

LAMDA-SSL: Semi-Supervised Learning in Python

Lin-Han Jia

JIALH@LAMDA.NJU.EDU.CN

*National Key Laboratory for Novel Software technology
Nanjing University
Nanjing 210023, China*

Lan-Zhe Guo

GUOLZ@LAMDA.NJU.EDU.CN

*National Key Laboratory for Novel Software technology
Nanjing University
Nanjing 210023, China*

Zhi Zhou

ZHOUZ@LAMDA.NJU.EDU.CN

*National Key Laboratory for Novel Software technology
Nanjing University
Nanjing 210023, China*

Yu-Feng Li

LIYF@LAMDA.NJU.EDU.CN

*National Key Laboratory for Novel Software technology
Nanjing University
Nanjing 210023, China*

Editor:

Abstract

Semi-supervised learning (SSL) aims to improve learning performance by exploiting unlabeled data when labels are limited or expensive to obtain. SSL is an important research field in machine learning and many SSL algorithms have been proposed. However, there still lacks a comprehensive SSL toolkit to make machine learning users apply SSL algorithms conveniently. In this paper, we provide LAMDA-SSL, a comprehensive Python SSL toolkit, to support the development and wide application of SSL. LAMDA-SSL supports more than 30 representative SSL algorithms, including both statistical SSL algorithms and deep SSL algorithms; 4 data types including image, text, tabular, and graph; and 3 machine learning tasks including classification, regression, and clustering. To the best of our knowledge, it is the most comprehensive SSL toolkit available. Benefiting from its *powerful functions*, *simple interfaces*, and *extensive documentation*, LAMDA-SSL is a comprehensive, easy-to-use, open-source toolkit for researchers to develop follow-up studies and for engineers to solve real-world tasks. The source code is available at: <https://github.com/YGZWQZD/LAMDA-SSL>.

Keywords: semi-supervised learning, toolkit, python, statistical learning, deep learning

1. Introduction

In many real-world applications of machine learning, large-scale well-labeled datasets are expensive to obtain, as the acquisition of labels requires huge human labor and financial costs (Zhou, 2018; Li et al., 2021; Guo and Li, 2022). SSL is one of the most promising learning paradigms to ease the scarcity of labeled data by leveraging an abundance of unlabeled data.

beled data (Chapelle et al., 2006; Oliver et al., 2018). However, immense knowledge barriers make it difficult for non-professionals to apply SSL algorithms to solve practical problems conveniently. At present, there is still a lack of comprehensive and easy-to-use SSL toolkits. Only the SSL module of scikit-learn (Pedregosa et al., 2011) and TorchSSL (Zhang et al., 2021) which is a Pytorch-based (Paszke et al., 2019) toolkit are developed for statistical SSL and deep SSL respectively. Unfortunately, these SSL toolkits are unsound, for example, scikit-learn only contains 3 statistical SSL algorithms and does not support deep SSL, TorchSSL contains 9 deep SSL algorithms but only supports classification tasks for images.

In this paper, we present LAMDA-SSL, an open-sourced toolkit in Python for SSL. LAMDA-SSL has integrated statistical SSL algorithms and deep SSL algorithms into the same framework and is compatible with both scikit-learn and Pytorch. Currently, LAMDA-SSL has implemented 30 SSL algorithms, including 12 statistical SSL algorithms and 18 deep SSL algorithms. LAMDA-SSL is designed considering both two aspects: data and model (as shown in Figure 1). In the data module, LAMDA-SSL can perform data management and data transformation for 4 types of data: tabular, image, text and graph. In the model module, LAMDA-SSL can perform model application and model deployment for 3 types of tasks: classification, regression and clustering. For entry-level users, LAMDA-SSL provides simple interfaces and well-tuned default parameters. For professional users, LAMDA-SSL provides flexible component replacement and customization interfaces.

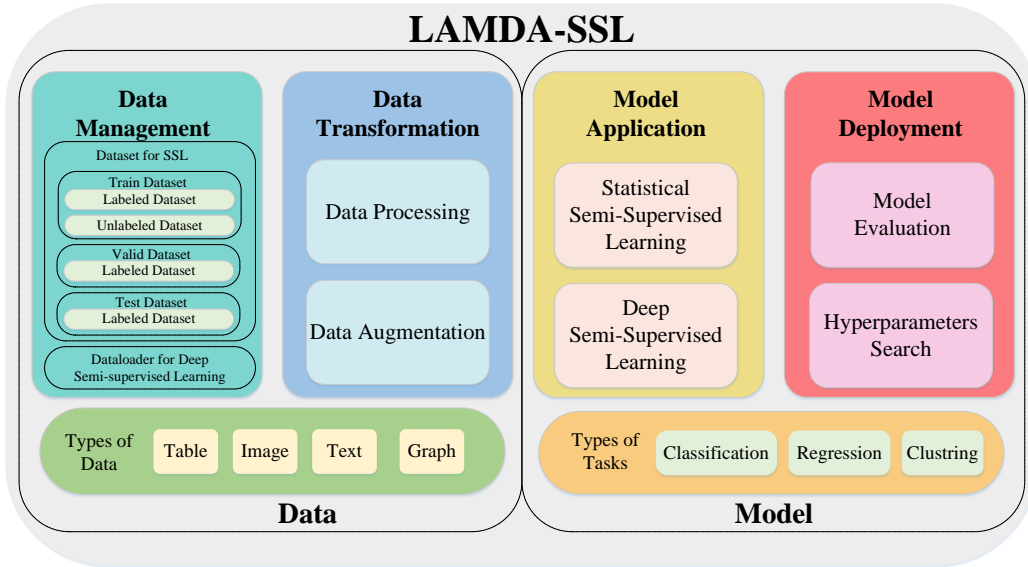


Figure 1: An overview of LAMDA-SSL.

We have compared LAMDA-SSL with the SSL module of scikit-learn and TorchSSL (as shown in Table 1). To our best knowledge, LAMDA-SSL is the first SSL toolkit that has integrated statistical SSL algorithms and deep SSL algorithms into the same framework. In the field of statistical SSL, LAMDA-SSL is more suitable for SSL in applications compared with scikit-learn. In the field of deep SSL, LAMDA-SSL has significant advantages in terms of the number of algorithms, data types, task types, functions and documentation compared with TorchSSL.

Table 1: The comparison of LAMDA-SSL with other related toolkits.

Toolkit	scikit-learn	TorchSSL	LAMDA-SSL
The number of statistical SSL algorithms	3	0	12
The number of deep SSL algorithms	0	9	18
Types of data	Tabular Image Text	Image	Tabular Image Text Graph
Types of task	Classification Regression Clustering	Classification	Classification Regression Clustering
Hyper-parameters search	✓	×	✓
GPU acceleration	×	✓	✓
Distributed learning	×	✓	✓
Documentation	✓	×	✓

2. Superiority of LAMDA-SSL

In this section, we described the key superiority of LAMDA-SSL, including powerful functions, simple interfaces, and extensive documentation.

2.1 Powerful Functions

At present, LAMDA-SSL has implemented 30 SSL algorithms, including 12 statistical SSL algorithms and 18 deep SSL algorithms.

For statistical SSL, algorithms in LAMDA-SSL can be used for classification, regression and clustering (Zhou, 2021). The algorithms used for classification include generative method SSGMM (Shahshahani and Landgrebe, 1994); semi-supervised support vector machine methods TSVM (Joachims et al., 1999) and LapSVM (Belkin et al., 2006); graph-based methods Label Propagation (Zhu and Ghahramani, 2003) and Label Spreading (Zhou et al., 2003); disagreement-based methods Co-Training (Blum and Mitchell, 1998) and Tri-Training (Zhou and Li, 2005b); ensemble methods SemiBoost (Bennett et al., 2002) and Assemble (Mallapragada et al., 2008). The algorithm used for regression is CoReg (Zhou and Li, 2005a). The algorithms used for clustering include Constrained K -Means (Wagstaff et al., 2001) and Constrained Seed K -Means (Basu et al., 2002).

For deep SSL, algorithms in LAMDA-SSL can be used for classification and regression (Yang et al., 2021). The algorithms used for classification include consistency methods Ladder Network (Rasmus et al., 2015), Π Model, Temporal Ensembling (Laine and Aila, 2017), Mean Teacher (Tarvainen and Valpola, 2017), VAT (Miyato et al., 2018) and UDA (Xie et al., 2020); pseudo label-based methods Pseudo Label (Lee, 2013) and S4L (Zhai et al., 2019); hybrid methods ICT (Verma et al., 2019), MixMatch (Berthelot et al., 2019), ReMixMatch (Berthelot et al., 2020), FixMatch (Sohn et al., 2020) and FlexMatch (Zhang et al., 2021); deep generative methods ImprovedGAN (Salimans et al., 2016)

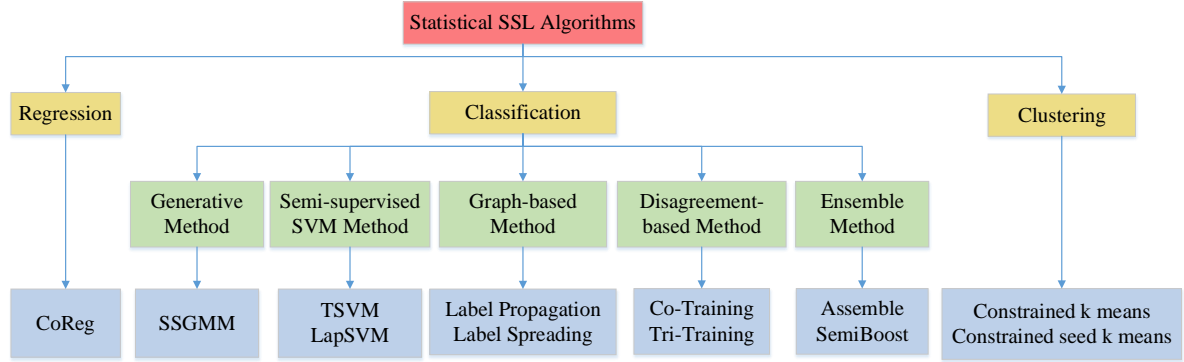


Figure 2: Statistical SSL algorithms in LAMDA-SSL.

and SSVAE (Kingma et al., 2014); deep graph-based methods SDNE (Wang et al., 2016), GCN (Kipf and Welling, 2017) and GAT (Veličković et al., 2018). The algorithms for regression include Π Model Reg, Mean Teacher Reg and ICT Reg.

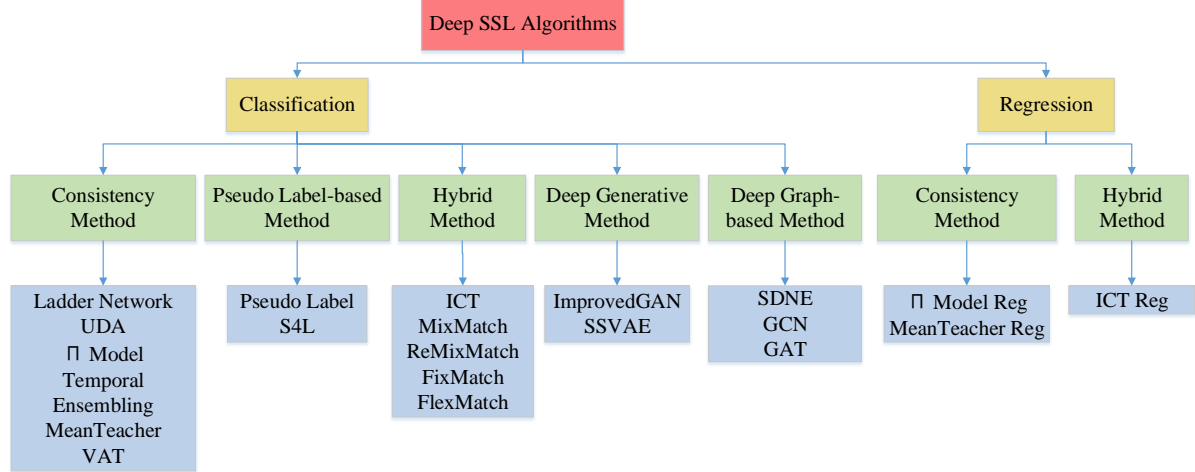


Figure 3: Deep SSL algorithms in LAMDA-SSL.

In addition to the supported algorithms, LAMDA-SSL provides 44 data transformation methods and 16 metrics for model evaluation. LAMDA-SSL also provides flexible component replacement and customization interfaces for professional users. Especially for deep SSL, users can arbitrarily replace and customize modules of deep SSL algorithms such as Dataset, Dataloader, Sampler, Augmentation, Network, Optimizer and Scheduler without worrying about affecting other modules. Users can also achieve low-code implementation for customized deep SSL algorithms by inheriting a component in LAMDA-SSL called DeepModelMixin which provides many default processing functions used for deep SSL. Moreover, LAMDA-SSL is compatible with both scikit-learn and Pytorch and has inherited their mechanisms and functions. Like scikit-learn, LAMDA-SSL supports the Pipeline mechanism and has the function of hyper-parameters search. Like Pytorch, LAMDA-SSL can use GPU to accelerate training process and support distributed learning.

2.2 Simple Interfaces

The APIs of LAMDA-SSL refer to scikit-learn and all the learners have two basic methods: `fit()` and `predict()`. The only difference from the APIs of scikit-learn is that the `fit()` method of LAMDA-SSL needs three data items of X , y and $unlabeled_X$ to be input. For deep SSL algorithms, LAMDA-SSL uses DeepModelMixin component to make the APIs of deep SSL algorithms and statistical SSL algorithms unified. Lots of elaborate examples can be found in the online documentation and the Example module of the source code.

```
from LAMDA_SSL.Dataset.Vision.CIFAR10 import CIFAR10
from LAMDA_SSL.Algorithm.Classification.FixMatch import FixMatch
from LAMDA_SSL.Evaluation.Classifier.Accuracy import Accuracy
# Initialize CIFAR10 dataset
dataset=CIFAR10(root='../Download\\cifar-10-python',labeled_size=4000)
labeled_X, labeled_y=dataset.labeled_X,dataset.labeled_Y
unlabeled_X=dataset.unlabeled_X
test_X, test_y=dataset.test_X, dataset.test_y
# Initialize FixMatch algorithm
model=FixMatch(threshold=0.95,lambda_u=1.0,T=0.5,mu=7,
               epoch=1,num_it_epoch=2**20,device='cuda:0')
# Call the fit() method to Train the model
model.fit(X=labeled_X,y=labeled_y,unlabeled_X=unlabeled_X)
# Call the predict() method to predict the labels of new samples
y_pred=model.predict(test_X)
# Evaluate the model's performance.
performance=Accuracy().scoring(test_y,y_pred)
```

Figure 4: A basic example of LAMDA-SSL.

2.3 Extensive Documentation

LAMDA-SSL is open-sourced on GitHub and its detailed usage documentation is available at <https://ygzqwzd.github.io/LAMDA-SSL/>. This documentation introduces LAMDA-SSL in detail from various aspects and can be divided into four parts. The first part introduces the design idea, features and functions of LAMDA-SSL. The second part shows the usage of LAMDA-SSL by abundant examples in detail. The third part introduces all algorithms implemented by LAMDA-SSL to help users quickly understand and choose SSL algorithms. The fourth part shows the APIs of LAMDA-SSL. This detailed documentation greatly reduces the cost of familiarizing users with LAMDA-SSL toolkit and SSL algorithms.

3. Quality Standards

In the following, we evaluate LAMDA-SSL according to several quality standards of open source software.

Availability. LAMDA-SSL’s packages for Python 3.7 and above are available for Linux, macOS, and Windows, and can be acquired via Pypi easily using ‘`pip install LAMDA-SSL`’.

Reliability. The code coverage of LAMDA-SSL is higher than 90% and the testing report is open at <https://coveralls.io/github/YGZWWQZD/LAMDA-SSL>. The performances

of all algorithms in LAMDA-SSL are evaluated on multiple datasets and the experimental results are available on the homepage.

Openness. LAMDA-SSL is distributed under the MIT license. Contributions from the community are strongly welcome and easy enough because the documentation provides numerous examples showing how to customize modules of LAMDA-SSL.

4. Conclusions and Discussions

In this paper, we present LAMDA-SSL, an easy-to-use, powerful and open-source toolkit in Python for SSL with ample functions, simple interfaces, complete documentation, and the best support for algorithms, data types, and tasks compared with other related toolkits.

It is expected that LAMDA-SSL can promote the research and applications of SSL to ease the scarcity of labeled data. In the future, we are interested in incorporating more advanced algorithms into LAMDA-SSL and expanding the application scope of SSL in open environments (Zhou, 2022).

Acknowledgments

This research was supported by the National Science Foundation of China (62176118, 61921006). Lan-Zhe Guo and Yu-Feng Li are the corresponding authors.

References

- Sugato Basu, Arindam Banerjee, and Raymond Mooney. Semi-supervised clustering by seeding. In *Proceedings of the 19th International Conference on Machine Learning*, pages 27–34, 2002.
- Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(11):2399–2434, 2006.
- Kristin P Bennett, Ayhan Demiriz, and Richard Maclin. Exploiting unlabeled data in ensemble methods. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 289–296, 2002.
- David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019.
- David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.
- Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference Computational Learning Theory*, pages 92–100, 1998.

- Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 2006.
- Lan-Zhe Guo and Yu-Feng Li. Class-imbalanced semi-supervised learning with adaptive thresholding. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8082–8094, 2022.
- Thorsten Joachims et al. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, pages 200–209, 1999.
- Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.
- Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proceedings of the 30th International Conference on Machine Learning Workshop*, page 896, 2013.
- Yu-Feng Li, Lan-Zhe Guo, and Zhi-Hua Zhou. Towards safe weakly supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):334–346, 2021.
- Pavan Kumar Mallapragada, Rong Jin, Anil K Jain, and Yi Liu. Semiboost: Boosting for semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2000–2014, 2008.
- Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin 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, 2018.
- Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3239–3250, 2018.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

- Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.
- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016.
- Behzad M Shahshahani and David A Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095, 1994.
- Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, pages 596–608, 2020.
- Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.
- Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3635–3641, 2019.
- Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Proceedings of the 18th International Conference on Machine Learning*, pages 577–584, 2001.
- Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1225–1234, 2016.
- Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, pages 6256–6268, 2020.
- Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *arXiv preprint arXiv:2103.00550*, 2021.
- Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1476–1485, 2019.

- Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, pages 18408–18419, 2021.
- Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, pages 321–328, 2003.
- Zhi Zhou, Lan-Zhe Guo, Zhanzhan Cheng, Yu-Feng Li, and Shiliang Pu. Step: Out-of-distribution detection in the presence of limited in-distribution labeled data. In *Advances in Neural Information Processing Systems*, pages 29168–29180, 2021.
- Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.
- Zhi-Hua Zhou. *Machine learning*. Springer Nature, 2021.
- Zhi-Hua Zhou. Open environment machine learning. *National Science Review*, 9(8), 2022.
- Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 908–913, 2005a.
- Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, 2005b.
- Xiaojin Zhu and Zoubin Ghahramani. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919, 2003.