Semi-Supervised Learning for Materials Informatics

Anonymous Authors[1]

Abstract

This work presents a review and extension of the "FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling" paper by implementing it in the field of materials informatics. The original results of the paper are first reimplemented, and demonstrate that the FlexMatch algorithm (87%) yields lower accuracy than FixMatch (93%) on the CIFAR-10 dataset, contradictory to the paper results. Both of these models are then implemented on a custom materials informatics dataset of crystal structure graph embeddings to predict the stability of materials. Both models yield an accuracy of 58%, indicating that more work is needed to refine this dataset to obtain comparable levels of accuracy to standard semi-supervised datasets.

1. Introduction

Image classification and materials informatics are two distinct fields that share the same fundamental premise: working with large amounts of data to make high-fidelity predictions where labeled data is far outweighed by unlabeled data. In materials science, acquiring new labels of experimental material properties is comparatively more expensive and slower than simulations. In image classification, the cost of labeling images is prohibitively expensive. This has led to the dominance of semi-supervised learning (SSL) algorithms in both fields, which provide a valuable approach for leveraging unlabeled data in cases where labeled data is scarce.

This work reviews and extends FlexMatch, a semi-supervised learning algorithm which uses a flexible confidence threshold to identify high-confidence predictions. This algorithm is extended to a dataset of materials informatics to predict stability of materials from images of their crystal structure. This work is important because it extends the state of the art in semi-supervised learning to a new field, and provides a new set of use cases for novel semi-supervised algorithms to be applied to. In addition, another SSL algorithm, NPMatch, is reviewed. A third paper describing recent advances in materials informatics is reviewed for additional context on the field.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Paper Review: FlexMatch

This work is a review and reimplementation of "FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling", a collaboration between the Tokyo Institute of Technology and Microsoft (Zhang et al., 2021). This reimplementation aims to extend the FlexMatch algorithm proposed in this paper to an additional set of benchmarks in the field of materials informatics.

2.1. Storyline

2.1.1. High-level motivation/problem

The overall goal of this paper is to improve the field of semi-supervised learning, a blend of supervised and unsupervised learning that involves extending supervised models using context of unlabeled data. This is especially useful in cases where labels are rare and unlabeled data is abundant. This paper serves as a benchmark of various semi-supervised methods to advance this field, and addresses concerns with several of the prominent methods used in this field.

2.1.2. Prior work on this problem

Recent semi-supervised learning models such as FixMatch achieved state-of-the-art performance metrics on semi-supervised learning benchmarks (Sohn et al., 2020). Models such as FixMatch use entropy minimization and self-paced learning approaches. Entropy minimization is the method of increasing confidence in predictions via the minimization of entropy in predictions on unlabeled data. Self-paced learning is another method which increases confidence by slowly increasing the difficulty of training samples. However, these approaches have severe limitations, as addressed in the next section.

2.1.3. RESEARCH GAP

This paper identifies an important limitation of FixMatch and other related semi-supervised learning algorithms: these models use a fixed threshold for unsupervised loss, ignoring predictions with confidence metrics below this threshold. This discounts large portions of unlabeled data, often resulting in misleading error metrics.

Another limitation addressed by this paper is that modern SSL algorithms such as pseudo-labeling and unsupervised data augmentation treat all classes equally, independent of their learning difficulties. This rigid structure therefore lacks the architecture to adjust thresholds for each class based on their learning difficulties.

2.1.4. CONTRIBUTIONS

The main contribution of this paper was the introduction of Curriculum Pseudo-Labeling (CPL), a new approach for SSL algorithms to use context from unlabeled data to improve the supervised predictions by using flexible thresholds. This paper improved upon the state-of-the-art FixMatch algorithm by leveraging CPL to develop FlexMatch, an improved algorithm which yields better performance across SSL benchmarks, in addition to converging more quickly. This algorithm was implemented using TorchSSL, a PyTorch-based codebase to aid in the development of new SSL algorithms.

2.2. Proposed Solution

The proposed solution of this paper is a new semi-supervised learning algorithm called FlexMatch. This algorithm uses a flexible threshold to identify high-confidence predictions, and during training adjusts the confidence thresholds to control the quality of the pseudo-labels. This is done by rendering the pseudo labels at different time steps to different classes. The learning effect of class c at time step t is shown in Equation 1.

$$\sigma_t(c) = \sum_{n=1}^{N} 1(\max(p_{m,t}(y|u_n)) > \tau) \cdot 1(\text{argmax}(p_{m,t}(y|u_n)) = c) \quad (1)$$

In this equation, $p_{m,t}(y|u_n)$ is the prediction of unlabeled data u_n at time t. This flexible threshold τ is important to remove noisy, low-confidence predictions, while balancing the confirmation bias of high-confidence predictions. The FlexMatch algorithm is outlined in Algorithm 1:

The FlexMatch algorithm calculates the learning effects for each class individually, then uses these to set flexible thresholds for each class. The inclusion of a warm-up parameter helps prevent overfitting when unused data dominate. This algorithm helps address the issues of a fixed threshold for unsupervised loss and treating all class predictions equally.

Algorithm 1 FlexMatch Algorithm with Active Selection

Require: Labeled dataset D_l, unlabeled dataset D_u, neural network f, number of iterations T, learning rate α, threshold warm-up parameter μ_B, and active selection threshold τ.
Ensure: Trained neural network f.
0: Initialize the neural network f.
0: Initialize the estimated learning effects $\hat{u}_n$ for all unlabeled samples u_n to -1.
0: Initialize the flexible thresholds T_c for all classes c to 0.
0: Initialize the prediction probabilities $p(u_n|f)$ for all unlabeled samples u_n to 0.
0: **while** not converged **do**
0: **for** each class c **do**
0: Compute the estimated learning effect $\hat{u}_n$ for all unlabeled samples u_n in class c.
0: **if** $\max \sigma_t(c) < 1$ **then**
0: Calculate the threshold warm-up parameter $\beta(c)$.
0: **else**
0: Calculate the normalized estimated learning effect $\beta(c)$.
0: **end if**
0: Calculate the flexible threshold T_c for class c.
0: **end for**
0: **for** each unlabeled sample u_b in batch b **do**
0: **if** $p(u_b|f) > \tau$ **then**
0: Update the prediction of unlabeled sample u_b.
0: **end if**
0: **end for**
0: Compute the loss.
0: Update the parameters of f using backpropagation and the learning rate α.
0: **end while**
0: **return** f. =0

2.3. Claims-Evidence

2.3.1. CLAIM 1

Cumulative pseudo-labeling algorithms such as FlexMatch improve the accuracy and efficiency of existing semi-supervised algorithms such as FixMatch.

2.3.2. EVIDENCE 1

When comparing FixMatch and Flexmatch, the authors find that FlexMatch converges faster when training on the

CIFAR-100 dataset (Krizhevsky & Hinton, 2009). Within 50k iterations, FlexMatch reaches similar error levels as 1M iterations of FixMatch. FlexMatch achieves an error rate of 39.94%, while FixMatch achieves an error of 46.43%.

2.3.3. Claim 2

With a high confidence threshold τ, the learning effect of class c is correlated with the number of predictions of c above τ.

2.3.4. Evidence 2

A high confidence threshold removes noisy pseudo labels and reduces confirmation bias. This can be shown through the inverse; a low threshold results in overfitting to initial unlabeled data, leading to confirmation bias. The error rate as a function of threshold τ when training on ImageNet indicates that the error decreases as the threshold increases, then eventually increases as the threshold becomes too high, and no predictions are added to the model. This suggests the existence of a local minimum in error rate as a function of threshold.

2.3.5. Claim 3

A convex mapping function yields the lowest error rate, compared with concave and linear mapping functions.

2.3.6. Evidence 3

This paper tested three mapping functions: concave, linear, and convex. The convex function demonstrated the best performance, while the concave function yielded the worst, with a 10% higher error rate.

2.4. Critique and Discussion

This paper does not effectively explain their choice of datasets: they selected CIFAR-10/100 (Krizhevsky & Hinton, 2009), SVHN (Netzer et al., 2011), STL-10 (Coates et al., 2011), and ImageNet (Deng et al., 2009) and cited them as "common SSL datasets". While it may be true that these are the datasets commonly used in SSL papers, this restricts the possible implementations of SSL in other fields.

In addition, the authors chose to implement a deterministic semi-supervised framework, despite modern probabilistic frameworks yielding comparable benchmark metrics. In general, more explanation of the choices they made and a discussion of the state of the art in this field would have been helpful, rather than solely comparing benchmark scores.

3. Paper Review: NP-Match

The second paper for review is "NP-Match: When Neural Processes meet Semi-Supervised Learning", a collaboration between the University of Oxford and Microsoft Research (Wang et al., 2022). This paper applies neural processes to the task of semi-supervised image classification. This paper will be helpful to the materials informatics reimplementation as it provides a novel semi-supervised learning algorithm to compare to the FlexMatch algorithm.

3.1. Storyline

3.1.1. High-level motivation/problem

This paper identifies semi-supervised learning as a useful class of machine learning algorithms for cases where labeled data is scarce. This is most common in cases where acquiring new labels is prohibitively expensive. By leveraging context from unlabeled data, supervised approaches can yield higher accuracies. A semi-supervised, probabilistic monte carlo (MC) dropout approach is identified as the main approach to be implemented in this field.

3.1.2. Prior work on this problem

This paper summarizes the state of the art in SSL for image classification, gaussian processes, and neural processes. In image classification, pseudo-labeling and consistency regularization methods dominate. For Gaussian processes, this paper identifies that Gaussian processes often require high computational costs. For neural processes (NPs), this paper identifies that state of the art papers incorporate translational equivariance into NPs to create convolutional conditional NPs.

3.1.3. Research gap

This paper identifies that many state-of-the-art models, such as FlexMatch (Zhang et al., 2021) and FixMatch (Sohn et al., 2020), are deterministic pseudo-label approaches. In contrast, probabilistic appoaches fail to achieve comparable accuracies. A new monte carlo probabilistic algorithm, NP-Match, is proposed to address this gap. This is the first work using monte carlo dropout for semi-supervised image classification.

3.1.4. Contributions

This paper proposed NP-Match, which maps neural processes to semi-supervised learning, specifically image classification. The introduction of NP-match disrupts the dominance of MC dropout as the predominant probabilistic model for semi-supervised learning. The source code is available at https://github.com/Jianf-Wang/NP-Match.

3.2. Proposed Solution

This paper proposes a neural-process based, monte carlo semi-supervised machine learning framework called NP-

match. Neural processes are stochastic processes which approximate finite-dimensional marginal distributions. For a probability space (Ω, Σ, Π) and index set $\mathcal{X}$, there exists a mapping $F : \mathcal{X} \rightarrow \mathcal{Y}$ for any point in Ω. NPs parameterize the function F using a vector sampled from a high dimensional multivariate Gaussian distribution. The NP-Match architecture uses this basis, combined with a p neural network, to perform semi-supervised classification.

The NP-Match pipeline is constructed in two modes: training and inference. During training, a set of labeled and unlabeled images are weakly augmented and passed to the NP-Match model. After data augmentation, inference is performed on the weakly-augmented, unlabeled data, and these psuedo-labels are used to train the remaining unlabeled data.

3.3. Claims-Evidence

3.3.1. Claim 1

Kullback-Leibler (KL) Divergence in the evidence lower bound is not a good choice for semi-supervised learning.

3.3.2. Evidence 1

The authors propose a new uncertainty-based skew-geometric Jensen-Shannon (JS) divergence metric. This metric generalizes a previously established JS divergence from 2020, with abstract means and a scalar α to control the divergence skew. This transforms the general JS divergence to the more relevant skew-geometric JS divergence.

3.3.3. Claim 2

NP-Match achieves state-of-the-art performance on public benchmarks.

3.3.4. Evidence 2

In CIFAR10/100 (Krizhevsky & Hinton, 2009) and STL-10 (Coates et al., 2011), NP-Match outperforms all other methods. The error rates for Top-1 models are lower with NP-Match (41.1) than other probabilistic methods (42.69) or deterministic methods (43.66, 41.85, 42.17).

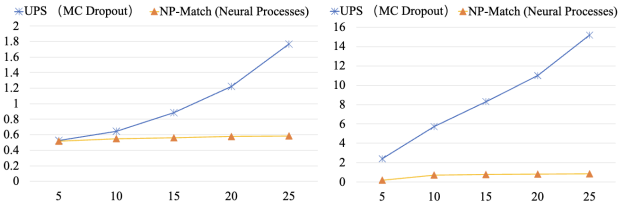
3.3.5. Claim 3

NP-Match estimates uncertainty faster.

3.3.6. Evidence 3

As shown in Figure 1, this is supported by the paper, where they plot the time consumption of NP-Match and MC-dropout methods.

In this figure, the horizontol axes denote the number of predictions in the uncertainty quantification, with time consumption on the vertical axis.



(a) WRN-28-2 on CIFAR-10 (b) WRN-28-8 on CIFAR-100

Figure 1. Time consumption of NP-Match and MC-dropout methods (Wang et al., 2022).

3.4. Critique and Discussion

In this paper, they combined neural processes with a deep learning framework with semi-supervised learning. However, semi-supervised learning frameworks often require a highly iterative approach, and significant re-tuning of hyperparameters. Using a deep neural network therefore seems to be a poor choice for this application, given these constraints, and the authors did not disclose why they decided on a deep learning supervised model. In addition, the same restriction on datasets present in (Zhang et al., 2021) is present: benchmarking on commonly used SSL datasets. While this is useful for comparing models in this subfield, this systemic approach narrows the potential applications of these models.

4. Paper Review: Empirical Interatomic Potentials

This section is a review of "Injecting Domain Knowledge from Empirical Interatomic Potentials to Neural Networks for Predicting Material Properties" from the University of Minnesota (Shui et al., 2022). This paper is fit for the materials informatics reimplementation as it adapts existing materials informatics models to an SSL framework.

4.1. Storyline

4.1.1. High-level motivation/problem

Atomistic modeling has emerged as a critical tool for the prediction of new material properties, due to its relatively cheaper and more efficient cost than empirical property prediction. However, extremely accurate simulations require extensive computational time, and this trade off-has led to the emergence of neural-network based potentials which shortcut this computation cost and aim to preserve the same accuracy. This paper aims to improve the generalizability neural-network-based interatomic potentials trained on DFT data using techniques from semi-supervised learning.

4.1.2. Prior work on this problem

Prior work in this area has focused on developing machine learning EIPs, particularly those based on neural networks, as a more accurate alternative to traditional EIPs. These machine-learned potentials employ general-purpose regression algorithms. However, these NN potentials require large volumes of training data labeled using first-principles quantum mechanical methods, which can be expensive.

4.1.3. Research gap

While neural network potentials have dramatically reduced the computational cost of materials property prediction, this often sacrifices accuracy. These potentials are trained on density-functional theory data, which is often entirely ab initio and removed from experiment. By including information from empirical interatomic potentials, this paper aims to improve the accuracy of these NN potentials.

4.1.4. Contributions

This work identifies three primary contributions. First, that emprical interatomic potentials (EIPs) can be used to improve the accuracy of neural network potentials (NNPs). Second, the paper proposes two strategies to overcome the label scarcity barrier. And third, these modified potentials are benchmarked and compared to other neural network potentials currently being used.

4.2. Proposed Solution

This paper proposes two strategies to improve the accuracy of neural network potentials. The first strategy is to train an auxiliary classifier on empirical interatomic potentials, and select the best-performing EIP for each configuration. The second strategy is to adopt a transfer learning approach by way of multi-task pretraining. This pretrains the NN potential on a set of EIP energies. The first strategy is based on weakly supervised learning, while the second strategy is based on transfer learning.

4.3. Claims-Evidence

4.3.1. Claim 1

The first strategy, based on weakly supervised learning, improves baseline NN performance by 5% to 51%.

4.3.2. Evidence 1

Comparison to existing NN potentials, such as SOAPNet, SchNet, CGCNN, and GemNet on three benchmark datasets (KIM-Si, ANI-Al, AgAu) indicates a lower absolute error (30%, 18%, and 8% improvement) against SoapNet, with similar metrics for the other NN potentials. This is fitting with the claim.

4.3.3. Claim 2

The second strategy, based on transfer learning, improves baseline performance by up to 55%.

4.3.4. Evidence 2

Comparison to existing NN potentials, such as SOAPNet, SchNet, CGCNN, and GemNet on three benchmark datasets (KIM-Si, ANI-Al, AgAu) indicates a lower absolute error (25%, 19%, and 29% improvement) against SoapNet, with similar metrics for the other NN potentials. This claim is true.

4.3.5. Claim 3

Combining the two strategies further boosts performance.

4.3.6. Evidence 3

Experimental results on three benchmark datasets demonstrate that the proposed strategies successfully inject domain knowledge from EIPs to NN potentials and improve the performance of the NN potentials by up to 55

4.4. Critique and Discussion

The results in the abstract are presented as ”improving baseline performance by up to 55 %”. This is misleading, as it makes no claims as to how significantly the baseline performance is increased by only providing an upper bound. Based on the data, this value ranges from 2% to 55%. In addition, it is somewhat surprising that across the dozens of tests and combinations performed, their models never yielded a lower accuracy than the state of the art. This could suggest some form of overfitting - more hyperparameter optimization may have benefited this paper.

5. Implementation

5.1. Implementation Motivation

The objective of this work is to rerun the experiments in the FlexMatch paper, and to extend these SSL algorithms to existing benchmarks in materials informatics. This is a valuable addition to semi-supervised literature because the field of materials informatics is extremely well suited to semi-supervised learning. For example, the task of predicting material synthesizability has available to it a large amount of unlabeled data via computational methods such as density-functional theory, molecular dynamics, and graph-based neural network potentials. Labeled data on successful examples of materials synthesis is limited, and expensive to acquire; therefore, semi-supervised learning plays a significant role in solving this problem.

A successful extension of this paper's work on the field of

materials informatics would demonstrate which SSL algorithms are best suited to this problem, allow comparison to existing SSL methods in materials informatics, and in doing so, extend the original results of the paper to another field which is not as well documented in the machine learning community.

5.2. Implementation Plan and Setup

The authors of the FlexMatch paper have published their implementation open-source at https://github.com/TorchSSL/TorchSSL. This codebase contains 12 semi-supervised algorithms and 5 datasets (CIFAR-10/100, SVHN, STL-10, and ImageNet).

To extend this codebase into the field of materials informatics, there are two axes of study: semi-supervised algorithms to materials informatics datasets, and adding materials informatics algorithms to SSL datasets. While both axes were intended to be pursued, due to time constraints the first axis is the primary focus of this work and the second is left as future work. The materials informatics datasets were obtained from https://github.com/materialsproject/matbench, a repository of datasets for benchmarking in materials informatics hosted by the Materials Project (Dunn et al., 2020). This repository is known as an "ImageNet for materials science". The prediction of stability was the primary machine learning target, as it is important in the synthesis of new materials and well-suited to semi-supervised learning. Connecting the SSL algorithms studied in this work to these datasets of materials informatics will significantly expand the domain of models applied to materials informatics, and provide a new set of use cases for novel semi-supervised algorithms to be applied to.

The codebase structure used by TorchSSL, the repository associated with the original FlexMatch paper, is unflexible in its implementation. It was therefore required to adapt these models to a more general framework, as it was determined that TorchSSL was no longer being maintained and failed to pass basic tests. Therefore, the decision was made to instead switch the primary implementation to https://github.com/microsoft/Semi-supervised-learning. This *semilearn* repository is maintained by Microsoft, and is a more general framework for semi-supervised learning which implements several state-of-the-art semi-supervised machine learning frameworks.

5.3. Implementation Details

5.3.1. Configuring a Working SSL Environment

The majority of the code for this project was run using Google Colab, as this allowed for the use of a T4 GPU to train the models in the required timeline. In addition, the use of Colab made setting up the required dependencies for each of the environments much easier, as packages could simply be `!pip install semilearn`. The FixMatch and FlexMatch semi-supervised models were obtained from the *semilearn* codebase, which provides a friendly wrapper to load pre-trained models. The materials informatics dataset was obtained and preprocessed with my own code, while the experiments of the SSL models on this custom dataset were a combination of the *semilearn* codebase and my own code.

5.3.2. Running Basic Tests with FixMatch/FlexMatch

To validate the primary claims of the FlexMatch paper, the first task was to replicate the results of the paper on the CIFAR-10 dataset. This was done by testing the codebase on the CIFAR-10 dataset for both FixMatch and FlexMatch algorithms, and comparing the results to the results reported in the paper. A 40-label split size was selected for this experiment, mirroring the FlexMatch paper. The reported parameters from the model include accuracy, precision, recall, and F1 score. This was repeated for the ImageNet dataset, to have a benchmark on another publicly available machine learning dataset.

To validate the convergence rate of FlexMatch, the training loss of both FixMatch and FlexMatch on the CIFAR-10 dataset was plotted using a plotly graph to compare the relative losses of each model over the same training period.

5.3.3. Creating the Custom MatBench Dataset

The primary contribution of this paper is a system for connecting materials informatics datasets to the semi-supervised algorithms implemented in the *semilearn* codebase. This repository provides a loose structure for adding a custom dataset by instantiating a new `BasicDataset` object. These objects take in a set of image data and labels, and provide a method to load the data into the model.

As FlexMatch and FixMatch are image-based classification algorithms, the materials dataset was required to fit this input and output format. The chosen dataset for this experiment was to predict if a material is *stable*, a binary classification task. Specifically, these labels correspond to whether a given material structure and composition lies on the convex hull of the potential energy surface, as determined by *ab initio* quantum mechanics simulations, namely, density functional theory with the PBE (Perdew-Burke-Ernzerhof) exchage-correlation functional. While not a perfect measure of experiment, this metric is often used in the field of materials informatics as a proxy for synthesizability. The MatBench repository uses this metric as one of its prediction tasks, with the input of a crystallographic structure. However, this structure file is not in the format required by the *semilearn* codebase, and therefore required preprocessing.

To convert the crystallographic information into a format readable by an image-based classification algorithm, the following steps were taken:

1. Crystallographic and stability information were queried using Materials Project's API
2. The crystallographic structure was converted into a graph representation using the pymatgen library
3. This graph representation was transformed into a 32x32 image

The crystal structure and derived graph representation for NaCl are shown in Figure 2.

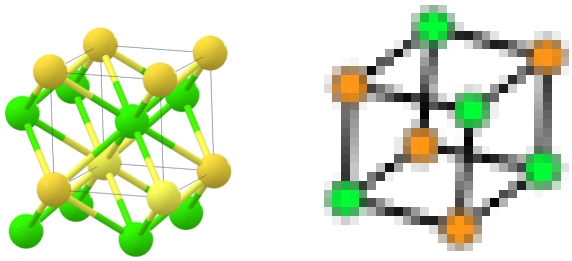


Figure 2. Left: NaCl crystal structure, from MaterialsProject. Right: 32x32 graph representation of NaCl crystal structure.

This idea to use the graph embedding instead of the crystal structure itself was inspired by recent advancements achieved using graph neural network-based prediction of material properties, as described in (Shui et al., 2022). These embeddings map crystallographic structures to graphs with atoms as graph nodes and atomic bonds as graph edges. In this work, this embedding was then transformed into an image, where color denotes the literature valency of the atom at that graph node. This was done to add some basic bonding information to the image, as the graph representation alone does not contain this information. This image was then resized to 32x32, as this is the required input size for the FixMatch and FlexMatch models. This was done on 10,000 structures obtained from Materials Project, a sampling of which is shown in Figure 3.

Prior to training, the image dataset was subject to augmentation using the following techniques: random horizontal flipping, random cropping of a 32x32 image section with padding, conversion of the image to a tensor, and normalization of the pixel values.

5.3.4. Running FlexMatch on the custom dataset

The final step of this implementation was to run the FlexMatch and FixMatch algorithms on the custom dataset. This

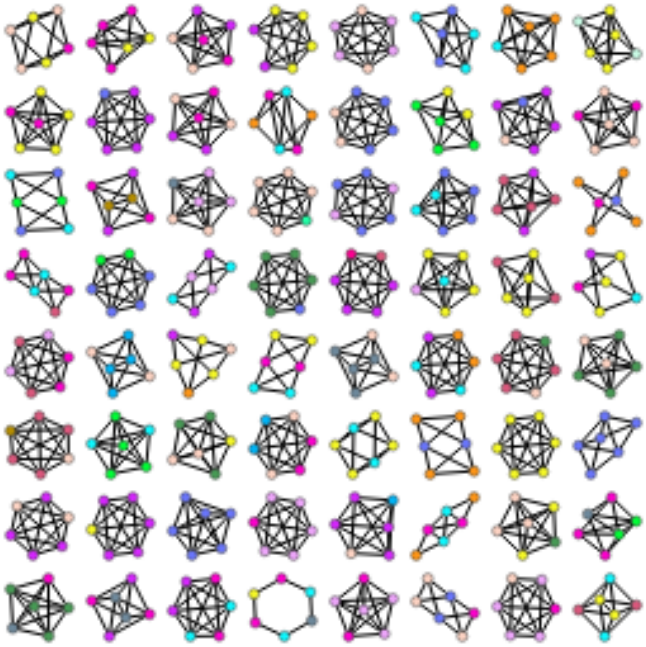


Figure 3. A sample of the images used in the custom dataset to represent crystal structures.

was done by loading the newly created dataset into the *semilearn* codebase and training the FlexMatch and FixMatch neural networks on this dataset. The models used the vit_tiny_patch architecture with pretrained weights. The training process consists of 1 epoch, with 5000 training iterations, 500 evaluation iterations, and logging every 50 iterations. Within the custom materials dataset, there were 10000 entries. Of these, 500 were used for labeled training, 1000 were used for validation, and 8500 were used for unlabeled training.

5.4. Results and Interpretation

5.4.1. Validating FlexMatch Paper Results

After setting up a working installation of the USB library, the first task was to validate an experiment from the FlexMatch paper. The following results are reported from the as-implemented codebase, using a preliminary comparison of FixMatch and FlexMatch in Table 1 below to investigate Claim 1.

Table 1. Measured FixMatch and FlexMatch Performance on CIFAR-1 dataset

Method	Accuracy	Precision	Recall	F1
FixMatch [8]	0.9362	0.9380	0.9362	0.9359
FlexMatch [9]	0.8668	0.8820	0.8670	0.8534

As a comparison, the results of the FlexMatch paper are

shown in Table 2 below.

Table 2. Reported FixMatch and FlexMatch Performance on CIFAR-10 dataset

Method	Accuracy
FixMatch [8]	0.9253
FlexMatch [9]	0.9503

The information in Table 1 and 2 does not support the claim that FlexMatch yields a higher accuracy than FixMatch, as demonstrated by the higher accuracy reported by FlexMatch than FixMatch in Table 1. We do not expect this behavior, as FlexMatch implements a flexible threshold to identify high-confidence predictions, while FixMatch does not. This is a surprising result, and may be due to the difference in codebases used to implement these models. The FlexMatch paper used the *TorchSSL* codebase, while this implementation used the *semilearn* codebase. This may indicate that the *semilearn* codebase is not as well suited to the FlexMatch algorithm as the *TorchSSL* codebase.

To validate that the convergence rate of FlexMatch is faster than FixMatch, the following plot was constructed from the training loss of both FixMatch and FlexMatch on the CIFAR-10 dataset. This plot is shown in Figure 4.

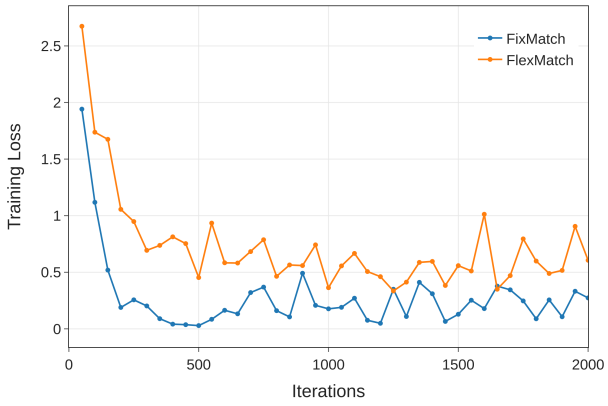


Figure 4. Convergence of FixMatch and FlexMatch on CIFAR-10 dataset

From this figure, we can also reject the claim that FlexMatch converges faster than FixMatch. The convergence as a function of training iterations is similar for both models, and does not indicate that FlexMatch converges faster than FixMatch.

5.4.2. Testing FlexMatch on MatBench Dataset

The obtained accuracy of the FixMatch and FlexMatch algorithms on the custom MatBench dataset is shown in Table 3 below.

Table 3. Measured FixMatch and FlexMatch Performance on MatBench dataset

Method	Accuracy	Precision	Recall	F1
FixMatch	0.582	0.570	0.579	0.562
FlexMatch	0.586	0.579	0.590	0.569

The convergence of the FixMatch and FlexMatch algorithms on the MatBench dataset is shown in Figure 5.

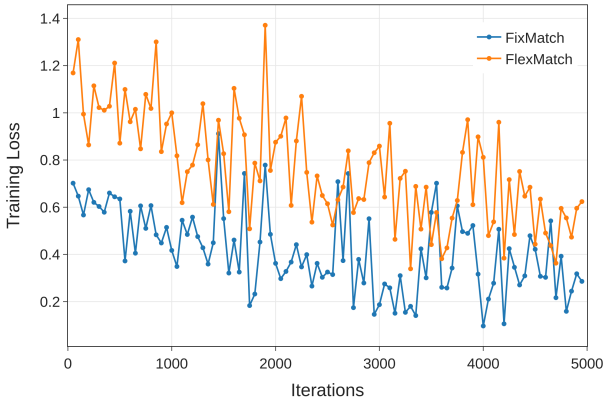


Figure 5. Convergence of FixMatch and FlexMatch on MatBench dataset

From these results, it appears that the model does better than random chance, but does not perform as well as the models on the CIFAR-10 dataset. This is likely due to the difference in the datasets, as the CIFAR-10 dataset is a well-established benchmark for semi-supervised learning, while the MatBench dataset is a custom dataset created for this project. In addition, the convergence plot seems to indicate that more training may have continued to lower the total error, though there was not time for this in the project timeline. This is left to future work.

It is worth nothing that the images in the MatBench dataset are much harder to classify, as while theoretically there is enough information in the graph embedding to determine its stability, this is not a trivial task. The image itself requires more information describing the chemical system, and a better representation of the bonding information than fuzzy lines. This is a limitation of the graph embedding used in this project, and is a potential area for future work.

6. Conclusion and Discussion

The objective of this project was to implement the FlexMatch and FixMatch algorithms for semi-supervised learning on custom materials informatics datasets. We conducted experiments on both the CIFAR-10 dataset and the MatBench dataset to evaluate the performance of these algorithms.

From our results on the CIFAR-10 dataset, it was observed that the FlexMatch did not yield a higher accuracy than FixMatch, contrary to the claim made in the FlexMatch paper. This is likely due to the differences in the codebases used for implementation, as the original paper used the TorchSSL codebase and this work used the semilearn codebase. This training also demonstrated that FlexMatch did not converge faster than FixMatch, and in fact both converged similarly.

On the MatBench dataset, we achieved accuracy levels that exceeded random chance but were less than the performance on the CIFAR-10 dataset. This discrepancy can be attributed to the difficulty of classifying the chemical bonding environments represented in the MatBench dataset. In summary, this work is a promising start to the merging of two exciting fields, but requires more work to be a useful contribution to the field of materials informatics.

References

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011. URL https://proceedings.mlr.press/v15/coates11a.html.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009. URL https://ieeexplore.ieee.org/abstract/document/5206848/.

Dunn, A., Wang, Q., Ganose, A., Dopp, D., and Jain, A. Benchmarking materials property prediction methods: the Matbench test set and Automatminer reference algorithm. *npj Computational Materials*, 6(1):1–10, September 2020. ISSN 2057-3960. doi: 10.1038/s41524-020-00406-3. URL https://www.nature.com/articles/s41524-020-00406-3. Number: 1 Publisher: Nature Publishing Group.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009. URL http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf. Publisher: Toronto, ON, Canada.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011. URL https://research.google/pubs/pub37648/.

Shui, Z., Karls, D., Wen, M., Nikiforov, I., Tadmor, E., and Karypis, G. Injecting Domain Knowledge from Empirical Interatomic Potentials to Neural Networks for Predicting Material Properties. *Advances in Neural Information Processing Systems*, 35:14839–14851, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/5ef1df239d6640a27dd6ed9a59f518c9-Abstract-Conference.html.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems*, volume 33, pp. 596–608. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/06964dce9addb1c5cb5d6e3d9838f733-Abstract.html.

Wang, J., Lukasiewicz, T., Massiceti, D., Hu, X., Pavlovic, V., and Neophytou, A. NP-Match: When Neural Processes meet Semi-Supervised Learning. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 22919–22934. PMLR, June 2022. URL https://proceedings.mlr.press/v162/wang22s.html. ISSN: 2640-3498.

Zhang, B., Wang, Y., Hou, W., WU, H., Wang, J., Okumura, M., and Shinozaki, T. FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18408–18419. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/995693c15f439e3d189b06e89d145dd5-Abstract.html.