Mixing Up Real Samples and Adversarial Samples for Semi-Supervised Learning

Yun Ma*
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
mayun371@gmail.com

Xudong Mao*
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
xudong.xdmao@gmail.com

Yangbin Chen
Department of Computer Science
City University of Hong Kong
Hong Kong, China
robinchen2-c@my.cityu.edu.hk

Qing Li
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
qing-prof.li@polyu.edu.hk

***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 that our AdvMixup can achieve better performance and robustness than state-of-the-art methods for semi-supervised learning.**

***Index Terms*—semi-supervised learning, adversarial samples, mixup**

I. 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, in that labeling a large amount of data is not feasible due to the high cost in time and finances. To address this problem, semi-supervised learning (SSL) relieves the demand for labeled data and improves the generalization performance of the model by using more easily-obtained unlabeled data.

Cluster assumption [1] 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 [2], entropy minimization [3], and pseudo-labeling [4]. Recently, the consistency regularization based methods [5]–[9] have renewed the state-of-the-art results across many semi-supervised learning tasks. Consistency regularization enforces the predictions of an unlabeled sample x and its neighborhood sample $\hat{x}$ to be consistent, thus encouraging the decision boundary to lie in low-density regions. Different methods concentrate on different types of neighborhood samples $\hat{x}$.

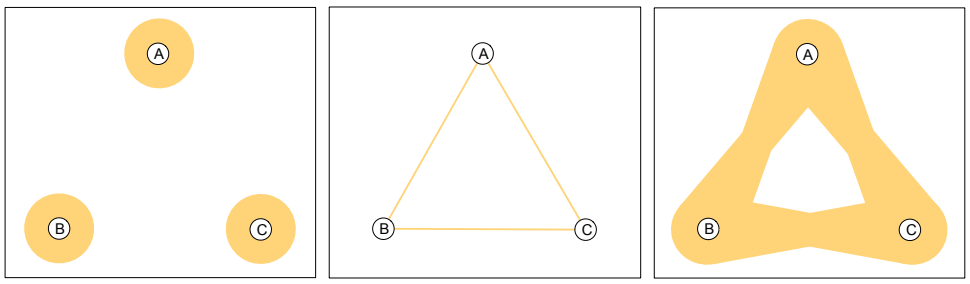
One branch of the consistency regularization methods focuses on the local neighborhood around the training samples. The Π model [6] obtained $\hat{x}$ by adding a random noise to x. However, models regularized with such isotropic noise have shown vulnerabilities to the perturbations in the adversarial direction [10], [11]. Inspired by this, Miyato et al. [8] 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 Fig. 1a, where the possible areas for the selection of $\hat{x}$ are centered around the training samples.

Another branch of the consistency regularization methods considers the in-between neighborhood of two training samples. The mixup model [12], proposed for supervised learning, picked $\hat{x}$ along the interpolation path between a pair 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 the interpolation between their ground-truth labels $\lambda y_i + (1-\lambda)y_j$. The Interpolation Consistency Training (ICT) model [9] generalized the mixup model to semi-supervised learning by replacing the ground-truth labels with the predicted labels of a teacher model [7]. These interpolation-based methods can be visualized as in Fig. 1b, where the possible areas for the selection of $\hat{x}$ are along the paths between pairs of training samples.

Both of the two branches are limited in terms of their consistency regularization areas. On one hand, the perturbation based methods pay attention to the neighborhood of single data points while ignoring the space in-between them, leaving the model unpredictable in these regions. On the other hand, the interpolation-based methods only consider the convex

*The first two authors contributed equally to this work.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) (b) (c)

Fig. 1: Visualization of the consistency regularization areas for (a) perturbation based methods, (b) interpolation based methods, and (c) our AdvMixup. 'A','B', and 'C' denote three training points, and the orange regions denote the regularization areas.

combination of available training samples without constraining the points outside, thus losing control in extreme cases where the model oscillates in a position near a training sample but not covered by any interpolation path.

In this paper, we propose a novel consistency regularization technique, called Adversarial Mixup (AdvMixup), by unifying the local neighborhood and the in-between neighborhood. In particular, we consider the neighborhood formed by *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 $x_j^{(adv)}$ of x_j, i.e., $\hat{x} = \lambda x_i + (1 - \lambda)x_j^{(adv)}$. Then we enforce the consistency between $f(\hat{x})$ and $\lambda f(x_i) + (1 - \lambda)f(x_j)$. By mixing up a training sample with an adversarial sample, we fuse the benefits of both branches. The regularization area for our AdvMixup can be visualized as in Fig. 1c.

We evaluate our AdvMixup on both synthetic data and benchmark datasets, and the experimental results demonstrate that AdvMixup outperforms the baseline methods which consider only local neighborhood or in-between neighborhood, especially when fewer labeled data are given. Furthermore, we justify the robustness of our method under white-box attacks and black-box attacks. In both scenarios, our AdvMixup show significantly better robustness compared to the state-of-the-art.

II. Related Work

With the aid of unlabeled data, SSL methods aim to design a regularization term to encourage the model to comply with the cluster assumption [1], which favors decision boundaries lying in low-density regions and smooth model behaviors. In the following, we briefly review the state-of-the-art consistency regularization methods for SSL.

An important research line in consistency regularization constrains the model to have consistent predictions in the local neighborhood around training inputs, where the local neighborhood is usually represented as variants of the input or model parameters. The Π model from [5] and [6] constructed different input variants with stochastic image transformation and additive Gaussian noise, as well as different model variants with dropout layers. Wei et al. [13] integrated the Π model with the generative adversarial networks (GAN) based semi-supervised learning approaches [14], where the classifier was forced to correctly classify labeled samples and distinguish real unlabeled samples and fake samples from a generator. Laine et al. [6] proposed a Temporal Ensembling approach by applying the consistency constraint between current model prediction and the exponential moving average (EMA) of all historical predictions for a given input. Tarvainen et al. [7] further improved Temporal Ensembling by considering the consistency between the predictions given by current model parameters and the EMA of model parameters. Considering the insufficient power of the isotropic perturbations, Miyato et al. [8] proposed the VAT model by using adversarial perturbations which point out the model's most vulnerable directions, to better represent the local neighborhood.

Another promising research line in consistency regularization considers the consistency between pairs of training samples. Luo et al. [15] enhanced the local neighborhood based methods by pulling similar sample pairs towards each other while pushing the dissimilar pairs away in the low-dimensional feature space. Under supervised setting, Zhang et al. [12] proposed the mixup model which encourages 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. [16] extended the mixup operation to the hidden layers. Guo et al. [17] proposed to adaptively generate the mixing parameter for a specific pair, to avoid overlapping between the mixed samples and the real ones. Verma et al. [9] generalized the mixup model to the semi-supervised setting where the labels are substituted by the soft labels from a teacher model. The MixMatch [18] model further generalized the mixup mechanism with several techniques such as multiple data augmentation and label sharpening, obtaining strong empirical results on semi-supervised learning.

Orthogonal to the consistency regularization methods, generative models based methods try to improve SSL by learning better data representations with the aid of unlabeled data and generative models like variational auto-encoders [19] or GANs [14], [20]. We leave the integration of our model with this research line for future work.

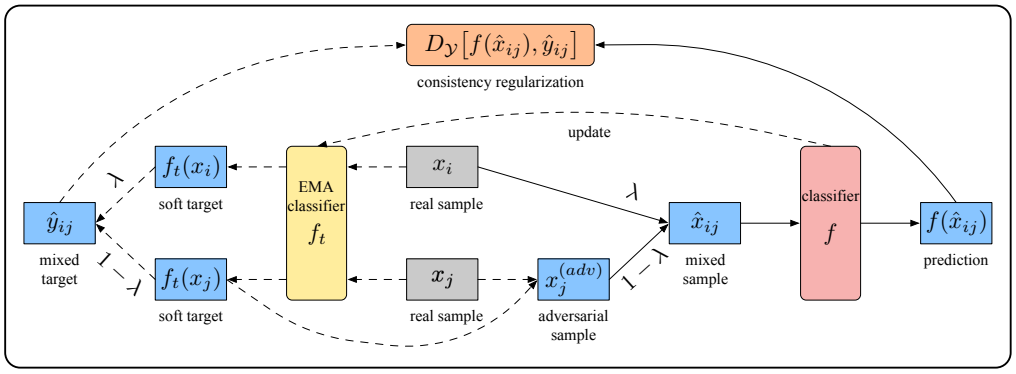


Fig. 2: Overview of the proposed AdvMixup framework. The two gray rectangles, x_i and x_j, denote the entry point. The dashed lines denote the computational paths where gradient back-propagation is disabled.

III. Our Approach: AdvMixup

A. 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 the 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 objective is to learn a mapping function $f : \mathcal{X} \to \mathcal{Y}$ which can generalize to the unseen (x, y) data pairs sampled from the joint probability distribution $P(\mathcal{X}, \mathcal{Y})$.

B. AdvMixup

Standing on the cluster assumption [1], 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, where each pair is composed of a real sample and an adversarial sample.

Formally, given two random 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 $(\hat{x}_{i,j}, \hat{y}_{i,j})$ by 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 [9], we employ the predictions from the EMA model f_t as the soft labels for better target quality [7].

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

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

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

Finally, we arrive at the full objective function for AdvMixup, namely, to minimize

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

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 $\mathcal{L}_{\text{reg}}$. We summarize the training procedure of AdvMixup in Algorithm 1.

Adversarial Sample Generation. An adversarial sample is a 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 [10], [11]. In this paper, we adopt the virtual adversarial example generation method from [8], where "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

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

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 [8], which requires a low additional computational cost to the basic mixup model.

Generality. The proposed AdvMixup can generalize to both the perturbation-based regularization method VAT [8] and the mixup-based regularization method ICT [9]. If $\lambda \to 0$, the constructed virtual data sample is $(x_j^{(adv)}, f_t(x_j))$ in (1), reducing to the VAT model. If the adversarial perturbation degenerate to zero, i.e., $x_j^{(adv)} = x_j$, (1) reduces to the ICT model.

IV. Why AdvMixup?

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

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 random parameters θ;
▷ f's EMA version f_t with random parameters θ_t
▷ perturbation norm ϵ in (4)
▷ mixup parameter α for the Beta distribution;
▷ regularization weight β; f_t's update ratio γ
▷ **FOR** $k = 1, \ldots,$ **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 using (4)

$$B_{u^+}^{(adv)} = \{(x_i^{(adv)} = x_i + r_i^{(adv)}, f_t(x_i)) | x_i = B_u[i]\}_{i=1}^{n_u}$$

▷ Shuffle $B_{u^+}^{(adv)}$ as $B_{u^+,s}^{(adv)}$
▷ Sample $\lambda \sim Beta(\alpha, \alpha)$
▷ Construct a virtual data batch $\hat{B}_{u^+} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{n_u}$ with

$$\hat{x}_i = \lambda x_i^1 + (1-\lambda) x_i^2,$$
$$\hat{y}_i = \lambda y_i^1 + (1-\lambda) y_i^2$$

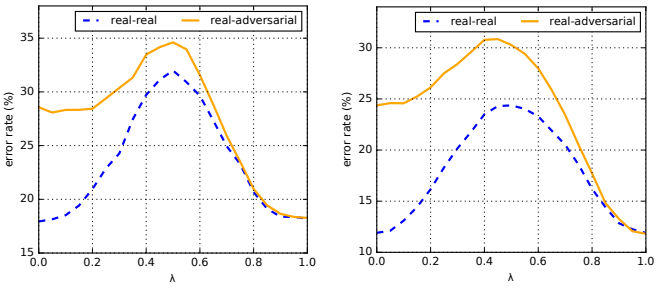
where $(x_i^1, y_i^1) = B_{u^+}[i], (x_i^2, y_i^2) = B_{u^+,s}^{(adv)}[i]$
▷ Compute the consistency regularization term

$$\mathcal{L}_{\text{reg}} = \frac{1}{n_u} \sum_{(\hat{x}_i, \hat{y}_i) \in \hat{B}_{u^+}} D_{\mathcal{Y}}[f(\hat{x}_i), \hat{y}_i]$$

▷ Evaluate the full objective function $\mathcal{L} = \mathcal{L}_{\text{nll}} + \beta \mathcal{L}_{\text{reg}}$
▷ Update θ based on the gradient $\nabla_\theta \mathcal{L}$
▷ Update $\theta_t = (1-\gamma)\theta_t + \gamma\theta$
▷ **END FOR**
▷ **OUTPUT:** θ **and** θ_t

Reasonableness. The mixup model [12] and ICT model [9] encourage the classifier to have linear transition in-between real samples, thus pushing the decision boundary to low-density areas. Our AdvMixup takes one additional step by creating an adversarial variant for one sample in each 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 $\langle x_i, x_j \rangle$ 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 the classifier's flaws which violate the cluster assumption. An effective approach is expected to detect these flaws **more significantly** and **more comprehensively**. Compared with the methods seeking for the flaws in-between neighborhood of training samples like ICT [9], our AdvMixup can create the virtual samples that violate the cluster assumption more 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 Fig. 3, the supervised model exhibits much larger error rates along the real-adversarial interpolation path of AdvMixup (orange solid line) than the real-real interpolation path of ICT (blue dashed line). Compared with the methods seeking for flaws in the local neighborhood of training samples like VAT [8], 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 for the in-between neighborhood when $\lambda > 0$. The usefulness of regularizing the in-between neighborhood has been validated by [9], [12] and also illustrated in Fig. 3 where the error rate reaches the maximum when λ is around 0.5.



(a) CIFAR-10 (b) SVHN

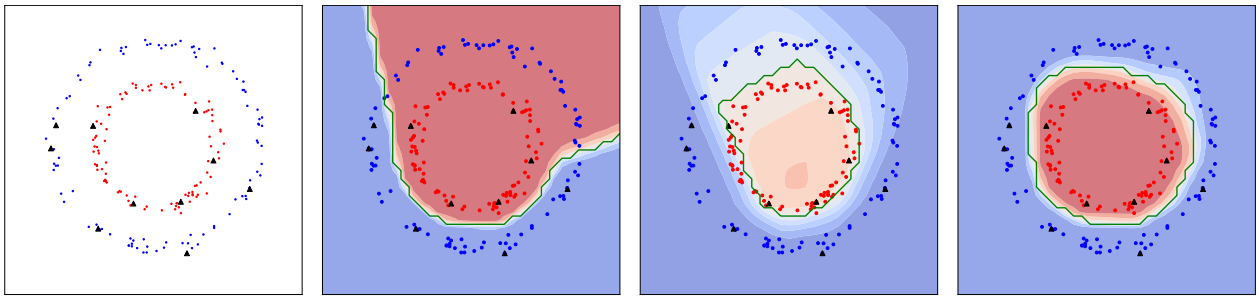
Fig. 3: Prediction error rates of the supervised model (trained with only labeled data) on the virtual samples along the *real-real* interpolation paths (blue dashed line) defined by the ICT model [9] and the *real-adversarial* interpolation paths (orange solid line) defined by the proposed AdvMixup. (a) Results on the CIFAR-10 dataset where 4000 labeled data samples are used to train the supervised model. (b) Results on the SVHN dataset where 1000 labeled data samples are used to train the supervised model.

A. Case Study on Synthetic Data

We evaluate the proposed AdvMixup against VAT and ICT on a synthetic dataset with two classes. As shown in Fig. 4a, the training points form 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 to, 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 Fig. 4, and we have the following major observations. First, VAT can not successfully classify the two classes: it mistakenly predicts a proportion of blue points as red points. Since these blue points have a relatively larger distance to other surrounding



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

Fig. 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. 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.

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 roughly achieves to distinguish the two classes. However, the decision boundary stays too close to some data points, making the model vulnerable to even small noises. Note that the points next to the decision boundary lie in a lighter area of the contour and thus have lower confidence scores. ICT regards this result as a feasible solution since it complies with their constraint for in-between training samples. Third, our 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.

V. 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, robustness analysis, and parameter analysis to validate the effectiveness of our model.

A. 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 [5]–[9], [21], 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 samples on CIFAR-10 and 1000 samples on SVHN, respectively.

B. Implementation Details

Data preprocessing. Unless otherwise stated, we adopt standard data augmentation and data normalization in the preprocessing phase following our baselines. 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 convolution neural network architecture as in the ICT model [9], which eliminates the dropout layers compared to the variants in [5]–[8], [15].

Hyper-parameters. We directly use the perturbation norm ϵ following the code[1] of the VAT model [8], 8.0 for the CIFAR-10 dataset and 3.5 for the SVHN dataset, respectively. The update ratio γ of the EMA model is set to 0.001 following [9]. We search the optimal mixup parameter α in the beta distribution and the regularization weight β in (3) via the validation performance. As a result, α is set as 2.0 on CIFAR-10 and 0.1 on SVHN, respectively; β is set as 50, 100, and 100 on CIFAR-10 for 1000, 2000, and 4000 labeled samples correspondingly, and as 50, 100, and 100 on SVHN for 250, 500, and 1000 labeled samples correspondingly.

Model training. We adopt the mean squared error as the divergence metric in (2) and (4). The batch size is 32 for labeled data and 128 for unlabeled data. We follow the rest of settings as in [9]: 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 strategy [22]; 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.

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

TABLE I: Test error rates (%) of different methods on CIFAR-10. Results for the Supervised method in the first block are duplicated from [9]. 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
Π model [6]	31.65 ± 1.20	17.57 ± 0.44	12.36 ± 0.31
TempEns [6]	23.31 ± 1.01	15.64 ± 0.39	12.16 ± 0.24
MT [7]	21.55 ± 1.48	15.73 ± 0.31	12.31 ± 0.28
VAT [8]	–	–	11.36 ± 0.34
VAT+EntMin [8]	–	–	10.55 ± 0.05
VAdD [23]	–	–	11.32 ± 0.11
VAdD + VAT [23]	–	–	9.22 ± 0.10
TempEns+SNTG [15]	18.41 ± 0.52	13.64 ± 0.32	10.93 ± 0.14
VAT+EntMin+SNTG [15]	–	–	9.89 ± 0.34
CT-GAN [13]	–	–	9.98 ± 0.21
CVT [24]	–	–	10.11 ± 0.15
MT+ fast-SWA [25]	15.58 ± 0.12	11.02 ± 0.23	9.05 ± 0.21
ICT [9]	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 II: Test error rates (%) of different methods on SVHN. Results for the Supervised method in the first block are duplicated from [9]. 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
Π model [6]	9.93 ± 1.15	6.65 ± 0.53	4.82 ± 0.17
TempEns [6]	12.62 ± 2.91	5.12 ± 0.13	4.42 ± 0.16
MT [7]	4.35 ± 0.50	4.18 ± 0.27	3.95 ± 0.19
VAT [8]	–	–	5.42 ± 0.22
VAT+EntMin [8]	–	–	3.86 ± 0.11
VAdD [23]	–	–	4.16 ± 0.08
VAdD + VAT [23]	–	–	3.55 ± 0.05
Π+SNTG [15]	5.07 ± 0.25	4.52 ± 0.30	3.82 ± 0.25
MT+SNTG [15]	4.29 ± 0.23	3.99 ± 0.24	3.86 ± 0.27
ICT [9]	4.78 ± 0.68	4.23 ± 0.15	3.89 ± 0.04
AdvMixup	$\mathbf{3.95 \pm 0.70}$	$\mathbf{3.37 \pm 0.09}$	$\mathbf{3.07 \pm 0.18}$

C. Results

The evaluation results of our proposed AdvMixup against several state-of-the-art methods on CIFAR-10 and SVHN are shown in Table I and Table II, respectively. The baseline semi-supervised learning methods encompass consistency regularization methods based on local neighborhoods [6]–[8], [23]–[25], in-between neighborhoods [9], and those combining them [15]. From Table I and Table II, we have the following observations.

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 have smaller variance compared to the natural images of CIFAR-10, and the baselines already achieve pretty high accuracy. Nevertheless, AdvMixup still demonstrates a clear improvement over all the baselines across different numbers of labeled data. In particular, AdvMixup achieves an error rate of 3.95% for 250 labeled samples, which already beats the results of all baselines with 500 labeled samples.

Thirdly, following [9], we also compare with the supervised method (the method in the first block of Table I and Table II), where only the labeled samples are used. For both CIFAR-10 and SVHN, AdvMixup exhibits significant improvement over the supervised baseline across different numbers of labeled data.

TABLE III: Test error rates (%) of different ablated versions on CIFAR-10. Results are averaged over 3 runs.

Method	Test error rates (%)		
	1000 labels	2000 labels	4000 labels
AdvMixup	$\mathbf{9.67 \pm 0.08}$	$\mathbf{8.04 \pm 0.12}$	$\mathbf{7.13 \pm 0.08}$
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

D. Ablation Study

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

- **Adv-Adv Mixup** as an alternative of integrating the local and in-between neighborhood approaches, by defining the interpolation paths between two adversarial samples, i.e., replacing x_i with $x_i^{(adv)}$ in (1).
- **AdvMixup w/o teacher model**, which uses 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 (1).

The results of these ablated variants are shown in Table III. Firstly, interpolating between adversarial examples clearly degrades the performance across different numbers of labeled samples. One possible 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. Secondly, eliminating the teacher model degrades 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 [9].

E. Robustness Analysis

Deep models have been discovered to be particularly vulnerable to adversarial perturbations [10], [11]. To investigate the robustness of our proposed AdvMixup in semi-supervised learning, we compare our AdvMixup with the strongest and most related baseline ICT as well as a simple supervised model under white-box attacks and black-box attacks on CIFAR-10 and SVHN. Models are learned with 4000 labeled samples on CIFAR-10 and 1000 labeled samples on SVHN, respectively.

TABLE IV: Test error rates (%) of different methods on CIFAR-10 and SVHN under white-box attacks. The white-box attacks are generated using the fast gradient method with the perturbation norm ϵ_w.

Method	CIFAR-10					SVHN				
	$\epsilon_w = 1.0$	$\epsilon_w = 2.0$	$\epsilon_w = 3.0$	$\epsilon_w = 5.0$	$\epsilon_w = 8.0$	$\epsilon_w = 0.1$	$\epsilon_w = 0.5$	$\epsilon_w = 1.0$	$\epsilon_w = 2.0$	$\epsilon_w = 3.0$
Supervised	58.50	77.73	86.73	94.2	96.91	19.81	51.71	69.94	82.28	86.46
ICT [9]	24.77	43.28	56.24	69.42	78.38	7.72	28.57	41.87	52.35	58.00
AdvMixup	**17.40**	**30.91**	**42.52**	**58.59**	**70.82**	**5.11**	**14.59**	**24.39**	**37.84**	**47.63**

TABLE V: Test error rates (%) of different methods on CIFAR-10 and SVHN under black-box attacks. The black-box attacks are generated using the fast gradient method with the perturbation norm ϵ_b.

Method	CIFAR-10					SVHN				
	$\epsilon_b = 1.0$	$\epsilon_b = 2.0$	$\epsilon_b = 3.0$	$\epsilon_b = 5.0$	$\epsilon_b = 8.0$	$\epsilon_b = 0.1$	$\epsilon_b = 0.5$	$\epsilon_b = 1.0$	$\epsilon_b = 2.0$	$\epsilon_b = 3.0$
Supervised	29.25	39.38	48.83	63.06	75.75	14.37	24.76	36.92	52.91	62.05
ICT [9]	9.78	12.68	16.03	24.85	37.83	4.19	8.17	15.59	30.43	41.29
AdvMixup	**8.62**	**10.17**	**12.34**	**17.34**	**25.77**	**3.47**	**6.62**	**12.31**	**24.92**	**35.39**

We attack each model with adversarial perturbations crafted towards a source model by using the Fast Gradient Method [11]. In the white-box setting, the source model is the same as the target model for testing. In the black-box setting, the source model is different from the target model. In our experiments, we independently train another supervised model (only using the labeled samples) as the source model for the black-box setting.

Table IV and Table V show the results of different models against white-box attacks and black-box attacks with different values of perturbation norm. Firstly, by regularizing the model using unlabeled data, both ICT and AdvMixup significantly improve the supervised model. Secondly, with 4000 labeled samples for CIFAR-10 and 1000 labeled samples for SVHN, while the difference between AdvMixup and ICT is not quite significant for classifying real images (as shown in Table I and II), AdvMixup shows a clear advantage over ICT for predicting the adversarial images. Specifically, in the white-box setting, AdvMixup reduces the error rate by 7%-13% on CIFAR-10 and by 2%-17% on SVHN; in the black-box setting, AdvMixup reduces the error rate by 1%-12% on CIFAR-10 and by 1%-6% on SVHN. Therefore, we can conclude that the integration of local neighborhood with in-between neighborhood gives AdvMixup an edge in robustness against adversarial perturbations.

F. Parameter Analysis

We investigate the effects of three hyper-parameters, the perturbation norm ϵ, the mixup parameter α, and the regularization weight β. The proposed AdvMixup is compared with VAT (for ϵ and β) and ICT (for α and β), which are most related to our work from the viewpoint of the local neighborhood and the in-between neighborhood, across different values of the hyper-parameters. To make the comparison more direct, no data augmentation is utilized.

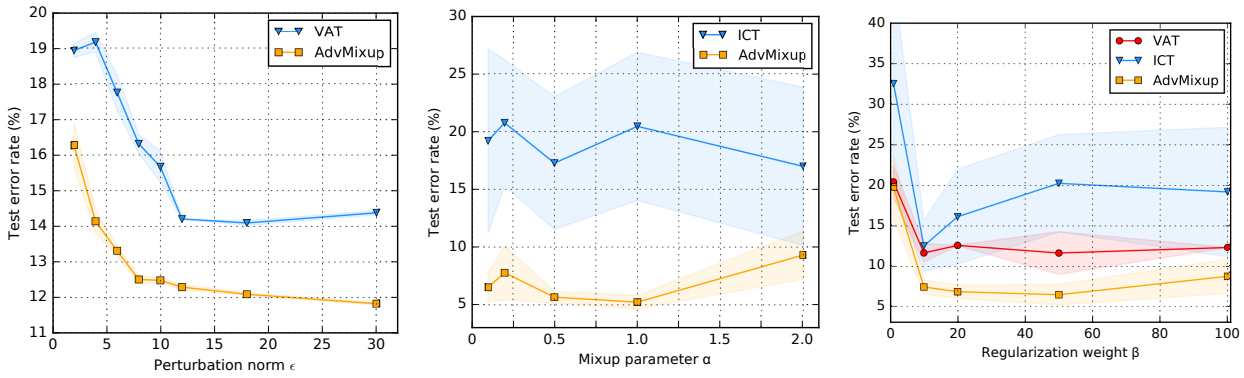
The perturbation norm ϵ controls our model from the local neighborhood perspective. Fig. 5a presents the test performance of our AdvMixup against VAT as a function of ϵ on CIFAR-10 with 4000 labeled samples. We observe that AdvMixup consistently outperforms VAT across different values of ϵ. Besides, the test error rates for $\epsilon \geq 4$ vary in a smaller range in our model than in VAT, showing that AdvMixup is less sensitive to ϵ by incorporating the in-between neighborhood as well.

The mixup parameter α controls our model from the in-between neighborhood perspective. Fig. 5b presents the test performance of AdvMixup against ICT as a function of α on SVHN with 250 labeled samples. We observe that AdvMixup consistently outperforms ICT and shows better stability with smaller variances across different values of α. We notice that the benefits of AdvMixup over ICT is more obvious when no data augmentation is used. This result demonstrates, by considering the local neighborhood, that our AdvMixup can better generalize to different model settings.

The regularization weight β balances the model between fitting the labeled data and satisfying the consistency constraint. Fig. 5c presents the test performance of AdvMixup against VAT and ICT as a function of β on SVHN with 250 labeled samples. We observe that AdvMixup consistently outperforms both VAT and ICT across different values of β, verifying the strength by unifying the local neighborhood and the in-between neighborhood.

VI. 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 for regularization, thus empowering the model with better generalization ability and robustness. Our experiments demonstrate that the proposed AdvMixup constantly outperforms the baselines in terms of both predictions on real samples and on adversarial samples. For further performance improvement, promising directions include fitting AdvMixup into the MixMatch framework [18]



(a) Effect of perturbation norm ϵ (b) Effect of mixup parameter α (c) Effect of regularization weight β

Fig. 5: Effects of different hyper-parameters. (a) Test error rate (%) of VAT and AdvMixup as a function of ϵ on CIFAR-10 with 4000 labeled samples; (b) Test error rate (%) of ICT and AdvMixup as a function of α on SVHN with 250 labeled samples; (c) Test error rate (%) of VAT, ICT and AdvMixup as a function of β on SVHN with 250 labeled samples. Results are averaged over three runs, with the standard deviations indicated by the shaded regions. No data augmentation is utilized.

and integrating AdvMixup with generative models based semi-supervised learning methods [14], [20].

The main limitation of the proposed AdvMixup is the computational overhead brought by the adversarial sample generation, which requires an additional forward-backward pass. For our future work, we plan to evaluate AdvMixup with different adversarial sample generation strategies, study the trade-off between model efficiency and classification performance, and explore the possibility of generating adversarial samples with neglectable cost and without sacrificing performance.

REFERENCES

[1] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation." in *International Conference on Artificial Intelligence and Statistics*, 2005.

[2] T. Joachims, "Transductive inference for text classification using support vector machines," in *International Conference on Machine Learning*, 1999.

[3] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in Neural Information Processing Systems*, 2005.

[4] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, International Conference on Machine Learning*, 2013.

[5] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2016.

[6] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *International Conference on Learning Representations*, 2017.

[7] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, 2017.

[8] T. Miyato, S. ichi Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[9] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," in *International Joint Conference on Artificial Intelligence*, 2019.

[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[12] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[13] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, "Improving the improved training of wasserstein gans: A consistency term and its dual effect," in *International Conference on Learning Representations*, 2018.

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016.

[15] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth neighbors on teacher graphs for semi-supervised learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[16] V. Verma, A. Lamb, C. Beckham, A. Courville, I. Mitliagkis, and Y. Bengio, "Manifold mixup: Encouraging meaningful on-manifold interpolation as a regularizer," *International Conference on Machine Learning*, 2019.

[17] H. Guo, Y. Mao, and R. Zhang, "Mixup as locally linear out-of-manifold regularization," in *Association for the Advancement of Artificial Intelligence*, 2019.

[18] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2019.

[19] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014.

[20] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," in *Advances in Neural Information Processing Systems*, 2017.

[21] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*, 2018.

[22] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017.

[23] S. Park, J. Park, S.-J. Shin, and I.-C. Moon, "Adversarial dropout for supervised and semi-supervised learning," in *Association for the Advancement of Artificial Intelligence*, 2018.

[24] K. Clark, M.-T. Luong, C. D. Manning, and Q. V. Le, "Semi-supervised sequence modeling with cross-view training," in *Conference on Empirical Methods in Natural Language Processing*, 2018.

[25] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson, "There are many consistent explanations of unlabeled data: Why you should average," in *International Conference on Learning Representations*, 2019.