A Unified Batch Selection Policy for Active Metric Learning

Priyadarshini K[1], Siddhartha Chaudhuri[1,2], Vivek Borkar[1], and Subhasis Chaudhuri[1]

[1] IIT Bombay
[2] Adobe Research
priyadarshini.k@iitb.ac.in,sidch@adobe.com,{borkar,sc}@ee.iitb.ac.in

Abstract. Active metric learning is the problem of incrementally selecting high-utility batches of training data (typically, ordered triplets) to annotate, in order to progressively improve a learned model of a metric over some input domain as rapidly as possible. Standard approaches, which independently assess the informativeness of each triplet in a batch, are susceptible to highly *correlated* batches with many redundant triplets and hence low overall utility. While a recent work [20] proposes *batch-decorrelation* strategies for metric learning, they rely on ad hoc heuristics to estimate the correlation between two triplets at a time. We present a novel batch active metric learning method that leverages the Maximum Entropy Principle to learn the least biased estimate of triplet distribution for a given set of prior constraints. To avoid redundancy between triplets, our method collectively selects batches with maximum *joint entropy*, which simultaneously captures both informativeness *and* diversity. We take advantage of the submodularity of the joint entropy function to construct a tractable solution using an efficient greedy algorithm based on Gram-Schmidt orthogonalization that is provably $\left(1-\frac{1}{e}\right)$-optimal. Our approach is the first batch active metric learning method to define a unified score that balances informativeness and diversity for an entire batch of triplets. Experiments with several real-world datasets demonstrate that our algorithm is robust, generalizes well to different applications and input modalities, and consistently outperforms the state-of-the-art.

Keywords: Batch active learning · Perceptual metric · Submodular optimization · Maximum Entropy Principle.

1 Introduction

Understanding similarity between two objects is fundamental to many vision and machine learning tasks, e.g. object retrieval [33], clustering [35] and classification [30]. Most existing methods model a *discrete* measure of similarity based on class labels: all inter-class samples are considered equally dissimilar, even though their features differ by different degrees. But human estimation of perceptual (dis)similarity is often more fine-grained. We may choose, for example, *continuous* measures such as the *degree* of perceived similarity in taste or visual appearance for comparing two food dishes, rather than discrete categorical labels (Figure 1). Thus, it is important to build a *continuous* perceptual space to model human-perceived similarity between objects. Recent studies

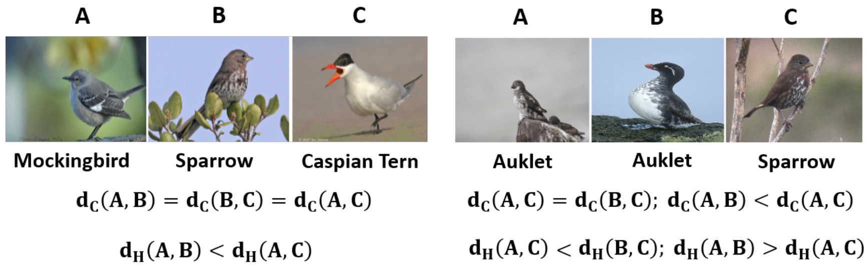

A
B
C
Mockingbird
Sparrow
Caspian Tern
$\mathbf{d_C(A,B) = d_C(B,C) = d_C(A,C)}$
$\mathbf{d_H(A,B) < d_H(A,C)}$
A
B
C
Auklet
Auklet
Sparrow
$\mathbf{d_C(A,C) = d_C(B,C);\ d_C(A,B) < d_C(A,C)}$
$\mathbf{d_H(A,C) < d_H(B,C);\ d_H(A,B) > d_H(A,C)}$

Fig. 1: Difference between class-based and perceptual distances on two different types of triplets. In each case, the class-based metric d_C fails to capture intra-class variations and inter-class similarities and is not compatible with the perceptual metric d_H.

demonstrate the importance of perceptual metrics in several tasks in computer vision and cognitive science [36,15,19].

Early work on perceptual metric learning focuses on non-parametric methods (e.g. Multidimensional scaling (MDS) [18]) which use numerical measurements of pairwise similarity for training. These are hard to gather and suffer from inconsistency. Instead, similarity *comparisons* of the form "Is object x_i more similar to object x_j than object x_k?" are easier to gather and more stable [14]. They form a useful foundation for several tasks, including perceptual metric learning. However, the number of possible triplets of n objects is $O(n^3)$, making it infeasible to label even a significant fraction of them. Fortunately, many triplets are redundant and we can effectively model the metric using only a few high-utility triplets (Figure 2). Thus it is imperative to identify and annotate a subset of high-quality triplets that are jointly informative for the model, *without knowing the annotations of any triplets in advance.* We stress this last point since it renders common triplet sampling strategies such as (semi-)hard negative mining, which rely on access to a fully annotated dataset, inadmissible.

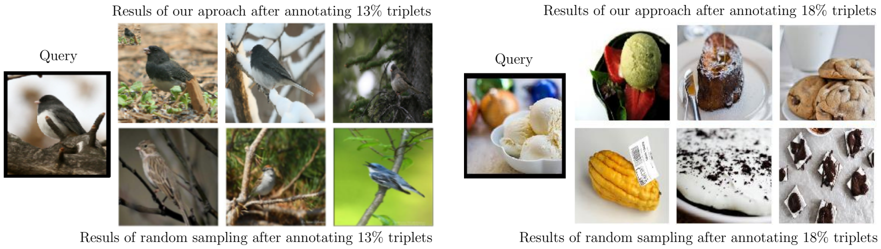

Resuls of our aproach after annotating 13% triplets
Query
Resuls of random sampling after annotating 13% triplets
Results of our approach after annotating 18% triplets
Query
Results of random sampling after annotating 18% triplets

Fig. 2: Top-3 retrieved images ranked from most to least similar by a perceptual metric (visual appearance for birds, and taste for food) trained on randomly selected (but correctly annotated) triplets vs. high-quality triplets identified for annotation by our method. For a fair comparison, both methods run for equal training rounds and solicit annotations for equal amounts of training data – 13% of the CUB-200 bird dataset on the left, and 18% of the Yummly-Food dataset on the right.

Active learning is a standard technique that addresses this issue by iteratively identifying small batches of informative samples and soliciting labels for them. While extensively studied for class label-based learning tasks, there exists very little literature [31,9,20] on active learning which focuses on perceptual/general metric learning. Further, these works merely assess the informativeness of *individual* triplets with uncertainty measures, which assume a triplet with high prediction uncertainty is more crucial to label. Although effective in many scenarios, such an uncertainty measure makes a myopic decision based solely on the current model's prediction and fails to capture the triplets' collective distribution as a whole. Independently assessed triplets may themselves have much redundancy even if they are individually informative. Hence the triplets should be not merely informative but also *diverse* or *decorrelated*.

Kumari *et al.* [20] proposed a method for selecting informative *and* decorrelated batches of triplets for active metric learning. However, their approach suffers from three major limitations: (1) The active learning strategy is based on a two stage optimization for informativeness (choice of an overcomplete batchpool of individually informative triplets) and diversity (subsequent trimming of the batchpool), applied sequentially. It does not always ensure an optimal tradeoff between the two criteria. (2) The proposed diversity measures are all ad-hoc with no principled connection to informativeness. Being heuristic, no single measure works consistently well in all cases, making it harder for a user to select which measure to use in practice. (3) The informativeness of a triplet is determined using a point estimate of the perceptual metric. Bias in the latter, e.g., because of suboptimal batch selection in prior iterations, directly translates to bias in informativeness, which can misguide the strategy.

To mitigate these issues, we propose a new batch active learning algorithm developed specifically for triplet-based metric learning. Our **key insight** is to express a set of (unannotated) triplets as a vector of random variables, and select batches of triplets that maximize the *joint entropy* measure. Thus, instead of separately expressing and optimizing informativeness for individual triplets and diversity for pairs of triplets, we develop a single probabilistic informativeness measure *for a batch of triplets*. We also provide computationally efficient approximate solutions with provable guarantees. Specifically, our main technical contributions are:

1. We propose to use the joint entropy of the distribution of triplet margins to rank a batch of unannotated triplets. We estimate the second-order statistics (mean and covariance) of *triplet margins* by randomly perturbing the current model trained on prior batches as in [7], to characterize the distribution.
2. Using the Maximum Entropy Principle, we arrive at a Gaussian distribution compatible with the given empirical mean and covariance, whose entropy is characterized by the determinant of the covariance matrix. As exact maximization of the joint entropy is prohibitively expensive (there are $\binom{m}{b}$ possible batches of size b from m triplets), we use the fact that entropy is monotone increasing and submodular to justify a greedy policy which is provably $(1-\frac{1}{e})$-optimal [22].
3. We achieve further computational efficiency by using the fact that the covariance matrix is a Gram matrix, and its determinant can be computed using efficient recursion. Our method recursively maximizes successive projection errors of a set of vectors, picked one at a time, when projected onto the span of previous choices.

This amounts to successive maximization of the conditional entropy, and is easily implemented using Gram-Schmidt orthogonalization.

We demonstrate the effectiveness of our approach through extensive experiments on different applications and data in different modalities (image, taste and haptic). In addition to having a sound theoretical justification, our method provides a significant performance gain over the current state-of-the-art.

2 Related Work

The prior work can be roughly divided into three categories. We review representative techniques in each and discuss how our work differs from the existing methods.

2.1 Perceptual Metric Learning

While there is extensive recent research on distance metric learning, most of the algorithms are specific to class-based learning tasks such as classification [30] and clustering [35], which consider two objects similar if they belong to the same class. See Bellet *et al.* [3] for a comprehensive review. In contrast, our goal is to define a perceptual distance that captures the degree of similarity between any two objects irrespective of their classes. Recently, a whole new literature has emerged that emphasizes the importance of learning such continuous measures of similarity for various applications, e.g. for measuring image similarity [36], face recognition [5], concept learning [33,32] and perceptual embedding of objects [19,11,1]. The closest application to ours is perceptual embedding of objects, where the embedding function is learned so as to model the human-perceived inter-object similarity. While multidimensional scaling (MDS) techniques have been extensively applied for this [11,1,18], they are non-parametric and require numerical similarity measurement as inputs, which are hard to gather [14]. Recent works [36,21] address these limitations by developing parametric models using non-numeric relative comparisons. A relevant method is the triplet-based deep metric learning method of Kumari *et al.* [19]. Although our method borrows base metric learning architectures from [19],[19] doesn't aim to make the metric learning algorithm data-efficient by developing an active data sampling technique.

2.2 Active Learning for Classification

Active learning (AL) methods have been well explored for vision and learning tasks, see Settles [27] for a detailed review of active learning methods for class-based learning. Typically, the AL methods select a single instance with the maximum individual utility for annotation in each iteration. The utility of an instance is decided by different heuristics, e.g. uncertainty sampling [20], query-by-committee (QBC) [8], expected gradient length (EGL) [2], and model-output-change (MOC) [6]. The simplest and most widely applicable uncertainty sampling approach has been extended to modern deep learning frameworks and variational inference [29]. However, in all these methods, each sample's utility is evaluated independently without considering dependence between them.

In batch-mode active learning, data items are assessed not one at a time but in batches, to reduce the number of times the model is retrained. To avoid selecting correlated batches, some recent attempts evaluate the whole batch's utility by taking mutual information between samples into account. In contrast to our work, most of them are developed for classification tasks [2,17,26,24]. For example, Kirch *et al.* [17] define the utility score as the mutual information between data points in a batch and model parameters and then pick a subset with the maximum score. Pinsler *et al.* [24] formulate the active learning problem as a sparse subset selection problem approximating the complete data posterior of the model parameters. Both methods have a similar motivation to our work, but they are developed for the classification task, and their informativeness measures are not easy to extend to the metric learning task. Ash *et al.* [2] use the norm of the gradient at a sample to implicitely capture both informativeness and diversity, and select a subset of the farthest samples in the gradient space. This ensures both informativeness and diversity by a single gradient-based measure, which does not work well in the metric learning task, as shown by Kumari *et al.* [20]. Sener and Savarese [26] follow a similar strategy in a different feature space. Shui *et al.* [28] introduce a unified approach for training and batch selection process and explicitly define uncertainty-diversity trade-off by adopting Wasserstein distance.

2.3 Active Learning of Perceptual Metrics

There are only a few works on active learning of a perceptual metric. Most of these, e.g. [31,9], are based on a single instance evaluation criterion. They define the utility of a single triplet and select a batch of the individually highest-utility triplets to annotate. In contrast, we define a utility score for a batch taking joint information between triplets into account. The closest work to ours is a very recent paper by Kumari *et al.* [20]. The algorithm involves a two-stage process. First, it selects an overcomplete set of individually highly informative samples, and then subsamples a less correlated subset, using different triplet-based decorrelation heuristics, as the current batch. This method, in essence, is still based on a single triplet selection strategy. In contrast, we present a new, rigorous approach to define utility for a batch as a whole based on *joint entropy*, providing a unified utility function to balance both informativeness and diversity.

3 Proposed Method

In this section, we first briefly describe the perceptual metric learning setup and the underlying neural network-based learner called PerceptNet [19]. Next, we introduce our novel batch selection policy explicitly designed for triplet-based active metric learning.

3.1 Triplet-Based Active Metric Learning

Let $X = \{x_i\}_1^n$ represent a set of n objects, each described by a d-D feature vector x_i. Also, let T_L be a set of ordered triplets, where each triplet (x_i, x_j, x_k) indicates that the object x_i is more similar to object x_j than to x_k. For brevity we denote (x_i, x_j, x_k) by ijk. We frame the perceptual metric learning problem as learning an embedding

$\phi : R^d \rightarrow R^{\hat{d}}$, s.t. the L_2 distance between any two objects in the embedding space $d_\phi(x, y) = \|\phi(x) - \phi(y)\|$ reflects the perceptual distance between them. In recent work, ϕ is typically modeled with a neural network: in our experiments, we choose the existing PerceptNet model [19], where three copies of the same network, with shared weights, process three objects x_i, x_j and x_k during training. The output is optimized with an exponential triplet loss $\mathcal{L} = \sum_{T_L} e^{-\left(d_\phi^2(x_i,x_k) - d_\phi^2(x_i,x_j)\right)}$ to maximize the distance margins (a.k.a "triplet margins"), as defined by the exponent, for training triplets.

The number of possible triplets is cubic in the number of objects, so annotating a significant fraction of them is often intractable, e.g. in domains such as haptics and food tasting where annotation is especially slow. However, an effective embedding can be modeled with far fewer comparisons if triplets are sampled selectively based on *how much information* they would provide if annotated. This calls for active learning. The model is trained iteratively: batches of triplets informative to the current model are selected for annotation in each iteration, after which the model is retrained. However, the efficiency gain of selecting larger batches may be undone by *correlation* among triplets in a batch implying low overall information, a common issue in independent optimization of individual informativeness of each triplet. To mitigate this, prior works have studied *batch decorrelation* strategies for classification [17,2,24]. Recently, Kumari *et al.* [20] developed a decorrelation strategy for metric learning with separate steps for optimizing individual triplet informativeness and then batch diversity. However, as already noted, this work suffers from limitations related to their design choices. In contrast, we develop a method that jointly defines and optimizes the informativeness of an entire batch while implicitly ensuring diversity. The method is grounded in the Maximum Entropy Principle and leads to an attractive computational scheme.

3.2 Joint Entropy Measure for Batch Selection

The key to a good batch mode active learning is an effective informativeness measure for a batch of triplets. For tractability, earlier work typically defines a measure adding up the individual informativeness scores of triplets. A popular score is the Shannon entropy of the prediction probability p_y of the current model trained on prior batches, for a triplet t taking one of two possible orderings $y \in \{ijk, ikj\}$, $H(t) = -\sum_{y \in \{ijk,ikj\}} p_y \log p_y$ [31]. While often termed "uncertainty", this is not a good predictor of actual model uncertainty due to possible bias in the current model [7]. Further, individually high-entropy triplets may also have high mutual information, hence simply adding up the scores may overestimate the actual utility of the batch.

We propose a novel batch selection algorithm based on the *joint entropy* of an entire batch of triplets B, capturing their mutual dependence. We define the joint probability distribution of a set of unannotated triplets on some feature space such as their distance margins. This probability is defined using the *distribution of likely models* given prior batches, reducing any bias due to model training. Note that this is **quite different** from the *prediction probability* of a single fixed model, described above. The joint distribution over a set of triplets naturally captures the notion of interdependence among them.

3.3 Maximum-Entropy Model of the Joint Distribution

Our goal is to postulate the joint probability distribution of unannotated triplets in a batch, preferably in a form that allows efficient computation of its entropy. We represent each triplet t by its distance margin $\xi_t = d_\phi^2(x_i, x_k) - d_\phi^2(x_i, x_j)$. Then a batch, denoted $B = \{t_1, t_2, \ldots, t_b\}$ is represented by the vector of distance margins given by $\vec{\xi}_B = [\xi_{t_1}, \xi_{t_2}, \ldots, \xi_{t_b}]$.[3] We assume there is uncertainty about these margin predictions arising from the fact that there is a *distribution of plausible models* given the previously annotated data. Hence, each distance margin ξ_{t_i} is a 1D random variable taking different values for different choices of model parameters ϕ. As discussed above, simply looking at the predicted ordering probabilities of individual triplets is both error-prone and fails to consider correlation between triplets. Fortunately, if the model is a neural network, it has been shown that random dropout yields a good Bayesian approximation of model uncertainty [7]. We stochastically apply the dropout K times to the model, evaluating $\vec{\xi}_B$ each time, to sample the joint margin vector distribution of the batch and to compute the corresponding b-dimensional mean and covariance matrix. We invoke the Maximum Entropy Principle [12] which maximizes the Shannon entropy subject to constraints on prescribed averages. The maximum entropy distribution, consistent with all prior constraints, ensures the largest amount of uncertainty with respect to unknown, and hence introduces no additional biases in the estimation. Empirical estimates of the entropy of the batch from samples are susceptible to noise, and lead to a hard combinatorial optimization over batches. So we constrain the mean and covariance matrix of the triplet margins to match their empirical values $\vec{\mu}_B$ and Σ_B and maximize the differential entropy $H(B) = -\int p(\vec{\xi}_B) \log p(\vec{\xi}_B) d\vec{\xi}_B$ subject to these constraints. This leads to a multivariate gaussian distribution $N(\vec{\mu}_B, \Sigma_B)$ with entropy

$$H(B) = \frac{1}{2} \log \left((2\pi e)^b \det(\Sigma_B) \right) \tag{1}$$

Note that this score takes into account inter-triplet correlation, unlike measures depending only on individual marginals. The next task is to efficiently select an optimum batch of size b with maximum informativeness: $B^* = \arg\max_{B \subset T_U, |B|=b} H(B)$, where T_U is the set of currently unannotated triplets.

3.4 Greedy Algorithm for Batch Selection

Since the maximization of the joint entropy function $H(B)$ over subsets is computationally prohibitive, we use the fact that entropy is monotone increasing and submodular to justify a greedy policy which is provably $(1 - \frac{1}{e})$-optimal by the results of Nemhauser *et al.* [22]. The greedy algorithm builds up the set B^* incrementally. In step k, we pick the triplet t_k which has maximum conditional entropy given triplets B_{k-1} selected in

[3] Other triplet based representations are possible: we found the above to be a consistent and more useful feature in practice.

Algorithm 1 Greedy algorithm to maximize $H(B)$

```
Input: Unlabeled triplets $T_U$, batch size $b$, entropy function $H : 2^{T_U} \to \mathbb{R}$ as in Eq. 1.
Output: Batch $B$ that is an $(1 - \frac{1}{e})$-approximation to $\arg\max_{B \subset T_U, |B|=b} H(B)$.

$B_0 \leftarrow \emptyset$, $H(B_0) = 0$
for k=1,..., b do
    $t_k \leftarrow \arg\max_{t \in T_U \setminus B_{k-1}} \log\left(\det\left(\Sigma_{B_{k-1} \cup \{t\}}\right) / \det\left(\Sigma_{B_{k-1}}\right)\right)$
    $B_k \leftarrow B_{k-1} \cup \{t_k\}$
end for
return Final subset $B_k$
```

previous steps. Specifically,

$$\begin{aligned} t_k &= \arg \max_{t \in T_U \setminus B_{k-1}} H(\{t\} \mid B_{k-1}) \\ &= \arg \max_{t \in T_U \setminus B_{k-1}} H(B_{k-1} \cup \{t\}) - H(B_{k-1}) \\ &= \arg \max_{t \in T_U \setminus B_{k-1}} \log \left(\frac{\det\left(\Sigma_{B_{k-1} \cup \{t\}}\right)}{\det\left(\Sigma_{B_{k-1}}\right)} \right) . \end{aligned} \tag{2}$$

This step is repeated $|T_U|$ times. The greedy policy has low complexity (quantified later) and scales well to large datasets. The overall batch selection algorithm is listed in 1. The remaining challenge is to efficiently compute the increment in the determinant of the covariance matrix in each step. We present a recursive algorithm for this, which also clarifies why the method selects a decorrelated batch.

3.5 Recursive Computation of Determinant of Covariance Matrix

The covariance matrix is a Gram matrix, i.e. its $(i, j)^{\text{th}}$ element can be written as the dot product of the i^{th} and j^{th} vectors from a given family of vectors. This allows us to recursively compute its determinant and choose the recursion order according to the greedy policy for approximate optimization. Let u_t denote the zero-mean vector of all sampled distance margins, $[\xi_t(\phi_1), \cdots, \xi_t(\phi_K)] - [\mu_t, \cdots, \mu_t]$, for a single triplet t. The covariance matrix $\Sigma_{B_{k-1}}$ has the form UU^T, where each column of U is $u_s, s \in B_{k-1}$. In the k^{th} step, a new row and column vector for a new triplet t are appended to U. Using the Gram matrix property, we have $\det(\Sigma_{B_{k-1} \cup \{t\}}) - \det(\Sigma_{B_{k-1}}) = \|\tilde{u}_k\|^2$, where $\tilde{u}_k$ is the normal from u_t onto span$\{u_s \mid s \in B_{k-1}\}$. Thus the scheme successively maximizes the squared projection error $\|\tilde{u}_k\|^2$, over the remaining vectors $\{u_t \mid t \in T_U \setminus B_{k-1}\}$. Thus we select at each step the triplet that is least correlated with the already chosen triplets. The orthogonal projections are computed using the modified Gram-Schmidt orthogonalization scheme from [10], with complexity dn^2, where d is the dimension of the ambient vector space and n the number of vectors. Since we compute the projection error for all $|T_U| - n \approx |T_U|$ remaining triplets at each step (because $|T_U| >> n$), the overall complexity of the scheme is $dn^2|T_U|$.

In summary, the submodularity of the joint entropy function naturally combines informativeness, diversity, and representativeness, which are precisely the desired properties for batch mode active learning.

4 Experiments

We perform several experiments to answer the following questions: (1) Is our method competitive with standard baselines, including the state-of-the-art method(s), for different choices of hyperparameters, feature dimension, applications, and datasets? (2) How good is our assumption that the second-order statistics (mean and covariance) are sufficient statistics for estimating the reasonable distribution? (3) How robust is our method to labeling error? We address these questions by conducting several experiments on real-world datasets with different modalities: image, food and haptic. For each of these datasets, we select an appropriate neural network architecture – for the haptic and food datasets we ensure that these architectures exactly match those of Kumari et al. [20] so that the comparison is fair ([20] did not present any result on images, requiring us to implement their method on image databases). We test with different initial pools and varying batch sizes. We also simulated random errors in the triplet orderings to test robustness to labeling error.

Datasets. We evaluate the performance of our method on five real-world datasets for which triplets defining perceptual metrics are available: Yummly food dataset [34]; TUM haptic texture dataset [30]; Abstract500 image dataset [25]; CUB-200 image dataset [32], and Scoot facial sketch dataset [5]. The **Yummly-Food** dataset has 72148 triplets defined over 73 food items based on taste similarity. Each food item is represented by a 6D feature vector (this is an experiment with a low feature dimension) with each component indicating different taste properties. We use 20K training and 20K test triplets sampled from the entire set of triplets. The **TUM-Haptic** dataset contains signals from 108 different types of surface materials. Each type of material has 32-D spectral feature vectors for 10 representative acceleration traces. The triplets are generated from a given ground-truth perceptual matrix, which has user-recorded perceived similarity responses. Like the Yummly-Food dataset, we have training and test sets of 20K triplets each. We also evaluate our method on a comparatively larger dataset (but relatively small for image data), the **Abstract500** image dataset [25], which contains 500 images of 128×128 pixels, with pairwise perceptual similarities between them. Each image is represented by a 512-D GIST feature (an example of a relatively high-dimensional feature vector) extracted using 32 Gabor filters at four scales and eight orientations [23]. We use perceptual matrix to generate 20K training and 20K test triplets. Next, we use the popular and much larger **CUB-200** bird database that contains 200 bird species with roughly 30 images in each class. We choose five representative images for each class and generate its features using a pretrained ResNeXt-101-32x8d model. The network takes segmented images as input and outputs 2048-D feature vectors. The training and test sets each have 10K triplets sampled from the entire set of 93530 triplets. Finally, the results on the **Scoot** dataset, which is relatively small, consisting of just 1282 triplets, are presented in the supplementary material because of space constraints.

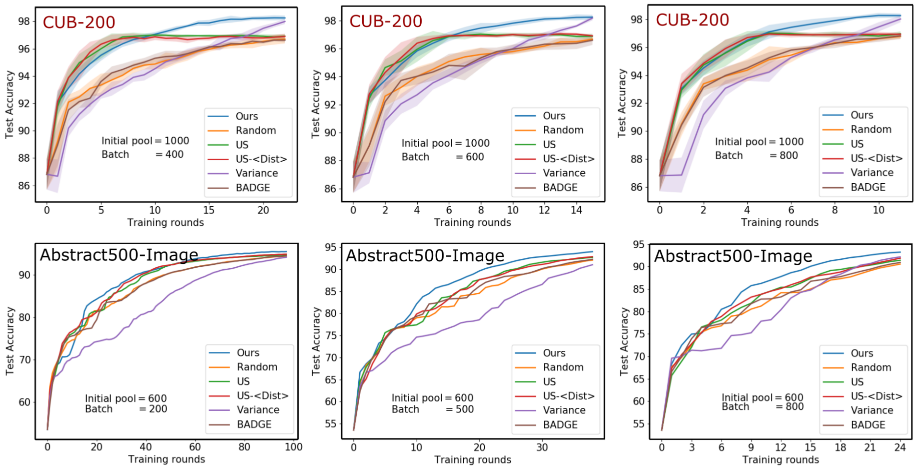


Fig. 3: Performance of different active learning methods on two image datasets CUB-200 and Abstract500 for increasing batch sizes (400/600/800 or 200/500/800, from left to right). Here accuracy means what fraction of test triplets have been detected with the correct ordering. To avoid clutter, standard deviations are shown only for the CUB-200 dataset and the rest are shown in the supplementary material.

Baselines. We compare our method with five baselines, including the state-of-the-art method: (1) **US-$\langle$Dist$\rangle$:** A batch of individually high-entropy triplets is pruned subjected to different (denoted by $\langle$Dist$\rangle$) decorrelation measures to select a diverse batch of informative triplets [20]. It is the current state-of-the-art for batch mode active metric learning, and outperforms other alternatives like BADGE [2] (adapted to metric learning). We pick $\langle$Dist$\rangle$ to be the highest-performing variant in each individual experiment. (2) **Variance:** Triplets with the highest individual distance-margin variance across a collection of models generated using dropout [13]. This method simulates the effect of replacing the joint entropy of a batch with the sum of individual entropies of triplets in the batch. (3) **Random:** A passive learning strategy that uniformly samples each batch of triplets at random. Though naïve, this choice often results in reasonably good accuracy. (4) **US:** Uncertainty method, which picks the top b triplets with highest uncertainty in predicted triplet ordering (i.e. the model's (lack of) ordering confidence), without taking correlation among them into account [31]. (5) **BADGE:** A diverse set of triplets with maximum loss gradients for the most probable label, selected using k-means ([2] adapted to the triplet scenario).

Active Learning Setup. For the CUB-200 dataset, we begin each experiment with an initial pool of 1000 annotated triplets, and for the other three datasets with 600 annotated triplets, and pretrain the model ϕ_0, which is used as a common starting point for all compared methods. In each active learning iteration, we select the best fixed-size batch of unannotated triplets, using the chosen batch selection method, and acquire their orderings. To make convergence faster, we update the current model ϕ_i to obtain ϕ_{i+1} using the available additional annotated triplets instead of training *ab initio*. The performance of the learned model is evaluated by its *triplet generalization accuracy*,

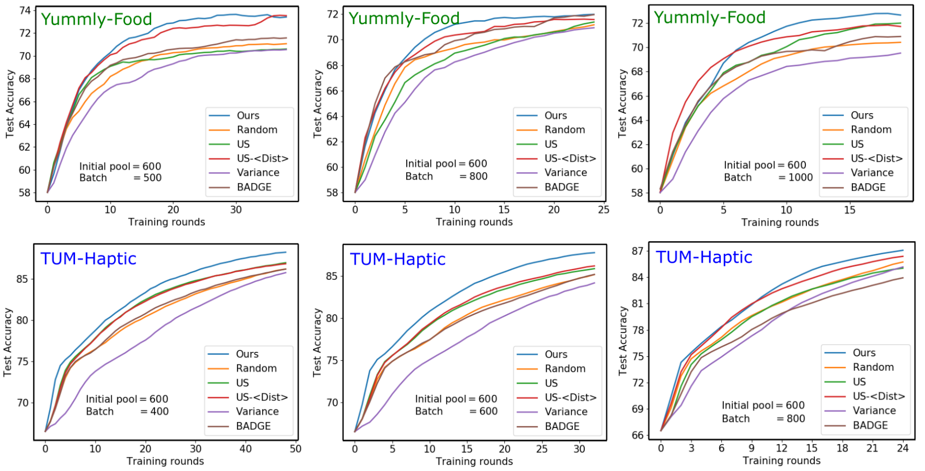


Fig. 4: Performance of different active learning methods on Yummly-Food and TUM-Haptic texture datasets for varying batch sizes (500/800/1000 or 400/600/800).

which denotes the fraction of triplets whose ordering is correctly predicted [19]. Each experiment is repeated with five random train/test splits, and the average performance along with the standard deviation is reported. (For most plots, the standard deviation is shown in supplementary material, for clarity.)

Implementation Details. The architecture and training hyperparameters used for different datasets are as follows: Yummly-Food: 3 fully-connected (FC) layers with 6, 12 and 12 neurons; TUM-Haptic: 4 FC layers with 32, 32, 64 and 32 neurons; Abstract500-Image: 6 FC layers with 512, 256, 128, 64, 32 and 16 neurons; CUB-200: 3 FC layers with 2048, 512 and 32 neurons. Each layer is followed by a dropout layer with a dropout probability of 0.02. The Adam optimizer [16] is used for training all models with a learning rate of 10^{-4} for Yummly-Food, TUM-Haptic, and Abstract500-Image dataset, and 10^{-5} for CUB-200 dataset. The model is trained with an SGD batch size of 500 for 1000 epochs for all four datasets.

Active Learning Performance. The performance of our method against the baselines described earlier is plotted in Figure 3 for the image datasets CUB-200 and Abstract500-Image, shown as a function of the number of active learning iterations. In Figure 4, we compare the performances of all methods for the data from other modalities, i.e., haptic and food. We observe that our method is consistently better than the state-of-the-art **US-$\langle$Dist$\rangle$** method (for clarity, we only show the specific variant offering the best performance in each experiment). Our method reaches higher accuracies quicker and also tends to converge to a higher final accuracy on both Yummly-Food and TUM-Haptic datasets. For the large CUB-200 image dataset, our method is neck-and-neck with the state-of-the-art for the first few iterations and then rapidly overtakes it, widening the gap with additional iterations. For the smaller Abstract500 image dataset, the improvements are more prominent with larger batch sizes, reflecting the focus of our work on batch-mode learning. Additionally, for the CUB-200 dataset, we plot the standard devi-

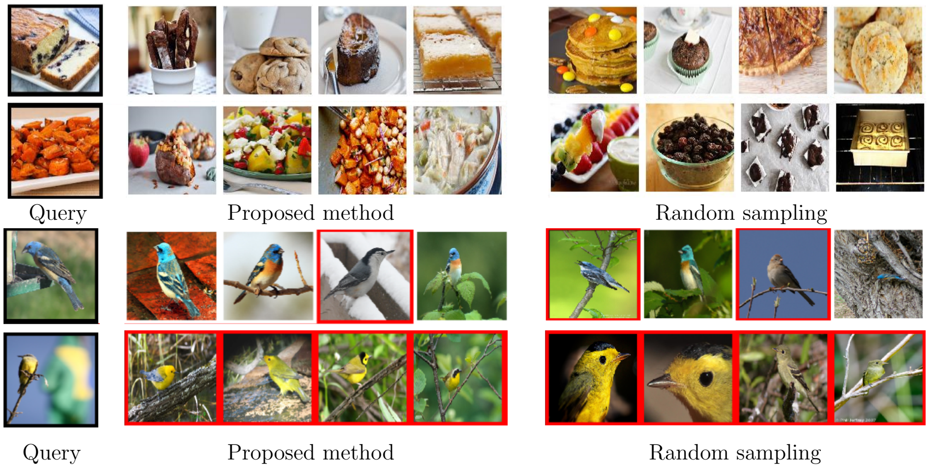


Fig. 5: Top-4 retrieved images in the order of increasing perceptual distance, left to right, using our method and random sampling (randomly-selected batches are annotated for training) on different modalities datasets. On both datasets, each model is trained for twelve training rounds, constituting 18% of training triplets. Images different from the query class are bounded by a red box, substantiating that two images from different classes can be perceptually more similar than two from the same class. The *triplet order accuracy*, defined here as the number of test triplets whose order is preserved by the ranked list of retrieved images, for our method vs random sampling after the 12th round of training is $M^{12}_{\text{Ours}} = 96.7\%$, $M^{12}_{\text{Random}} = 92\%$ for image dataset and $M^{12}_{\text{Ours}} = 72.9\%$, $M^{12}_{\text{Random}} = 69.3\%$ for food dataset. More results with different queries and learned metrics are shown in the supplementary material.

ation in the same plot as the shaded region (of the same color) around the performance curves for different methods (standard deviations on other datasets are shown in supplementary). Even though the figure looks a little cluttered, one can see that the standard deviation for the proposed method is better than that of the next-best method, signifying a more consistent performance. This substantiates our claim that joint entropy is a better batch score than an ad-hoc combination of independent informativeness and diversity heuristics. Further, our method does not require the user to select a suitable decorrelation heuristic to manually fine-tune the performance.

We also outperform the other two baselines: **Random** and **Variance**. It is particularly informative to see the generally poor performance of **Variance** (lower than the Random). Because of high correlations among informative triplets, individually selecting the most informative triplets does not learn the entire metric space as well as just picking triplets at random. In contrast, our method as well as that of Kumari *et al.* [20] both incorporate batch decorrelation and outperform random sampling. This shows the critical importance of batch diversity in an active learning strategy.

Next, we evaluate the effectiveness of our method for an **object retrieval** task. Specifically, we compare our method with the random sampling baseline at different training rounds. We show the retrieval results on two different modalities, food and image. We split the Yummly-Food dataset into 40000 training and 32148 test triplets, and

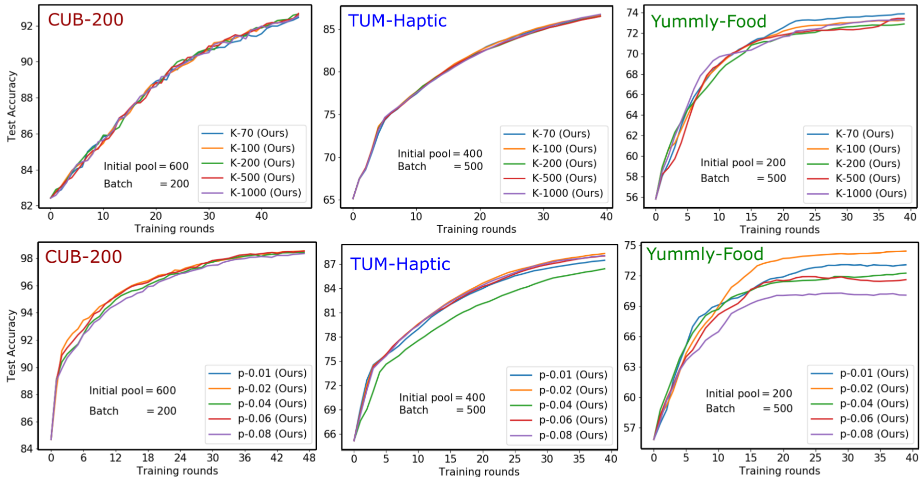


Fig. 6: Ablation study to analyze the robustness of our method to different values of K (# of prior models sampled by dropout) and p (dropout probability) on different datasets.

the CUB-200 dataset into 40000 training and 33000 test triplets. On both datasets, we perform active learning with a batch size of 600 and an initial pool of 500 triplets. For a given query image, the top four instances from the retrieval set are shown (ranked from most similar to least similar) in Figure 5. As we can see, retrieval results of our method resemble the query in taste or visual appearance better than the random sampling. Please see the supplementary material for further results from this experiment.

Ablation Study. In order to get an estimate of the covariance matrix, we perform random dropouts in the neural network K times. Naturally, as K increases, one gets a better estimate of the covariance matrix. However, this may increase the computation time. We perform an ablation study to see how this hyperparameters (i.e., variation in K and dropout probability p) affect the triplet order accuracy, and the results are shown in Figure 6 for three different modalities: image, haptic and food. It can be seen from the plots that a moderate value of about $K = 70$ or 100 is good enough as the performance is not significantly dependent on the choice of K. We also observe that the performance is robust to variation in dropout probability; however, there is a significant variation for the Yummly-Food dataset, with optimal $p = 0.02$.

Runtime Analysis. We also compare the computational requirement of the proposed method with that of Kumari *et al.* [20]. The key computational step in [20] involves searching for the subset of maximally apart (in the feature space) triplet at each training round, apart from the computation of the gradients. They also use a greedy search technique for subset selection. For the proposed method, the subset selection process is efficient, but the computation of determinant of covariance matrix at each iteration does consume a good amount of time. Overall, both the methods were found to consume a nearly equal amount of computation time when the Gram-Schmidt orthogonalization is used. For instance runtimes (in secs) of different batch selection policies to select a 500-triplet batch from Yummly-food are: US: 0.109, Variance: 0.083, BADGE: 60.104, US-⟨Dist⟩: 8.110, Ours: 7.803. While the computation complexity varies with the feature dimension and model size, the relative performance remains similar.

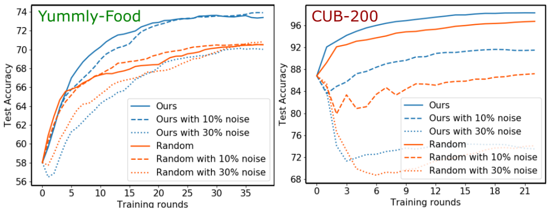


Fig. 7: Performance of our method vs random sampling in the presence of labeling error.

***Robustness to Labeling Error*.** To evaluate the robustness of our method against labeling error, we corrupt 10% and 30% of the ground-truth training triplets in the food and image datasets by flipping their orders. Figure 7 shows how the noisy training set affects the performance of our method vs the random sampling baseline. For the food data (top), with a relatively low 10% labeling error, our method takes slightly more iterations to gain accuracy, but eventually converges to a comparably high accuracy as the noise-free case, while random batch selection fails to achieve the same performance even with clean data. As the percentage of noisy triplets increases, the performance of both methods degrade, showing vulnerability to large scale labeling error. In the absence of abnormally high levels of outliers, our method shows robust performance. For the more complex image dataset (bottom), labeling error has a stronger negative effect (the first selected batch actually decreases overall accuracy), but at each noise level our method still outperforms the baseline.

***Comparison of Data Distribution to Theoretical Distribution*.** We study the validity of the Gaussian embedding, though it already has justification as "worst-case analysis" due to the Maximum Entropy Principle. A standard test is the quantile-quantile (QQ) plot [4], which indicates how close the empirical distribution is to the theoretical distribution. For ease of visualization, we show the QQ plot and histogram for a single randomly-selected unlabeled triplet, for a particular model trained on the initial triplet pool in each dataset. (We cannot visualize a full multivariate QQ plot over all possible batches.) In the QQ plot, the x-axis denotes the theoretical quantiles, which in our case is a Gaussian distribution with the empirical mean and variance, and the observed ordered distance margins are on the y-axis. The *goodness of fit* is indicated by the alignment of points with the straight line having a unit slope. As shown in Figure 8, in all four datasets, the plotted curve closely approximates the corresponding straight lines shown in red. Our approximation is further validated in the histogram, where our data distribution shows a reasonable fit with the theoretical distribution (shown in green) for the most part, except that the actual distribution is a little more peaked.

5 Conclusion, Limitations, and Future Work

We have introduced a novel approach for batch-mode active metric learning based on maximizing the joint entropy of a batch. We found that a batch of individually informative triplets does not form an optimal subset, even if decorrelation heuristics are applied to reduce their correlation. Instead of defining separate measures for informativeness

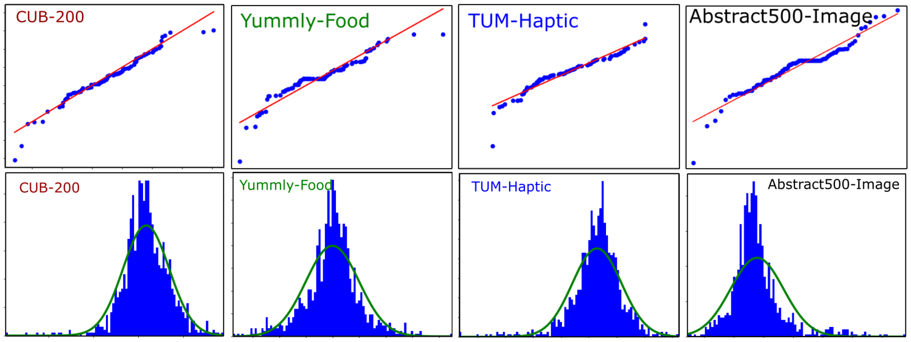


Fig. 8: QQ plot and histogram for all four datasets to demonstrate how closely the actual distribution follows the theoretical distribution.

and diversity, our method defines the joint entropy of a batch of triplets as a unified measure that jointly optimizes both. The overall method involves no heuristic parameter selection and has no control parameter to tweak, other than the number of dropout samples and dropout probability, once the network architecture is chosen.

While our method shows promising results, it does have a few limitations. First, approximating the joint distribution of data using the Maximum Entropy Principle gives the most general distribution for a given prior, which in the case of second-order statistics as constraints is a Gaussian. However, in some cases, where the actual distribution may be quite non-Gaussian, the joint entropy measure defined with the second-order statistics may misguide the batch selection policy. One important direction for future work is extending our framework beyond second-order statistics to learn the joint distribution of data closer to empirical distribution. Another important extension would be to modify our framework to dynamically learn the optimal batch size and batch selection policy, which we believe would further improve the performance and generalize well to diverse inputs and applications.

References

1. Agarwal, S., Wills, J., Cayton, L., Lanckriet, G., Kriegman, D., Belongie, S.: Generalized non-metric multidimensional scaling. In: AIS (2007) 4
2. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: ICLR (2020) 4, 5, 6, 10
3. Bellet, A., Habrard, A., Sebban, M.: A survey on metric learning for feature vectors and structured data. arXiv preprint arXiv:1306.6709 (2013) 4
4. Ben, M.G., Yohai, V.J.: Quantile–quantile plot for deviance residuals in the generalized linear model. Journal of Computational and Graphical Statistics (2004) 14
5. Fan, D.P., Zhang, S., Wu, Y.H., Liu, Y., Cheng, M.M., Ren, B., Rosin, P.L., Ji, R.: Scoot: A perceptual metric for facial sketches. In: ICCV (2019) 4, 9
6. Freytag, A., Rodner, E., Denzler, J.: Selecting influential examples: Active learning with expected model output changes. In: ECCV (2014) 4
7. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: ICML (2016) 3, 6, 7
8. Gilad-Bachrach, R., Navot, A., Tishby, N.: Query by committee made real. In: NeurIPS (2006) 4

9. Heim, E., Berger, M., Seversky, L., Hauskrecht, M.: Active perceptual similarity modeling with auxiliary information. In: AAAI (2015) 3, 5
10. Hoffmann, W.: Iterative algorithms for gram-schmidt orthogonalization. Computing (1989) 8
11. Jamieson, K.G., Nowak, R.D.: Low-dimensional embedding using adaptively selected ordinal data. In: Allerton (2011) 4
12. Jaynes, E.T.: Information theory and statistical mechanics. Physical Review (1957) 7
13. Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv preprint arXiv:1511.02680 (2015) 10
14. Kendall, M.G.: Rank correlation methods. Griffin (1948) 2, 4
15. Kim, S., Seo, M., Laptev, I., Cho, M., Kwak, S.: Deep metric learning beyond binary supervision. In: CVPR (2019) 2
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 11
17. Kirsch, A., van Amersfoort, J., Gal, Y.: BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In: NeurIPS (2019) 5, 6
18. Kruskal, J.B., Wish, M.: Multidimensional scaling. Elsevier (1978) 2, 4
19. Kumari, P., Chaudhuri, S., Chaudhuri, S.: PerceptNet: Learning perceptual similarity of haptic textures in presence of unorderable triplets. In: WHC (2019) 2, 4, 5, 6, 11
20. Kumari, P., Goru, R., Chaudhuri, S., Chaudhuri, S.: Batch decorrelation for active metric learning. In: IJCAI-PRICAI (2020) 1, 3, 4, 5, 6, 9, 10, 12, 13
21. McFee, B., Lanckriet, G.: Learning multi-modal similarity. JMLR (2011) 4
22. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions – I. Mathematical programming (1978) 3, 7
23. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV (2001) 9
24. Pinsler, R., Gordon, J., Nalisnick, E., Hernández-Lobato, J.M.: Bayesian batch active learning as sparse subset approximation. In: NeurIPS (2019) 5, 6
25. Robb, D.A., Padilla, S., Kalkreuter, B., J.Chantler, M.: Crowdsourced feedback with imageryrather than text: Would designers use it? In: SIGCHI (2015) 9
26. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. In: ICLR (2018) 5
27. Settles, B.: Active learning. SLAIML (2012) 4
28. Shui, C., Zhou, F., Gagné, C., Wang, B.: Deep active learning: Unified and principled method for query and training. In: AISTATS (2020) 5
29. Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: ICCV (2019) 4
30. Strese, M., Boeck, Y., Steinbach, E.: Content-based surface material retrieval. In: WHC (2017) 1, 4, 9
31. Tamuz, O., Liu, C., Belongie, S., Shamir, O., Kalai, A.T.: Adaptively learning the crowd kernel. In: ICML (2011) 3, 5, 6, 10
32. Wah, C., Maji, S., Belongie, S.: Learning localized perceptual similarity metrics for interactive categorization. In: WACV (2015) 4, 9
33. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: CVPR (2014) 1, 4
34. Wilber, M.J., Kwak, I.S., Belongie, S.J.: Cost-effective hits for relative similarity comparisons. In: AAAI (2014) 9
35. Xing, E.P., Jordan, M.I., Russell, S.J., Ng, A.Y.: Distance metric learning with application to clustering with side-information. In: NeurIPS (2003) 1, 4
36. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 2, 4

A Unified Batch Selection Policy for Active Metric Learning

Supplementary Material

In this supplementary material, we provide (a) plots with standard deviations for three datasets where only the mean accuracies were provided in the main paper, for clarity; (b) performance evaluation of different active learning methods on one additional dataset and (c) qualitative and quantitative results on a retrieval task with metrics trained with different methods.

1 Standard Deviation Plots

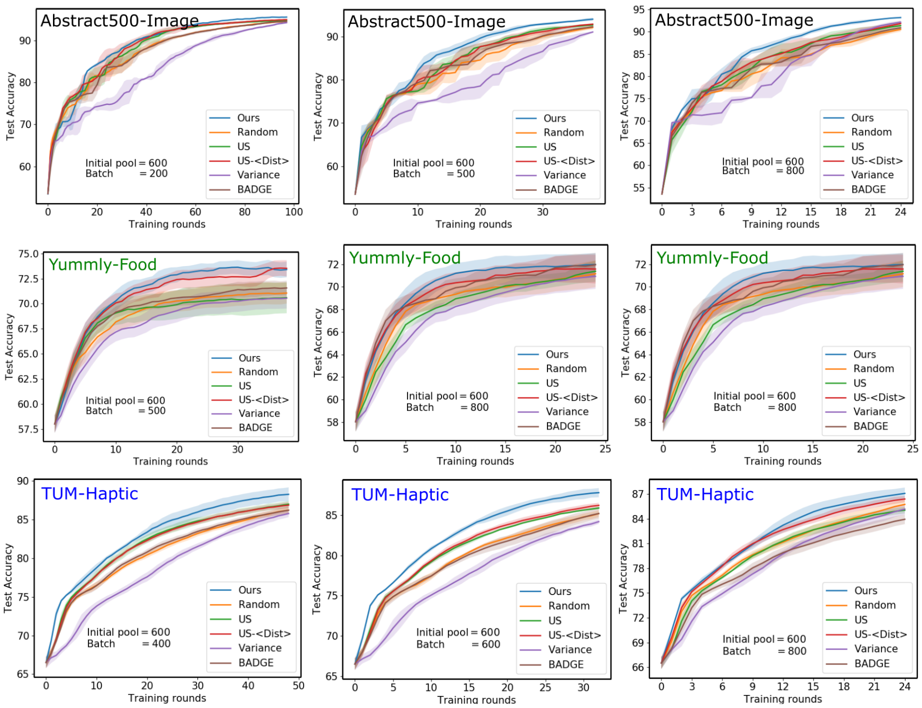


Fig. 1: Mean accuracy and standard deviation plots (computed over five random train/test splits) of different active learning methods for Abstract500-Image, Yummly-Food and TUM-Haptic datasets with increasing batch sizes.

In Figure 1, we augment the plots shown in the main paper with standard error bands across the five experimental runs for each method+hyperparameter combination for three real-world datasets: Abstract500-Image, Yummly-Food, and TUM-Haptic. (Bands for CUB-200 are directly shown in the main paper.) The standard error of our method is small for all datasets except the Yummly-Food dataset, where all methods have high variance. While our method always exceeds or matches the best alternative, the performance gain is observed to be higher for a larger batch size.

2 Performance of Active Learning Methods on Scoot Facial Sketch Dataset

The Scoot facial sketch dataset is relatively small, consisting of just 1282 triplets, where each triplet represents similarity ordering between three sketched faces of a person. The facial sketch is represented by a 512-D GIST feature extracted using 32 Gabor filters at four scales and eight orientations. The training and test set contains 800 and 200 triplets, respectively, sampled from the entire triplet set. The architecture and training hyperparameters used for Scoot facial sketch dataset are: 6 FC layers with 512, 256, 128, 64, 32 and 16 neurons. Each layer is followed by a dropout layer with a dropout probability of 0.02. The Adam optimizer is used for training the model with a learning rate of 10^{-6}. The model is trained with an SGD batch size of 500 for 1000 epochs.

As can be seen in Figure 2, in the initial training rounds with only a few annotated triplets, our method performs marginally better than other baselines, but the accuracy margin improves as the number of annotated triplets increases.

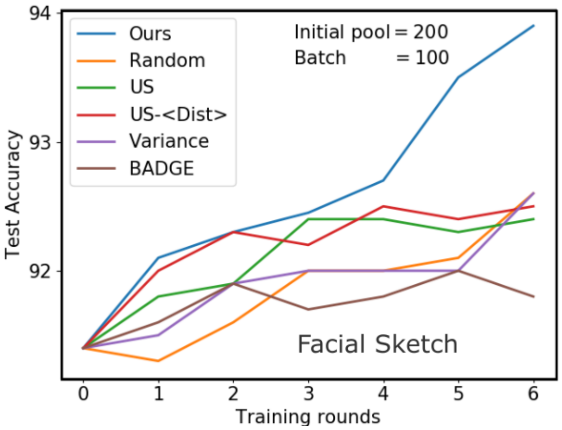


Fig. 2: Performance of different active learning methods on the Scoot dataset. Test accuracy indicates the fraction of test triplets correctly ordered by the learned perceptual metric.

3 Image Retrieval Task

In this section, we further evaluate the effectiveness of our method for an image retrieval task. We compare our method with the random sampling baseline at different training rounds on the CUB-200 and Yummly-Food datasets.

3.1 Retrieval Results on CUB-200 Dataset

The CUB-200 dataset consists of 200 classes with roughly 30 images in each class. Triplets are defined based on visual (perceptual) similarity of classes. We split the dataset into 40000 training and 33000 test triplets, and performed active learning with a batch size of 600 and an initial pool of 500 triplets. For a given query image, the top four instances from the retrieval set are shown (ranked from most similar to least similar) in Figures 3 and 4, after two different training rounds.

3.2 Retrieval Results on Yummly-Food Dataset

The Yummly-Food dataset consists of images of 73 food items and 72148 triplets defined on the basis of similarity in taste. We split the dataset into 40000 training and 32148 test triplets, and performed active learning with a batch size of 600 and an initial pool of 500 triplets. For a given query image, the top nine instances from the retrieval set are shown (ranked from most similar to least similar) in Figures 5 (9 training rounds) and 6 (20 training rounds).

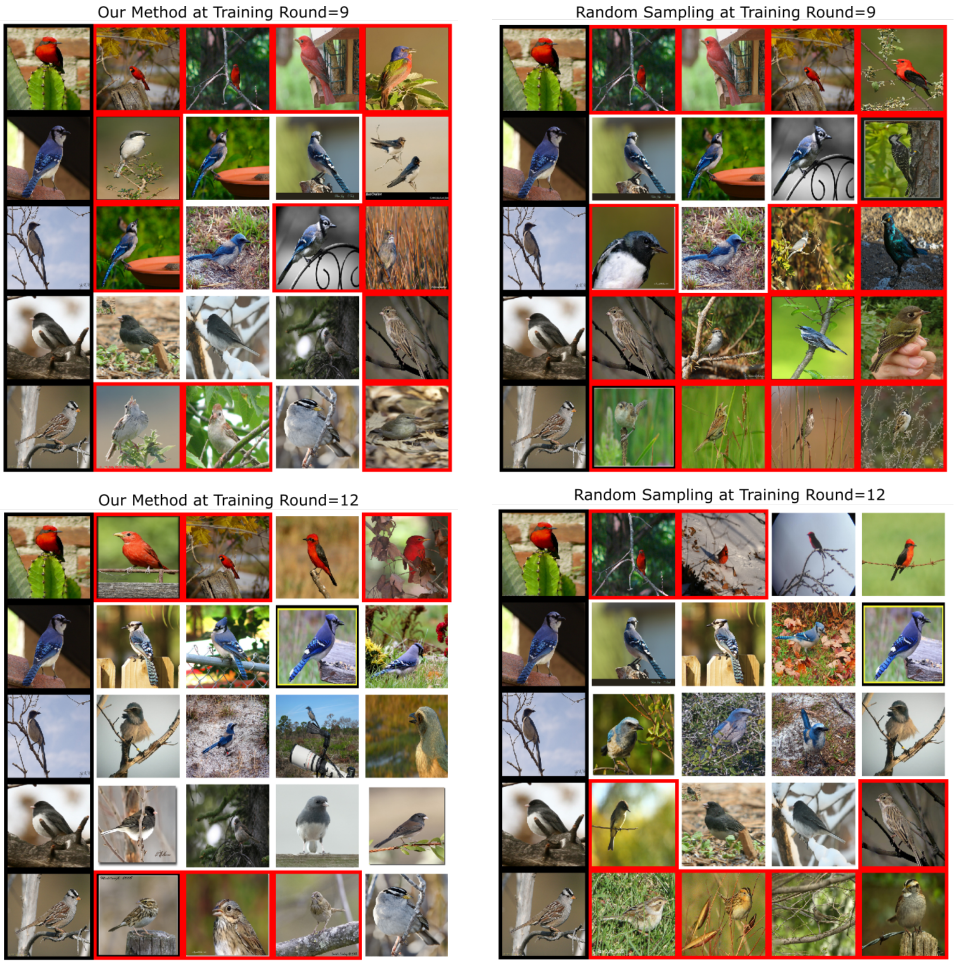
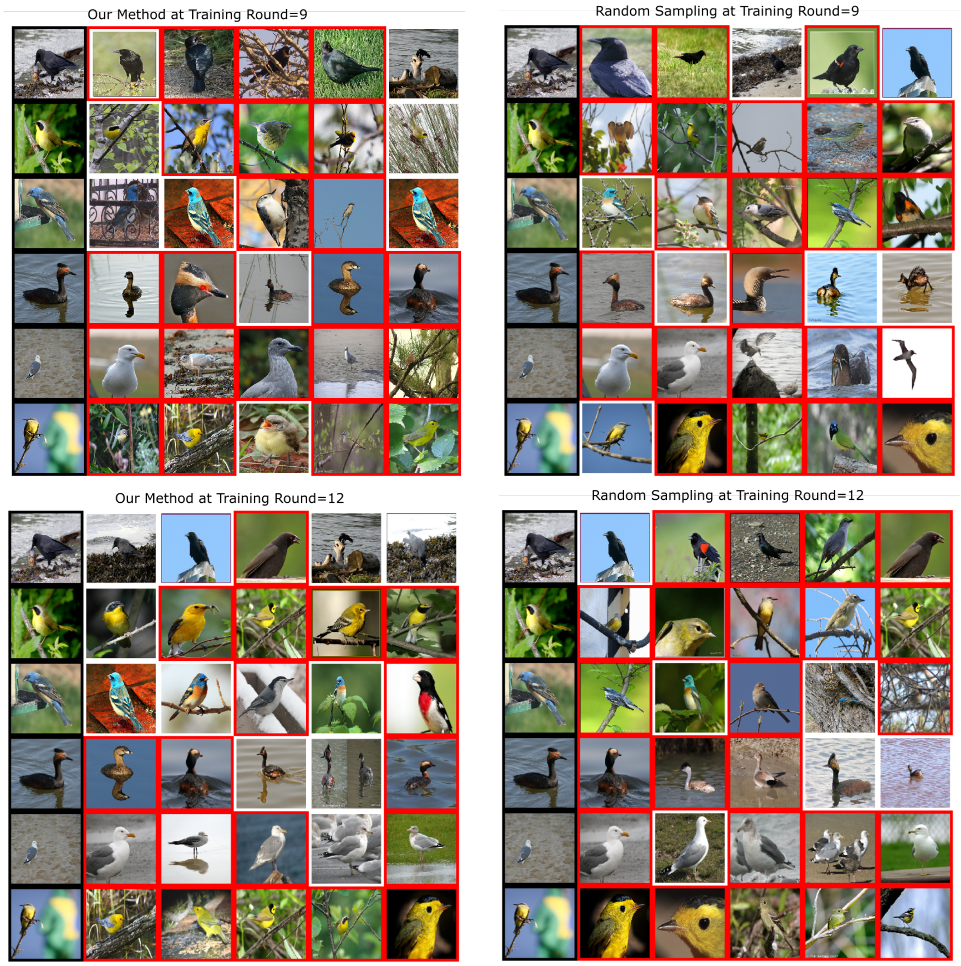

Our Method at Training Round=9
Random Sampling at Training Round=9
Our Method at Training Round=12
Random Sampling at Training Round=12

Fig. 3: Performance comparison between our method and random sampling for the image retrieval task after two different training rounds. The leftmost column presents a query, and the remaining columns present the first four retrieved images in the order of increasing perceptual distance, left to right. Images with red squares belong to classes different from the query class. Our results indicate that two images from different classes can be visually more similar than two from the same class, highlighting the distinction between perceptual and class-based metrics. The triplet ordering accuracy for $M^k_{Ours/Random}$ (model trained on triplets selected by our method vs random sampling resp. at k^{th} training round): $M^9_{\text{Ours}} = 95.3\%$, $M^9_{\text{Random}} = 90.3\%$, $M^{12}_{\text{Ours}} = 96.7\%$, $M^{12}_{\text{Random}} = 92\%$.



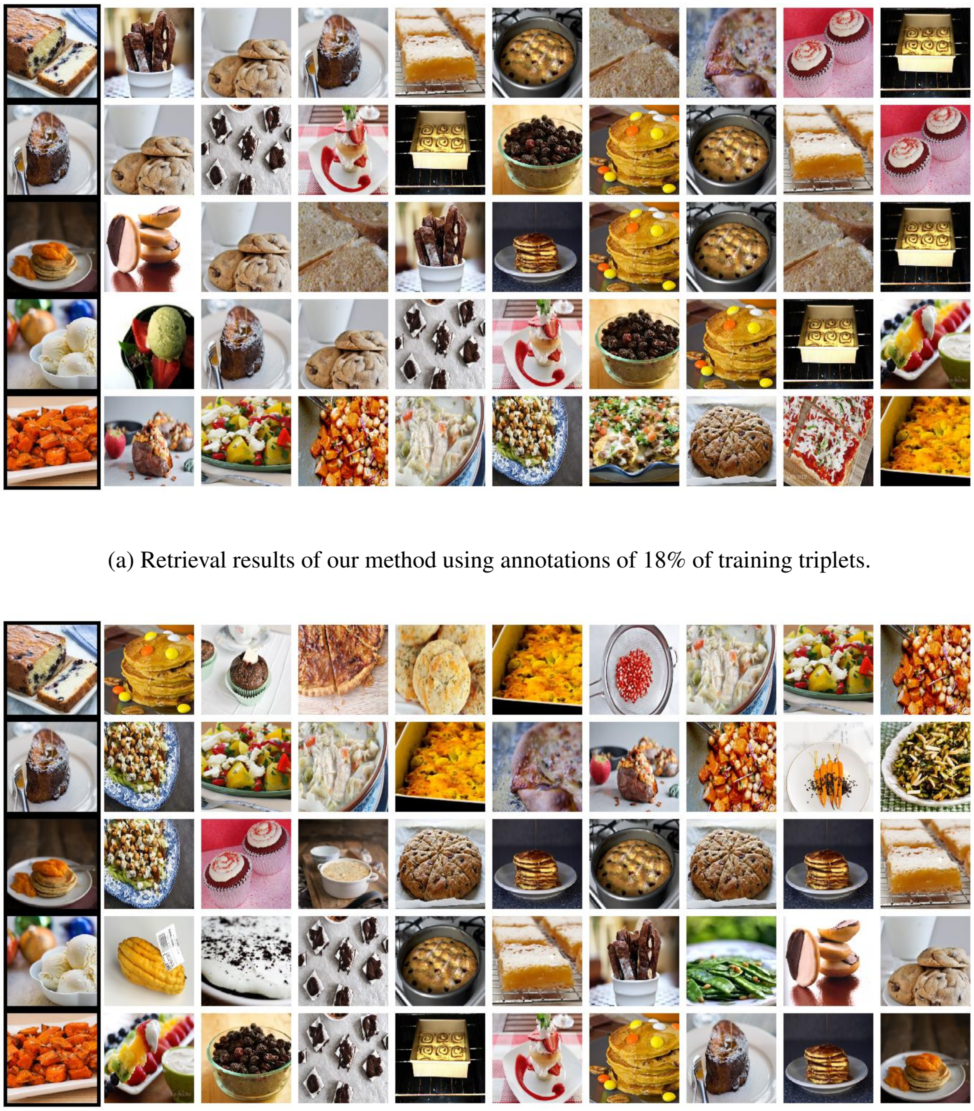
Our Method at Training Round=9

Random Sampling at Training Round=9

Our Method at Training Round=12

Random Sampling at Training Round=12

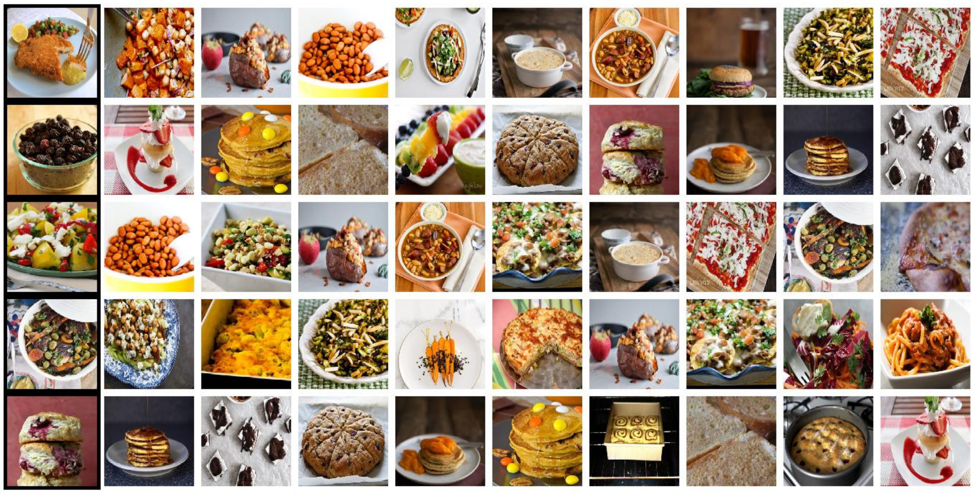
Fig. 4: Similar results as Figure 3, shown for a different set of query images in the leftmost column.



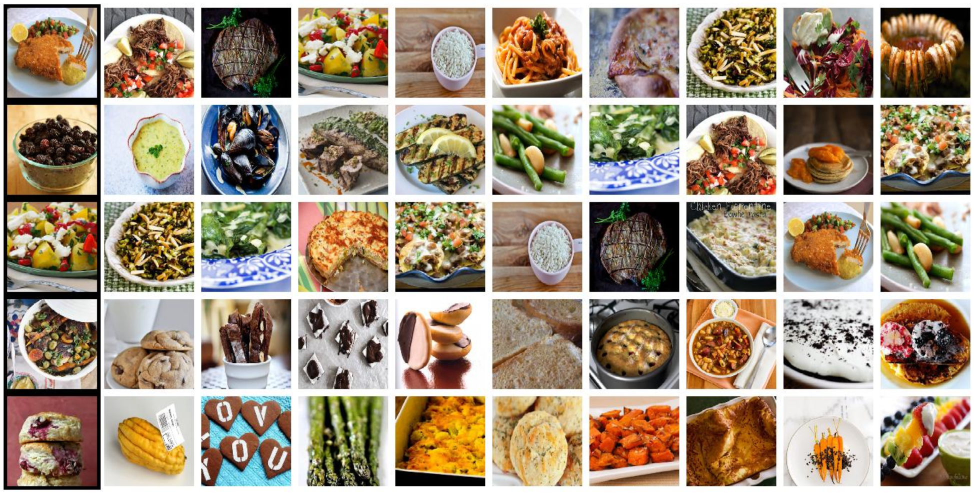
(a) Retrieval results of our method using annotations of 18% of training triplets.

(b) Retrieval results of the random sampling baseline using annotations of 18% of training triplets.

Fig. 5: Top-9 food dishes, retrieved according to taste similarity, by our method vs random sampling after twelve training rounds, for query images in the leftmost columns. The triplet ordering accuracy for $M^k_{Ours/Random}$ (model trained on triplets selected by our method vs random sampling resp. at k^{th} training round): $M^{12}_{\text{Ours}} = 72.9\%$, $M^{12}_{\text{Random}} = 69.3\%$. Note that both our method and random sampling solicit annotations of 18% of the training triplets; however, our method's retrieval results better resemble the query in taste, than those from random sampling.



(a) Retrieval results of our method using annotations of 30% of training triplets.



(b) Retrieval results of the random sampling baseline using annotations of 30% of training triplets.

Fig. 6: Top-9 retrieved food dishes by our method vs random sampling after twenty training rounds, for query images in the leftmost columns. The triplet ordering accuracy: $M^{20}_{\text{Ours}} = 73.6\%$, $M^{20}_{\text{Random}} = 69.7\%$.