Semi-supervised Image Classification

Eyad Alshami
7015715
eyal00001@stud.uni-saarland.de

Shreyash Arya
7015279
shar00001@stud.uni-saarland.de

Abstract

The projects aims to tackle the problem of data scarcity for training data-hungry neural networks using the semi-supervised learning which relies on small amount of labelled dataset. Three techniques namely - Pseudo-labeling, VAT and Attention-based methods have been implemented and analyzed on CIFAR-10 and CIFAR-100 datasets.

1 Introduction

Neural networks have been vastly applied to the image classification problems due to its ability to scale and generalise on a wide multitude of tasks. The main bottleneck is the data as training neural networks requires a large amount of dataset and acquiring such dataset is an expensive task. Hence, the method of Semi Supervised Learning (SSL) is applied which uses a very less amount of labelled samples to train the deep networks. In this project, we explored three methods namely - Pseudo Labeling [5], Virtual Adversarial Training (VAT) [6] and a custom attention-based method built on Fixmatch [7] that is our novel idea for performing SSL. We tested these methods over two datasets - CIFAR10 and CIFAR-100 [4] with 250 and 4000 and, 2500 and 10000 labelled samples respectively.

The Semi-Supervised Learning (SSL) techniques are used to improve generalization using labeled and unlabeled data. The clustering assumes that in order to improve generalization performance the decision boundary should be on the low-density region [5]. Also, entropy is a measure of class overlap and as the overlap decreases, the density of data points get lower at the decision boundary. Hence, task 1 and 2 both works on entropy minimization principle that in-turn improves generalisation. In our experiments, we notice with increasing complexity of SSL techniques, results improve and generalisation power increases.

Pseudo-labelling [5] is the earliest techniques in all the three tasks discussed in this project and shown to prove simple and better for semi-supervised learning tasks from that time. Before it, methods were introduced for combined minimization of the loss [11-13]. The method in effect shown to be equivalent to Entropy Regularization introduced by Grandvalet et al. [14].

In reference to task 2 (VAT), many methods have been introduced in past for the regularizing the random perturbation for improving the robustness of the model and hence can also be applied to the task of SSL. Related work section in [6] gives a nice dives into the methods with best performers as the adversarial training based methods [10] (on which idea VAT is build) and the generative models [15-18].

Task 3 is based on the Fixmatch [7] training framework, where the model is presented with two versions of the input image, one is weakly augmented and the other is strongly augmented. The prediction of the weakly augmented image is considered as a pseudo label if its probability exceeded a specific threshold, which then is treated as a ground truth label for calculating the loss of the strongly augmented image's prediction. By this, the model learns to ignore the augmentation and will be more robust against unseen examples from the same classes it was trained on.

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

The following sections are divided as follows: In section 2, we discuss the methodology followed in three tasks and the experimental settings. Further, in section 3, we discuss the results from all the tasks and finally in section 4, we conclude.

2 Methodology

The following section is divided into three techniques namely Pseudo Labelling [5], VAT [6] and Attention-based techniques for SSL corresponding to each task. Each task (from 1 to 3) has increasing technique complexity.

2.1 Task 1: Pseudo-labeling

The name of this task conveys the approach it uses to generate labels and do self-supervised learning. We start with an empty set of pseudo labels and the labeled dataset. For each training epoch, we train on a combination of those sets, after which we pass the entire unlabeled dataset through the model and predict the labels for each unlabeled example. Based on the probability of the assigned class we decide whether to consider it as pseudo label or not. This is done by comparing the probability against a user-specified threshold (τ) and accumulating only the ones who exceeded this threshold.

For the model to generate a high probability for one class, either the mode is pretrained, or the predicted class is easy to classify. That is why we trained the model for several epochs only on the labeled data, so it can have some good point to start learning from. However, this learning framework puts some real risks on the classification, because models might be caught in a confirmation bias scenario, where the model parameters are close to one data mode and recognize one class better than the other. In this case, the model starts to accumulate pseudo examples from only that class which is the easiest class for it to predict, and as more pseudo examples are added and trained on as the confidence of the model raises for that specific class and not for other classes. During experiments, we noticed the classes predicted by the model at the beginning while training only succeeded at predicting one class for quite some time, and depending on the hyper-parameters, optimization method, and the initialization of the parameters, it can stuck and never converges to a better spot.

2.2 Task 2: VAT

Task 2 uses regularisation-based method which tries to make model robust against the random and local perturbations. It is built upon the idea of adversarial training [10] which tries to improve generalisation by assigning similar label to data point as the neighbor in the adversarial direction. Adversarial direction is in which the prediction of the model deviates the most while VAT works on distribution level and hence can be applied to semi-supervised setting. It looks for the most anisotopic direction and smooths it to reduce the deviation in the current inferred output distribution by regularising the sensitivity of output with respect to input (hence, parametric invariant regularization). In implementation, there are three parameters for norm contraints (xi, eps) and the power iteration for backpropogration in VAT ($iter$) which are tuned for the best results. The model is trained on the combined loss function of the cross-entropy loss on the labelled dataset and the VAT loss on the set of randomly sampled unlabelled dataset which works as regularizer term. Conditional entropy minimization loss can also be incorporated with the VAT loss which can be toggled using use_entmin parameter. The VAT loss calculation is implemented following the algorithm given in problem statement document.

2.3 Task 3: Attention-based distance loss

The Task 3 builds upon improving the state-of-the-art results achieved by Kihyuk Sohn et al. [7] which uses the idea of strong vs. weak augmentation learning. The learning framework consists of two pipelines, the first one is fed with a weakly augmented version of an input image, and the second pipeline is fed with a strongly augmented version of the same input image. The same model is used in both pipelines. However, in the second pipeline, the model computes its loss based on the output of the first pipeline. The authors tried different combinations of augmentations and provided a thorough analysis of what works best. In our approach, we used their framework of using weak vs. strong augmentation training and their recommendations regarding the augmentation transforms. The framework used in Fixmatch [7] aims to push the model to learn and ignore the augmentation and transformation, and focus on the high-level features that help in correctly classifying the image. It gave us an idea of extending this learning to include attention learning.

Our intuition is that the model must concentrate on similar high-level features in the two pipelines. To make it simple, the idea is to generate attention maps in both the pipelines and compute the euclidean distance between the coordinates of the four (later reduced to two) largest values in the attention maps of the two pipelines. However, the gradient can't be computed with respect to coordinates and parameters can't be updated according to that, so *Softmax* function is used to get a distribution over the attention map and at the same time get indication where the highest values reside. By this, we argue that besides the max values we got in the shape of probabilities, we also get an indication of where the max values are located. Also, rather than comparing coordinates, two distributions are compared using the KL divergence distance which is included in the final loss.

As deep convolutional neural networks capture high-level features from images, adding attention to it guide the network for image classification. Our approach uses attention-based loss, where attention is computed using an attention module consisting of two 1×1 *Conv layers*, a *Batch normalization layer* and a *ReLU non-linearity function*. Given the features' maps generated by the last non-classifying layer in the model (*Wideresnet*), two attention maps are generated. After generating the attention maps, an element-wise summation is applied over them. The idea is to combine maps in one representative attention where peaks of magnitude representing each attention map are present. These attention maps are preserved to compute the attention loss in later steps.

Further, bi-pooling is done by applying an element-wise multiplication between the generated attention map and the features maps. If the probability of the predicted class exceeded the specified threshold, then the attention maps from the two pipelines are flatten and *Softmax* is computed for each map. It gives two distributions that are compared using the KL divergence distance. Finally, the calculated distance is added to the final loss.

The method can be summarized in the following steps (only for examples that exceed the success threshold):

- Feed the input image and generate features maps.
- Apply the attention module on the maps to generate two attention maps.
- Apply element-wise summation of the attention maps to generate one final attention map.
- Apply bi-pooling by computing element-wise multiplication between the final attention map and the features maps.
- Compute the *Softmax* for both attention maps.
- Compute the KL divergence of the two probability distributions.

The main limitation in our approach is the difference between the two augmented versions of the input image, especially if the strongly augmented version has gone through a cut-out transform. In that case, the high-level features generated must be lacking the features corresponding to the cut-outed part which is present in the weakly augmented images. This makes the KL distance useless if the main features are absent in one input and present in the other.

2.4 Experimental Setting

For all tasks, CIFAR-10 and CIFAR-100 datasets are used for training and testing. For task 1, two settings for number of labelled samples - 250 and 4000 for CIFAR-10 and, 2500 and 4000 for CIFAR-100 are considered with threshold values (τ = 0.95, 0.75, 0.6). Also, the warm-up epochs (training only on labelled data available initially) are kept at 10 and normal epochs at 300. For task 2, same four settings (two for each dataset) of number of available labelled samples are used as first task. The warmup epochs are set at 35 and normal epochs at 300 for task 2. Also, the VAT parameters are kept at $xi = 10, eps = 10, iter = 5$ for the best results. For task 3, we have trained model on 200 epochs.

For task 1 and 2, warm-up training is done to make the model learn the structure from the labelled dataset provided in each case. It helps model to predict better on the unlablled dataset and have a generalisation boost. Note, the warm-up epochs are included in the total number of epochs. Also, all the tasks used randomly initialized models, no pre-trained models were used. For task 3, we did not used a randomly initialized Wideresnet model, and trained for 100 epochs for all the datasets. We used the three thresholds (0.95, 0.75, 0.6) specified in the instructions.

3 Results

In this section, we present the results for all the tasks. The evaluation metric used is the test error rate which which is defined as (1 - test accuracy).

3.1 Task 1 results discussion

Table 1: Test Error Rates for Pseudo-Labelling

	CIFAR-10		CIFAR-100	
Threshold	250 labels	4000 labels	2500 labels	10000 labels
0.95	66.682	30.042	80.583	63.644
0.75	68.908	26.534	81.916	62.475
0.6	73.936	25.113	80.654	61.36

Comparing datasets:

Both the datasets given (CIFAR-10 and CIFAR-100) are balanced with same number of samples per class in all the four settings (250, 4000, 2500 and 10000 labelled samples) but CIFAR-100 has more number of classes. Hence, it makes it harder for model to predict 100 classes as compared to 10 classes in CIFAR-10. So, we can notice a general pattern of increased error rates with increasing number of classes for the same hyperparameter settings.

Comparing available labelled samples settings:

The settings with 250 labelled samples (CIFAR-10) and 2500 labelled samples (CIFAR-100) are comparable in terms of having same number of samples per class (25) but CIFAR-100 has more classes. Also, the case with 4000 labelled samples in CIFAR-10 will see the largest examples per class (400) and has less number of classes (10) giving the lowest test error rate ($25.113\%; \tau = 0.6$). In contrast, 2500 labelled sample case for CIFAR-100 has lowest examples per class (25) and highest number of classes (100) giving the highest test error rate ($81.916\%; \tau = 0.75$).

Comparing vertically (changing thresholds):

For CIFAR-10 with 250 labelled samples, as the threshold value (τ) goes down, the model becomes more lenient and makes more errors. This is because the model only sees 25 examples per class and is under-fitted due to which it predicts near random pseudo-labels for the unlabelled data. The warmup training done before predicting the pseudo-labels helps but with very less labelled data, it still performs badly. Also, for higher threshold ($\tau = 0.95$) as compared to the lower ($\tau = 0.6$), the model is less reliant on the predictions learnt for the labelled data and chooses more randomly. Only the exact samples or samples with slight change would be classified correctly.

The case with 4000 labelled samples in CIFAR-10 is the best model as it sees the most number of labelled samples per class (400) and hence the results get better (30.042% to 25.113%) as we reduce the threshold ($\tau = 0.95$ to 0.6; increasing the reliance on model for pseudo label predictions). This can be understood as the generalisation power of the model gets better as it encounter more variations of samples per class and is more confident of the class label prediction.

For CIFAR-100, the case with 2500 labelled samples is similar to the CIFAR-10 with 250 labelled samples but it is much worse due to more number of classes for prediction. It requires more samples to improve it's predictions (as can be seen in 10000 labelled samples case). This case is worse in terms of performance (approx 81%) and change of threshold (τ) doesn't affect the performance as it is almost random.

The 10000 labelled samples case in CIFAR-100 shows improving results (63.644% to 61.38%) as the threshold decreases and reliance on model prediction increases. It is similar to the case of 4000 labelled samples in CIFAR-10 but due to more number of classes, the performance is worse compared to it.

In Figure 1 in Appendix, we show the loss and the accuracy curves of the model trained on CIFAR-10 and CIFAR-100 datasets respectively.

3.2 Task 2 results discussion

Table 2: Test Error Rates for VAT

CIFAR-10		CIFAR-100	
250 lables	4000 labels	2500 labels	10000 labels
68.362	27.557	83.454	60.683

The results for task 2 follows a similar pattern to task 1 in terms of change in test error rates with respect to the different datasets (CIFAR-10 and CIFAR-100), number of labelled samples and the number of classes. Miyato et al. [6] mentioned the best parameter settings for the CIFAR-10 (4000 labelled samples) to be ($xi = 8, eps = 8, iter = 1$) while in out experimentation, we found ($xi = 10, eps = 10, iter = 5$) giving the best results shown in Table 2.

The test error rates are less for the more number of labelled samples - 27.557$\%$ for 4000 labels in CIFAR-10 and 60.683$\%$ for 10000 labels in CIFAR-100. In general, CIFAR-10 performs better than CIFAR-100 due to lesser number of classes. Also, similar to task 1, the best (27.557$\%$) performance overall is for the case with 4000 labels and 10 classes (400 samples per class) and worst (83.454$\%$) for 2500 labels and 100 classes (25 labels per class). Similar intuition could be followed that the generalisation power of the model gets better as it encounter more variations of samples per class and is more confident of the class label prediction.

Ideally, VAT is a more advanced technique compared to Pseudo-labelling using the idea of adversarial training and modifying the loss to make the model robust to do predictions on the similar samples. This in turn could be used for labelling unlabelled samples. The objective function looks at the max component (most anisotropic direction to be smoothed) as compared to the expectation (in case of random perturbation training [10]) which works in favour of performance. But in our case, we don't see any significant improvements (except for CIFAR-100, 10000 labelled samples case) which could be due to various implementation decisions taken for the VAT - random initialization, learning rate scheduling, non-pretrained model usage, number of epochs etc. We also experimented with incorporating entropy minimization in loss for boosting the performance as suggested in [6] but results were not significantly improved. Also, the power iteration parameter ($iter$) is stated to saturate at at value 1 but in our case we got better results at ($iter = 5$). This could be possible as different labelled settings have different spectrum of Hessian (as also mentioned in [6]).

In Appendix, in Figure 2, we show combined loss and accuracy curves for the models trained on CIFAR-10 and CIFAR-100 datasets respectively. In Figure 3 and 4, we show tensorboard output for CIFAR-10, 4000 labelled samples and CIFAR-100, 2500 labelled samples respectively. Also, in Figure 5, we visualize the original and the perturbed images for both datasets.

3.3 Task 3 results discussion

Table 3: Test Error Rates for Task3

	CIFAR-10		CIFAR-100	
Threshold	250 labels	4000 labels	2500 labels	10000 labels
0.95	73.47	24.43	87.58	63.43
0.75	90.01	26.93	87.68	65.51
0.6	75.31	36.03	89.79	70.23

For this task with attention loss, results did not show significant improvements from the previous tasks. From table 3, we can see that when the number of labeled examples are high, the results are good as compared to when the number of labeled examples are low - in which case the error rates are really high. This effect can be attributed to the usage of attention loss and thresholding.

For 250 labeled samples, the model does not have enough examples for the attention module to adjust well to attend the best features, and decreasing the threshold makes the error grow because the attention is now seeing more and more stochastic inputs which have no patterns to detect. Hence, a low threshold allows more randomness to the process. Whereas, when we have more labeled samples, we can see that with high threshold the model's error rate is the lowest between all the other trials, and as the threshold gets lower the error rate gets higher. We argue that this is because the attention module gets confused when seeing examples with low probability classes.

Our intuition is that attention needs more training examples and more confident predictions as it does not build its output as a convolutional neural network that generates its prediction based on a hierarchy of features that build on top of each other and could form and rely on multiple detected features. On the contrary, we argue that the attention module we used needs a stable input to detect patterns and focus on what matters more for the output.

In experiments with CIFAR-100 dataset, we can see that the performance is quite bad when the number of labeled samples is 25 per class similar to the case of CIFAR-10 with 250 labels. Also, with a larger number of classes, the job becomes even harder. We do not see that sharp decrease when lowering the threshold, and this couldn't be attributed to a specific reason. In the case of 10000 labeled samples, we can see that it has a lower error rate compared with the model trained with only 2500 labeled samples. This can be attributed to the increase in the number of labeled examples and thus more certain attention maps are generated. Though like all other trials, we can see that the error rates increase with lowering the threshold and that again in our opinion can be attributed to lower certainty rates the model is producing for the labels.

In general what we learned from task 3 are as follows:

- We think that attention is a sensitive method that relies on robust classifications and large number of examples.
- Attention modules tend to focus on areas rather than generating features specific, thus it needs more data to learn what to focus on and what not.
- Building a new loss and what the semantics of it are, is not an easy thing.

4 Discussion and Conclusion

In our experiments, we found Pseudo-labelling and VAT to perform very similar to each other in terms of test error rate performance. Pseudo-Labeling training works well on classification problems by decreasing the density of data points at the decision boundary. According to the stated above assumption of clustering, using the Pseudo-Labeling approach we can get a better generalization performance. Besides its efficient performance, this technique is quite simple in implementation and does not require high computational operations.

Although the idea of using pseudo labels is simple and effective in deep learning, sometimes it might happen that the incorrect pseudo labels will impair the generalization performance and slow down the training of deep networks. Wrong pseudo labels of unlabeled examples (and fake examples) greatly deteriorate the accuracy of prior methods based on pseudo labels, but VAT exhibits a better tolerance to wrong pseudo labels. VAT is computationally less expensive and have only three parameters to tune.

In task 3, we tried to go beyond features and explore the relationships between the areas at which the model should concentrate. However, it turned out that it is difficult to achieve better performance in the given context where the number of labelled samples are low. Also, with lower thresholds, uncertainty in the framework increases. Compared to pseudo-labeling and VAT, it did not outperform them. Though, with CIFAR-10, 4000 labelled samples, we saw lower error rates even with a lesser number of training epochs. This indicates that with a decent amount of labelled samples, training time and hyperparameter tuning, good results can be achieved. Though, we did not thoroughly study the effect of the number of attention maps generated, but we believe that by introducing a distance loss, which is calculated, based on the generated attention maps, to encourage ones that are more diverse, the attention module could be improved for better results.

References

[1] Ekin D Cubuk et al. “Randaugment: Practical automated data augmentation with a reduced search space”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020, pp. 702–703.

[2] Chengyue Gong, Dilin Wang, and Qiang Liu. “AlphaMatch: Improving Consistency for Semi-supervised Learning with Alpha-divergence”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 13683–13692.

[3] Zijian Hu et al. “SimPLE: Similar Pseudo Label Exploitation for Semi-Supervised Classification”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 15099–15108.

[4] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).

[5] Dong-Hyun Lee et al. “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks”. In: Workshop on challenges in representation learning, ICML. Vol. 3. 2. 2013, p. 896.

[6] Takeru Miyato et al. “Virtual adversarial training: a regularization method for supervised and semi-supervised learning”. In: IEEE transactions on pattern analysis and machine intelligence 41.8 (2018), pp. 1979–1993.

[7] Kihyuk Sohn et al. “Fixmatch: Simplifying semi-supervised learning with consistency and confidence”. In: arXiv preprint arXiv:2001.07685 (2020).

[8] Sergey Zagoruyko and Nikos Komodakis. “Wide Residual Networks”. In: BMVC. 2016.

[9] Bowen Zhang et al. “Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling”. In: Advances in Neural Information Processing Systems 34 (2021).

[10] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In ICLR, 2015.

[11] Ranzato, M., and Szummer, M. Semi-supervised learning of compact document representations with deep networks. In Proceedings of the 25th international conference on Machine learning. ACM, 2008. p. 792- 799.

[12] Larochelle, H., and Bengio, Y. Classification using discriminative restricted Boltzmann machines. In Proceedings of the 25th international conference on Machine learning. ACM, 2008. p. 536-543.

[13] Weston, J., Ratle, F., and Collobert, R. Deep learning via semi-supervised embedding. In Proceedings of the 25th international conference on Machine learning. ACM, 2008. p. 1168-1175.

[14] Yves Grandvalet and Yoshua Bengio, Entropy Regularization, In: Semi-Supervised Learning, pages 151–168, MIT Press, 2006.

[15] Diederik Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In NIPS, 2014.

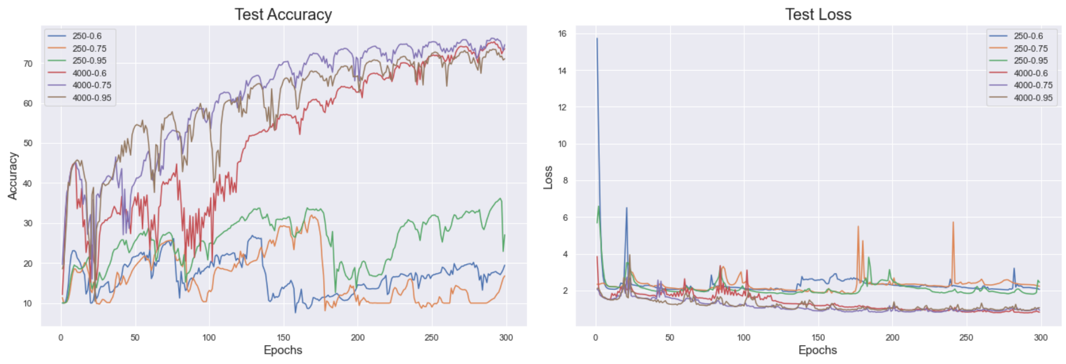
[16] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. In ICML, 2016.

[17] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In NIPS, 2016.

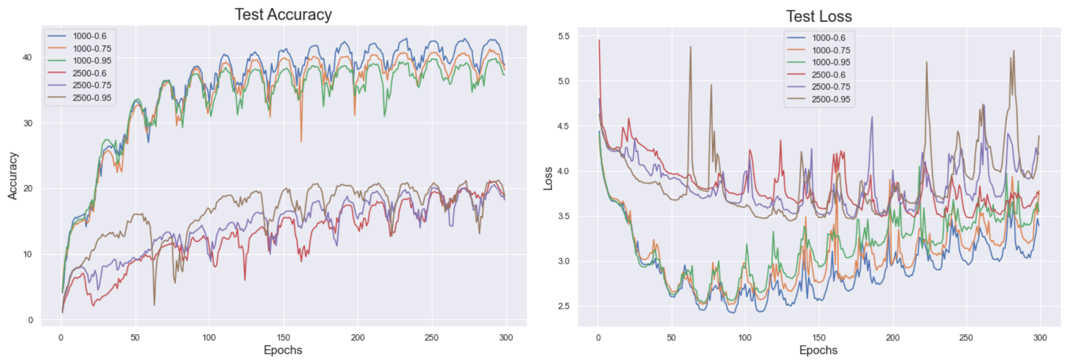
[18] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In ICLR, 2015.

A Appendix

If you need additional space, you may use an appendix after your references but your appendix should contain only figures, result tables and graphs. Your appendix must not contain any substantial text.

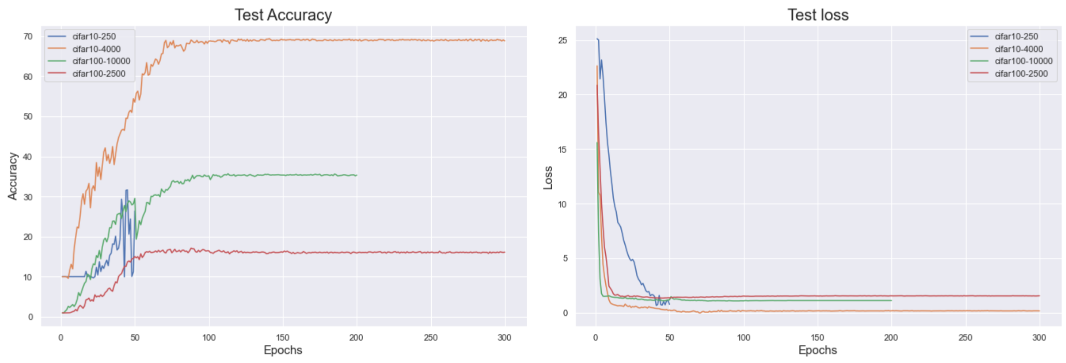


(a)



(b)

Figure 1: Task 1 (Pseudo-labelling) accuracy (left) and loss (right) plots for (a) CIFAR-10 and (b) CIFAR-100 test dataset.



(a)

Figure 2: Task 2 (VAT) tensorboard accuracy (left) and log-loss (right) plots for CIFAR-10 and CIFAR-100 test dataset.

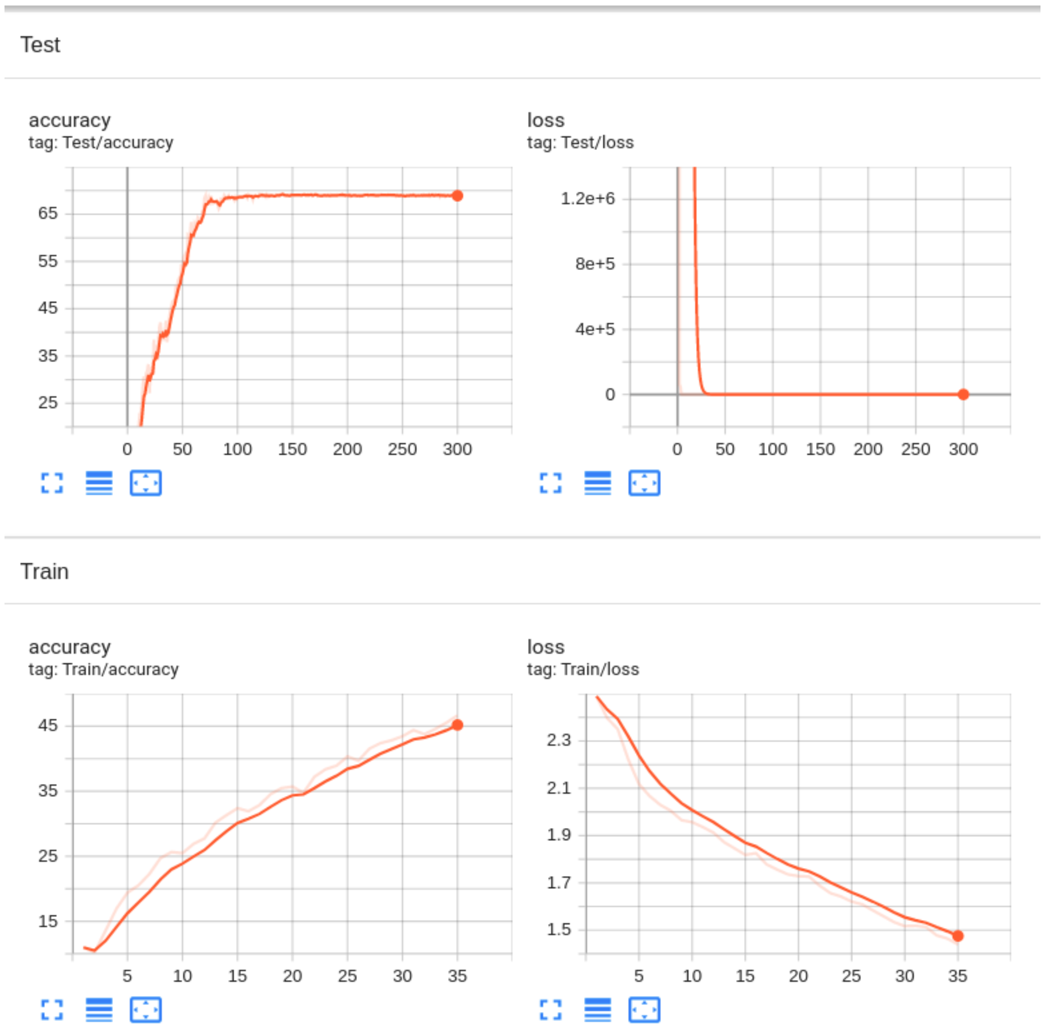


Figure 3: Task 2 (VAT) tensorboard accuracy and loss plots of CIFAR-10 dataset (4000 labels).

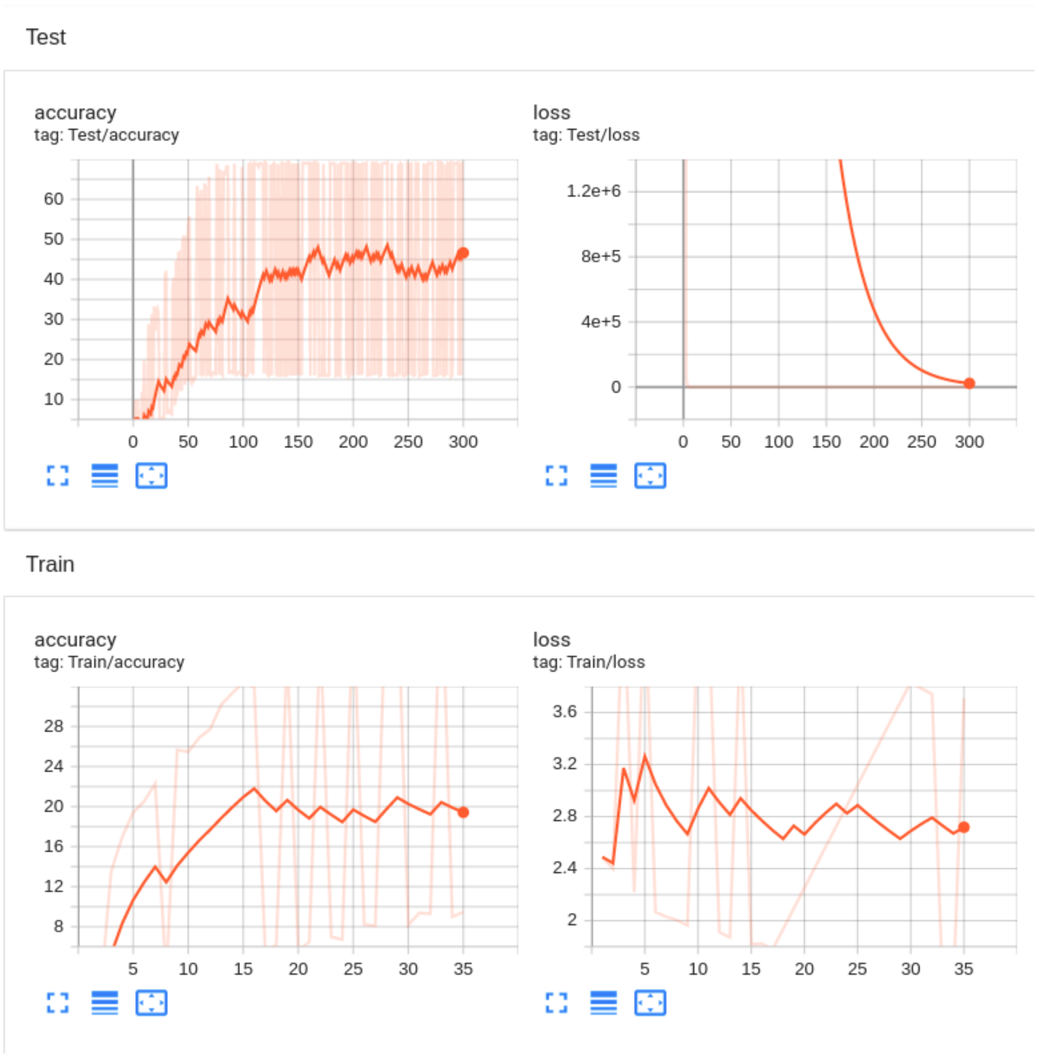
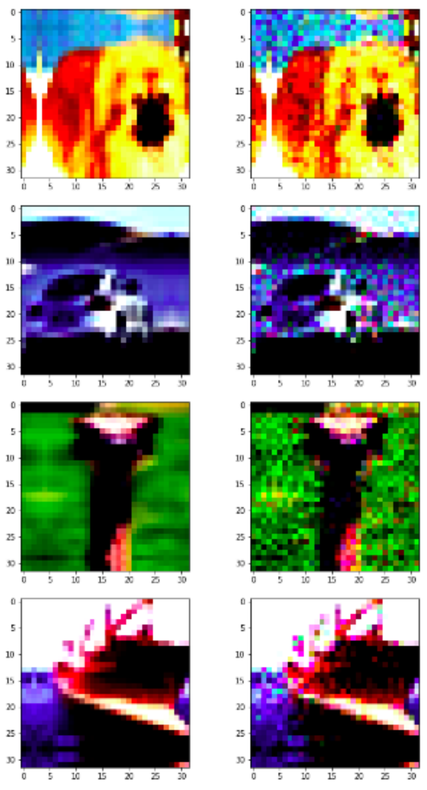


Figure 4: Task 2 (VAT) accuracy and loss plots of CIFAR-100 dataset (2500 labels).


(a) CIFAR-10

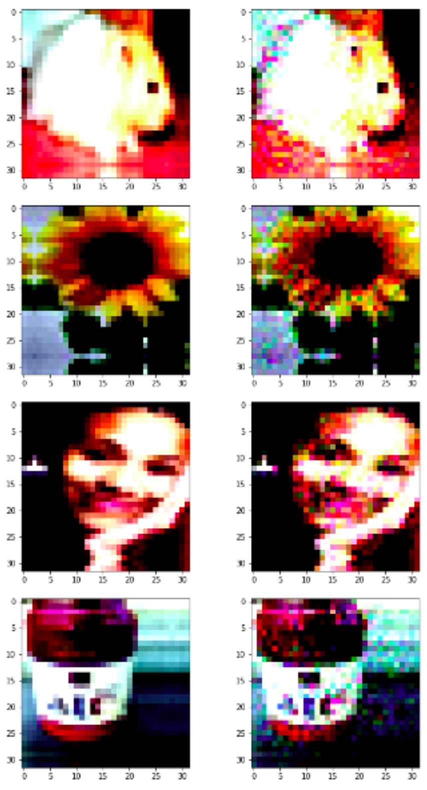

(b) CIFAR-100

Figure 5: For (a) CIFAR-10 and (b) CIFAR-100, original (left image) and perturbed (right image) examples respectively in Task 2 (VAT) with $xi = eps = 10$.