Learning with Label Noise for Image Retrieval by Selecting Interactions

Sarah Ibrahimi[†*] Arnaud Sors[‡] Rafael Sampaio de Rezende[‡] Stéphane Clinchant[‡]
[†] University of Amsterdam [‡] NAVER LABS Europe

Abstract

Learning with noisy labels is an active research area for image classification. However, the effect of noisy labels on image retrieval has been less studied. In this work, we propose a noise-resistant method for image retrieval named Teacher-based Selection of Interactions, T-SINT[1]*, which identifies noisy interactions,* i.e. *elements in the distance matrix, and selects correct positive and negative interactions to be considered in the retrieval loss by using a teacher-based training setup which contributes to the stability. As a result, it consistently outperforms state-of-the-art methods on high noise rates across benchmark datasets with synthetic noise and more realistic noise.*

1. Introduction

Deep Learning models need large amounts of data to train, but the collection and annotation of large scale datasets is highly time consuming and expensive. One way to avoid data labeling is by employing self-supervised or semi-supervised learning techniques, that require little to no labels [4, 8, 14]. Another way is to collect annotations from web sources by downloading user-written or automatically generated captions or tags and use them as labels when training supervised models [3, 27, 42]. These annotations often contain noise due to incorrect captions or tags. As a consequence, noisy labels will prevent models from obtaining their potential best performance. Apart from correcting label noise by manual relabeling, one can think of methods that can distinguish correctly from incorrectly labeled samples during training to avoid the need for manual correction.

Over the past few years, learning with noisy labels has been studied extensively for image classification by using a wide variety of techniques such as robust losses [6, 12], sample selection [15, 18], regularization [23, 45], and meta learning [20]. One of the key ideas of noise-resistant methods for classification is that samples with a high probability of being wrongly labeled should be either ignored or assigned a lower weight during the training process [36]. The effect of having noisy labels has been less studied for image retrieval. One recent study on noisy labels for image retrieval was done by Liu *et al*. [22]. Following classification-based methods for learning with noisy labels, it applies the principle of discarding samples with noisy labels on image retrieval and this method performs better than not using any noise-resistant component during training.

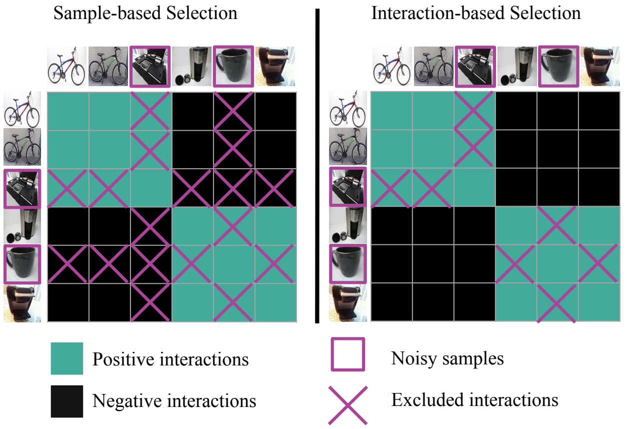

Figure 1: Imagine a set of 6 data samples, with 2 classes (bike and coffee maker) and 3 samples per class. We represent distance matrices between each possible pair in this minibatch and call each pairwise distance an interaction. In this set, one sample of a sofa is mislabeled as a bike and a mug is mislabeled as a coffee maker. Traditional sample-based selection approaches exclude all interactions of these mislabeled samples. Our interaction-based selection approach only excludes false positive interactions, since negative interactions have a high probability to remain negatives after mislabeling. This is indicated by the purple crosses only appearing on the green, positive interactions.

An important difference between classification and retrieval methods in general is that retrieval methods use *interaction losses*, *i.e*. loss functions defined on a pair or tuple of samples, compared to sample-based losses for classification [13, 24]. Interaction losses aim to bring representations of samples from the same class (positives) closer together and

[*] Work done while interning at NAVER LABS Europe.

[1] `https://europe.naverlabs.com/research/machine-learning-and-optimization/tsint`

those of different classes (negatives) further apart. Using sample-based selection mechanisms for retrieval has two main disadvantages. First, discarding samples will not only result in removing false positive interactions, but also many true negatives. For large datasets with small noise rates, sufficiently many interactions will remain to learn effectively. However, for high noise rates or small datasets, sample-based techniques might remove too many samples and will most likely break the learning process. The difference between these two approaches is visualized in Fig. 1. Second, some datasets only contain labels for positive and negative pairs and have no class labels, such as SfM-120K [32] for landmark retrieval and MSMARCO [26] for document retrieval. Using interactions provides a solution for these two weaknesses by keeping true negative interactions and removing the need for class labels.

In this work we present *T-SINT*, which stands for **t**eacher-based **s**election of **int**eractions for learning with label noise. *T-SINT* selects true positive and negative interactions to be considered in the retrieval loss by using a teacher-based training setup. This simple yet effective mechanism allows us to train robust models even under high noise rates. We show that our method is easy to tune and much more stable than existing methods across different datasets and different noise levels. We evaluate these on several datasets under uniform noise and close to realistic noise. For uniform noise, we beat the best competing method by almost 7% on average on all noise levels and more than 13% on the highest noise level. For more realistic noise, we beat the best competing method by up to 3.5%.

2. Related work

Noisy labels in image classification. Noise-resistant methods for image classification use a wide variety of techniques such as robust losses [6, 12], sample selection [7, 15, 18, 35, 43], regularization [16, 23, 45], and meta learning [20]. All methods aim to detect noisy samples. A common way to do this is by using a small clean subset of the data and learn characteristics about this set [39, 44]. Clean subsets are available in common benchmark datasets for image classification such as Clothing1M [42], but in real world scenarios they often don't exist. Another detection technique for noisy samples, is the so called 'small-loss trick' [36]. This trick separates correctly labeled samples from wrongly labeled samples by considering the loss values, since correctly labeled samples have in general smaller loss values. This also relates to the memorization effect [1], stating that Deep Neural Networks learn from clean samples first and then learn from noisy samples.

Noisy labels in image retrieval. Learning with noisy labels has not been extensively studied for image retrieval, although two recent works took a first step [5, 22]. The SuperLoss [5] is a general noise resistant method that can be used for various tasks such as regression, classification, object detection, and retrieval. It consists of a loss function that has to be applied on top of a task-specific loss function and acts similar to an activation function by reducing the importance of hard, and therefore probably wrongly labeled, samples. For retrieval, the SuperLoss uses an interaction-based selection approach by weighing interactions. While the concept of the SuperLoss is easy and widely applicable, the image retrieval experiments in [5] have shown to be highly sensitive to techniques as hard-negative mining and hyperparameter tuning. Liu *et al.* present to the best of our knowledge the first study that solely focuses on noise-resistant image retrieval [22]. It shows that existing noise-resistant methods for classification [15, 28, 43] are not effective for retrieval and are in some cases even worse on image retrieval than commonly used interaction losses without an explicit noise-resistant component. PRISM, the noise-resistant method of [22] is a class-center approach that compares the similarity of a feature with all other features of the same class with the help of a memory bank [41], to determine whether the sample is noisy. Any samples detected as noisy are discarded during training.

Where PRISM discards noisy samples and the SuperLoss weighs interactions based on noise estimations, our method combines the best of both approaches. Like the SuperLoss, our method uses an interaction-based selection approach and therefore does not discard true negative interactions as happens with sample-based selection approaches such as PRISM. Our work differs from the SuperLoss by not using a weighing scheme for interactions. By selecting or discarding interactions, like PRISM, we present a more stable method that requires less hyperparameter tuning. Also, our method does not require a memory bank.

Self-distillation. Knowledge distillation has gained much popularity across application domains such as computer vision and natural language processing [11, 21, 34, 46]. It aims to transfer knowledge or features learned from a teacher network to a student network. When the teacher and student have identical architectures, this is called self-distillation. Fang *et al.* show how distillation can help in a self-supervised setup [11]. Li *et al.* present how distillation can be used to learn from noisy labels for image classification [21]. With the help of a small set of images with clean labels and a knowledge graph, a model is guided to learn from the entire noisy dataset. Nguyen *et al.* use the model ensembling combined with a prediction ensembling method together with a filtering mechanism for noisy labels to create a noise-resistant method [25].

Our method uses a teacher-student setup with a model ensemble technique as presented in [25, 38]. However, where distillation methods aim for consistency between the teacher and the student output, our method does not need

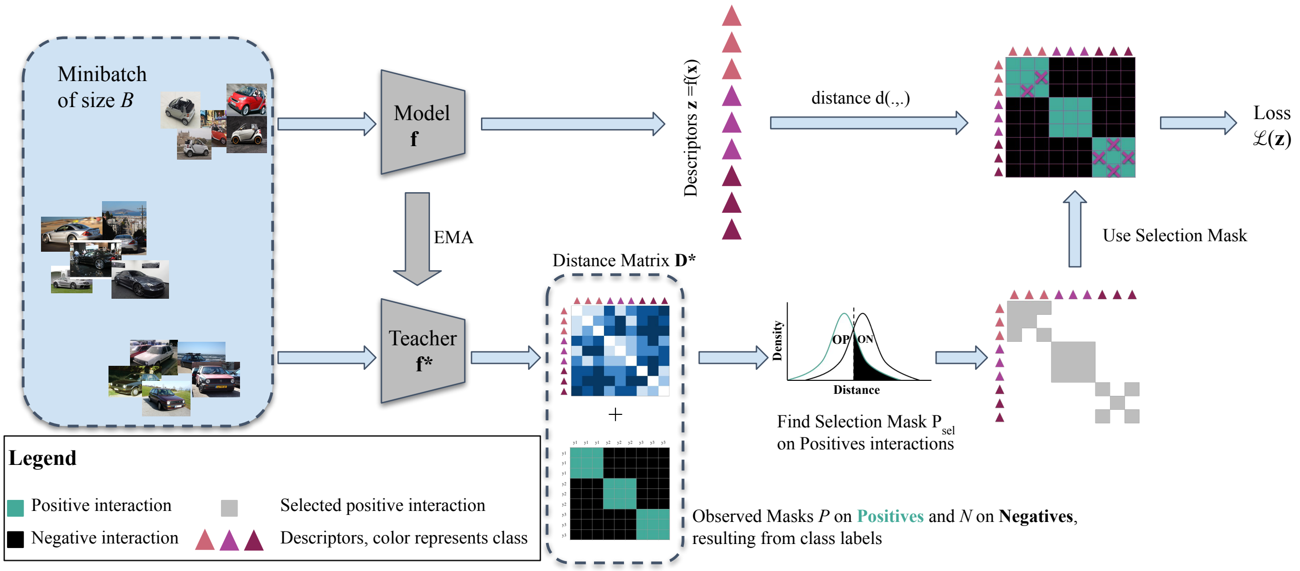
Minibatch
of size B
Model
f
EMA
Teacher
f*
Descriptors z = f(x)
distance d(.,.)
Loss
ℒ(z)
Use Selection Mask
Distance Matrix D*
+
Density
OP
ON
Distance
Find Selection Mask P_sel
on Positives interactions
Observed Masks P on Positives and N on Negatives,
resulting from class labels
Legend
Positive interaction
Selected positive interaction
Negative interaction
Descriptors, color represents class

Figure 2: A model f extracts descriptors that are used in a distance function on all pairs of descriptors to obtain a pairwise distance matrix and to build relevance annotations from class annotations. A teacher model $f^\star$ computes the interaction matrix for the same samples in the batch as the main model. We deduce a selection mask with the help of a cutting value and apply it on the distance matrix of the main model to then let it calculate the loss. Best viewed in color.

this constraint and only requires a task specific loss.

3. Method

Our method starts with a typical image retrieval pipeline. We have a model f that extracts descriptors and we use a distance function on all pairs of descriptors to obtain a pairwise distance matrix and to build relevance annotations from class annotations. This is shown in the top part of Figure 2. The relevance annotation for a batch is presented by the interaction matrix on the right, where green interactions represent predicted positive pairs and black interactions negative pairs. Since we deal with noisy samples, some of the interactions are incorrect and should be discarded for optimal performance of the model. The key component of our method produces a mask used to select interactions (bottom branch of Figure 2), which is then applied to interactions in the main branch. This is further described later in this section.

3.1. Loss

Our method is based on the contrastive margin loss [9], except it calculates it on *selected* interactions rather than using them all. Before we explain how to select, we introduce this loss. Image retrieval methods use interaction losses that bring descriptors of samples from the same class closer together and those of different classes further apart. For a minibatch $\mathcal{B} = \{(x_1, y_1), ..., (x_B, y_B)\}$ of size B, we can compute the real-vector feature representations $z_i = f_\theta(x_i)$ at the output of model f parameterized by θ. We denote $\mathbf{D}$ the $B \times B$ matrix of all possible pairwise distances between elements of $\mathcal{B}$:

$$\mathbf{D} = \{D_{i,j}, (i,j) \in [\![1, B]\!]^2 \mid D_{i,j} = d(z_i, z_j)\}, \qquad (1)$$

where d is usually the euclidean or cosine distance. For any batch indices $1 \leq i, j \leq B$, the contrastive margin loss works on all pairs of descriptors (z_i, z_j) as

$$\begin{cases} \ell_{i,j} & = 1_{y_i = y_j} \ell_p(z_i, z_j) + 1_{y_i \neq y_j} \ell_n(z_i, z_j); \\ \ell_p(z_i, z_j) & = D_{i,j}{}^q; \\ \ell_n(z_i, z_j) & = \max(0, m - D_{i,j})^q, \end{cases} \qquad (2)$$

where the form of positive terms (ℓ_p) and negative terms (ℓ_n) are different, $m > 0$ is the margin and the distance exponent q is usually 1 or 2 (in our case 1). Denoting by $\mathcal{P} = \{(i,j) \in [\![1, b]\!]^2 | y_i = y_j\}$ the set of positive pairs and $\mathcal{N} = \{(i,j) \in [\![1, b]\!]^2 | y_i \neq y_j\}$ the set of negative pairs, the loss $\mathcal{L}$ over the batch is an aggregation of positive and negative interactions:

$$\begin{cases} \mathcal{L}(\mathbf{z}) & = B^{-2}(\bar{\ell}_p(\mathbf{z}) + \bar{\ell}_n(\mathbf{z})); \\ \bar{\ell}_p(\mathbf{z}) & = \mathbb{E}_{(i,j)\sim\mathcal{U}(\mathcal{P})}[\ell_p(z_i, z_j)]; \\ \bar{\ell}_n(\mathbf{z}) & = \mathbb{E}_{(i,j)\sim\mathcal{U}(\mathcal{N})}[\ell_n(z_i, z_j)]. \end{cases} \qquad (3)$$

3.2. Interaction Selection

When a dataset has noisy labels, it will affect interactions. For a sample with a corrupted label, when the number of classes is large an observed positive interaction using this sample is almost always a false positive. For negative interactions, this has less impact: whenever the number of different instances in a batch is much smaller than the number of total classes in the dataset, chances are very small that an observed negative interaction is actually a positive interaction. A study on false negative interactions is provided in the Supplementary material.

We are working on a distance matrix $\mathbf{D}$ where *observed positives* elements $\{D_{i,j} \mid (i,j) \in \mathcal{P}\}$ are noisy and observed negative elements $\{D_{i,j} \mid (i,j) \in \mathcal{N}\}$ can be considered all clean. We would like to identify *false positive* interactions and exclude them from the aggregation in the loss function. As many noise correction methods [5, 15, 18, 43], we rely on the fact that if we have a trained retrieval model $f^\star$, the *distance value* on positive elements according to this model gives us an indication on the likelihood of the interaction being a true or a corrupted one. Interactions between clean samples are expected to have a small distance value and for interactions that are noisy the distance values will most likely be larger.

Figure 3 exemplifies this idea. (a) Consider two distributions $p_{TP}(d)$ and $p_{TN}(d)$ of true positive and true negative interactions respectively. If we have at our disposal a hypothetical *perfect* model in the sense of a retrieval metric, it will attribute smaller distance values to all true positive interactions compared to any negative interaction. (b) Therefore, when in an observed setting, we can find a distance value of this model that can be used to separate true positive from false positive interactions (amongst observed positives), as indicated by the red line. (c) In actual fact, such a perfect model is obviously not available, but we can still use a non-perfect $f^\star$, according to which $p_{TP}(d)$ and $p_{TN}(d)$ will likely overlap for difficult interactions, but which can already help identify easier noisy interactions. (d) By using a cutting value, a subset of the positive interactions will be removed and therefore not considered by f.

3.3. Using a Teacher

Now comes the question of what to use for $f^\star$. In practice a perfect model $f^\star$ is not available, otherwise the retrieval problem would already be solved. To create a model which approaches it, we take inspiration from knowledge distillation, where a strong teacher network provides knowledge to a student network that learns the knowledge [17]. More specifically, we take inspiration from Mean Teacher [38]. It consists of a teacher-student setup with two similar model architectures, where the student model weights are updated through gradient descent and the teacher model weights as an exponential moving average of the student weights. Mean Teacher uses two loss functions, a classification loss on the student model and a consistency loss that compares the softmax outputs of both networks, usually the MSE loss or KL-divergence loss. Strictly speaking, Mean Teacher is not a self-distillation method, since it optimizes for consistency during training, while for self-distillation methods the consistency regularization is usually performed after training.

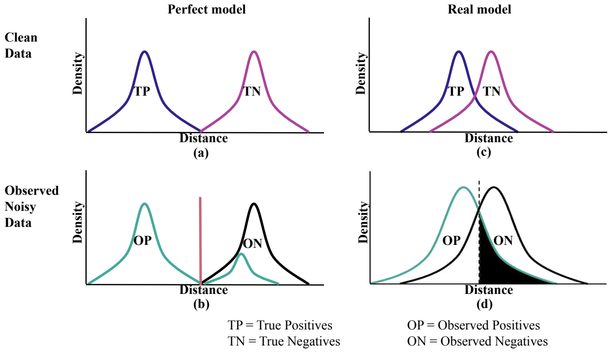

Figure 3: A perfect retrieval model, represented on the left, is also able to identify noise. A real model can approach this idea. Section 3.4 details this.

We decide to use a teacher only for the task-specific loss and do not aim to optimize for consistency between the teacher and student models outputs. Therefore, we do not consider this as a typical knowledge distillation approach. Our main model and teacher model have the same architecture and same initialization. Our main model is updated by backpropagation and we choose the update for the teacher model to be an exponential moving average of the parameters of the main model every iteration.

$$\theta^\star_t = \alpha\theta^\star_{t-1} + (1-\alpha)\theta_t, \tag{4}$$

where $\theta, \theta^\star$ are the main model and teacher's weights, respectively. Our network updating step is similar to [38].

3.4. Selection algorithm

As described in Section 3.1 we want to select and keep only a subset of interactions amongst *observed positives*, according to their distance values in the output space of a teacher model $f^\star$. To do so, we could simply select out the top-K interactions in terms of distance value, with K chosen such as to reach a predefined selection ratio. However, if we have a dataset with noise rate r, because minibatches have a limited size, each of them will not contain exactly rB noisy samples but a number which varies around rB from minibatch to minibatch. In order to allow for a selection ratio which also varies from minibatch to minibatch, if we want to keep on average a proportion τ of positive interactions, we keep a multiplicative running-average d_{cut} (with

Algorithm 1: Teacher-based selection of interactions: calculation of the minibatch loss

```
input : $\mathbf{z} = \{z_i = f(x_i)\ i \in [\![1, B]\!]\}$,
        $\mathbf{z}^\star = \{z_i = f^\star(x_i)\ i \in [\![1, B]\!]\}$,
        $\mathbf{y} = \{y_i, i \in [\![1, B]\!]\}$: a minibatch of main model features, teacher model features, and class labels;
        $\tau$: selection hyperparameter;
        $\beta$: momentum on cutting threshold upd.;
        $d_{cut}$: current value of cutting threshold;
        $\mathbf{P},\ \mathbf{N} \in \{0, 1\}^{B \times B}$ : observed masks;
output: loss value $\mathcal{L}(\mathbf{z})$
initialization $d_{cut} \leftarrow None$
Build distance matrices between all pairs of features, for both the main model and teacher model):
    $\mathbf{D} = \{D_{i,j} \mid i, j \in [\![1, B]\!]^2,\ D_{i,j} = d(z_i, z_j)\}$
    $\mathbf{D}^\star = \{D^\star_{i,j} \mid i, j \in [\![1, B]\!]^2,\ D^\star_{i,j} = d(z^\star_i, z^\star_j)\}$
Gather observed positive distance values according to teacher model: $\mathcal{D}^\star_P = \{D^\star_{i,j} \mid P_{i,j} = 1\}$
Denote $F$ the cumulative distribution of $\mathcal{D}^\star_P$, calculate the $\tau$-percentile of $F$: $d_B = F^{-1}(\tau)$
Update cutting value with moving average:
    if $d_{cut}$ is None then $d_{cut} \leftarrow d_B$; else
    $d_{cut} \leftarrow \beta d_{cut} + (1 - \beta) d_B$
Deduce selection mask on positives:
    $\mathbf{P}_{sel} = \mathbb{1}(\mathbf{D}^\star < d_{cut})\ \&\ \mathbf{P}$
Calculate loss using the main model's distance matrix: $\bar{\ell}_p(\mathbf{z}) = \mathbb{E}_{\mathbf{P}_{sel}=1}[D_{i,j}]$,
    $\bar{\ell}_n(\mathbf{z}) = \mathbb{E}_{\mathbf{N}=1}[\max(0, m - D_{i,j})]$
    $\mathcal{L}(\mathbf{z}) = B^{-2}(\bar{\ell}_p(\mathbf{z}) + \bar{\ell}_n(\mathbf{z}))$
```

momentum β) of the τ-percentile of observed positive distance values, and do the selection on each minibatch using this cutting value. Algorithm 1 details this selection step.

Now we explain how to choose the threshold τ. In general, noise-resistant methods that use the loss value as a selection mechanism to find noisy samples, use one or more hyperparameters to set the desired selection level. This is necessary to decide which samples to ignore or how to weigh them within a batch. When applying the method to a practical problem, it is useful to have guidelines about how to choose these hyperparameters. For example, [22] uses the noise rate that corresponds to the noise level they introduced in their synthetic uniform noise. In our approach, we make the following estimation. Assume we have a noise rate r on examples in our dataset and k instances per class in each mini-batch. Then we can estimate the proportion of true positives amongst all observed off-diagonal positives as $\tilde{p} \sim (1 - r)^2$, or equivalently including the diagonal,

$$\tilde{p} \sim \frac{(1 - r)^2(k^2 - k) + k}{k^2}, \tag{5}$$

where k^2 is the total number of interactions for an instance, $(1 - r)^2$ is the estimated proportion of clean interactions and the $(k^2 - k)$ multiplier and the $1/k$ addition account for the fact that noise affects only non-diagonal elements, although the selection is performed on all. This expression is valid if the per-batch noise level has low variance around r, i.e. when $B >> 1$, and when the number of classes is sufficiently high so that the case where two samples forming a positive interaction get corrupted to the same class is rare. $\tau = \tilde{p}$ constitutes a reasonable starting value for the selection hyperparameter if an estimate of the noise rate on class annotations is available. However, this is only an initial guess, and the best value may vary a little. Indeed, the above calculation pre-supposes that the optimal number of interactions to filter is equal to the true number of corrupted interactions. This is true in the case of a perfect teacher, but in the case of a real teacher this may not exactly be. Also, the noise rate in various experimental datasets may be slightly different than the expected one.

3.5. Comparison with state of the art

We compare T-SINT with two existing noise-resistant approaches for image retrieval: SuperLoss and PRISM.

SuperLoss [5]. This is a loss that can be applied on top of any other loss. Assume we have an input loss value l_i, then the SuperLoss SL can be applied in the following way

$$SL_\lambda(l_i) = \min_{\sigma_i} \left((l_i - \tau)\sigma_i + \lambda(\log \sigma_i)^2\right),$$

where τ is a threshold that separates easy from hard samples based on their loss values, λ is a regularization hyperparameter and σ_i is a confidence parameter which in practice will be replaced by the converged value at the limit [2]. Therefore only τ and λ should be tuned, where τ is usually a running average of the input loss during training.

Whenever the SuperLoss is applied on an interaction loss, such as the contrastive loss, this can be described as follows:

$$SL^{CL}_\lambda(f(x_i), f(x_j), y_{ij}) = \begin{cases} SL_\lambda(l^{CL}_+(f(x_i), f(x_j))) \text{ if } y_{ij} = 1, \\ SL_\lambda(l^{CL}_-(f(x_i), f(x_j))) \text{ if } y_{ij} = 0. \end{cases}$$

where the two losses use two independent thresholds τ_+ and τ_- and share the same λ value.

PRISM [22]. This method starts with estimating which samples are clean. Therefore it compares each sample representation z_i with all stored features v_j of the same class k in the memory bank M. When the estimated probability, calculated by the Softmax of a sample being clean, $P_{\text{clean}}(i)$,

[2] An analytical expression is given in [5].

is below a threshold m, it will be considered as a noisy sample and discarded. m is calculated through a smooth top-R method that uses an estimated noise rate R. Specifically,

$$m = \frac{1}{\tau} \sum_{j=t-\tau}^{t} Q_j,$$

where Q_j is the R^{th} percentile $P_{\text{clean}}(i)$ value in the j-th minibatch and τ stands for the number of last batches.

After selecting the clean data samples according to the threshold m, these are used to calculate the selected loss, for example the contrastive loss. Furthermore, a memory bank loss is calculated, to enable more positive and negative pairs in the loss. The main hyperparameter to tune or estimate is R, which is compared to the SuperLoss more intuitive.

4. Experiments

4.1. Datasets

We report results on four commonly-used datasets for image retrieval *Caltech-UCSD Birds-200-2011 (CUB)* [40], *CARS* [19], *Stanford Online Products (SOP)* [37], *Revisited Paris (RParis6k)* [31] and a recent dataset created for learning with label noise for image retrieval, *CARS-98N* [22].

CUB consists of 11,788 images in 200 classes, where the first 100 classes are used for training and the rest for evalation. CARS has 16,158 images with 196 classes, where the first 98 are used for training and the remaining classes for evaluation. SOP contains 120,053 product images from 22,634 classes. The first 11,318 classes (59,551 images) are for training and the other 11,316 (60,502 images) classes are used for evaluation.

RParis6k consists of 6,412 images of 12 landmarks in Paris. This dataset is a cleaned version of the Paris dataset [30] provided by Radenović *et al.* [31]. We use this dataset for evaluation when training on two noisy datasets with landmarks: Oxford [29] and Landmarks [2]. The Oxford dataset consists of 5,062 images of 11 Oxford landmarks. We use the original version of this dataset that has not been cleaned. The Landmarks dataset, also known as Landmarks-full, consists of 167.231 images in its train set. These images are retrieved by querying a search engine and therefore considered as noisy.

CARS-98N is a recent training set with noisy labels, created by crawling 9,558 images for 98 car models from Pinterest. These models correspond to the 98 labels from the CARS training set. Since the test set of the CARS dataset is considered as clean, it is used for evaluation.

4.2. Noise Types and Rates

CUB, CARS, and SOP are considered as clean datasets and have only been used in a noise study by [22] by adding synthetic noise to the train sets of these datasets. However, we do not rule out that any noise in these original datasets is present. We follow [22] by applying 10%, 20%, and 50% uniform noise to these datasets. Following noise studies for classification, we add 70% of uniform noise to our study.

More realistic noise is present in Oxford, Landmarks and CARS-98N, since their images are collected with the help of search engines and not manually cleaned. Radenović *et al.* showed that the Oxford dataset has noise in its annotation that is not related to the building category but to the difficulty of recognizing the building which is indicated by 'positive', 'junk', or 'negative' [31]. There is no exact estimation of the noise rate. Gordo *et al.* [13] showed that Landmarks contains a non-negligible amount of unrelated images. A clean version of this dataset was presented by [13] and left only 25% of the training images and 87% of the classes. We use this as a guideline to estimate the noise level in this dataset. For evaluation purposes we use the cleaned test set of the Paris dataset [30], RParis6k, that has been cleaned manually by [31]. The level of noise in CARS-98N has been estimated by the creators of this dataset at 50%.

4.3. Baselines & Implementation Details

We compare our method T-SINT against three methods:

- PRISM [22], which is the current State-of-the-Art approach for learning with label noise for image retrieval on the benchmarks we consider.
- SuperLoss [5], which has been effective for label noise on image retrieval, in particular on landmarks.
- Contrastive Margin Loss [9], which is a simple baseline without a noise-resistant component. We use this to compare to PRISM, the SuperLoss and T-SINT since the three methods all use the contrastive margin loss.

T-SINT uses a CLIP [33] model with a ViT-B/32 backbone [10] and a similar teacher, both without a head, resulting in a 512 dim output feature. We choose CLIP since it covers a wide range of domains and are therefore more suitable than domain specific image retrieval datasets. We rerun all baseline methods by using the same CLIP model as a backbone. Results on the original backbones as described in [22] can be found in the Supplementary material.

For all methods, we tune the learning rate and the batch size. Details about the best values of these hyperparameters for each method and dataset are provided in the Supplementary material. The noise-resistant methods need some additional tuning. As explained in Section 3.5, the SuperLoss needs tuning for the regularization parameter λ and two thresholds τ_+ and τ_- for the positive and negative interactions. For each dataset and each noise level, we did a hyperparameter search for λ and τ_+ and τ_-. For λ we tried the values 0.001, 0.01, 0.05, 0.1, 0.25, 1.0 as these

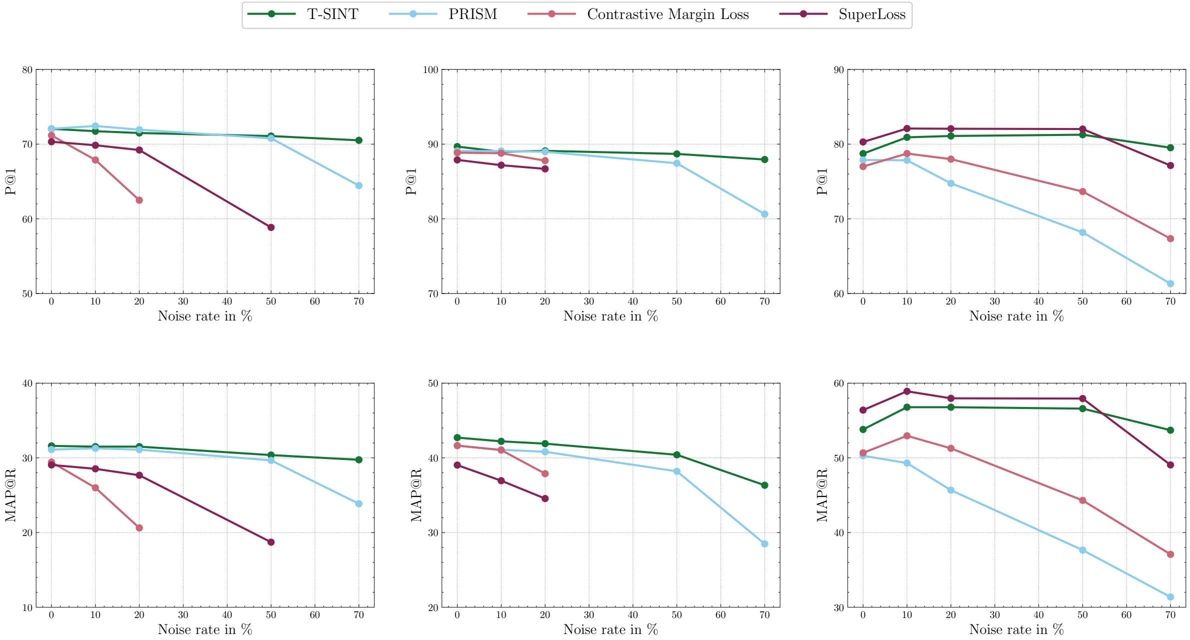

Figure 4: **Precision@1** ((a),(c),(e)) and **MAP@R** ((b),(d),(f)) results on CUB, CARS and SOP respectively. All methods depicted use the CLIP model as a teacher. Whenever a method has no measure point for a specific noise rate, it means that the model was not able to perform better than its initial state when using the CLIP model. This is the case for the Contrastive Margin Loss and the SuperLoss on CUB and CARS. Note that the y-axis has a different range for each dataset and metric. Best viewed in color.

values were recommended in [5]. For the thresholds, [5] recommends three options: a global average, an exponential running average with a fixed smoothing parameter or a fixed value given by prior knowledge on the task. Therefore, we experimented with the global average and the exponential running average. The best values can be found in the Supplementary material. For PRISM, we use the estimated noise rate as provided [22]. For Oxford and Landmarks we tune this value. For T-SINT , we estimate the proportion of true positives by equation 5.

Following [22, 24], we use Precision@1 (P@1), Mean Average Precision@R (MAP@R), and mean Average Precision (mAP) for evaluation, since a ranked list of nearest neighbours on the test set is used for evaluation.

4.4. Comparison with State-of-the-Art

Uniform Noise. We analyze the effect of several levels of uniform noise (0%, 10%, 20%, 50%, and 70%) on CUB, CARS, and SOP. The results are presented in Figure 4. Note that the scores for 0% noise are reference scores that we aim to match in the case of label noise.

In general, the CLIP model performs quite well on all three datasets. However, the CLIP model is not noise-resistant. Combined with the contrastive margin loss, it performs well on 0-10% noise, but starts to drop severely for larger noise rates. For CUB and CARS, this method is not even able to achieve a score higher than when using a CLIP model as a feature extractor, which is indicated in Figure 4 by missing points. Therefore we can conclude that a strong backbone is not sufficient by itself to be resistant to noise.

Considering PRISM, we observe that for CUB and CARS, T-SINT performs on par with this method on noise levels of 10%, 20% and 50%. However, for 70% noise, PRISM shows a large drop in performance compared to 50% noise, whereas our method is much more stable. The gap between PRISM and our method is 5 to 6 points for both the P@1 and the MAP@R scores. For SOP we notice a larger gap over all noise levels, where in the case of 70% noise our method outperforms PRISM by more than 20% for P@1 and MAP@R scores. On average on CUB, CARS, and SOP at 70% noise, our method scores more than 10% higher on P@1 and more than 13% higher on MAP@R.

The SuperLoss performs significantly worse than our method on CUB and CARS for 10% and 20% noise. For CUB the gap even increases to a difference of 10% in P@1 and MAP@R at 50% noise. However, for 70% uniform noise on CUB and for 50% and 70% uniform noise on CARS using the SuperLoss breaks the training at each combination of hyperparameters that has been tested. We have

Method	Training set	Easy	Medium	Hard
CTRM$_{\text{BN-inception}}$[9]	Oxford	64.66	50.37	25.26
	Landmarks	77.33	62.67	37.94
PRISM$_{\text{BN-inception}}$[22]	Oxford	62.72	49.28	24.38
	Landmarks	76.33	62.03	37.05
CTRM$_{\text{ViT-B/32}}$[9]	Oxford	74.90	62.63	38.40
	Landmarks	85.62	73.53	51.27
PRISM$_{\text{ViT-B/32}}$[22]	Oxford	77.92	65.50	41.74
	Landmarks	85.52	72.99	49.52
SuperLoss$_{\text{ViT-B/32}}$[5]	Oxford	**82.59**	**70.35**	**47.68**
	Landmarks	86.72	76.27	56.11
T-SINT $_{\text{ViT-B/32}}$ (Ours)	Oxford	82.21	70.18	47.36
	Landmarks	**87.55**	**77.66**	**57.17**

Table 1: mAP scores for training on Oxford and Landmarks, testing on RParis-6k.

Method	P@1	MAP@R
CTRM$_{\text{BN-inception}}$	44.91	4.76
PRISM$_{\text{BN-inception}}$[22]	57.95	8.04
CTRM$_{\text{ViT-B/32}}$	82.59	30.38
PRISM$_{\text{ViT-B/32}}$[22]	81.75	29.93
SuperLoss$_{\text{ViT-B/32}}$[5]	82.55	31.31
T-SINT$_{\text{ViT-B/32}}$ (Ours)	**86.10**	**34.93**

Table 2: Precision@1 (%) and MAP@R (%) for CARS-98N.

EMA teacher	EMA d_{cut}	Teacher model	Model backbone	P@1	MAP@R
✓	✓	ViT-B/32	ViT-B/32	86.10	34.39
✗	✓	ViT-B/32	ViT-B/32	84.80	33.22
✓	✗	ViT-B/32	ViT-B/32	84.87	32.54
✗	✓	ViT-B/32	BN-Inc	38.93	3.98
✓	✓	BN-Inc	BN-Inc	38.10	3.60

Table 3: Ablation study on T-SINT , evaluated on CARS-98N. Precision@1 (%) and MAP@R (%).

no clear indication why the set of hyperparameters that we use and has been suggested by [5] for the SuperLoss does not work on these noise rates for CUB and CARS. However, Castells *et al*. [5] indicate that some learning setups for image retrieval for this method lead to poor performance and suggest to therefore tune more elements in the training setup including explicit hard negative mining, GeM pooling and descriptor whitening. We decided not to add these components and additional tuning for the SuperLoss to be able to make a fair comparison to our method and PRISM. This additional tuning is not necessarily required for all datasets. For example on SOP, we see that training with the SuperLoss does not break the training and even slightly outperforms our method, except for a noise rate of 70 %. Nevertheless, our method T-SINT is the only method that does not show a significant drop in performance from 0% to 70% noise for SOP and is therefore the most stable.

More realistic noise. We study the effectiveness of our method on more realistic noise by experiments on Oxford, Landmarks, and CARS-98N. For experiments on Oxford and Landmarks we evaluate on RParis6k by differentiating between easy, medium and hard samples following [31]. In Table 1, we see that when training on Oxford, the SuperLoss performs slightly better than our method. However, when training on a much larger dataset, our method outperforms the SuperLoss. In both cases, we see a large gap between PRISM and our method.

Table 2 presents the results on CARS-98N. Our method outperforms all other methods by a large margin. Compared to PRISM, we report a gap of almost 5% for both the P@1 and MAP@R and for the SuperLoss this gap is 3.5% for both metrics.

4.5. Ablation Study

We present an ablation study on our method to see which components contribute to its performance. The results are presented in Table 3. We notice that updating a teacher by taking the exponential moving average of the parameters in the main model compared to freezing the teacher gains slightly more than 1% of performance. Updating the cutting value with a moving average compared to keeping the cutting value fixed adds almost 2% to the score. We also study the effect of backbones. In case our main model and teacher do not have the same backbone architecture, there is no exponential moving average for the teacher's weights. Whenever we keep the ViT-B/32 backbone of the teacher and take a BN-inception pretrained on ImageNet for the main model, our model completely breaks and performs even worse than only using the contrastive margin loss with a BN-inception backbone. The reason for this is unclear so far as T-SINT is in principle compatible with the use of different architectures for the teacher and main model. Whenever we replace both ViT-B/32 backbones of the main model and the teacher by a BN-inception backbone, our method also breaks. This emphasizes that our method requires a good teacher.

5. Conclusion

We propose T-SINT , an effective noise-resistant method for image retrieval. With the help of a teacher, it identifies noisy interactions and selects correct positive and negative interactions in the distance matrix to be used by the retrieval loss. Our simple selection mechanism achieves state-of-the-art results on both synthetic noise and more realistic noise, consistently outperforming existing methods on high noise rates and being the most stable across all noise rates.

References

[1] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *ICML*, 2017.

[2] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014.

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[5] Thibault Castells, Philippe Weinzaepfel, and Jérôme Revaud. Superloss: A generic loss for robust curriculum learning. In *NeurIPS*, 2020.

[6] Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama. On symmetric losses for learning from corrupted labels. In *ICML*, 2019.

[7] Pengfei Chen, Benben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, 2019.

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

[9] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[10] 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.

[11] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. SEED: self-supervised distillation for visual representation. In *ICLR*, 2021.

[12] Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017.

[13] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 2017.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *NeurIPS*, 2020.

[15] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

[16] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

[18] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018.

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013.

[20] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Learning to learn from noisy labeled data. In *CVPR*, 2019.

[21] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *ICCV*, 2017.

[22] Chang Liu, Han Yu, Boyang Li, Zhiqi Shen, Zhanning Gao, Peiran Ren, Xuansong Xie, Lizhen Cui, and Chunyan Miao. Noise-resistant deep metric learning with ranking-based instance selection. In *CVPR*, 2021.

[23] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020.

[24] Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, 2020.

[25] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi-Phuong-Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. SELF: learning to filter noisy labels with self-ensembling. In *ICLR*, 2020.

[26] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NeurIPS*, 2016.

[27] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, 2017.

[28] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017.

[29] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[30] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[31] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018.

[32] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. 41(7):1655–1668, 2018.

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[34] Karsten Roth, Timo Milbich, Björn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. S2SD: simultaneous similarity-based self-distillation for deep metric learning. *CoRR*, abs/2009.08348, 2020.

[35] Haozhi Zhang Chenfan Zhuang Dengke Dong Matthew R. Scott Sheng Guo, Weilin Huang and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *ECCV*, 2018.

[36] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*, 2020.

[37] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

[38] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

[39] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J. Belongie. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, 2017.

[40] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[41] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *CVPR*, 2020.

[42] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

[43] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, 2019.

[44] Bodi Yuan, Jianyu Chen, Weidong Zhang, Hung-Shuo Tai, and Sara McMains. Iterative cross learning on noisy labels. In *WACV*, 2018.

[45] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[46] Zhilu Zhang and Mert R. Sabuncu. Self-distillation as instance-specific label smoothing. In *NeurIPS*, 2020.