arXiv:1903.03825v3 [stat.ML] 19 May 2019

Interpolation Consistency Training for Semi-Supervised Learning

Vikas Verma
Aalto Univeristy, Finland
Montréal Institute for Learning Algorithms
vikas.verma@aalto.fi

Alex Lamb
Montréal Institute for Learning Algorithms
lambalex@iro.umontreal.ca

Juho Kannala
Aalto Univeristy, Finland
juho.kannala@aalto.fi

Yoshua Bengio
Montréal Institute for Learning Algorithms
CIFAR Senior Fellow
yoshua.umontreal@gmail.com

David Lopez-Paz
Facebook AI Research
dlp@fb.com

Abstract

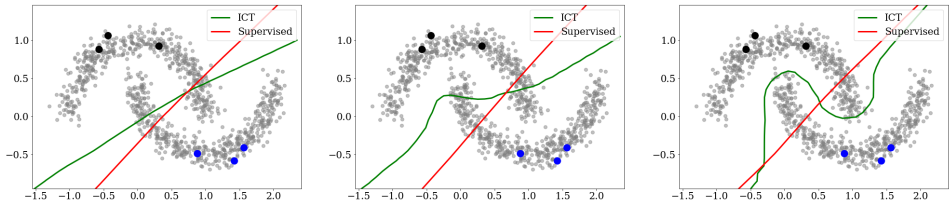
We introduce Interpolation Consistency Training (ICT), a simple and computation efficient algorithm for training Deep Neural Networks in the semi-supervised learning paradigm. ICT encourages the prediction at an interpolation of unlabeled points to be consistent with the interpolation of the predictions at those points. In classification problems, ICT moves the decision boundary to low-density regions of the data distribution. Our experiments show that ICT achieves state-of-the-art performance when applied to standard neural network architectures on the CIFAR-10 and SVHN benchmark datasets.

1 Introduction

Deep learning achieves excellent performance in supervised learning tasks where labeled data is abundant (LeCun et al., 2015). However, labeling large amounts of data is often prohibitive due to time, financial, and expertise constraints. As machine learning permeates an increasing variety of domains, the number of applications where unlabeled data is voluminous and labels are scarce increases. For instance, recognizing documents in extinct languages, where a machine learning system has access to a few labels, produced by highly-skilled scholars (Clanuwat et al., 2018).

The goal of Semi-Supervised Learning (SSL) (Chapelle et al., 2010) is to leverage large amounts of unlabeled data to improve the performance of supervised learning over small datasets. Often, SSL algorithms use unlabeled data to learn additional structure about the input distribution. For instance, the existence of cluster structures in the input distribution could hint the separation of samples into different labels. This is often called the *cluster assumption*: if two samples belong to the same cluster in the input distribution, then they are likely to belong to the same class. The cluster assumption is equivalent to the *low-density separation assumption*: the decision boundary should lie in the low-density regions. The equivalence is easy to infer: A decision boundary which lies in a high-density region, will cut a cluster into two different classes, requiring that samples from different classes lie in the same cluster; which is the violation of the *cluster assumption*. The *low-density separation assumption* has inspired many recent *consistency-regularization* semi-supervised learning

To appear in IJCAI 2019. Code available at https://github.com/vikasverma1077/ICT



(a) After 100 updates (b) After 500 updates (c) After 1000 updates

Figure 1: Interpolation Consistency Training (ICT) applied to the "two moons" dataset, when three labels per class (large dots) and a large amount of unlabeled data (small dots) is available. When compared to supervised learning (red), ICT encourages a decision boundary traversing a low-density region that would better reflect the structure of the unlabeled data. Both methods employ a multilayer perceptron with three hidden ReLU layers of twenty neurons.

techniques, including the Π-model (Sajjadi et al., 2016; Laine & Aila, 2016), temporal ensembling (Laine & Aila, 2016), VAT (Miyato et al., 2018), and the Mean-Teacher (Tarvainen & Valpola, 2017).

Consistency regularization methods for semi-supervised learning enforce the low-density separation assumption by encouraging invariant prediction $f(u) = f(u+\delta)$ for perturbations $u+\delta$ of unlabeled points u. Such consistency and small prediction error can be satisfied simultaneously if and only if the decision boundary traverses a low-density path.

Different consistency regularization techniques vary in how they choose the unlabeled data perturbations δ. One simple alternative is to use random perturbations δ. However, random perturbations are inefficient in high dimensions, as only a tiny proportion of input perturbations are capable of pushing the decision boundary into low-density regions. To alleviate this issue, Virtual Adversarial Training or VAT (Miyato et al., 2018), searches for small perturbations δ that maximize the change in the prediction of the model. This involves computing the gradient of the predictor with respect to its input, which can be expensive for large neural network models.

This additional computation makes VAT (Miyato et al., 2018) and other related methods such as (Park et al., 2018) less appealing in situations where unlabeled data is available in large quantities. Furthermore, recent research has shown that training with adversarial perturbations can hurt generalization performance (Nakkiran, 2019; Tsipras et al., 2018).

To overcome the above limitations, we propose the Interpolation Consistency Training (ICT), an efficient consistency regularization technique for state-of-the-art semi-supervised learning. In a nutshell, ICT regularizes semi-supervised learning by encouraging consistent predictions $f(\alpha u_1 + (1-\alpha)u_2) = \alpha f(u_1) + (1-\alpha) f(u_2)$ at interpolations $\alpha u_1 + (1-\alpha)u_2$ of unlabeled points u_1 and u_2.

Our experimental results on the benchmark datasets CIFAR10 and SVHN and neural network architectures CNN-13 (Laine & Aila, 2016; Miyato et al., 2018; Tarvainen & Valpola, 2017; Park et al., 2018; Luo et al., 2018) and WRN28-2 (Oliver et al., 2018) outperform (or are competitive with) the state-of-the-art methods. ICT is simpler and more computation efficient than several of the recent SSL algorithms, making it an appealing approach to SSL. Figure 1 illustrates how ICT learns a decision boundary traversing a low density region in the "two moons" problem.

2 Interpolation Consistency Training

Given a mixup (Zhang et al., 2018) operation:

$$\text{Mix}_\lambda(a, b) = \lambda \cdot a + (1-\lambda) \cdot b,$$

Interpolation Consistency Training(ICT) trains a prediction model f_θ to provide consistent predictions at interpolations of unlabeled points:

$$f_\theta(\text{Mix}_\lambda(u_j, u_k)) \approx \text{Mix}_\lambda(f_{\theta'}(u_j), f_{\theta'}(u_k)),$$

where θ' is a moving average of θ (Figure 2). But, why do interpolations between unlabeled samples provide a good consistency perturbation for semi-supervised training?

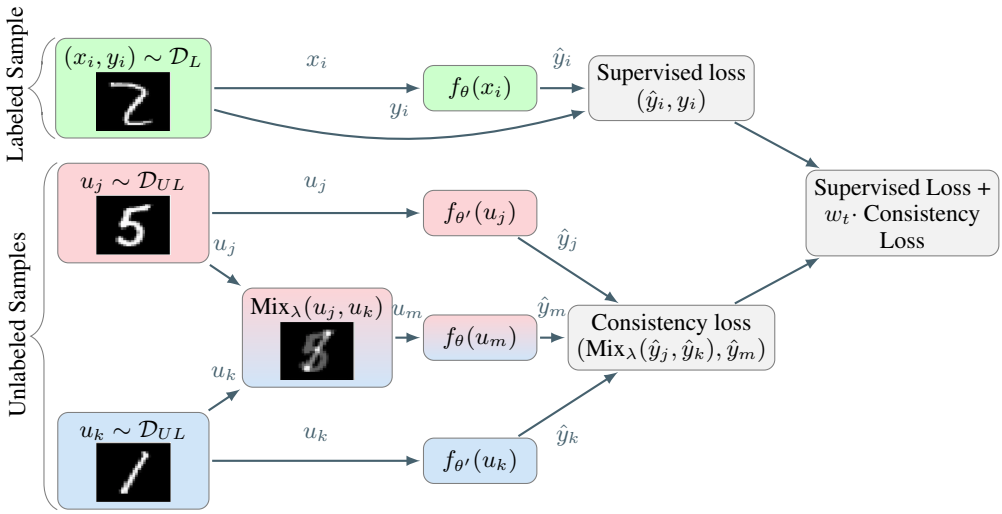


Figure 2: Interpolation Consistency Training (ICT) learns a student network f_θ in a semi-supervised manner. To this end, ICT uses a mean-teacher $f_{\theta'}$, where the teacher parameters θ' are an exponential moving average of the student parameters θ. During training, the student parameters θ are updated to encourage consistent predictions $f_\theta(\text{Mix}_\lambda(u_j, u_k)) \approx \text{Mix}_\lambda(f_{\theta'}(u_j), f_{\theta'}(u_k))$, and correct predictions for labeled examples x_i.

To begin with, observe that the most useful samples on which the consistency regularization should be applied are the samples near the decision boundary. Adding a small perturbation δ to such low-margin unlabeled samples u_j is likely to push $u_j + \delta$ over the other side of the decision boundary. This would violate the *low-density separation assumption*, making $u_j + \delta$ a good place to apply consistency regularization. These violations do not occur at high-margin unlabeled points that lie far away from the decision boundary.

Back to low-margin unlabeled points u_j, how can we find a perturbation δ such that u_j and $u_j + \delta$ lie on opposite sides of the decision boundary? Although tempting, using random perturbations is an inefficient strategy, since the subset of directions approaching the decision boundary is a tiny fraction of the ambient space.

Instead, consider interpolations $u_j + \delta = \text{Mix}_\lambda(u_j, u_k)$ towards a second randomly selected unlabeled examples u_k. Then, the two unlabeled samples u_j and u_k can either:

1. lie in the same cluster,
2. lie in different clusters but belong to the same class,
3. lie on different clusters and belong to the different classes.

Assuming the cluster assumption, the probability of (1) decreases as the number of classes increases. The probability of (2) is low if we assume that the number of clusters for each class is balanced. Finally, the probability of (3) is the highest. Then, assuming that one of (u_j, u_k) lies near the decision boundary (it is a good candidate for enforcing consistency), it is likely (because of the high probability of (3)) that the interpolation towards u_k points towards a region of low density, followed by the cluster of the other class. Since this is a good direction to move the decision, the interpolation is a good perturbation for consistency-based regularization.

Our exposition has argued so far that interpolations between random unlabeled samples are likely to fall in low-density regions. Thus, such interpolations are good locations where consistency-based regularization could be applied. But how should we label those interpolations? Unlike random or adversarial perturbations of single unlabeled examples u_j, our scheme involves two unlabeled examples (u_j, u_k). Intuitively, we would like to push the decision boundary as far as possible from the class boundaries, as it is well known that decision boundaries with large margin generalize better (Shawe-Taylor et al., 1996). In the supervised learning setting, one method to achieve large-margin decision boundaries is mixup (Zhang et al., 2018). In mixup, the decision boundary is pushed far away from the class boundaries by enforcing the prediction model to change linearly in between samples. This is done by training the model f_θ to predict $\text{Mix}_\lambda(y, y')$ at location $\text{Mix}_\lambda(x, x')$, for

random pairs of labeled samples $((x, y), (x', y'))$. Here we extend mixup to the semi-supervised learning setting by training the model f_θ to predict the "fake label" $\text{Mix}_\lambda(f_\theta(u_j), f_\theta(u_k))$ at location $\text{Mix}_\lambda(u_j, u_k)$. In order to follow a more conservative consistent regularization, we encourage the model f_θ to predict the fake label $\text{Mix}_\lambda(f_{\theta'}(u_j), f_{\theta'}(u_k))$ at location $\text{Mix}_\lambda(u_j, u_k)$, where θ' is a moving average of θ, also known as a *mean-teacher* (Tarvainen & Valpola, 2017).

We are now ready to describe in detail the proposed Interpolation Consistency Training (ICT). Consider access to labeled samples $(x_i, y_i) \sim \mathcal{D}_L$, drawn from the joint distribution $P(X, Y)$. Also, consider access to unlabeled samples $u_j, u_k \sim \mathcal{D}_{UL}$, drawn from the marginal distribution $P(X) = \frac{P(X,Y)}{P(Y|X)}$ Our learning goal is to train a model f_θ, able to predict Y from X. By using stochastic gradient descent, at each iteration t, update the parameters θ to minimize

$$L = L_S + w(t) \cdot L_{US}$$

where L_S is the usual cross-entropy supervised learning loss over labeled samples $\mathcal{D}_L$, and L_{US} is our new interpolation consistency regularization term. These two losses are computed on top of (labeled and unlabeled) minibatches, and the ramp function $w(t)$ increases the importance of the consistency regularization term L_{US} after each iteration. To compute L_{US}, sample two minibatches of unlabeled points u_j and u_k, and compute their fake labels $\hat{y}_j = f_{\theta'}(u_j)$ and $\hat{y}_k = f_{\theta'}(u_k)$, where θ' is an moving average of θ (Tarvainen & Valpola, 2017). Second, compute the interpolation $u_m = \text{Mix}_\lambda(u_j, u_k)$, as well as the model prediction at that location, $\hat{y}_m = f_\theta(u_m)$. Third, update the parameters θ as to bring the prediction $\hat{y}_m$ closer to the interpolation of the fake labels $\text{Mix}_\lambda(\hat{y}_j, \hat{y}_k)$. The discrepancy between the prediction $\hat{y}_m$ and $\text{Mix}_\lambda(\hat{y}_j, \hat{y}_k)$ can be measured using any loss; in our experiments, we use the mean squared error. Following (Zhang et al., 2018), on each update we sample a random λ from $\text{Beta}(\alpha, \alpha)$.

In sum, the population version of our ICT term can be written as:

$$\mathcal{L}_{US} = \mathop{\mathbb{E}}_{u_j, u_k \sim P(X)} \; \mathop{\mathbb{E}}_{\lambda \sim \text{Beta}(\alpha,\alpha)} \ell(f_\theta(\text{Mix}_\lambda(u_j, u_k)), \text{Mix}_\lambda(f_{\theta'}(u_j), f_{\theta'}(u_k))) \tag{1}$$

ICT is summarized in Figure 2 and Algorithm 1.

3 Experiments

3.1 Datasets

We follow the common practice in semi-supervised learning literature (Laine & Aila, 2016; Miyato et al., 2018; Tarvainen & Valpola, 2017; Park et al., 2018; Luo et al., 2018) and conduct experiments using the CIFAR-10 and SVHN datasets, where only a fraction of the training data is labeled, and the remaining data is used as unlabeled data. We followed the standardized procedures laid out by (Oliver et al., 2018) to ensure a fair comparison.

The CIFAR-10 dataset consists of 60000 color images each of size 32×32, split between 50K training and 10K test images. This dataset has ten classes, which include images of natural objects such as cars, horses, airplanes and deer. The SVHN dataset consists of 73257 training samples and 26032 test samples each of size 32×32. Each example is a close-up image of a house number (the ten classes are the digits from 0-9).

We adopt the standard data-augmentation and pre-processing scheme which has become standard practice in the semi-supervised learning literature (Sajjadi et al., 2016; Laine & Aila, 2016; Tarvainen & Valpola, 2017; Miyato et al., 2018; Luo et al., 2018; Athiwaratkun et al., 2019). More specifically, for CIFAR-10, we first zero-pad each image with 2 pixels on each side. Then, the resulting image is randomly cropped to produce a new 32×32 image. Next, the image is horizontally flipped with probability 0.5, followed by per-channel standardization and ZCA preprocessing. For SVHN, we zero-pad each image with 2 pixels on each side and then randomly crop the resulting image to produce a new 32×32 image, followed by zero-mean and unit-variance image whitening.

3.2 Models

We conduct our experiments using CNN-13 and Wide-Resnet-28-2 architectures. The CNN-13 architecture has been adopted as the standard benchmark architecture in recent state-of-the-art SSL

Algorithm 1 The Interpolation Consistency Training (ICT) Algorithm

Require: $f_\theta(x)$: neural network with trainable parameters θ

Require: $f_{\theta'}(x)$ mean teacher with θ' equal to moving average of θ

Require: $\mathcal{D}_L(x, y)$: collection of the labeled samples

Require: $\mathcal{D}_{UL}(x)$: collection of the unlabeled samples

Require: α: rate of moving average

Require: $w(t)$: ramp function for increasing the importance of consistency regularization

Require: T: total number of iterations

Require: Q: random distribution on [0,1]

Require: $\text{Mix}_\lambda(a, b) = \lambda a + (1 - \lambda)b$.

for $t = 1, \ldots, T$ **do**

Sample $\{(x_i, y_i)\}_{i=1}^B \sim \mathcal{D}_L(x, y)$ ▷ Sample labeled minibatch

$L_S = \text{CrossEntropy}(\{(f_\theta(x_i), y_i)\}_{i=1}^B)$ ▷ Supervised loss (cross-entropy)

Sample $\{u_j\}_{j=1}^U, \{u_k\}_{k=1}^U \sim \mathcal{D}_{UL}(x)$ ▷ Sample two unlabeled examples

$\{\hat{y}_j\}_{j=1}^U = \{f_{\theta'}(u_j)\}_{j=1}^U, \{\hat{y}_k\}_{k=1}^U = \{f_{\theta'}(u_k)\}_{k=1}^U$ ▷ Compute fake labels

Sample $\lambda \sim Q$ ▷ sample an interpolation coefficient

$(u_m = \text{Mix}_\lambda(u_j, u_k), \hat{y}_m = \text{Mix}_\lambda(\hat{y}_j, \hat{y}_k))$ ▷ Compute interpolation

$L_{US} = \text{ConsistencyLoss}(\{(f_\theta(u_m), \hat{y}_m)\}_{m=1}^U)$ ▷ e.g., mean squared error

$L = L_S + w(t) \cdot L_{US}$ ▷ Total Loss

$g_\theta \leftarrow \nabla_\theta L$ ▷ Compute Gradients

$\theta' = \alpha\theta' + (1 - \alpha)\theta$ ▷ Update moving average of parameters

$\theta \leftarrow \text{Step}(\theta, g_\theta)$ ▷ e.g. SGD, Adam

end for

return θ

methods (Laine & Aila, 2016; Tarvainen & Valpola, 2017; Miyato et al., 2018; Park et al., 2018; Luo et al., 2018). We use its variant (i.e., without additive Gaussian noise in the input layer) as implemented in (Athiwaratkun et al., 2019). We also removed the Dropout noise to isolate the improvement achieved through our method. Other SSL methods in Table 1 and Table 2 use the Dropout noise, which gives them more regularizing capabilities. Despite this, our method outperforms other methods in several experimental settings.

(Oliver et al., 2018) performed a systematic study using Wide-Resnet-28-2 (Zagoruyko & Komodakis, 2016), a specific residual network architecture, with extensive hyperparameter search to compare the performance of various consistency-based semi-supervised algorithms. We evaluate ICT using this same setup as a mean towards a fair comparison to these algorithms.

3.3 Implementation details

We used the SGD with nesterov momentum optimizer for all of our experiments. For the experiments in Table 1 and Table 2, we run the experiments for 400 epochs. For the experiments in Table 3, we run experiments for 600 epochs. The initial learning rate was set to 0.1, which is then annealed using the cosine annealing technique proposed in (Loshchilov & Hutter, 2016) and used by (Tarvainen & Valpola, 2017). The momentum parameter was set to 0.9. We used an L2 regularization coefficient 0.0001 and a batch-size of 100 in our experiments.

In each experiment, we report mean and standard deviation across three independently run trials.

Table 1: Error rates (%) on CIFAR-10 using CNN-13 architecture. We ran three trials for ICT.

Model	1000 labeled 50000 unlabeled	2000 labeled 50000 unlabeled	4000 labeled 50000 unlabeled
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, 2016)	31.65 ± 1.20	17.57 ± 0.44	12.36 ± 0.31
TempEns (Laine & Aila, 2016)	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)	–	–	$11.36 \pm$ NA
VAT+Ent (Miyato et al., 2018)	–	–	$10.55 \pm$ NA
VAdD (Park et al., 2018)	–	–	11.32 ± 0.11
SNTG (Luo et al., 2018)	18.41 ± 0.52	13.64 ± 0.32	10.93 ± 0.14
MT+ Fast SWA (Athiwaratkun et al., 2019)	$15.58 \pm$ NA	$11.02 \pm$ NA	$9.05 \pm$ NA
ICT	$\mathbf{15.48 \pm 0.78}$	$\mathbf{9.26 \pm 0.09}$	$\mathbf{7.29 \pm 0.02}$

Table 2: Error rates (%) on SVHN using CNN-13 architecture. We ran three trials for ICT.

Model	250 labeled 73257 unlabeled	500 labeled 73257 unlabeled	1000 labeled 73257 unlabeled
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, 2016)	9.93 ± 1.15	6.65 ± 0.53	4.82 ± 0.17
TempEns (Laine & Aila, 2016)	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)	–	–	$5.42 \pm$ NA
VAT+Ent (Miyato et al., 2018)	–	–	$3.86 \pm$ NA
VAdD (Park et al., 2018)	–	–	4.16 ± 0.08
SNTG (Luo et al., 2018)	$\mathbf{4.29 \pm 0.23}$	$\mathbf{3.99 \pm 0.24}$	$\mathbf{3.86 \pm 0.27}$
ICT	4.78 ± 0.68	4.23 ± 0.15	3.89 ± 0.04

The consistency coefficient $w(t)$ is ramped up from its initial value 0.0 to its maximum value at one-fourth of the total number of epochs using the same sigmoid schedule of (Tarvainen & Valpola, 2017). We used MSE loss for computing the consistency loss following (Laine & Aila, 2016; Tarvainen & Valpola, 2017). We set the decay coefficient for the mean-teacher to 0.999 following (Tarvainen & Valpola, 2017).

We conduct hyperparameter search over the two hyperparameters introduced by our method: the maximum value of the consistency coefficient $w(t)$ (we searched over the values in $\{$1.0, 10.0, 20.0, 50.0, 100.0$\}$) and the parameter α of distribution $Beta(\alpha, \alpha)$ (we searched over the values in $\{$0.1, 0.2, 0.5, 1.0$\}$). We select the best hyperparameter using a validation set of 5000 and 1000 labeled samples for CIFAR-10 and SVHN respectively. This size of the validation set is the same as that used in the other methods compared in this work.

We note the in all our experiments with ICT, to get the supervised loss, we perform the interpolation of labeled sample pair and their corresponding labels (as in *mixup* (Zhang et al., 2018)). To make sure, that the improvements from ICT are not only because of the supervised *mixup* loss, we provide the direct comparison of ICT against supervised *mixup* and *Manifold Mixup* training in the Table 1 and Table 2.

3.4 Results

We provide the results for CIFAR10 and SVHN datasets using CNN-13 architecture in the Table 1 and Table 2, respectively.

Table 3: Results on CIFAR10 (4000 labels) and SVHN (1000 labels) (in test error %). All results use the same standardized architecture (WideResNet-28-2). Each experiment was run for three trials. † refers to the results reported in (Oliver et al., 2018). We did not conduct any hyperparameter search and used the best hyperparameters found in the experiments of Table 1 and 2 for CIFAR10(4000 labels) and SVHN(1000 labels)

SSL Approach	CIFAR10 4000 labeled 50000 unlabeled	SVHN 1000 labeled 73257 unlabeled
Supervised †	20.26 ± 0.38	12.83 ± 0.47
Mean-Teacher †	15.87 ± 0.28	5.65 ± 0.47
VAT †	13.86 ± 0.27	5.63 ± 0.20
VAT-EM †	13.13 ± 0.39	5.35 ± 0.19
ICT	$\mathbf{7.66 \pm 0.17}$	$\mathbf{3.53 \pm 0.07}$

To justify the use of a SSL algorithm, one must compare its performance against the state-of-the-art supervised learning algorithm (Oliver et al., 2018). To this end, we compare our method against two state-of-the-art supervised learning algorithms (Zhang et al., 2018; Verma et al., 2018), denoted as Supervised(Mixup) and Supervised(Manifold Mixup), respectively in Table 1 and 2. ICT method passes this test with a wide margin, often resulting in a two-fold reduction in the test error in the case of CIFAR10 (Table 1) and a four-fold reduction in the case of SVHN (Table 2)

Furthermore, in Table 1, we see that ICT improves the test error of other strong SSL methods. For example, in the case of 4000 labeled samples, it improves the test error of best-reported method by $\sim 25\%$. The best values of the hyperparameter max-consistency coefficient for 1000, 2000 and 4000 labels experiments were found to be 10.0, 100.0 and 100.0 respectively and the best values of the hyperparameter α for 1000, 2000 and 4000 labels experiments were found to be 0.2, 1.0 and 1.0 respectively. In general, we observed that for less number of labeled data, lower values of max-consistency coefficient and α obtained better validation errors.

For SVHN, the test errors obtained by ICT are competitive with other state-of-the-art SSL methods (Table 2). The best values of the hyperparameters max-consistency coefficient and α were found to be 100 and 0.1 respectively, for all the ICT results reported in the Table 2.

(Oliver et al., 2018) performed extensive hyperparameter search for various consistency regularization SSL algorithm using the WRN-28-2 and they report the best test errors found for each of these algorithms. For a fair comparison of ICT against these SSL algorithms, we conduct experiments on WRN-28-2 architecture. The results are shown in Table 3. ICT achieves improvement over other methods both for the CIFAR10 and SVHN datasets.

We note that unlike other SSL methods of Table 1, Table 2 and Table 3, we do not use Dropout regularizer in our implementation of CNN-13 and WRN-28-2. Using Dropout along with the ICT may further reduce the test error.

3.5 Ablation Study

- Effect of not using the mean-teacher in ICT: We note that Π-model, VAT and VAdD methods in Table 1 and Table 2 do not use a mean-teacher to make predictions on the unlabeled data. Although the mean-teacher (Tarvainen & Valpola, 2017) used in ICT does not incur any significant computation cost, one might argue that a more direct comparison with Π-model, VAT and VAdD methods requires not using a mean-teacher. To this end, we conduct an experiment on the CIFAR10 dataset, without the mean-teacher in ICT, i.e. the prediction on the unlabeled data comes from the network $f_\theta(x)$ instead of the mean-teacher network $f_{\theta'}(x)$ in Equation 1. We obtain test errors of $19.56 \pm 0.56\%$, $14.35 \pm 0.15\%$ and $11.19 \pm 0.14\%$ for 1000, 2000, 4000 labeled samples respectively (We did not conduct any hyperparameter search for these experiments and used the best hyperparameters found in the ICT experiments of Table 1). This shows that even without a mean-teacher, ICT has major a advantage over methods such as VAT (Miyato et al., 2018) and VAdD (Park et al., 2018) that it does not require an additional gradient computation yet performs on the same level of the test error.

- Effect of not having the *mixup* supervised loss: In Section 3.3, we noted that to get the supervised loss, we perform the interpolation of labeled sample pair and their corresponding labels (*mixup* supervised loss as in (Zhang et al., 2018)). Will the performance of ICT be significantly reduced by not having the *mixup* supervised loss? We conducted experiments with ICT on both CIFAR10 and SVHN with the *vanilla* supervised loss. For CIFAR10, we obtained test errors of 14.86 ± 0.39, 9.02 ± 0.12 and 8.23 ± 0.22 for 1000, 2000 and 4000 labeled samples respectively. We did not conduct any hyperparameter search and used the best values of hyperparameters (max-consistency coefficient and α) found in the experiments of the Table 1. We observe that in the case of 1000 and 2000 labeled samples, there is no increase in the test error (w.r.t having the *mixup* supervised loss), whereas in the case of 4000 labels, the test error increases by approximately 1% . This suggests that, in the low labeled data regimes, not having the *mixup* supervised loss in the ICT does not incur any significant increase in the test error.

4 Related Work

This work builds on two threads of research: consistency-regularization for semi-supervised learning and interpolation-based regularizers.

On the one hand, consistency-regularization semi-supervised learning methods (Sajjadi et al., 2016; Laine & Aila, 2016; Tarvainen & Valpola, 2017; Miyato et al., 2018; Luo et al., 2018; Athiwaratkun et al., 2019) encourage that realistic perturbations $u + \delta$ of unlabeled samples u should not change the model predictions $f_\theta(u)$. These methods are motivated by the *low-density separation assumption* (Chapelle et al., 2010), and as such push the decision boundary to lie in the low-density regions of the input space, achieving larger classification margins. ICT differs from these approaches in two aspects. First, ICT chooses perturbations in the direction of another randomly chosen unlabeled sample, avoiding expensive gradient computations. When interpolating between distant points, the regularization effect of ICT applies to larger regions of the input space.

On the other hand, interpolation-based regularizers (Zhang et al., 2018; Tokozume et al., 2018; Verma et al., 2018) have been recently proposed for supervised learning, achieving state-of-the-art performances across a variety of tasks and network architectures. While (Zhang et al., 2018; Tokozume et al., 2018) was proposed to perform interpolations in the input space, (Verma et al., 2018) proposed to perform interpolation also in the hidden space representations. Furthermore, in the unsupervised learning setting, (Berthelot et al., 2019) proposes to measure the realism of latent space interpolations from an autoencoder to improve its training.

Other works have approached semi-supervised learning from the perspective of generative models. Some have approached this from a consistency point of view, such as (Lecouat et al., 2018), who proposed to encourage smooth changes to the predictions along the data manifold estimated by the generative model (trained on both labeled and unlabeled samples). Others have used the discriminator from a trained generative adversarial network (Goodfellow et al., 2014) as a way of extracting features for a purely supervised model (Radford et al., 2015). Still, others have used trained inference models as a way of extracting features (Dumoulin et al., 2016).

5 Conclusion

Machine learning is having a transformative impact on diverse areas, yet its application is often limited by the amount of available labeled data. Progress in semi-supervised learning techniques holds promise for those applications where labels are expensive to obtain. In this paper, we have proposed a simple but efficient semi-supervised learning algorithm, Interpolation Consistency Training(ICT), which has two advantages over previous approaches to semi-supervised learning. First, it uses almost no additional computation, as opposed to computing adversarial perturbations or training generative models. Second, it outperforms strong baselines on two benchmark datasets, even without an extensive hyperparameter tuning. As for the future work, extending ICT to interpolations not only at the input but at hidden representations (Verma et al., 2018) could improve the performance even further. Another direction for future work is to better understand the theoretical properties of interpolation-based regularizers in the SSL paradigm.

Acknowledgments

Vikas Verma was supported by Academy of Finland project 13312683 / Raiko Tapani AT kulut. We would also like to acknowledge Compute Canada for providing computing resources used in this work.

References

Athiwaratkun, B., Finzi, M., Izmailov, P., and Wilson, A. G. There are many consistent explanations of unlabeled data: Why you should average. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkgKBhA5Y7.

Berthelot, D., Raffel, C., Roy, A., and Goodfellow, I. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1fQSiCcYm.

Chapelle, O., Schlkopf, B., and Zien, A. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010. ISBN 0262514125, 9780262514125.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. *CoRR*, abs/1610.02242, 2016. URL http://arxiv.org/abs/1610.02242.

Lecouat, B., Foo, C.-S., Zenati, H., and Chandrasekhar, V. Manifold regularization with gans for semi-supervised learning. *arXiv preprint arXiv:1807.04307*, 2018.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436, 2015.

Loshchilov, I. and Hutter, F. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016. URL http://arxiv.org/abs/1608.03983.

Luo, Y., Zhu, J., Li, M., Ren, Y., and Zhang, B. Smooth neighbors on teacher graphs for semi-supervised learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 8896–8905, 2018.

Miyato, T., ichi Maeda, S., Koyama, M., and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Nakkiran, P. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*, 2019.

Oliver, A., Odena, A., Raffel, C., Cubuk, E. D., and Goodfellow, I. J. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. In *Neural Information Processing Systems (NIPS)*, 2018.

Park, S., Park, J., Shin, S.-J., and Moon, I.-C. Adversarial dropout for supervised and semi-supervised learning. *AAAI*, 2018.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 1171–1179, USA, 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9. URL http://dl.acm.org/citation.cfm?id=3157096.3157227.

Shawe-Taylor, J., Bartlett, P., C. Williamson, R., and Anthony, M. A framework for structural risk minimisation. pp. 68–76, 01 1996. doi: 10.1145/238061.238070.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30*, pp. 1195–1204, 2017.

Tokozume, Y., Ushiku, Y., and Harada, T. Between-class learning for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. *stat*, 1050:11, 2018.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Courville, A., Lopez-Paz, D., and Bengio, Y. Manifold Mixup: Better Representations by Interpolating Hidden States. *arXiv e-prints*, art. arXiv:1806.05236, Jun 2018.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In Richard C. Wilson, E. R. H. and Smith, W. A. P. (eds.), *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL https://dx.doi.org/10.5244/C.30.87.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.