Data-Efficient Semi-Supervised Learning by Reliable Edge Mining

Peibin Chen[1], Tao Ma[1], Xu Qin[1], Weidi Xu[2], Shuchang Zhou[3],
[1]Peking University, [2]Ant Financial Services Group, [3]Megvii
{pbchen, taoma, qinxu}@pku.edu.cn, weidi.xwd@alibaba-inc.com, zsc@megvii.com

Abstract

Learning powerful discriminative features is a challenging task in Semi-Supervised Learning, as the estimation of the feature space is more likely to be wrong with scarcer labeled data. Previous methods utilize a relation graph with edges representing 'similarity' or 'dissimilarity' between nodes. Similar nodes are forced to output consistent features, while dissimilar nodes are forced to be inconsistent. However, since unlabeled data may be wrongly labeled, the judgment of edges may be unreliable. Besides, the nodes connected by edges may already be well fitted, thus contributing little to the model training. We propose Reliable Edge Mining (REM), which forms a reliable graph by only selecting reliable and useful edges. Guided by the graph, the feature extractor is able to learn discriminative features in a data-efficient way, and consequently boosts the accuracy of the learned classifier. Visual analyses show that the features learned are more discriminative and better reveals the underlying structure of the data. REM can be combined with perturbation-based methods like Π-model, TempEns and Mean Teacher to further improve accuracy. Experiments prove that our method is data-efficient on simple tasks like SVHN and CIFAR-10, and achieves state-of-the-art results on the challenging CIFAR-100.

1. Introduction

Deep neural network has shown promising advantages in many applications of machine learning, such as computation vision, speech recognition, and nature language process [15]. One of the key reasons why the technique of deep neural networks achieved such rapid developments is there are huge amounts of labeled datasets. However, it usually takes a lot of time and human efforts to construct fully-labeled datasets because of the complicated works of determining the exact labels for different samples. By contrast, since it is much easier to collect unlabeled data, there have been a lot of efforts made for utilizing the information of unlabeled data, and Semi-Supervised Learning (SSL) is an important branch among them.

SSL aims to benefit from the limited labeled data and large amounts of unlabeled data. In order to generalize better with the unlabeled data, the methods of SSL suppose that the points which have the close proximity in a high-density region should have close outputs [2, 36]. Based upon this assumption, many perturbation-based methods have been proposed [25, 26, 33, 24]. Π method [14] and Mean Teacher [14] force consistent output between student network and teacher network. VAT [21] generates a virtual adversarial example for each input and expects the model to give close output. Although these methods have achieved promising results, they only consider of the unlabeled examples but ignore the associated relationships between these examples. Some methods, such as Luo *et al*. [19], utilize the underlying structure of data by building a teacher graph in the embedding space. The nodes represent the data and the edges represent the consistency of labels between nodes[1]. Then the feature extractor is expected to output similar features for nodes from the same class (the edge is 1), while output dissimilar features for nodes from different classes (the edge is 0). However, on the one hand, they ignore the reliability of the edges (the values of these edges may be wrong), which may leads to the wrong guidance of model training. On the other hand, they ignore the usefulness[2] of the edges, which may leads to the inefficient utilization of data.

We are concerned with the task of constructing a reliable sub-graph given the original graph mentioned before. To guide the model training with the whole dataset, we expect the sub-graph to maintain the nodes in the original graph. But for efficient data utilization and reliable guidance, only useful and reliable edges are expected to be added to the sub-graph. We call the constructed sub-graph as a 'reliable graph'.

In this work, we propose Reliable Edge Mining (REM) to construct such a reliable graph (see Fig. 1). Specifically, we add two attributes to each edge in the original graph: usefulness and certainty. According to the attribute of usefulness, we select useful edges to form some candidate sets, from which the reliable edges are mined according to the

[1] Unlabeled data use the model prediction as their labels.

[2] Usefulness means that the edge contributes to model training.

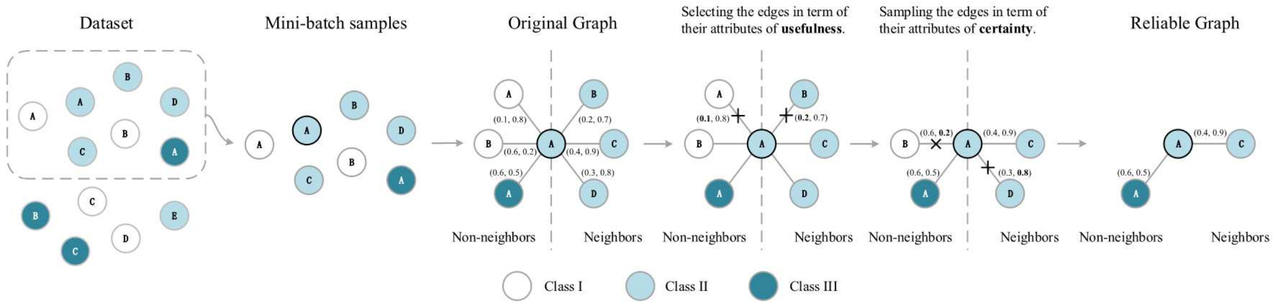


Figure 1: Illustration of REM on a synthetic example. We explain the whole process from the perspective of A from Class II. Firstly, we utilize the label information to form a original graph. Then, the attributes of usefulness (U) and certainty (C) are calculated for each edge (written as (usefulness, certainty) in the figure). We preserve k_i^+ neighbors and k_i^- non-neighbors in the graph according to the attribute of usefulness (in this example, k_i^+ and k_i^- are set to be 2). These preserved edges are further filtered according to the attribute of certainty. Edges with higher value of certainty and usefulness are more likely to be added to the reliable graph.

attribute of certainty. Given these reliable edges and all nodes in the original graph, we are able to construct a reliable graph, with which the feature extractor is expected to learn well using fewer epochs. REM is complementary with currently advanced perturbation-based methods. Experiments on simple tasks like SVHN and CIFAR-10 illustrate that REM achieves comparable results with other state-of-the-art methods with fewer epochs. On more challenging tasks like CIFAR-100, REM surpasses current start-of-the-art results, reducing the best known error rate from 33.62% to 31.95% and from 35.09% to 33.73% with and without augmentation, respectively. Moreover, on Tiny ImageNet, REM reduces the error rate of the baselines from 64.21% to 61.72%. Visualization experiments demonstrate the efficiency of REM in proving useful and reliable edges for model training. We also find that the reliable graph encourages the model for confident outputs, which has been shown beneficial in SSL [7, 16].

The contribution of this paper can be summarized as: (1) We propose REM to construct a reliable graph in the embedding space. The reliable graph is a sub-graph of the original graph, containing only reliable and useful edges. (2) The reliable graph is able to guide the model learning discriminative features in a data-efficient way. (3) We demonstrate that the model becomes confident in its output after training with the reliable graph. (4) REM surpasses previous teacher graph based methods by a obvious margin and achieves currently state-of-the-art results on several benchmarks.

2. Related Work

There has been a long history of developments in Semi Supervised Learning (SSL) [36]. Recently, due to the development of deep learning [15, 10, 13, 28], many SSL ideas have been renovated and achieved impressive improvement compared to full-supervised learning [3, 23, 30, 12, 17]. In this section, we focus on the closely related work. For a detailed review of SSL, we refer readers to [36].

SSL assumes the decision boundary should lie in low-density regions. Based on this assumption, many methods have been proposed [25, 26, 21, 24]. Entropy Regularization [7] minimizes the entropy of softmax output of unlabeled data to encourage low density separation between classes. Pseudo-Labeling (PL) [16] pseudo-labels unlabeled data if the maximum output probabilities are larger than a predefined threshold. These methods encourage the model to give confident outputs, and believe there is a positive correlation between probability and correctness. Our work is also based on this assumption. Given a original graph, we determine which edges are reliable according to the probabilities and select these edges to form a reliable sub-graph. Experiments show that such a graph implicitly encourages the model to give confident output (see Fig. 6) .

Graph-based methods construct a graph with labeled and unlabeled data. Each node represents an example and each edge represents the similarity of examples. There have been a lot of traditional works on building a graph [35, 9, 34]. However, these works fix the graph and only update the weights of edges during training. Luo *et al*. [19] constructs a 'joint-training' sparse graph based on the predicted labels. Then the graph serves as a guideline for metric learning. However, since the predicted labels can be wrong, the edges may be unreliable (see Fig. 2). In addition, as the graph can be large if considering all data in the dataset, stochastic sampling is used for generating a sub-graph from the original graph. We argue that this kind of sub-graph is data-inefficient, thus contributing little to the learning of the em-

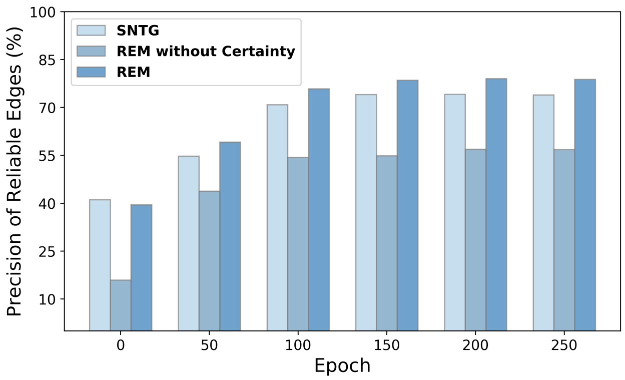


Figure 2: Reliability-Epoch curve under different methods. The precision of reliable edges is defined as the ratio of reliable edges to all edges in the sub-graph. The method of REM without Certainty represents generating the sub-graph without the certainty attribute, only considering the usefulness attribute.

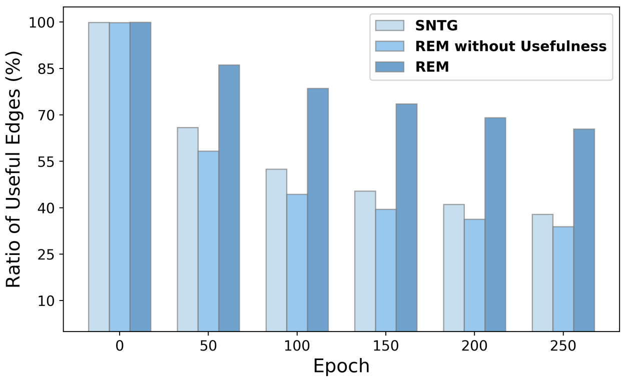


Figure 3: Usefulness-Epoch curve under different methods. The ratio of useful edges is defined as the ratio of useful edges to all edges in the graph. REM without Usefulness means generating the sub-graph only considering the certainty attribute. From the figure, REM without Usefulness brings in a graph with less useful edges compared to SNTG. Taking the usefulness attribute into consideration, REM maintains more useful edges.

bedding space (see Fig. 3). Our work adds two attributes to the edges: reliability and usefulness. Using these attributes, we build a reliable and useful sub-graph, with which the model is trained more efficient (see Fig. 7).

Deep metric learning is a field aiming for an embedding space in which similar data are closer than dissimilar data [22, 27, 29]. As generating all possible pairs would lead to inefficient model training, hard example mining has been widely used for generating valuable pairs. However, hard example mining requires the label information of data, which is lacking in SSL. If we directly use the predictions to be pseudo-labels, a truly positive unlabeled example may be mistaken for a hard negative when the classifier gives incorrect prediction. In our work, we filter out unreliable edges according to the predefined certainty attribute, and then sample useful edges according to the usefulness attribute, which reduces the risk of choosing wrong data. Fig. 3 and Fig. 4 show the efficiency of our method.

3. Preliminaries

Our method is described under the setting of the semi-supervised image classification, in which a training set $\mathcal{D}$ is consist of L labeled examples $\{x_i, y_i\}_{i=1}^{L} \in \mathcal{L}$ and U unlabeled examples $\{x_i\}_{i=1}^{U} \in \mathcal{U}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{1 \dots K\}$. Here K is the number of different classes. For each example $x_i \in \mathcal{D}$, let $\tilde{y}_i = \arg\max_j f_\theta\left(x_i\right)_j$ where $f_\theta\left(x_i\right)$ is the output of network for the i-th example.

The goal of SSL is to train a classifier with $\mathcal{L}$ and $\mathcal{U}$, which can be written as

$$L_\theta = \sum_{i=1}^{L} L_s\left(f_\theta\left(x_i\right), y_i\right) + \lambda L_u(\theta, \mathcal{L}, \mathcal{U}), \qquad (1)$$

where L_s is the supervised part and L_u is the unsupervised part. A common choice of L_s is cross-entropy, and L_u is different based on different assumptions.

To utilize the information in the data structure, we can construct a graph in the embedding space, and use the graph to guide the learning of discriminative features with the help of metric learning. Suppose given a graph G with nodes representing the data and edges representing the consistency of labels between nodes, contrastive embedding can be learned using the following loss,

$$\begin{aligned} L_u(\theta, \mathcal{L}, \mathcal{U}) = \sum_{\{i,j\}_{e_{ij}=1}} & \left[D_{ij} - m^{+}\right]_{+}^{2} + \\ \sum_{\{i,j\}_{e_{ij}=0}} & \left[m^{-} - D_{ij}\right]_{+}^{2} . \end{aligned} \qquad (2)$$

Here, $[.]_+$ is the ReLU function. $e_{ij} \in E$ where E is the set of edges. The values of edges are 0 or 1, where 0 and 1 represent 'dissimilarity' and 'similarity' respectively. D_{ij} represents the distance between node x_i and x_j from the set of nodes V. m^+ and m^- are hyper-parameters. In order to learn discriminative features, e_{ij} is set to 1 if x_i and x_j come from the same class. Otherwise, e_{ij} is set to 0. For unlabeled data, the predictions from the model are used in this process, thereby may introducing unreliable edges if the predictions are incorrect.

To reduce negative effects of this unreliable graph, previous methods like Luo *et al.* [19] have tried to build a sub-graph with randomly sampled edges from G, but they still didn't consider the reliability and usefulness of E, thus suffering the risk of learning the incorrect contrastive embedding or learning inefficiently.

4. Our Method

Under the settings of SSL, the accuracy of predictions for unlabeled nodes differs in different steps of training. Since the values of edges are depended on the labels of labeled nodes and predictions of unlabeled nodes, reliable edges are also changing in the training. Therefore, our target is to build different 'dynamic' reliable graphs at different training steps. In this section, we describe the proposed Reliable Edge Mining (REM) for constructing such a 'dynamic' graph in detail, which is formalized by answering the following questions: (1) how to measure the reliability of the edges? (2) does all reliable edges attribute to the training? and (3) how to train the model with the help of the graph? The overall algorithm is illustrated in Alg. 1.

4.1. Measuring the Reliability of the Edges

We define a reliable edge as an edge with high certainty about its value. Since the values of edges represent the 'dissimilarity' and 'similarity' between nodes, the certainties of edges depend on that of nodes.

There have been many methods on calculating the certainties of the nodes. Liu *et al.* [18] builds a reliable classifier which outputs the labels as well as the corresponding certainties. However, the algorithm needs a generative model as well as a discrimination model, thereby introducing more parameters and training time. Temperature scaling is effective at calibrating predictions [8], but it is a post-processing method, which doesn't match our need for constructing a dynamic reliable graph during training.

In the proposed method, the certainties of nodes are expected to be related to the network predictions. Previous methods, like Pseudo-Labeling [16], use a threshold to filter the data with high probabilities. While high network probability does not guarantee correctness, there is a positive correlation between probability and correctness [5]. We consider the entropy of the model outputs as a measure of certainty. Given the softmax output s_i of node x_i, the calculation of its certainty q_i can be formally defined as:

$$q_i = 1 - \frac{\mathrm{H}(s_i)}{\log(K)}, \tag{3}$$

where $\mathrm{H}(\cdot)$ is the entropy function and K is the number of classes. Given the certainties of nodes, we are able to measure the reliabilities of edges. We define the certainty of an edge e_{ij} as C_{ij}, which depends on the certainty values of x_i and x_j:

$$C_{ij} = \frac{q_i + q_j}{2}. \tag{4}$$

If only considering reliable edges, we are able to obtain a pure reliable sub-graph from the original graph. However, we argue that this kind of sub-graph contributes little to the training. According to Eq. (2), a useful edge e_{ij} is an edge whose D_{ij} is larger than m^+ if $e_{ij} = 1$ or smaller than m^- if $e_{ij} = 0$. Those who don't meet the requirements are useless for the training. When we construct a sub-graph only considering the attribute of reliability, the edges are likely to be useless since the nodes are well-trained (see Fig. 3 for detail).

4.2. Mining Edges from the Graph

To mine reliable and useful edges from the original graph, we prefer the edges connecting the nodes which are not well-trained. Specially, an attribute U_{ij} representing the usefulness of the edge e_{ij} is calculated as follows:

$$U_{ij} = e^{\mathcal{I}(\tilde{y_i}=\tilde{y_j})\cdot D_{ij}}, \tag{5}$$

where $\mathcal{I}(\cdot)$ is a indicator function that takes on a value of 1 if its augment is true, and -1 otherwise. In practice, we use the Euclidean distance to compute D_{ij}.

Now each edge is attached with the attributes of certainty and usefulness. For each node x_i, according to the attribute of usefulness, we firstly select the top k_i^+ most useful edges from the list $\{e_{ij} | e_{ij} = 1, j = 1, 2, ..., L + U\}$ to form the neighbor candidate set P_i, and select the top k_i^- most useful edges from the list $\{e_{ij} | e_{ij} = 0, j = 1, 2, ..., L + U\}$ to form the non-neighbor candidate set N_i. Then we sample one edge from P_i and another edge from N_i according to the attributes of certainty. These two edges and the corresponding nodes are added to the sub-graph. With repeating this process for each node on the original graph, the reliable graph is constructed, which maintains the nodes of the original graph and contains less edges. In practice, k_i^+ and k_i^- are set as:

$$\begin{aligned} k_i^+ &= \sum_{j, e_{ij}=1} [D_{ij} > m^+], \\ k_i^- &= \sum_{j, e_{ij}=0} [D_{ij} < m^-], \end{aligned} \tag{6}$$

where $[\cdot]$ is the Iverson bracket that takes on a value of 1 if its augment is true, and 0 otherwise.

To avoid over-sampling the edges and nodes with high certainties, we decay the certainty of the nodes every time the corresponding edges are added to the reliable sub-graph. Since the value of the certainty is between 0 and 1, we simply use the square function to decay the certainty.

4.3. Guiding the Model with the Graph

The reliable graph is 'joint-trained' with the model. At the start of each iteration, the reliable graph is constructed according to Alg. 1. Then, for nodes connected by the edges with value 1, we force the feature extractor to output 'similar' features. For nodes connected by the edges with value 0, we force the extractor to output 'dissimilar' features. This can be achieved by the loss function Eq. (2). Given the class-separable features, the classifier is expected

Algorithm 1 Generation of a Graph with REM

```
Require: $G$ = the original graph
Require: $h_i$ = the feature of node $x_i$ in $\mathcal{X}$
Require: $s_i$ = the softmax output of node $x_i$ in $\mathcal{X}$
Require: $K$ = the number of different classes
for each node $x_i$ do
    Compute $q_i$ according to Eq. (3) given $s_i$ and $K$
end for
$G_s = (V_s, E_s)$
for each node $x_i$ do
    Calculate the certainty of all edges connected to $x_i$ in $G$ according to Eq. (4)
    Calculate the usefulness of all edges connected to $x_i$ in $G$ according to Eq. (5)
    Form the neighbor candidate set $P_i$ and non-neighbor candidate set $N_i$ according to the attribute of usefulness
    Sample an edge $e_{ij}$ from $P_i$ and an edge $e_{ik}$ from $N_i$ according to the attribute of certainty
    Add $e_{ij}$ and $e_{ik}$ to $E_s$
    Add $x_i$, $x_j$ and $x_k$ to $V_s$
    Decay $q_i$, $q_j$ and $q_k$
end for
return $G_s$
```

to be trained easier, thus providing more reliable edges to the graph. Guided by the graph with more reliable edges, the feature extractor learns better, and provides the classifier with more class-separable features.

Consider the synthetic example in Fig. 1. REM firstly calculates the attributes of certainty and usefulness for each edge. Then for each node x_i in the original graph, k_i^+ edges and k_i^- edges are selected to form P_i and N_i according to their usefulness attributes, from which e_{ij} and e_{ik} are sampled. The edges and the corresponding nodes are added to the sub-graph. As we construct the sub-graph per mini-batch, the sub-graph is memory-saving.

5. Experiments

To verify the efficiency of REM, a set of experiments are conducted in this section. Specifically, we firstly compare REM with recently competitive algorithms, especially the previous teacher graph based method named SNTG [19], on the widely adopted semi-supervised learning benchmarks, Then, we visualize trained pairs and discriminative features to prove the reliability and usefulness of the graph. We highlight the data-efficiency of our method by comparing REM with SNTG on several benchmarks. Finally, we show that the model guided by the reliable graph is confident in its output.

5.1. Setup

REM is evaluated on the popularly used SVHN, CIFAR-10 and CIFAR-100 datasets. In most of our experiments, we use a standard network architecture (13-layer convolutional neural network), which has been adopted as a benchmark architecture in previous methods [14, 31, 19]. We use a softmax function with certainty as input to sample two edges from the candidates for each node. m^+ and m^- in Eq. (2) are set to be 0 and 1, respectively. The other hyper-parameters are kept to be the same as previous methods.

5.2. Comparison to Other Methods

Previous advanced perturbation-based methods, including Π-model [14], temporal ensembling (Tempens) model [14] and Mean Teacher [31], are used as the baseline methods for comparison. Π-model and Tempens generate teacher predictions based on perturbed models. Mean Teacher averages model weights to get a teacher model, from which the teacher predictions are obtained to guide the student model. As these methods only enforce smoothness on each single example, we naturally wonder if we can combine them with REM. We also compare REM with a previous method named SNTG [19], which can be seen as a 'random' version of REM. In detail, SNTG randomly selects edges from the original graph, while REM selects useful and reliable edges according to the attributes of edges.

We randomly sample 250, 500, 1000 labels for SVHN, 1000, 2000, 4000 labels for CIFAR-10 and 10000 labels for CIFAR-100, respectively. Table 1 and 3 show the results reported by averaging over 10 runs. The results of SNTG using the same seeds as REM are also reported (marked with *). Guided by the reliable graph, the test error rate of perturbation-based methods are reduced by a large margin, like from 56.57% to 38.30% using Π model on CIFAR-100. Besides, REM surpasses SNTG on most benchmarks. For instance, the test error rates are reduced from 21.23% to 18.64% and from 39.07% to 35.44% using Π model on CIFAR-10 with 1000 labels and CIFAR-100 with 10000 labels. CIFAR-100 is a more difficult task containing 100 classes, and our method can still make significant improvements. It suggests that the reliable graph constructed by our method really helps to improve the generalization performance of the model. Note that when the data are all labeled (CIFAR-100 all labels with/without augmentation), REM still surpasses SNTG. We explain that the improvement is because the useful-edge-mining mechanism in our method.

5.3. Stronger Baselines

FastSWA [1] is a stronger baseline. Different from Mean Teacher in Sec. 5.2, FastSWA averages weights along the trajectory of SGD with a cyclical learning rate schedule. The authors reveal that this kind of ensembling can obtain a solution centered in a more flat region of the loss, resulting

Table 1: Test error rates (%) on CIFAR-100 with/without standard augmentation and CIFAR-10 with standard augmentation.

DataSet	CIFAR-100				CIFAR-10		
Model	with augmentation		without augmentation		with augmentation		
	10000 labels	all labels	10000 labels	all labels	1000 labels	2000 labels	4000 labels
Π model [14]	39.19±0.36	26.32±0.04	56.57±0.54	29.06±0.21	31.65 ± 1.20	17.57 ± 0.44	12.36 ± 0.31
Π+SNTG*	39.07±0.38	25.49±0.17	43.48±0.39	28.24±0.22	21.23 ± 1.27	14.65 ± 0.31	11.00 ± 0.13
Π+REM (**ours**)	35.44±0.23	24.68±0.18	38.30±0.38	**26.89±0.24**	18.64 ± 1.23	13.65 ± 0.33	11.09 ± 0.16
TempEns [14]	38.65±0.51	26.30±0.15	–	–	23.31 ± 1.01	15.64 ± 0.39	12.16 ± 0.24
TempEns+SNTG*	38.68±0.33	25.48±0.23	43.61±0.34	28.23±0.13	18.86 ± 1.07	13.88 ± 0.30	11.01 ± 0.20
TempEns+REM (**ours**)	35.62±0.33	**24.59±0.13**	38.77±0.30	26.96±0.19	**17.66 ± 1.13**	**13.33 ± 0.35**	10.61 ± 0.16
MT [31]	35.96±0.77	–	36.90±0.62	–	19.58 ± 1.03	14.76 ± 0.66	11.57 ± 0.31
MT+SNTG*	35.81±0.27	–	36.71±0.41	–	18.69 ± 1.38	13.79 ± 0.60	10.74 ± 0.56
MT+REM (**ours**)	**33.22±0.28**	–	**35.09±0.33**	–	18.23 ± 1.26	13.37 ± 0.53	**10.56 ± 0.20**

Table 2: Test error rates (%) on CIFAR-100 and Tiny ImageNet. CIFAR10k-aug and CIFAR10k-woaug represent training on CIFAR-100 using 10000 labels with and without augmentation. TIN10k-aug represents training on Tiny ImageNet using 10000 labels with augmentation.

Model	CIFAR10k-aug	CIFAR10k-woaug	TIN10k-aug
Supervised-only [14]	44.56±0.30	51.21±0.33	68.91±NA
LP [11]	35.92±0.47	–	–
CCN [32]	35.28±0.23	–	–
Π+FastSWA [1]	34.25±0.16	36.19±0.19	63.57±0.44
Π+FastSWA+REM (**ours**)	32.81±0.69	34.25±0.28	61.88±0.15
MT+FastSWA [1]*	33.62±0.54	35.09±0.47	64.21±NA
MT+FastSWA+SNTG [19]*	33.60±0.36	34.70±0.54	64.26±0.53
MT+FastSWA+REM (**ours**)	**31.95±0.27**	**33.73±0.56**	**61.72±0.37**

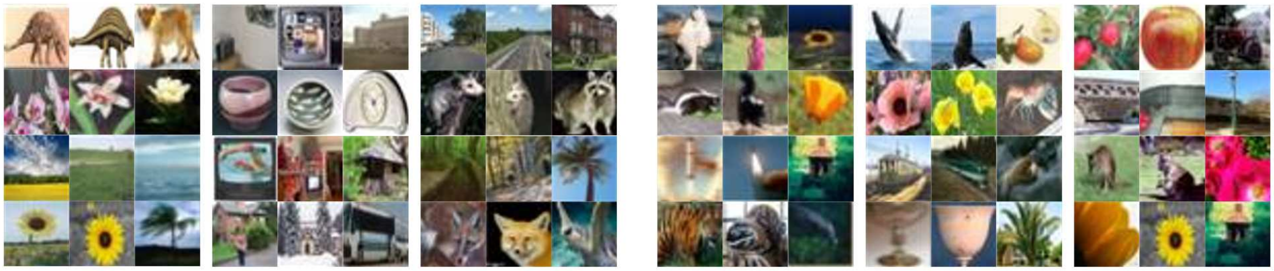
Table 3: Test error rates (%) on SVHN with standard augmentation, averaged over 10 runs.

Model	250 labels	500 labels	1000 labels
Supervised-only [31]	42.65 ± 2.68	22.08 ± 0.73	14.46 ± 0.71
TempEns [14]	12.62 ± 2.91	5.12 ± 0.13	4.42 ± 0.16
TempEns+SNTG [19]	5.36 ± 0.57	4.46 ± 0.26	3.98 ± 0.21
TempEns+REM (**ours**)	**5.07** ± **0.38**	**4.40** ± **0.29**	**3.87** ± **0.15**

in better generalization performance. However, FastSWA still doesn't utilize the information in the embedding space, motivating us to think about whether this strong baseline can be further improved with REM. To verify this, we combine FastSWA with REM and test on CIFAR-100 and Tiny ImageNet. Tiny ImageNet is a subset of ImageNet [4]. It contains 200 classes with 500 training images, 50 validation images, and 50 test images for each class, which is more challenging. On CIFAR-100, we use the same standard architecture and hyper-parameters as before. But on Tiny ImageNet, a 12-block (26-layer) Residual Network [10] with Shake-Shake regularization [6] is used following [1]. The results are reported by averaging over 3 runs on CIFAR-100 and 2 runs on Tiny ImageNet. As the comparison presented in Table 2, REM reduces the error rate from 33.62% to 31.95% and from 35.09% to 33.73% with and without augmentation on CIFAR-100, surpassing previous advanced methods like LP [11] and CCN [32]. Moreover, in spite of large number of classes in Tiny ImageNet, REM still makes significant improvements compared with SNTG (from 64.26% to 61.27%). It suggests again that the graph constructed by REM helps improve the generalization ability of the model.

5.4. Visualization

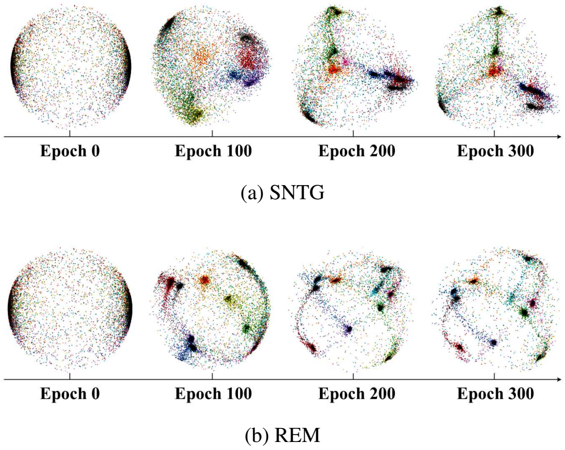
To check whether REM constructs reliable graphs, we randomly select edges from the sub-graph generated by REM and SNTG, respectively, and visualize the data connected by the edges. The experiment is done on CIFAR-100 with 10000 labeled data. As shown in Fig. 4, on one



(a) REM

(b) SNTG

Figure 4: We visualize the data connected by edges in the sub-graph of REM and SNTG on CIFAR-100. The first column represents each x_i. The second and the third column are the neighbor (x_j) and non-neighbor (x_k) of this sample. Our method is able to select more challenging and reliable pairs, while SNTG may select incorrect data for training.See Sec. 5.4 for detail.



(a) SNTG

(b) REM

Figure 5: Comparison of the 2D features generated by SNTG and REM on the CIFAR-10 test data. At epoch 100, our method is able to generate some compact clusters, while SNTG only generates divergent clusters.

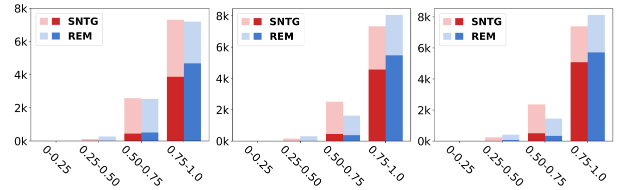
hand, REM finds more challenging and reliable edges (take the first row as example, a dinosaur can find another dinosaur as its neighbor and a lion as its non-neighbor). On the other hand, SNTG may find wrong neighbors or easy non-neighbors (take the first row as example, a flatfish finds a girl as its neighbor and a sunflower as its non-neighbor). This observation supports our analysis in Sec. 3 and Sec. 4. Furthermore, to explore whether REM guides the feature extractor to learn discriminative feature efficiently, we further visualize the last hidden layer on test data of CIFAR-10 with PCA [20]. The models of REM and SNTG are trained on CIFAR-10 with 500 labels using the same hyperparameters and training strategy. Fig. 5 demonstrates that REM encourages concentrated clusters while keeping distances between clusters after fewer epochs of training. On the contrary, the clusters are more divergent and closer to each other even after 300 epochs of training with SNTG. Therefore, REM is more efficient in learning more discriminative features.

5.5. Encouraging Confident Output of the Model

To clarify the relationships between the certainty and the correctness, we evenly split the certainty interval into four bins and add data to the corresponding bin according to their certainties. As shown in Fig. 6, for bins with high certainty, the predictions are more likely to be correct, which supports our assumption that there is a positive correlation between certainty and correctness. Moreover, when we compare SNTG and REM in Fig. 6, it can be seen that REM encourages more confident outputs than SNTG, meaning that the reliable graph encourages confident output of the model. With these two observations, we can describe REM in a joint learning way: the model first predicts some labels, based on which REM constructs a reliable sub-graph with reliable edges. In turn, the reliable graph encourages the model to give more confident outputs. As there are more reliable edges than in the original graph, REM constructs graphs with more reliable edges, which guides the model to learn better.

5.6. Effectiveness of the Attributes

To investigate whether the attributes of certainty and usefulness really contribute to the construction of a reliable graph, we compare the following methods: (1) SNTG: generating a sub-graph randomly, which can be seen as our baseline; (2) REM: generating a sub-graph considering the certainty and usefulness; (3) REM without Certainty/Usefulness: generating a sub-graph without considering the certainty or the usefulness. As shown in Fig. 2, the precision of reliable edges of REM without Certainty is much lower than SNTG, and that of REM is higher than SNTG. Similarly, Fig. 3 demonstrates that adopting the at-



(a) Epoch 100 (b) Epoch 200 (c) Epoch 300

Figure 6: Splitting the certainty interval into four bins and adding data to the corresponding bin according to their certainties. The bars with 🟦 or 🟥 represent the number of data in the interval. The bars with 🟦 or 🟥 represents the correctly predicted number of data in the interval.

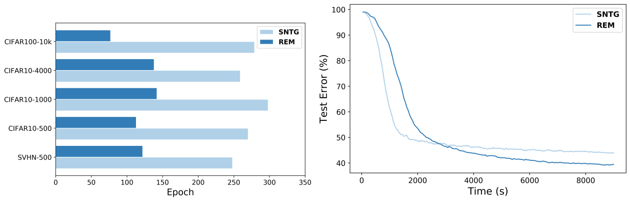
tribute of Usefulness can help REM surpasses SNTG in the ratio of useful edges mined. Therefore, we can draw a conclusion that both these two attributes of edges really contribute to the construction of a reliable graph.

5.7. Comparison on Data Utilization

As the reliable graph efficiently utilizes data and correctly guides the model, we are curious if it can help model train with seeing data less time. We study the efficiency of REM and SNTG on SVHN, CIFAR-10 and CIFAR-100 (CIFAR-10 with 500, 1000 and 4000 labels, SVHN with 500 labels, CIFAR-100 with 10000 labels, both with data augmentation). Fig. 7 shows that REM achieves the best results of SNTG with much less training iterations (about 1/3 epochs on CIFAR-100 with 10000 labels) and less time consumption, which means only changing the randomly constructed sub-graph to a reliable one can help model train more efficient.

5.8. Ablation Study

To clarify the effectiveness of the insights in REM, we compare the performances of respectively removing these components from REM. Specially, we measure the effect of (1) REM without Certainty: Randomly selecting edges from the hard neighbor candidate set and non-neighbor candidate set, which can be seen as ignoring certainty; (2) REM without Usefulness: Sampling edges considering only the attribute of certainty, which can be seen as ignoring usefulness; and (3) REM without Decaying: the certainty of the edge will not be decayed when added to the sub-graph. SNTG is used as the baseline, as it generates a sub-graph without Certainty/Usefulness/Decaying. We carry out the experiments on CIFAR-100 using 10000 labels with standard augmentation. Each result is reported by averaging over 10 runs. As shown in Table 4, each component is important for our final performance. Specially, the Certainty component and the Usefulness component can reduce the test error of SNTG by at least 2% points on CIFAR-100 with 10000 labels. And the combination of them finally reduces the result by 3% points.



(a) (b)

Figure 7: (a) Comparing the number of iterations required for SNTG and REM to achieve the same accuracy on test set. The best test accuracy of SNTG is used in the experiment. (b) Comparing the error rate obtained by SNTG and REM at the same time consumption.

Table 4: Test error rates (%) on CIFAR-100 using 10000 labels with standard augmentation, averaged over 10 runs.

Model	CIFAR-100 with 10000 labels
SNTG [19]	38.68 ± 0.33
REM without Certainty	36.69 ± 0.44
REM without Usefulness	35.98 ± 0.36
REM without Decaying	35.69 ± 0.30
REM	35.62 ± 0.33

6. Conclusion

This work explores how to build a reliable graph in the embedding space, for better guidance of the training of the model. We find that the random graphs generated by previous Teacher Graph based methods, can lead to data-inefficient training due to the wrongly tagged or useless edges. To solve the problem, we propose Reliable Edge Mining to build a reliable graph, which only contains carefully selected edges according to two attributes: reliability and usefulness. Guided by the graph, the feature extractor is able to learn discriminative feature with less iterations overs data. Our experiments demonstrate that REM utilizes data more efficiently on simple tasks like SVHN and CIFAR-10, and achieves state-of-the-art results on more difficult tasks like CIFAR-100. We also show that the model guided by the reliable graph is confident in the outputs, meaning that the method implicitly encourages the decision boundary to lie in low-density regions.

Acknowledgement

This paper is supported by the National key RD plan of the Ministry of science and technology (“Grid function expansion technology and equipment for community risk prevention”, Project No. 2018YFC0809704), and Beijing Academy of Artificial Intelligence, BAAI.

References

[1] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 5, 6

[2] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. Introduction to semi-supervised learning. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, pages 1–12. The MIT Press, 2006. 1

[3] Zihang Dai, Zhilin Yang, Fan Yang, William W. Cohen, and Ruslan Salakhutdinov. Good semi-supervised learning that requires a bad GAN. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6510–6520, 2017. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. 6

[5] Geoffrey French, Michal Mackiewicz, and Mark H. Fisher. Self-ensembling for visual domain adaptation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 4

[6] Xavier Gastaldi. Shake-shake regularization. *CoRR*, abs/1705.07485, 2017. 6

[7] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In François Denis, editor, *Actes de CAP 05, Conférence francophone sur l'apprentissage automatique - 2005, Nice, France, du 31 mai au 3 juin 2005*, pages 281–296. PUG, 2005. 2

[8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017. 4

[9] Jingrui He, Jaime G. Carbonell, and Yan Liu. Graph-based semi-supervised learning as a generative model. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2492–2497, 2007. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 2, 6

[11] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5070–5079. Computer Vision Foundation / IEEE, 2019. 6

[12] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3581–3589, 2014. 2

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012. 2

[14] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1, 5, 6

[15] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1, 2

[16] 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, page 2, 2013. 2, 4

[17] Chongxuan Li, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4088–4098, 2017. 2

[18] Xiuming Liu, Dave Zachariah, and Johan Wågberg. Robust semi-supervised learning when labels are missing at random. *CoRR*, abs/1811.10947, 2018. 4

[19] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. 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*, pages 8896–8905. IEEE Computer Society, 2018. 1, 2, 3, 5, 6, 8

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 7

[21] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993, 2019. 1, 2

[22] Mohammad Norouzi, David J. Fleet, and Ruslan Salakhutdinov. Hamming distance metric learning. In Peter L. Bartlett,

Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1070–1078, 2012. 3

[23] Augustus Odena. Semi-supervised learning with generative adversarial networks. *CoRR*, abs/1606.01583, 2016. 2

[24] Sungrae Park, Jun-Keon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3917–3924. AAAI Press, 2018. 1, 2

[25] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3546–3554, 2015. 1, 2

[26] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1163–1171, 2016. 1, 2

[27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823. IEEE Computer Society, 2015. 3

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[29] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4004–4012. IEEE Computer Society, 2016. 3

[30] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 2

[31] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 1195–1204, 2017. 5, 6

[32] Si Wu, Jichang Li, Cheng Liu, Zhiwen Yu, and Hau-San Wong. Mutual learning of complementary networks via residual correction for improving semi-supervised classification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6500–6509. Computer Vision Foundation / IEEE, 2019. 6

[33] Bing Yu, Jingfeng Wu, Jinwen Ma, and Zhanxing Zhu. Tangent-normal adversarial regularization for semi-supervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10676–10684. Computer Vision Foundation / IEEE, 2019. 1

[34] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University, 2002. 2

[35] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In Tom Fawcett and Nina Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 912–919. AAAI Press, 2003. 2

[36] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005. 1, 2