Debugging and Explaining Metric Learning Approaches: An Influence Function Based Perspective

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep metric learning (DML) learns a generalizable embedding space where the representations of semantically similar samples are closer. Despite achieving good performance, the state-of-the-art models still suffer from the generalization errors such as similar samples are farther and dissimilar samples are closer in the space. In this work, we design empirical influence function (EIF), a debugging and explaining technique for the generalization errors of the state-of-the-art metric learning models. EIF is designed to efficiently identify and quantify how a subset of training samples contribute to the generalization errors. Moreover, given a user-specific error, EIF can be used to relabel a potentially noisy training sample as a mitigation. In our quantitative experiment, EIF outperforms the traditional baseline in identifying root-cause training samples (e.g., noisy samples) by mitigating 1-3% more confusing pairs and 33.5% less time. In the field study on the well-known datasets such as CUB200, CARS196, and InShop, EIF identifies 4.4%, 6.6%, and 17.7% labelling mistakes, indicating the direction of the DML community to further improve the model performance.

1 Introduction

Deep metric learning (DML) learns a generalizable embedding space of a dataset, where semantically similar samples are closer [12]. It has been widely applied in face recognition [25], image retrieval [36], and clustering [5]. Recently, the record-breaking methodologies have been generally evolving from pairwise-based approaches (e.g., triplet-based [12] and pair-based [11]) to proxy-based approaches [19, 13, 29, 10, 22]. However, many recent works [26, 13, 29] begin to achieve only marginal improvements on the classical datasets [35, 17, 18]. Thus, the explanation approaches of DML are in need for understanding *why the trained model can confuse the dissimilar samples and cannot recognize the similar samples*.

This research starts with our investigation on popular classical datasets (i.e., CUB200, CARS196, and InShop) that, for state-of-the-art metric learning approaches, (1) different approaches share not only similar performance metrics (e.g., Recall@1), but the same types of generalization errors, and (2) the human inspection sometimes has no less generalization errors on existing DML datasets. We report the results at Section 5 and our anonymous website [2]. The observation leads us to design an influence function based explanation framework to investigate the existing datasets, consisting of:

- **Scalable training-sample attribution:** We propose empirical influential function (EIF) to (1) identify what training samples contribute to the generalization errors, and (2) quantify how much contribution they make to the errors. Technically, we replace the Hessian matrix in traditional influence function [14] with light-weighted Newton step estimation to improve both its effectiveness and efficiency.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

- **Dataset relabelling recommendation:** We further aim to identify the potentially "buggy" training samples with mistaken labels and generate their relabelling recommendation.

Technique Evaluation. We design one sample-locating experiment and one sample-relabelling experiment on 3 datasets to evaluate our framework. In both experiments, we introduce 10% noisy training samples to train a DML model. In the sample-locating experiment, we evaluate EIF's performance on locating the noisy training samples for both individual-pair confusion (i.e., model is confused with one pair of samples) and group-pair confusion (i.e., model is confused with a group of pairs of samples). The results show that, compared to the traditional influence function [14], EIF identifies root-cause training samples (e.g., noisy samples) by mitigating 1-3% more confusing pairs with 33.5% less time. The sample-relabelling experiment shows that we can accurately recommend on average 91.5% of the mis-labelled training samples.

Empirical Investigation. Based on the proposed framework, we investigate the classical datasets such as on CUB200 [35], CARS196 [17], and InShop [18]. We find that the labels annotated in the datasets are more unreliable than expected. We summarize a taxonomy of dataset problems (e.g., see Section 5), We further conclude that the major barriers for DML performance is not the model design, but the confusing labels in the classical datasets.

In summary, this work makes the following contributions:

- We propose empirical influence function (EIF) for DML approaches, which can attribute root-cause training samples for arbitrary number of pairs of unseen test samples.
- We propose a sample-relabelling technique based on EIF for mitigating potential dataset problems.
- We identify and categorize labelling problems of the well-known classical datasets for DML, indicating the potential direction to further improve the performance of DML approaches.

2 Problem Setting

We denote the input space as $\mathcal{X} \subset \mathbb{R}^d$ (d is the input dimension), the embedding space as $\mathcal{Z} \subset \mathbb{R}^m$ (m is the embedding dimension), and the class label space $\mathcal{Y} \subset Z^+$. A DML network $f(.)$ parameterized by θ is denoted as $f_\theta : \mathcal{X} \rightarrow \mathcal{Z}$. Given a distance measure $d(.,.)$ where $d : \mathcal{Z} \times \mathcal{Z} \rightarrow \{0, \mathbb{R}^+\}$, we can calculate the distance between any input pairs $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{X}$ by $d(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}_j))$. Typically, cosine distance and Euclidean distance are common choices of $d(.,.)$. In addition, we define a labelling function $l(.)$ where $l : \mathcal{X} \rightarrow \mathcal{Y}$.

The DML techniques aim to optimize θ such that $\forall \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in \mathcal{X}$ $(y_i, y_j, y_k \in \mathcal{Y}, y_i = y_j \neq y_k)$, $d(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}_j)) < d(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}_k))$. Pair-based losses try to optimize the inequality directly on pairs / triplets [11, 12, 27]. However, studies show that they suffer from (1) slow and noisy convergence, (2) high computational complexity. Therefore, proxy-based losses have been proposed to address these issues [19, 29, 13, 22, 6], which significantly outperform the pair-based losses. Most of proxy-based losses are defined on a per-sample basis [19, 29, 22, 6]. Therefore, in this work, we follow the loss of form $L(\mathbf{x}; \theta)$.

Given the training dataset set $\mathcal{X}_{train}$ with labels $\mathcal{Y}_{train}$ and the testing dataset $\mathcal{X}_{test}$ with labels $\mathcal{Y}_{test}$ where training and testing are class disjoint, i.e., $\mathcal{Y}_{train} \cap \mathcal{Y}_{test} = \emptyset$. The generalization error can be defined as a testing sample not sharing the same class label as its nearest neighbor in the space, i.e.,

Definition 2.1. We define a testing-sample pair $p = (\mathbf{x}_i, \mathbf{x}_j)$ $(\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}_{test})$ as a **confusion pair** if:

1. $\mathbf{z}_j = f_\theta(\mathbf{x}_j)$ is the nearest neighbour of $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$ in the space regarding the measure $d(.,.)$;
2. $y_i \neq y_j$ and $y_i, y_j \in \mathcal{Y}_{test}$.

Given a set of confusion pairs $P_c = \{p_1, p_2, ..., p_n\}$, we aim to achieve the following two goals:

G1. Influential Sample Identification We locate the set of root-cause training samples $\mathcal{X}_r \subset \mathcal{X}_{train}$ such that retraining with re-weighted $\mathcal{X}_r$ can increase the average distance of confusion pairs in P_c,

$$
\begin{aligned}
\hat{\theta}_r &= \arg\min_{\theta} \frac{1}{|\mathcal{X}_{train}|} \sum_{\mathbf{x_i} \in \mathcal{X}_{train}} \Big(L(\mathbf{x_i}, y_i; \theta) + \epsilon_i \mathbb{1}(\mathbf{x_i} \in \mathcal{X}_r) L(\mathbf{x_i}, y_i; \theta) \Big) \\
\mathcal{X}_r &= \arg\max_{\mathcal{X}_r \subset \mathcal{X}_{train}} \frac{1}{|P_c|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in P_c} d(\mathbf{x}_i, \mathbf{x}_j; \hat{\theta}_r)
\end{aligned} \tag{1}
$$

In Equation 1, $\hat{\theta}_r$ is the retrained model by re-weighting the training sample $\mathbf{x}_i$ by ϵ_i. Specifically, $|\epsilon_i|$ is the re-weighting magnitude, if $\epsilon_i > 0$, $\mathbf{x_i}$ is a helpful training sample; if $\epsilon < 0$, $\mathbf{x_i}$ is a harmful training samples. Under the measure $d(.,.)$ regarding the learned $\hat{\theta}_r$, the average distance of the confusion pair set P_c is maximized.

G2. Influential Sample Relabelling We locate the set of root-cause training samples $\mathcal{X}_l \subset \mathcal{X}_{train}$ and a relabelling function $\mathcal{R} : \mathcal{X} \rightarrow \mathcal{Y}$ such that retraining θ by changing the labels of $\mathcal{X}_l$ with $\mathcal{R}(.)$ can increase the average distance of confusion pairs in P_c the most. Specifically,

$$
\begin{aligned}
\hat{\theta}_l &= \arg\min_{\theta} \frac{1}{|\mathcal{X}_{train}|} \sum_{\mathbf{x}_i \in \mathcal{X}_{train}} \Big(L(\mathbf{x}_i, y_i; \theta) + \mathbb{1}(\mathbf{x}_i \in \mathcal{X}_l) \big(L(\mathbf{x}_i, \mathcal{R}(\mathbf{x}_i); \theta) - L(\mathbf{x}_i, y_i; \theta) \big) \Big) \\
\mathcal{X}_l &= \arg\max_{\mathcal{X}_l \subset \mathcal{X}_{train},\ l: \mathcal{X} \rightarrow \mathcal{Y}} \frac{1}{|P_c|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in P_c} d(\mathbf{x}_i, \mathbf{x}_j; \hat{\theta}_l)
\end{aligned} \tag{2}
$$

3 Approach

Recaping Influence Function Given that a learned model parameterized by $\hat{\theta}$, a sample $\mathbf{x}$, and its loss value on the model $L(\mathbf{x}; \hat{\theta})$. The influence of up-weighting a training sample $\mathbf{x}_{train}$ by on a testing sample $\mathbf{x}_{test}$ is [14]:

$$
I(\mathbf{x}_{train}, \mathbf{x}_{test}) = L(\mathbf{x}_{test}; \hat{\theta}') - L(\mathbf{x}_{test}; \hat{\theta}) \approx -\nabla_{\hat{\theta}} L(\mathbf{x}_{test}; \hat{\theta})^{\intercal} H_{\hat{\theta}}^{-1} \nabla_{\hat{\theta}} L(\mathbf{x}_{train}; \hat{\theta}) \tag{3}
$$

In Equation 3, $\hat{\theta}$ is the original model, $\hat{\theta}'$ is the retrained model after $\mathbf{x}_{train}$ is adjusted, $H_{\theta}^{-1} = \frac{1}{|\mathcal{X}_{train}|} \sum_{i=1}^{|\mathcal{X}_{train}|} \nabla_{\theta}^2 L(\mathbf{x}_i, \hat{\theta})$. Given a confusion pair $p_c = (\mathbf{x}_i, \mathbf{x}_j)$ with distance $d_{\theta}(p_c; \hat{\theta})$, the first item can be replaced with $d_{\theta}(p_c; \hat{\theta})$ if we need to calculate the influence of a training sample, i.e.,

$$
I(\mathbf{x}_{train}, p_c) = -\nabla_{\hat{\theta}} d(p_c; \hat{\theta})^{\intercal} H_{\hat{\theta}}^{-1} \nabla_{\hat{\theta}} L(\mathbf{x}_{train}; \hat{\theta}) \tag{4}
$$

However, computing $I(\mathbf{x}_{train}, p_c)$ still suffers from two drawbacks, i.e., (1) high computational cost and (2) inaccurate approximation for group-pair confusion.

High Computational Cost Computing the Hessian function $H_{\hat{\theta}}^{-1}$ is non-trivial, which requires the complexity of $O(np^2 + p^3)$ where n is training dataset size and p is the parameter size. In [14], the complexity is further optimized to $O(np + rtp)$ where $rt \sim O(n)$. However, the complexity to calculate $\nabla_{\hat{\theta}} d(p_c; \hat{\theta})$ and $\nabla_{\hat{\theta}} L(\mathbf{x}; \hat{\theta})$ are $O(p)$ and $O(np)$ respectively. With the increase of parameter size (e.g., millions) and the training set, the runtime cost is still considerably large.

Accuracy of Influence of Group-pair Confusion When we are locating the influential training samples for a group of confusion pairs, Equation 4 cannot hold in practice because the derivation of the Hessian matrix $H_{\hat{\theta}}^{-1}$ in Equation 3 assumes that $\hat{\theta}' \sim \hat{\theta}$ so that a series of higher-order terms in the Talyer expansion can be omitted [14]. However, the number of influential training samples to a group of confusion pairs (i.e., $\mathcal{X}_r$ in Equation 1) can be large, indicating that the learned parameters $\hat{\theta}'$ can be very different from the original $\hat{\theta}$. Thus, such omission can lead to inaccurate estimation.

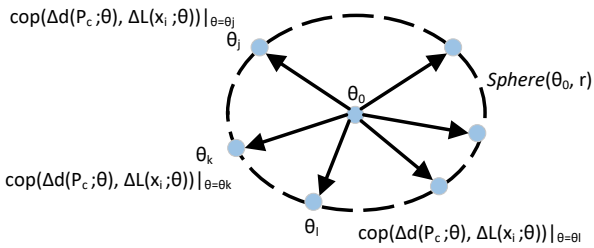


Figure 1: Illustration of EIF calculation by sampling most representative θ on the hypersphere.

Empirical Influence Function (EIF) In this work, we design empirical influence function (EIF) to address the above challenges. Our rationale lies in that, (1) fitting harmful samples contribute to more generalization errors, and (2) fitting helpful samples contribute to mitigating generalization errors. We estimate the influence score by the empirical co-change product between the distance of confusion pairs $\triangle d(P_c;\theta)$ and the training loss $\triangle L(\mathbf{x};\theta)$ ($\mathbf{x} \in \mathcal{X}_{train}$) as Equation 5:

$$I(\mathbf{x}, P_c) = E_\theta[cop(\triangle d(P_c;\theta), \triangle L(\mathbf{x};\theta))] = \int \frac{\triangle d(P_c;\theta)\triangle L(\mathbf{x};\theta)}{||\theta - \theta_0||_2} P(\theta) d\theta \tag{5}$$

Assume that we train the network by upweighting $\mathbf{x}$ to have $\theta_0 \to \theta$, the co-change product is defined as $cop(\triangle d(P_c;\theta), \triangle L(\mathbf{x};\theta)) = \frac{\triangle d(P_c;\theta)\triangle L(\mathbf{x};\theta)}{||\theta-\theta_0||_2}$. $cop(\triangle d(P_c;\theta), \triangle L(\mathbf{x};\theta)) > 0$ indicates that $\mathbf{x}$ is harmful; and $cop(\triangle d(P_c;\theta), \triangle L(\mathbf{x};\theta)) < 0$ indicates that $\mathbf{x}$ is helpful. However, retraining each individual sample is prohibitively expensive, therefore we need to estimate $cop(\triangle d(P_c;\theta), \triangle L(\mathbf{x};\theta))$ by taking the integral over the possible distribution of θ, i.e. $P(\theta)$.

As shown in Figure 1, assume that θ_0 is the model parameter, the distribution of θ forms a high-dimensional Sphere $P(\theta) \sim Sphere(\theta_0, r)$ with a radius r. Given that Equation 5 is intractable, the integral is estimated by drawing Monte Carlo samples (e.g., θ_j, θ_k, θ_l) on the sphere. i.e., we sample $\Theta \subset Sphere(\theta_0, r)$

$$E_\theta[cop(\triangle d(P_c;\theta), \triangle L(\mathbf{x};\theta))] \approx \frac{1}{|\Theta|} \sum_{\theta' \in \Theta} \frac{\triangle d(P_c;\theta')\triangle L(\mathbf{x};\theta')}{||\theta' - \theta_0||_2} \tag{6}$$

In this work, we heuristically construct Θ, regarding how sensitive each $\theta' \in \Theta$ is to the change of $d(P_c;\theta)$. Specifically, we first sample $\theta_{max} \in Sphere(\theta_0, r)$ pointing to the steepest ascent direction of $d(P_c;\theta)$, which can be considered as the "repairing" the network. Symmetrically, θ_{min} is taken as the steepest descent direction of $d(P_c;\theta)$, which can be considered as the "worsening" network. Next, given a user-defined threshold N_θ, we repetitively sample $\theta' \in Sphere(\theta_0, r)$ which is orthogonal to $\forall\theta \in \Theta$, until N_θ exhausts. Our experiment shows that even using $\Theta = \{\theta_{max}\}$ can achieve an accurate influence estimation, with well improved runtime efficiency.

The complexity of our empirical influence function on an individual training sample is $O(N_\theta \times p)$, where p is the parameter size. As a result, we can attach each $\mathbf{x} \in \mathcal{X}_{train}$ with an influence function score $I(\mathbf{x}, P_c)$, which can be used to rank $\mathcal{X}_{train}$ and select the most influential training samples.

Sample Relabelling Recommendation Based on the empirical influence function, we propose the data relabelling as a data cleansing strategy. Given the identified influential harmful training samples $\mathcal{X}_{harm}$ from EIF. Assuming that the noisy samples are the minority in the training dataset, we recommend a label $l \in \mathcal{Y}_{train}$ according to the labels of its neighbors through a weighted-KNN algorithm (See Algorithm 1). Intuitively, the label supported by more weighted neighbours is recommended to correct a harmful sample $\mathbf{x}$. The neighbours are weighted by their proximity to $\mathbf{x}$. Note that, we use hard label re-assignment (one-hot label) in Algorithm 1 (line 5). We can also revise it to soft label re-assignment by using the probability score $P(y_{new})$ as the calibration label.

Algorithm 1 Training Sample Relabelling

Input : training dataset $\mathcal{X}_{train}$, K
Output : relabelled dataset $S = \{(x, l)\}$

$\mathcal{X}_{harmful} = training_sample_location(\hat{\theta}, \mathcal{X}_{train})$
$S = \emptyset$
for $\mathbf{x} \in \mathcal{X}_{harm}$ **do**
  $\{(\mathbf{x}_{nni}, y_{nni})\} \leftarrow$ KNN-Set($f_{\hat{\theta}}(\mathbf{x})$)
  for $l \in Set(\mathcal{Y}_{train})$ **do**
    $P(y_{new} = l) = \frac{\sum_{i=1}^{K} exp(-d(\mathbf{x}, \mathbf{x}_{nni}))\mathbf{1}(y_{nni}=l)}{\sum_{l'} \sum_{i=1}^{K} exp(-d(\mathbf{x}, \mathbf{x}_{nni}))\mathbf{1}(y_{nni}=l')}$
  $S = S \cup \{(\mathbf{x}, \arg\max_l P(y_{new} = l))\}$
return S

Table 1: Paired t-test results for alternative hypothesis on individual 50 pairs $H_1 : \triangle d_{EIF}(p) > \triangle d_{IF}(p)$. And Wilcoxon signed-rank test results for alternative hypothesis on 10 group of pairs $H_1 : \triangle d_{EIF}(G_p) > \triangle d_{IF}(G_p)$

Method \ Dataset	Statistics	CUB200		CARS196		InShop	
		Individual	Group	Individual	Group	Individual	Group
ProxyNCA++	Improvement	0.02	0.02	0.01	0.02	0.01	-0.00
	p-value	1.4e-15	9.8e-04	1.0e-07	9.8e-04	3.0e-07	0.9
SoftTriple	Improvement	0.02	0.03	0.01	0.02	0.01	0.005
	p-value	1.9e-17	9.8e-04	3.5e-07	9.8e-04	1.4e-02	1.4e-02

4 Experiment

Experimental Settings In this study, we use Proxy-NCA++ [29] and SoftTriple [22] loss to train DML models with ResNet-50 model architecture on three datasets, i.e., CUB200 [35], CARS196 [17], and InShop [18]. The influential sample location capability is evaluated by a DML training experiment and a noisy data detection experiment; and the sample relabelling capability is evaluated by the noisy data detection experiment. In the DML training experiment, we evaluate whether retraining the model by upweighting or downweighting the reported influential samples can mitigate the confusion pairs. In the noisy data detection experiment, we flip 10% of the labels in the above training datasets, and evaluate whether EIF can identify and correct those noisy samples. We choose the influence function with modified testing loss function (as in Equation 4) as our baseline. More details of training configuration can be referred on our anonymous website [2].

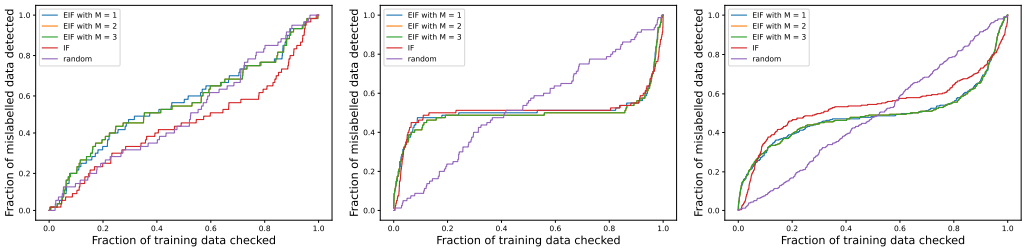
DML Training Experiment In the experiment, for each dataset, we select the 50 most confusing pairs in its testing dataset. For each selected confusion pair p, we evaluate the identified influential training samples (either helpful or harmful) by comparing :

$$\triangle d(p) = d(p; \theta') - d(p; \theta) \tag{7}$$

In Equation 7, θ represents the original network, θ' represents the network *actually* trained by down-weighting harmful samples and up-weighting helpful samples for one epoch, and $d(p; \theta)$ represents the normalized distance of the confusion pair p on θ.

Furthermore, we select 10 groups of confusion pairs as follows. We select the top-10 testing classes with the most generalization errors, denoted as $\mathcal{C} = \{c_1, c_2, ..., c_{10}\}$. For each $c_i \in \mathcal{C}$, we take all the confusion pairs including samples in c_i as its confusion counterpart. Therefore, for each confusion pair group G_p, we evaluate the change of average distance after retraining the influential training samples.

$$\triangle d(G_p) = \frac{1}{|G_p|} \sum_{p \in G_p} d(p; \theta') - d(p; \theta) \tag{8}$$



(a) CUB200 (b) CARS196 (c) InShop

Figure 2: The performance of detecting 10% mislabelled samples, M stands for N_θ (see Section 3)

Table 2: Mislabelled Sample Recommendation

Mislabelling Ratio	Method \ Dataset	CUB200	CARS196	InShop
1%	ProxyNCA++	84.62%	100.00%	96.52%
	SoftTriple	100.00%	96.42%	97.12%
5%	ProxyNCA++	88.00%	91.30%	95.24%
	SoftTriple	100.00%	91.89%	95.17%
10%.	ProxyNCA++	86.36%	97.44%	97.74%
	SoftTriple	96.00%	91.89%	92.19%

Noisy Data Detection Experiment In the experiment, we evaluate EIF and the influence function by how well they can locate those noisy data. In addition, we evaluate how many labels can our relabelling algorithm recommend accurately.

Results: Influential Sample Identification Table 1 shows the comparison of the score $\triangle d(p)$ between EIF and the original influence function (IF). We can see that EIF outperforms IF on the average improvement on "deconfusing" the confusion pair. Compared to IF, EIF identifies samples with which retraining the model can increase larger distance for the confusion pair. Moreover, the improvement is of statistical significance (all p-values are smaller than 0.001 but the group confusion on the InShop dataset). Compared to other datasets, InShop is a few-shot dataset with more noises (i.e., 50K over 8K classes), which may require a larger training batch size for EIF to be more effective.

On the 10% noisy dataset, Figure 2 compares how well EIF and IF locate the 10% mislabelled samples in the three datasets. The x-axis represents the ranked training samples regarding their influence score in descending order, the y-axis represents the ratio of mislabelled samples being detected. We plot random choice (in purple), IF (in red), and EIF with different M (in blue, yellow, and green). Readers can refer to Section 3 for the definition of M (i.e., N_θ).

Overall, both IF and EIF have comparable performance on mislabelled detection in top-ranked training samples. In CUB200 and CARS196, EIF outperforms IF in the effectiveness of detecting noisy samples. In InShop, EIF leads the advantage in the top 5% samples, while IF catches up at lower rankings. In addition, we observe that increasing M may not necessarily improve the performance of EIF, while incurring the runtime cost. Therefore, we recommend to set $\Theta = \{\theta_{max}\}$, for the sake of a more stable influence function and minimum runtime cost.

Results: Relabelling Recommendation Table 2 shows the performance of recommending the relabelling suggestion. Overall, the sampling relabelling algorithm performs well in predicting and re-calibrating the labels. Note that, with the increase in mislabelling ratio, the relabelling algorithm can still preserve its performance.

Results: Runtime Performance Table 3 shows the runtime cost of EIF and IF in the above experiment. We use $M = 1$ for (see Section 4) recording the runtime cost of EIF. In Table 3, we report the mean$\pm$std runtime for mislabelled sample experiments. Overall, we can see that EIF can boost the runtime efficiency of IF by $\sim$33.5% on average.

Table 3: Average Runtime Statistics (in seconds)

Dataset	App	Mislabelled Sample	
		Detection	Recommendation
CUB200	EIF	260.20 ± 29.28	1.85 ± 0.01
	IF	375.56 ± 68.06	/
CARS196	EIF	453.61 ± 28.76	3.74 ± 0.03
	IF	588.11 ± 23.23	/
InShop	EIF	1306.43 ± 27.31	43.24 ± 0.11
	IF	2454.11 ± 9.55	/

Table 4: Agreeable and Disagreeable Confusion Pairs

Dataset	#Confusion		#Mis-similar	
	#Agreeable	#Disagreeable	#Agreeable	#Disagreeable
CUB200	15	5	11	9
CARS196	6	14	2	18
InShop	10	10	5	15

5 Field Study on Popular Datasets

Based on our EIF framework, we further investigate the generalization errors of the state-of-the-art models make on the popular datasets such as CUB200 [35], CARS196 [17], and InShop [18]. We investigate (1) *what does the generalization errors look like and how human agree with the errors?* (2) *what are the root causes for the erroneous model decision?* In this study, we choose Proxy-NCA++ [29] in this study as its leading performance in metric learning community.

Study Design For each dataset, we investigate two types of generalization errors: confusion pair and mis-similar pair. The mis-similar pair corresponds to the alternative definition in Section 2, i.e., the semantically similar pair with large distance. For each erroneous pair, we manually evaluate and classify them into: (1) agreeable error and (2) disagreeable error. For agreeable errors, humans agree with the model's decision. For disagreeable errors, humans disagree with the model. In this study, we recruit two volunteers (university graduate students majoring in computer science) to independently verify the reported confusion and mis-similar pairs. For the pairs where they disagree with each other, we let them discuss and reach a consensus. In addition, we use Algorithm 1 to identify and generate relabelling suggestions for the harmful training samples of the erroneous pairs. Based on the recommendation, we further confirm the recommendation and qualitatively analyze the reported problems in training datasets.

Generalization Errors and Their Agreeability Table 4 shows that human may share a considerable number of generalization errors with the model. Overall, human agrees with 51.6% (31 out of 60) of the confusion pairs and 30% (18 out of 60) of the mis-similar pairs. We show agreeable and disagreeable confusion pairs in Figure 3. Overall, human investigators agree with many of the "erroneous" model decisions. Readers can check mis-similar pairs on our anonymous website [2].

Root Cause of Erroneous Decision Table 5 shows that EIF generates relabelling suggestion for 41% (41 out of 100) training classes in CUB200 dataset, 46.9% (46 out of 98) training classes in the CARS196 dataset, and 22.5% (901 out of 3997) training classes in the InShop dataset. Moreover, we further investigate the training classes with more than 10% of their samples recommended to change their labels. We can see that, compared to CUB200 and CARS196, the InShop dataset has more confusing training classes. Figure 4 show some relabelled training samples in the InShop dataset.

We further manually sample the training classes with relabelling recommendations, regarding the following criteria:

- **Centralized Relabelling**: The training classes with more than 10% samples are recommended to be relabelled, and the recommended labels lean towards a single label.
- **Diversified Relabelling**: The training classes with more than 10% samples are recommended to be relabelled, and the recommended labels lean towards diversified labels.

(a) under class 143 (confused with Figure 3b) (b) under class 140 (confused with Figure 3a) (c) under class 100 (confused with Figure 3d (d) under class 143 (confused with Figure 3c

Figure 3: Figure 3a and Figure 3b are reported as a confusion pair but human investigator agree with the mode decision; Figure 3c and Figure 3d are reported as a confusion pair and human investigator disagree with the mode decision (the birds can be distinguished by the color of their heads).

Table 5: Relabelling suggestions on the popular datasets. The InShop dataset has 803 classes where over 10% of the samples recommended to be relabelled.

Dataset	**#class**	**#class with relabelling suggestions**	
		total	with significance (>10%)
CUB200	100	41	5
CARS196	98	46	12
InShop	3997	901	803

- **Individual Relabelling**: The training classes with less than 5% samples are recommended to be relabelled.

We distinguish centralized and diversified relabelling cases by introducing a threshold th_H (we use 0.35 in this study). Given the entropy of the relabelling-class distribution of a class $\mathcal{C}$ as H_c, if $H_c < th_H$, we consider C as a centralized relabelling class; otherwise, we consider C as a diversified relabelling class. Generally, a dataset, if with centralized relabelling requirements, needs to have its relevant training classes merged. In contrast, a dataset, if with diversified and individual relabelling requirements, needs to clean the samples under the relevant training classes.

We sample 3 training classes from each category on each dataset and report the confirmed relabelled suggestions in Table 6. Overall, EIF achieves high recommendation accuracy on the training classes with centralized and diversified relabelling in CUB200 and CARS196; and acceptable accuracy on the training classes with individual relabelling classes in CARS196 and InShop. We provide more details on our anonymous website [2].

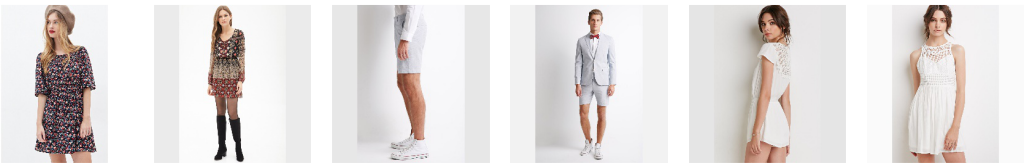
Summary and Discussion In this study, we conclude that the following problems are universal among the popular DML training and testing datasets:

- The testing dataset includes a number of arguably confusing samples, thus an "erroneous" model decision by the labels of testing samples may not necessary be *really* erroneous.
- Some classes are confusing with each other, i.e., over 10% of the samples have the potential to be merged into other classes.
- Many training classes involve outliers that look very different from other samples in the same class, but similar to other classes.

We provide more detailed examples on our website [2]. We argue that those dataset problems are one of the most important barriers to further improve new state-of-the-art DML approaches. The future work of DML should revolve around dataset cleaning and merging to improve the metrics in a more significant manner.

6 Related Work

Deep Metric Learning Deep metric learning (DML) learns an embedding space such that intra-class samples are located closer than inter-class samples. Loss functions are usually the key for learning such an embedding, which has been evolved from pairwise-based loss such as [11], [12],



(a) under class 24 (b) under class 652 (c) under class 28 (d) under class 2961 (e) under class 99 (f) under class 2216

Figure 4: Samples in the InShop training dataset, which look similar but under different class. Figure 4a is recommend to relabel to class 652, Figure 4c is recommend to relabel to class 2961, Figure 4e is recommend to relabel to class 2216,

Table 6: Manually verified relabelling suggestions on the datasets

Type	**Manual Evaluation**		
	CUB200	CARS196	InShop
Centralized Relabelling	100%	100%	66.67%
Diversified Relabelling	92.59%	100%	50.00%
Individual Relabelling	66.67%	85.71%	80.00%

[7], and [27] to proxy-based loss such as [19, 29, 6, 32, 8, 22, 13]. Proxy-based DML approach now leads the world-record, and Proxy-NCA++ is the representative.

While new approaches emerge to outperform the state-of-the-art with marginal improvements [20, 10, 23, 38], few work has been proposed towards understanding of generalization error in DML, from the dataset perspective. This work makes the first step, and our findings shed light on the potential problems of the datasets. Moreover, our EIF technique can further facilitate their fixes.

Model Explanation Despite its great successes in multiple disciplines, the deep learning model has remained to be black-box mystery for decades. There are two types of explanations: feature-level and instance-level. While feature-level explanations would like to interpret the semantics and importance of features, instance-level explanations would like to quantify the individual training sample's contribution to prediction. In [14], the idea of influence function is introduced to measure the change in testing loss upon removal of certain training sample. Variations of influence functions are later developed to solve the overestimation for outliers [3], the low diversity in high-influence points [4], the high computaional cost in Hessian estimation [24, 21]. RPS [37] proposes an alternative view from the Representer Point Theorem: they use the weighted kernels of training points as the influence measure. Since RPS has restrictions on model regularizers, RPS-LJE [28] has been later proposed to generalize RPS to models without regularization. However, as stated by the authors of RPS-LJE, the final definition has subtle differences to the original influence function [14].

Influence function has been applied to various tasks such as VAE [16], GAN [30], data poisoning [9], causal inference [1], data subsampling [31, 34, 33], data relabelling [15]. As far as we know, we are the first work which designs influence function catering to DML problems.

7 Conclusion

In this work, we design an empirical influence function (EIF) to debug and understand the generalization errors in state-of-the-art deep metric learning models. Comparing to the traditional influence function, EIF can (1) guide us to locate the influential harmful and helpful training samples and (2) recommend the potential relabelling suggestion for the harmful training samples. Our extensive experiments have proved its effectiveness. With the support of EIF, we further identify the problems of existing datasets for metric learning, which suggests the improvement of the dataset for achieving further world-record performance.

References

[1] A. Alaa and M. Van Der Schaar. Validating causal inference models via influence functions. In *International Conference on Machine Learning*, pages 191–201. PMLR, 2019.

[2] A. Anonymous. Anonymous webiste for empirical influence function. `https://sites.google.com/view/empirical-influence-function/home`. Accessed: 2022-01-28.

[3] E. Barshan, M.-E. Brunet, and G. K. Dziugaite. Relatif: Identifying explanatory training samples via relative influence. In *International Conference on Artificial Intelligence and Statistics*, pages 1899–1909. PMLR, 2020.

[4] U. Bhatt, I. Chien, M. B. Zafar, and A. Weller. Divine: Diverse influential training points for data visualization and model refinement. *arXiv preprint arXiv:2107.05978*, 2021.

[5] D. Bouchacourt, R. Tomioka, and S. Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[6] M. Boudiaf, J. Rony, I. M. Ziko, E. Granger, M. Pedersoli, P. Piantanida, and I. B. Ayed. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *European Conference on Computer Vision*, pages 548–564. Springer, 2020.

[7] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2017.

[8] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[9] M. Fang, N. Z. Gong, and J. Liu. Influence function based data poisoning attacks to top-n recommender systems. In *Proceedings of The Web Conference 2020*, pages 3019–3025, 2020.

[10] G. Gu, B. Ko, and H.-G. Kim. Proxy synthesis: Learning with synthetic classes for deep metric learning. *arXiv preprint arXiv:2103.15454*, 2021.

[11] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[12] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

[13] S. Kim, D. Kim, M. Cho, and S. Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3238–3247, 2020.

[14] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

[15] S. Kong, Y. Shen, and L. Huang. Resolving training biases via influence-based data relabeling. In *International Conference on Learning Representations*, 2021.

[16] Z. Kong and K. Chaudhuri. Understanding instance-based interpretability of variational auto-encoders. *Advances in Neural Information Processing Systems*, 34, 2021.

[17] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[18] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[19] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017.

[20] Y. Patel, G. Tolias, and J. Matas. Recall@ k surrogate loss with large batches and similarity mixup. *arXiv preprint arXiv:2108.11179*, 2021.

[21] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930, 2020.

[22] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6450–6458, 2019.

[23] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *International Conference on Machine Learning*, pages 8242–8252. PMLR, 2020.

[24] A. Schioppa, P. Zablotskaia, D. Vilar, and A. Sokolov. Scaling up influence functions. *arXiv preprint arXiv:2112.03052*, 2021.

[25] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[26] J. Seidenschwarz, I. Elezi, and L. Leal-Taixé. Learning intra-batch connections for deep metric learning. *arXiv preprint arXiv:2102.07753*, 2021.

[27] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016.

[28] Y. Sui, G. Wu, and S. Sanner. Representer point selection via local jacobian expansion for post-hoc classifier explanation of deep neural networks and ensemble models. *Advances in Neural Information Processing Systems*, 34, 2021.

[29] E. W. Teh, T. DeVries, and G. W. Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 448–464. Springer, 2020.

[30] N. Terashita, H. Ohashi, Y. Nonaka, and T. Kanemaru. Influence estimation for generative adversarial networks. *arXiv preprint arXiv:2101.08367*, 2021.

[31] D. Ting and E. Brochu. Optimal subsampling with influence functions. *Advances in neural information processing systems*, 31, 2018.

[32] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[33] T. Wang, J. Huan, and B. Li. Data dropout: Optimizing training data for convolutional neural networks. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 39–46. IEEE, 2018.

[34] Z. Wang, H. Zhu, Z. Dong, X. He, and S.-L. Huang. Less is better: Unweighted data subsampling via influence function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6340–6347, 2020.

[35] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[36] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017.

[37] C.-K. Yeh, J. S. Kim, I. E. Yen, and P. Ravikumar. Representer point selection for explaining deep neural networks. *arXiv preprint arXiv:1811.09720*, 2018.

[38] W. Zhao, Y. Rao, Z. Wang, J. Lu, and J. Zhou. Towards interpretable deep metric learning with structural matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9887–9896, 2021.