Research in Astronomy and Astrophysics, 25:104005 (9pp), 2025 October
© 2025. National Astronomical Observatories, CAS and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.
https://doi.org/10.1088/1674-4527/adf712
CSTR: 32081.14.RAA.adf712

Semi-supervised Learning for Galaxy Morphology Classification Based on Diversity Enhancement and Uncertainty Estimation

Zushou Chen, Zhen Xu, and Weibin Chen
College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China; sun@wzu.edu.cn
Received 2025 April 30; revised 2025 June 10; accepted 2025 June 18; published 2025 September 12

Abstract

Our knowledge of galaxy creation and evolution has significantly improved in recent years because of the substantial advancements made in the classification of galaxy morphology using machine learning techniques. Nevertheless, the vast amount of unlabeled astronomical image data restricts the application of these techniques, and it is expensive to manually label enormous volumes of astronomical data. Current semi-supervised approaches have used pseudo-labeling and consistency regularization to get good results. However, many pseudo-labels are underutilized because of their excessively high confidence levels. Furthermore, confirmation bias in the model may result from the overconfidence commonly seen in softmax outputs, which is usually disregarded in current research. In order to decrease confirmation bias throughout the learning process, this study introduces uncertainty quantification. Specifically, the suggested approach fully utilizes pseudo-labeled data by using distinct training procedures according to different confidence levels. This method lowers errors brought on by the model's overconfidence while simultaneously improving the quality of pseudo-labels. Results from experiments show that our approach performs well on the Galaxy Zoo 2 data set and the Galaxy10 DECaLS data set. In particular, our approach obtains an accuracy of 76.8% on the Galaxy10 DECaLS data set, which contains 2000 labeled samples, while fully supervised approaches only achieve 70.8%. In comparison to the supervised approach, our strategy lowers the error rate by 32.34% for the 1000 labeled samples in the Galaxy Zoo 2 data set. Our method accomplishes accurate categorization of galaxy morphology by using a large amount of unlabeled data and a small amount of labeled data. The public can get the source code at https://github.com/CZSGC/SSLGMCBDEUE.

Key words: techniques: image processing – methods: data analysis – surveys – catalogs

1. Introduction

Classification of galaxy morphology is important for studying celestial evolution (Maddox et al. 1990) and celestial attributes. It is possible to gain more understanding of the motion patterns of celestial bodies by measuring and analyzing astronomical data. Galaxy morphology categorization that combines machine learning and computer vision allows highly automated analytical tasks (Barchi et al. 2020), greatly decreasing the workload for astronomy specialists in comparison to costly and time-consuming spectroscopic data processing. Galaxy morphology classification classifies galaxies into distinct classes by extracting characteristics from galaxy images, such as form and the number of spiral arms. It is possible to determine the evolutionary patterns of galaxies throughout different historical times and cosmic conditions by examining the morphological traits of galaxies at different evolutionary stages and their places in the cosmos (Parry et al. 2009). Over the past ten years, machine learning—and deep learning in particular—has advanced quickly. The notion of using deep learning techniques to classify galaxy morphology naturally arose as deep learning applications in picture classification grew in maturity. However, categorizing galaxy morphology data necessitates a high level of astronomical skill, and traditional deep learning algorithms require a lot of labeled data for model training. Large-scale sky surveys that currently exist, like the Sloan Digital Sky Survey (SDSS; Blanton et al. 2017), offer a significant amount of unlabeled data that requires additional annotation. Moreover, with the development of large-scale telescopes, such as the Euclid Survey (Euclid Collaboration et al. 2022), the Large Synoptic Survey Telescope (LSST; Robertson et al. 2019), and the Chinese Space Station Telescope (Gong et al. 2025), the challenge lies in effectively utilizing these precious yet massive unlabeled data sets. The Galaxy Zoo (GZ; Lintott et al. 2008) project addresses this issue by engaging online volunteers to answer simple multiple-choice questions about galaxy images, thereby performing basic classification tasks and iteratively refining the classification scheme for data set annotation (Willett et al. 2013). However, this approach still faces challenges: the recruited volunteers lack astronomical expertise and are unable to address complex galaxy morphology classification tasks, leading to the introduction of significant noise in the labeled data. Even with high threshold

settings, mislabeled data may still be selected. Unsupervised learning techniques have been investigated in certain research to make use of unlabeled data. However, additional classifiers must still be trained for downstream tasks because unsupervised learning lacks defined classification objectives (Cheng et al. 2020; Martin et al. 2020; Han et al. 2022; Wei et al. 2022). Some works also focus on supervised learning to improve the model's capabilities (Wu et al. 2022).

By combining supervised and unsupervised methods, semi-supervised learning (SSL) makes use of both a lot of unlabeled data and the feature distribution of labeled data to achieve notable performance gains. By integrating the large amounts of unlabeled data accessible in the field of astronomy with a modest amount of labeled data annotated by experts, SSL efficiently leverages the domain expertise of astronomy specialists. This method makes complex classification tasks possible while lessening the effort for specialists. In the field of natural image classification, methods like FixMatch (Sohn et al. 2020) have achieved remarkable results. FixMatch screens data by setting a fixed confidence threshold and utilizes the filtered data for consistency regularization. Specifically, it applies cross-entropy loss to pseudo-labels generated from both weakly and strongly augmented versions of the data. This effectively leverages unlabeled data. However, setting a fixed threshold may result in the loss of many valuable data.

Using the Galaxy Zoo 2 (GZ2) data set as our benchmark, we assess the performance of various SSL algorithms on this data set. In this work, we propose a novel training method to mitigate the confirmation bias caused by overconfident pseudo-labels By classifying labeled data into easy and difficult data according to their confidence levels and applying different training loss strategies accordingly, we also measure the uncertainty of data using multiple prediction results and select data with higher diversity to prevent the model from forgetting when learning similar data. For the definition and discussion of confirmation bias, please refer to Section 4.

2. Preliminaries

2.1. Data Set

The GZ2 project data set and the Galaxy10 DECaLS data set are two significant data sets that we use in this study to assess the model's performance.

Galaxy Zoo 2 Data Set. The SDSS Data Release 7 (DR7) is the source of the photos used in the GZ2 project. There are 61,578 galaxy photos in this data set. These photos have been categorized through several rounds of stringent volunteer voting and screening procedures. There are still some out-of-distribution samples in the data set, though. Our tests show that out-of-distribution samples can still be detected and filtered using our model.

Galaxy10 DECaLS Data Set. The Galaxy10 DECaLS data set (Leung & Bovy 2019) substitutes high-quality, high-resolution photos from the DECaLS survey project for the original SDSS images and incorporates the categorization data from the GZ2 study. There are over 18,000 photos in this data set, which are separated into ten categories. The initial classifications have been adjusted to improve the orthogonality across various categories in order to maximize classification performance. Notably, in order to preserve data validity and balance, the category "Edge-on Disk with Boxy Bulge," which only has 17 photos, has been eliminated from the final data set.

2.2. Data Augmentation

In order to guarantee their proximity in feature space, we separate the augmentation algorithms into strong augmentation and weak augmentation using consistency regularization.

Resizing, random cropping, horizontal flipping, and normalizing are examples of weak augmentation. In addition to the operations in weak augmentation, Cutout (DeVries & Taylor 2017) applies three strong perturbation operations at random in strong augmentation. Among the severe disturbances are rotation, Sharpness, AutoContrast, Posterize, Solarize, ShearX/ShearY, Brightness, Contrast, Saturation, Color Jitter, and Sharpness.

2.3. Overview of Previous Work

Before presenting our method, we briefly summarize the approaches taken by other researchers as background information.

SSL mitigates the demand for labeled data by leveraging unlabeled data, addressing the high cost of data annotation. This has led to the development of numerous SSL methods tailored for deep learning. One prevalent SSL approach involves predicting outcomes for unlabeled data using a neural network during training and treating high-confidence predictions as labels for further training. This method is known as pseudo-labeling, or self-training. Pseudo-labeling encourages the model to produce more confident predictions by pushing the decision boundary into sparse regions through the use of numerous pseudo-labels. A notable example is Pseudo-Label (Lee 2013), which directly uses network-generated labels for cross-entropy loss computation. Noisy Student (Xie et al. 2020) enhances robustness by incorporating knowledge distillation during training. SimCLRv2 (Chen et al. 2020) demonstrates the adaptability of self-supervised learning frameworks to semi-supervised settings by employing a self-supervised approach. Another widely adopted method relies on the manifold assumption, enforcing consistency regularization by applying different levels of data augmentation or varying model parameters. This is known as consistency regularization. For instance, the Π Model (Sajjadi et al. 2016) applies random data augmentation to both labeled and unlabeled data,

enforcing consistency constraints to ensure similar predictions. Temporal Ensembling (Laine & Aila 2017) improves the Π Model by using Exponential Moving Average (EMA) to reduce the computational cost of multiple forward passes. Mean Teacher (Tarvainen & Valpola 2017) consists of a student-teacher network, where EMA is used to average the student's weights, and consistency constraints are directly imposed between the student and teacher networks. Deep semi-supervised generative models learn data distributions through generative models, primarily utilizing networks such as Generative Adversarial Networks (GANs; Goodfellow et al. 2020), Variational Autoencoders (VAEs; Kingma & Welling 2022), and their variants. Other methods include hybrid models that integrate the aforementioned approaches. Notably, FixMatch combines consistency regularization and pseudo-labeling, filtering out-of-distribution data through high-confidence predictions and achieving state-of-the-art performance with a simple training strategy. Our approach builds upon this method. Graph-based SSL (Wang et al. 2016) constructs a graph structure from the data set, leveraging the graph's properties to propagate labels to unlabeled data through techniques like label propagation. SemiReward (Li et al. 2024) employs adversarial training to determine whether an image is suitable for training, directly utilizing a discriminator to select images. U2PL+ (Wang et al. 2025) enhances performance by improving the utilization rate of unlabeled data.

In the field of astronomy, SSL methods have demonstrated promising applications. Jiang et al. (2023) proposed a dynamic threshold approach to enhance the utilization of unlabeled data in galaxy morphology classification. Slijepcevic et al. (2022) employed semi-supervised algorithms to improve both classification accuracy and robustness in radio galaxy morphology identification. Wang et al. (2025) developed a hybrid method that incorporates reconstruction results as pseudo-labels, which outperformed all baseline methods in terms of reconstruction quality, robustness, and generalization capability. Furthermore, Walmsley et al. (2022) achieved significant results by leveraging data similarity in their SSL framework.

Next, we will introduce the notation and mathematical formulations used in FixMatch, which will help clarify our proposed methodology. The following notation is used to denote the labeled and unlabeled data within a batch. labeled data: $\mathcal{X} = \{(x_i, y_i): i \in (1, 2,\ldots,B_l)\}$; unlabeled data: $\mathcal{U} = \{u_i: i \in (1, 2,\ldots,B_u)\}$. We define the weak augmentation mapping for images as $\omega(\cdot)$ and the strong augmentation mapping as $\Omega(\cdot)$. Detailed augmentation strategies are described in the experimental section.

The supervised loss for the labeled data can be expressed as

$$\mathcal{L}_s = \frac{1}{B_l}\sum_{i=1}^{B_l} H(y_i, p_\theta(\omega(x_i))), \tag{1}$$

where B_l is the number of data per batch, $p_\theta(\cdot)$ represents the pseudo-labels predicted by the neural network, and $H(\cdot)$ denotes the cross-entropy loss function. This supervised loss is defined similarly to that in traditional supervised learning.

For the unlabeled data, current state-of-the-art SSL methods achieve strong performance by combining consistency regularization and pseudo-labeling. The consistency regularization loss function requires the neural network to produce similar predictions for both weakly and strongly augmented versions of the same sample. High-confidence pseudo-labels are selected to provide more reliable training data.

$$\mathcal{L}_u = \frac{1}{B_u}\sum_{i=1}^{B_u} \mathbb{I}(\max(q_i) > \tau) H(\hat{y}_i, p_\theta(\Omega(u_i))), \tag{2}$$

where $q_i = p_\theta(\omega(u_i))$, we use $\hat{y}_i = \arg\max(q_i)$ as a pseudo-label, $\max(q_i)$ represents the confidence value of the most probable class after applying the softmax function, $\mathbb{I}$ is an indicator function used to filter high-confidence pseudo-labels, and τ is the threshold.

Thus, the total loss function is given by

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_s + \beta\mathcal{L}_u, \tag{3}$$

where α and β are hyperparameters that control the weight of unsupervised learning in the total loss. The entire process is illustrated in Figure 1.

3. Methodology

Our approach focuses on optimizing the training process by utilizing uncertainty quantification. Softmax's overconfidence exacerbates the problem of confirmation bias in SSL, which limits the model's capacity to generalize, despite its benefits, which include cheap computing cost and simplicity of implementation for uncertainty estimation in classification tasks.

On the other hand, our study greatly improves the model's generalization performance by adding more uncertainty information to aid in the learning of a wider variety of representations. In particular, we find that approaches such as FixMatch have a tendency to underuse potentially useful information by discarding a significant proportion of low-confidence pseudo-labels. In order to overcome this constraint, our study applies the concepts of noisy learning and places restrictions on data uncertainty in order to choose trustworthy training data.

According to experimental results, our strategy not only provides faster convergence to a better solution but also exceeds the majority of existing approaches in terms of accuracy. This emphasizes how crucial it is to use uncertainty information in the SSL framework and how much it can do to enhance model performance.

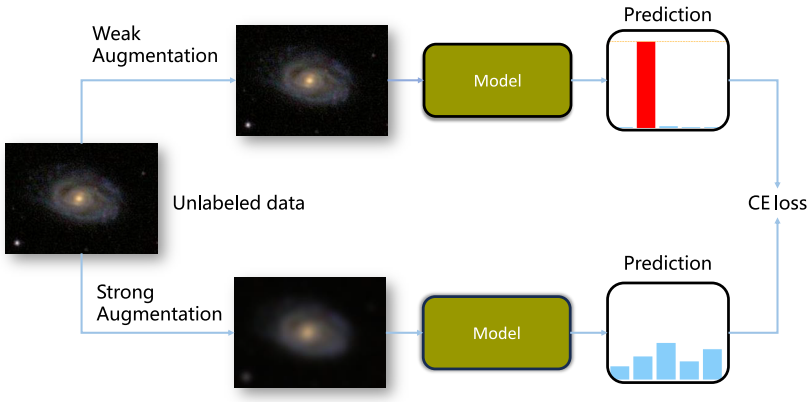


Figure 1. A schematic diagram of the current state-of-the-art SSL framework, trained using consistency regularization and pseudo-labeling methods.

3.1. Differentiated Loss Strategies for Handling Sample Uncertainty and Confidence Levels

Indiscriminate learning is carried out on high-confidence data that surpass a specific threshold in contemporary sophisticated SSL techniques, which introduces several effects on the model's generalization performance.

First off, concentrating on high-confidence data does improve the model's comprehension of features, but it also causes the model to predict erroneously labeled data with excessive confidence. To put it another way, we achieve entropy minimization for pseudo-labels by learning high-confidence data, but this results in an excessive sharpening of the model's outputs. In the early phases of training, this aids the model in rapidly learning the distribution of data; however, in the latter phases, it invariably leads to overconfident and inaccurate predictions for unlabeled data.

Second, this approach makes the issue of class inequality worse. The algorithm improves its ability to identify majority classes as it trains by choosing high-confidence data. The imbalance is exacerbated by the majority classes' quick learning, which raises their percentage in the high-confidence pseudo-labels.

To tackle these problems, as seen in Figure 2, we classify unlabeled data according to confidence and uncertainty scores into easy, ambiguous, and unlearnable data, and apply distinct training techniques to various kinds of data. For detailed calculations and pseudocode of the entire process, please refer to Algorithm 1.

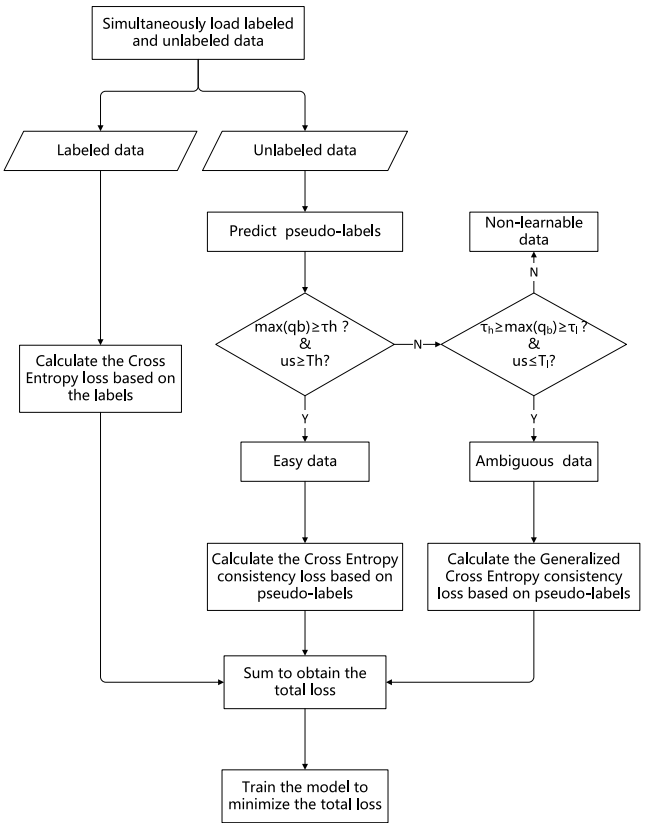


Figure 2. Our method focuses on enhancing the diversity of high-confidence data and utilizing low-confidence data.

3.2. Diversity-based Data Selection Strategy for Easy Data

To find and choose easy-to-learn data, we use a technique that filters high-confidence pseudo-labels by combining uncertainty scores with confidence levels. With this method, the model will be able to learn more quickly and aggressively while still being able to retain and comprehend a wider variety of data representations. The experimental section will include more specifics regarding uncertainty measuring techniques (Section 4).

We compute the entropy of the average prediction probabilities to quantify the model's prediction uncertainty given input data. In information theory, entropy is a metric that quantifies a probability distribution's degree of uncertainty. In particular, the uncertainty of the model's average prediction distribution

following several Monte Carlo iterations is reflected in the entropy of the average prediction probabilities. For the entropy of the average prediction probabilities, we define

$$-\sum_c \left(\frac{1}{T}\sum_{t=1}^{T} p_c^t\right)\log\left(\frac{1}{T}\sum_{t=1}^{T} p_c^t\right), \tag{4}$$

where p_c^t represents the predicted probability of class c during the tth Monte Carlo iteration; T denotes the total number of Monte Carlo iterations. The term $\frac{1}{T}\sum_{t=1}^{T} p_c^t$ computes the average predicted probability for class c across all T iterations.

For the model's uncertainty, we use the average of the prediction entropy

$$\frac{1}{T}\sum_{t=1}^{T}\sum_c p_c^t \log p_c^t. \tag{5}$$

Here, $\frac{1}{T}\sum_{t=1}^{T} p_c^t$ computes the average predicted probability for class c across all T iterations.

Thus, the total uncertainty score can be expressed as

$$u(U, \mathcal{P}) = -\sum_c \left(\frac{1}{T}\sum_{t=1}^{T} p_c^t\right)\log\left(\frac{1}{T}\sum_{t=1}^{T} p_c^t\right) + \frac{1}{T}\sum_{t=1}^{T}\sum_c p_c^t \log p_c^t. \tag{6}$$

Finally, we perform normalization

$$us(U, \mathcal{P}) = \text{clamp}\left(\frac{u(U, \mathcal{P})}{\log(C)}, 0, 1\right), \tag{7}$$

where C represents the total number of classes, and clamp $(\cdot, 0, 1)$ restricts the input value within the range [0, 1]. For the uncertainty score, a higher value indicates greater uncertainty.

We choose data that show both high confidence and relatively high uncertainty during model training in order to improve the model's capacity to learn a variety of representations. High-confidence pseudo-labels, on the one hand, give the model a more solid and trustworthy training basis, which speeds up model convergence and allows for a quick entropy minimization procedure. Conversely, a higher uncertainty score suggests that these data include more diversified and instructive information. This is essential for improving the model's capacity for adaptation and generalization. Combining these two features results in data that not only enhance the model's optimization performance on data that are already available, but also provide a strong basis for tackling issues presented by data that are not visible. As a result, this approach successfully strikes a balance between stability and exploration during model training, providing fresh perspectives and techniques to enhance model performance even further.

Ultimately, we use the following formula as the indicator function

$$\mathbb{I}_e = \mathbb{I}((us >= T_h) \wedge (\max(q_i) >= \tau_h)). \tag{8}$$

3.3. Noisy Learning for Ambiguous Data

Low-confidence pseudo-labels are frequently underutilized for the majority of SSL techniques. Nonetheless, some research has tried to use contrastive learning to take advantage of these low-confidence pseudo-labels. The Generalized Cross-Entropy (GCE) loss function, which is frequently employed in noisy learning, is introduced in this paper (DeVries & Taylor 2017). It is combined with uncertainty scores to assess sample stability and efficiently extract the value of low-confidence data. This method improves the robustness and adaptability of the model by expanding the range of usable data and, to some extent, mitigating the effect of noisy labels on model training.

Therefore, for ambiguous data, we adopt a conservative learning strategy to ensure reliability. This is achieved by setting a minimum confidence threshold and a maximum uncertainty score threshold, combined with the GCE loss function for learning. For ambiguous data, the indicator function is defined as

$$\mathbb{I}_a = \mathbb{I}((T_l \leqslant us \leqslant T_h) \wedge (\max(q_i) \leqslant \tau_l)). \tag{9}$$

3.4. Computation of Loss

The main steps of our algorithm are summarized in Algorithm 1.

Algorithm 1. Pseudocode for our method

Require: labeled batch $\mathcal{X} = \{(x_i, y_i): i \in (1, 2,\ldots,B_l)\}$, unlabeled batch $\mathcal{U} = \{u_i: i \in (1, 2,\ldots,B_u)\}$, unsupervised loss weight w_u; Number of classes C; Model $p_\theta(\cdot)$ with network parameter θ; Strong augmentation function $\Omega(\cdot)$; Weak augmentation function $\omega(\cdot)$.

1: Initialize model parameter θ randomly;

2: Calculate $\mathcal{L}_s$ using:

$$\mathcal{L}_s = \frac{1}{B_l}\sum_{i=1}^{B_l} H(y_i, p_\theta(\omega(x_i)))$$

3: Calculate $u(U, \mathcal{P})$ using Equation (7).

4: Select easy data and ambiguous data from the batch using Equations (9) and (8).

5: For easy data, we apply cross-entropy loss on pseudo-labels for consistency regularization:

$$\mathcal{L}_e = \frac{1}{B_e}\sum_{i=1}^{B_e} \mathbb{I}_e H(\hat{y}_i, p_\theta(\Omega(u_i)))$$

6: For ambiguous data, we use the GCE loss function with noise learning capabilities on pseudo-labels:

$$\mathcal{L}_a = \frac{1}{B_a}\sum_{i=1}^{B_a} \mathbb{I}_a GCE(\hat{y}_i, p_\theta(\Omega(u_i)))$$

7: Compute the total loss:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_s + \eta\mathcal{L}_s + \nu\mathcal{L}_u$$

8: Return: $\mathcal{L}_{total}$.

We have explained the differences among unlabeled data and their corresponding indicator functions. For the supervised learning component, the loss function remains unchanged. For the unsupervised loss, it is defined as follows

$$\mathcal{L}_e = \frac{1}{B_e}\sum_{i=1}^{B_e} \mathbb{I}_e H(\hat{y}_i, p_\theta(\Omega(u_i))), \tag{10}$$

$$\mathcal{L}_a = \frac{1}{B_a}\sum_{i=1}^{B_a} \mathbb{I}_a \mathrm{GCE}(\hat{y}_i, p_\theta(\Omega(u_i))), \tag{11}$$

$$\mathcal{L}_u = \lambda \mathcal{L}_a + \mu \mathcal{L}_e. \tag{12}$$

Here, GCE(·) represents the GCE loss function, and λ and μ are hyperparameters that balance the contributions of the respective loss components.

4. Experiments and Analysis

4.1. Evaluation

A number of important metrics are frequently used in data science and machine learning to assess how well categorization models perform. These consist of Top-K Accuracy, Running Time, F1 Score, Accuracy, and Recall (also called True Positive Rate). A thorough description of each metric may be found below.

4.1.1. Accuracy

The percentage of accurately predicted samples out of all the samples is known as accuracy. It is described as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}},$$

TP: Number of true positives (correctly predicted positive instances); TN: Number of true negatives (correctly predicted negative instances); FP: Number of false positives (incorrectly predicted positive instances); FN: Number of false negatives (incorrectly predicted negative instances).

When the data set's class distribution is reasonably balanced, accuracy works well.

4.1.2. Recall (True Positive Rate)

The percentage of real positive examples that the model accurately detects is known as recall. It is described as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Recall is particularly important in applications where minimizing false negatives is critical, such as medical diagnosis or fault detection.

4.1.3. F1 Score

The F1 Score is a balanced indicator of both precision and recall, calculated as the harmonic mean of the two. It is particularly helpful for data sets that are unbalanced. The definition of the F1 Score is

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Table 1
Comparison of Different Methods on Accuracy

Data Set	Galaxy10 DECaLS			Galaxy Zoo 2	
	Label = 200	400	2000	994	2338
Supervised	0.397	0.545	0.708	0.759	0.835
Fixmatch	0.470	0.587	0.711	0.786	0.863
Ours	0.475*	0.644*	0.768*	0.837*	0.867*

Table 2
Part of the Performance on the Galaxy10 DECaLS Data Set with 2000 Labels

	Top-1	Top-5	Runtime (sec./iter.)	Recall	F1
Supervised	0.7081	0.9698	0.0535*	0.6904	0.6692
Fixmatch	0.7107	0.9780	0.1246	0.6966	0.6576
Ours	0.7683*	0.9815*	0.5060	0.7679*	0.7130*

4.1.4. Top-K Accuracy

In multi-class classification problems, Top-K Accuracy is a performance statistic that is particularly useful when the output is a probability distribution across classes. If the true label is among the top K anticipated classes, the prediction is deemed accurate.

Top-1 Accuracy is the conventional notion of accuracy. A prediction is considered correct only if the class with the highest predicted probability matches the true label.

Top-5 Accuracy considers a prediction correct if the true label is among the top 5 classes with the highest predicted probabilities.

These metrics provide a comprehensive view of a model's performance and help in selecting the most appropriate model and configuration based on specific application requirements.

4.2. Results

As can be seen, our method has achieved excellent results on both the Galaxy10 DECaLS Data set and the GZ2 Data set. When the amount of labeled data is the same, its performance surpasses both supervised learning and existing SSL algorithms. As shown in Table 1, when the labeled data size is only 200, our accuracy is 7.8% higher than that of supervised methods, and when the labeled data size is 2000, our accuracy exceeds the existing FixMatch method by 6%. Table 2 presents additional performance metrics, demonstrating that our method achieves superior results across all metrics except for runtime. Notably, the reported runtime refers to training



Figure 3. The confusion matrix of our method on Galaxy10 DECaLS Data set.

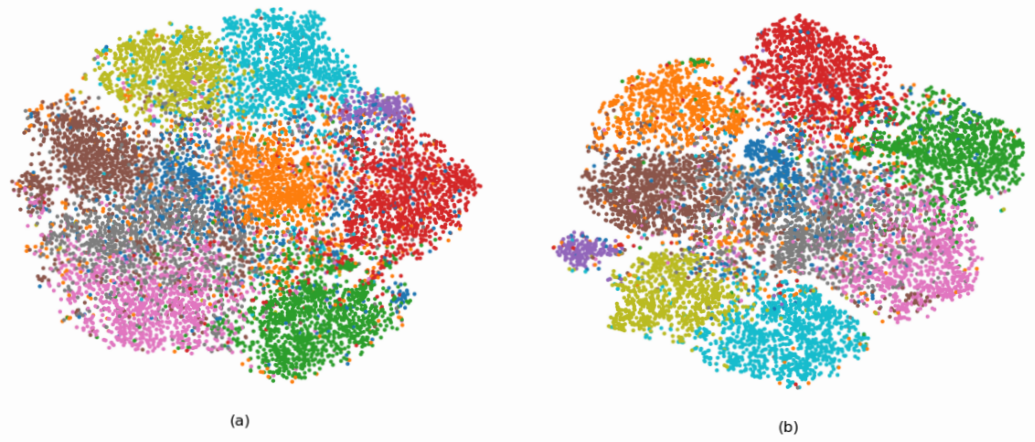


Figure 4. Visualization of features on Galaxy10 DECaLS Data set. (a) supervised. (b) Ours.

time and does not impact the testing time post-training. The higher runtime of our method compared to others is attributed to the additional forward passes required to obtain uncertainty information. However, the remaining evaluation metrics highlight the effectiveness of our approach in mitigating the impact of class imbalance.

In the confusion matrix, displayed in Figure 3, our method successfully learns most categories. However, due to the excessive sparsity of some individual classes, the model experiences forgetting, leading to a class imbalance problem.

By visualizing the generated features in Figure 4, we observe that our method effectively places decision boundaries in sparse regions, thereby improving the classification performance of the model. In particular, minority classes (represented by the purple regions in the figure) can be correctly classified, rather than being misclassified within the clusters of incorrect classes.

4.3. Analysis of Misclassified Samples

4.3.1. Confirmation Bias

In this study, we define confirmation bias as the model's tendency to overfit incorrect samples due to insufficient information (Arazo et al. 2020; Gilhuber et al. 2023). When labeled data are scarce or data utilization is suboptimal, the

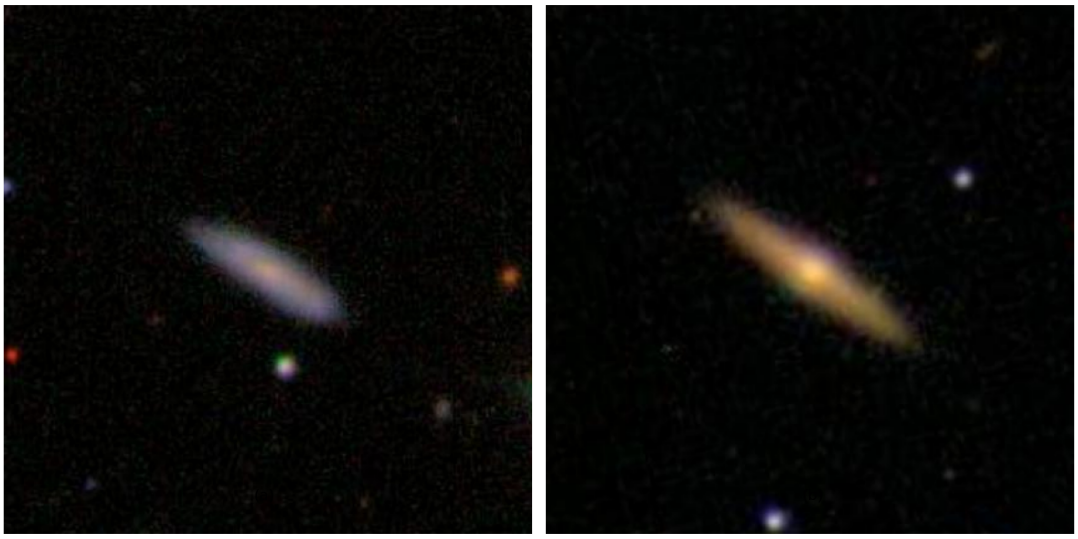


Figure 5. Similar categories in the GZ2 data set. (Left) lenticular. (Right) cigar-shaped.

model is prone to generating inaccurate predictions for complex unlabeled data. During the early stages of training, the model rapidly learns from easily classifiable samples, forming relatively accurate feature representations. However, for data rich in information, the correct predictions are not immediately obvious. This leads to misclassified pseudo-labels in the early stages, which the model fails to correct. When similar data are encountered again, the model continues to train on these erroneous pseudo-labels, resulting in an accumulation of errors. Pseudo-labeling, by minimizing entropy, places decision boundaries in sparse regions, exacerbating confirmation bias, particularly in minority classes within labeled data. Research indicates that techniques such as mixup, setting a minimum number of samples per mini-batch, and other methods (e.g., dropout and data augmentation) can mitigate confirmation bias. To address this issue, we propose training with a diverse set of samples to enable the model to learn more informative feature representations, thereby enhancing data utilization efficiency.

4.3.2. Characteristics of the Data Set

Astronomical data sets possess unique characteristics that present significant challenges for model training, often leading to misclassifications. First, in galaxy morphology classification, the distinctions between certain classes, such as barred lenticular galaxies and cigar-shaped galaxies (Figure 5), are subtle, resulting in increased difficulty for fine-grained classification tasks. Second, galaxy morphology offers limited contextual information, as the background typically consists of empty cosmic space. Finally, galaxy image classification is not always clear-cut, with many irregular galaxies contributing to reduced classification accuracy.

4.4. Implementation Details

For the backbone, we aim to select a simple baseline model without adding too many modules. Both supervised learning and SSL adopt Wide Residual Network (WRN) as our base model. WRN (Zagoruyko & Komodakis 2017) has a 28-layer network structure and can be simply described as a convolutional neural network (CNN) network. However, compared to ResNet, it has a larger width. We use a width factor of 2.

For uncertainty score sampling, we employ MC Dropout (Gal & Ghahramani 2016). MC Dropout performs multiple forward passes during inference, randomly dropping neurons in each forward pass to obtain multiple different outputs. The distribution of these outputs can be used to estimate prediction uncertainty. Meanwhile, for the uncertainty score, we refer to Tran et al. (2019).

After conducting a series of comparative experiments, we selected Stochastic Gradient Descent (SGD) as the optimization algorithm for our model. For the remaining hyperparameters, we adopted default settings: a learning rate of 0.03, a batch size of 16, and a dropout rate of 0.3. To compute the uncertainty score for each batch of data, we perform 10 forward passes during training.

Our training environment consists of a workstation equipped with dual Intel Xeon Gold 8368Q processors, 256 GB of RAM, and two NVIDIA GeForce RTX 4090D graphics cards.

5. Conclusion

In this study, we employ a computational uncertainty score to measure uncertainty and combine it with existing confidence-based methods to enable the model to rapidly learn diverse representations. Additionally, we categorize the data into three types based on classification difficulty–easy data, ambiguous data,

and unlearnable data–using both uncertainty scores and confidence levels. For easy data, the model learns robust and reliable features. For ambiguous data, we selectively train on more credible samples, thereby improving the utilization of low-confidence samples. Compared with supervised learning approaches and other semi-supervised algorithms, our method enhances both the accuracy and generalization capability of the model.

There remains significant potential for further work to enhance the applicability of this research in astronomy. First, while we have not specifically addressed the class imbalance problem, the disparity in the number of different galaxy types in astronomical data sets has led to suboptimal learning for minority classes. Addressing the class imbalance issue could significantly improve model performance. Second, integrating SSL into the data processing pipeline as a component of a broader data engine presents an important direction for future research.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China(NSFC, grant No. 62201398) and the Open Project of Center for Applied Mathematics of Jiangsu Province (Nanjing University of Information Science and Technology).

ORCID iDs

Weibin Chen https://orcid.org/0000-0002-1689-7583

References

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., & McGuinness, K. 2020, in Int. Joint Conf. on Neural Networks, 1 (Los Alamitos, CA: IEEE)
Barchi, P. H., de Carvalho, R. R., Rosa, R. R., et al. 2020, A&C, 30, 100334
Blanton, M. R., Bershady, M. A., Abolfathi, B., et al. 2017, AJ, 154, 28
Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G. 2020, in Advances in Neural Information Processing Systems (Red Hook, NY: Curran Associates, Inc), 22243
Cheng, T.-Y., Li, N., Conselice, C. J., et al. 2020, MNRAS, 494, 3750
DeVries, T., & Taylor, G. W. 2017, arXiv:1708.04552
Euclid Collaboration, Scaramella, R., Amiaux, J., et al. 2022, A&A, 662, A112
Gal, Y., & Ghahramani, Z. 2016, in Proc. Int. Conf. on Machine Learning (New York, NY: PMLR), 1050
Gilhuber, S., Hvingelby, R., Fok, M. L. A., & Seidl, T. 2023, Machine Learning and Knowledge Discovery in Databases: Research Track (Cham: Springer), 330
Gong, Y., Miao, H., Zhou, X., et al. 2025, SCPMA, 68, 280402
Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. 2020, Communications of the ACM, 63, 139
Han, Y., Zou, Z., Li, N., & Chen, Y. 2022, RAA, 22, 085006
Jiang, J., Zhang, J., Li, X., Li, H., & Du, P. 2023, RAA, 23, 115019
Kingma, D. P., & Welling, M. 2022, arXiv:1312.6114
Laine, S., & Aila, T. 2017, arXiv:1610.02242
Lee, D.-H. 2013, in Int. Conf. on Machine Learning, Workshop on Challenges in Representation Learning (Atlanta, GA: JMLR: W&CP), 896
Leung, H. W., & Bovy, J. 2019, MNRAS, 483, 3255
Li, S., Jin, W., Wang, Z., et al. 2024, arXiv:2310.03013
Lintott, C. J., Schawinski, K., Slosar, A., et al. 2008, MNRAS, 389, 1179
Maddox, S. J., Sutherland, W. J., Efstathiou, G., Loveday, J., & Peterson, B. A. 1990, MNRAS, 247, 1P
Martin, G., Kaviraj, S., Hocking, A., Read, S. C., & Geach, J. E. 2020, MNRAS, 491, 1408
Parry, O. H., Eke, V. R., & Frenk, C. S. 2009, MNRAS, 396, 1972
Robertson, B. E., Banerji, M., Brough, S., et al. 2019, NatRP, 1, 450
Sajjadi, M., Javanmardi, M., & Tasdizen, T. 2016, Advances in Neural Information Processing Systems (Red Hook, NY: Curran Associates, Inc), 1171
Slijepcevic, I. V., Scaife, A. M. M., Walmsley, M., et al. 2022, MNRAS, 514, 2599
Sohn, K., Berthelot, D., Li, C.-L., et al. 2020, in Advances in Neural Information Processing Systems (Red Hook, NY: Curran Associates, Inc), 596
Tarvainen, A., & Valpola, H. 2017, in Advances in Neural Information Processing Systems (Red Hook: Curran Associates, Inc), 1195
Tran, T., Do, T.-T., Reid, I., & Carneiro, G. 2019, in Proc. Int. Conf. on Machine Learning (Long Beach, CA: PMLR), 6295
Walmsley, M., Scaife, A. M. M., Lintott, C., et al. 2022, MNRAS, 513, 1581
Wang, D., Cui, P., & Zhu, W. 2016, in Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (New York: ACM), 1225
Wang, H., Wang, Y., Shen, Y., et al. 2025, IJCV, 133, 1476
Wang, R., Wang, H., Luo, Q., Wang, F., & Wu, H. 2025, in Proc. AAAI Conf. on Artificial Intelligence (Washington, DC: AAAI Press), 852
Wei, S., Li, Y., Lu, W., et al. 2022, PASP, 134, 114508
Willett, K. W., Lintott, C. J., Bamford, S. P., et al. 2013, MNRAS, 435, 2835
Wu, D., Zhang, J., Li, X., & Li, H. 2022, RAA, 22, 115011
Xie, Q., Luong, M.-T., Hovy, E., & Le, Q. V. 2020, in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (Los Alamitos, CA: IEEE), 10684
Zagoruyko, S., & Komodakis, N. 2017, arXiv:1605.07146