MIXING UP REAL SAMPLES AND ADVERSARIAL SAMPLES FOR SEMI-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Consistency regularization methods have shown great success in semi-supervised learning tasks. Most existing methods focus on either the local neighborhood or in-between neighborhood of training samples to enforce the consistency constraint. In this paper, we propose a novel generalized framework called Adversarial Mixup (AdvMixup), which unifies the local and in-between neighborhood approaches by defining a virtual data distribution along the paths between the training samples and adversarial samples. Experimental results on both synthetic data and benchmark datasets exhibit the benefits of AdvMixup on semi-supervised learning.

1 INTRODUCTION

Deep neural networks have achieved remarkable performance in various areas thanks to their excellent capability on data representation learning. However, successful training of deep learning models usually requires a large amount of labeled data. Such property poses a challenge to many practical tasks where labeling a larget amount of data is not feasible due to the high cost in time and finances. To address this problem, semi-supervised learning leverages the unlabeled data to improve the generalization performance of the model over a small amount of labeled data.

Cluster assumption Chapelle & Zien (2005) has been a basis for many successful semi-supervised learning models, which states that the data distribution forms discrete clusters and samples in the same cluster tend to share the same class label. This assumption has motivated many traditional semi-supervised learning approaches such as transductive support vector machines Joachims (1999), entropy minimization Grandvalet & Bengio (2005), and pseudo-labeling Lee (2013). Recently, the consistency regularization based methods Sajjadi et al. (2016); Laine & Aila (2017); Tarvainen & Valpola (2017); Miyato et al. (2018); Verma et al. (2019) have renewed the state-of-the-art results across many semi-supervised learning tasks. Basically, consistency regularization enforces the predictions of an unlabeled sample x and its neighborhood sample $\hat{x}$ to be the same, which encourages the decision boundary to lie in the low-density regions. Different methods concentrate on different types of the neighborhood samples $\hat{x}$.

One branch of the consistency regularization methods focuses on the local neighborhood around the training samples. The Π model Laine & Aila (2017) obtained $\hat{x}$ by adding a random noise to x. However, Szegedy et al. (2014); Goodfellow et al. (2015) have shown that models regularized with such isotropic noise can still be vulnerable to the perturbations in the adversarial direction Goodfellow et al. (2015). Inspired by this, Miyato et al. (2018) proposed the Virtual Adversarial Training (VAT) model, where $\hat{x}$ is selected as the adversarial example of x, thus regularizing the model in the most non-smooth regions. These perturbation-based methods can be visualized as in Figure 1a, where the possible areas for the selection of $\hat{x}$ are centred around the training samples.

Another branch of the consistency regularization methods considers the in-between neighborhood of two training samples. The mixup model Zhang et al. (2018) picked $\hat{x}$ along the interpolation path between pairs of training samples x_i and x_j, i.e., $\hat{x} = \lambda x_i + (1-\lambda) x_j$, and enforced a linear transition along this path by requiring $f(\hat{x})$ to approximate $\lambda y_i + (1-\lambda) y_j$, which is originally proposed for supervised learning. Verma et al. (2019) generalized the mixup model to semi-supervised learning by replacing the real labels with the predicted labels by a teacher model Tarvainen & Valpola (2017). These interpolation-based methods can be visualized as in Figure 1b, where the possible areas for the selection of $\hat{x}$ are along the path between pairs of training samples.

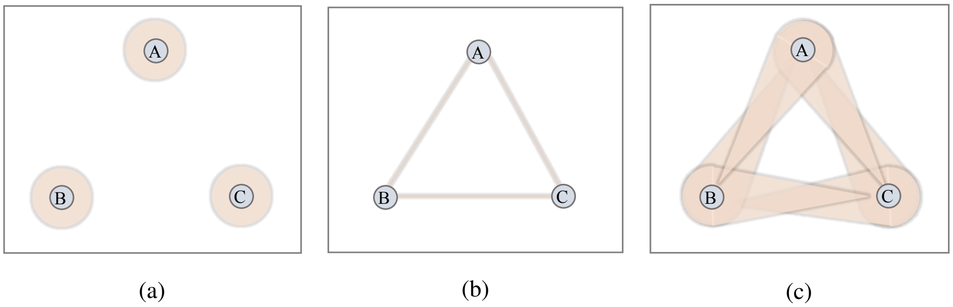


Figure 1: Visualization of the consistency regularization areas for different methods. (a) Perturbation based methods. (b) Interpolation based methods. (c) Our AdvMixup.

In this paper, we propose a novel consistency regularization technique, called Adversarial Mixup (AdvMixup), by unifying both the local neighborhood and in-between neighborhood. In particular, we define the neighborhood as the samples lying along the paths between the real samples and adversarial samples. For two random training samples x_i and x_j, we sample $\hat{x}$ along the interpolation path between x_i and the adversarial sample $x_j^{(adv)}$, i.e., $\hat{x} = \lambda x_i + (1-\lambda)x_j^{(adv)}$. Then we enforce the consistency between $f(\hat{x})$ and $\lambda f(x_i) + (1-\lambda)f(x_j)$. Note that we use the predicted label $f(x_j)$ for $x_j^{(adv)}$, which implicitly incorporates the local neighborhood regularization method. The regularization area for our AdvMixup can be visualized as in Figure 1c.

We evaluate our AdvMixup on one synthetic dataset and several commonly used benchmark datasets, and the experimental results demonstrate AdvMixup outperforms the baseline methods which consider only local neighborhood or in-between neighborhood, especially when few labeled data is given.

2 AdvMixup

2.1 Problem Definition

In this paper, we focus on the standard semi-supervised learning task. Formally, Let $\mathcal{X}$ denote the input feature space and $\mathcal{Y}$ denote target label space. Given a labeled dataset $\mathcal{S}_l = \{(x_i, y_i)|i = 1, \ldots, N_l\}$ and an unlabeled dataset $\mathcal{S}_u = \{x_i|i = 1, \ldots, N_u\}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, our aim is to learn a mapping function $f : \mathcal{X} \to \mathcal{Y}$ which can generalize to the unseen (x_i, y_i) data pairs sampled from the joint probability distribution $P(\mathcal{X}, \mathcal{Y})$.

2.2 AdvMixup

Standing on the cluster assumption Chapelle & Zien (2005), we propose Adversarial Mixup (AdvMixup), a new consistency regularization approach for semi-supervised learning. AdvMixup implicitly defines a virtual data distribution $\hat{P}$ sampling along the interpolation paths between pairs of points from the real samples and the adversarial samples.

Formally, given a random pair of unlabeled training samples x_i and x_j, we first craft an adversarial sample $x_j^{(adv)}$ for x_j, then construct a virtual data sample $(\hat{x}_{i,j}, \hat{y}_{i,j})$ using the interpolation between x_i and $x_j^{(adv)}$ as the virtual input and the interpolation between the soft labels of x_i and x_j as the virtual target:

$$\begin{aligned} \hat{x}_{i,j} &= \lambda x_i + (1-\lambda)x_j^{(adv)}, \\ \hat{y}_{i,j} &= \lambda f_t(x_i) + (1-\lambda)f_t(x_j), \end{aligned} \tag{1}$$

where $\lambda \in [0, 1]$ is sampled from the distribution $P_\lambda = Beta(\alpha, \alpha)$ with $\alpha \in [0, \infty]$. Following the ICT model Verma et al. (2019), we employ the predictions from the exponential moving average (EMA) model f_t as the soft labels for better target quality Tarvainen & Valpola (2017).

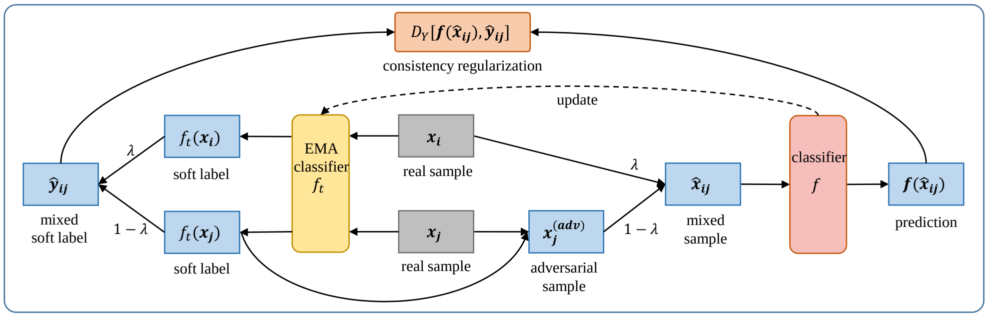


Figure 2: Overview of the proposed AdvMixup framework.

The goal of AdvMixup is to fit the contructed virtual data samples by minimizing the divergence between the model prediction on the virtual input $f(\hat{x}_{i,j})$ and the virtual target $\hat{y}_{i,j}$, which can be formulated as

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{x_i, x_j \sim \mathcal{S}_u} \big[D_{\mathcal{Y}}[f(\hat{x}_{i,j}), \hat{y}_{i,j}] \big], \tag{2}$$

where $D_{\mathcal{Y}}$ is a divergence metric defined on the $\mathcal{Y}$ space. The overview of our AdvMixup regularization is shown in Figure 2.

Finally, we arrive at the full objective function for AdvMixup, i.e., minimizing

$$\mathcal{L}_{\text{nll}} + \beta \mathcal{L}_{\text{reg}} \tag{3}$$

where $\mathcal{L}_{\text{nll}} = \mathbb{E}_{(x_i, y_i) \sim \mathcal{S}_l} \big[-y_i^\top \ln f(x_i) \big]$ is the typical negative log-likelihood loss for the labeded data, and β is a hyper-parameter controlling the importance of regularization term $\mathcal{L}_{\text{reg}}$. The training procedure of our model is illustrated in Algorithm 1.

Adversarial Sample Generation. An adversarial sample Szegedy et al. (2014); Goodfellow et al. (2015) is an slightly and carefully perturbed variant of a real data sample, with the aim of misleading a given classifier to make different predictions from the original real data sample. In this paper, we adopt the virtual adversarial example generation method from Miyato et al. (2018), where the "virtual" means no ground-truth target labels are used to cater for the semi-supervised setting. Specifically, we craft an adversarial sample $x_j^{(adv)} = x_j + r_j^{(adv)}$ for x_j by optimizing

$$r_j^{(adv)} = \operatorname*{arg\,max}_{\|r\|_2 \le \epsilon} D_{\mathcal{Y}} \big[f(x_j), f(x_j + r) \big] \tag{4}$$

where $\epsilon > 0$ is the norm constraint for the adversarial perturbation. The maximization problem can be approximated by the power iteration method. In practice, one step of iteration is enough to achieve strong performance Miyato et al. (2018), which requires a low additional computational cost to the basic mixup model.

Generality. The proposed AdvMixup can generalize to both the perturbation-based regularization (e.g., VAT Miyato et al. (2018)) and the mixup-based regularization (e.g., ICT Verma et al. (2019)). If $\lambda \to 0$, the constructed virtual data sample is $(x_j^{(adv)}, f_t(x_j))$ in Equation 1, reducing to the VAT model. If the adversarial perturbation degenerate to zero, i.e., $x_j^{(adv)} = x_j$, Equation 1 reduces to the ICT model.

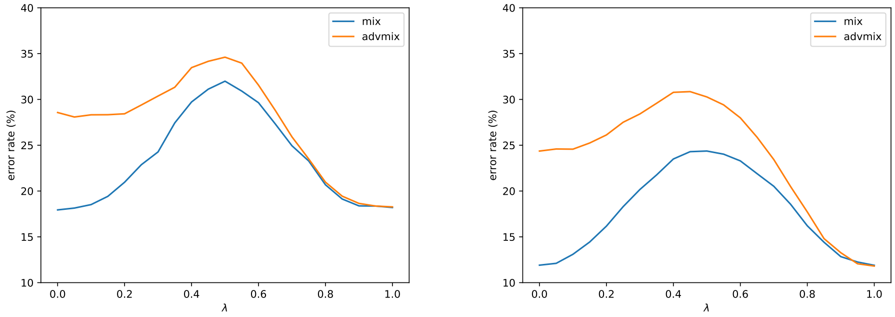
3 Why AdvMixup?

The AdvMixup model regularizes the classifier f along the interpolation paths between training samples and adversarial samples. In the following, we elaborate the reasonableness and advantages of this regularization scheme, and validate the effectiveness of AdvMixup via a case study on synthetic data.

Reasonableness. The mixup model Zhang et al. (2018) and ICT model Verma et al. (2019) encourage the classifier to have linear transition in-between real samples, thus pushing the decision

Algorithm 1 Minibatch training of AdvMixup for semi-supervised learning

```
▷ Require: labeled training set $\mathcal{S}_l$; unlabeled training set $\mathcal{S}_u$
▷   classification model $f$ with parameters $\theta$; $f$'s EMA version $f_t$ with parameters $\theta_t$
▷   Beta distribution parameter $\alpha$;weight of regularization term $\beta$; $f_t$'s update ratio $\gamma$
▷ for $k = 1, \dots,$ num_iterations do
▷   Sample a labeled batch $B_l = \{(x_i, y_i)\}_{i=1}^{n_l} \sim \mathcal{S}_l$
▷   Sample an unlabeled batch $B_u = \{x_i\}_{i=1}^{n_u} \sim \mathcal{S}_u$
▷   Compute the negative log-likelihood loss using $B_l$: $\mathcal{L}_{\text{nll}} = \frac{1}{n_l} \sum_{(x_i,y_i) \in B_l} \left[-y_i^\top \ln f(x_i)\right]$
▷   Associate the samples in $B_u$ with soft labels $B_{u+} = \{(x_i, f_t(x_i))\}_{i=1}^{n_u}$
▷   Craft an adversarial batch $B_{u+}^{(adv)} = \{(x_i^{(adv)} = x_i + r_i^{(adv)}, f_t(x_i)) | x_i = B_u[i]\}_{i=1}^{n_u}$ using Eq. 4
▷   Shuffle $B_{u+}^{(adv)}$ as $B_{u+,s}^{(adv)}$
▷   Sample $\lambda \sim Beta(\alpha, \alpha)$ as the interpolation parameter
▷   Construct a virtual data batch $\hat{B}_{u+} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{n_u}$ with
                     $\hat{x}_i = \lambda x_i^1 + (1 - \lambda) x_i^2,$
                     $\hat{y}_i = \lambda y_i^1 + (1 - \lambda) y_i^2$
    where $(x_i^1, y_i^1) = B_{u+}[i], (x_i^2, y_i^2) = B_{u+,s}^{(adv)}[i]$
▷   Compute the consistency regularization term $\mathcal{L}_{\text{reg}} = \frac{1}{n_u} \sum_{(\hat{x}_i,\hat{y}_i) \in \hat{B}_{u+}} D_{\mathcal{Y}}[f(\hat{x}_i), \hat{y}_i]$
▷   Evaluate the full objective function $\mathcal{L} = \mathcal{L}_{\text{nll}} + \beta \mathcal{L}_{\text{reg}}$
▷   Update $\theta$ based on the gradient $\nabla_\theta \mathcal{L}$
▷   Update $\theta_t = (1 - \gamma)\theta_t + \gamma\theta$
▷ end for
▷ Output: $\theta$ and $\theta_t$
```



(a) CIFAR-10

(b) SVHN

Figure 3: Prediction error rates of the model (trained with only labeled data) on the virtual samples along the interpolation paths (blue line) defined by the ICT model Verma et al. (2019) and the interpolation paths (orange line) defined by the proposed AdvMixup. (a) Results on the CIFAR-10 dataset where 4000 labeled data samples are used. (b) Results on the SVHN dataset where 1000 labeled data samples are used. Best viewed in color.

boundary to low-density areas. Our AdvMixup takes one additional step by creating an adversarial sample for one of the real sample pair. The created adversarial sample is supposed to share the same class label with its corresponding real sample. Therefore, given a random real sample pair x_i and x_j as well as the adversarial sample $x_j^{(adv)}$ for x_j, it is reasonable to enforce the classifier's predictions to linearly change from the (soft) target label $f(x_i)$ of x_i to the (soft) target label $f(x_j)$ of $x_j^{(adv)}$ along the path from x_i to $x_j^{(adv)}$.

Advantages. Consistency regularization approaches are actually fixing possible flaws of the classifier which violates the cluster assumption. An effective approach is expected to detect the flaws violating the cluster assumption 1) more significantly and 2) more comprehensively. Compared with the methods seeking for the flaws in-between neighborhood of training samples like ICT Verma et al. (2019), our AdvMixup considers the adversarial samples that violate the cluster assumption more

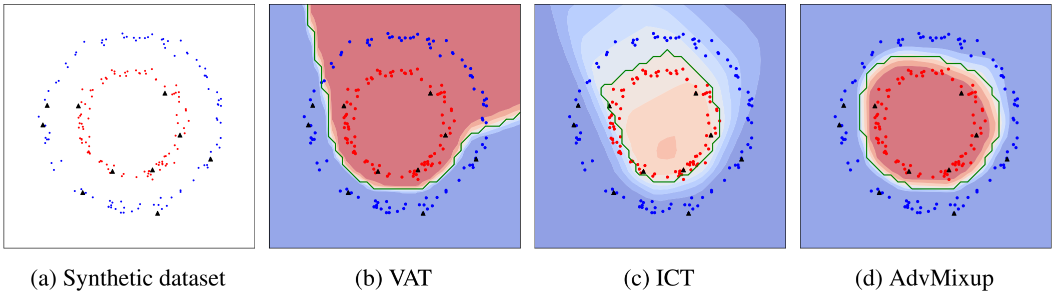

(a) Synthetic dataset (b) VAT (c) ICT (d) AdvMixup

Figure 4: Comparison between VAT, ICT, and proposed AdvMixup on the concentric circles dataset. Red and blue points denote unlabeled samples from the two classes, i.e., inner circle and outter circle, and labeled data are marked with black triangles. (a) The concentric circles dataset. (b-d) The contour plot and the decision boundaries (green curve) learned by VAT (b), ICT (c), and AdvMixup (d). Best viewed in color.

significantly. To verify this, we first train a supervised model without using any regularization techniques on the CIFAR-10 and SVHN datasets, and then use it to predict the virtual samples defined by ICT and AdvMixup. As shown in Figure 3, the supervised model exhibits much larger error rates along the real-adversarial interpolation path of AdvMixup than the real-real interpolation path of ICT. Compared with the methods seeking for flaws in the local neighborhood of training samples like VAT Miyato et al. (2018), our AdvMixup explores a more comprehensive searching area. In particular, AdvMixup incorporates the local neighborhood based regularization as special cases when $\lambda \to 0$, while allowing regularization on in-between areas when $\lambda > 0$. The usefulness of such in-between areas has been validated by Zhang et al. (2018); Verma et al. (2019) and also illustrated in Figure 3 where the error rate reaches the maximum around $\lambda = 0.5$.

3.1 Case Study on Synthetic Data

We evaluate the proposed AdvMixup against VAT and ICT on a synthetic dataset with two classes. As shown in Figure 4a, the training points forms two concentric circles with different radiuses, and the Gaussian noise ($\mu = 0$, $\sigma = 0.01$) is applied to these points. The task is to classify these two classes of points, and each class contains 5 labeled samples and 100 unlabeled samples. Note that sometimes the distance between neighbor points within the same class can be comparable, if not larger than, the distance between neighbor points from different classes, making the problem a non-trivial task.

We utilize a neural network model as the classifier, which includes two hidden layers with 100 and 50 hidden units and ReLU activation functions. We fix the weight of the regularization terms as 10 for different methods, and search the optimal hyper-parameters specific to different methods (i.e., ϵ for VAT, α for ICT, and ϵ and α for AdvMixup) via a validation set.

The learned decision boundaries are shown in Figure 4, and we have the following major observations. First, VAT can not successfully classify the two classes, since it mistakenly predicts a proportion of blue points as red points. Since these blue points have a relatively larger distance to other surrounding blue points and no labeled data lie in this region, it is possible for VAT to fail as the result has almost no objections to their local neighborhood based constraint. However, with the local neighborhood based constraint, the decision boundary of VAT desirably keeps a safe distance with the training samples. Second, ICT basically achieves to distinguish the two classes. However, the decision boundary stays too close to some data points, making the model vulnerate to even small noises. This is possible for ICT, because the points next to the decision boundary lie in a lighter area of the contour and thus have lower confidence scores, making the result still comply with the constraint for in-between training samples. Third, the proposed AdvMixup, considering both local and in-between neighborhood, is capable of learning a decision boundary which can both differentiate points from the two classes and stay a certain distance with the training samples.

4 Experiments

In this section, we evaluate the proposed AdvMixup against various strong baselines for semi-supervised learning on benchmark datasets. We also conduct an ablation study to validate the effectiveness of our model.

4.1 Datasets

We conduct experiments on the widely-used CIFAR-10 and SVHN datasets. The CIFAR-10 dataset is composed of 32×32 colored images drawn from 10 natural classes, with a split of 50,000 training samples and 10,000 test samples. The SVHN dataset is composed of 32×32 colored images drawn from 10 digit classes, with a split of 73,257 training samples and 26,032 test samples. Following common practice Sajjadi et al. (2016); Laine & Aila (2017); Tarvainen & Valpola (2017); Miyato et al. (2018); Verma et al. (2019); Oliver et al. (2018), we randomly select a small ratio of training samples as labeled data and use the rest as unlabeled data for semi-supervised learning. In particular, we provide results with 1000, 2000, and 4000 labeled samples on the CIFAR-10 dataset, and 250, 500, and 1000 labeled samples on the SVHN dataset. The hyper-parameters are tuned on a validation set with 5000 (CIFAR-10) and 1000 (SVHN) samples.

4.2 Implementation Details

Data preprocessing. Following our baselines, we adopt standard data augmentation and data normalization in the preprocessing phase. On the CIFAR-10 dataset, we first augment the training data by random horizontal flipping and random translation (in the range of [-2,2] pixels), and then apply global contrast normalization and ZCA normalization based on statistics of all training samples. On the SVHN dataset, we first augment the training data by random translation (in the range of [-2,2] pixels), and then apply zero-mean and unit-variance normalization.

Model architecture. We adopt the exactly same 13-layer convoluation neural network architecture as in the ICT model Verma et al. (2019), which eliminates the dropout layers compared to the variant in Sajjadi et al. (2016); Laine & Aila (2017); Tarvainen & Valpola (2017); Miyato et al. (2018); Luo et al. (2018).

Hyper-parameters. We directly use the norm constraint ϵ in the code[1] of the VAT model Miyato et al. (2018), 8.0 for the CIFAR-10 dataset and 3.5 for the SVHN dataset. The update ratio γ of the EMA model is set to 0.001 following Verma et al. (2019).We search the optimal beta distribution parameter α and the weight β of the regularization term in Equation 3 via the validation performance. As a result, α is set as 2.0 on CIFAR-10 and 0.1 on SVHN; β is set as 50, 100, and 100 for CIFAR-10 with 1000, 2000, and 4000 labeled samples, and as 50, 100, and 100 for SVHN with 250, 500, and 1000 labeled samples.

Model training. We adopt the mean squared error as the divergence metric in Equation 2 and 4 as in Verma et al. (2019). The batch size is 32 for labeled data and 128 for unlabeled data. We follow the settings of Verma et al. (2019) for other details: The model is trained for 400 epochs, and optimized using the SGD algorithm with a momentum factor 0.9 and weight decay factor 1×10^{-4}; the learning rate is set to 0.1 initially and then decayed using the cosine annealing strategyLoshchilov & Hutter (2017); a sigmoid warm-up schedule is utilized to increase the regularization weight β from 0 to its maximum value within the first 100 epochs. Our code will be made publicly available soon.

4.3 Results

The evaluation results of the proposed AdvMixup against several state-of-the-art methods on CIFAR-10 and SVHN are shown in Table 1 and Table 2, respectively. The baseline semi-supervised learning methods encompass consistency regularization methods based on local neighborhoods (Laine & Aila (2017); Tarvainen & Valpola (2017); Miyato et al. (2018); Park et al. (2018); Athiwaratkun et al. (2019); Clark et al. (2018)) , in-between neighborhoods (Verma et al. (2019)), as well as those combining them (Luo et al. (2018)). From Table 1 and Table 2, we have the following observations.

[1] `https://github.com/takerum/vat_tf`

Table 1: Test error rates (%) of different methods on CIFAR-10. Random horizontal flipping and random translation are used to augment training data. Results for methods in the first block (i.e., Supervised, Supervised (Mixup), and Supervised (Manifold Mixup)) are duplicated from Verma et al. (2019). Results of AdvMixup are averaged over 3 runs.

Method	Test error rates (%)		
	1000 labels	2000 labels	4000 labels
Supervised	39.95 ± 0.75	31.16 ± 0.66	21.75 ± 0.46
Supervised (Mixup)	36.48 ± 0.15	26.24 ± 0.46	19.67 ± 0.16
Supervised (Manifold Mixup)	34.58 ± 0.37	25.12 ± 0.52	18.59 ± 0.18
Π model (Laine & Aila, 2017)	31.65 ± 1.20	17.57 ± 0.44	12.36 ± 0.31
TempEns (Laine & Aila, 2017)	23.31 ± 1.01	15.64 ± 0.39	12.16 ± 0.24
MT (Tarvainen & Valpola, 2017)	21.55 ± 1.48	15.73 ± 0.31	12.31 ± 0.28
VAT (Miyato et al., 2018)	–	–	11.36 ± 0.34
VAT+EntMin (Miyato et al., 2018)	–	–	10.55 ± 0.05
VAdD (Park et al., 2018)	–	–	11.32 ± 0.11
VAdD + VAT (Park et al., 2018)	–	–	9.22 ± 0.10
TempEns+SNTG (Luo et al., 2018)	18.41 ± 0.52	13.64 ± 0.32	10.93 ± 0.14
VAT+EntMin+SNTG (Luo et al., 2018)	–	–	9.89 ± 0.34
CT-GAN (Wei et al., 2018)	–	–	9.98 ± 0.21
CVT (Clark et al., 2018)	–	–	10.11 ± 0.15
MT+ fast-SWA (Athiwaratkun et al., 2019)	15.58 ± 0.12	11.02 ± 0.23	9.05 ± 0.21
ICT (Verma et al., 2019)	15.48 ± 0.78	9.26 ± 0.09	7.29 ± 0.02
AdvMixup	$\mathbf{9.67 \pm 0.08}$	$\mathbf{8.04 \pm 0.12}$	$\mathbf{7.13 \pm 0.08}$

Table 2: Test error rates (%) of different methods on SVHN. Random translation is used to augment training data. Results for methods in the first block (i.e., Supervised, Supervised (Mixup), and Supervised (Manifold Mixup)) are duplicated from Verma et al. (2019). Results for AdvMixup are averaged over 3 runs.

Method	Test error rates (%)		
	250 labels	500 labels	1000 labels
Supervised	40.62 ± 0.95	22.93 ± 0.67	15.54 ± 0.61
Supervised (Mixup)	33.73 ± 1.79	21.08 ± 0.61	13.70 ± 0.47
Supervised (Manifold Mixup)	31.75 ± 1.39	20.57 ± 0.63	13.07 ± 0.53
Π model (Laine & Aila, 2017)	9.93 ± 1.15	6.65 ± 0.53	4.82 ± 0.17
TempEns (Laine & Aila, 2017)	12.62 ± 2.91	5.12 ± 0.13	4.42 ± 0.16
MT (Tarvainen & Valpola, 2017)	4.35 ± 0.50	4.18 ± 0.27	3.95 ± 0.19
VAT (Miyato et al., 2018)	–	–	5.42 ± 0.22
VAT+EntMin (Miyato et al., 2018)	–	–	3.86 ± 0.11
VAdD (Park et al., 2018)	–	–	4.16 ± 0.08
VAdD + VAT (Park et al., 2018)	–	–	3.55 ± 0.05
Π+SNTG (Luo et al., 2018)	5.07 ± 0.25	4.52 ± 0.30	3.82 ± 0.25
MT+SNTG (Luo et al., 2018)	4.29 ± 0.23	3.99 ± 0.24	3.86 ± 0.27
ICT (Verma et al., 2019)	4.78 ± 0.68	4.23 ± 0.15	3.89 ± 0.04
AdvMixup	$\mathbf{3.95 \pm 0.70}$	$\mathbf{3.37 \pm 0.09}$	$\mathbf{3.07 \pm 0.18}$

Firstly, for CIFAR-10, AdvMixup outperforms all the baselines across different numbers of labeled data. In particular, AdvMixup improves the second-best method ICT by nearly 6% when only 1000 labeled samples are given.

Secondly, for SVHN, it is much easier than the task on CIFAR-10 as the house number images of SVHN has smaller variance compared to the natural images of CIFAR-10, and the baselines already achieve a high accuracy. Nevertheless, AdvMixup still demonstrates a clear improvement over all

Table 3: Test error rates (%) of different ablated versions on CIFAR-10. Random horizontal flipping and random translation are used to augment training data. Results are averaged over 3 runs.

Method	Test error rates (%)		
	1000 labels	2000 labels	4000 labels
AdvMixup	$\mathbf{9.67 \pm 0.08}$	$\mathbf{8.04 \pm 0.12}$	$\mathbf{7.13 \pm 0.08}$
ICT + VAT	9.91 ± 0.25	8.90 ± 0.20	7.97 ± 0.03
Adv-Adv Mixup	10.90 ± 0.11	8.99 ± 0.14	8.22 ± 0.09
AdvMixup w/o teacher model	11.64 ± 0.41	9.78 ± 0.11	8.20 ± 0.17

the baselines across different numbers of labeled data. In particular, AdvMixup achieves an error rate of 3.95% for 250 labels, which beats the results of 500 labels of all baselines.

Thirdly, following Verma et al. (2019), we also compare with the supervised methods (methods in the first block of Table 1 and Table 2), where only the labeled samples are used. For both CIFAR-10 and SVHN, AdvMixup exhibits significant improvement over the supervised baselines across different numbers of labeled data.

4.4 Ablation Study

To provide more insights, we present the performance of the following three variants of our model on the CIFAR-10 dataset:

- **ICT+VAT**, which is an alternative of integrating the local neighborhood and in-between neighborhood approaches by simply combining the losses of the ICT model and the VAT model.
- **Adv-Adv Mixup**, which is another alternative of integrating the local neighborhood and in-between neighborhood approaches by defining the interpolation paths between two adversarial samples, i.e., replacing x_i with $x_i^{(adv)}$ in Equation 1.
- **AdvMixup w/o teacher model**, which use the prediction of current model instead of the EMA model to compute the soft labels for the samples, i.e., replacing $f_t(x_i)$ and $f_t(x_j)$ with $f(x_i)$ and $f(x_j)$ in Equation 1.

The results of these ablated variants are shown in Table 3. Firstly, ICT+VAT shows a major improvement over ICT with 4000 labels, but underperforms AdvMixup across different numbers of labeled samples. We conjecture the reason is: by enforcing consistency constraint in local neighborhood and the interpolation paths between training samples, ICT+VAT stills have no guarantee on the more difficult interpolation paths between adversarial samples and training samples. Secondly, interpolating between adversarial examples clearly degrade the performance across different numbers of labeled samples. Our explanation is that there can be a gap between the true data distribution and the virtual data distribution defined by this interpolation scheme where real samples are not utilized, thus increasing the prediction errors on the test samples lying in the true data distribution. Thirdly, eliminating the teacher model clearly degrade the performance by 1%-2% across different numbers of labeled samples. However, this difference resulted from the teacher model is smaller than the difference between ICT and ICT w/o teacher model, which is about 4% as reported in Verma et al. (2019).

5 Related Work

Semi-supervised learning has been a hot topic for a long time to address the problem of limited labeled data which hinders the learning based models. By leveraging unlabeled data, semi-supervised learning methods are dedicated to designing a regularization term to encourage the model to comply with the cluster assumption Chapelle & Zien (2005), which favors decision boundaries lying in low-density regions and stable model behaviors without abrupt changes. In the following, we mainly concentrate on the consistency regularization methods which represent the state-of-the-art and are mostly related to our work.

An important research line in consistency regularization constrains the model to have consistent predictions around local neighborhood around training inputs, where the local neighborhood is usually represented as variants of the input or model parameters. The Π model from Sajjadi et al. (2016) and Laine & Aila (2017) constructed different input variants with stochastic image transformation and additive Gaussian noise, as well as different model variants with dropout layers. Wei et al. (2018) integrated the Π model with the generative adversarial networks (GAN) based semi-supervised learning approaches Salimans et al. (2016), where the classifier was forced to correctly classify labeled samples and distinguish real unlabeled samples and fake samples from the generator. Clark et al. (2018) proposed cross-view learning which formed input variants by randomly masking partial inputs. Laine & Aila (2017) also proposed a Temporal Ensembling approach to apply the consistency constraint between current model prediction and the EMA of all historical predictions for a specific input. Tarvainen & Valpola (2017) further improved Temporal Ensembling by considering the consistency between the variants of model parameters, i.e., predictions from current parameters and from the EMA of parameters. Considering the insufficient power of the isotropic perturbations, Miyato et al. (2018) proposed the VAT model by using adversarial perturbations, which point out the model's most vulnerable direction, to better represent the local neighborhood.

Another promising research line in consistency regularization considers the consistency between pairs of training samples. Luo et al. (2018) enhanced the local neighborhood based methods by narrowing down the distance between similar sample pairs while pushing the dissimilar pairs away in the low-dimensional feature space. Under supervised setting, Zhang et al. (2018) proposed the mixup model encouraging the prediction on the linear combination of two samples to approach the linear combination of their labels. The mixup model has been extended from different perspectives owing to its efficiency and strong regularization ability. Verma et al. (2018) extended the mixup operation to the hidden layers for flattening representations. Guo et al. (2019) proposed to adaptively generate the mixing parameter for a specific pair, so as to avoid overlapping between the mixed samples and the real ones. Verma et al. (2019) generalized the mixup model to the semi-supervised setting where the labels are substituted by the soft labels from a teacher model. The concurrent work MixMatch Berthelot et al. (2019) generalized the mixup mechanism with several techniques such as multiple data augmentation and label sharpening, obtaining strong empirical results on semi-supervised learning.

6 Conclusion

In this paper, we propose a new consistency regularization method, AdvMixup, for semi-supervised learning. AdvMixup enforces the model to fit the virtual data points sampled from the interpolation paths between adversarial samples and real samples. Such an interpolation scheme integrates the local neighborhood around training samples and the neighborhood in-between the training samples. Our experiments demonstrate the proposed AdvMixup constantly outperforms the baselines, especially when the labeled data are scarce. In future work, we plan to explore AdvMixup with different adversarial sample generation approaches. Also, it is promising to fit AdvMixup into the recent MixMatch framework Berthelot et al. (2019) for further performance improvement.

References

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *AISTATS*, volume 2005, pp. 57–64. Citeseer, 2005.

Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pp. 529–536, 2005.

Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019.

Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Icml*, volume 99, pp. 200–209, 1999.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, pp. 2, 2013.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8896–8905, 2018.

Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 3235–3246, 2018.

Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2226–2234, 2016.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pp. 1195–1204, 2017.

Vikas Verma, Alex Lamb, Christopher Beckham, Aaron Courville, Ioannis Mitliagkis, and Yoshua Bengio. Manifold mixup: Encouraging meaningful on-manifold interpolation as a regularizer. *arXiv preprint arXiv:1806.05236*, 2018.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. Improving the improved training of wasserstein gans: A consistency term and its dual effect. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.