ONLINE DEEP METRIC LEARNING VIA MUTUAL DISTILLATION

Gao-Dong Liu[1], *Wan-Lei Zhao*[1,*], *Jie Zhao*[2]

[1]Xiamen University, gdliu@stu.xmu.edu.cn, wlzhao@xmu.edu.cn; [2]Boden.ai, jie.zhao@bodenai.com.

ABSTRACT

Deep metric learning aims to transform input data into an embedding space, where similar samples are close while dissimilar samples are far apart from each other. In practice, samples of new categories arrive incrementally, which requires the periodical augmentation of the learned model. The fine-tuning on the new categories usually leads to poor performance on the old, which is known as "catastrophic forgetting". Existing solutions either retrain the model from scratch or require the replay of old samples during the training. In this paper, a complete online deep metric learning framework is proposed based on mutual distillation for both one-task and multi-task scenarios. Different from the teacher-student framework, the proposed approach treats the old and new learning tasks with equal importance. No preference over the old or new knowledge is caused. In addition, a novel virtual feature estimation approach is proposed to recover the features assumed to be extracted by the old models. It allows the distillation between the new and the old models without the replay of old training samples or the holding of old models during the training. A comprehensive study shows the superior performance of our approach with the support of different backbones.

Index Terms— Deep metric learning, knowledge distillation, mutual learning, feature estimation, online learning

1. INTRODUCTION

Owing to the seminal learning framework from [1], deep metric learning has been successfully applied in various tasks such as face recognition [1, 2], person re-identification [3], and fine-grained image search [4, 5], etc. The research focus in recent years has been on mining hard training samples [2, 4, 6]. In most of these works, the visual categories to be trained in the training set are fixed. No mechanism is designed to allow new categories to join in the training incrementally. Only a few research works [7–10] shed light on this learning issue, which is known as "online deep metric learning", or "incremental deep metric learning". In this scenario, the trained model has to be updated to adapt to new categories on the one hand. On the other hand, it is required to maintain the performance on old categories as much as possible. Due to the well-known "catastrophic forgetting" [11], these two competing requirements are hardly balanced.

There are several possible practices for the online deep metric learning. An intuitive solution is to retrain a new model based on both the old and new categories, which is also known as joint training. The major disadvantage of this type of approaches is that it requires large working memory to store and replay the past training samples [12, 13]. Such storage and replay may not be viable in practice. For instance, the samples of old categories are no longer available for streaming data. Another way is to fine-tune the trained model with the samples of new categories only. The embedding space constructed based on old categories has been transformed to adapt new categories, which leads to the considerable degradation on old tasks. This is essentially the cause of "catastrophic forgetting".

In the literature, most of the online learning is addressed under the context of visual class categorization task. In [12], the new model is trained based on the samples from both new and old categories that are produced by a generator. It requires the replay of samples from old categories. Essentially, it is still a joint training approach based on the generative model. Recent works [14, 15] address this issue under a more stringent condition where old data is not available during the training of new tasks. Research works in [7, 8] explore the online deep metric learning under the same constraint. Although different frameworks have been proposed, they all intend to maintain the distribution of old categories the same as before in the augmented embedding space. As no old data is used, different ways have been introduced to recover the features produced by the old models. The recovered old features will facilitate the training of a new model to keep balanced performance on both old and new tasks.

In all of the aforementioned approaches [7, 8, 14, 15], in order to maintain the knowledge learned from the old tasks, the performance in the new tasks has been sacrificed. In this paper, a novel online deep metric learning approach is proposed based on mutual learning [16], where *three* models are involved to strike a balance between stability and plasticity. *Two* student models learn collaboratively throughout the training, leading to the better generalization to the new task [16]. *One* teacher model transfers previous knowledge to student models to preserve the performance on the old task. Moreover, we extend this one-stage online learning framework to the scenario of multiple stages, where new categories are allowed to join in multiple batches. In order to enable the

mutual learning of knowledge from earlier stages, virtual features which are assumed to be produced from previous models are estimated. This allows the acquired knowledge to be transferred from previous stages to the current stage without the replay of old training samples or loading of old models.

2. RELATED WORK

Online deep metric learning has been addressed under different assumptions in the literature. In the case that the replay of old training samples is allowed [9, 10], it is essentially a variant of joint training. The drawback is that the old training samples are not always available due to the issue of privacy concerns and the high maintenance costs. Under the more widely recognized assumption, the old training samples are not allowed to join in the training of new tasks. Recent approaches [7, 8] are all proposed under this assumption. Although different in the design of loss functions, they all regularize the new model with the old ones to preserve the inherent feature distribution that is learned from the old data.

Usually, knowledge distillation [17] is adopted to distill the feature information of the old categories from the old model to the new model [7, 8]. Intuitively, the current model inherits all the trained categories from the previous models. The last model preserves the feature distribution of all the previously trained categories. It is possible to correlate the feature distribution of all the previous categories based on the latest model. However, the errors caused by the incremental learning could be aggregated. The discriminativeness on the old tasks is therefore eroded gradually. As a result, recovering the feature space of the previous stages is still necessary. In [8], a Maximum Mean Discrepancy (MMD) based regularization loss is introduced to minimize the discrepancy between features of newly added categories from the original and adaptive networks. In [7], features from models of previous stages are estimated. The old features are recovered according to the variation of mAP before and after the training on the new task. The disadvantages of this approach lie in two aspects. Firstly, not all the variations in the feature space can be reflected by the accuracy variation in the image retrieval task. Moreover, the variation of feature space cannot simply be modeled as a linear transformation.

Mutual learning [16], as an extension of knowledge distillation, is an ensemble training strategy to improve generalization by transferring individual knowledge to each other. In such kind of framework, it is not required to have a fully-trained teacher model. Instead, two peer student models are trained and learn from each other via a mutual loss. As the framework shows no preference over any individual model, neither old nor new knowledge will dominate over each other.

In this paper, the online deep metric learning is addressed under the assumption that the replay of old training samples is not allowed. Different from existing approaches, mutual learning instead of a teacher-student framework is adopted for knowledge transfer from one stage to the next. We will show that it is more suitable for online deep metric learning. Additionally, a novel feature estimation strategy is proposed. Based on the models of previous stages, the drifting of feature space during the training of multiple stages can be captured. This in turn allows us to recover the feature distribution in old models more precisely than that of FECD [7].

3. METHOD

3.1. Problem Formulation

In the real world, the scale of learning issues increases incrementally on many occasions. For example, for a shopping website, more and more categories of products are put on sale periodically. It is required the trained model to be updated periodically. Given one stage of the incremental training is defined as one training task τ, the online deep metric learning is composed of a sequential set of training tasks $T = \{\tau_1, \tau_2, \cdots, \tau_i \cdots\}$. In one training task τ_i, it consists of a training set with n new categories, namely $\tau_i = \{(X_i^c, y_i^c)|c = 1, 2, ..., n\}$, where X_i^c are the set of training samples x. All the samples in X_i^c share the same class label y_i^c. Without loss of generality, we assume there is no intersection between any two training tasks, $\tau_i \cap \tau_j = \emptyset$.[1] Correspondingly, we expect a series of models are learned with the given training sets, namely $M = \{F_1(X_1^c, w_1), F_2(X_2^c, w_2), \cdots, F_i(X_i^c, w_i), \cdots\}$, where w_i are the trained weights of a model F_i. For each trained model F_i, it is essentially a mapping function, through which a given image x is mapped to a fixed-length feature vector f_x^i.

On the condition that the old training sample replay is not allowed, model F_i could be trained in two different ways. In the first way, all the previous models $F_1, F_2, \cdots, F_{i-1}$ along with the training set $\tau_i = \{(X_i^c, y_i^c)\}$ are available. Alternatively, only model F_{i-1} and $\tau_i = \{(X_i^c, y_i^c)\}$ are available for training task τ_i. In our solution, the online deep metric learning is addressed in the first way. In the following, we are going to first present a solution for the online deep metric learning that only involves two training tasks. We assume the initial model $F_o(X_o^c, w_o)$ is well trained on task τ_o with *n* old categories (X_o^c, y_o^c). The weights w_o already converges on task τ_o. Now new task τ_p with *m* new categories (X_p^c, y_p^c) are joined in. This is called "one-task online learning". Upon the basis of one-task online learning, the solution to the multi-task online learning is presented in the section followed.

3.2. One-task Online Learning

A two-student mutual learning framework [16] is adopted in our one-task online learning. The framework is shown in Fig. 1. Basically, there are three branches in the framework.

[1] Otherwise, the training on the overlapped categories is a fine-tuning of the trained model.

The first branch copies the model trained at the "Initial" stage and its weights are frozen during the training. Given a training sample x, it produces a feature, namely $f_x^o = F_o(x)$, which is used as a reference during the training of F_p.

The second branch initially copies weights from the first branch, namely $w_p \leftarrow w_o$. It is designed to train a model F_p that maintains the discriminativeness on the old categories and adapts to the new categories in τ_p. The third branch is a supporting model that is initialized with random weights. On the one hand, F_s learns the new categories from scratch. The triplet loss is adopted in F_p and F_s to separately learn the new categories in task τ_p. On the other hand, F_s and F_p also learn from each other, which learned knowledge is shared by the introduction of mutual loss on both branches. All these three branches share the same network structure. F_p and F_s in combination are called student models [16]. F_o is as the teacher model. The objective for one-task online learning is

$$\begin{aligned} L(X_p^c; w_o; w_p; w_s) &= \lambda_1 L_{triplet}(X_p^c; w_p) \\ &+ \lambda_1 L_{triplet}(X_p^c; w_s) \\ &+ \lambda_2 L_{corr}(X_p^c; w_o; w_p) \\ &+ \lambda_3 L_{mutual}(X_p^c; w_s; w_p), \end{aligned} \tag{1}$$

where λ_1, λ_2, and λ_3 are the hyperparameters used for weighting of different loss functions. In Eqn. 1, λ_1 is set to *1* and λ_2 is set to *10* [7]. λ_3 is empirically set to *8*, of which leads to the best performance according to our observation. $L_{triplet}$ is the triplet loss using hard sampling strategy. $L_{corr} = \frac{1}{N}\sum KL(\sigma(G_o), \sigma(G_p))$ is the correlation distillation loss between F_o and F_p [7]. It applies Kullback-Leibler divergence between two Gram matrices G_o, G_p which are calculated with features extracted by F_o, F_p and normalized with Softmax function $\sigma(\cdot)$. F_s also generates a Gram matrix G_s to regularizes the updating of F_p. The mutual distillation loss is defined as follows

$$\begin{aligned} L_{mutual-p} &= \frac{1}{N}\sum KL(\sigma(G_p), \sigma(G_s)) \\ L_{mutual-s} &= \frac{1}{N}\sum KL(\sigma(G_s), \sigma(G_p)) \\ L_{mutual} &= \frac{1}{2}(L_{mutual-p} + L_{mutual-s}). \end{aligned} \tag{2}$$

According to Eqn. 1, F_p is required to mimic F_o on the one hand. On the other hand, F_p also learns from F_s, which is better trained on the new task τ_p. Since F_s also learns from F_p, F_p and F_s converge to similar models as the training continues. Compared to the popular teacher-student distillation framework, mutual distillation helps us to find a wider/flatter robust minimum that generalizes better to the new task. In the multi-task scenario where tasks are added incrementally, performing online learning from flat minima will effectively mitigate forgetting of previous tasks [18]. In the following, the online deep metric learning is addressed in the multi-task context based on our one-task online learning solution.

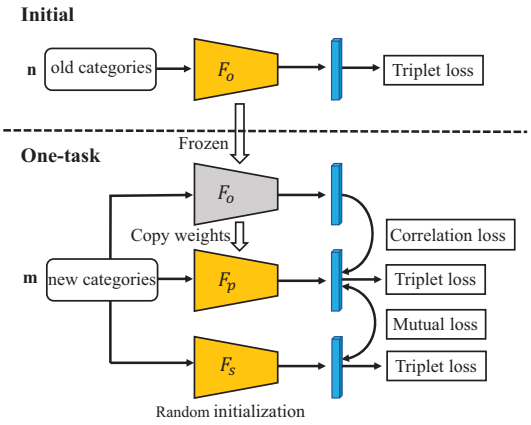

Fig. 1. One-task online learning overview. Initial: A model F_o is trained on n old categories (X_o^c, y_o^c). One-task: F_o is frozen as the teacher model and duplicated as the initialization of the student model F_p. A randomly initialized model F_s as the supporting student model is employed to solve the new task with F_p. Only samples of m new categories (X_p^c, y_p^c) are available during the training.

3.3. Multi-task Online Learning

In practice, new categories arrive in multiple stages. The model has to be trained periodically. At each stage, the model is trained on an aggregated new task. Namely, given there are $i-1$ $(i-1 \geq 2)$ tasks have been trained, we are now going to integrate a new group of categories $\tau_i = \{(X_i^c, y_i^c)|c = 1, 2, ..., n\}$ into the trained model F_{i-1} and work out a new model F_i. Although all the trained categories are kept in model F_{i-1}, the feature correlation for early tasks is distorted due to the aggregated concept drifting, which is more serious for the earlier tasks. It is, therefore, insufficient to correlate model F_i with F_{i-1} for the earlier tasks. Hereby, we assume all the models trained on the previous tasks are available. These models will be used to guide the more precise correlation of F_i with them.

Given a category k in τ_b $(b < i-1)$, prototype $\mu_{b,k}^b$ is defined as the centroid of kth category in task τ_b, which is calculated by the mean of features in this category extracted by F_b. When our training reaches Stage-i, the distribution of category k has drifted already. On the one hand, the image sample $x \in \tau_i$ should be fed into F_b to supervise the correlation of F_i to F_b. However, it is computationally expensive to load all the previous models to support the feature correlation and there is only one loaded previous model F_{i-1}. On the other hand, the feature from F_b cannot be directly estimated based on F_{i-1} since the drift of feature between F_b and F_{i-1} is unknown. Hereby, a way to estimate this drift is proposed.

Given a training sample x $(x \in \tau_{i-1})$, features f_x^b and f_x^{i-1} are extracted from F_b $(b = 1 \cdots i-2)$ and F_{i-1} respec-

Datasets	Training set (#Image/#Class)			Testing set (#Image/#Class)		
	Old task	New task	All	Old task	New task	All
CUB-200	3,504/100	3,544/100	7,048/200	2,360/100	2,380/100	4,740/200
Cars196	4,796/98	4,842/98	9,638/196	3,258/98	3,289/98	6,547/196
DeepFashion2	54,898/10,595	55,532/10,595	110,430/21,190	43,615/10,595	44,095/10,595	87,710/21,190

Table 1. Statistics of three datasets.

tively[2]. The drift of this individual feature is $\Delta f_x^{b\rightarrow i-1} = f_x^{i-1} - f_x^b$. Following with [19], the drift of the prototype $\mu_{b,k}^b$ at Stage-$i-1$ can be estimated as

$$\Delta\mu_{b,k}^{b\rightarrow i-1} = \frac{\sum_x S(f_x^b, \mu_{b,k}^b)\Delta f_x^{b\rightarrow i-1}}{\sum_x S(f_x^b, \mu_{b,k}^b)}, x \in \tau_{i-1}, \quad (3)$$

where $S(f_x^b, \mu_{b,k}^b)$ is the *Cosine* similarity between the feature f_x^b and the learned prototype $\mu_{b,k}^b$ at Stage-b. As we can see from Eqn. 3, the overall drift between two prototypes $(\mu_{b,k}^b, \mu_{b,k}^{i-1})$ is estimated by the weighted drift of each individual feature. Therefore, the prototype for category k $(k \in \tau_b)$ at Stage-$i-1$ is updated as $\mu_{b,k}^{i-1} = \mu_{b,k}^b + \Delta\mu_{b,k}^{b\rightarrow i-1}$.

During the training of Stage-i, only the previous model F_{i-1} is loaded as the teacher model. For a previous task τ_b, all of the prototype drifts $\Delta\mu_{b,k=1\cdots n}^{b\rightarrow i-1}$ are calculated and kept for updating the prototypes $\mu_{b,k=1\cdots n}^b \rightarrow \mu_{b,k=1\cdots n}^{i-1}$. Given a training sample $x \in \tau_i$, its feature f_x^b on F_b is not extracted directly as F_b is not loaded. Instead, it is estimated based on the teacher feature f_x^{i-1}. Namely, given f_x^{i-1}, $\mu_{b,k=1\cdots n}^{i-1}$ and $\Delta\mu_{b,k=1\cdots n}^{b\rightarrow i-1}$, we estimate how much f_x^{i-1} deviates from f_x^b

$$\Delta f_x^{i-1\rightarrow b} = -\frac{\sum_{k=1}^n S(f_x^{i-1}, \mu_{b,k}^{i-1})\Delta\mu_{b,k}^{b\rightarrow i-1}}{\sum_{k=1}^n S(f_x^{i-1}, \mu_{b,k}^{i-1})}, k \in \tau_b. \quad (4)$$

Similar as Eqn. 3, Eqn. 4 calculates a weighted drift for the feature f_x^{i-1} with respect to all the drifted prototypes and their drifts at Stage-$i-1$. Consequently, virtual feature $\hat{f}_x^b$ is estimated as $\hat{f}_x^b = f_x^{i-1} + \Delta f_x^{i-1\rightarrow b}$. These estimated virtual features at Stage-b regularize the updates of the current model F_i by constructing a correlation loss

$$L_{corr}^{b\rightarrow i} = \frac{1}{N}\sum KL(\sigma(G_b), \sigma(G_i)). \quad (5)$$

As shown in Fig. 2, we further estimate virtual features for tasks $\tau_1, \tau_2, \cdots, \tau_{i-2}$. With these virtual features, more Gram matrices are built, which provide more auxiliary knowledge rather than the only distilled knowledge from F_{i-1} for the current task τ_i

$$L_{corr} = \sum_t L_{corr}^{t\rightarrow i}, t = 1, 2, \cdots, i-1. \quad (6)$$

During the training, the 3rd term in Eqn. 1 is replaced with Eqn. 6, which becomes the overall loss function for multi-task online deep metric learning.

[2]This operation is undertaken before we train F_i. It is, therefore, offline.

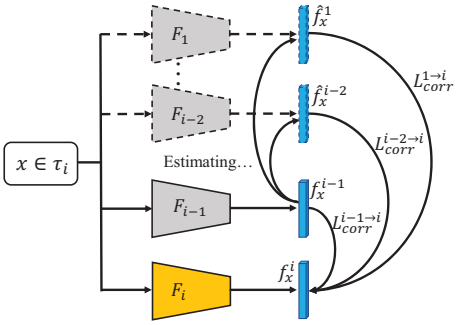

Fig. 2. Correlation regularization in multi-task online learning. Features of prior tasks are estimated from the teacher feature instead of extracting them with prior models. The Gram matrices produced by virtual features are used as supervision for learning the current task.

4. EXPERIMENT

In this section, the effectiveness of our online deep metric learning approach is studied on three datasets, Caltech-UCSD Birds 200 (CUB-200) [20], Cars196 [21], and DeepFashion2 [22]. The major information of these three datasets is summarized in Tab. 1. In the one-task learning case, the training set and the testing set of three datasets are evenly divided into two halves. The first half categories are treated as the old task, and the second half categories are treated as the new task. The performance of our approach is compared to LwF [15], EWC [14], FGIR [8], and FECD [7]. For all the approaches, BN-Inception [23] is adopted as the backbone, and triplet loss is used as the basic loss function. The results from FECD [7] are treated as the comparison baseline. The result from joint training is supplied as a reference, which learns all the categories as one training task. The performance is evaluated by Recall@1 for the retrieval on each dataset.

We noticed the experimental flaws pointed out by [24] in the current literature of deep metric learning. In our experiment design, no sophisticated image augmentation is adopted. The parameters in Batch-Norm are frozen.

4.1. Performance on One-task Online Learning

The experimental results on three datasets are reported on Tab. 2. The results for LwF, EWC, and FGIR are cited from [8] directly. The results from the "Initial" model and "Fine-tuning" model are reported. The "Initial" model is

Datasets	Approaches	Old	New	All
CUB-200	Initial	79.79	55.55	62.76
	Fine-tuning	68.81	79.92	69.01
	Joint Training	77.67	79.66	76.75
	LwF	54.92	75.76	-
	EWC	62.03	73.32	-
	FGIR	74.41	73.11	-
	FECD	77.20	76.09	72.43
	Ours	**77.29**	**78.07**	**73.73**
Cars196	Initial	82.23	63.58	-
	Fine-tuning	67.77	90.00	-
	Joint Training	82.26	92.28	85.15
	FECD	80.82	91.21	82.36
	Ours	**81.06**	**92.40**	**83.49**
DeepFashion2	Initial	55.79	55.04	-
	Fine-tuning	54.81	54.97	-
	Joint Training	56.02	56.21	49.66
	FECD	56.26	56.49	50.18
	Ours	**56.33**	**56.65**	**50.35**

Table 2. Recall@1 (%) on one-task online learning. "Initial", "Fine-tuning" and "Joint Training" are as the references.

Datasets	Tasks	Joint	LwF	EWC	FGIR	FECD	**Ours**
CUB-200	1-100	77.67	33.31	36.82	66.40	73.64	**73.90**
	101-125	82.14	49.83	57.99	70.07	**77.21**	**77.21**
	126-150	77.83	48.00	50.67	69.00	73.00	**75.33**
	151-175	88.44	67.17	64.15	73.87	81.07	**83.75**
	176-200	87.90	83.70	82.02	85.21	87.23	**88.40**
	1-200	76.75	-	-	-	**67.95**	67.91
Cars196	1-98	82.26	-	-	-	73.76	**75.54**
	99-123	97.13	-	-	-	94.86	**96.42**
	124-148	98.19	-	-	-	96.38	**96.86**
	149-173	97.38	-	-	-	97.26	**98.45**
	174-196	94.52	-	-	-	95.54	**96.56**
	1-196	85.15	-	-	-	75.00	**76.42**
DeepFashion2	1-10595	56.02	-	-	-	55.16	**56.04**
	10596-13256	67.30	-	-	-	66.94	**67.08**
	13257-15890	67.62	-	-	-	67.56	**68.33**
	15891-18558	68.42	-	-	-	68.22	**69.01**
	18559-21190	66.22	-	-	-	67.41	**67.76**
	1-21190	49.66	-	-	-	49.78	**50.58**

Table 3. Recall@1 (%) on multi-task online learning. "Joint" (namely Joint Training) serves as the reference.

trained on the old task only. The "Fine-tuning" model is trained on the old task and then fine-tuned on the new task.

As shown on the table, the initial model and fine-tuned model perform poorly either on CUB-200 or Cars196. This does indicate the necessity of an online metric learning approach to fit in. Among all the online approaches, our approach outperforms the rest constantly on all three tasks (old, new, and all). The performance gap between offline and online approaches on DeepFashion2 is minor. This is mainly because the training task is in large-scale (*21,190* categories). The discriminativeness of the trained model is already saturated after the first half categories have been trained. Nevertheless, our approach shows the constant improvement over existing approaches, as it shows the good trade-off between the old and new tasks.

4.2. Performance on Multi-task Online Learning

In order to simulate the online learning of multiple stages, the second half of the categories in three datasets are divided into four subsets evenly. So for CUB-200, Cars196, and DeepFashion2, there are *25*, *25*, and *2,661* new categories respectively joining in the training as a new task at each stage. In order to be in line with [7], the performance of the model in multi-task online learning is reported in two ways. Firstly, the Recall@1 of the final model is reported on each individual stage. Moreover, the overall performance of the final model on all the trained categories is reported.

The performance from our approach and LwF [15], EWC [14], FGIR [8], and FECD [7] are reported in Tab. 3. As shown from the table, FECD and our approach outperform the rest online approaches by a large margin. Compared to FECD, our approach shows constantly better performance in most of the stages. In particular, our approach outperforms FECD across all the tasks on Cars196 and DeepFashion2 datasets. The superior performance of our approach owes to both the adoption of mutual learning and novel virtual feature estimation for the old models.

Datasets	Approaches	ResNet-50			Vision Transformer		
		Old	New	All	Old	New	All
CUB-200	Initial	79.53	33.53	50.42	83.81	65.00	71.12
	Fine-tuning	51.10	79.62	57.81	77.25	80.59	75.46
	Joint Training	79.07	78.49	76.50	82.97	81.97	80.57
	FECD	75.59	75.08	70.44	**82.25**	79.03	78.08
	Ours	**76.69**	**75.34**	**70.55**	82.16	**80.17**	**78.59**
Cars196	Initial	85.21	41.81	57.49	80.63	52.87	60.68
	Fine-tuning	62.49	94.44	71.16	57.58	87.81	65.66
	Joint Training	84.99	93.86	88.24	77.56	86.01	78.45
	FECD	**83.36**	92.13	83.24	**77.53**	83.98	75.82
	Ours	82.81	**93.58**	**83.78**	76.40	**85.89**	**75.93**
DeepFashion2	Initial	53.06	50.87	45.65	58.40	58.36	52.51
	Fine-tuning	52.70	53.30	46.57	58.37	58.14	52.34
	Joint Training	54.03	53.99	47.66	58.40	58.46	52.49
	FECD	**54.47**	54.45	47.94	58.29	58.35	52.37
	Ours	54.11	**54.88**	**48.14**	**58.44**	**58.44**	**52.47**

Table 4. Ablation study with different backbones on one-task online learning. Evaluated by Recall@1 (%).

As shown on Tab. 5, when the mutual learning is integrated with the feature estimation of the way in FECD [7], the performance of "FECD(Mutual)" is inferior to ours in most of the cases. Moreover, it can be observed that FECD(Mutual) performs better than FECD on the last three tasks, which reveals the effectiveness of mutual loss over the conventional teacher-student model.

4.3. Ablation Study

The performance of our approach is also studied when the BN-Inception backbone is replaced with ResNet-50 and Vision Transformer. The study is conducted on both one-task and multi-task scenarios. The one-task and multi-task online learning results are shown on Tab. 4 and Tab. 6 respectively. Although the performance fluctuates considerably for all the approaches across different backbones, our approach outperforms the competing approach FECD in most of the cases. It confirms the stability of our approach on different backbones.

Tasks	Mutual	FECD	FECD(Mutual)	Ours
1-100	71.57	73.64	**73.98**	73.90
101-125	75.68	**77.21**	76.19	**77.21**
126-150	70.50	73.00	73.17	**75.33**
151-175	81.24	81.07	82.08	**83.75**
176-200	87.56	87.23	88.07	**88.40**
1-200	66.12	**67.95**	67.76	67.91

Table 5. Ablation study with different feature estimation approaches on CUB-200 on multi-task online learning. Evaluated by Recall@1 (%).

Tasks	Joint	FECD	**Ours**
1-100	82.97	80.51	**81.40**
101-125	85.54	81.46	**82.48**
126-150	77.33	78.17	**78.50**
151-175	87.60	83.92	**86.60**
176-200	90.42	**88.91**	88.40
1-200	80.57	76.62	**77.83**

Table 6. Ablation study with Vision Transformer as the backbone on CUB-200 on multi-task online learning. Recall@1 (%) is the evaluation metric.

5. CONCLUSION

We have presented our solution for online deep metric learning both for one stage and multiple stage scenarios. Different from existing solutions, our approach is built upon a mutual learning structure, which makes a good balance between the old and new learning tasks. Moreover, a novel virtual feature estimation approach is proposed. In combination with mutual learning, superior performance is achieved in the multi-task learning scenario. In most cases, it even outperforms the joint training. The effectiveness of our approach is validated on different datasets and with different network backbones.

Acknowledgment

This work is supported by National Natural Science Foundation of China under grants 61572408 and 61972326.

6. REFERENCES

[1] Sumit Chopra et al., "Learning a similarity metric discriminatively, with application to face verification," in *IEEE CVPR*. IEEE, 2005, vol. 1, pp. 539–546.

[2] Florian Schroff et al., "Facenet: A unified embedding for face recognition and clustering," in *IEEE CVPR*, 2015, pp. 815–823.

[3] Dong Yi et al., "Deep metric learning for person re-identification," *ICPR*, pp. 34–39, 2014.

[4] Hyun Oh Song et al., "Deep metric learning via lifted structured feature embedding," in *IEEE CVPR*, 2016, pp. 4004–4012.

[5] Xun Wang et al., "Multi-similarity loss with general pair weighting for deep metric learning," in *IEEE CVPR*, 2019, pp. 5022–5030.

[6] Chao-Yuan Wu et al., "Sampling matters in deep embedding learning," in *IEEE ICCV*, 2017, pp. 2840–2848.

[7] Wei Chen et al., "Feature estimations based correlation distillation for incremental image retrieval," *IEEE TMM*, 2021.

[8] Wei Chen et al., "On the exploration of incremental learning for fine-grained image retrieval," *arXiv preprint arXiv:2010.08020*, 2020.

[9] Xing Tian et al., "Complementary incremental hashing with query-adaptive re-ranking for image retrieval," *IEEE TMM*, 2020.

[10] Dayan Wu et al., "Deep incremental hashing network for efficient image retrieval," in *IEEE CVPR*, 2019, pp. 9069–9077.

[11] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of learning and motivation*, 1989.

[12] Hanul Shin et al., "Continual learning with deep generative replay," *arXiv preprint arXiv:1705.08690*, 2017.

[13] Sylvestre-Alvise Rebuffi et al., "icarl: Incremental classifier and representation learning," *IEEE CVPR*, 2017.

[14] James Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *PANS*, vol. 114, no. 13, pp. 3521–3526, 2017.

[15] Zhizhong Li et al., "Learning without forgetting," *IEEE TPAMI*, vol. 40, no. 12, pp. 2935–2947, 2017.

[16] Ying Zhang et al., "Deep mutual learning," *IEEE CVPR*, pp. 4320–4328, 2018.

[17] Geoffrey Hinton et al., "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[18] Guangyuan Shi et al., "Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima," *NIPS*, vol. 34, 2021.

[19] Lu Yu et al., "Semantic drift compensation for class-incremental learning," in *IEEE CVPR*, 2020, pp. 6982–6991.

[20] P. Welinder et al., "Caltech-UCSD Birds 200," Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.

[21] Jonathan Krause et al., "3D object representations for fine-grained categorization," in *IEEE 3dRR*, Sydney, Australia, 2013.

[22] Yuying Ge et al., "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *IEEE CVPR*, 2019, pp. 5337–5345.

[23] Sergey Ioffe et al., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*. PMLR, 2015, pp. 448–456.

[24] Kevin Musgrave et al., "A metric learning reality check," in *ECCV*. Springer, 2020, pp. 681–699.