we offer a **new SSL benchmark using real uncurated medical images that can assess cross-hospital generalization**. Using three inter-operable open-access datasets TMED [9, 10], CAMUS [15], and Unity [8], we pursue a clinically-relevant problem: recognizing the view type of an echocardiogram image of the heart. Future methods for learning from limited data can follow our reproducible code (link on page 1). An extended manuscript describing this work in more detail is available [11].

2 Method

The dominant approaches for semi-supervised training of deep image classifiers today continue to modify standard objectives for discriminative neural nets by adding a regularization term using unlabeled data [19, 24]. This approach trains a neural net via multi-task optimization:

$$\min_w \quad \sum_{x,y \in \mathcal{D}^L} \ell^L(y, f_w(x)) + \lambda \sum_{x \in \mathcal{D}^U} \ell^U(x; w) \tag{1}$$

Let w denote weight parameters, x input features, y labels, and f the probabilistic classifier output of the network. The labeled-set loss ℓ^L is *cross entropy*; the unlabeled-set loss ℓ^U is method-specific.

Our proposed method, Fix-A-Step, follows this multi-task approach, with two key modifications to how parameters are updated during gradient descent, detailed below. Fix-A-Step can "fix" (improve robustness to uncurated data) many common SSL methods with different losses ℓ^U.

First, in the *augmentation* phase our insight is the unlabeled set might be *helpful* for creating useful augmentations, even when uncurated, by injecting realistic diversity (motivating example in App. A). Inspired by MixMatch [1], we transform each labeled pair (x, y) using another pair (x', y') drawn either from the labeled set or the unlabeled set (if only x' is known we apply soft pseudo-label predictions for y', see Alg. D.2). Given x, y and x', y', we build a new labeled pair $\tilde{x}, \tilde{y}$ via MixUp [29] interpolation (see Alg. D.3), then use that pair to compute the labeled loss. While the success of MixUp for standard SSL via MixMatch [1] is widely known, it is under-explored whether this technique is beneficial with *uncurated* data. We further show that MixMatch alone is not enough (see Fig. 1).

Second, in the *step direction* phase, we prioritize the labeled loss in parameter updates, only using the unlabeled loss if it improves the labeled loss. At each batch, we compute two gradient vectors, one for each term in the loss: Let $g^L = \nabla_w \ell^L$ and let $g^U = \nabla_w \ell^U$. The update for weights w is then

$$w \leftarrow \begin{cases} w - \epsilon(g^L + \lambda g^U) & \text{if } \sum_d g^L_d g^U_d > 0 \\ w - \epsilon g^L & \text{otherwise.} \end{cases} \tag{2}$$

where $\epsilon > 0$ is a step size. In the top case, we do the standard steepest descent update that minimizes the two-term SSL objective in Eq. (1). In the bottom case, we perform an alternative update, using only the labeled-term gradient. This two-case construction tries to ensure that SSL learning does not harm labeled set performance. Formally, we can show that each possible update in Eq. (2) adjusts weights w in a *descent direction* for the labeled set loss at the current minibatch (proof in E). Our step modification phase is inspired by the *Transfer-Interference trade-off* [22, 17] in continual learning. Similar ideas have also been explored in multi-task learning [6] where the goal is maximize the performance of a "main" task no matter what, while using an "auxiliary" task if helpful.

Geometric intuition Recall that two vectors g^L and g^U have positive inner product (top case) only if the angle between the vectors is below 90 degrees. At angles larger than 90 (bottom case), g^L and g^U are pointing in different directions, and minimizing the unlabeled loss would hinder the labeled loss. In SSL, we care most about (heldout) classifier accuracy. Any improvement on the unlabeled loss is useful only if it helps improve accuracy. When g^U points in a different direction than g^L, our update ignores the unlabeled gradient and updating parameters using only g^L.

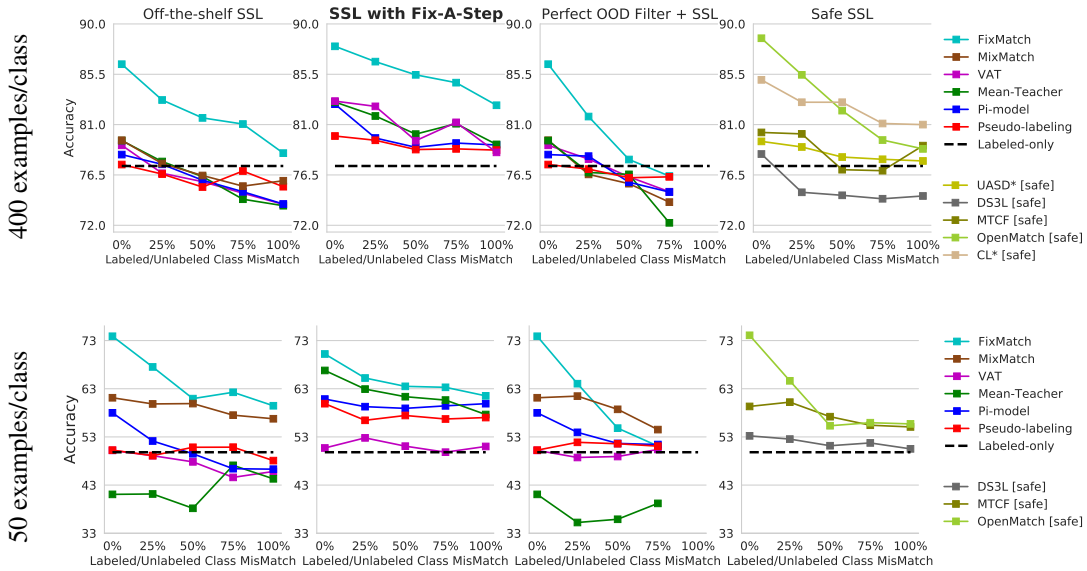

Figure 1: **Accuracy on CIFAR-10 6 animal task.** Accuracy on test images of animals (y-axis) as unlabeled set mismatch (percentage of non-animal classes represented, x-axis) increases. Training details in App C.1.

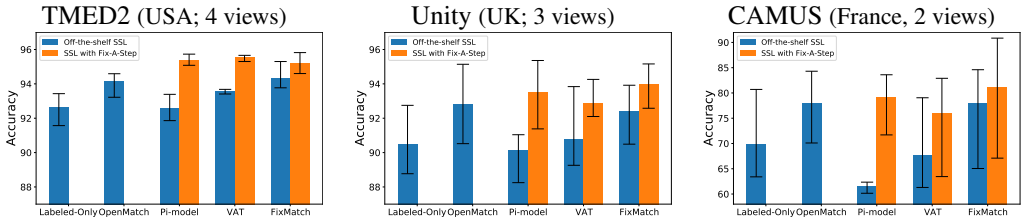

Figure 2: **Balanced accuracy for echocardiogram view classification (Heart2Heart benchmark).** *Left:* Evaluation on heldout TMED-2 images. *Center:* Evaluation on Unity dataset (17 sites in the UK). *Right:* Evaluation on CAMUS dataset (1 site in France). Methods are trained using only TMED-2. Details in App C.

3 Experimental Results

CIFAR-10 Experiments. We evaluate on CIFAR-10 "6-animal" task designed by [21]. We build a labeled set of the 6 animal classes (dog, cat, horse, frog, deer, bird) in CIFAR-10, across two training set sizes: 50 labeled images per class and 400 per class. We form an unlabeled set of $\sim$4100 images/class from 4 selected classes, some animal and some non-animal (car, truck, ship, airplane). In Fig. 1 We compared to 6 SSL methods (Pi-Model [14], Mean-Teacher [25], Pseudo-label [16], VAT [19], MixMatch [1], and FixMatch [24]), a "labeled only" baseline and five state-of-the-art open-set/safe SSL methods: UASD [5], DS3L [7], MTCF [28], OpenMatch [23] and Curriculum-labeling [3]. Test accuracy is reported on the held-out test set of 6 animal classes. The *perfect OOD filtering* column shows the best-possible case for methods under the OOD-is-harmful paradigm.

The key takeaways from Fig. 1 are: **1. Fix-A-Step improves all SSL methods in almost all settings**, despite its relative simplicity. **2. Perfect OOD filtering is not enough.** The gains of our Fix-A-Step over this best-case suggest that our *OOD-is-helpful* paradigm should be prioritized over OOD filtering.

Heart2Heart Experiments. In pursuit of realistic evaluation, we propose a clinically-relevant SSL task called *Heart2Heart*. The key question is: can we generalize classifiers of ultrasound images of the heart from one hospital to images from different hospitals in other countries. We train classifiers to recognize view (PLAX, PSAX, A4C, or A2C) using images from TMED-2 [10], including 56 labeled studies as well as 353,500 *uncurated* unlabeled images. We report *balanced accuracy* on TMED-2's test set, as well as *external generalization balanced accuracy* on 7231 available PLAX, A2C, and A4C images from the Unity dataset (17 hospitals in the UK) [8], and 2000 images (A2C and A4C views only) in the CAMUS dataset from a hospital in France [15]. Fig. 2 shows that **Fix-A-Step yields gains across all tested SSL methods (Pi-Model, VAT, FixMatch)**. With Fix-A-Step, all methods convincingly *outperform* the labeled-only baseline. External evaluation on Unity and

CAMUS further suggests that **these gains can transfer to new patients at different hospitals**. Compared to OpenMatch, a state-of-the-art safe SSL method, Fix-A-Step yields better accuracy while being much simpler and faster to train (2-3x speedup, see App B.1).

4 Discussion and Broader Impacts

This paper makes three contributions to deep SSL image classification. First, we argue that uncurated or OOD data in the unlabeled set can be quite *helpful*, reinforcing other parallel work that finds OOD data for SSL "not completely useless" [12]. Second, we propose Fix-A-Step, a new method remarkable for its simplicity as well as its effectiveness. Finally, we propose a realistic medical imaging benchmark for SSL called Heart2Heart to inspire robust studies of clinical model transportability across global populations. Throughout, our methodology emphasizes *simplicity*: Fix-A-Step can repair many different SSL methods without introducing any new neural networks, loss functions, or expensive optimization procedures.

Semi-supervised image classification has many positive applications. Indeed, work on SSL is often specifically motivated by the promise of improved efficiency in environments where labels are expensive and time-consuming as is the case in medical imaging [9, 18]. However, care must be taken to ensure that automated methods actually benefit patients and do not widen current disparities [4]. Our present Heart2Heart evaluations are an important step beyond single-center evaluations though do not reflect the true geographic and racial diversity of many patient populations. While all images used in our Heart2Heart task are completely de-identified and come from public open-access datasets (and thus did not require ethics review), we stress the responsibility we carry as researchers to protect the best interests of the individuals who contributed data.

Acknowledgments and Disclosure of Funding

Authors ZH, BSW, and MCH gratefully acknowledge financial support from the Pilot Studies Program at the Tufts Clinical and Translational Science Institute (Tufts CTSI NIH CTSA UL1TR002544). Author MJS is partially supported by the Tufts Summer Scholars program. We gratefully acknowledge computing hardware support from the U.S. National Science Foundation under grant NSF OAC-2018149.

References

[1] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, 2019. URL http://arxiv.org/abs/1905.02249.

[2] S. P. Boyd and L. Vandenberghe. Sec. 9.2: Descent methods. In *Convex Optimization*. Cambridge University Press, Cambridge, UK ; New York, 2004.

[3] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6912–6920, 2021.

[4] L. A. Celi, J. Cellini, M.-L. Charpignon, E. C. Dee, F. Dernoncourt, R. Eber, W. G. Mitchell, L. Moukheiber, J. Schirmer, et al. Sources of bias in artificial intelligence that perpetuate healthcare disparities—A global review. *PLOS Digital Health*, 1(3):e0000022, 2022.

[5] Y. Chen, X. Zhu, W. Li, and S. Gong. Semi-Supervised Learning under Class Distribution Mismatch. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3569–3576, 2020.

[6] Y. Du, W. M. Czarnecki, S. M. Jayakumar, M. Farajtabar, R. Pascanu, and B. Lakshminarayanan. Adapting Auxiliary Losses Using Gradient Similarity, 2020.

[7] L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou. Safe Deep Semi-Supervised Learning for Unseen-Class Unlabeled Data. In *International Conference on Machine Learning*, page 10, 2020. URL http://proceedings.mlr.press/v119/guo20i/guo20i.pdf.

[8] J. P. Howard, C. C. Stowell, G. D. Cole, K. Ananthan, C. D. Demetrescu, K. Pearce, R. Rajani, J. Sehmi, K. Vimalesvaran, et al. Automated Left Ventricular Dimension Assessment Using Artificial Intelligence Developed and Validated by a UK-Wide Collaborative. *Circulation: Cardiovascular Imaging*, 14(5):e011951, 2021.

[9] Z. Huang, G. Long, B. Wessler, and M. C. Hughes. A New Semi-supervised Learning Benchmark for Classifying View and Diagnosing Aortic Stenosis from Echocardiograms. In *Proceedings of the 6th Machine Learning for Healthcare Conference*. PMLR, 2021. URL https://proceedings.mlr.press/v149/huang21a.html.

[10] Z. Huang, G. Long, B. S. Wessler, and M. C. Hughes. TMED 2: A Dataset for Semi-Supervised Classification of Echocardiograms. In *In DataPerf: Benchmarking Data for Data-Centric AI Workshop*, 2022. URL https://tmed.cs.tufts.edu/papers/HuangEtAl_TMED2_DataPerf_2022.pdf.

[11] Z. Huang, M.-J. Sidhom, B. S. Wessler, and M. C. Hughes. Fix-a-step: Effective semi-supervised learning from uncurated unlabeled sets. *arXiv preprint arXiv:2208.11870*, 2022. URL https://arxiv.org/abs/2208.11870.

[12] Z. Huang, J. Yang, and C. Gong. They are Not Completely Useless: Towards Recycling Transferable Unlabeled Data for Class-Mismatched Semi-Supervised Learning. *IEEE Transactions on Multimedia*, pages 1–1, 2022.

[13] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[14] S. Laine and T. Aila. Temporal Ensembling for Semi-Supervised Learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/pdf?id=BJ6oOfqge.

[15] S. Leclerc, E. Smistad, J. Pedrosa, A. Østvik, F. Cervenansky, F. Espinosa, T. Espeland, E. A. R. Berg, P.-M. Jodoin, et al. Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography. *IEEE Transactions on Medical Imaging*, 38(9):2198–2210, 2019.

[16] D.-H. Lee. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In *Workshop on Challenges in Representation Learning at ICML*, number 2, 2013. URL http://deeplearning.net/wp-content/uploads/2013/03/pseudo_label_final.pdf.

[17] D. Lopez-Paz and M. Ranzato. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems*, page 10, 2017. URL https://proceedings.neurips.cc/paper/2017/file/f87522788a2be2d171666752f97ddebb-Paper.pdf.

[18] A. Madani, J. R. Ong, A. Tibrewal, and M. R. Mofrad. Deep echocardiography: Data-efficient supervised and semi-supervised deep learning towards automated diagnosis of cardiac disease. *NPJ digital medicine*, 1(1):1–11, 2018.

[19] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019. URL https://ieeexplore.ieee.org/document/8417973/.

[20] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. URL http://ufldl.stanford.edu/housenumbers.

[21] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, 2018. URL https://papers.nips.cc/paper/2018/file/c1fea270c48e8079d8ddf7d06d26ab52-Paper.pdf.

[22] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

[23] K. Saito, D. Kim, and K. Saenko. OpenMatch: Open-set Consistency Regularization for Semi-supervised Learning with Outliers. In *Advances in Neural Information Processing Systems*, page 12, 2021. URL https://proceedings.neurips.cc/paper/2021/file/da11e8cd1811acb79ccf0fd62cd58f86-Paper.pdf.

[24] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, C.-L. Li, et al. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/06964dce9addb1c5cb5d6e3d9838f733-Paper.pdf.

[25] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30:1195–1204, 2017.

[26] J. E. van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

[27] N. Wu, Z. Huang, Y. Shen, J. Park, J. Phang, T. Makino, S. Gene Kim, K. Cho, L. Heacock, L. Moy, et al. Reducing false-positive biopsies using deep neural networks that utilize both local and global image context of screening mammograms. *Journal of Digital Imaging*, 34(6): 1414–1423, 2021.

[28] Q. Yu, D. Ikami, G. Irie, and K. Aizawa. Multi-Task Curriculum Framework for Open-Set Semi-Supervised Learning. In *European Conference on Computer Vision (ECCV)*. arXiv, 2020.

[29] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. Mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[30] X. Zhu. Semi-Supervised Learning Literature Survey. Technical Report Technical Report 1530, Department of Computer Science, University of Wisconsin Madison., 2005.

Appendices

A Illustration: OOD unlabeled data can be helpful

We motivate the hypothesis that unlabeled data even from out-of-distribution (OOD) classes could be useful by an experiment testing the off-the-shelf performance of MixMatch [1] on the CIFAR-10 6 animal task (with 400 images per class in training set). We compare MixMatch with and without perfect OOD filtering under three mismatch levels $\zeta = 25\%, 50\%$ and 75%. Results are shown in Fig. A.1. *Counter-intuitively* we see that perfect OOD Filtering leads to clearly worse performance **at all mismatch levels**. This finding appears robust across 5 random train/test splits. This result suggests to us that unlabeled data, even with OOD classes, could be useful via MixMatch style augmentation. Note that our suggested Fix-A-Step procedure provides further safeguards via the gradient step modification, which are not used in Fig. A.1.

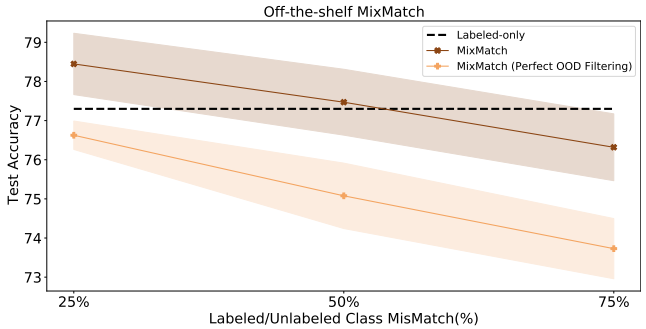

Figure A.1: **CIFAR-10 6-animal, 400 examples/class**. Results average across 5 train/test splits (shaded area shows standard deviation). Both methods use the same hyper-parameters for fair comparison.

B Comparison of computation cost and performance on TMED2

Methods	split0		split1		split2	
	Acc	Runtime	Acc	Runtime	Acc	Runtime
Pi-model+Fix-A-Step	95.33	233	95.08	240	95.73	218
VAT+Fix-A-Step	95.58	392	95.30	343	95.66	356
OpenMatch	94.54	1244	94.59	1282	93.22	879

Table B.1: **Comparison of runtime and test balanced accuracy on TMED-2 view classification task**. Runtime in minutes. Each model is trained on a Nvidia A100. In practice, we found OpenMatch converges slower than alternatives compared, we thus train about 2x more iterations for OpenMatch (otherwise its accuracy performance would be worse).

C Training Details

Hyperparameters for CIFAR-10 experiments. Table C.1 lists the experimental settings (dataset sizes, etc.) and hyperparameters used for all CIFAR-10 experiments. We did *not* tune any hyperparameters specifically for Fix-A-Step. Each model is train on a Nvidia A-100 GPU.

BASIC SETTINGS CIFAR-10

TRAIN LABELED SET SIZE	2400/300
TRAIN UNLABELED SET SIZE	16400/17800
VALIDATION SET SIZE	3000
TEST SET SIZE	6000

Labeled only

Labeled Batch size	64
Learning rate	3e-3
Weight decay	2e-3

VAT

Labeled batch size	64
Unlabeled batch size	64
Learning rate	3e-2
Weight decay	4e-5
Max consistency coefficient	0.3
Unlabeled loss warmup iterations	419430
Unlabeled loss warmup schedule	linear
VAT ξ	1e-6
VAT ϵ	6

Pseudo-label

Labeled batch size	64
Unlabeled batch size	64
Learning rate	3e-2
Weight decay	5e-4
Max consistency coefficient	10.0
Unlabeled loss warmup iterations	419430
Unlabeled loss warmup schedule	linear

Mean Teacher

Labeled batch size	64
Unlabeled batch size	64
Learning rate	3e-2
Weight decay	5e-4
Max consistency coefficient	50.0
Unlabeled loss warmup iterations	419430
Unlabeled loss warmup schedule	linear

Pi-Model

Labeled batch size	64
Unlabeled batch size	64
Learning rate	3e-2
Weight decay	5e-4
Max consistency coefficient	10.0
Unlabeled loss warmup iterations	419430
Unlabeled loss warmup schedule	linear

MixMatch

Labeled batch size	64
Unlabeled batch size	64
Learning rate	3e-2
Weight decay	4e-5
Max consistency coefficient	75.0
Unlabeled loss warmup iterations	1048576
Unlabeled loss warmup schedule	linear
Sharpening temperature	0.5
Beta shape α	0.75

FixMatch

Labeled batch size	64
Unlabeled batch size	448
Learning rate	3e-2
Weight decay	5e-4
Max consistency coefficient	1.0
Unlabeled loss warmup iterations	No warmup
Unlabeled loss warmup schedule	No warmup
Sharpening temperature	1.0
Pseudo-label threshold	0.95

Table C.1: **Hyperparameters used for CIFAR-10 experiments.** All settings represent the recommended defaults suggested in implementations by original authors for the 400 examples/class setting. We did *not* tune any hyperparameters specifically for Fix-A-Step.

Hyperparameters for Heart2Heart

Hyper-parameters are only tuned for the supervised-only baseline and the non Fix-A-Step version of the Pi-model, VAT and FixMatch. We ran 100 trials[2] of Tree-structured Parzen Estimator (TPE) based black box optimization using an open source AutoML toolkit[3] for each algorithm and each data split. The chosen hyper-parameters are then directly applied to Fix-A-Step without retuning. After hyper-parameter selection, each algorithm is then trained for 1000 epochs, the balanced test accuracy at maximum validation balanced accuracy is then reported.

Labeled-only: we search learning rate in $\{0.001, 0.003, 0.01, 0.03, 0.1, 0.3\}$, weight decay in $\{0.0, 0.00005, 0.0005, 0.005, 0.05\}$, optimizer in $\{\text{Adam}, \text{SGD}\}$, learning rate schedule in $\{\text{Fixed}, \text{Cosine}\}$. Batch size is set to 64.

Pi-model: We search learning rate in $\{0.003, 0.01, 0.03, 0.1\}$, weight decay in $\{0.0, 0.0005, 0.005, 0.05\}$, optimizer in $\{\text{Adam}, \text{SGD}\}$, learning rate schedule in $\{\text{Fixed}, \text{Cosine}\}$, Max consistency coefficient in $\{1.0, 5.0, 10.0, 20.0, 100.0\}$, unlabeled loss warmup iterations in $\{0, 17000, 34000\}$. Labeled batch size is set to 64 and unlabeled batch size is set to 64.

VAT: We search learning rate in $\{0.0002, 0.0006, 0.002, 0.006\}$, weight decay in $\{0.000004, 0.00004, 0.0004\}$, optimizer in $\{\text{Adam}, \text{SGD}\}$, learning rate schedule in $\{\text{Fixed}, \text{Cosine}\}$, Max consistency coefficient in $\{0.3, 0.1, 0.9, 0.03, 3\}$, unlabeled loss warmup iterations in $\{0, 17000, 34000\}$. Labeled batch size is set to 64, unlabeled batch size is set to 64. ξ is set to 0.000001 and ϵ is set to 6.

FixMatch: We search learning rate in $\{0.003, 0.01, 0.03, 0.1\}$, weight decay in $\{0.0005, 0.005, 0.05\}$, optimizer in $\{\text{Adam}, \text{SGD}\}$, learning rate schedule in $\{\text{Fixed}, \text{Cosine}\}$, Max consistency coefficient in $\{0.5, 1.0, 5.0, 10.0\}$, Labeled batch size is set to 64, unlabeled batch size is set to 320. We set sharpening temperature to 1.0 and pseudo-label threshold is set to 0.95 (as in CIFAR experiments).

Labeled loss implementation: Weighted cross entropy

On many realistic SSL classification tasks, even the labeled set will have noticeably *imbalanced* class frequencies. For example, in the TMED-2 view labels, the four view types (PLAX, PSAX, A4C, A2C) differ in the number of available examples, with the rarest class (A2C) roughly 3x less common than the most common class (PLAX). To counteract the effect of class imbalance, we use weighted cross-entropy for labeled loss, following prior works [9, 27]. Let integer $c \in \{1, 2, \dots C\}$ index the classes in the labeled set, and let N_c denote the number of images for class c. Then when we compute the labeled loss ℓ^L, we assign a weight $\omega_c > 0$ to the true class c that is inversely proportional to the number of images N_c of the class in the training set:

$$\ell^L(x, c; w) = -\omega_c \log f_w(x)[c], \qquad \omega_c = \frac{\prod_{k \neq c} N_k}{\sum_{j=1}^{C} \prod_{k \neq j} N_k} \iff \omega_c \propto \frac{1}{N_c} \tag{3}$$

Here c denotes the integer index of the true class corresponding to image x, w denotes the neural network weight parameters, and $f_w(x)[c]$ denotes the c-th entry of the softmax output vector produced by the neural network classifier.

Cosine-annealing of learning rate.

We found that several baselines were notably improved using the cosine-annealing schedule of learning rate suggested by [24]. Cosine-annealing sets the learning rate at iteration i to $\eta cos(\frac{7\pi i}{16I})$, where η is the initial learning rate, and I is the total iterations.

To be extra careful, we tried to allow all open-set/safe SSL baselines to also benefit from cosine annealing.

[2] in practice, for each trial we train for only 180 epochs to speed up the hyper-parameters selection process

[3] https://github.com/microsoft/nni

- MTCF is trained using Adam following the author's implementation [28]. Although the author did not originally. use cosine learning rate schedule, we found that adding cosine learning rate schedule substantially improve MTCF's performance. We thus report the performance for MTCF *with cosine annealing*.
- DS3L is trained using Adam following the author's implementation [7]. We tried to add Cosine learning rate to DS3L but result in worse performance. We thus report the performance for DS3L without cosine learning rate.

D Pseudo-code for Fix-A-Step and Subprocedures

Here, we provide implementation details of Fix-A-Step training (Alg. D.1). Submodules AUG+SOFTLABEL D.2 and MIXMATCHAUG D.3 were originally proposed by MixMatch [1] under no contamination SSL settings.

Algorithm D.1: Fix-A-Step Training

Input: Labeled set $\mathcal{D}^L$, Unlabeled set $\mathcal{D}^U$ (uncurated)
Output: Trained weights w^*
Hyperparameters (†*: unique to Fix-A-Step*)

- Sharpening temperature $\tau{>}0$ for SOFTPSEUDOLABEL†
- Shape $\alpha{>}0$ of Beta(α, α) dist. for MIXMATCHAUG†
- Max. iterations I, Step size ϵ, Initial weights w
- Unlabeled-loss weight per iter $\lambda_1, \dots \lambda_I$

for iter $i \in 1, 2, \dots I$ until converged **do**
  $\{\mathbf{x}^L, \mathbf{y}^L\}, \mathbf{x}^U \leftarrow$ GETNEXTMINIBATCH()
  $\tilde{\mathbf{x}}_1^U, \tilde{\mathbf{x}}_2^U, \tilde{\mathbf{y}}^U \leftarrow$ AUG+SOFTPSEUDOLABEL$(\mathbf{x}^U; w, \tau)$
  $\tilde{\mathbf{x}}^L, \tilde{\mathbf{y}}^L \leftarrow$ MIXMATCHAUG$(\{\mathbf{x}^L, \mathbf{y}^L\}, \tilde{\mathbf{x}}_1^U, \tilde{\mathbf{x}}_2^U, \tilde{\mathbf{y}}^U; \alpha)$
  $g^L \leftarrow \nabla_w \ell^L(\tilde{\mathbf{x}}^L, \tilde{\mathbf{y}}^L; w)$
  $g^U \leftarrow \nabla_w \ell^U(\tilde{\mathbf{x}}_1^U, \tilde{\mathbf{y}}^U; w)$
  if INNERPRODUCT$(g^L, g^U) > 0$ **then**
    $w \leftarrow w - \epsilon(g^L + \lambda_i g^U)$
  else
    $w \leftarrow w - \epsilon g^L$
  end if
end for
return w

Algorithm D.2: Augment and Soft-Pseudo-Label

Input: Unlabeled batch features $\mathbf{x}^U$
Output: Augmented features $\mathbf{x}_1^U, \mathbf{x}_2^U$, Soft pseudo labels $\tilde{\mathbf{y}}^U$
Hyperparameters

- Sharpening temperature $\tau > 0$

Procedure

for each image x in $\mathbf{x}^U$ **do**
  $x^{(1)} \leftarrow \text{BasicImageAugment}(x_n)$
  $x^{(2)} \leftarrow \text{BasicImageAugment}(x_n)$
  $\rho^{(1)} \leftarrow f_w(x^{(1)})$ `// Probability vector predicted by neural net`
  $\rho^{(2)} \leftarrow f_w(x^{(2)})$
  $\tilde{r} \leftarrow \left(\frac{1}{2}\rho^{(1)} + \frac{1}{2}\rho^{(2)}\right)^{1/\tau}$ `// Non-negative vector, sharpened by element-wise power`
  $S \leftarrow \sum_c \tilde{r}_c$
  $\tilde{y} \leftarrow [\frac{\tilde{r}_1}{S}, \frac{\tilde{r}_2}{S}, \ldots \frac{\tilde{r}_C}{S}]$ `// Normalize to "soft" label (proba. vector)`
  Add $x^{(1)}$ to $\tilde{\mathbf{x}}_1^U$
  Add $x^{(2)}$ to $\tilde{\mathbf{x}}_2^U$
  Add $\tilde{y}$ to $\tilde{\mathbf{y}}^U$
end for
return $\tilde{\mathbf{x}}_1^U, \tilde{\mathbf{x}}_2^U, \tilde{\mathbf{y}}^U$

Algorithm D.3: MixMatchAug : Transformation of Labeled Set

Input: Labeled batch $\mathbf{x}^L, \mathbf{y}^L$, Unlabeled batch $\mathbf{x}^U, \tilde{\mathbf{y}}$,
Output: Transformed labeled batch $\tilde{\mathbf{x}}^L, \tilde{\mathbf{y}}^L$
Hyperparameters

- Shape $\alpha > 0$ of Beta(α, α) dist.

for image-label pair x, y in labeled batch $\mathbf{x}^L, \mathbf{y}^L$ **do**
  $x', y' \leftarrow \textsc{SampleOnePair}([\mathbf{x}^L, \tilde{\mathbf{x}}_1^U, \tilde{\mathbf{x}}_2^U], [\mathbf{y}^L, \tilde{\mathbf{y}}^U, \tilde{\mathbf{y}}^U])$
  $\beta' \sim \textsc{SampleFromBeta}(\alpha, \alpha)$
  $\beta \leftarrow \textsc{Max}(\beta', 1 - \beta')$
  $\tilde{x} \leftarrow \beta x + (1 - \beta)x'$
  $\tilde{y} \leftarrow \beta y + (1 - \beta)y'$
  Add $\tilde{x}$ to $\tilde{\mathbf{x}}^L$
  Add $\tilde{y}$ to $\tilde{\mathbf{y}}^L$
end for
return $\tilde{\mathbf{x}}^L, \tilde{\mathbf{y}}^L$

E Proof: Fix-A-Step Update is Descent Direction of Labeled Loss

Definition 1: Descent direction of loss ℓ. For any loss function ℓ parameterized by weight vector $w \in \mathbb{R}^D$, a vector $v \in \mathbb{R}^D$ is a *descent direction* of ℓ at w if it satisfies $v^T \nabla_w \ell < 0$ [2].

Lemma 1: The update in Eq. (2) steps in a descent direction of the labeled loss ℓ^L at the current minibatch. We prove for each of the two cases in Eq. (2). *Top case*: Here by assumption the inner product $\sum_d g_d^L g_d^U$ is positive. This implies that $v = -(g^L + \lambda g^U)$ is a descent direction, because $\lambda > 0$ and thus

$$v^T g^L = \underbrace{-\sum_d (g_d^L)^2}_{\text{always negative}} \quad -\lambda \quad \underbrace{\sum_d g_d^L g_d^U}_{\text{pos. by assumption}} \quad < 0 \tag{4}$$

Bottom case: $-g^L$ is a descent direction for ℓ^L by definition.

While Lemma 1 provides a justification for our approach, we cannot formally guarantee the labeled-set loss will not increase after each step, for the same reasons that stochastic gradient descent (SGD) does not always decrease its loss after each minibatch update. First, a descent direction of a small minibatch may not be a descent direction of the entire dataset. Second, even though the direction of the step points locally downhill, the length of the step matters; if the step size $\epsilon > 0$ is too large, the loss may increase. Nevertheless, with proper step size tuning SGD has been wildly successful despite following minibatch-specific descent directions without formal guarantees of non-increasing loss. Thus far, we find our approach also successful in practice.