Data Pruning Can Do More: A Comprehensive Data Pruning Approach for Object Re-identification

Anonymous authors
Paper under double-blind review

Abstract

Previous studies have demonstrated that not each sample in a dataset is of equal importance during training. Data pruning aims to remove less important or informative samples while still achieving comparable results as training on the untruncated dataset, thereby reducing storage and training costs. We present the first data pruning approach in an object re-identification (ReID) setting. By fully leveraging the logit history during training, our approach offers a more accurate and comprehensive metric for quantifying sample importance, as well as correcting mislabeled samples and recognizing outliers. Furthermore, our approach is highly efficient, reducing the importance score estimation cost by 10 times compared to existing methods. Our approach is a plug-and-play, architecture-agnostic framework that can eliminate/reduce 35%, 30%, and 5% of samples/training time on the standard VeRi, MSMT17 and Market1501 datasets, respectively, with negligible loss in accuracy ($< 0.1\%$). The lists of important, mislabeled, and outlier samples from these ReID datasets will be released upon acceptance.

1 Introduction

Object re-identification (ReID) is a computer vision task that aims to identify the same object across multiple images. ReID has a wide range of applications in security surveillance (Khan et al., 2021), robotics (Wang et al., 2019b), and human-computer interaction (Wang et al., 2019a) among others. Similar to several other downstream computer vision applications, the performance of ReID algorithms is contingent upon the quality of data. ReID datasets are constructed by first collecting a pool of bounding boxes containing the intended object (e.g., people, vehicles); these images are typically obtained by applying category-specific object detectors to raw videos (Zheng et al., 2015; Fu et al., 2022). As a final step, the bounding boxes are assigned identity labels by manual annotators. Due to the mostly fixed position of cameras, extracted bounding boxes from a sequence of images can contain less information as these bounding boxes are likely to have the same background, lighting condition, and similar actions, e.g., Fig. 1(c). Such *less informative* samples do not provide additional value to model training. Instead, they increase the training time leading to lowered training efficiency (Toneva et al., 2018; Feldman & Zhang, 2020; Paul et al., 2021; Sorscher et al., 2022). Furthermore, ReID datasets are known to have a significant number of noisy labels and outliers (Yu et al., 2019; Yuan et al., 2020), which can ultimately lead to reduced model performance.

The issues of less informative samples and noise are not limited to ReID datasets. Several methods (Toneva et al., 2018; Feldman & Zhang, 2020; Paul et al., 2021; Sorscher et al., 2022) have shown that a significant portion of training samples in standard image classification benchmarks (Krizhevsky et al., 2009; Deng et al., 2009) can be pruned without affecting test accuracy. To this end, various data pruning approaches (Sener & Savarese, 2017b; Ducoffe & Precioso, 2018; Paul et al., 2021; Yang et al., 2022; Sorscher et al., 2022) have been proposed to remove unnecessary samples. A standard data pruning workflow is depicted in Fig. 1(a): 1) One or multiple models are trained on a raw dataset. 2) After training for some or all epochs, the importance score of each sample is estimated based on difficult metrics, e.g., EL2N (Paul et al., 2021) - L2 norm of error, forgetting score (Toneva et al., 2018) - the number of times a sample has been forgotten during

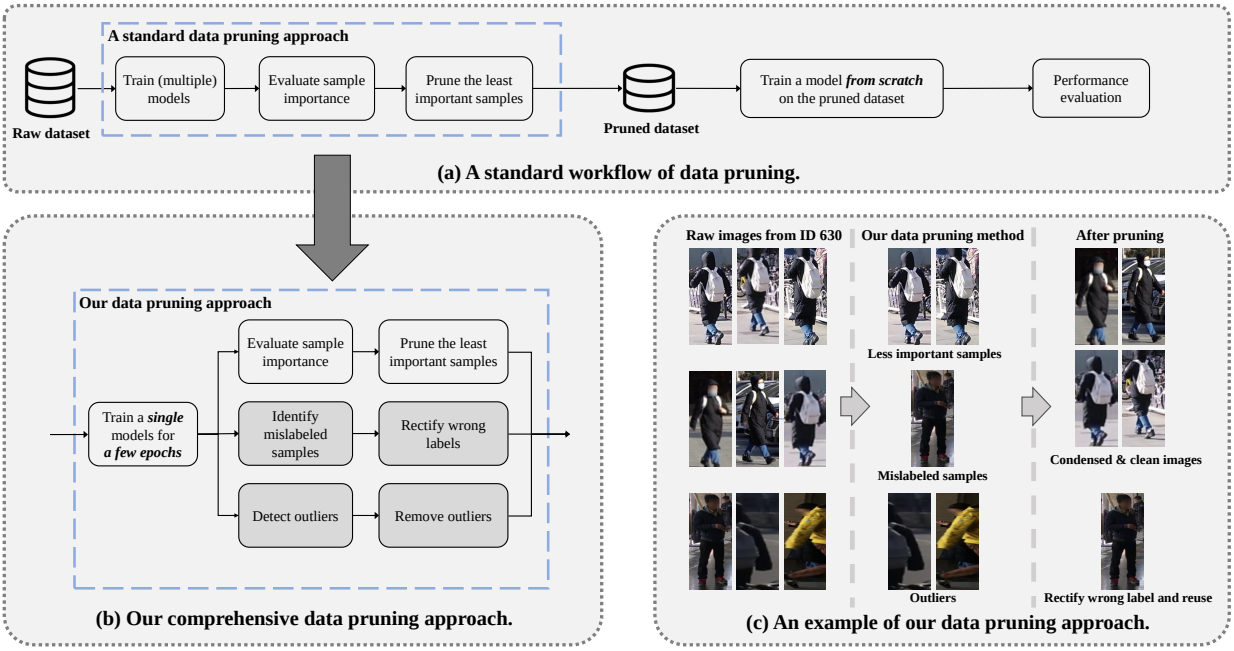


Figure 1: The workflow of data pruning. Our data pruning approach not only identifies *less important* samples, but also rectifies *mislabeled* samples and removes *outliers* (boxes highlighted in grey). Figure (c) demonstrates an example of our method, where all images are from the same person (id 630, MSMT17 dataset). Our approach serves as a "pre-processing" step, reducing the dataset size to save storage and training costs of ReID models while having minimal impact on their accuracy.

the training process. In general, the more difficult the sample, the more important it is (Toneva et al., 2018; Paul et al., 2021; Yang et al., 2022). 3) The samples with low importance scores are considered as easy or less informative samples and are discarded (Toneva et al., 2018; Feldman & Zhang, 2020; Paul et al., 2021). The first to third steps constitute a complete data pruning approach. 4) As a final step, we train the model *from scratch* on the pruned dataset and evaluate its performance. Although standard data pruning methods have yielded notable achievements, these standard methods primarily emphasize the removal of easy samples and do not explicitly discard noisy samples or rectify mislabeled samples (Toneva et al., 2018; Paul et al., 2021; Sorscher et al., 2022).

Furthermore, the majority of these data pruning methods quantify the importance of samples at a specific training step (Paul et al., 2021; Yang et al., 2022; Sorscher et al., 2022) and tend to overlook the sample's behavior during the training, i.e., how the logits (un-normalized class probability scores) vary and evolve across epochs during training. We illustrate such a scenario in Fig. 2 and observe that the EL2N score (Paul et al., 2021) cannot differentiate between two *hard* samples, i.e., (b) and (c), as it relies solely on the model's prediction at the *last* training epoch, discarding the sample's training history. On the other hand, the forgetting score (Toneva et al., 2018) estimates the importance score by recording how many times the model *forgets* this sample, i.e., how many times a sample experiences a transition from being classified correctly to incorrectly during training. However, the forgetting score is still coarse-grained, as it exclusively focuses on tracking the number of forgetting events and does not leverage the full training information.

To address current limitations, we propose a comprehensive data pruning approach, which not only prunes less informative samples more efficiently, but also corrects mislabeled samples and removes outliers in an end-to-end framework, as shown in Fig. 1(b). We hypothesize that instead of quantifying sample importance using discrete events (Toneva et al., 2018) or coarse inter-class relations (Paul et al., 2021; Sorscher et al., 2022) during training, fully leveraging the logit trajectory offers more insights. Accordingly, we propose a novel data pruning metric by fully utilizing the logit trajectory during training. We demonstrate the effectiveness

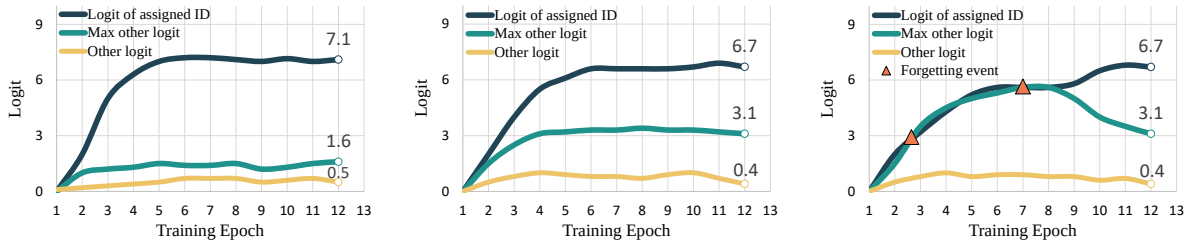
of our method on three standard ReID benchmarks (Zheng et al., 2015; Wei et al., 2018; Liu et al., 2016) and a classification dataset CIFAR-100 (Krizhevsky et al., 2009). Let us summarize our contributions:

1. We propose a novel pruning metric that utilizes soft labels generated over the course of training to estimate robust and accurate importance score for each sample.
2. Our method demonstrates significant efficiency, reducing the cost of importance score estimation by 10 times compared to existing state-of-the-art methods.
3. Our proposed framework not only removes less informative samples but also corrects mislabeled samples and eliminates outliers.
4. We benchmark our comprehensive data pruning approach on three standard ReID datasets. It eliminates/reduces 35%, 30%, and 5% of samples/training time on VeRi, MSMT17 and Market1501 dataset, respectively, with negligible loss in accuracy ($< 0.1\%$).

2 Related Works

Data Pruning aims to remove superfluous samples in a dataset and achieve comparable results when trained on all samples (Toneva et al., 2018; Feldman & Zhang, 2020; Paul et al., 2021; Sorscher et al., 2022). Different to dynamic sample selection (Chen et al., 2017; Hermans et al., 2017) and online hard example mining (Shrivastava et al., 2016), which emphasize on informative sample selection for a mini-batch at each iteration, (static) data pruning aims to remove redundant or less informative samples from the dataset in one shot. Such methods can be placed in the broader context of identifying coresets, i.e., compact and informative data subsets that approximate the original data (Sener & Savarese, 2017a). K-Center (Sener & Savarese, 2017b) and supervised prototypes (Sorscher et al., 2022) employ geometry-based pruning methods, which estimate the importance of a sample by measuring its distance to other samples or class centers in the feature space. The forgetting score (Toneva et al., 2018) tracks the number of "forgetting events", as illustrated in Fig. 2 (c). GraNd and EL2N scores (Paul et al., 2021) propose to estimate the sample importance using gradient norm and error L2 norm with respect to the cross-entropy loss. The main limitations of these approaches are: (i) their importance estimations may be highly affected by randomness due to underestimation of the influence of the whole training history, leading to insufficient accuracy; and (ii) the significantly high computational overhead for the metric estimation required by these methods. For instance, the forgetting score (Toneva et al., 2018) and supervised prototype score estimation (Sorscher et al., 2022) require an additional number of training epochs equivalent of the whole training time.

Object Re-identification (ReID) aims at identifying specific object instances across different viewpoints based primarily on its appearance (Zheng et al., 2016; Yadav & Vishwakarma, 2020; Zahra et al., 2022). ReID datasets (Zheng et al., 2015; Liu et al., 2016; Wei et al., 2018) consist of images of the object instances from multiple viewpoints typically extracted from image sequences. Therefore, images from the same sequence can often exhibit high similarity, i.e., the same background, consistent lighting conditions, and mostly indistinguishable poses. Furthermore, due to the subjective nature of manual annotation, ReID datasets often suffer from label noise and outliers (Yu et al., 2019; Yuan et al., 2020), such as heavily occluded objects and multi-target coexistence. During training, ReID tasks share similarity with the image classification task - given a dataset with a finite number of classes or identities, and models are trained to differentiate these distinct identities (Zheng et al., 2016; Yadav & Vishwakarma, 2020). As a result, the majority of ReID methods adopt the classification loss as an essential component (Wu et al., 2019; Si et al., 2019; Quispe & Pedrini, 2019; He et al., 2020). Owing to such similarity, several existing data pruning methods (Toneva et al., 2018; Paul et al., 2021; Sorscher et al., 2022) designed for image classification can directly be applied to ReID. As a first step, we apply existing pruning methods (Toneva et al., 2018; Paul et al., 2021) designed for the image classification task to ReID tasks. We observe that a portion of ReID datasets can be pruned without any noticeable loss in accuracy. Next, we propose a novel approach that offers improvements in terms of importance score estimation and computational overhead over existing methods. To the best of our knowledge, our work is the first to comprehensively study the impact of data pruning in a ReID setting.



(a) Easy sample.
Forgetting score: 0
EL2N score: 0.01
Ours: $H =$ 0.09

(b) Hard sample.
Forgetting score: 0
EL2N score: 0.04
Ours: $H =$ 0.30

(c) Harder sample.
Forgetting score: 2
EL2N score: 0.04
Ours: $H =$ 0.66

Figure 2: Logit trajectories for three samples (i.e., the evolution of the log probabilities of each sample belonging to each class over the course of training): (a) easy sample, (b) hard sample and (c) harder sample. There are three classes in total and each logit trajectory corresponds to one of such classes. The difference between the logits for each class can reflect the level of difficulty of this sample. In general, the more difficult the sample, the more important it is. The forgetting scores cannot distinguish (a) and (b), while the EL2N score relies solely on the model's prediction at the last epoch (thus without considering the history or "training dynamics"), hence it cannot differentiate between (b) and (c). Our approach fully exploits the training dynamics of a sample by utilizing the *average* logits values over all epochs to generate a more robust soft label. Then, the entropy of this soft label is employed to summarize the importance of a sample.

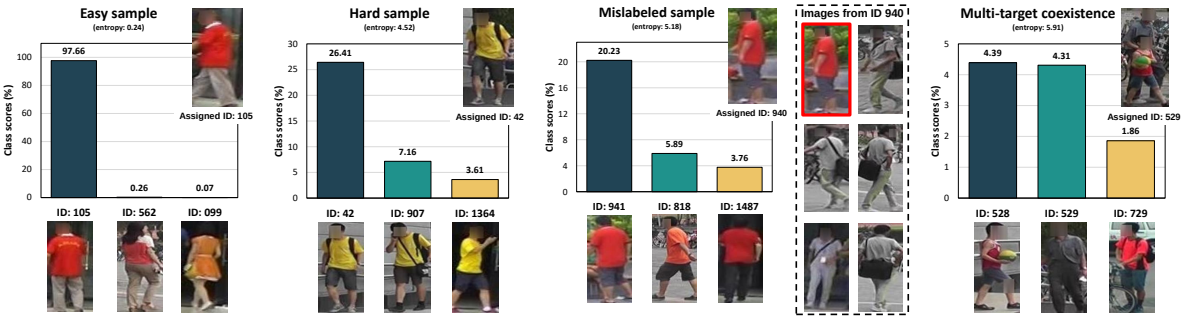
3 Identifying Important Samples

3.1 Preliminaries

Our work focuses on supervised ReID. Similar to image classification (Zeng et al., 2014; Rawat & Wang, 2017), a standard ReID network includes a feature extraction backbone, e.g., CNN-based (He et al., 2016) or transformer-based model (Dosovitskiy et al., 2021), followed by a classification head (Song et al., 2019; Liao & Shao, 2022). The backbone network extracts representations from images of persons (Wei et al., 2018) or vehicles (Liu et al., 2016). The classification head, typically a fully connected layer, maps these extracted features to class labels, which in ReID refer to the individuals or vehicles in the respective dataset. The training of ReID model varies from its classification counterpart in its use of loss functions. In addition to the standard cross-entropy loss, ReID training often involves metric losses, e.g., triplet loss (Schroff et al., 2015) and center loss (MacDonald, 2013), to learn more discriminative features: $L_{\text{ReID}} = L_{\text{CE}} + \alpha L_{\text{metric}}$, where α is a hyper-parameter to balance the loss terms.

3.2 Motivation

Typically, data pruning approaches estimate the sample importance after training for several or all epochs. However, the majority of data pruning methods do not fully exploit or take into account the training dynamics in evaluating sample importance. We observe that underestimating the training dynamics can lead to less accurate and unreliable estimations of the sample's importance. Figure 2 illustrates the trajectories of logits for three samples: (a) a simple sample, (b) a hard sample, and (c) a 'harder' sample. Let us consider a classification task with three distinct classes, and each logit trajectory corresponds to one class. The difficulty of a sample can be inferred by analyzing the discrepancy in logit values among different classes. In general, the more difficult the sample, the more important it is (Toneva et al., 2018; Paul et al., 2021; Yang et al., 2022). EL2N (Paul et al., 2021) and other geometry-based methods (Sorscher et al., 2022; Yang et al., 2022) utilize the output of a trained model after a few or all training epochs to evaluate the importance of samples. However, a comprehensive evaluation of importance cannot be achieved solely based on the model's output at a single epoch. As shown in Fig. 2 (b) and (c), although the last output logits of two hard samples are equal, the difficulties of learning them should not be equal: the model can correctly classify sample (b) from the early stage of training, while it struggles significantly and even mis-classifies sample (c) before the



(a) Easy sample: $H = 0.24$ (b) Hard sample: $H = 4.52$ (c) Mislabeled sample: $H = 5.18$ (d) Outlier: $H = 5.91$

Figure 3: Illustration of the generated soft labels averaged over 12 training epochs for different sample types (a)–(d) and their entropies. Multi-target coexistence (d) is one such type of outlier. We show the top 3 identities. Notably, our soft label can accurately indicate the ground-truth label of the mislabeled sample (c) without being influenced by the erroneous label. Images are from Market1501 (Zheng et al., 2015) dataset.

final training stage. Consequently, sample (c) in Fig. 2 should be considered more difficult or important. Inspired by this observation, we fully exploit the complete logit trajectories in order to achieve a more robust importance estimation, which we verify empirically. To be specific, for each sample, we leverage the complete logit trajectory to generate a soft label, which captures the relationship between a sample and all target classes. The entropy of the sample's soft label can represent the degree of confusion between different classes. A sample with high entropy could be wavering between different classes or a fuzzy boundary sample, thus making it more difficult or important.

3.3 Methodology

Let $(\mathbf{x}, y) \in \mathcal{D}_{\text{train}}$ be an image-label pair and $\mathbf{z}^{(t)}(\mathbf{x}) \in \mathbb{R}^c$ be logit vector of dimension c at epoch t, i.e., the un-normalized output of the classification head (pre-softmax). Here c refers to the number of classes in the training set. A logit value, presented on a logarithmic scale, represents the network's predicted probability of a particular class (Anderson et al., 1988), with the larger value indicating the higher class probability. We first calculate the average logits of a sample over all training epochs T. We then apply Softmax $\sigma(\cdot)$ to generate the soft label $\tilde{\mathbf{y}}$ of this sample, which portrays the relationship between this sample and all classes,

$$\tilde{\mathbf{y}} = \sigma\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{z}^{(t)}(\mathbf{x})\right) \in \mathbb{R}^c. \tag{1}$$

This soft label $\tilde{\mathbf{y}}$ summarizes the model's predictions at each training epoch. We conjecture that this soft label can effectively and accurately depict the ground truth characteristics of a sample. Next, we quantify the importance score of a sample using the soft label entropy,

$$H(\tilde{\mathbf{y}}) = -\sum_{i=1}^{c} p_i \log_2 p_i, \tag{2}$$

where $\tilde{\mathbf{y}} = (p_1, \ldots, p_c)^\top$ and c is the total number of classes. The importance of a sample can generally be evaluated by the difficulty of this sample (Sener & Savarese, 2017b; Toneva et al., 2018; Paul et al., 2021). The entropy is a straightforward metric to quantify the difficulty of a sample: A sample with a small entropy value implies that it contains the typical characteristics of a single class, and the classifier can easily distinguish this sample. In contrast, samples with larger values of entropy could have different characteristics of more than one class at the same time (e.g., samples on classification boundary), thus making it more difficult or important. Figure 3 illustrates our generated soft labels and corresponding entropies of several typical samples. Please refer to Appendix E for more examples.

The work of AUM (Pleiss et al., 2020) leverages training dynamics but for the identification of noisy samples in a dataset. The *area under margin* or AUM is defined as the difference between the logit values for a sample's assigned class and its highest non-assigned class, averaged over several epochs (Pleiss et al., 2020). This metric primarily captures the assigned class-related knowledge while ignoring the information across non-assigned classes. However, knowledge from non-assigned classes can potentially be of greater importance than that from the assigned class, as demonstrated in (Zhao et al., 2022). Different from AUM, our method utilizes information from all classes to depict a comprehensive relationship between a sample and the label space. Additionally, as the AUM metric does not fully encapsulate the characteristics of the sample, it is only capable of identifying mislabeled samples, but cannot correct them. We elaborate further on our approach for label correction and outlier detection in the next section.

4 Data Purification

Although most samples with high importance scores contribute significantly to model training, some high-scoring samples may be noisy (Feldman & Zhang, 2020; Paul et al., 2021) and lead to a reduction in model performance. To address this limitation, in this section we further exploit the generated soft label to "purify" samples in two ways: by correcting mislabeled samples and by eliminating outliers.

Dataset Noise. Noise in ReID datasets can be broadly categorized into (Yu et al., 2019) - (i) label noise, i.e., mislabeled samples, caused by human annotator errors, and (ii) data outliers caused by object detector errors, e.g., heavy occlusion, multi-target coexistence, and object truncation (some outlier examples are presented in Fig. 14). Our method aims to reuse mislabeled samples by label correction and eliminating outliers.

Correcting Mislabeled Samples. We hypothesize that the soft label generated from the average logits of a sample can reflect the ground-truth characteristics of this sample more accurately. Accordingly, we directly utilize soft label $\tilde{\mathbf{y}}$ from Eq. 1 to identify whether a sample is mislabeled or not. When the assigned label differs from the class with the highest score in the soft label, i.e., $\text{argmax}(\tilde{\mathbf{y}}) \neq y$, we consider it is mislabeled. The ground-truth label of this sample is then accordingly corrected to $\text{argmax}(\tilde{\mathbf{y}})$. Indeed, our ablation experiments offer empirical evidence that accumulated logits over epochs lead to a much more accurate prediction of the ground-truth label than solely relying on logits from a single epoch (more details in Sec. 5.6).

Identifying Outliers. The entropy of the soft label reflects the difficulty of a sample, allowing us to infer whether it is an outlier or not. A straightforward approach is to set a threshold on the entropy - if the entropy of a sample surpasses the predefined threshold, this sample is likely an outlier. However, theoretically, this threshold of the entropy should be changed depending on the number of classes: the upper bound of the entropy is determined by the number of classes, i.e., $H(\tilde{\mathbf{y}}) \leq \log_2(c)$, rendering it unsuitable for use across datasets with different numbers of classes. Therefore, we employ the highest class score as an indicator to determine outliers. If the highest class score of a soft label is relatively low, i.e., $\max(\tilde{\mathbf{y}}) \leq \delta$, it is highly likely that this sample is an outlier. We conduct a sensitivity analysis to evaluate the impact of this threshold in Sec. 5.6 and observe our method is robust to this threshold δ.

5 Experiments

We demonstrate the effectiveness of our method via detailed experimentation. In Sec. 5.3, we evaluate our pruning metric on ReID and classification datasets, and verify that it effectively quantifies the sample importance. In Sec. 5.4, we train models on noisy datasets after correcting labels and removing outliers to validate the data purification ability. In Sec. 5.5, we present the results of our comprehensive data pruning method, which includes removing less important samples, correcting mislabeled samples, and eliminating outliers. Finally, we perform several detailed ablation studies in Sec. 5.6 to evaluate the impact of hyper-parameters and verify the importance of logit accumulation.

5.1 Datasets and Evaluation

Datasets. We evaluate our method across three standard ReID benchmarks: Market1501 (Zheng et al., 2015) and MSMT17 (Wei et al., 2018) for pedestrian ReID, and VeRi-776 (Liu et al., 2016) for vehicle ReID. Market1501 is a popular ReID dataset, which includes $32,668$ images featuring $1,501$ pedestrians, captured across 6 cameras. Compared to Market1501, MSMT17 is a more challenging and large-scale person ReID dataset(Wei et al., 2018). MSMT17 contains $126,441$ images from $4,101$ pedestrians. VeRi-776 is a widely used vehicle ReID benchmark with a diverse range of viewpoints for each vehicle; it contains $49,357$ images of 776 vehicles from 20 cameras.

Evaluation. Given a query image and a set of gallery images, the ReID model is required to retrieve a ranked list of gallery images that it best matches the query image. Based on this ranked list, we evaluate ReID models using the following metrics: the cumulative matching characteristics (CMC) at rank-1 and mean average precision (mAP), which are the most commonly used and standard evaluation metrics for ReID tasks (Bedagkar-Gala & Shah, 2014; Ye et al., 2021; Zheng et al., 2016). We adopt, as evaluation metric, the mean of rank1 accuracy and mAP (He et al., 2020), i.e., $\frac{1}{2}(\text{rank1} + \text{mAP})$, to present the results in a single plot.

5.2 Implementation Details

Estimating the Sample Importance Score. For ReID tasks, we follow the training procedure in (Luo et al., 2019): a ResNet50 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009) is used as the backbone and trained on a ReID dataset for 12 epochs to estimate sample importance. For optimization, we use Adam (Kingma & Ba, 2014) with a learning rate of 3.5×10^{-4}. We do not apply any warm-up strategy or weight decay. The batch size is set to 64. Following Luo et al. (2019), We use a combination of the cross-entropy loss and the triplet loss as the objective function. During training, we record the logit values of each sample after each forward pass. Then, we generate the soft label of each sample based on Eq. 1 and calculate its entropy as the importance score. For classification tasks, we use the model architecture and training parameters specified in Toneva et al. (2018). In contrast to the ReID tasks, we do *not* employ any pre-trained models in classification tasks, i.e., the model weights are randomly initialized. Please refer to Appendix A and B for more implementation details.

Data Pruning. Given the importance scores of all samples, we sort them to obtain a sample ordering from easy (low importance score) to difficult (high importance score). Next, given the pruning rate, we remove the corresponding number of easy samples from the original training set. For instance, if the pruning rate is 10%, we remove the top 10% of the easiest samples. Subsequently, following Luo et al. (2019), we train four ReID models from scratch on this pruned dataset independently, each with a different random seed, and report their mean accuracy. The total number of training iterations is directly proportional to the number of samples. Therefore, a reduction in sample size leads to a proportionate decrease in training time.

5.3 Find Important Samples

This section aims to verify the effectiveness of our proposed importance score by data pruning experiments. Because our work predominantly concentrates on data pruning applied to the ReID tasks, we demonstrate the efficacy of our proposed importance score on three ReID datasets in Sec. 5.3.1. Considering the substantial similarity between ReID and classification tasks, we also extend our data pruning experiments to a CIFAR dataset (Krizhevsky et al., 2009) in Sec. 5.3.2, thereby ensuring a more comprehensive validation of our method.

5.3.1 Data Pruning on ReID Datasets

We observe that the majority of data pruning methods have been evaluated on standard image classification datasets. However, to date, no pruning metric has been evaluated on ReID datasets. To this end, we perform a systematic evaluation of four pruning metrics on ReID datasets: EL2N score (Paul et al., 2021), forgetting score (Toneva et al., 2018), supervised prototypes (Sorscher et al., 2022), moving avg loss (Zhou et al., 2020) and our proposed metric. Following the standard protocol (Toneva et al., 2018; Sorscher et al., 2022), the

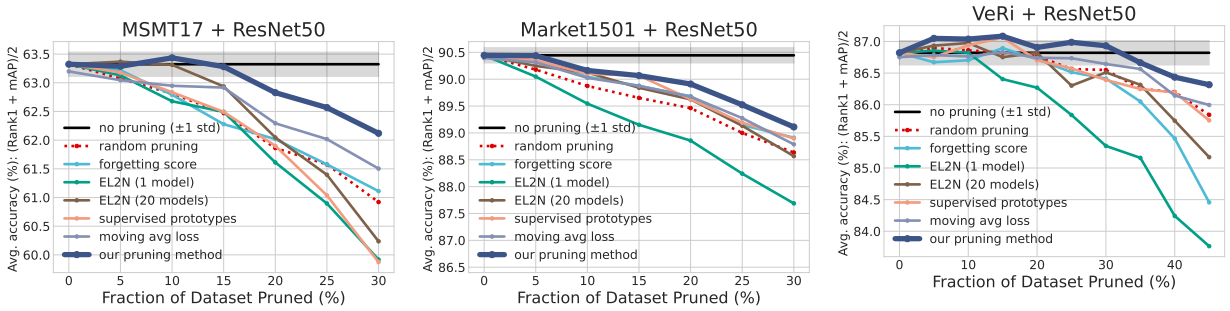


Figure 4: Data pruning on ReID datasets. We report the mean of Rank1 and mAP (He et al., 2020) on 3 ReID datasets (labeled at the top), obtained by training on the pruned datasets. For each method, we carry out four independent runs with different random seeds and report the mean.

Table 1: Extra training epochs and time needed for metric estimation on ReID dataset. The total training epochs for a ReID model is 120.

Method	EL2N(20 models)	Forgetting score	Supervised prototypes	Ours
Extra training epochs	240	120	120	**12**
Extra training time[1]	5.2 hours	2.6 hours	2.6 hours	**15.8 mins**

forgetting scores and supervised prototype scores are calculated at the end of training. The EL2N score (Paul et al., 2021) and our proposed metric are estimated early in training (i.e., after 12 epochs - which is 10% of the total training epochs and recommended by the standard EL2N protocol). For EL2N, we report two scores: one generated by a single trained model and the other one by 20 trained models, each trained with a different random seed. Training epochs for the importance score estimation of different methods are shown in Tab. 1. The training sets are constructed by pruning different fractions of the lowest-scored samples. In all experiments, training on the full dataset (*no pruning*) and a random subset of the corresponding size (*random pruning*) are used as baselines (Sorscher et al., 2022). We present the results of data pruning approaches on three different ReID datasets in Fig. 4.

Better Pruning Metric by Leveraging the Training Dynamics. The results in Fig. 4 validate our assumption that fully exploiting training dynamics can lead to a more precise and comprehensive evaluation of the sample importance. Our method surpasses all three competing methods across all datasets. The importance of incorporating the training dynamics is also observed in the forgetting score. Even though employed coarsely, the forgetting score outperforms single model EL2N and surpasses the random pruning baseline at low pruning rates on all datasets.

EL2N Score Suffers from Randomness. We observe that EL2N (1 model) under-performs on all three datasets (Fig. 4), being even worse than random pruning. The primary reason is that the EL2N score approximates the gradient norm of a sample at a certain epoch, and it assumes that the importance score of a sample remains static throughout the training process, ignoring the impact of evolving training processes. Disregarding the training dynamics leads to unreliable estimations that are highly susceptible to random variations. By training multiple models (20 models) and using the mean E2LN score, the performance improves significantly. However, it comes at the price of a notable growth in training time: it necessitates 20 times the training time required by our method.

Differences in Datasets. We notice considerable variations in the levels of sample redundancy across the three ReID datasets (Fig. 4). Using our method only allows for the removal of 5% of samples from Market1501 without compromising accuracy. While on MSMT and VeRi datasets our method can eliminate 15% and 30% of the samples, respectively. We hypothesize this is due to the rigorous annotation process of

[1]Extra training time is measured using a single NVIDIA V100 GPU on MSMT17 dataset.

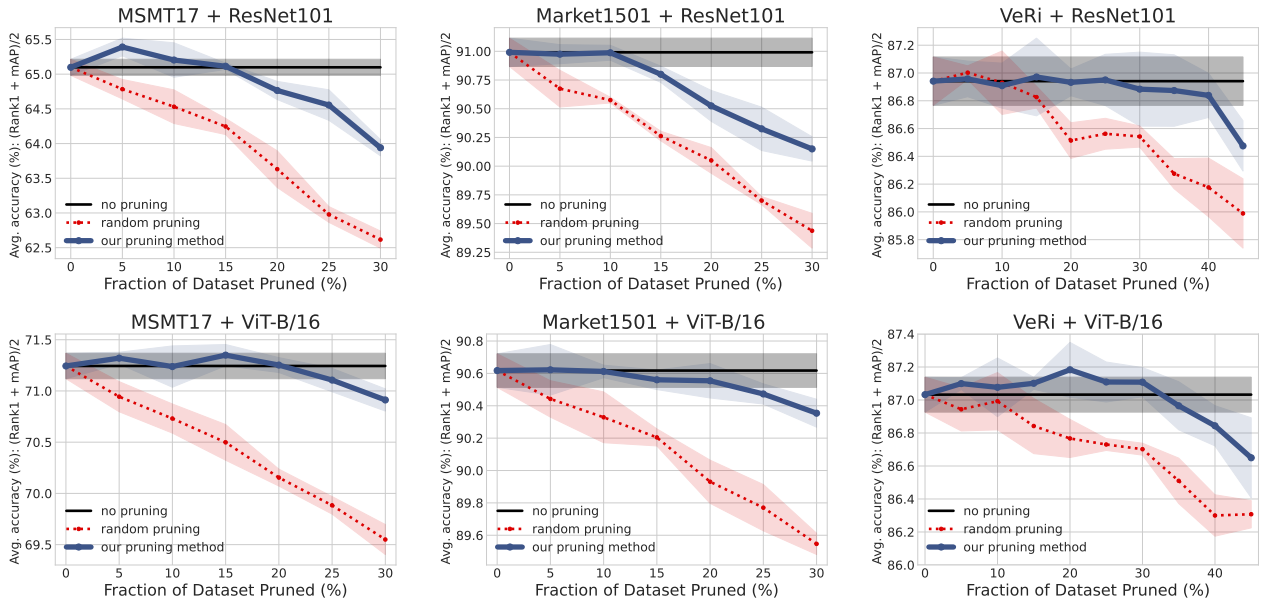


Figure 5: Generalization performance. We train a ResNet101 and a ViT model using the **sample ordering of ResNet50** on the MSMT17 dataset. For each method, we carry out four independent runs with different random seeds and we report the mean values. Shaded areas mean +/- one standard deviation of four runs.

Market1501 dataset. Additionally, Market1501 is considerably smaller in scale, i.e., Market1501 training set is merely 43% the size of MSMT17 and 34% the size of VeRi, thereby reducing the likelihood of redundant or easy samples.

Generalization from ResNet50 to Deeper ResNet and ViT. Previous experiments have demonstrated that our proposed metric can better quantify the importance of samples. Next, we validate if the ordering (ranking) obtained using a simpler architecture, i.e., ResNet50 (He et al., 2016), can also work with a more complex architecture, i.e., deeper ResNet and ViT (Dosovitskiy et al., 2021). To this end, we train a ResNet101 and a vision transformer ViT-B/16 on a pruned dataset using the sample ordering from ResNet50. We plot generalization performance of ResNet101 and ViT in Fig. 5. The network still performs optimally with even more than 15% samples from MSMT17 removed or 10% samples from Market1501 pruned. These results verify that our metric can reflect the ground-truth characteristics of samples, independent of the model architecture used for training.

5.3.2 Data Pruning on Classification Dataset

To comprehensively verify the effectiveness of our proposed metric, we conduct supplementary experiments on two classification datasets. We demonstrate the efficacy of our method on CIFAR-100 (Krizhevsky et al., 2009) and CUB-200-2011 (Wah et al., 2011) datasets and compare it with the forgetting (Toneva et al., 2018) and EL2N (Paul et al., 2021) scores in Fig. 6. For a fair comparison, all methods employ the same training time or cost to estimate the importance scores of samples. In detail, following the standard protocol of EL2N (Paul et al., 2021) we solely train a model for 20 epochs on CIFAR-100 and 3 epochs on CUB-200-2011, equivalent to 10% of all training epochs, and then apply three methods, i.e., forgetting, EL2N and our scores to estimate the sample importance. In line with the results in ReID tasks, our method consistently demonstrates superior performance over two competing methods. Notably, in the CIFAR-100 classification experiment, we intentionally exclude any pre-training methodologies; specifically, the model weights are randomly initialized. These results confirm the consistent efficacy of our approach, regardless of the utilization of pre-trained models.

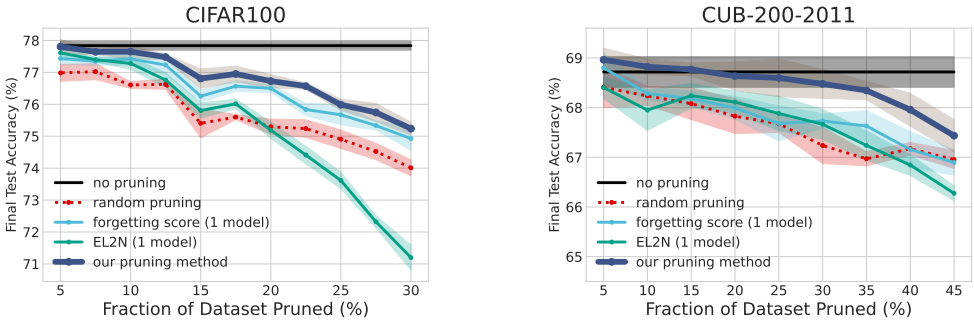


Figure 6: Data pruning on classification datasets. We report the mean of final test accuracy on two classification datasets (labeled at the top), obtained by training on the pruned datasets. For each method, we carry out four independent runs with different random seeds and report the mean.

5.4 Robust Training on Noisy Datasets

We test the efficacy of our data purification method on random noisy datasets. Herein, we start with the original images with the assigned labels. A certain percentage (i.e., from 10% to 50%) of training images are randomly selected and intentionally assigned the wrong labels. We compare our approach with a state-of-the-art method for identifying mislabeled samples, i.e., AUM (Pleiss et al., 2020), and evaluate each of our used components, i.e., label correction and removing outliers.

From Fig. 7, we observe that label correction plays a pivotal role, while merely removing outliers only slightly improves the performance. The reason is that outlier removal merely discards a few data points, leaving behind a substantial amount of mislabeled data, which ultimately results in minimal model improvement. Besides, we observed these two components are interdependent and complementary to each other, and the simultaneous utilization of them significantly boosts the performance.

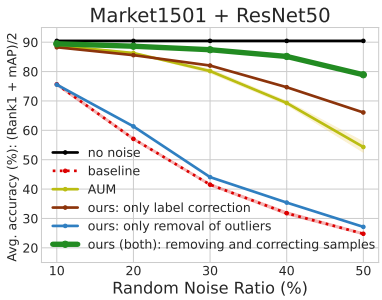


Figure 7: Data purification on noisy datasets. We report accuracy under different random noise ratios. Each curve represents the mean accuracy over four independent runs.

Our proposed data purification approach, *removing and correcting samples*, demonstrates superior performance when compared to the AUM metric (Pleiss et al., 2020). The rationale behind this advantage lies in the fact that the AUM metric tends to discard a greater amount of training data, whereas our method selectively removes fewer samples and capitalizes on mislabeled data by employing label correction to maximize the benefit of each sample.

5.5 Putting It All Together

We integrate data purification to build a comprehensive data pruning approach: easy samples (i.e., samples whose soft labels have low entropy) are pruned; outliers (i.e., samples whose highest class score is lower than 10%) are removed; samples with incorrect labels are rectified. We test our approach on three ReID datasets and evaluate the benefit of our data purification approach. In Fig. 8, we observe marked performance differences between datasets. Data purification on MSMT17 shows a notable effect, potentially due to the presence of more outliers and mislabeled samples in this dataset. The performance on VeRi is slightly improved as well. However, the impact on Market1501 is comparatively limited, plausibly owing to stricter annotation protocols and fewer outliers compared to the other datasets. In summary, the seamless integration of data purification and data pruning can effectively boost overall performance. Especially, on MSMT17 our approach can even eliminate/reduce 30% of samples/training time with almost no compromise in performance.

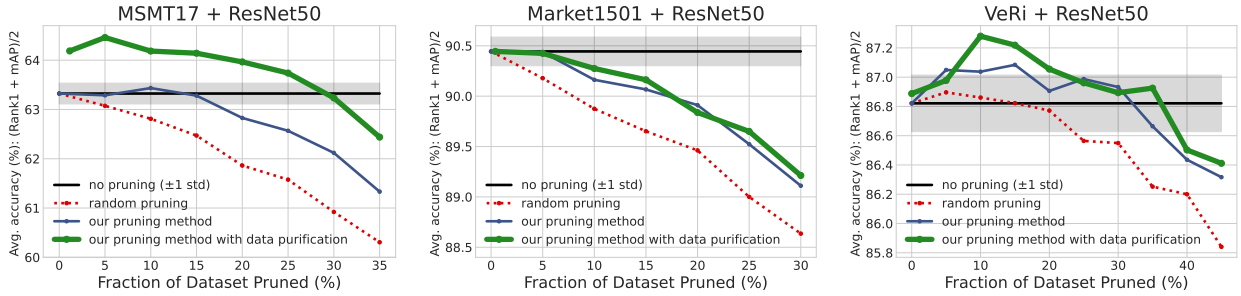


Figure 8: Data pruning with data purification. Accuracy is achieved by training on the pruned dataset (pruning ratios on X-axis). We report the mean values from four independent runs with different seeds.

5.6 Sensitivity Analysis and Ablation Studies

Our approach involves two hyper-parameters: (i) the number of training epochs required to generate the soft labels, i.e., T from Eq. 1, and (ii) the threshold δ to remove outliers. We *first* explore the impact of the hyper-parameter T. *Next*, we verify the importance of logit accumulation through ablation experiments, and *finally* analyze the effect of the threshold δ on the performance of data purification.

Impact of Score Computation Epoch T. We investigate how early in training is our metric effective at estimating the importance scores of samples. In Fig. 9, we compare the model performance from training on 85% training data but pruned based on the forgetting score (Toneva et al., 2018), EL2N score (Paul et al., 2021) and our scores computed at different epochs. All scores are estimated using the identical random seed. We observe that after 12 epochs, the model performance of using our pruning method essentially stabilizes and surpasses that of random pruning as well as other competing methods. Considering EL2N's vulnerability to randomness, a single model is often insufficient to precisely estimate sample importance. The forgetting score typically necessitates many training epochs (Toneva et al., 2018), thereby hindering its capability to accurately estimate sample importance in early training stages. Additionally, we explore the impact of score computation epoch T on label correction and present results in Appendix D.1. We observe training for 12 epochs is sufficient to achieve desirable results of label correction as well.

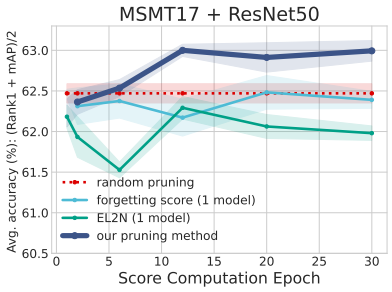


Figure 9: Impact of score computation epoch T. Model performance achieved by training on 85% training data comprised of examples with maximum forgetting, EL2N, and our scores computed at different epochs.

Importance of Logit Accumulation. We conduct two experiments to demonstrate the importance of logit accumulation in (i) importance score estimation for data pruning, and (ii) label correction for noisy label datasets. In both experiments, our soft labels are generated using different frequencies of logit accumulation, i.e., once at the 12^{th} epoch and every 1-6 epochs. For example, *"every 4 epochs"* means we use logits at the 4^{th}, 8^{th}, and 12^{th} epochs to calculate our proposed importance scores. Likewise, *"every 6 epochs"* means logits are accumulated on the 6^{th} and 12^{th} epoch. Figure 10 depicts the results of data pruning experiment, and we observe the progressive improvement in pruning performance with increasing frequency of logit accumulation. Consistent with the previous result, Fig. 11 illustrates that as more logits at different epochs are accumulated, the performance of label correction improves. Both results validate our claim that the soft label generated from leveraging the full logit trajectory can better capture the ground-truth characteristics of a sample.

Impact of Outlier Removal Threshold δ. In our method, samples whose highest class score is lower than δ are marked as outliers and removed (Sec. 4). We perform the ablation study to quantify of impact of δ under varying levels of noise. We keep the noise ratios fixed at 10%, 20% and 30% and vary δ. As shown

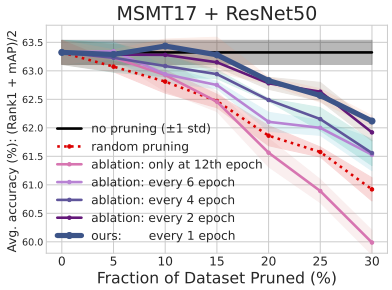


Figure 10: Ablation study of logit accumulation for *data pruning*. Model performance when trained with the importance scores computed using different frequencies of logit accumulation.

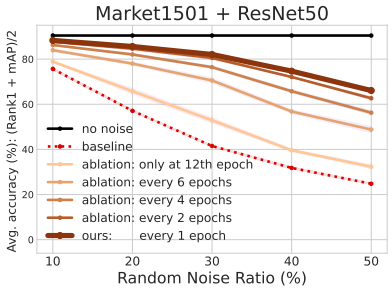


Figure 11: Ablation study of logit accumulation on *noisy datasets*. Model performance *only* using label correction and the soft labels are generated using different frequencies of logit accumulation.

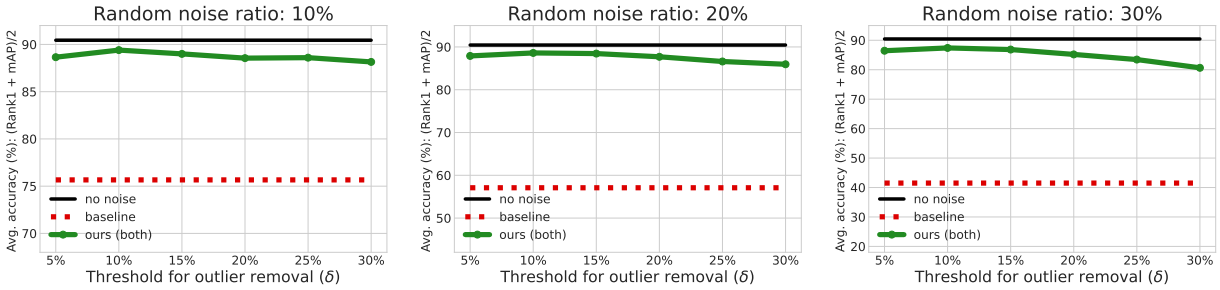


Figure 12: Impact of outlier removal threshold δ. Model performance under noise ratio 10%, 20% and 30%.

in Fig. 12, we observe that setting $\delta = 10\%$ yields slightly superior performance. Overall, δ has a limited impact on the final performance, especially when δ lies between 5% and 15%. It demonstrates our approach is robust and insensitive to the threshold values.

6 Discussion and Conclusion

In our work, we propose a plug-and-play architecture-agnostic data pruning framework and apply it to ReID. *Firstly*, by leveraging the training dynamics, we provide a more accurate and robust pruning metric with extremely low training overhead. Through the generalization test, we demonstrate that our metric reflects the ground-truth characteristics of samples independently of the model architecture used for training, i.e., the sample ordering (ranking) obtained via a ResNet remains effective in training a vision transformer. For completeness, we also benchmark our method for image classification on CIFAR-100 and we observe that our approach exhibits superior performance over competing methods as well. *Secondly*, we propose an efficient data purification method, enabling the correction of mislabeled samples and the removal of outliers. Experiments on the noisy datasets validate that our data purification method exhibits remarkable robustness, achieving impressive results. *Finally*, by integrating data purification we build a comprehensive data pruning framework and demonstrate that it achieves state-of-the-art performance when pruning ReID datasets, allowing for the removal of 35%, 30%, and 5% of samples from the VeRi, MSMT17, and Market1501 dataset respectively thereby leading to an equivalent reduction in training time. Like previous works (Toneva et al., 2018; Paul et al., 2021), our data pruning method requires labels. In future, we intend to investigate data pruning methods based on logit trajectories in an unsupervised setting.

References

Simon P Anderson, André De Palma, and J-F Thisse. A representative consumer theory of the logit model. *International Economic Review*, pp. 461–466, 1988.

Apurva Bedagkar-Gala and Shishir K Shah. A survey of approaches and trends in person re-identification. *Image and vision computing*, 32(4):270–286, 2014.

Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 403–412, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

Dengpan Fu, Dongdong Chen, Hao Yang, Jianmin Bao, Lu Yuan, Lei Zhang, Houqiang Li, Fang Wen, and Dong Chen. Large-scale pre-training for person re-identification with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2476–2486, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 2020.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

Samee Ullah Khan, Tanveer Hussain, Amin Ullah, and Sung Wook Baik. Deep-reid: Deep features and autoencoder assisted image patching strategy for person re-identification in smart cities surveillance. *Multimedia Tools and Applications*, pp. 1–22, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Shengcai Liao and Ling Shao. Graph sampling based deep metric learning for generalizable person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7359–7368, 2022.

Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 869–884. Springer, 2016.

Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Norman MacDonald. *Time lags in biological models*, volume 27. Springer Science & Business Media, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.

Rodolfo Quispe and Helio Pedrini. Improved person re-identification based on saliency and semantic parsing with deep neural network models. *Image and Vision Computing*, 92:103809, 2019.

Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017a.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017b.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

Tongzhen Si, Zhong Zhang, and Shuang Liu. Compact triplet loss for person re-identification in camera sensor networks. *Ad Hoc Networks*, 95:101984, 2019.

Jifei Song, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Generalizable person re-identification by domain-invariant mapping network. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 719–728, 2019.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Yujiang Wang, Jie Shen, Stavros Petridis, and Maja Pantic. A real-time and unsupervised face re-identification system for human-robot interaction. *Pattern Recognition Letters*, 128:559–568, 2019a.

Yujiang Wang, Jie Shen, Stavros Petridis, and Maja Pantic. A real-time and unsupervised face re-identification system for human-robot interaction. *Pattern Recognition Letters*, 128:559–568, 2019b.

Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 79–88, 2018.

Di Wu, Si-Jia Zheng, Wen-Zheng Bao, Xiao-Ping Zhang, Chang-An Yuan, and De-Shuang Huang. A novel deep model with multi-loss and efficient training for person re-identification. *Neurocomputing*, 324:69–75, 2019.

Ankit Yadav and Dinesh Kumar Vishwakarma. Person re-identification using deep learning networks: A systematic review. *arXiv preprint arXiv:2012.13318*, 2020.

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. *arXiv preprint arXiv:2205.09329*, 2022.

Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):2872–2893, 2021.

Tianyuan Yu, Da Li, Yongxin Yang, Timothy M Hospedales, and Tao Xiang. Robust person re-identification by modelling feature uncertainty. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 552–561, 2019.

Ye Yuan, Wuyang Chen, Yang Yang, and Zhangyang Wang. In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 354–355, 2020.

Asmat Zahra, Nazia Perwaiz, Muhammad Shahzad, and Muhammad Moazam Fraz. Person re-identification: A retrospective on domain specific open challenges and future trends. *arXiv preprint arXiv:2202.13121*, 2022.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pp. 2335–2344, 2014.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11953–11962, 2022.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*, 2015.

Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.

Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: from clean label detection to noisy label self-correction. In *International Conference on Learning Representations*, 2020.

A Implementation Details for ReID Tasks

A.1 Estimating the Importance Score of Samples from ReID datasets

All experiments are implemented in PyTorch (Paszke et al., 2019) using the FastReID (He et al., 2020) toolbox[1] on 4 NVIDIA Tesla V100 GPUs. A general workflow for supervised ReID is shown in Fig. 13. In our work, we follow the training procedure[2] of Luo et al. Luo et al. (2019). For feature extraction, we employ a ResNet50 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009); the model is trained for 12 epochs. All images from Market1501 and MSMT17 are resized to 256×128 pixels, while images from VeRi are resized to 256×256 pixels. During training, we record the logits for each sample after each forward pass. For optimization, we use a combination of the cross-entropy loss and the triplet loss, i.e., $L_{\text{train}} = L_{\text{ce}} + L_{\text{triplet}}$.

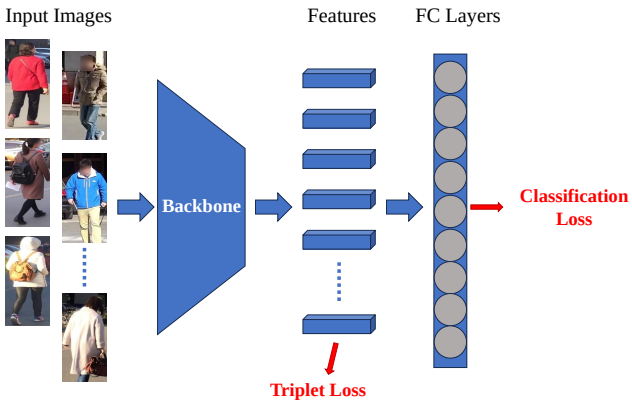


Figure 13: A general supervised ReID workflow.

A.2 Existing Methods

E2LN. We use the same model configuration and settings as in our approach, except for the loss function. When calculating the EL2N score, we strictly follow its procedure and definition (Paul et al., 2021), i.e., we only use the cross-entropy loss and do not utilize additional loss functions (e.g., metric loss). We train till 12 epochs and then compute the EL2N score for each sample. For EL2N (20 models), we run the same experiments 20 times but with different random seeds and obtain the score by averaging over the runs.

Forgetting Score. We train a single model for 120 epochs using the default settings[2]. During training, we record the number of forgetting events for each sample.

Supervised Prototypes. To ensure a fair comparison with other methods, we replace the self-supervised learning proposed in Sorscher et al. (2022) with supervised learning. Using the default settings and configuration[2], we train a single model for 120 epochs. After training, we calculate the supervised prototype score of each sample, which is defined by the L2 distance between a sample and its class centroid (Sorscher et al., 2022). Additionally, following Sorscher et al. (2022), we also apply a simple 50% class balancing ratio.

A.3 Data Pruning Experiments on ReID datasets

Following the default model configuration and settings[2], we train four independent ReID models using different random seeds on the pruned dataset, where the samples are pruned based on different types of importance score, i.e., the forgetting, EL2N, supervised prototype and our scores. We then report the mean performance across these four models.

[1] `https://github.com/JDAI-CV/fast-reid`

[2] `https://github.com/JDAI-CV/fast-reid/blob/master/configs/Base-bagtricks.yml` - Model configuration and hyper-parameter setting.

A.4 Generalization Experiments

We initialize a vision transformer ViT-B/16 (Dosovitskiy et al., 2021) with its weight pre-trained on ImageNet21K and use the default configuration[3]. The model is trained on a pruned dataset, which is pruned based on the ranking list of samples generated by a ResNet50. We report the average performance across four runs with different random seeds.

A.5 Robust Training on Noisy Datasets

To evaluate the performance of AUM, we follow their proposal (Pleiss et al., 2020) to fine-tune the threshold value using threshold samples. For our method, we set the threshold of outlier removal to $\delta = 10\%$ on all three ReID datasets. After the outlier removal and label correction, we train four ReID models using the default settings[2] and present the mean accuracy derived from these four runs.

A.6 Putting It All Together

In our proposed framework, samples are either removed or rectified in parallel:

- Removing easy samples, as described in Sec. 3.3.
- Rectifying mislabeled samples. Details are presented in Sec. 4.
- Eliminating outliers according to the highest class scores of the samples, as described in Sec. 4.

Please note that the total number of pruned samples equals the sum of outliers and the pruned easy samples.

B Implementation Details for Classification Tasks

For image classification on CIFAR-100, we begin with computing the forgetting score (Toneva et al., 2018), EL2N score (Paul et al., 2021), and our proposed metric for each sample. We use the model architecture and training parameters[4] specified in Toneva et al. (2018). For our approach, during training, we compute the average logit value for each sample across 20 epochs (10% of the total training epochs). After computing the individual metrics for each sample, we remove the corresponding number of easy samples from the original training set. For instance, if the pruning rate is 20%, we remove the top 20% of the easiest samples (samples with the lowest importance score). Following Toneva et al. (2018), we train four models with ResNet18 (He et al., 2016) backbone *from scratch* (i.e., weights are randomly initialized without the use of any pre-trained model) on this pruned dataset independently. For each run, we use a different random seed. Finally, we report the mean test accuracy across these four independent runs.

C Examples of different outliers from MSMT17.

In Fig. 14 we present three different types of outliers from MSMT17, i.e., heavy occlusion, multi-target coexistence and object truncation

(a) Heavy occlusion (b) Multi-target coexistence (c) Object truncation

Figure 14: Examples of different outliers from MSMT17.

[3] `https://github.com/JDAI-CV/fast-reid/blob/master/configs/Market1501/bagtricks_vit.yml` - ViT configuration
[4] `https://github.com/mtoneva/example_forgetting`

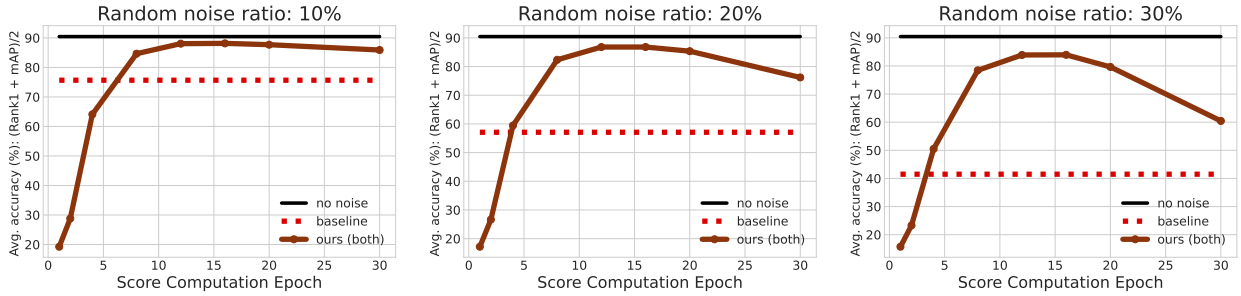


Figure 15: Accuracy achieved by training on the Market1501 dataset under different noise ratios **only using label correction**, where the soft labels are generated at different early epochs.

D Ablation Study

D.1 Impact of Score Computation Epoch *T* on Label Correction

In Sec. 5.6, we investigate how early in training is our metric effective at estimating the importance scores of samples and observe that after 12 epochs the importance scores of samples estimated by our method tend to be stabilized. However, it beckons the consideration: Are the generated soft labels trained for 12 epochs sufficient to achieve a good performance in label correction? To answer this question, we explore the impact of score computation epoch T, which is the soft labels generation epoch as well, on label correction. We present results in Fig. 15. Our sensitivity analysis reveals that the soft labels generated between the $8^{\text{th}} - 20^{\text{th}}$ epochs demonstrate a superior label correction capability, even approaching the model performance trained on the clean dataset. This observation clarifies our earlier question: Even in the relatively early stages of training (after 8 epochs), the soft labels we proposed still exhibit a notable capability of label correction.

Furthermore, we observe that the label correction capability of soft labels generated after 20 epochs exhibits a decline, particularly pronounced under high noise ratios such as 30%. The underlying rationale is that, as the model undergoes prolonged training on a noisy dataset, the model starts memorizing a huge amount of mislabeled samples, leading to fitting these erroneous labels (Pleiss et al., 2020). Overall, our approach exhibits robustness to hyper-parameter T, particularly within an acceptable range of noise ratios., e.g. 10%, 20%.

E Example Images

We present some samples from three ReID datasets and CIFAR-100 dataset sorted from easy to hard based on our proposed pruning metric. From Figs. 16-19, we observe that the samples with the smallest important scores tend to be simple and are canonical representations of each class. In contrast, the samples with higher scores are harder to identify: they have different backgrounds or even suffer from occlusion or truncation.

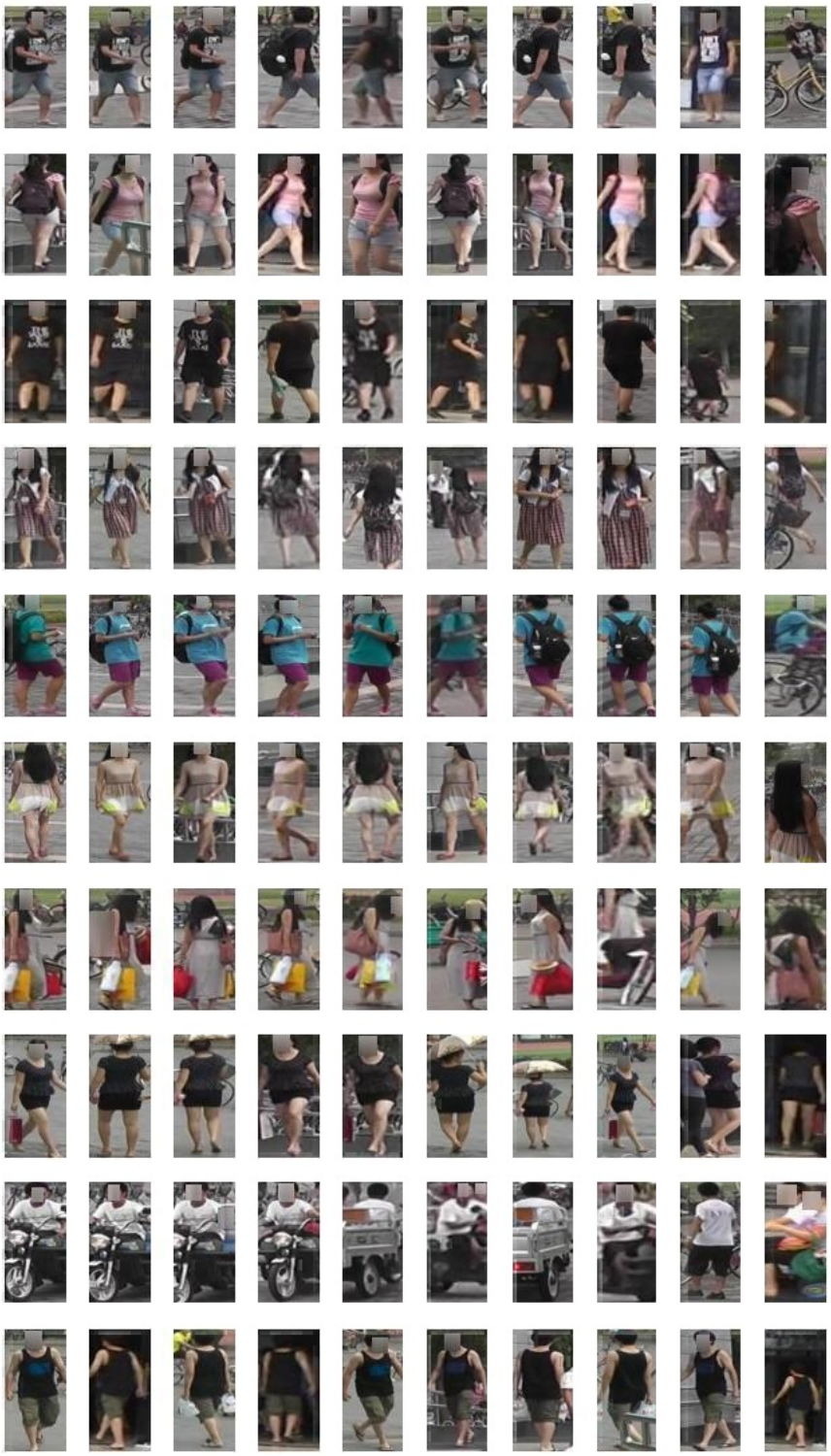


Figure 16: Samples from Market1501 sorted from *easy* (the first column) to *hard* (the last column) based on our proposed importance scores. Images in each row belong to the same identity.

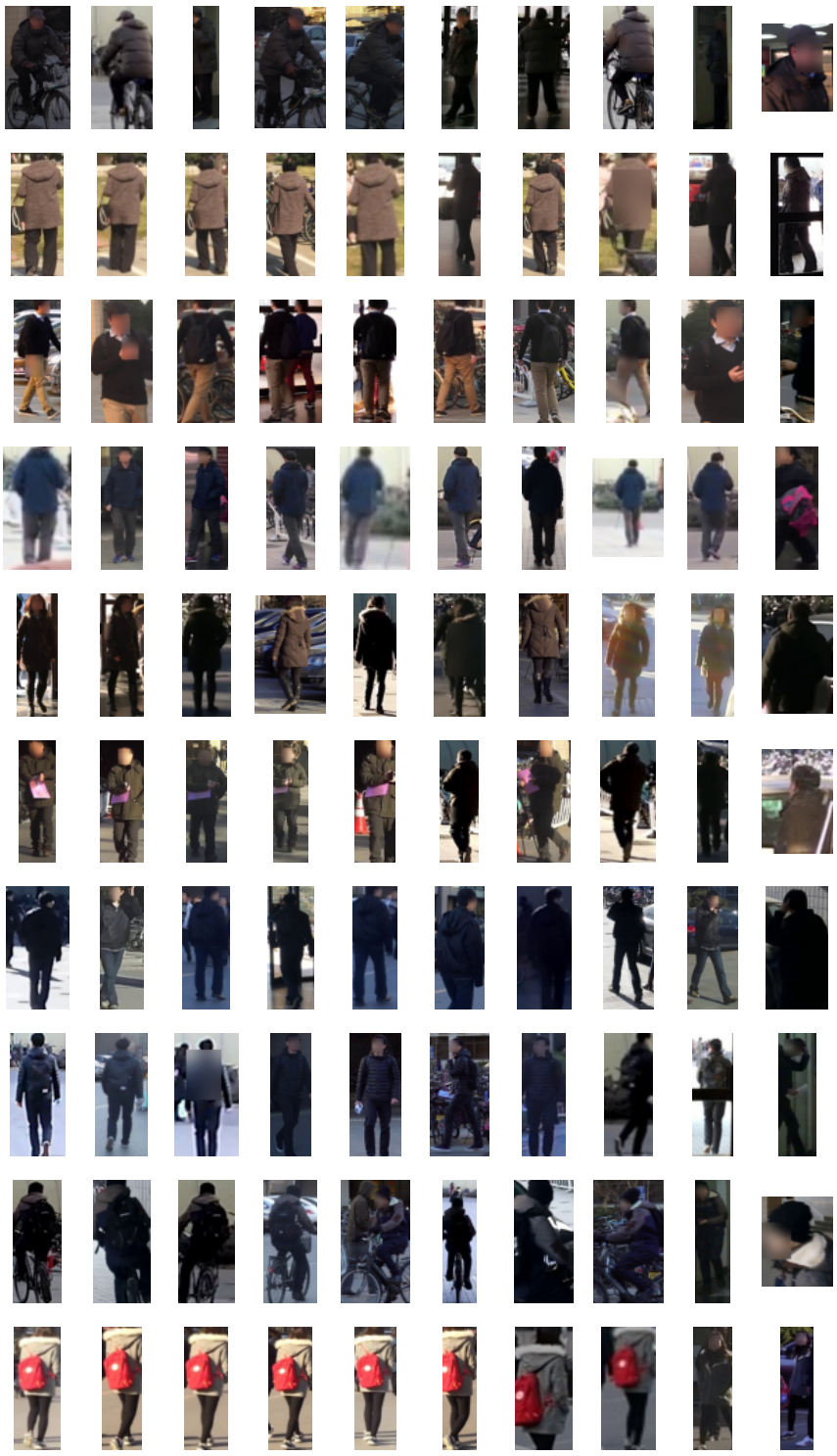


Figure 17: Samples from MSMT17 sorted from *easy* (the first column) to *hard* (the last column) based on our proposed importance scores. Images in each row belong to the same identity.

Figure 18: Samples from VeRi sorted from *easy* (the first column) to *hard* (the last column) based on our proposed importance scores. Images in each row belong to the same identity.

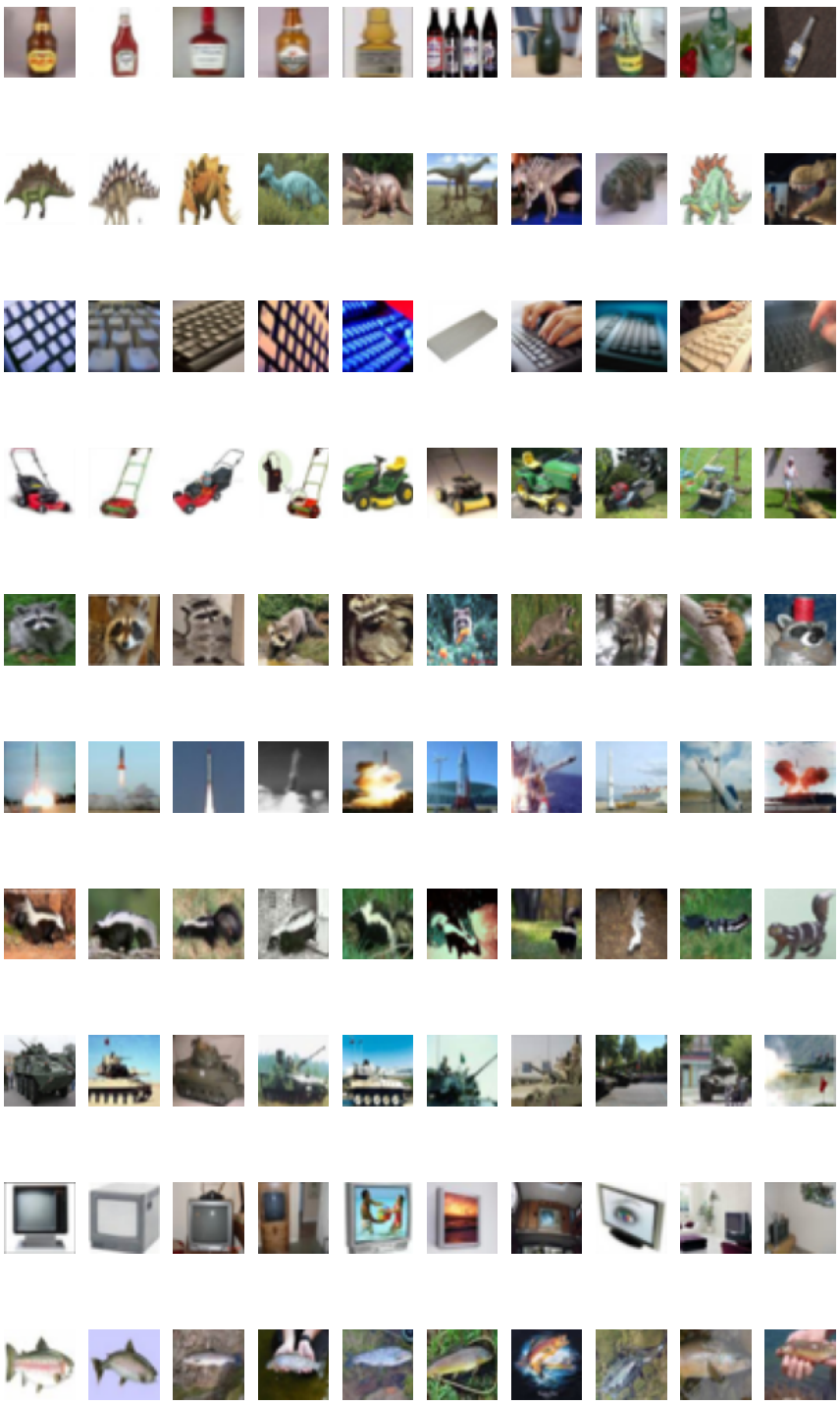


Figure 19: Samples from CIFAR-100 sorted from *easy* (the first column) to *hard* (the last column) based on our proposed importance scores. Images in each row belong to the same identity.