Active Learning by Feature Mixing

Amin Parvaneh[1] Ehsan Abbasnejad[1] Damien Teney[1,2] Reza Haffari[3]
Anton van den Hengel[1,4] Javen Qinfeng Shi[1]

[1]Australian Institute for Machine Learning, University of Adelaide
[2]Idiap Research Institute [3]Monash University [4]Amazon

{amin.parvaneh, ehsan.abbasnejad,javen.shi, anton.vandenhengel}@adelaide.edu.au
damien.teney@idiap.ch gholamreza.haffari@monash.edu

Abstract

*The promise of active learning (AL) is to reduce labelling costs by selecting the most valuable examples to annotate from a pool of unlabelled data. Identifying these examples is especially challenging with high-dimensional data (*e.g. *images, videos) and in low-data regimes. In this paper, we propose a novel method for batch AL called ALFA-Mix. We identify unlabelled instances with sufficiently-distinct features by seeking inconsistencies in predictions resulting from interventions on their representations. We construct interpolations between representations of labelled and unlabelled instances then examine the predicted labels. We show that inconsistencies in these predictions help discovering features that the model is unable to recognise in the unlabelled instances. We derive an efficient implementation based on a closed-form solution to the optimal interpolation causing changes in predictions. Our method outperforms all recent AL approaches in 30 different settings on 12 benchmarks of images, videos, and non-visual data. The improvements are especially significant in low-data regimes and on self-trained vision transformers, where ALFA-Mix outperforms the state-of-the-art in 59% and 43% of the experiments respectively*[1].

1. Introduction

The success of machine learning applications depends on the quality and volume of the annotated datasets. High quality data annotations can be slow and expensive. Active learning (AL) aims to actively select the most valuable samples to be labelled in the training process iteratively, to boost the predictive performance. A popular setting called *batch* AL [34] fixes a budget on the size of the batch of instances to be sent to an oracle for labelling. The process is repeated over multiple rounds, allowing the model to be updated iteratively. The core challenge is therefore to identify the most valuable instances to be included in this batch at each round, depending on the current model.

[1]The code is available at https://github.com/aminparvaneh/alpha_mix_active_learning

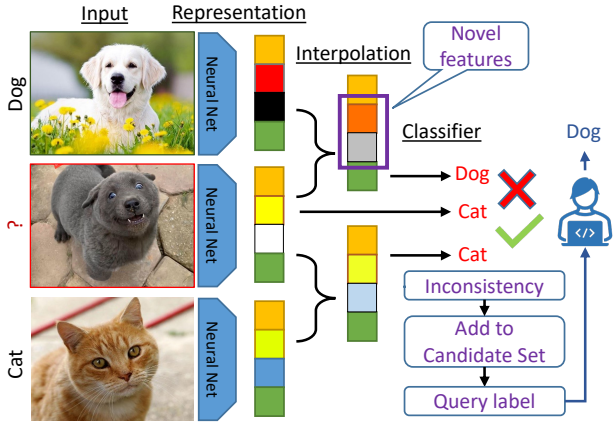
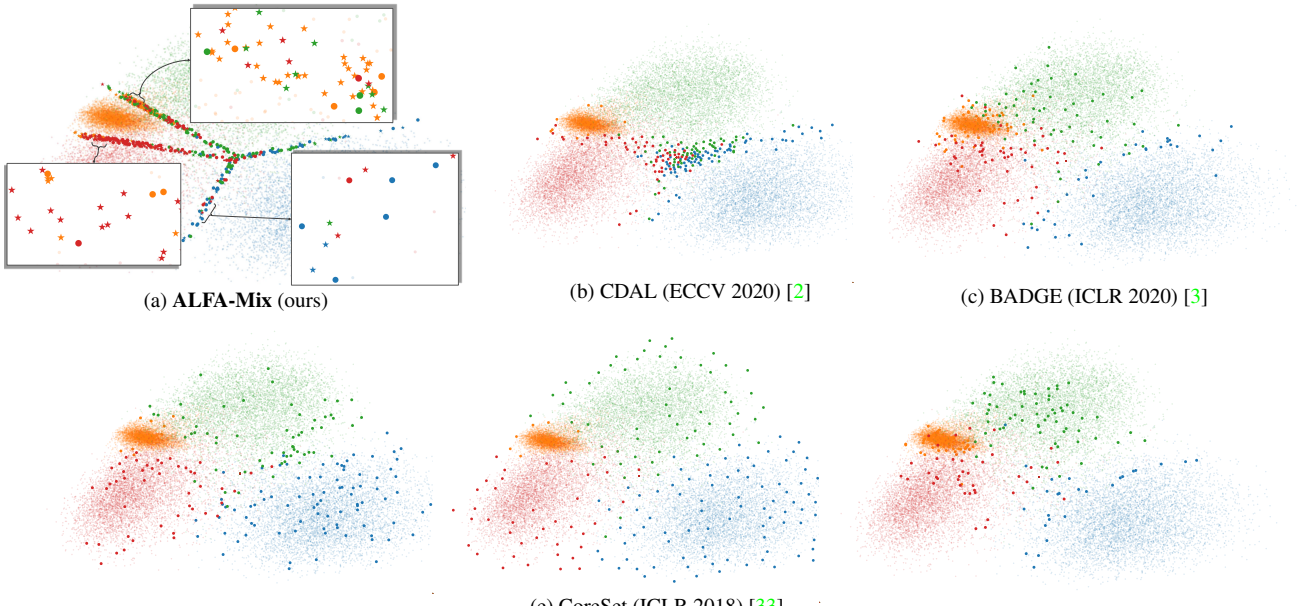


Figure 1. We propose to form linear combinations (*i.e.* interpolations or mixing) of the features of an unlabelled instance (middle image) and of labelled ones (top and bottom images). The interpolated features are passed through the current classifier. We show that inconsistencies in the predicted labels indicate that the unlabelled instance may have novel features to learn from.

Various AL strategies have been proposed differing in predicting (1) how informative a particular unlabelled instance will be (*i.e.* uncertainty estimation [12, 15, 31, 38]) or (2) how varied a set of instances will be (*i.e.* diversity estimation [33, 39]), or both [2, 17, 19]. Recent deep learning based AL techniques include, for example, the use of an auxiliary network to estimate the loss of unlabelled instances [40], the use of generative models like VAEs to capture distributional differences [20, 35], and the use of graph convolutional networks to relate unlabelled and labelled instances [5].

Despite much progress made, current AL methods still struggle when applied to deep neural networks, with high-dimensional data, and in a low-data regime. We hypothesised that the representations learned within deep neural networks may be leveraged to reason about the model's uncertainty while alleviating the challenges associated with high-dimensional data. Some existing methods only consider the model's output, but we believe that this cannot convey a complete picture of the model's current state. Assessing the uncertainty in the model is particularly important in a low-data regime since the number of available training



(a) **ALFA-Mix** (ours) (b) CDAL (ECCV 2020) [2] (c) BADGE (ICLR 2020) [3]

(d) GCNAL (CVPR 2021) [5] (e) CoreSet (ICLR 2018) [33] (f) BALD (ICML 2017) [15]

Figure 2. Visualization of sample selection behaviours of various AL methods in the latent space (see the Appendix for additional methods). The larger dots represent the selected samples to label; smaller dots represent unlabelled ones. Our approach finds a candidate set (demonstrated by stars in 2a) of unlabelled instances with inconsistencies in their label prediction when interpolated with labelled representations. It selects a diverse set of samples lying close to the all four borders for the labelling (with three zoom-in windows). The demonstration problem is that of identifying 4 classes from MNIST (illustrated above by 4 colours) using a MLP. An initial training set of 200 randomly selected points and their labels was provided, with each method given a budget of 200 additional labels. The features are projected to two-dimensions for visualization.

examples is small. This motivation has led to methods like BADGE [3] which uses gradients through the classifier layer of the network. Besides its relatively poor performance in low-data regimes [3], the drawback is a high computational cost due to the high dimensionality of the gradient embeddings, making the method impractical for deep models with latent representations of high dimensions, large datasets, and large numbers of classes.

In this paper, we present a novel and efficient AL method, named Active Learning by FeAture Mixing (ALFA-Mix), based on the manipulation of latent representations of the data. We identify informative unlabelled instances by evaluating the variability of the labels predicted for perturbed versions of these instances. These perturbed versions are instantiated in feature space as convex combinations of unlabelled and labelled instances (see Figure 1). This approach effectively explores the neighbourhood surrounding an unlabelled instance by interpolating its features with those of previously-labelled ones. Convex combinations of features have been already used in other contexts such as data augmentation, using random interpolations [36, 37, 41, 42] or actual solutions to an optimisation problem [1, 29].

We provide a theoretical support for the method. In particular, under a norm-constraint on the interpolation ratio, we show that the interpolation is equivalent to considering (1) the difference between the features of the unlabelled instance and the labelled ones and (2) the gradient of the model w.r.t the features at the unlabelled point. Discovering new features considering (1) and (2) leads us to finding an optimal interpolated point deterministically, at a minimal computing cost. Rather than using all the labelled data for these interpolations, we choose a subset we call anchors to capture the common features for each class. Subsequently, we construct a candidate set by choosing the instances from the unlabelled set that when mixed with these anchors lead to a change in the model's prediction for those instances. Then, to ensure selected instances are diverse, we perform a simple clustering in the candidate set and choose their centroids as the points to be queried.

The contributions of this paper are as follows.

- Instead of interrogating an unlabelled instance directly, we interpolate its representation features from the labelled instances to uncover its hidden traits. To the best of our knowledge, it is the first of its kind in AL. Unlike existing methods that reply solely on the predicted output, we harness useful information from the feature representations as an indication of which features are novel for the model.
- We show that optimal interpolation/mixing for each instance that underscores the novel features with which the model could change prediction, has a closed-form solution making our approach efficient and scalable.
- We show that our approach outperforms its counterparts

over 9 image, 2 OpenML, and one video datasets in various settings of architecture, network initialisation, and budget choice. Our approach consistently achieves higher accuracy than existing methods, with particularly significant gains in a low-data regime.

- We provide the first investigation into using AL in vision transformers: we demonstrate the effectiveness of ALFA-Mix on a self-trained vision transformer [6], performing better than random selection in all tests, and 43% better than the state-of-the-art. In addition, our approach performs significantly better that its counterparts for video classification using transformers [14].

2. Related Work

Active learning strategies can be broadly categorised into three types: diversity-based, uncertainty-based, and hybrid sampling, according to the nature of their acquisition function. Diversity-based approaches aim to select samples that best represent the whole of the available unlabelled set. A variety of approaches have been proposed that cluster the unlabelled samples based on feature representations [39], or construct a core-set over the latent features to identify a suitably diverse set of samples [33].

Uncertainty-based methods seek to identify the unlabelled samples that are most ambiguous to the current model that has been trained over the present labelled set based on the target objective function. The assumption here is that having these uncertain samples labelled will add the most value to the next model training round. Entropy and the confidence of the predictions [38], the margin between the confidence of the highest and second highest predicted classes [31], the information gain in the model parameters in a Bayesian framework [15], and the variance between the predicted probabilities within the ensemble [4] have all been proposed as measures of uncertainty. These methods favour points that lie close to the decision boundary, but as they rely entirely on the predicted class likelihoods they ignore the value of the feature representation itself. The closest method to that which we propose here is the deep fool attack learning (DFAL) approach [12] where the distance to the decision boundary is approximated by perturbation, using techniques originally developed for adversarial attacks [28]. Adversarial examples may expose vulnerability of the network architecture to particular patterns in the input rather than the distribution of the labels over latent space. That may lead to incorrect selection of instances that have patterns that are easily manipulated rather than helping to shape a more consistent decision boundary. Random perturbations are unlikely to lie within the true data distribution, and thus risk wasting labelling cost on feature values that can never arise in practice. Rather than repeatedly adding random noise in the input space, the method we propose here (ALFA-Mix) interpolates in latent space. ALFA-Mix is not only faster, it also significantly outperforms the DFAL approach.

Recently, a series of model-based active learning have been developed whereby a separate model is trained for active instance selection. Various objectives, either task-agnostic (*e.g.* variational adversarial active learning [35], graph convolutional active learning [5]) or task-aware (*e.g.* target loss prediction [40]), have been proposed as for training these models. Additionally, [8] has married model-based algorithms with conventional ones by combining a variational Bayes network with feature representations from the target model. In addition to sensitivity to hyper-parameters and additional computational cost, these AL methods do not consider the diversity of the selected samples and are prone to selecting samples with repetitive patterns. Moreover, our experiments show their poor performances in low-data regime.

Hybrid AL methods exploit both diversity and uncertainty in their sample selection methodologies. A mini-max strategy was proposed in [19], for example, that maximises both the informativeness and representativeness of the samples. Interestingly, a method that learns to combine different AL strategies was presented in [17]. Additionally, [2] exploits the predicted probabilities in images to select samples from diverse contexts (*i.e.* images of objects with varied backgrounds). Recently, [3] proposed to cluster the gradients of the final output layer of the target model as the features of the unlabelled samples that implicitly encompass the uncertainty information. Despite their state-of-the-art results on some image and non-image datasets, their approach is not scalable to larger tasks with numerous number of classes. Our approach not only consistently outperforms their method by a large margin in different settings, but it also is extremely efficient and scalable to large tasks.

3. Methodology

3.1. Problem Definition

Without loss of generality, we consider our learning objective to be training a supervised multiclass classification problem with K classes. A learner is actively trained in iterations of interactions with an oracle. At each iteration, this active learner has access to a small set of labelled data $\mathcal{D}^l = \{(\boldsymbol{x}_i, y_i)\}_{i=0}^{M}$ where $\boldsymbol{x}_i \in \mathcal{X}$ represents the input (*e.g.* an image or a video clip) and $y_i \in \{1, \dots, K\}$ stands for the associated class label. The learner also has access to a set of unlabelled data $\mathcal{D}^u$ from which B number of instances are chosen to be labelled by the oracle. The labelled samples are then added to $\mathcal{D}^l$ to update the model. The performance of the model is evaluated on an unseen test dataset.

The learner is a deep neural network $f = f_c \odot f_e$ parameterised by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_e, \boldsymbol{\theta}_c\}$. Here, $f_e : \mathcal{X} \to \mathbb{R}^D$ is the backbone which encodes the input to a D-dimensional representation in a latent space, *i.e.* $\boldsymbol{z} = f_e(\boldsymbol{x}; \boldsymbol{\theta}_e)$. Further, $f_c : \mathbb{R}^D \to \mathbb{R}^K$ is a classifier *e.g.* multi-layer perceptron (MLP)

that maps the instances from their representations to their corresponding logits which can be converted to class likelihoods by $p(y \mid \boldsymbol{z}; \boldsymbol{\theta}) = \text{softmax}(f_c(\boldsymbol{z}; \boldsymbol{\theta}_c))$. We optimise the parameters end-to-end by minimising the cross-entropy loss over the labelled set: $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}^l}[\ell(f_c \odot f_e(\boldsymbol{x}; \boldsymbol{\theta}), y)]$. The prediction of the label (*i.e.* pseudo-label) for an unseen instance is $y^*_{\boldsymbol{z}} = \arg\max_y f^y_c(\boldsymbol{z}; \boldsymbol{\theta}_c)$ where $\boldsymbol{z} = f_e(\boldsymbol{x}; \boldsymbol{\theta}_e)$ and f^y_c is the logit output for class y. Additionally, the logit of the predicted label is denoted as $f^*_c(\boldsymbol{z}) := f^{y^*_{\boldsymbol{z}}}_c(\boldsymbol{z})$[2]. We also denote $\boldsymbol{Z}^u = \{f_e(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{D}^u\}$ the set for representations of the unlabelled data and $\boldsymbol{Z}^l$ its labelled counterpart. We compute the average representation $\boldsymbol{z}^\star$ of the labelled samples per class, and call it anchor. The anchors for all classes form the anchor set $\boldsymbol{Z}^\star$, and serve as representatives of the labelled instances.

3.2. Feature Mixing

The characteristics of the latent space plays a crucial role in identifying the most valuable samples to be labelled. Our intuition is that the model's incorrect prediction is mainly due to novel "features" in the input that are not recognisable. Thus, we approach the AL problem by first probing the features learned by the model. To that end, we use a convex combination (*i.e.* interpolation) of the features as a way to explore novel features in the vicinity of each unlabelled point. Formally, we consider our interpolation between the representations of the unlabelled and labelled instances, $\boldsymbol{z}^u$ and $\boldsymbol{z}^\star$ respectively (we use the labelled anchor here for efficiency) as $\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} = \boldsymbol{\alpha}\boldsymbol{z}^\star + (1-\boldsymbol{\alpha})\boldsymbol{z}^u$ using an interpolation ratio $\boldsymbol{\alpha} \in [0, 1)^D$. This process can be seen as a way of sampling a new instance without explicitly modelling the joint probability of the labelled and unlabelled instances [1, 24, 29, 41], *i.e.*

$$\boldsymbol{z} \sim p(\boldsymbol{z} \mid \boldsymbol{z}^u, \boldsymbol{Z}^\star, \boldsymbol{\alpha}) \equiv \boldsymbol{\alpha}\boldsymbol{z}^\star + (1-\boldsymbol{\alpha})\boldsymbol{z}^u, \;\; \boldsymbol{z}^\star \sim \boldsymbol{Z}^\star. \quad (1)$$

We consider interpolating an unlabelled instance with all the anchors representing different classes to uncover the sufficiently distinct features by considering how the model's prediction changes. For that, we investigate the change in the pseudo-label (*i.e.* y^*) for the unlabelled instance and the loss incurred with the interpolation. We expect that a small enough interpolation with the labelled data should not have a consequential effect on the predicted label for each unlabelled point.

Using a first-order Taylor expansion w.r.t. $\boldsymbol{z}^u$, the model's loss for predicting the pseudo-label of an unlabelled instance at its interpolation with a labelled one can be re-written as[3]:

$$\ell\left(f_c\left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}\right), y^*\right) \approx \ell\left(f_c(\boldsymbol{z}^u), y^*\right) + (\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u))^\intercal . \nabla_{\boldsymbol{z}^u} \ell\left(f_c\left(\boldsymbol{z}^u\right), y^*\right), \quad (2)$$

which for a sufficiently small $\boldsymbol{\alpha}$, *e.g.* $\|\boldsymbol{\alpha}\| \leq \epsilon$ is almost exact. Consequently, for the full labelled set, by choosing the max loss from both sides we have:

$$\max_{\boldsymbol{z}^\star\sim\boldsymbol{Z}^\star} \left[\ell\left(f_c\left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}\right), y^*\right)\right] - \ell\left(f_c(\boldsymbol{z}^u), y^*\right) \approx \max_{\boldsymbol{z}^\star\sim\boldsymbol{Z}^\star} \left[(\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u))^\intercal . \nabla_{\boldsymbol{z}^u} \ell\left(f_c\left(\boldsymbol{z}^u\right), y^*\right)\right]. \quad (3)$$

Intuitively, when performing interpolation, the change in the loss is proportionate to two terms: (a) the difference of features of $\boldsymbol{z}^\star$ and $\boldsymbol{z}^u$ proportionate to their interpolation $\boldsymbol{\alpha}$, and (b) the gradient of the loss w.r.t the unlabelled instance. The former determines which features are novel and how their value could be different between the labelled and unlabelled instance. On the other hand, the later determines the sensitivity of the model to those features. That is, if the features of the labelled and unlabelled instances are completely different but the model is reasonably consistent, there is ultimately no change in the loss, and hence those features are not considered novel to the model.

The choice of $\boldsymbol{\alpha}$ is input specific and determines the features to be selected. As such, in Sec 3.3 we introduce a closed form solution for finding a suitable value for $\boldsymbol{\alpha}$. Finally, we note that the interpolations utilised here have some interesting properties that are further discussed in the supplements.

3.3. Optimising the Interpolation Parameter $\boldsymbol{\alpha}$

Since manually choosing a value for $\boldsymbol{\alpha}$ is non-trivial, we devise a simple optimisation approach to choose the appropriate value for a given unlabelled instance. To that end, we note that, as observed from Eq. (3), the worst case of maximum change in the loss is when we choose $\boldsymbol{\alpha}$ that maximises the loss at the interpolation point (details are in the supplement). However, using the r.h.s of the Eq. (3), we devise the objective for choosing $\boldsymbol{\alpha}$ as:

$$\boldsymbol{\alpha}^* = \arg\max_{\|\boldsymbol{\alpha}\|\leq\epsilon} (\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u))^\intercal . \nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*), \quad (4)$$

where ϵ is a hyper-parameter governing the magnitude of the mixing. Intuitively, this optimisation chooses the hardest case of $\boldsymbol{\alpha}$ for each unlabelled instance and anchor. We approximate the solution to this optimisation using dual norm formulation, which in the case of using 2-norm yields:

$$\boldsymbol{\alpha}^* \approx \epsilon \frac{\|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*)}{\|\nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*)\|_2} \oslash (\boldsymbol{z}^\star - \boldsymbol{z}^u), \quad (5)$$

where $\oslash$ represents element-wise division (further details in the Supplement). This approximation makes the optimisation of the interpolation parameter efficient and our experiments show that it will not have significant detrimental effects on the final results compared to directly optimising for $\boldsymbol{\alpha}$ to maximise the loss.

[2]For brevity, when the parameters $\boldsymbol{\theta}_e$ and $\boldsymbol{\theta}_c$ are clear from the context, we refrain from explicitly including them.

[3]This statement is true for any given instance and any convex combination of points in the latent space. For AL, we particularly focus on unlabelled instances though. More details are provided in the Supplements.

Algorithm 1: Our active learning algorithm.

Inputs: initial labelled set $\mathcal{D}^l$; unlabelled pool $\mathcal{D}^u$; labelling budget at each round B; mixing parameter ϵ;
for $i = 1$ ***to*** *max_rounds* **do**
  Train the model f using the labelled data $\mathcal{D}^l$.
  Initialise $\boldsymbol{Z}^\star$ based on the representations of $\mathcal{D}^l$.
  $\mathcal{I} = \{\}$.
  for $\boldsymbol{x}^u \in \mathcal{D}^u$ **do**
    $\boldsymbol{z}^u = f_e(\boldsymbol{x}^u)$.
    for $\boldsymbol{z}^\star \in \boldsymbol{Z}^\star$ **do**
      Calculate $\boldsymbol{\alpha}^*$ using ϵ and Eq. 5.
      $\tilde{\boldsymbol{z}} = \boldsymbol{\alpha}^* \boldsymbol{z}^\star + (1 - \boldsymbol{\alpha}^*)\boldsymbol{z}^u$.
      if $\arg\max_y(f_c^y(\boldsymbol{z}_u)) \neq \arg\max_y(f_c^y(\tilde{\boldsymbol{z}}))$ **then**
        $\mathcal{I} = \mathcal{I} \cup (\boldsymbol{x}^u, \boldsymbol{z}^u)$.
        Break
  Cluster the samples in $\mathcal{I}$ into B clusters.
  Select samples at the centre of each cluster ($\mathcal{C}$).
  $Y^{new} = \text{Query}(\mathcal{C})$.
  $\mathcal{D}^l = \mathcal{D}^l \cup (\mathcal{C}, Y^{new})$, $\mathcal{D}^u = \mathcal{D}^u \setminus \mathcal{C}$.

3.4. Candidate Selection

For AL it is reasonable to choose instances to be queried whose loss substantially change with interpolation according to Eq. (3). This corresponds to those instances for which the model's prediction change and have novel features. Intuitively, as depicted in Figure. 2a, these samples are placed close to the decision boundary in the latent space. Alternatively, we expect a small interpolation should not affect the model's loss when it is reasonably confident in its recognition of the features of the input. We, then, create our candidate set as:

$$\mathcal{I} = \left\{ \boldsymbol{z}^u \in \boldsymbol{Z}^u \;\middle|\; \exists \boldsymbol{z}^\star \in \boldsymbol{Z}^\star,\ f_c^*(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}) \neq y^*_{\boldsymbol{z}^u} \right\}. \tag{6}$$

The size of the selected set $\mathcal{I}$ could potentially be larger than the budget B. In addition, ideally we seek *diverse* samples since most instances in $\mathcal{I}$ could be chosen from the same region (*i.e.* they might share the same novel features). To that end, we propose to cluster the instances in $\mathcal{I}$ into B groups based on their feature similarities and further choose the closest samples to the centre of each cluster to be labelled by oracle. This ensures the density of the space represented by $\mathcal{I}$ samples, is reasonably approximated using B instances. We simply use k-MEANS which is widely accessible. Similar strategy is also used by [3] to encourage diversity. Our approach is summarised in Algorithm 1.

4. Experiments and Results

4.1. Baselines

We compare ALFA-Mix with the following AL baselines:

– **Random**: a simple baseline that randomly selects B samples from the unlabelled pool at each round.
– **Entropy** [38]: A conventional AL approach that picks unlabelled instances with highest entropy.
– **BALD** [15]: An uncertainty model relying on Bayesian approaches that selects a set of samples with the highest mutual information between label predictions and posterior of the model approximated using dropout (Figure 2f).
– **Coreset** [33]: An approach based on the core-set technique that chooses a batch of diverse representative samples of the whole unlabelled set (Figure. 2e).
– **Adversarial Deep Fool** [12]: An uncertainty method that utilises deep fool attacks to select a batch of unlabelled samples whose predictions flip with small perturbations.
– **GCNAL** [5]: A model-based approach that learns a graph convolutional network to measures the relation between labelled and unlabelled instances (Figure. 2d)[4].
– **BADGE** [3]: A hybrid approach that queries the centroids obtained from the clustering of the gradient embeddings (Figure. 2c).
– **CDAL** [2]: A hybrid approach that exploits the contextual information in the predicted probabilities to choose samples with varied contexts (Figure. 2b)

4.2. Experiment Settings

Setting and Datasets: We conducted a comprehensive set of experiments in 30 different settings on multiple datasets to evaluate how ALFA-Mix compares to its counterparts. We define an AL setting as a combination of a specific dataset, a limited set of initial labelled samples, a particular type of deep neural network, a limited number of AL rounds, and a fixed labelling budget (batch) for each round.

Specifically, we experimented on MNIST [23], EMNIST [9], CIFAR10 [21], CIFAR100 [21], Mini-ImageNet [32], DomianNet-Real [30] as well as two subsets of DomainNet-Real for image datasets. Additionally, we extended our experiments to two more non-visual datasets from the OpenML[5] repository. Furthermore, to reveal the effectiveness of each AL method in different data regimes, we utilised both small ($10 \times K$) and large ($100 \times K$) budget sizes. More importantly, the network architecture and its initial parameters are two more factors that we considered in our experiments. As for the choice of the architecture, we employed different common deep neural networks; including Multi-Layer Perceptron (MLP) [3], ResNet-18 [16], DenseNet-121 [18], as well as Vision Transformers [11]. Regarding the network initialisation, we considered three scenarios where at the start of each AL round[6], the parameters are initialised randomly, from the model trained in the previous round (denoted as "Continue" in Figure. 3), or using pre-trained models (either from supervised or self-supervised [6] pre-training on ImageNet [10]). Please find for more details in the Appendix.

[4] We employed CoreGCN variation in our experiments as results reported in [5] show its superiority over the UncertainGCN version.
[5] https://www.openml.org
[6] After a new batch of samples are selected by AL method and added to the labelled set and before the model training.

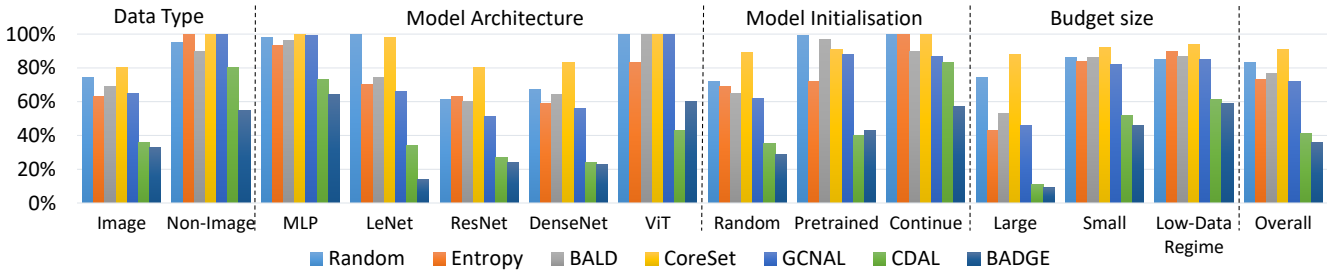


Figure 3. A summary of the performance of our proposed AL method (ALFA-Mix) compared with state-of-the-art across all the 30 settings considered. Each bar represents the percentage of AL rounds in which our approach outperforms others (lower indicates stronger baseline). It is worth noting that our approach (ALFA-Mix) under-performs others in close to zero cases.

We followed the supervised training setting proposed in [3] and optimised the network using all the labelled set (without any validation set) based on a cross-entropy loss and an Adam optimiser with a learning rate of $1e-3$ and $1e-4$ for image and non-image datasets, respectively. Similarly, we continued the training using a batch size of 64 until the model reaches a certain early stopping condition (*i.e.* reaching a training accuracy above 99% [3]).

We set the number of rounds for each setting to 10, except for DomainNet-Real where we continue for 5 rounds. Additionally, to eliminate the effect of randomness in the results, we repeated each experiment 5 times with different random seeds. To have a better understanding about the performance of each method, in addition to the quantitative results, we provided the penalty matrix [3] that facilitates the pairwise comparisons between different approaches across all the settings.

Video classification: Since video classification is a more challenging task with higher annotation cost, we compare the AL performance on video classification tasks. All the experiments are conducted on HMDB [22], a widely used dataset consisting of 5,412 training videos belonging to 51 classes representing different actions. For each video, we randomly sampled a video clip with 32 frames of size 224×224 using a temporal stride of 2. Regarding the network architecture, we employed the state-of-the-art Multi-Scale Vision Transformer (MViT) backbone pre-trained on Kinetics-600 [7]. Starting with a labelled set consisting of two labelled instances from each class (a total of 102 video clips), we provide each AL method with budget of the varied sizes ($2 \times K$, $4 \times K$, $7 \times K$ and $15 \times K$) in the next AL rounds. At each AL round, we train the network for 50 epochs with a batch size of 8 using AdamW [27] optimiser with a base learning rate of $1e-4$ that warms up linearly for the first 30 epochs and then decays to $5e-5$ for the rest of the iterations using a cosine scheduler [26]. We repeated each experiment twice to cancel out the effect of random selection of the initial labelled set.

4.3. Overall Results

Image and non-image results. In Figure. 4 we summarise all the results across various datasets, budget sizes and architectures (30 different settings in total) for image and non-image tasks into a matrix C. While each element $C_{i,j}$ in the matrix reveals in how many experiments the method shown in row i outperforms the one in column j in terms of accuracy of an unseen test set (higher is better for the approach shown in the row). The last row indicates the average number of experiments in which the method in the column has been outperformed by others (lower is better). The maximum value for each cell in the matrix is 30. This matrix clearly shows the superior performance of our approach compared to the baselines. In particular, we outperform CDAL [2] and BADGE [3] in a significant number of experiments (12.3 and 10.6 out of 30, respectively) but ours under-performed in only 0.3 of the times. Generally as shown in the last column, our approach is rarely outperformed (lower than 0.3). In other words, except in 3 AL rounds, for the rest of 282 ones (around 99% of the rounds), our approach is capable of matching or outperforming the best-performing baselines (BADGE and CDAL).

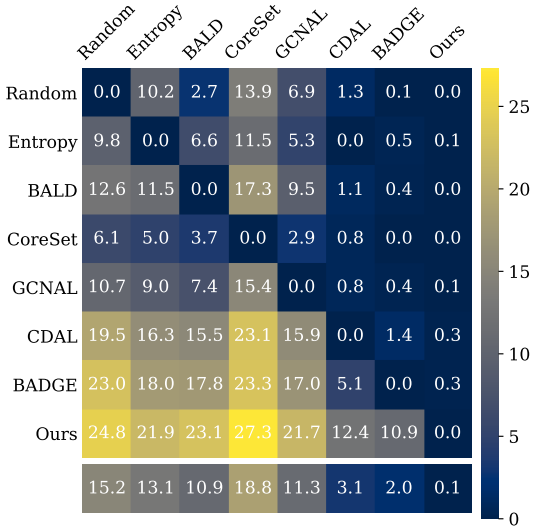
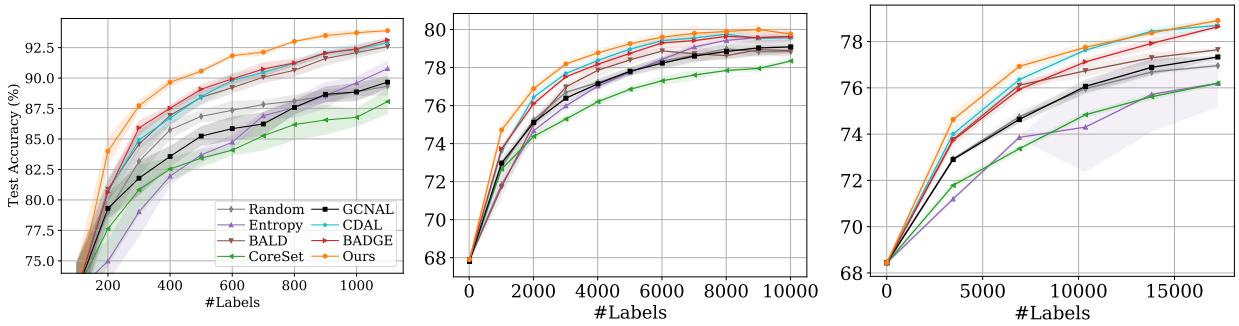


Figure 4. Pairwise comparison [3] of different approaches. Lower values shown at each column reveal the better performances of that AL method across all the experiments. Maximum value of each cell is 30, which represents the number of experimental settings.

Video Classification results. Table. 1 summarises the results for applying various AL methods for the activity recog-



(a) MNIST (MLP) (b) MiniImageNet (ViT-Small) (c) DomainNet-Real (ViT-Base)

Figure 5. Test accuracy plots across some of the employed settings. Each experiment has been repeated 5 times.

Method	AL Rounds 204*	408	765	1530
MViT (initial accuracy with 102 instances: 50.9±1.2)				
Random	56.7±1.4	64.1±1.2	72.0±1.1	75.3±0.4
Entropy [38]	55.5±0.6	65.5±0.3	70.2±2.0	76.5±0.7
BALD [15]	56.7±0.4	65.5±0.6	72.4±1.3	76.6±1.8
CoreSet [33]	59.3±1.3	65.8±1.2	72.8±1.6	78.5±0.7
GCNAL [5]	54.9±1.4	63.3±2.2	70.8±1.4	77.0±1.3
CDAL [2]	60.9±0.1	67.2±0.4	74.6±0.2	78.4±0.5
BADGE [3]	60.6±1.3	67.3±0.2	73.2±1.1	**78.7**±0.2
Ours	**62.5**±0.6	**69.4**±0.7	**75.1**±0.3	78.3±0.1

Table 1. Top-1 test accuracy of various AL methods on HMDB [22]. * Values on top of each column reveal the size of the labelled set at the end of each round.

nition in videos where our approach outperforms the baselines. Interestingly, compared to the Random sampling, we are able to improve the Top-1 test accuracy by more than 5% in the first two AL rounds and 3% in the last ones. This signifies the effectiveness of our proposed approach in reducing the labelling cost which is particularly an obstacle for video classification tasks. Moreover, ALFA-Mix outperforms all other strong baselines with a large margin (more than 2%) in the first three AL rounds. Interestingly, this is similar to what we observe from our experiments on other data types and show the effectiveness of our approach when applied to pre-trained transformers and/or in low-data regimes.

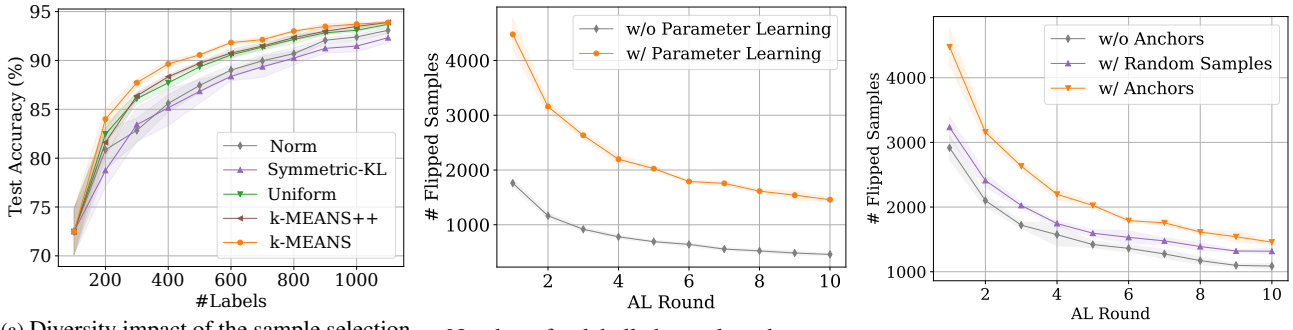
4.4. Ablation Study

Learning Ablations. Figure. 3 demonstrates the percentage of AL rounds where ALFA-Mix performs better than the baselines considering input data type, network architecture, network parameter initialisation and the budget size. The results indicate our approach, irrespective of other factors, consistently outperforms other AL baselines. Interestingly, when employing pre-trained networks, which is a common practice for transferring learnt representations to new tasks, ALFA-Mix 99% of occasions assists the model to generalise better than random sampling. Additionally, in these settings, our approach surpasses the strongest baselines (CDAL and BADGE) in more than 40% of the rounds. Above all, using Vision Transformer networks pre-trained in a self-supervised manner, ALFA-Mix not only outperforms Random, BALD, CoreSet and GCNAL in all the AL settings, it also significantly improves on BADGE and CDAL in 60% and 43% of the rounds respectively.

Interestingly, we observe a significant advantage from our proposed AL method when it is applied on small budget setting (Figure. 3). In fact, the test performance of our approach exceeds BADGE (the best performing baseline) in 46% of the small budget experiments. Moreover, we observe a more evident gap between our approach and others when it comes to AL in low-data regime. For that, we consider the performance in the first 5 rounds of AL using a small budget; *i.e.* starting from $10 \times K$ randomly selected labelled samples, each method queries for the maximum of $50 \times K$ unlabelled samples overall during 5 AL iterations. Figure. 3 demonstrates the dominance of our approach in this setting as it eclipses all other approaches in at least 60% of the experiments. When using a large budget, our approach is able to slightly surpass BADGE which previously has shown great success in this setting.

Diversification. Figure. 6a illustrates the effectiveness of the batch diversification on selecting final instances from the set of samples whose predictions have been changed ($\mathcal{I}$) during the interpolation process. In addition to *uniformly* sampling instances from the candidate set, we consider two heuristics: (1) the *norm* of the interpolation parameter $\|\boldsymbol{\alpha}\|_2$ in which a lower norm indicates with smaller intervention the model changed prediction; and, (2) the *symmetric KL-Divergence* between the predicted label distributions from the unlabelled instance $p(y|\boldsymbol{z}^u; \boldsymbol{\theta}_c)$ and that of the interpolated variant $p(y|\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}; \boldsymbol{\theta}_c)$. The latter evaluates the distributions change in the output (*i.e.* prefers samples with highest values of symmetric KL-Divergence). Interestingly, both heuristics show poor performances even in comparison with the uniform selection from the candidate set. While this highlights how hard the candidate selection could be, one explanation is that these approaches might use a considerable proportion of the budget on samples that reside in a small region of the space. Consequently, the selected batch does not carry the whole information obtained through the interpolation process.

In addition to the heuristic measures, we considered k-



(a) Diversity impact of the sample selection from the candidate set ($\mathcal{I}$). *k-MEANS is our proposed full model.

(b) Number of unlabelled samples whose predictions flip with and without learning the interpolation parameter $\boldsymbol{\alpha}$.

(c) The impact of anchors on identifying samples whose labels flip during the interpolation.

Figure 6. Ablations of our AL approach. Experiments are conducted on MNIST datasets using an MLP model and a small AL budget.

MEANS++, a simpler variation of k-MEANS that has shown better performance in [3], as another contender. In contrast to what found in [3], in our experiments, k-MEANS outperforms k-MEANS++ considerably as it better representations found using interpolation.

Learning the Interpolation Parameter. As it is evident in Figure. 6b, skipping the learning process for the interpolation parameter $\boldsymbol{\alpha}$ (see section 3.3) significantly reduces the number of samples chosen in the candidate set. This can have detrimental consequence on the diversity of samples that are selected during the clustering.

Anchors. Figure. 6c shows the impact of using different anchors $\boldsymbol{Z}^\star$. Evidently, the proposed method based on anchors outperforms other plausible alternatives including picking random samples from the labelled set and removing $\boldsymbol{z}^\star$ during the interpolation. The latter resembles adding noise to the sample and is similar to applying adversarial attack in the latent space.

Acquisition Time. We measured the time required to choose instances for labelling during each AL round. As demonstrated in Table 2, using a simple MLP network or a deep DenseNet-121, our approach performs competitive with the fastest baselines. This is mainly because of the fact that we only back-propagates to a latent representation layer (not the whole network). Additionally, our approaches reduces the time required for BADGE (the best performing baseline) by a factor of more than 2 when applied to datasets with a small number of classes. We should note that running BADGE on large-scale datasets with numerous classes requires a considerable time and computing resources. The main reason is the large dimensionality of the gradient embedding in tasks with large number of classes and instances. More importantly, Table 2 shows the time needed for DFAL method for MNIST dataset, which makes it impossible to apply to deep models and large datasets in a reasonable time.

Method	**Time** (seconds) **MNIST** (MLP)	**SVHN** (DenseNet)
Entropy [38]	$1_{\pm 0}$	$169_{\pm 44}$
BALD [15]	$16_{\pm 4}$	$1723_{\pm 445}$
Coreset [33]	$7_{\pm 2}$	$185_{\pm 49}$
DFAL [12]	$242_{\pm 69}$	–
GCNAL [5]	$12_{\pm 4}$	$187_{\pm 65}$
CDAL [2]	$5_{\pm 2}$	$179_{\pm 52}$
BADGE [3]	$50_{\pm 13}$	$523_{\pm 135}$
Ours	$5_{\pm 7}$	$210_{\pm 50}$

Table 2. Label acquisition run times of different methods. Our approach is significantly faster than BADGE and about 50x quicker than its Adversarial counterpart.

5. Conclusions and Limitations

In this paper, we proposed a simple AL method based on the interpolation between labelled and unlabelled samples. We effectively applied ALFA-Mix to a wide variety of image, non-image and video datasets and demonstrate its state-of-the-art results across various settings. Attractively, when the labelled set is small and the budget is limited, our approach is able to gain the most performance boost–it surpassed all other baselines in around 60% of all evaluated rounds.

Further, the feature representations are not generally disentangled [13, 25] and interpolation in the high dimensional space may yield representations for unexpected inputs. Nevertheless, our approach indicates such interpolations highlight reasonable variations in the input that may otherwise remain unexplored. For future, we consider using disentangled representations to explore novel factors of variations.

Limitations: AL consciously selects a small subset of a large pool of unlabelled samples to be labelled and used to train a model. AL will be essential in catastrophes, like pandemics, where the time to reach a model at a particular level of accuracy becomes vital and would directly impact the lives of people. In spite of that, its a common practice to evaluate AL in a simulated environment mainly due to financial constraints. However, AL community at large and our approach in particular could heavily benefit from real-world evaluations.

Acknowledgements

This material is based on research sponsored by Air Force Research Laboratory and DARPA under agreement number FA8750-19-2-0501. The U.S. Government is authorised to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

[1] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 4

[2] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *European Conference on Computer Vision*, pages 137–153. Springer, 2020. 1, 2, 3, 5, 6, 7, 8

[3] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020. 2, 3, 5, 6, 7, 8

[4] William H. Beluch, Tim Genewein, Andreas Nurnberger, and Jan M. Kohler. The power of ensembles for active learning in image classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018. 3

[5] Razvan Caramalau, Binod Bhattarai, and Tae-Kyun Kim. Sequential graph convolutional network for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9583–9592, June 2021. 1, 2, 3, 5, 7, 8

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 3, 5

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6

[8] Jongwon Choi, Kwang Moo Yi, Jihoon Kim, Jinho Choo, Byoungjip Kim, Jinyeop Chang, Youngjune Gwon, and Hyung Jin Chang. Vab-al: Incorporating class imbalance and difficulty with variational bayes for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6749–6758, June 2021. 3

[9] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, 2017. 5

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 5

[11] 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. In *International Conference on Learning Representations*, 2021. 5

[12] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. In *arXiv:1802.09841*, 2018. 1, 3, 5, 8

[13] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*, 2020. 8

[14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *CoRR*, abs/2104.11227, 2021. 3

[15] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR, 06–11 Aug 2017. 1, 2, 3, 5, 7, 8

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[17] Wei-Ning Hsu and Hsuan-Tien Lin. Active learning by learning. In *AAAI Conference on Artificial Intelligence*, 2015. 1, 3

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 5

[19] Sheng-jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 892–900. Curran Associates, Inc., 2010. 1, 3

[20] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8166–8175, June 2021. 1

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 6, 7

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[24] Damian Lesniak, Igor Sieradzki, and Igor T. Podolak. Distribution-interpolation trade off in generative models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 4

[25] Francesco Locatello, Ben Poole, Gunnar Raetsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine*

Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 6348–6359. PMLR, 13–18 Jul 2020. 8

[26] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 6

[27] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam, 2018. 6

[28] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3

[29] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Javen Shi, and Anton van den Hengel. Counterfactual vision-and-language navigation: Unravelling the unseen. In *Advances in Neural Information Processing Systems*, volume 33, 2020. 2, 4

[30] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 5

[31] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 413–424, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 1, 3

[32] Hugo Larochelle Sachin Ravi. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017. 5

[33] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. 1, 2, 3, 5, 7, 8

[34] Burr Settles. Active learning literature survey. 2009. 1

[35] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 3

[36] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *International Conference on Machine Learning (ICML)*, 2019. 2

[37] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019. 2

[38] D. Wang and Y. Shang. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 112–119, 2014. 1, 3, 5, 7, 8

[39] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G. Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. In *International Journal of Computer Vision*, 2015. 1, 3

[40] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 3

[41] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2, 4

[42] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations (ICLR)*, 2021. 2