Self-supervised Semi-supervised Learning for Data Labeling and Quality Evaluation

Haoping Bai* **Meng Cao** **Ping Huang** **Jiulong Shan**

Apple

{haoping_bai, mengcao, huang_ping, jlshan}@apple.com

Abstract

As the adoption of deep learning techniques in industrial applications grows with increasing speed and scale, successful deployment of deep learning models often hinges on the availability, volume, and quality of annotated data. In this paper, we tackle the problems of efficient data labeling and annotation verification under the human-in-the-loop setting. We showcase that the latest advancements in the field of self-supervised visual representation learning can lead to tools and methods that benefit the curation and engineering of natural image datasets, reducing annotation cost and increasing annotation quality. We propose a unifying framework by leveraging self-supervised semi-supervised learning and use it to construct workflows for data labeling and annotation verification tasks. We demonstrate the effectiveness of our workflows over existing methodologies. On active learning task, our method achieves 97.0% Top-1 Accuracy on CIFAR10 with 0.1% annotated data, and 83.9% Top-1 Accuracy on CIFAR100 with 10% annotated data. When learning with 50% of wrong labels, our method achieves 97.4% Top-1 Accuracy on CIFAR10 and 85.5% Top-1 Accuracy on CIFAR100.

1 Introduction

As deep learning models and algorithms continue to evolve with increasing capacity and complexity, existing research has shown success across a plethora of domains and tasks [1, 2, 3] at the cost of a growing amount of data and compute. However, many industrial applications do not have readily available high-quality datasets. As a result, a large part of the machine learning life cycle is data engineering [4], which often requires painstaking manual annotation and inspection that are expensive and time-consuming [5].

To reduce the amount of human effort, it is necessary to automate the data curation process and reduce the number of labels needed for good performance. For example, active learning can reduce the amount of manual labor required by prioritizing the most informative data sample for labeling. Recent progress in active learning has shown promising results in speeding up human-in-the-loop annotation [6, 7, 8, 9, 10]. To help model learn with fewer labels, both self-supervised learning [11, 12, 13, 14, 15, 16] and semi-supervised learning [17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28] has shown substantial progress, achieving competitive results against supervised baselines with limited supervisions.

Inspired by the latest advances in self-supervised learning and semi-supervised learning, we combine the best of both worlds and build a simple yet versatile image similarity-based framework for robust and efficient labeling and data verification. The contributions of the paper are

*Corresponding author: Haoping Bai haoping_bai@apple.com.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

- We leverage the latest advances in self-supervised learning and computer vision architectures to obtain image representations that is useful for versatile downstream usages.
- We demonstrate the effectiveness of the self-supervised representation in a variety of scenarios including active learning-based human-in-the-loop annotation, label error detection, and robust classification with noisy labels.
- Unlike expensive semi-supervised approaches that require updating neural networks to incorporate label information [28], our approach leverages label propagation based on nearest neighbor graph to quickly incorporate new label information via simple matrix multiplication. As a result, our method can be seamlessly incorporated into real-time human-in-the-loop systems without sacrificing throughput.

We hope our approach will be a simple and strong baseline to motivate future progress in building labor-efficient data curation pipelines and label-efficient machine learning systems. Code will be made available.

2 Methods

Our general workflow consists of two parts. We first leverage self-supervised learning, specifically, contrastive learning methods to obtain an unsupervised representation for the unlabeled data. Then we construct a nearest neighbor graph over data samples based on the learned representations. Finally, we can use the nearest neighbor graph for various downstream tasks. In this section, we will discuss each component of our method in detail.

2.1 Problem formulation

We assume a dataset of n examples $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \mathbb{R}^{n \times d}$ with d being the feature dimension. For our purpose, we define $l \subseteq [n]$ to be the index set for samples with human verified labels and $u \in [n]$ to be the index set for data samples without trusted labels. The label matrix $Y = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \in \mathbb{R}^{n \times c}$ with c being the number of classes. Our goal is to leverage feature matrix X and known label matrix Y_l to generate and improve the estimate $\tilde{Y}_u$ for the unknown label matrix Y_u.

2.2 Self-supervised Learning

Recent developments in self-training have seen a substantial progress exemplified by a series of work [11, 14, 12, 15, 16] in contrastive learning, where the goal is to learn representation that is invariant across two views of the same image created via data augmentation. Specifically, we leverage BYOL [14], which uses an asymetric siamese architecture including online encoder f_θ, online projector g_θ and predictor q_θ for one branch and a target encoder f_ξ and projector g_ξ for the other branch with polyak averaged weights $\xi \leftarrow \tau\xi + (1-\tau)\theta$. Given two views v_1 and v_2 of the same image x, we obtain projections $p_i = g_\xi \circ f_\xi(v_i)$ and predictions $z_i = q_\theta \circ g_\theta \circ f_\theta(v_i)$ for $i \in \{1, 2\}$, and we train θ with the following loss

$$\mathcal{L} = \ell(p_1, z_2)/2 + \ell(p_2, z_1)/2, \quad \text{where } \ell(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2} \tag{1}$$

After training is complete, we use $\ell(f_\theta(x_i), f_\theta(x_j))$ as a similarity metric between x_i and x_j.

2.3 Nearest Neighbor Graph

Based on the metric $\ell(f_\theta(x_i), f_\theta(x_j))$, we can build a nearest neighbor graph in the form of sparse adjacency matrix W where

$$W_{ij} = \begin{cases} \exp(\ell(f_\theta(x_i), f_\theta(x_j))/T), & \text{if } j \in \text{NN}(i,k) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\text{NN}(i,k)$ denotes the index set of the k nearest neighbors of x_i, and T is a temperature parameter. The symmetrically normalized counterpart of W is given by $\mathcal{W} = D^{-1/2} W D^{-1/2}$, where $D = \text{diag}(W\mathbf{1}_n)$ is the degree matrix and $\mathbf{1}_n$ is an all-ones n-vector.

2.4 Semi-supervised Pseudo-Labeling with Label Propagation

Based on the consistency assumption [29] that nearby nodes are likely to have the same label, we can perform label propagation (LP) on the nearest neighbor graph to propagate information from samples with known labels to samples without label or with noisy labels as follows

$$\tilde{Y}^{(t+1)} = \mathcal{W}Y^{(t)},\ Y_u^{(t+1)} = \tilde{Y}_u^{(t+1)},\ Y_l^{(t+1)} = Y_l^{(0)} \tag{3}$$

where $Y^{(0)} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\}$ is the initial label matrix, $l \subseteq [n]$ denotes index set for samples with annotated/verified labels, $u \subseteq [n]$ denotes index set for samples without trusted labels. $\tilde{Y}^{(t+1)}$ is the soft label matrix at iteration $t+1$, from which we take $\tilde{Y}_u^{(t+1)}$ as new soft label for the unlabeled split u and reset $Y_l^{(t+1)}$ to the given ground truth $Y_l^{(0)}$. Since LP is mostly sparse matrix multiplication, incorporating information from added labels into the pseudo-labels is much faster than training a deep learning model with partial labels.

3 Experimental Analysis and Results

In this section, we showcase the performance of two convenient workflows that we built around the aforementioned techniques with minimal modifications.

Experiment Settings and Implementation Details We use CIFAR10 and CIFAR100 [30] as our benchmark datasets. To perform self-supervised learning on target dataset, we leverage the Vision Transformer [1], ViT-B/16, as our encoder architecture while following the exact projector and predictor definition in [1]. We initialize the ViT encoder with BEiT [31] pretrained weights. We perform all training with batchsize 64 with images resized to 224×224 resolution. We use the AdamW optimizer [32]. For each experiment, we determine the learning rate, weight decay, and τ through grid search. On CIFAR10, we run BYOL for 3k steps with a learning rate of $2.3e-5$, $5.e-4$ weight decay, $\tau = 0.998$. On CIFAR100, we run BYOL for 5k steps with a learning rate of $4.6e-5$, $2.1e-6$ weight decay, $\tau = 0.9993$. Following [1], we gradually anneal τ to 1 during training. To construct k-NN graph, we use $k = 10$ and $T = 0.01$ for CIFAR10, and $k = 15$ and $T = 0.02$ for CIFAR100. We use $t = 20$ iterations for LP.

k-NN Classification Performance with Learned Representations To assess the quality of our learned representation, we directly performed weighted k-NN classification based on the nearest neighbor graph in Equation 2. We achieved 98.45% validation Top-1 accuracy on CIFAR10, and 89.58% validation Top-1 accuracy on CIFAR100.

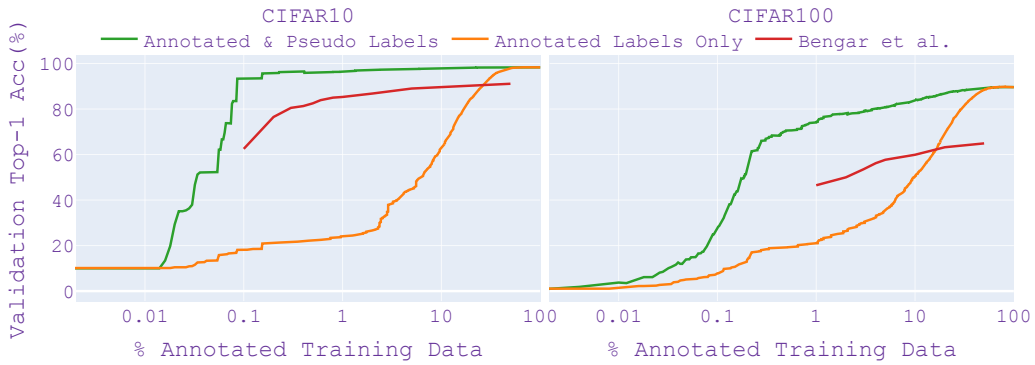

Figure 1: Active Learning Performance on CIFAR10 (left) and CIFAR100 (right). Orange line denotes using only labeled training data to predict validation labels. Red line denotes results from Bengar et al. [33]. Green line denotes using both labeled training data and the unlabeled training data with LP generated pseudo-labels to predict validation labels.

3.1 Efficient Annotation with Active Learning

For this task, we perform simulation using both datasets and start with no training label. We simulate the human-in-the-loop annotation process by iteratively performing LP and randomly sampling data for oracle labeling. Following the observation in [33], we choose the random sampling as a simple and strong baseline. As shown in Figure 1, we achieve exponential gain in Top-1 Accuracy when

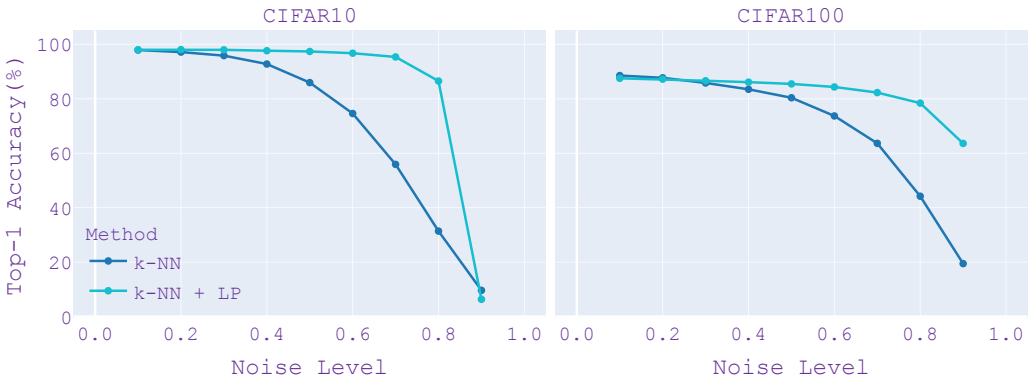

Figure 2: Classification Performance Under Label Noise on CIFAR10 (left) and CIFAR100 (right)

annotating fewer than $< 0.1\%$ data in CIFAR10 and $< 1\%$ data in CIFAR100. We outperform previous work by Bengar et al. [33] substantially.

To assess the value of pseudo-labels generated by LP on the unlabeled training data, we perform an ablation study by performing LP only on the annotated data and the validation data. As shown in Figure 1, without pseudo-labels from the unlabeled training data, the orange curves show substantial drops in validation top-1 accuracy when using the same amount of annotation. Thus, having a reliable nearest neighbor graph allows us to effectively scale performance with the amount of unlabeled data by propagating information from labeled data across the data manifold.

3.2 Robust Classification with Noisy Labels

We showcase our second workflow by demonstrating the effectiveness of using LP to correct corrupted labels and maintain robust classification performance under noise.

Effect of LP on Noise Reduction We first examine the evidence for the consistency assumption [29] by simulating random noise in labels and then performing LP. If our nearest neighbor graph faithfully captures the similarity among data, LP will aggregate and smooth out the inconsistency from noisy neighbor labels so that the neighbors with the correct label can stand out. As illustrated in Figure 3, with a noise level below 0.8, LP quickly reduces the noise level in pseudo label with an increasing number of iterations. At extreme noise levels such as 0.9, the consistency assumption no longer holds due to highly corrupted labels, and performing LP hurts performance instead.

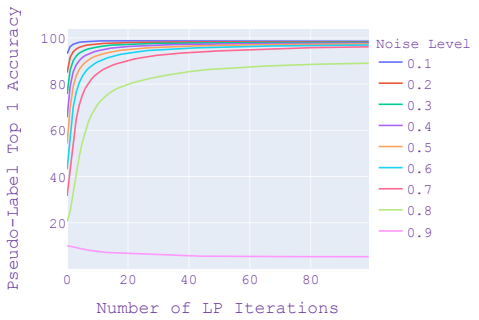

Figure 3: The Noise Level in Pseudo Label vs. Number of LP Iterations

Classification under Noisy Label After obtaining smoothed pseudo labels via LP, we use the pseudo labels with the nearest neighbor graph to perform weighted k-NN classification. For comparison, we use weighted k-NN based on the corrupted labels as a baseline. As shown in Figure 2, pseudo labels obtained via LP offer more robust performance than directly using the corrupted labels on both CIFAR10 and CIFAR100.

4 Conclusion

In summary, leveraging the latest advances in self-supervised learning, we developed a nearest neighbor graph-based approach that can perform versatile downstream tasks and quickly incorporate new information in a semi-supervised manner, suitable for integration with real time human-in-the-loop systems. We demonstrated the effectiveness of our methods on a combination of datasets (CIFAR10, CIFAR100) and tasks (active learning, label error detection, and learning under noise) and achieved competitive performance. We hope our work can serve as a simple and strong baseline for the development of practical tools in industrial settings.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[2] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019.

[4] Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, and Martin Zinkevich. Data management challenges in production machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, page 1723–1726, New York, NY, USA, 2017. Association for Computing Machinery.

[5] Clas Blank. Automatic vs. manual data labeling: A system dynamics modeling approach, 2020.

[6] Wenbin Cai, Ya Zhang, Siyuan Zhou, Wenquan Wang, Chris Ding, and Xiao Gu. Active learning for support vector machines with maximum model change. In *Machine Learning and Knowledge Discovery in Databases*, pages 211–226. Springer, 2014.

[7] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *IJCV*, 113(2):113–127, 2015.

[8] Yuhong Guo. Active instance sampling via matrix partition. In *NIPS*, pages 1–9, 2010.

[9] Suyog Dutt Jain and Kristen Grauman. Active image segmentation propagation. In *CVPR*, pages 2864–2873, 2016.

[10] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[13] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[15] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[16] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[17] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[18] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[19] Augustus Odena. Semi-supervised learning with generative adversarial networks, 2016.

[20] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[21] Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning, 2018.

[22] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning, 2019.

[23] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3635–3641. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[24] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. *ICLR*, 2019.

[25] Bin Liu, Zhirong Wu, Han Hu, and Stephen Lin. Deep metric transfer for label propagation with limited annotated data. 2018.

[26] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[27] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4182–4192. PMLR, 13–18 Jul 2020.

[28] Yuan-Hong Liao, Amlan Kar, and Sanja Fidler. Towards good practices for efficiently annotating large-scale image classification datasets, 2021.

[29] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, pages 321–328, 2003.

[30] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[31] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. 2021.

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[33] Javad Zolfaghari Bengar, Joost van de Weijer, Bartlomiej Twardowski, and Bogdan Raducanu. Reducing label effort: Self-supervised meets active learning, 2021.