Semi-Supervised Deep Learning with Memory

Yanbei Chen[1] Xiatian Zhu[2] Shaogang Gong[1]

Queen Mary University of London[1] Vision Semantics Ltd.[2]
{yanbei.chen,s.gong}@qmul.ac.uk eddy@visionsemantics.com

Abstract. We consider the semi-supervised multi-class classification problem of learning from sparse labelled and abundant unlabelled training data. To address this problem, existing semi-supervised deep learning methods often rely on the up-to-date "network-in-training" to formulate the semi-supervised learning objective. This ignores both the discriminative feature representation and the model inference uncertainty revealed by the network in the preceding learning iterations, referred to as the *memory* of model learning. In this work, we propose a novel Memory-Assisted Deep Neural Network (MA-DNN) capable of exploiting the memory of model learning to enable semi-supervised learning. Specifically, we introduce a memory mechanism into the network training process as an *assimilation-accommodation interaction* between the network and an external memory module. Experiments demonstrate the advantages of the proposed MA-DNN model over the state-of-the-art semi-supervised deep learning methods on three image classification benchmark datasets: SVHN, CIFAR10, and CIFAR100.

Keywords: Semi-Supervised Learning, Neural Network with Memory.

1 Introduction

Semi-supervised learning (SSL) aims to boost the model performance by utilising the large amount of unlabelled data when only a limited amount of labelled data is available [4, 37]. It is motivated that unlabelled data are available at large scale but labelled data are scarce due to high labelling costs. This learning scheme is useful and beneficial for many applications such as image search [6], web-page classification [2], document retrieval [21], genomics [29], and so forth. In the SSL literature, the most straightforward SSL algorithm is self-training where the target model is incrementally trained by additional self-labelled data given by the model's own predictions with high confidence [21, 2, 25]. This method is prone to error propagation in model learning due to wrong predictions of high confidence. Other common methods include Transductive SVM [10, 3] and graph-based methods [39, 1], which, however, are likely to suffer from poor scalability to large-scale unlabelled data due to inefficient optimisation.

Recently, neural network based SSL methods [23, 35, 15, 12, 30, 24, 19, 26, 16, 9, 32] start to dominate the progress due to the powerful representation-learning ability of deep neural networks. Most of these methods typically utilise the up-to-date in-training network to formulate the additional unsupervised penalty so

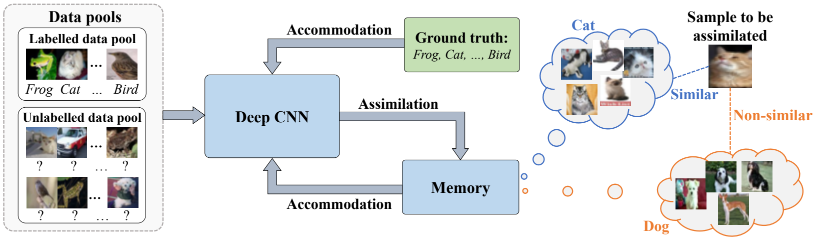

Fig. 1. Illustration of the memory-assisted semi-supervised deep learning framework that integrates a deep CNN with an external memory module trained concurrently. The memory module assimilates the incoming training data on-the-fly and generates an additional unsupervised memory loss to guide the network learning along with the standard supervised classification loss.

as to enable semi-supervised learning. We consider that this kind of deep SSL scheme is sub-optimal provided that the memorising capacity of deep networks is often incomplete and insufficiently compartmentalised to represent knowledge accrued in the past learning iterations [34]. To effectively leverage such knowledge, we introduce a memory mechanism into the deep network training process to enable semi-supervised learning from small-sized labelled and large-sized unlabelled training data. In spirit of the Piaget's theory on human's ability of *continual learning* [7], we aim to design a SSL scheme that permits the deep model to additionally learn from its memory (*assimilation*) and adjust itself to fit optimally the incoming training data (*accommodation*) in an incremental manner. To this end, we formulate a novel memory-assisted semi-supervised deep learning framework: Memory-Assisted Deep Neural Network (MA-DNN) as illustrated in Fig. 1. MA-DNN is characterised by an assimilation-accommodation interaction between the network and an external memory module.

The key to our framework design is two-aspect: (1) the class-level discriminative feature representation and the network inference uncertainty are gradually accumulated in an external memory module; (2) this memorised information is utilised to assimilate the newly incoming image samples on-the-fly and generate an informative unsupervised memory loss to guide the network learning jointly with the supervised classification loss.

Our contribution is two-fold: **(1)** We propose to exploit the *memory* of model learning to enable semi-supervised deep learning from the sparse labelled and abundant unlabelled training data, whilst fully adopting the existing end-to-end training process. This is in contrast to most existing deep SSL methods that typically ignore the memory of model learning. **(2)** We formulate a novel *Memory-Assisted Deep Neural Network* (MA-DNN) characterised by a memory mechanism. We introduce an unsupervised memory loss compatible with the standard supervised classification loss to enable semi-supervised learning. Extensive comparative experiments demonstrate the advantages of our proposed MA-DNN model over a wide variety of state-of-the-art semi-supervised deep learning methods.

2 Related Works

Semi-supervised deep learning has recently gained increasing attraction due to the strong generalisation power of deep neural networks [35, 15, 12, 30, 24, 19, 14]. A common strategy is to train the deep neural networks by simultaneously optimising a standard supervised classification loss on labelled samples along with an additional unsupervised loss term imposed on either unlabelled data [15, 27, 5] or both labelled and unlabelled data [35, 24, 19, 14]. These additional loss terms are considered as *unsupervised* supervision signals, since ground-truth label is not necessarily required to derive the loss values. For example, Lee [15] utilises the cross-entropy loss computed on the pseudo labels (the classes with the maximum predicted probability given by the up-to-date network) of unlabelled samples as an additional supervision signal. Rasmus et al. [24] adopt the reconstruction loss between one clean forward propagation and one stochastically-corrupted forward propagation derived for the same sample. Miyato et al. [19] define the distributional smoothness against local random perturbation as an unsupervised penalty. Laine et al. [14] introduce an unsupervised L_2 loss to penalise the inconsistency between the network predictions and the temporally ensembled network predictions. Overall, the rationale of these SSL algorithms is to regularise the network by enforcing smooth and consistent classification boundaries that are robust to random perturbation [24, 19]; or to enrich the supervision signals by exploiting the knowledge learned by the network, such as using the pseudo labels [15] or the temporally ensembled predictions [14].

Whilst sharing the generic spirit of introducing an unsupervised penalty, our method is unique in a number of fundamental ways: (i) Exploiting the memory of model learning: Instead of relying on the incomplete knowledge of a single up-to-date network to derive the additional loss [15], we employ a memory module to derive a memory loss based on the cumulative class-level feature representation and model inference uncertainty aggregated all through the preceding training iterations. (ii) Low computational cost: By utilising the memory mechanism, only one network forward propagation is required to compute the additional loss term for training the network, as opposed to more than one forward propagations required by other models [24, 19]. (iii) Low consumption of memory footprint: Instead of storing all the predictions of all training samples in a large mapped file [14], our online updated memory module consumes very limited memory footprint, therefore potentially more scalable to training data of larger scale.

Neural networks with memory are recently introduced to enable more powerful learning and reasoning ability for addressing several challenging tasks, such as question answering [34, 31, 18] and one-shot learning [28, 11]. Augmenting a network with an external memory component is attractive due to its flexible capability of storing, abstracting and organising the past knowledge into a structural and addressable form. As earlier works, Weston et al. [34] propose Memory Networks, which integrate inference components with a memory component that can be read and written to remember supporting facts from the past for question answering. Kaiser et al. [11] propose a life-long memory module to record network activations of rare events for one-shot learning. Our work is conceptually inspired by these works, but it is the first attempt to explore the memory mechanism in semi-supervised deep learning. Besides the basic storage functionality, our memory module induces an assimilation-accommodation interaction to exploit the memory of model learning and generate an informative unsupervised memory loss that permits semi-supervised learning.

3 Memory-Assisted Deep Neural Network

We consider semi-supervised deep learning in the context of multi-class image classification. In this context, we have access to a limited amount of labelled image samples $\mathcal{D}_L = \{(\boldsymbol{I}_{i,l}, \boldsymbol{y}_{i,l})\}_i^{n_l}$ but an abundant amount of unlabelled image samples $\mathcal{D}_U = \{(\boldsymbol{I}_{i,u})\}_i^{n_u}$, where $n_u \gg n_l$. Each unlabelled image is assumed to belong to one of the same K object categories (classes) $\mathcal{Y} = \{\boldsymbol{y}_i\}_i^K$ as the labelled data, while their ground-truth labels are not available for training. The key objective of SSL is to enhance the model performance by learning from the labelled image data $\mathcal{D}_L$ and the additional unlabelled image data $\mathcal{D}_U$ simultaneously. To that end, we formulate a memory-assisted semi-supervised deep learning framework that integrates a deep neural network with a memory module, We call this *Memory-Assisted Deep Neural Network* (MA-DNN).

3.1 Approach Overview

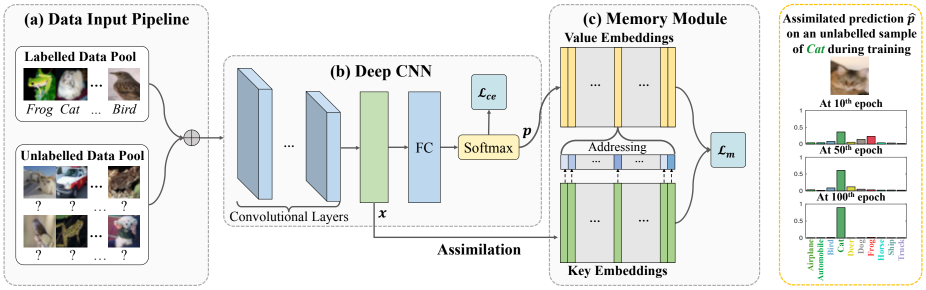

Fig. 2. An overview of Memory-Assisted Deep Neural Network (MA-DNN) for semi-supervised deep learning. During training, given **(a)** sparse labelled and abundant unlabelled training data, mini-batches of labelled/unlabelled data are feed-forward into **(b)** the deep CNN to obtain the up-to-date feature representation $\boldsymbol{x}$ and probabilistic prediction $\boldsymbol{p}$ for each sample. Given **(c)** the updated memory module, memory assimilation induces another multi-class prediction $\hat{\boldsymbol{p}}$ (Eq. (4)) for each sample via key addressing and value reading. In accommodation, a memory loss $\mathcal{L}_m$ (Eq. (7)) is computed from $\hat{\boldsymbol{p}}$ and employed as an additional supervision signal to guide the network learning jointly with the supervised classification loss. At test time, the memory module is no longer needed, so it does not affect the deployment efficiency.

The overall design of our MA-DNN architecture is depicted in Fig. 2. The proposed MA-DNN contains three parts: **(1)** A deep neural network (Section 3.2); **(2)** A memory module designed to record the memory of model learning (Section 3.3); and **(3)** An assimilation-accommodation interaction mechanism introduced for effectively exploiting the memory to facilitate the network optimisation in semi-supervised learning (Section 3.4).

3.2 Conventional Deep Neural Network

The proposed framework aims to work with existing standard deep neural networks. We select the Convolutional Neural Network (CNN) in this work due to its powerful representation-learning capability for imagery data. To train a CNN for image classification, the supervised cross-entropy loss function is usually adopted. During training, given any training sample $\boldsymbol{I}$, we feed-forward it through the up-to-date deep network to obtain a feature vector $\boldsymbol{x}$ and a multi-class probabilistic prediction vector $\boldsymbol{p}$ over all classes. Specifically, we predict the j-th class posterior probability of the labelled image sample $\boldsymbol{I}_i$ as

$$p(\boldsymbol{y}_j|\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{W}_j^\top \boldsymbol{x}_i)}{\sum_{j=1}^{|\mathcal{Y}|} \exp(\boldsymbol{W}_j^\top \boldsymbol{x}_i)} \tag{1}$$

where $\boldsymbol{x}_i$ refers to the embedded deep feature representation of $\boldsymbol{I}_i$ extracted by the deep CNN, and $\boldsymbol{W}_j$ is the j-th class prediction function parameter. The cross-entropy loss on $\boldsymbol{I}_i$ against the ground-truth class label k is computed as

$$\mathcal{L}_{\text{ce}}{=}-\sum_{j=1}^{K} \mathbb{1}[\boldsymbol{y}_j = k] \log\Big(p(\boldsymbol{y}_j|\boldsymbol{x}_i)\Big) \tag{2}$$

Obviously, the cross-entropy loss function is restricted to learn from the labelled samples alone. To take advantage of the unlabelled training samples, a straightforward way is to utilise the predicted labels given by the up-to-date model in training. This, however, may be error-prone and unreliable given immature label estimations particularly at the beginning of model training. This presents a catch-22 problem. We overcome this problem by introducing a memory module into the network training process to progressively estimate more reliable predictions on the unlabelled data.

3.3 Memory Module

To take advantage of the memorisable information generated in model learning, it is necessary for us to introduce a memory module. We consider two types of memory experienced by the network-in-training: **(1)** the class-level feature representation, and **(2)** the model inference uncertainty.

To manage these memorisable information, we construct the memory module in a key-value structure [18]. The memory module consists of multiple slots with each slot storing a symbolic pair of (*key*, *value*). In particular, the key embedding is the dynamically updated *feature representation* of each class in the feature space. Utilising an univocal representation per class is based on the assumption that deep feature embeddings of each class can be gradually learned to distribute around its cluster centroid in the feature space [33]. Based on this assumption, the global feature distribution of all classes is represented by their cluster centroids in the feature space, whilst these cluster centroids are cumulatively updated as the key embeddings in a batch-wise manner. On the other hand, the value embedding records the similarly updated *multi-class probabilistic prediction* w.r.t. each class. Hence, each value embedding is the accumulated network predictions of samples from the same class that encodes the overall model inference uncertainty at the class level.

To represent the incrementally evolving feature space and the up-to-date overall model inference uncertainty, memory update is performed every iteration to accommodate the most recent updates of the network. We only utilise the labelled data for memory update, provided that unlabelled samples have uncertainty in class assignment and hence potentially induce the risk of error propagation. Formally, suppose there exist n_j labelled image samples $\{\boldsymbol{I}_i\}$ from the j-th class ($j \in \{1, \cdots, K\}$) with their feature vectors and probabilistic predictions as $\{(\boldsymbol{x}_i, \boldsymbol{p}_i)\}_i^{n_j}$, the j-th memory slot $(\boldsymbol{k}_j, \boldsymbol{v}_j)$ is cumulatively updated over all the training iterations as follows.

$$\begin{cases} \boldsymbol{k}_j \leftarrow \boldsymbol{k}_j - \eta\nabla\boldsymbol{k}_j \\ \boldsymbol{v}_j \leftarrow \dfrac{\boldsymbol{v}_j - \eta\nabla\boldsymbol{v}_j}{\sum_{i=1}^{K}(\boldsymbol{v}_{j,i} - \eta\nabla\boldsymbol{v}_{j,i})} \end{cases} \text{with} \begin{cases} \nabla\boldsymbol{k}_j = \dfrac{\sum_{i=1}^{n_j}(\boldsymbol{k}_j - \boldsymbol{x}_i)}{1+n_j} \\ \nabla\boldsymbol{v}_j = \dfrac{\sum_{i=1}^{n_j}(\boldsymbol{v}_j - \boldsymbol{p}_i)}{1+n_j} \end{cases} \tag{3}$$

where η denotes the learning rate (set to $\eta = 0.5$ in our experiments). The value embedding $\boldsymbol{v}_j$ is normalised to ensure its probability distribution nature. Along the training process, as the gradients $(\nabla\boldsymbol{k}_j, \nabla\boldsymbol{v}_j)$ progressively get smaller, the key and value embeddings will become more reliable to reflect the underlying feature structures and multi-class distributions. To begin the training process without imposing prior knowledge, we initialise all the key and value embeddings to $\mathbf{0}$ and $\frac{1}{K} \cdot \mathbf{1}$ (a uniform probabilistic distribution over K classes), respectively. This indicates the memorised information is fully discovered by the network during training, without any specific assumption on the problem settings, therefore potentially applicable to different semi-supervised image classification tasks.

3.4 The Assimilation-Accomodation Interaction

Given the updated memory of model learning, we further employ it to enable semi-supervised deep learning. This is achieved by introducing an assimilation-accomodation interaction mechanism with two operations executed every training iteration: **(1)** *Memory Assimilation*: Compute the memory prediction for each training sample by key addressing and value reading; **(2)** *Accommodation*:

Compute the memory loss to formulate the final semi-supervised learning objective. We present the details of these operations in the following.

(1) Memory Assimilation. Given the forward propagated image representation $\boldsymbol{x}$ and network prediction $\boldsymbol{p}$ of the image $\boldsymbol{I}$, memory assimilation induces another multi-class probabilistic prediction $\hat{\boldsymbol{p}}$ based on the updated memory. We obtain this by *key addressing* and *value reading* [18]. Specifically, key addressing is to compute the addressing probability $w(\boldsymbol{m}_i|\boldsymbol{I})$, i.e., the probabilistic assignment to each memory slot $\boldsymbol{m}_i = (\boldsymbol{k}_i, \boldsymbol{v}_i)$, $i \in \{1, \cdots, K\}$, based on pairwise similarity w.r.t. each key embedding. In essence, $w(\boldsymbol{m}_i|\boldsymbol{I})$ is the cluster assignment in the feature space. Given the addressing probabilities over all K memory slots, value reading is then applied to compute the memory prediction $\hat{\boldsymbol{p}}$ by taking a weighted sum of all the value embeddings as follows.

$$\hat{\boldsymbol{p}} = \sum_{i=1}^{K} w(\boldsymbol{m}_i|\boldsymbol{I})\ \boldsymbol{v}_i \tag{4}$$

According to label availability, we adopt two addressing strategies. The first is *position-based* addressing applied to labelled training samples. Formally, suppose the training sample $\boldsymbol{I}$ is labelled as the k-th class, the addressing probability is attained based on the position k as

$$w(\boldsymbol{m}_i|\boldsymbol{I}) = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \tag{5}$$

The second is *content-based* addressing applied to unlabelled image samples. This strategy computes the addressing probability based on the pairwise similarity between the image sample $\boldsymbol{I}$ and the key embeddings $\boldsymbol{k}_i$ as

$$w(\boldsymbol{m}_i|\boldsymbol{I}) = \frac{e^{-\mathrm{dist}(\boldsymbol{x},\boldsymbol{k}_i)}}{\sum_{j=1}^{K} e^{-\mathrm{dist}(\boldsymbol{x},\boldsymbol{k}_j)}} \tag{6}$$

where $\boldsymbol{x}$ is the extracted feature vector of $\boldsymbol{I}$ and dist() denotes the Euclidean distance. Eq. (6) can be considered as a form of label propagation [38] based on the *cluster assumption* [35, 36], in the sense that the probability mass is distributed according to proximity to each cluster centroid in the feature space. That is, probabilistic assignment is computed based on cluster membership.

(2) Accommodation. This operation provides the deep network with a memory loss to formulate the final semi-supervised learning objective such that the network can learn additionally from the unlabelled data. Specifically, we introduce a *memory loss* on each training sample $\boldsymbol{x}$ as follows.

$$\mathcal{L}_m = H(\hat{\boldsymbol{p}}) + \max(\hat{\boldsymbol{p}}) D_{\mathrm{KL}}(\boldsymbol{p}||\hat{\boldsymbol{p}}) \tag{7}$$

where $H()$ refers to the entropy measure; max() is the maximum function that returns the maximal value of the input vector; $D_{KL}()$ is the Kullback-Leibler

(KL) divergence. Both $H()$ and $D_{KL}()$ can be computed without ground-truth labels and thus applicable to semi-supervised learning. The two loss terms in Eq. (7) are named as the Model Entropy (ME) loss and the Memory-Network Divergence (MND) loss, as explained below.

(i) The Model Entropy (ME) loss term $H(\hat{\boldsymbol{p}})$ is formally computed as

$$H(\hat{\boldsymbol{p}}) = -\sum\nolimits_{j=1}^{K} \hat{\boldsymbol{p}}(j) \log \ \hat{\boldsymbol{p}}(j) \tag{8}$$

which quantifies the amount of information encoded in $\hat{\boldsymbol{p}}$. From the information-theoretic perspective, the entropy reflects the overall model inference uncertainty. A high entropy on a *labelled* image sample indicates that $\hat{\boldsymbol{p}}$ is an ambiguous multimodal probability distribution, which corresponds to the retrieved value embedding of a specific class. This indicates that the network cannot well distinguish between this class and the other classes, which is resulted from assigning inconsistent probabilistic predictions to image samples within the same class. On the other hand, a high entropy on an *unlabelled* sample suggests the severe class distribution overlap between different classes in the feature space. This is because the unlabelled sample cannot be assigned to a certain class with high probability. Therefore, minimising the model entropy H is equivalent to reducing class distribution overlap in the feature space and penalising inconsistent network predictions at the class level, which is essentially motivated by the *entropy minimisation* principle [8].

(ii) The Memory-Network Divergence (MND) loss term $D_{\mathrm{KL}}(\boldsymbol{p}||\hat{\boldsymbol{p}})$ is computed between the network prediction $\boldsymbol{p}$ and the memory prediction $\hat{\boldsymbol{p}}$ as follows.

$$D_{\mathrm{KL}}(\boldsymbol{p}||\hat{\boldsymbol{p}}) = \sum\nolimits_{j=1}^{K} \boldsymbol{p}(j) \log \ \frac{\boldsymbol{p}(j)}{\hat{\boldsymbol{p}}(j)} \tag{9}$$

$D_{\mathrm{KL}}(\boldsymbol{p}||\hat{\boldsymbol{p}})$ is a non-negative penalty that measures the discrepancy between two distributions: $\boldsymbol{p}$ and $\hat{\boldsymbol{p}}$. It represents the additional information encoded in $\boldsymbol{p}$ compared to $\hat{\boldsymbol{p}}$ in information theory. Minimising this KL divergence prevents the network prediction from overly deviating from the probabilistic distribution derived from the memory module. When $D_{\mathrm{KL}}(\boldsymbol{p}||\hat{\boldsymbol{p}}) \rightarrow 0$, it indicates the network predictions match well with its memory predictions. Additionally, we also impose a dynamic weight: $\max(\hat{\boldsymbol{p}})$, the maximum probability value of $\hat{\boldsymbol{p}}$, to discount the importance of $D_{\mathrm{KL}}()$ when given an ambiguous memory prediction, i.e., a multimodal probability distribution. Hence, $\boldsymbol{p}$ is encouraged to match with $\hat{\boldsymbol{p}}$ particularly when $\hat{\boldsymbol{p}}$ corresponds to a confident memory prediction, i.e., a peaked probability distribution, where the peak corresponds to the assignment to a certain class with high probability.

The final **semi-supervised learning objective function** is formulated by merging Eq. (7) and Eq. (2) as follows.

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{m} \tag{10}$$

where λ is a hyper-parameter that is set to 1 to ensure equivalent importance of two loss terms during training.

3.5 Model Training

The proposed MA-DNN is trained by standard Stochastic Gradient Descent algorithm in an end-to-end manner. The algorithmic overview of model training is summarised in Algorithm 1.

Algorithm 1 Memory-Assisted Semi-Supervised Deep Learning.

```
Input: Labelled data $\mathcal{D}_L$ and unlabelled data $\mathcal{D}_U$.
Output: A deep CNN model for classification.
for $t = 1$ to $max\_iter$ do
    Sampling a mini-batch of labelled & unlabelled data.
    Network forward propagation (samples feed-forward).
    Memory update (Eq. (3)).
    Network supervised loss computation (Eq. (2)).
    Memory assimilation (Eq. (4)) and accommodation (Eq. (7)).
    Network update by back-propagation (Eq. (10)).
end for
```

4 Experiments

We validate the effectiveness of MA-DNN on three widely adopted image classification benchmark datasets, with comparison to other state-of-the-art methods in Section 4.2 and ablation studies in Section 4.2.

4.1 Evaluation on Semi-Supervised Classification Benchmarks

Datasets. To evaluate our proposed MA-DNN, we select three widely adopted image classification benchmark datasets as detailed in the following.
(1) SVHN [20]: A Street View House Numbers dataset including 10 classes (0∼9) of coloured digit images from Google Street View. The classification task is to recognise the central digit of each image. We use the format-2 version that provides cropped images sized at 32×32, and the standard 73,257/26,032 training/test data split.
(2) CIFAR10 [13]: A natural images dataset containing 50,000/10,000 training/test image samples from 10 object classes. Each class has 6,000 images with size 32×32.
(3) CIFAR100 [13]: A dataset (with same image size as CIFAR10) containing 50,000/10,000 training/test images from 100 more fine-grained classes with subtle inter-class visual discrepancy.
Experimental Protocol. Following the standard semi-supervised classification protocol [12, 24, 30, 19], we randomly divide the training data into a small labelled set and a large unlabelled set. The number of labelled training images is 1,000/4,000/10,000 on SVHN/CIFAR10/CIFAR100 respectively, with the remaining 72,257/46,000/40,000 images as unlabelled training data. We adopt the

Table 1. Evaluation on semi-supervised image classification benchmarks in comparison to state-of-the-art methods. **Metric**: Error rate (%) ± standard deviation, **lower is better**. "–" indicates no reported result. "*" indicates generative models.

Methods	SVHN [20]	CIFAR10 [13]	CIFAR100 [13]
DGM* [12]	36.02 ± 0.10	–	–
Γ-model [24]	–	20.40 ± 0.47	–
CatGAN* [30]	–	19.58 ± 0.58	–
VAT [19]	24.63	–	–
ADGM* [16]	22.86	–	–
SDGM* [16]	16.61 ± 0.24	–	–
ImpGAN* [27]	8.11 ± 1.3	18.63 ± 2.32	–
ALI* [5]	7.42 ± 0.65	17.99 ±1.62	–
Π-model [14]	4.82 ± 0.17	12.36 ± 0.31	39.19 ± 0.36
Temporal Ensembling [14]	4.42 ± 0.16	12.16 ± 0.24	37.34 ± 0.44
Mean Teacher [32]	**3.95 ± 0.19**	12.31 ± 0.28	–
MA-DNN (Ours)	4.21 ± 0.12	**11.91 ± 0.22**	**34.51 ± 0.61**

common classification *error rate* as model performance measure, and report the average error rate over 10 random data splits.

Implementation Details. We adopt the same 10-layers CNN architecture as [14]. More implementation details are given in the supplementary material.

Comparison with State-of-the-art Methods. In Table 1, we compare our model to 11 state-of-the-art competitive methods with their reported results on SVHN, CIFAR10 and CIFAR100. Among all these methods, Mean Teacher is the only one that slightly outperforms our MA-DNN on the digit classification task. On the natural image classification tasks, our MA-DNN surpasses the best alternative (Temporal Ensembling) with a margin of 0.25%(12.16-11.91) and 2.83%(37.34-34.51) on CIFAR10 and CIFAR100 respectively. This indicates the performance superiority of the proposed MA-DNN in semi-supervised deep learning among various competitive semi-supervised learning algorithms. Additionally, it can also be observed that MA-DNN outperforms more significantly on the more challenging dataset CIFAR100 with more fine-grained semantic structures among more classes. This suggests that the memory loss derived from the memory of model learning can enhance more fine-grained class discrimination and separation to facilitate better semi-supervised learning. Therefore, MA-DNN is potentially more scalable than the other competitors on the image classification tasks that involve a larger number of classes.

Computational Costs. The per-batch distance computation complexity induced by memory assimilation and memory update is $\mathcal{O}\big(N_u K\big)$ and $\mathcal{O}\big(N_l\big)$ respectively, where K is the number of memory slots, N_l , N_u are the numbers of labelled and unlabelled samples in each mini-batch. For computational efficiency, all the memory operations are implemented as simple matrix manipulation on GPU with single floating point precision. Overall, MA-DNN is computationally efficient in a number of ways: (i) Only one network forward propagation is required to compute the additional supervision signal, as opposed to more than one

Table 2. Evaluation on the effect of individual memory loss terms. **Metric**: Error rate (%) ± standard deviation, **lower is better**. ME: Model Entropy; MND: Memory-Network Divergence.

Methods	SVHN [20]	CIFAR10 [13]	CIFAR100 [13]
Full (ME+MND)	**4.21 ± 0.12**	**11.91 ± 0.22**	**34.51 ± 0.61**
W/O ME	4.59 ± 0.11	12.63 ± 0.26	39.93 ± 0.34
W/O MND	6.75 ± 0.40	17.41 ± 0.15	41.90 ± 0.39

forward propagations required by Γ-model, VAT, Π-model and Mean-Teacher. (ii) The consumption of memory footprint is limited. The memory size of the memory module in MA-DNN is only proportional to the number of classes; while Temporal Ensembling requires to store the predictions of all samples in a large mapped file with a memory size proportional to the number of training samples. (iii) Unlike generative models including DGM, CatGAN, ADGM, SDGM, ImpGAN, and ALI, our MA-DNN does not need to generate additional synthetic images during training, therefore resulting in more efficient model training.

4.2 Ablation Studies and Further Analysis

Effect of the Memory Loss. We evaluate the individual contribution of two loss terms in the memory loss formulation (Eq. (7)): (1) the Model Entropy (ME) (Eq. (8)), and (2) the Memory-Network Divergence (MND) (Eq. (9)). We measure the impact of each loss term by the *performance drop* when removing it from the memory loss formulation. Table 2 shows the evaluation results with comparison to the full memory loss formulation. We have the following observations: **(i)** Both loss terms bring positive effects to boost the model performance. The classification error rates increase when either of the two loss terms is eliminated. **(ii)** The MND term effectively enhances the model performance. Eliminating the MND term causes performance drop of 2.54%(6.75-4.21), 5.50%(17.41-11.91), 7.39%(41.90-34.51) on SVHN, CIFAR10, and CIFAR100 respectively. This indicates the effectiveness of encouraging the network predictions to be consistent with reliable memory predictions derived from the memory of model learning. **(iii)** The ME term is also effective. Eliminating the ME term causes performance drop of 0.38%(4.59-4.21), 0.72 %(12.63-11.91), 5.42%(39.93-34.51) on SVHN, CIFAR10, and CIFAR100 respectively. This suggests the benefit of penalising class distribution overlap and enhancing class separation, especially when the amount of classes increase – more classes are harder to be separated. Overall, the evaluation in Table 2 demonstrates the complementary joint benefits of the two loss terms to improve the model performance in semi-supervised deep learning.

Labelled Training Sample Size. We evaluate the robustness of MA-DNN over varying numbers of labelled training samples. We conduct this evaluation on SVHN by varying the number of labelled samples from 73,257 (all training samples are labelled) to 250. As comparison, we adopt the supervised counterpart *CNN-Supervised* trained only using the same labelled data without the

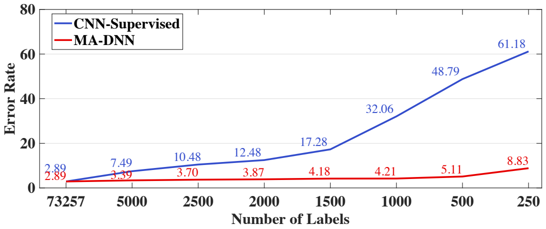

Fig. 3. Evaluation on the robustness of the MA-DNN on varying number of labelled training samples. **Metric**: Error rate, **lower is better**.

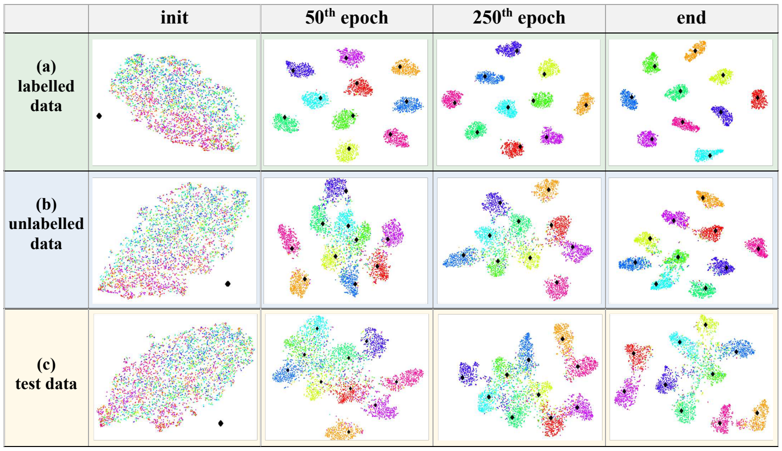

Fig. 4. Visualisation on the evolution of key embeddings (denoted as the *black dots*) and the multi-class data distribution (denoted as dots in colours) of **(a)** labelled data, **(b)** unlabelled data, **(c)** test data from CIFAR10 in the feature space during training. Data projection in 2-D space is attained by tSNE [17] based on the feature representation extracted on the *same* sets of data using the CNN at different training stages.

memory module. Fig. 3 shows that as the size of labelled data decreases, the model performance of CNN-Supervised drops from 61.18% (given 73,257 labelled samples) to 2.89% (given 250 labelled samples), with a total performance drop of 58.29% in error rate. In contrast, the performance of MA-DNN degrades only by 5.94%(8.83-2.89). This indicates the proposed MA-DNN can effectively leverage additional unlabelled data to boost the model performance when both small-sized labelled and large-sized unlabelled training data are provided.

Evolution of the Memory Module. As aforementioned, the two types of class-level memorisable information recorded in the memory module is **(1)** the class-level feature representation (key embeddings), and **(2)** the model inference uncertainty (value embeddings). To understand how the memory module is updated during training, we visualise the evolution of the key embeddings and value embeddings in Fig. 4, 5 and qualitatively analyse their effects as below.

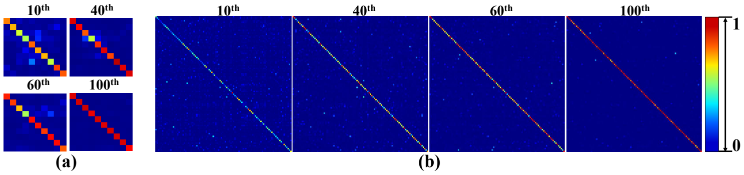

Fig. 5. Visualisation on the evolution of value embeddings on **(a)** CIFAR10 and **(b)** CIFAR100. In each block, each row corresponds to a per-class value embedding, i.e., a multi-class probabilistic prediction that encodes the class-level network inference uncertainty at different epochs during training.

Effect of the Key Embeddings. As Fig. 4 shows, the key embeddings (denoted as the *black dots*) are essentially updated as the cluster centroids to capture the global manifold structure in the feature space. In particular, we have the following observations: **(i)** Fig. 4(a) shows that although the key embeddings are initialised as **0** without imposing prior knowledge, they are consistently updated to capture the underlying global manifold structure of the labelled data in the projected 2-D feature space, as seen at the $50/250^{th}$ epochs. **(ii)** Fig. 4(b) shows that there is severe class distribution overlap of the unlabelled data initially; however, such class distribution overlap tends to be gradually mitigated as the model is trained. **(iii)** Fig. 4(c) shows that the key embeddings also roughly capture the global manifold structure of the unseen test data, even though the network is not optimised to fit towards the test data distribution. Overall, these observations are in line with our motivation of recording the accumulatively updated cluster centroids as the key embeddings for deriving the probabilistic assignment on unlabelled samples based on the *cluster assumption*. Moreover, the evolution of unlabelled data distribution in Fig. 4(b) also qualitatively suggests that our memory loss serves to penalise the class distribution overlap and render the class decision boundaries to lie in the low density region. Note that the 2-D tSNE visualisation of high-dimensional data may not perfectly reflect the underlying structure of how classes are separated in the feature space.

Effect of the Value Embeddings. As Fig. 5 shows, the value embeddings essentially record the model inference uncertainty at the class level. At the initial training stages, the value embeddings reflect much higher inference uncertainty (multimodal distribution with higher entropy), but progressively reflect much lower inference uncertainty (peaked distribution with lower entropy) as the model is progressively trained. In fact, when removing the value embeddings, the probabilistic assignment on unlabelled samples can become particularly unreliable at the earlier training stages, which even leads to performance drops of 0.69/1.94/2.78% on SVHN/CIFAR10/CIFAR100 as verified in our experiments. Hence, the value embeddings can serve to reflect the class separation in the label space, and be utilised to smooth the probabilistic assignment with model inference uncertainty for deriving more reliable memory predictions.

Evolution of Memory Predictions. We visualise the evolution of memory predictions on the unlabelled samples from CIFAR10 at different training stages

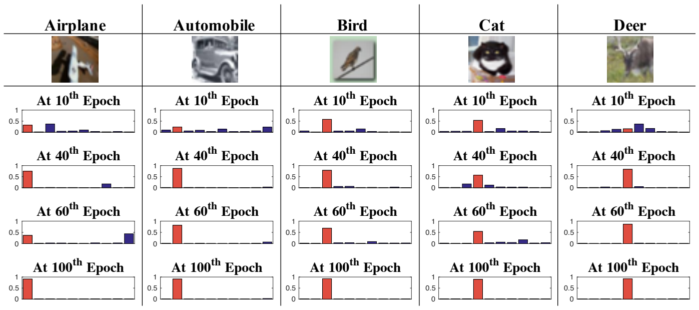

Fig. 6. Evolution of memory predictions of randomly selected *unlabelled* samples from CIFAR10. The *Red* bar corresponds to the *missing* ground-truth class.

in Fig. 6. It can be observed that the memory predictions are progressively improving from more uncertain (ambiguous) to more confident on the unlabelled training samples. This not only demonstrates the good convergence property of the MA-DNN, but also indicates how the memory loss takes effect in model learning – (1) penalising class distribution overlap when given uncertain memory predictions at the earlier training stages while (2) encouraging the network predictions to be consistent with confident memory predictions, such that the unlabelled data is fitted optimally towards the underlying manifold structure.

5 Conclusions

In this work, we present a novel Memory-Assisted Deep Neural Network (MA-DNN) to enable semi-supervised deep learning on sparsely labelled and abundant unlabelled training data. The MA-DNN is established on the idea of exploiting the memory of model learning to more reliably and effectively learn from the unlabelled training data. In particular, we formulate a novel assimilation-accommodation interaction between the network and an external memory module capable of facilitating more effective semi-supervised deep learning by imposing a memory loss derived from the incrementally updated memory module. Extensive comparative evaluations on three semi-supervised image classification benchmark datasets validate the advantages of the proposed MA-DNN over a wide range of state-of-the-art methods. We also provide detailed ablation studies and further analysis to give insights on the model design and performance gains.

Acknowledgements

This work was partly supported by the China Scholarship Council, Vision Semantics Limited, the Royal Society Newton Advanced Fellowship Programme (NA150459) and Innovate UK Industrial Challenge Project on Developing and Commercialising Intelligent Video Analytics Solutions for Public Safety (98111-571149).

References

1. Blum, A., Lafferty, J., Rwebangira, M.R., Reddy, R.: Semi-supervised learning using randomized mincuts. In: International Conference on Machine Learning (2004)
2. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory. ACM (1998)
3. Chapelle, O., Zien, A., Ghahramani, C.Z., et al.: Semi-supervised classification by low density separation. In: Tenth International Workshop on Artificial Intelligence and Statistics (2005)
4. Chapelle, O., Schlkopf, B., Zien, A.: Semi-supervised learning. The MIT Press (2010)
5. Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., Courville, A.: Adversarially learned inference. In: International Conference on Learning Representation (2017)
6. Fergus, R., Weiss, Y., Torralba, A.: Semi-supervised learning in gigantic image collections. In: Advances in Neural Information Processing Systems (2009)
7. Ginsburg, H.P., Opper, S.: Piaget's theory of intellectual development. Prentice-Hall, Inc (1988)
8. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in Neural Information Processing Systems (2005)
9. Haeusser, P., Mordvintsev, A., Cremers, D.: Learning by association-a versatile semi-supervised training method for neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
10. Joachims, T.: Transductive inference for text classification using support vector machines. In: International Conference on Machine Learning (1999)
11. Kaiser, Ł., Nachum, O., Roy, A., Bengio, S.: Learning to remember rare events. In: International Conference on Learning Representation (2017)
12. Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M.: Semi-supervised learning with deep generative models. In: Advances in Neural Information Processing Systems (2014)
13. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
14. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: International Conference on Learning Representation (2017)
15. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: ICML Workshop on Challenges in Representation Learning (2013)
16. Maaløe, L., Sønderby, C.K., Sønderby, S.K., Winther, O.: Auxiliary deep generative models. In: International Conference on Machine Learning (2016)
17. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. The Journal of Machine Learning Research (2008)
18. Miller, A., Fisch, A., Dodge, J., Karimi, A.H., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (2016)
19. Miyato, T., Maeda, S.i., Koyama, M., Nakae, K., Ishii, S.: Distributional smoothing with virtual adversarial training. In: International Conference on Learning Representation (2016)
20. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning (2011)

21. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Proceedings of the ninth international conference on Information and knowledge management. ACM (2000)
22. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. In: International Conference on Learning Representation (2017)
23. Ranzato, M., Szummer, M.: Semi-supervised learning of compact document representations with deep networks. In: International Conference on Machine Learning (2008)
24. Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T.: Semi-supervised learning with ladder networks. In: Advances in Neural Information Processing Systems (2015)
25. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised self-training of object detection models. In: Seventh IEEE Workshop on Applications of Computer Vision. Citeseer (2005)
26. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: Advances in Neural Information Processing Systems (2016)
27. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems (2016)
28. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: International Conference on Machine Learning (2016)
29. Shi, M., Zhang, B.: Semi-supervised learning improves gene expression-based prediction of cancer recurrence. Bioinformatics **27**(21) (2011)
30. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. In: International Conference on Learning Representation (2016)
31. Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: Advances in Neural Information Processing Systems (2015)
32. Tarvainen, A., 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 (2017)
33. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision (2016)
34. Weston, J., Chopra, S., Bordes, A.: Memory networks. In: International Conference on Learning Representation (2014)
35. Weston, J., Ratle, F., Mobahi, H., Collobert, R.: Deep learning via semi-supervised embedding. In: International Conference on Machine Learning (2008)
36. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems (2004)
37. Zhu, X.: Semi-supervised learning literature survey. Computer Science, University of Wisconsin-Madison **2**(3), 4 (2006)
38. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University (2002)
39. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using gaussian fields and harmonic functions. In: International Conference on Machine Learning (2003)