Deep Mutual Learning

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Computer Vision and Pattern Recognition 2018

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Deep Mutual Learning

Ying Zhang\textsuperscript{1,2}, Tao Xiang\textsuperscript{2}, Timothy M. Hospedales\textsuperscript{3}, and Huchuan Lu\textsuperscript{1}

\textsuperscript{1}Dalian University of Technology, China
\textsuperscript{2}Queen Mary University of London, UK \textsuperscript{3}University of Edinburgh, UK

zyd10907@mail.dlut.edu.cn, t.xiang@qmul.ac.uk, t.hospedales@ed.ac.uk, lhchuan@dlut.edu.cn

Abstract

Model distillation is an effective and widely used technique to transfer knowledge from a teacher to a student network. The typical application is to transfer from a powerful large network or ensemble to a small network, in order to meet the low-memory or fast execution requirements. In this paper, we present a deep mutual learning (DML) strategy. Different from the one-way transfer between a static pre-defined teacher and a student in model distillation, with DML, an ensemble of students learn collaboratively and teach each other throughout the training process. Our experiments show that a variety of network architectures benefit from mutual learning and achieve compelling results on both category and instance recognition tasks. Surprisingly, it is revealed that no prior powerful teacher network is necessary – mutual learning of a collection of simple student networks works, and moreover outperforms distillation from a more powerful yet static teacher.

1. Introduction

Deep neural networks achieve state of the art performance on many problems, but are often very large in depth and/or width, and contain large numbers of parameters \[7, 28\]. This has the drawback that they may be slow to execute or large to store, limiting their use in applications or platforms with low memory or fast execution requirements, e.g., mobile phones. This has led to a rapid growth of research in developing smaller and faster models. Achieving compact yet accurate models has been approached in a variety of ways including explicit frugal architecture design \[9\], model compression \[22\], pruning \[14\], binarisation \[18\] and most interestingly model distillation \[8\].

Distillation-based model compression relates to the observation \[3\] that small networks often have the same representation capacity as large networks; but compared to large networks they are simply harder to train and find the right parameters that realise the desired function. That is, the limitation seems to lie in the difficulty of optimisation rather than in the network size \[1\]. To better learn a small network, the distillation approach starts with a powerful (deep and/or wide) teacher network (or network ensemble), and then trains a smaller student network to mimic the teacher \[8, 1, 16, 3\]. Mimicking the teacher’s class probabilities \[8\] and/or feature representation \[1, 19\] conveys additional information beyond the conventional supervised learning target. The optimisation problem of learning to mimic the teacher turns out to be easier than learning the target function directly, and the student can match or even outperform \[19\] the much larger teacher.

In this paper we aim to solve the same problem of learning small but powerful deep neural networks. However, we explore a different but related idea to model distillation – that of mutual learning. Distillation starts with a powerful large and pre-trained teacher network and performs one-way knowledge transfer to a small untrained student. In contrast, in mutual learning we start with a pool of untrained students who simultaneously learn to solve the task together. Specifically, each student is trained with two losses: a conventional supervised learning loss, and a mimicry loss that aligns each student’s class posterior with the class probabilities of other students. Trained in this way, it turns out that each student in such a peer-teaching based scenario learns significantly better than when learning alone in a conventional supervised learning scenario. Moreover mutually learned student networks achieve better results than students trained by conventional distillation from a larger pre-trained teacher. Furthermore, while the conventional understanding of distillation requires a teacher larger and more powerful than the intended student, it turns out that in many cases mutual learning of several large networks also improves performance compared to independent learning. This makes the deep mutual learning strategy generally applicable, e.g., it can also be used in application scenarios where there is no constraint on the model size and the recognition accuracy is the only concern.

It is perhaps not obvious why the proposed learning strategy should work at all. Where does the additional knowledge come from, when the learning process starts out with
all small and untrained student networks? Why does it converge to a good solution rather than being hamstrung by groupthink as ‘the blind lead the blind’. Some intuition about these questions can be gained by considering the following: Each student is primarily directed by a conventional supervised learning loss, which means that their performance generally increases and they cannot drift arbitrarily into groupthink as a cohort. With supervised learning, all networks soon predict the same (true) labels for each training instance; but since each network starts from a different initial condition, they learn different representations, and consequently their estimates of the probabilities of the next most likely classes vary. It is these secondary quantities that provide the extra information in distillation [8] as well as mutual learning. In mutual learning the student cohort effectively pools their collective estimate of the next most likely classes. Finding out – and matching – the other most likely classes for each training instance according to their peers increases each student’s posterior entropy [4], which helps them to converge to a more robust (flatter) minima with better generalisation to testing data. This is related to very recent work on the robustness of high posterior entropy solutions (network parameter settings) in deep learning [4], but with a more informed choice of alternatives than blind entropy regularisation.

Overall, mutual learning provides a simple but effective way to improve the generalisation ability of a network by training collaboratively with a cohort of other networks. Extensive experiments are carried out on both object category recognition (image classification on CIFAR100 [12]) and instance recognition problems (person re-identification on Market1501 [33]). The results show that, compared with distillation by a pre-trained static large network, collaborative learning by small peers achieves better performance. In particular, on the person re-identification task, state-of-the-art results can be obtained using a much smaller network trained with mutual learning, compared to the latest competitors. Furthermore we observe that: (i) it applies to a variety of network architectures, and to heterogeneous cohorts consisting of mixed big and small networks; (ii) The efficacy increases with the number of networks in the cohort – a nice property to have because by training on small networks only, more of them can fit on given GPU resources for more effective mutual learning; (iii) it also benefits semi-supervised learning with the mimicry loss activated both on labelled and unlabelled data. Finally, we note that while our focus is on obtaining a single effective network, the entire cohort can also be used as a highly effective ensemble model.

2. Related Work

Model Distillation The distillation-based approach to model compression has been proposed over a decade ago but was recently re-popularised by [8], where some additional intuition about why it works – due to the additional supervision and regularisation of the higher entropy soft-targets – was presented. Initially, a common application was to distill the function approximated by a powerful model/ensemble teacher into a single neural network student [3, 8]. But later, the idea has been applied to distill powerful and easy-to-train large networks into small but harder-to-train networks [19] that can even outperform their teacher. Recently, distillation has been connected more systematically to information learning theory [15] and SVM+ [25] – an intelligent teacher provides privileged information to the student. This idea of using model distillation for learning with privileged information has been exploited by Zhang et al. [29] for action recognition: the more expensive optical flow field is treated as privileged information and an optical flow CNN is used to teach a motion vector CNN. In terms of representation of the knowledge to be distilled from the teacher, existing models typically use teacher’s class probabilities [8] and/or feature representation [11, 19]. Recently, Yim et al. [27] exploited flow between layers computed as the inner product of feature maps between layers. In contrast to model distillation, we address dispensing with the teacher altogether, and allowing an ensemble of students to teach each other in mutual distillation.

Collaborative Learning Other related ideas on collaborative learning include Dual Learning [6] where two cross-lingual translation models teach each other interactively. But this only applies in this special translation problem where an unconditional within-language model is available to be used to evaluate the quality of the predictions, and ultimately provides the supervision that drives the learning process. Furthermore, in dual learning different models have different learning tasks whilst in mutual learning the tasks are identical. Recently, Cooperative Learning [2] has been proposed to learn multiple models jointly for the same task but in different domains. E.g. recognising the same set of object categories but with one model inputting RGB images and the other inputting depth images. The models communicate via object attributes which are domain invariant. Again this is different from mutual learning where all models address the same task and domain.

3. Deep Mutual Learning

3.1. Formulation

We formulate the proposed deep mutual learning (DML) approach with a cohort of two networks (see Fig. 1). Extension to more networks is straightforward (see Sec. 3.3). Given $N$ samples $X = \{x_i\}_{i=1}^N$ from $M$ classes, we denote the corresponding label set as $Y = \{y_i\}_{i=1}^N$ with $y_i \in \{1, 2, ..., M\}$. The probability of class $m$ for sample
Figure 1. Deep Mutual Learning (DML) schematic. Each network is trained with a supervised learning loss, and a Kullback Leibler Divergence based mimicry loss to match the probability estimates of its peers.

Our KL divergence based mimicry loss is asymmetric, thus different for the two networks. One can instead use a symmetric Jensen-Shannon Divergence loss:

$$\frac{1}{2}(D_{KL}(p_1||p_2) + D_{KL}(p_2||p_1)).$$

(7)

However, we found empirically that whether a symmetric or asymmetric KL loss is used does not make any difference.

3.2. Optimisation

A key difference between model distillation and DML is that in DML, the two models are optimised jointly and collaboratively, with the optimisation processes for the two models being closely intervened. The mutual learning strategy is embedded in each mini-batch based model update step for both models and throughout the whole training process. The models are learned with the same mini-batches. At each iteration, we compute the predictions of the two network’s predictions $p_1$ and $p_2$, we use the Kullback Leibler (KL) Divergence.

The KL distance from $p_1$ to $p_2$ is computed as

$$D_{KL}(p_2||p_1) = \sum_{i=1}^{N} \sum_{m=1}^{M} I(y_i, m) \log \frac{p_2^m(x_i)}{p_2^m(x_i)}.$$

(4)

The overall loss functions $L_{\Theta_1}$ and $L_{\Theta_2}$ for networks $\Theta_1$ and $\Theta_2$ respectively are thus:

$$L_{\Theta_1} = L_{C_1} + D_{KL}(p_2||p_1).$$

(5)

$$L_{\Theta_2} = L_{C_2} + D_{KL}(p_1||p_2).$$

(6)

In this way each network learns both to correctly predict the true label of training instances (supervised loss $L_C$) as well as to match the probability estimate of its peer (KL mimicry loss).
we have added the coefficient $\frac{1}{K-1}$ to make sure that the training is mainly directed by supervised learning of the true labels. The optimisation for DML with more than two networks is a straightforward extension of Algorithm 1. This learning strategy naturally suits distributed learning [21]: It can be distributed by learning each network on one device and passing the small probability vectors between devices.

With more than two networks, an interesting alternative learning strategy for DML is to take the ensemble of all the other $K-1$ networks as a single teacher to provide a combined mimicry target, which would be very similar to the distillation approach but performed at each mini-batch model update. Then the objective function of $\Theta_k$ can be written as

$$L_{\Theta_k} = L_{C_k} + D_{KL}(p_{avg}||p_k), \quad p_{avg} = \frac{1}{K-1} \sum_{l=1, l\neq k}^K p_l. \tag{11}$$

In our experiments (see Sec. 4.8), we find that this DML strategy with a single ensemble teacher (denoted DML,e) leads to worse performance than DML with $K - 1$ teachers. This is because the model averaging step (Equation (11)) to build the teacher ensemble makes the teacher’s posterior probabilities more peaked at the true class, thus reducing the posterior entropy over all classes. It is therefore contradictory to one of the objectives of DML, which is to produce robust solutions with high posterior entropy.

3.4. Extension to Semi-supervised Learning

The proposed DML extends straightforwardly to semi-supervised learning. Under the semi-supervised learning setting, we only activate the cross-entropy loss for labelled data, while computing the KL distance based mimicry loss for all the training data. This is because the KL distance computation does not require class labels, so unlabelled data can also be used. Denote the labelled and unlabelled data as $L$ and $U$, where we have $X = L \cup U$, the objective function for learning network $\Theta_1$ can be reformulated as

$$L_{\Theta_1} = L_{C_1} + D_{KL}(p_2||p_1). \tag{12}$$

4. Experiments

4.1. Datasets and Settings

Datasets Four datasets are used in our experiments. The ImageNet [20] dataset contains 1000 object classes with about 1.2 million images for training and 50,000 images for validation. The CIFAR-10 and CIFAR-100 [12] datasets consist of $32 \times 32$ color images containing objects from 10 and 100 classes respectively. Both are split into a 50,000-image train set and a 10,000-image test set. The Top-1 classification accuracy is used as evaluation metric. The Market-1501 [33] dataset is a widely used benchmark in the person re-identification (re-id) problem [5]. Different from the object category recognition problem in CIFAR, re-id is an instance recognition problem that aims to associate person identities across different non-overlapping camera views. Market-1501 contains 32,668 images of 1,501 identities captured from six camera views, with 751 identities for training and 750 identities for testing. As per state of the art approaches to re-id [5], we train the network for 751-way classification and use the resulting feature output of the last pooling layer as a representation for nearest neighbour matching at testing. For evaluation, the standard Cumulative Matching Characteristic (CMC) Rank-k accuracy and mean average precision (mAP) metrics [33] are used.

Networks The networks used in our experiments include compact networks of typical student size: Resnet-32 [7] and MobileNet [9]; as well as larger networks of typical teacher size: InceptionV1 [24] and Wide ResNet WRN-28-10 [28]. Table 1 compares the number of parameters of all the networks for CIFAR-100.

<table>
<thead>
<tr>
<th>Network</th>
<th>CIFAR-32</th>
<th>MobileNet</th>
<th>InceptionV1</th>
<th>WRN-28-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-32</td>
<td>0.5M</td>
<td>3.3M</td>
<td>7.8M</td>
<td>36.5M</td>
</tr>
</tbody>
</table>

Table 1. Number of parameters on the CIFAR-100 dataset

Implementation Details We implement all networks and training procedures in TensorFlow and conduct all experiments on an NVIDIA GeForce GTX 1080 GPU. For CIFAR-100, we follow the experimental settings of [28]. Specifically, we use SGD with Nesterov momentum and set the initial learning rate to 0.1, momentum to 0.9 and mini-batch size to 64. The learning rate dropped by 0.1 every 60 epochs and we train for 200 epochs. The data augmentation includes horizontal flips and random crops from im-
Table 2. Top-1 accuracy (%) on the CIFAR-10 and CIFAR-100 dataset. “DML-Ind” measures the difference in accuracy between the network learned with DML and the same network learned independently.

<table>
<thead>
<tr>
<th>Network Types</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Independent</td>
<td>DML</td>
</tr>
<tr>
<td>ResNet-32</td>
<td>92.47</td>
<td>92.47</td>
</tr>
<tr>
<td>WRN-28-10</td>
<td>95.01</td>
<td>92.47</td>
</tr>
<tr>
<td>MobileNet</td>
<td>93.59</td>
<td>92.47</td>
</tr>
<tr>
<td>MobileNet</td>
<td>93.59</td>
<td>93.59</td>
</tr>
<tr>
<td>WRN-28-10</td>
<td>95.01</td>
<td>93.59</td>
</tr>
<tr>
<td>WRN-28-10</td>
<td>95.01</td>
<td>95.01</td>
</tr>
</tbody>
</table>

Table 3. Comparative results on the Market-1501 dataset. Each MobileNet is trained in a two-network cohort and the averaged performance of the two networks in the cohort is reported. “pre?” indicates whether ImageNet pretraining was carried out.

<table>
<thead>
<tr>
<th>Method</th>
<th>pre?</th>
<th>Single-Query</th>
<th>Multi-Query</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>mAP</td>
<td>Rank-1</td>
</tr>
<tr>
<td>Gated S-CNN [26]</td>
<td>no</td>
<td>39.55</td>
<td>65.88</td>
</tr>
<tr>
<td>k-reciprocal [35]</td>
<td>yes</td>
<td>63.63</td>
<td>77.11</td>
</tr>
<tr>
<td>MSCAN [13]</td>
<td>yes</td>
<td>57.53</td>
<td>80.31</td>
</tr>
<tr>
<td>PDC [23]</td>
<td>no</td>
<td>63.41</td>
<td>84.14</td>
</tr>
<tr>
<td>DLPAR [32]</td>
<td>no</td>
<td>63.40</td>
<td>81.00</td>
</tr>
<tr>
<td>MobileNet</td>
<td>no</td>
<td>50.15</td>
<td>76.87</td>
</tr>
<tr>
<td>MobileNet+DML</td>
<td>no</td>
<td>54.71</td>
<td>79.12</td>
</tr>
<tr>
<td>MobileNet</td>
<td>yes</td>
<td>65.06</td>
<td>85.01</td>
</tr>
<tr>
<td>MobileNet+DML</td>
<td>yes</td>
<td>70.51</td>
<td>89.34</td>
</tr>
</tbody>
</table>

4.2. Results on CIFAR-100

Table 2 compares the Top-1 accuracy of the CIFAR-100 dataset obtained by various architectures in a two-network DML cohort. From the table we can make the following observations: (i) All the network combinations among ResNet-32, MobileNet and WRN-28-10 improve performance when learning in a cohort compared to learning independently, indicated by the all positive values in the “DML-Ind” columns. (ii) The networks with smaller capacity (ResNet-32 and MobileNet) generally benefit more from DML. (iii) Although WRN-28-10 is a much larger network than MobileNet or ResNet-32 (Table 1), it still benefits from being trained together with a smaller peer. (iv) Training a cohort of large networks (WRN-28-10) is still beneficial compared to learning them independently. Thus in contrast to the conventional wisdom of model distillation, we see that a large pre-trained teacher does not necessarily bring large benefits, and multiple large networks can still benefit from our distillation-like process.

4.4. Results on ImageNet

Figure 2(a) compares MobileNet and InceptionV1 accuracy on ImageNet with Independent and DML training. We can see that the DML variants of both architectures consistently performs better than their independently trained counterparts. These results show that DML is applicable to large-scale problems.

4.5. Distributed Training of DML

To investigate the impact of training strategy on DML, we compared two DML variants: 1) sequential: train two networks according to Algorithm 1 on one GPU; two networks are updated one after the other; 2) distributed: each network is trained on a separate GPU and CPU is used for KL divergence communication; in this way, the predictions and parameters of two networks are updated simultaneously. We experiment on Market-1501 with 2 MobileNets, a simple, small, and general-purpose network. In contrast many recently proposed deep re-id networks such as those in [31, 23, 34] have complicated and specially designed architectures to handle the drastic pose-changes and body-part mis-alignment when matching people across camera views.
and show the convergence and mAP results in Fig. 2(b). It is interesting to observe that our DML’s performance is further boosted by distributed training. Comparing these two variants, the two networks are more ‘equal’ when trained distributed as they always have exactly the same number of training iterations. This result thus suggests that the students benefit the most from the DML peer-teaching when the discrepancy in their learning progress is minimised.

### 4.6. Comparison with Model Distillation

As our method is closely related to model distillation, we next provide a focused comparison to Distillation \[8\]. Table 4 compares our DML with model distillation where the teacher network (Net 1) is pre-trained and provides fixed posterior targets for the student network (Net 2). As expected the conventional distillation approach from a powerful pre-trained teacher does indeed improve the student performance compared to independently learning the student (1 distills 2 versus Net 2 Independent).

However, the results show that training both networks together in deep mutual learning provides a clear improvement compared to distillation (1 distills 2 versus DML Net 2). This implies that in the process of mutual learning, the network that would play the role of teacher actually becomes better than a pre-trained teacher, via learning from interactions with an a-priori untrained student.

### 4.7. DML with Larger Student Cohorts

The prior experiments studied cohorts of 2 students. We next investigate how DML scales with more students in the cohort. Figure 3(a) shows the results on Market-1501 with DML training of increasing cohort sizes of MobileNets. The figure shows average mAP, as well as the standard deviation. From Fig. 3(a) we can see that the performance of the average single network increases with the number of networks in the DML cohort, hence its gap to the independently trained networks. This demonstrates that the generalisation ability of students is enhanced when learning together with increasing numbers of peers. The performance of different networks is also more consistent with larger cohort size, indicated by the smaller standard deviations.

A common technique when training multiple networks is to use them as an ensemble and make a combined prediction. In Fig. 3(b) we use the same models as Fig. 3(a) but make predictions based on the ensemble (matching based on concatenated feature of all members) instead of reporting individual network predictions as expected (Fig. 3(b) vs. (a)). Moreover, the ensemble performance also benefits from training multiple networks as a DML cohort (Fig. 3(b) DML ensemble vs. Independent ensemble).

### 4.8. How and Why does DML Work?

In this section we attempt to give some insights about how and why our deep mutual learning strategy works. There has been a wave of recent research on the subject of “Why Deep Nets Generalise” \[4\] \[10\], which have provided some insights such as: While there are often many solutions (deep network parameter settings) that generate zero training error, some of these generalise better than oth-

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Network Types</th>
<th>Independent</th>
<th>1 distills 2</th>
<th>DML</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Net 1</td>
<td>Net 2</td>
<td>Net 1</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>WRN-28-10</td>
<td>ResNet-32</td>
<td></td>
<td>78.69</td>
</tr>
<tr>
<td></td>
<td>MobilNet</td>
<td>ResNet-32</td>
<td></td>
<td>73.65</td>
</tr>
<tr>
<td>Market-1501</td>
<td>Inception V1</td>
<td>MobileNet</td>
<td></td>
<td>63.91</td>
</tr>
<tr>
<td></td>
<td>MobileNet</td>
<td>MobileNet</td>
<td></td>
<td>50.15</td>
</tr>
</tbody>
</table>

Table 4. Comparison with distillation on CIFAR-100 (Top-1 accuracy (%)) and Market-1501 dataset (mAP (%))

(a) ImageNet  
(b) Market-1501

Figure 2. (a) Results on ImageNet. It shows top-1 acc. (%) w.r.t. training steps; (b) Convergence effect and test mAP (%) on Market-1501 with sequential and distributed training.

Figure 3. Performance (mAP (%)) on Market-1501 with different cohort size.
ers due to being in wide valleys rather than narrow crevices 
[4, 10] – so that small perturbations do not change the pre-
diction efficacy drastically; and that deep networks are bet-
ter than might be expected at finding these good solutions
[30], but that the tendency towards finding robust minima
can be enhanced by biasing deep nets towards solutions
with higher posterior entropy [4, 17].

**DML Leads to Better Quality Solutions with More Ro-
 bust Minima**  With these insights in mind we make some
observations about the DML process. Firstly we note that
in our experiments, the networks typically fit the training
data perfectly: Training accuracy goes to 100% and classi-
fication loss becomes minimal (e.g., Fig. 4(a)). However, as
we saw earlier, DML performs better on test data. There-
fore rather than helping to find a better (deeper) minimum
of training loss, DML appears to be helping us to find a
wider/more robust minimum that generalises better to test
data. Inspired by [4, 10], we perform a simple test to anal-
yse the robustness of the discovered minima on CIFAR-100
using MobileNet. For the DML and independent models,
we compare the training loss of the learned models before
and after adding independent Gaussian noise with variable
standard deviation $\sigma$ to each model parameter. We see that
the depths of the two minima were the same (Fig. 4(a)), but
after adding this perturbation the training loss of the inde-
pendent model jumps up while the loss of the DML model
increases much less. This suggests that the DML model has
found a much wider minimum, which is expected to provide
better generalisation performance [4, 17].

**How is a Better Minimum Found?**  When asking each
network to match its peer’s probability estimates, mis-
matches where a given network predicts zero and its
teacher/peer predicts non-zero are heavily penalised. There-
fore the overall effect of DML is that, where each network
independently would put a small mass on a small set of
secondary probabilities, all networks in the DML tend to
aggregate their prediction of secondary probabilities, and
both (i) put more mass on the secondary probabilities alto-
gether, and (ii) place non-zero mass on more distinct sec-

DML with Ensemble Teacher  In DML, each student is
taught by all other students in the cohort individually, re-
gardless how many students are in the cohort (Eq. (10)).
In Sec. 3.3, an alternative DML strategy (DML, e) is dis-
cussed, by which each student is asked to match the pre-
dictions of the ensemble of all other students in the cohort
(Eq. (11)). One might reasonably expect this approach to be
better: As the ensemble prediction is better than individual
predictions, it should provide a cleaner and stronger teach-

<table>
<thead>
<tr>
<th>Settings</th>
<th>mAP</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent</td>
<td>50.15</td>
<td>76.87</td>
</tr>
<tr>
<td>DML</td>
<td>54.71</td>
<td>79.12</td>
</tr>
<tr>
<td>Independent, Entropy</td>
<td>50.94</td>
<td>76.34</td>
</tr>
<tr>
<td>DML, L2</td>
<td>51.01</td>
<td>77.58</td>
</tr>
</tbody>
</table>

Table 5. Single-Query results on Market-1501 under different set-
ings with MobileNets. L2 indicates decreasing the feature dis-
tance of two networks.
ing signal – more like conventional distillation. In practice
the results of ensemble rather than peer teaching are worse
(see Fig. 5). By analysing the teaching signal of the en-
semble in comparison to peer teaching, the ensemble target
is much more sharply peaked at the true label than the
peer targets, resulting in larger prediction entropy value for
DML (0.2805) than DML_e (0.1562). Thus while the noise-
averaging property of ensembling is effective for making a
correct prediction, it is actually detrimental to providing a
teaching signal where the secondary class probabilities are
the salient cue in the signal and having high-entropy poste-
rior leads to more robust solutions to model training.

4.9. Does DML Makes Models More Similar?

We know that with the same training objective, the pre-
dicted class posterior would be similar for different mod-
els in a DML cohort. The question is do these model also
produce similar features, especially when the models have
identical architecture? Figure 5 shows the t-SNE visualisa-
tion of feature distribution on the Market-1501 test set by
two MobileNets. We can see that either with or without
DML, the two MobileNets do indeed different features, in-
dicating diverse models are obtained. This helps to explain
why different models in a DML cohort can teach each other:
Each has learned something that the others have not.

We note that in a number of model distillation studies
[1, 19], a feature distance loss is added to force the student
network to produce similar features to the teacher at corre-
sponding layers. This makes sense when the teacher is pre-
trained and fixed and the student aims to imitate the teacher.
However, in DML, aligning the internal representations dif-
ferent DML models would diminish the cohort diversity and
thus damage the ability of each network to teach its peers.
Table 5 shows that indeed, when a feature L2 loss is intro-
duced, DML becomes less effective (DML vs. DML, L2).

4.10. Semi-Supervised Learning

We finally explore semi-supervised learning in CIFAR-
100 and Market-1501. For CIFAR-100 , we randomly se-
lect a subset (from 10% to 100%) of the training images per
class as labeled, and treat the rest as unlabeled. For Market-
1501, we randomly select M identities as labelled in the
training set, varying M from 100 to 751. Experiments are
performed with 3 different training strategies: 1) training
on the labelled data only with single network; 2) training on
labelled data only with DML (DML-labelled). 3) training
on all data with DML, where the classification loss is com-
puted for labelled data only, and KL loss is calculated for
all the training data (DML-all).

From the results in Figure 6, we can see that: (1) Train-
ing two networks with DML consistently performs better
than training a single network – as before, but now with
varying amounts of labeled data. (2) Compared with adding
DML to only labelled data (DML-labelled), DML-all fur-
ther improves the performance by exploiting the unlabelled
data using the KL-distance based mimicry loss. The
improvement is bigger when the percentage of labelled data is
smaller. This confirms that DML benefits both supervised
and semi-supervised learning scenarios.

5. Conclusion

We have proposed a simple and generally applicable ap-
proach to improving the performance of deep neural net-
works by training them in a cohort with peers and mutual
distillation. With this approach we can obtain compact net-
works that perform better than those distilled from a strong
but static teacher. One application of DML is to obtain
compact, fast and effective networks. We also showed that
this approach is also promising to improve the performance
of large powerful networks, and that the network cohort
trained in this manner can be combined as an ensemble to
further improve performance.

Acknowledgements. This work was supported by the Nat-
ural Science Foundation of China under Grant 61725202 ,
61472060 and China Scholarship Council (CSC) and the
European Union’s Horizon 2020 research and innovation
program (grant agreement no. 640891)
References