Robust Deep Active Learning via Distance-Measured Data Mixing and Adversarial Training

Shinan Song , Xing Wang , Shike Dong , Jingyan Jiang *

Posted Date: 7 October 2025

doi: 10.20944/preprints202510.0404.v1

Keywords: Active Learning; Data Selection; Robustness; Uncertainty Estimation

 doi:10.20944/preprints202510.0404.v1

Article

Robust Deep Active Learning via Distance-Measured Data Mixing and Adversarial Training

Shinan Song [1], Xing Wang [1], Shike Dong [2] and Jingyan Jiang [3,*]

[1] School of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China;
[2] Sendelta International Academy Shenzhen, Shenzhen 518100, China;
[3] College of Bigdata and Internet, Shenzhen Technology University, Shenzhen 518118, China.
* Correspondence: Jingyan Jiang(jiangjingyan@sztu.edu.cn)

Abstract

Accurate uncertainty estimation in unlabeled data represents a fundamental challenge in active learning. Traditional deep active learning approaches suffer from a critical limitation: uncertainty-based selection strategies tend to concentrate excessively around noisy decision boundaries, while diversity-based methods may miss samples that are crucial for decision-making. This over-reliance on confidence metrics when employing deep neural networks as backbone architectures often results in suboptimal data selection. We introduce Distance-Measured Data Mixing (DM2), a novel framework that estimates sample uncertainty through distance-weighted data mixing to capture inter-sample relationships and the underlying data manifold structure. This approach enables informative sample selection across the entire data distribution while maintaining focus on near-boundary regions without overfitting to the most ambiguous instances. To address noise and instability issues inherent in boundary regions, we propose a boundary-aware feature fusion mechanism integrated with fast-gradient adversarial training. This technique generates adversarial counterparts of selected near-boundary samples and trains them jointly with the original instances, thereby enhancing model robustness and generalization capabilities under complex or imbalanced data conditions. Comprehensive experiments across diverse tasks, model architectures, and data modalities demonstrate that our approach consistently surpasses strong uncertainty-based and diversity-based baselines while significantly reducing the number of labeled samples required for effective learning.

Keywords: active learning; data selection; robustness; uncertainty estimation

1. Introduction

Deep neural networks typically require extensive labeled datasets for effective training, making data annotation a slow, expensive, and complex process [1]. Active learning addresses this challenge by strategically selecting the most informative samples from an unlabeled pool for annotation, thereby reducing the overall labeling burden [2]. A prevalent approach prioritizes samples with low-confidence predictions, as these high-uncertainty instances are empirically proven to provide valuable information for model improvement [3]. However, strategies that rely exclusively on uncertainty estimation often concentrate sample selection within narrow regions of the feature space, resulting in inadequate coverage of the overall data distribution and potential amplification of label noise [4]. Incorporating diversity considerations into the selection process can mitigate these issues by ensuring broader distributional coverage and capturing richer information content across the data manifold.

Recent active learning research has therefore pursued two complementary objectives: uncertainty estimation and diversity promotion. Uncertainty-based methods such as Least Confidence [3,5,6] prioritize samples exhibiting minimal predictive confidence. Deep Bayesian techniques further enhance uncertainty estimation by utilizing posterior predictive distributions to refine entropy-based and mutual-information-based selection criteria [7–11]. Alternative approaches employ auxiliary models

 doi:10.20944/preprints202510.0404.v1

to improve uncertainty estimation or guide the selection process [12–17]. Complementing these uncertainty-focused strategies, diversity-oriented methods such as VAAL [18] explore varied regions within the latent space, while BADGE [19] integrates gradient-based uncertainty estimation with clustering techniques to enhance distributional coverage [20]. Despite these advances, a fundamental trade-off remains: uncertainty-centric querying tends to over-sample points near decision boundaries that are often noisy or ambiguous, whereas diversity-only selection may overlook the most decision-relevant instances, particularly in complex, imbalanced, or noisy data environments.

From a decision-boundary perspective, samples with high uncertainty typically reside near class separators, where noise and label ambiguity are most prevalent. Excessive emphasis on such points can propagate labeling errors and compromise training quality. To investigate this phenomenon, we conducted a preliminary study using Least Confidence on CIFAR-10 [21] with MobileNet [22] as the backbone architecture. Following an initial training phase, we selected 1,000 samples per iteration based on lowest confidence scores and visualized the selections using t-SNE, comparing direct top-k selection (offset 0) with an offset strategy that skips the top 300 lowest-confidence samples. As illustrated in Figure 1, the offset strategy distributes selections more broadly across the feature space, enhancing both coverage and diversity. This approach achieved 84.11% accuracy compared to 83.5% without offset, demonstrating that strategically shifting away from the most uncertain samples can capture richer information content and improve overall performance.

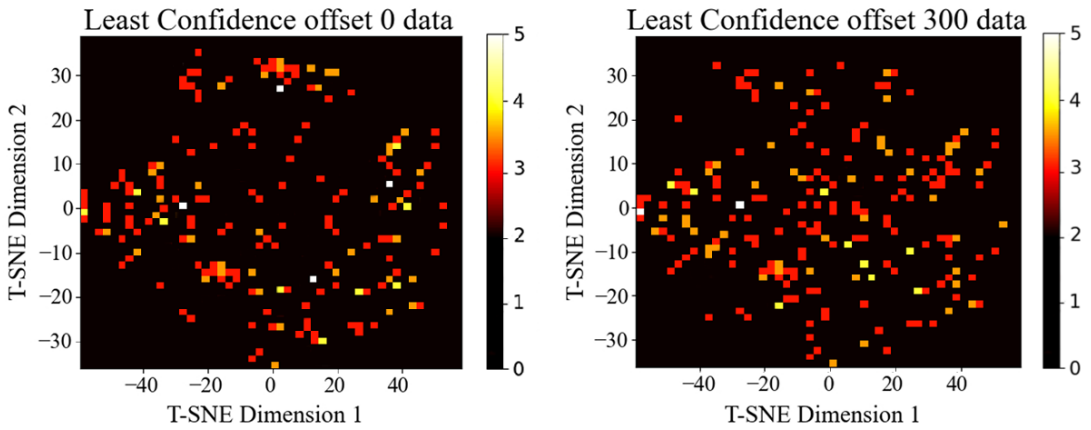


Figure 1. The left side of the figure illustrates a heat map of the data distribution for the Least Confidence method, achieving 83.5% accuracy. The right side depicts the data distribution when offsetting 300 data points, resulting in 84.11% accuracy. This comparison highlights the impact of offsetting on sample diversity and model performance.

Motivated by these observations and drawing inspiration from Alpha-Mix [1], which employs tuned mixing strategies to introduce variability while preserving salient features, we propose a data-mixing framework for deep active learning that simultaneously addresses uncertainty estimation, diversity promotion, and robustness enhancement. Our central premise involves inferring sample uncertainty through weighted mixtures that encode inter-sample relationships and underlying data distribution structure, with particular emphasis on near-boundary regions that are critical for class discrimination yet susceptible to noise contamination. Building upon this foundation, we further introduce a boundary-aware feature fusion mechanism via adversarial training: we generate adversarial counterparts for selected near-boundary samples using fast gradient methods and train them jointly with the original instances. This approach enhances generalization capabilities and robustness in complex, noisy environments by stabilizing the learning process around decision boundaries.

Our contributions are summarized as follows:

- We introduce Distance-Measured Data Mixing (DM2) Active Learning, a novel deep active learning framework that estimates sample uncertainty through distance-weighted mixing of data samples. By exploiting inter-sample relationships and distributional structure, this method

 doi:10.20944/preprints202510.0404.v1

selects informative instances across the data manifold, including near-boundary regions, thereby enhancing the diversity of queried samples.
- To address noise susceptibility in challenging scenarios, we augment Distance-Measured Data Mixing with adversarial training (DM2-AT). We generate fast-gradient adversarial samples for selected near-boundary instances and train them jointly with the original data, improving model robustness and generalization performance under complex data distributions.
- Comprehensive experiments across diverse tasks, model architectures, and data modalities demonstrate that our method achieves superior performance while significantly reducing labeling requirements compared to existing approaches.

2. Related Work

2.1. Uncertainty Estimation

The uncertainty-based approach selects the most ambiguous unlabeled samples from the current model. Since the model is initially trained with a limited dataset, these ambiguous samples provide valuable information for subsequent rounds of training: (1) Prominent uncertainty selection, which includes the Least Confidence method [5] and Entropy Sampling [23]. The Margin Sampling method [6] evaluates the difference between the confidence levels of the highest and second-highest prediction classes. BatchBALD [24] selects samples by maximizing the joint information gain of a batch. (2) Bayesian framework, information gain methods focus on model parameters [4,25], often integrating Bayesian belief networks with Monte Carlo sampling [26]. Deep Bayesian approximation methods like MC-Dropout [7] are employed to address the challenge of probabilistic prediction. Query by Committee (QBC) methods facilitate multi-model training [27], while adversarial training methods [10,11,18] provide additional robustness. Furthermore, the variance between predicted probabilities within a set [28] has been proposed as a measure of uncertainty. (3) Model-based active learning trains a separate model for active instance selection. Variational Autoencoders (VAE) [18] utilize a V-shaped autoencoder to model data distribution. CoreGCN [13] employs Graph Convolutional Networks (GCNs) to represent relationships between examples. LL4AL [14] integrates a lightweight module to learn the prediction error of unlabeled examples, capturing the learning loss in active learning. ProbCover [29] is a novel active learning algorithm designed for low-budget scenarios, aiming to maximize probability coverage. Methods like ISAL and ent-gn [15,16] propose using influence functions [17] to estimate potential model changes, thereby informing training strategies. These techniques often prioritize points near the decision boundary. However, they may overlook valuable data near the decision boundary by relying solely on predicted class likelihood.

2.2. Diversity Improvement

In active learning, diversity refers to selecting representative and varied samples for labeling. Methods often assign confidence scores based on classifier uncertainty and sample diversity [30]. One strategy uses entropy and mutual information within a CRF graphical model for query selection [31]. The state-of-the-art method, Coreset [12], focuses on selecting data with diverse representations. BADGE [19] explores the relationship between data diversity and uncertainty using Bayesian methods and clustering. CoreGCN [13] employs graph embeddings for diverse data selection. Hierarchical agglomerative clustering (HAC) [32] distributes uncertain examples. BatchBALD [24] also considers sample diversity by selecting complementary samples to avoid redundancy. Recent advances utilize parameters from the final neural layer, while Alpha-Mix [1] employs feature mixing with alpha values. Noise Stability [20] introduces a greedy algorithm that adds noise to highlight differences. These methods propose complex paradigms for modeling diversity, enhancing effectiveness but increasing complexity and coupling.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 October 2025 doi:10.20944/preprints202510.0404.v1

3. Distance-Measured Data Mixing

This section presents our theoretical framework for Distance-Measured Data Mixing (DM2), as illustrated in Figure 2. Our approach focuses on selecting samples that effectively balance diversity and uncertainty considerations. The process begins by feeding unlabeled data through the model to extract feature-layer embeddings. We then compute similarity distances between samples within these embedding spaces and select multiple similar instances for linear mixing according to predetermined proportions. The model evaluates these mixed samples to determine their confidence levels. Finally, we rank samples by confidence scores and select data indices with the lowest confidence values, adding the corresponding original samples to the labeled candidate set to complete each selection round.

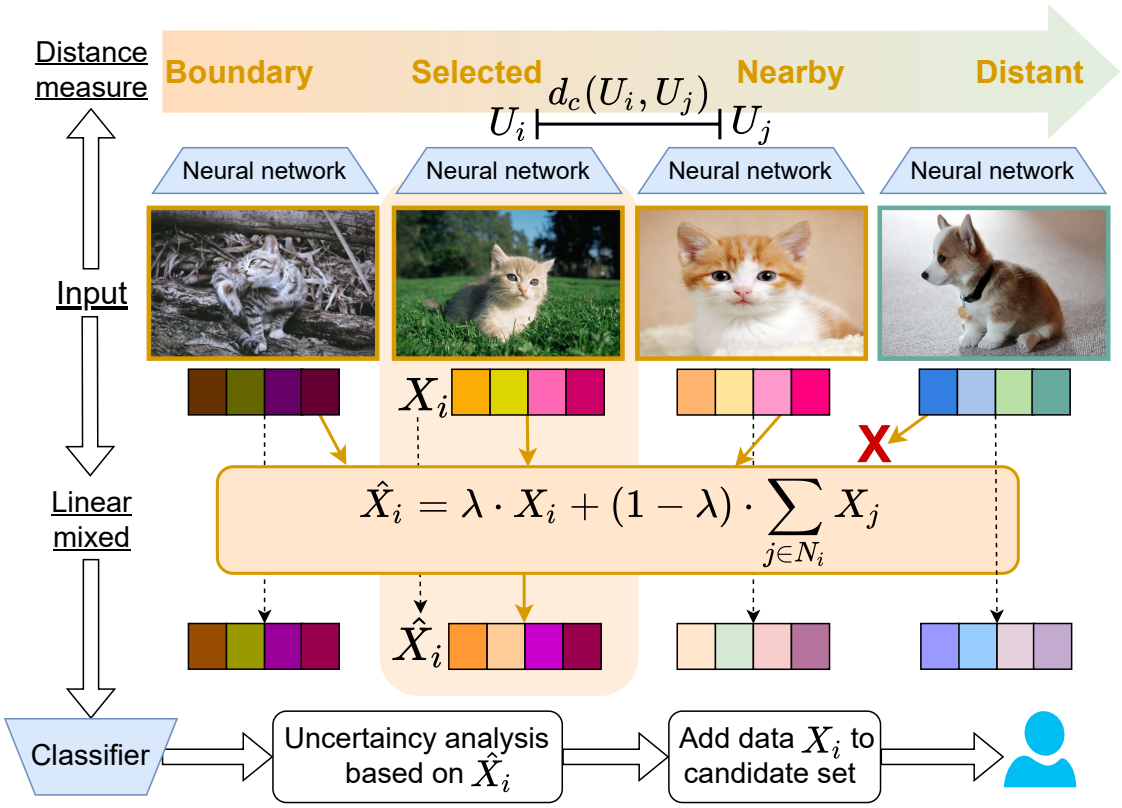


Figure 2. Overview of DM2. Data is input into the model to obtain feature layer embeddings. We calculate sample similarity distances on these embeddings and select similar samples for linear blending. The model is tested with mixed samples to determine confidence levels. Data with low confidence is identified, and the original data is added to the labeled data pool.

3.1. Formal Definition

Given an unlabeled data pool U and an initially empty labeled data pool L, the objective of active learning is to select a subset of samples X_i from U based on a predefined annotation budget (e.g., selecting 1,000 samples at a time from a pool of 50,000 samples). These selected samples are annotated by human experts and added to L, such that $L \leftarrow L \cup X_i, Y_i$, where Y_i represent the newly acquired annotations. The selected samples are subsequently removed from the unlabeled pool: $U \leftarrow U \setminus X_i$.

A neural network model is defined as a function f_θ, where θ denotes the model parameters. For input data $U = X_1, X_2, \ldots, X_n$, the model generates predictions $\hat{Y}i = f\theta(X_i)$. The model is trained by minimizing the cross-entropy loss function:

$$\ell(Y, \hat{Y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}), \tag{1}$$

 doi:10.20944/preprints202510.0404.v1

where N represents the number of samples, C denotes the number of classes, and $\hat{y}_{i,c}$ is the predicted probability that the i-th sample belongs to class c. The optimization objective is to minimize this loss function:

$$\theta^* = \arg\min_{\theta} \ell(Y, f_\theta(x)). \tag{2}$$

In subsequent data selection phases, the trained model evaluates samples from the unlabeled pool U to obtain confidence scores for each instance. Based on these confidence estimates, the method selects samples to be added to the labeled pool L, initiating the next cycle of training and data selection.

3.2. Feature Extraction

The feature representation from the final layer of a convolutional neural network (CNN) captures the highest-level, most abstract features from input images. These features effectively encapsulate global information and complex patterns, making them particularly valuable for tasks such as classification and recognition.

For each sample in the unlabeled data pool $U = X_1, X_2, \ldots, X_n$, we extract features from the last convolutional layer of the CNN. The output features are denoted as $F_{\text{Conv}}(X_i) \in \mathbb{R}^d$, where d represents the feature dimension. Here, U_i denotes the feature representation of the i-th sample extracted through this layer. The feature extraction process is formalized as:

$$U_i = F_{\text{Conv}}(X_i). \tag{3}$$

3.3. Distance Measured

Euclidean distance effectively captures geometric relationships in continuous feature spaces, making it well-suited for detecting subtle differences between samples and demonstrating high sensitivity to small variations in data. In contrast, Manhattan distance is particularly effective for measuring differences in discrete or sparse feature spaces.

By combining Euclidean and Manhattan distances, we can more comprehensively assess the relationships between samples. Let U_i and U_j represent the feature vectors of samples i and j, respectively. The Euclidean distance between two sample vectors is defined as:

$$d_{\text{E}}(U_i, U_j) = \sqrt{\sum_{k=1}^{n} (U_{ik} - U_{jk})^2}, \tag{4}$$

where U_{ik} and U_{jk} represent the k-th components of feature vectors U_i and U_j, respectively, and d is the feature dimension. The Manhattan distance metric is expressed as:

$$d_{\text{M}}(U_i, U_j) = \sum_{k=1}^{n} \left| U_{ik} - U_{jk} \right|. \tag{5}$$

The combined distance metric, obtained by averaging Equations 4 and 5, is expressed as:

$$d_c(U_i, U_j) = \frac{d_{\text{E}}(U_i, U_j) + d_{\text{M}}(U_i, U_j)}{2}. \tag{6}$$

3.4. Linear Data Mixing

The distance function $d_c(U_i, U_j)$ calculates the similarity between feature samples U_i and U_j. In this step, we select n samples $X_1, X_2, \ldots, X_n$ that are most similar to sample X_i for linear mixing. We denote the set of nearest neighbors of U_i based on $d_c(U_i, U_j)$ as N_i, and employ the parameter λ to control the degree of mixing. This process operates directly at the data level rather than on feature representations. The mixing formula is expressed as:

 doi:10.20944/preprints202510.0404.v1

$$\hat{X}_i = \lambda \cdot X_i + (1 - \lambda) \cdot \sum_{j \in N_i} X_j, \tag{7}$$

where $\hat{X}_i$ represents the mixed data sample derived from X_i, N_i denotes the index set of the n nearest neighbors of sample X_i, and λ controls the mixing weight of the original sample X_i. The linearly mixed sample $\hat{X}_i$ is then fed into the pre-trained model for prediction:

$$P_i = f_\theta(\hat{X}_i). \tag{8}$$

The output of the classification model, P_i, is typically a vector representing the predicted logits for each class. This output is converted into a probability distribution using the softmax function:

$$\text{Softmax}(P_i) = \left[\frac{\exp(P_{i1})}{\sum_c \exp(P_{ic})}, \ldots, \frac{\exp(P_{iC})}{\sum_c \exp(P_{ic})} \right], \tag{9}$$

where C represents the number of classes and P_{ic} denotes the model's logit score for sample $\hat{X}_i$ belonging to class c.

The probability distribution from the classification model's output is utilized to determine the confidence level for each sample. The highest predicted probability typically serves as a confidence indicator:

$$C_i = \max(\text{softmax}(P_i)). \tag{10}$$

This represents the model's maximum predicted probability for sample $\hat{X}_i$, indicating its confidence level. Samples are ranked by their confidence scores, and those with the lowest confidence are selected for the next active learning batch. When confidence levels are sorted in ascending order, sort$[0]$ corresponds to the index of the sample with the lowest confidence, while sort$[N-1]$ represents the highest confidence sample. From this sorted arrangement, we select the n samples with the lowest confidence scores for annotation. The indices of these selected samples constitute the active learning dataset D_a:

$$D_a = \{X_{\text{sort}[0]}, X_{\text{sort}[1]}, \ldots, X_{\text{sort}[N-1]}\}. \tag{11}$$

We return the selected n sample indices for active learning annotation. The corresponding samples are retrieved from the original unlabeled dataset $U = X_1, X_2, \ldots, X_m$ using these indices and added to the labeled pool L. This process is repeated iteratively throughout the entire active learning cycle.

 doi:10.20944/preprints202510.0404.v1

Algorithm 1 Distance-Measured Data Mixing Active Learning

Input: f: randomly initialized neural network, U: unlabeled data pool,
    L: labeled data pool
Output: L: updated labeled pool
Begin:
    Train the model f using the unlabeled pool U to obtain $F_{\text{conv}}(X_i)$
    for $k \leftarrow 1$ to K **do**
        for $i \leftarrow 1$ to S **do**
            $\delta(U_k, U_i) = \sqrt{(F_{\text{conv}}(U_k) - F_{\text{conv}}(U_i))^2} + |F_{\text{conv}}(U_k) - F_{\text{conv}}(U_i)|$
            get $N_k = \delta(U_k, U_i)/2$
        end for
    end for
    Number of mixed samples is N
    for $h \leftarrow 1$ to H **do**
        $\hat{X}_h = \lambda * X_h + (1 - \lambda) * \sum_{j \in N_h} X_j$
        $\xi(h) = \hat{X}_h$
    end for
    Train the model $f(\xi(h))$ to obtain $M = \text{sort}(\text{softmax}(P))$
    Select samples with indexes $\{M_{jc}\}_{c=1}^{N}$ in U as $\{X_i\}$, and obtain their labels $\{Y_i\}$
    Update L with $L = L \cup \{X_i, Y_i\}$
Return L

4. Adversarial Training for Boundary Data Feature Fusion

This section presents an active learning algorithm that integrates adversarial training with feature fusion for boundary data samples. This method employs adversarial training to enhance model efficiency, thereby strengthening performance when processing noisy and complex environmental data. Simultaneously, the active learning strategy reduces the required number of training samples, lowering overall training costs. Specifically, in our boundary data feature fusion approach for active learning, samples selected in each round are initially augmented through adversarial training to generate adversarial counterparts. These adversarial samples are subsequently merged with the existing labeled pool, enabling the model to fully exploit the augmented data during updates.

The advantage of this approach lies in its incorporation of active learning properties to reduce labeled data requirements while leveraging adversarial training to enhance the model's classification capabilities, particularly when confronting noise and interference in real-world scenarios. Through this combination, the model's information recognition performance is significantly improved, achieving more accurate classification in complex environments and demonstrating adaptability across diverse application scenarios. This algorithm not only improves model stability and classification accuracy but also reduces training sample requirements while adapting to large-scale dataset challenges, offering substantial practical value, especially in applications requiring rapid deployment and efficient training. The complete methodological process is illustrated in Figure 3.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 October 2025 doi:10.20944/preprints202510.0404.v1

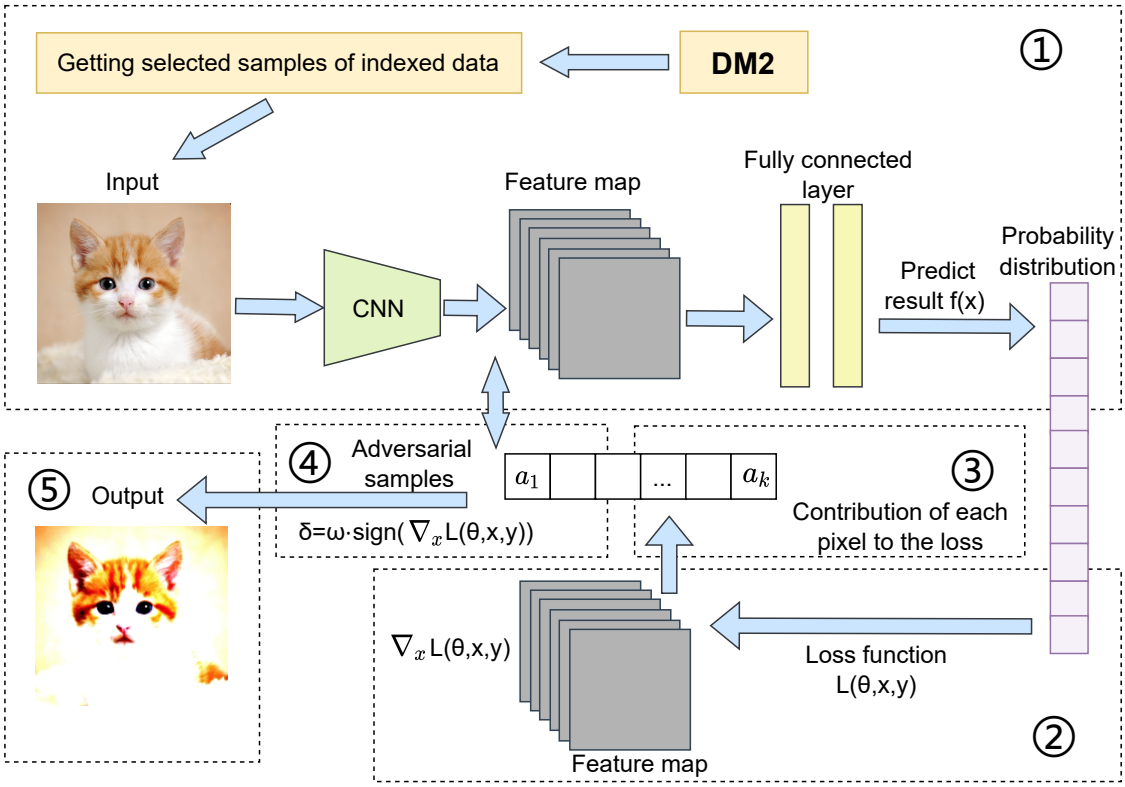


Figure 3. Overview of adversarial training for sample selection, where the input consists of near-boundary data selected by DM2 from Section 3. After generating adversarial samples, the original samples are added to the labeling pool alongside their adversarial counterparts. Through iterative selection of both original and adversarial sample sets, the model's robustness is enhanced.

4.1. FGSM Confrontation Training

Adversarial training serves as a method to improve model generalization by incorporating adversarial samples into the training dataset during the training process. This approach compels the model to learn from these challenging examples, thereby enhancing its ability to defend against adversarial perturbations. The Fast Gradient Sign Method (FGSM) represents one of the most widely used techniques for generating adversarial samples, and FGSM adversarial training constitutes a training methodology that employs FGSM to generate adversarial samples and incorporate them into the training set [33].

FGSM is an algorithm that efficiently generates adversarial perturbations by computing the gradient of the loss function with respect to the input. The fundamental principle involves applying a small perturbation along the direction of the loss function's gradient to input samples, thereby causing the model to produce erroneous predictions. This perturbation is computed individually for each input sample, making it inherently "sample-specific" [34]. The FGSM generation process follows these specific steps:

Calculate the gradient: For each input sample and its corresponding label, we first compute the gradient of the loss function with respect to the input:

$$\nabla_x \mathcal{L}(\theta, x, y), \tag{12}$$

where $\mathcal{L}(\theta, x, y)$ represents the loss function with model parameters θ, input sample x, and true label y. This gradient indicates the direction in which small changes to the input would most significantly increase the loss.

Using the computed gradient to generate adversarial perturbations, the key principle of FGSM involves computing the sign of the gradient (representing the gradient's direction) and adding perturbations along that direction. The perturbation magnitude is controlled by a small constant parameter:

 doi:10.20944/preprints202510.0404.v1

$$\text{adv}_{sample} = x + \omega^* sign(\nabla_x \mathcal{L}(\theta, x, y)), \tag{13}$$

where sign$(\cdot)$ is the sign function that extracts the sign of each element in the gradient vector, and ϵ is the hyperparameter that controls the perturbation magnitude. This formula represents the process of applying a small perturbation to input samples along the direction of the loss function's gradient. The sign function ensures that the perturbation moves in the direction that would maximally increase the loss, while the ϵ parameter bounds the perturbation size to maintain the adversarial sample's similarity to the original input.

The generated adversarial samples are incorporated alongside the original samples during the training process, enabling the model to learn correct predictions when confronted with adversarially perturbed inputs. This approach enhances the model's robustness by exposing it to challenging examples that lie near the decision boundary.

In adversarial training, the training process comprises two complementary components. First, positive sample training follows the traditional approach by utilizing original data for model training. Second, adversarial sample training incorporates adversarial samples generated using FGSM into the training data. During each training step, a batch of data is selected from the training set, where each sample consists of an input x and its corresponding label y. FGSM is then applied to generate adversarial perturbations for each sample, producing the corresponding adversarial examples.

The training procedure calculates losses for both the original samples and their adversarial counterparts, combining these losses for backpropagation to update model parameters:

$$\mathcal{L}_{\text{total}} = \frac{1}{2}(\mathcal{L}(\theta, x, y)) + \mathcal{L}(\theta, adv_{sample}, y)), \tag{14}$$

where $\mathcal{L}(\theta, x, y)$ represents the loss computed on the original sample and $\mathcal{L}(\theta, adv_{sample}, y)$ denotes the loss computed on the adversarial sample. This combined loss function ensures that the model simultaneously learns to make correct predictions on both normal and adversarially perturbed inputs.

FGSM adversarial training constitutes an effective method for improving model robustness. By generating adversarial samples and incorporating them into the training data, this approach enhances the model's ability to adapt to input perturbations, enabling the model to maintain performance when confronted with adversarial examples during inference.

4.2. Adversarial Training for Sample Selection

This method combines the principles of active learning and adversarial training to enhance model stability and performance. The specific process unfolds as follows:

Initially, the trained model performs forward propagation to obtain feature representations from the final layer for each sample, as expressed in Equation 3. This process effectively represents the information contained within samples as high-dimensional feature vectors. Subsequently, the most representative samples are selected through similarity calculations between samples. We employ a combination of Manhattan and Euclidean distances to produce a more reliable and efficient distance measure, as demonstrated in Equation 6. This combination leverages the strengths of both distance metrics to achieve more stable similarity calculations, particularly when handling features with varying scales.

Through inter-sample similarity calculations, we identify the most similar batch of samples for subsequent fusion and training. The n most similar samples are then merged using the MixUp fusion method, where samples are linearly combined to generate new training instances that enhance model generalization capability, as shown in Equation 7. Classification confidence is obtained by evaluating the model, and the corresponding index list is returned for active learning selection.

Following sample selection by the boundary data feature fusion algorithm, the original images undergo forward propagation through the neural network. Prediction results are obtained via the fully connected layer, yielding probability distributions as outputs. FGSM then calculates the gradient of

 doi:10.20944/preprints202510.0404.v1

the loss function with respect to the input image through backpropagation. This gradient indicates each pixel's contribution to the loss function, revealing how individual pixels should be modified during the perturbation process to maximize the loss. Larger gradient magnitudes indicate greater pixel impact on the loss function.

Finally, the perturbation is computed and adversarial samples are generated according to:

$$\delta = \omega^* sign(\nabla_x \mathcal{L}(\theta, x, y)). \tag{15}$$

In this formulation, ϵ represents the perturbation step size, $\nabla_x \mathcal{L}(\theta, x, y)$ denotes the gradient of the loss function with respect to input sample x, $\mathcal{L}(\theta, x, y)$ is the model's loss function, and sign$(\cdot)$ is the sign function. The parameter ϵ determines the magnitude of the adversarial perturbation, with larger values producing more pronounced perturbations. For simpler tasks such as MNIST or SVHN, a smaller value of $\epsilon = 0.03$ is typically chosen. For more complex tasks like CIFAR-10, a larger value of $\epsilon = 0.1$ is selected to enable the model to handle more substantial perturbations. The greater the difference between generated adversarial samples and their original counterparts, the more challenging it becomes to train the model for robust performance.

The calculated perturbations are added to the original samples to generate adversarial samples:

$$H = \mathbf{x} + \delta, \tag{16}$$

where H represents the adversarial sample resulting from adding the perturbation δ to the original input x.

Algorithm 2 Distance-Measured Data Mixing with Adersarial Training

Inputs: unlabeled pool U, labeled pool L, model f_θ, loss $\mathcal{L}$, batch B, group size n, Beta(α), step ϵ, epochs E
while not converged **do**
  Extract final-layer features $z(x) \leftarrow f_\theta$.`features`(x) for all $x \in U$
  For each $x \in U$, compute mixed distances s(x,x') using L1+L2; get top-n neighbors $N(x)$
  Build fused set S via MixUp on pairs from $x \cup N(x)$ with $\lambda \sim$ Beta(α, α)
  Score S by model confidence; select B lowest-confidence indices Q
  for each $(\tilde{x}_i, \tilde{y}_i) \in Q$ **do**
    compute $g = \nabla_x \mathcal{L}(\theta, \tilde{x}_i, \tilde{y}_i)$
    compute $\delta = \epsilon$, sign(g), $H_i = \tilde{x}i + \delta$
  end for
  Acquire labels for $\tilde{x}ii \in Q$; update $L \leftarrow L \cup (\tilde{x}i, y_i), (H_i, y_i)i \in Q$
  for epoch = 1 to E **do**
    for mini-batch (x, y) from L with paired (x^{adv}, y) if present **do**
      $\ell = \frac{1}{2}(\mathcal{L}(\theta, x, y) + \mathcal{L}(\theta, x^{adv}, y))$; update $\theta \leftarrow \theta - \eta \nabla \theta \ell$
    end for
  end for
  Remove labeled items from U; refresh features as needed
end while
Return θ

The proposed method integrates active learning with adversarial training to improve robustness and efficiency in boundary-focused sample selection. In each iteration, the model first extracts final-layer features for all unlabeled samples and computes pairwise similarities using a hybrid of L1 and L2 distances to identify top-n neighbors per seed sample. Candidate training instances are then synthesized via MixUp over seed–neighbor pairs, and the model's confidence on these fused samples is evaluated to select a batch of the most uncertain (near-boundary) examples. For each selected instance, FGSM is applied to generate an adversarial counterpart by taking the sign of the input gradient, scaling by a perturbation step, and adding it to the input. The original fused samples and their adversarial variants are labeled and added to the labeled pool.

 doi:10.20944/preprints202510.0404.v1

Model updates combine natural and adversarial losses with equal weight in mini-batch training over several epochs, promoting accurate predictions on both clean and perturbed inputs. By repeatedly selecting uncertain, feature-near-boundary samples constructed via MixUp and augmenting them with FGSM adversaries, the algorithm reduces labeling requirements while enhancing robustness against noise and perturbations.

5. Theoretical Analysis

5.1. Notation and Setup

Let $U = \{X_1, \ldots, X_m\}$ denote the unlabeled pool and L the labeled pool. A model f_θ with parameters θ produces class probabilities $\hat{Y}_i = f_\theta(X_i)$ and is trained by minimizing the cross-entropy loss in (1), with optimal parameters θ^* given by (2). Feature embeddings are extracted by $U_i = F_{\text{Conv}}(X_i) \in \mathbb{R}^d$ as in (3). Distances are measured by d_{E} and d_{M} in (4)–(5) and combined as d_c in (6). For each anchor X_i, a neighbor set N_i is defined as the indices of the n nearest neighbors under d_c. Mixed inputs are formed at the data level by

$$\hat{X}_i = \lambda X_i + (1-\lambda) \sum_{j \in N_i} X_j \quad \text{with} \quad \lambda \in [0,1], \tag{17}$$

as in (7). The model outputs logits $P_i = f_\theta(\hat{X}_i)$, which are mapped to probabilities via softmax (9), and the confidence is $C_i = \max(\text{softmax}(P_i))$ in (10). The acquisition set comprises the indices of the n lowest-confidence samples, cf. (11).

5.2. Geometric Rationale for Distance-Weighted Mixing

We analyze the effect of DM2 on two axes crucial for active learning: (i) uncertainty exposure at decision boundaries; and (ii) diversity preservation through local neighborhood mixing.

We work under standard conditions often met in deep representation spaces: (A1) The embedding F_{Conv} is locally Lipschitz: $\|F_{\text{Conv}}(X) - F_{\text{Conv}}(X')\| \le L\|X - X'\|$ for some $L > 0$. (A2) The classifier head of f_θ is L_f-Lipschitz in input space on compact domains. (A3) Nearby points under d_c have high label-correlation: there exists $\eta \in [0,1)$ such that for $j \in N_i$, $\mathbb{P}[Y_j \neq Y_i] \le \eta$; equivalently, neighborhoods are label-homogeneous with bounded noise. (A4) Calibration around the decision boundary: near regions where class posteriors are close (small margin), confidence $\max_c \hat{y}_{ic}$ decreases monotonically with the distance to the margin hyper-surface.

Assumption (A1)–(A3) capture that d_c is a surrogate for semantic proximity, while (A4) links geometric proximity to predictive uncertainty.

Consider a first-order expansion of f_θ w.r.t. the input:

$$f_\theta(\hat{X}_i) \approx f_\theta\Big(\lambda X_i + (1-\lambda) \sum_{j \in N_i} X_j\Big) \approx \lambda f_\theta(X_i) + (1-\lambda) \sum_{j \in N_i} f_\theta(X_j) + \varepsilon_i, \tag{18}$$

with a remainder term $\|\varepsilon_i\| \le \frac{1}{2} L_f \big\|\lambda X_i + (1-\lambda)\sum_{j \in N_i} X_j - X_i\big\|^2$ by (A2). Thus, to first order, the logits on the mixed input approximate an average of neighbor logits. When N_i is label-homogeneous, the average logit sharpens the predicted class; when N_i straddles a class boundary, the average logit becomes ambiguous, lowering confidence.

5.3. Uncertainty Amplification Near Class Boundaries

Define the pointwise margin for logits $P(x)$ as

$$\gamma(x) \triangleq P_{(1)}(x) - P_{(2)}(x), \tag{19}$$

the gap between the top two logits. By softmax monotonicity, smaller $\gamma(x)$ implies lower confidence $C(x)$.

 doi:10.20944/preprints202510.0404.v1

Lemma 1: Margin reduction under heterogeneous neighborhoods. Let X_i have neighbors N_i with class proportions π_c ($\sum_c \pi_c = 1$), and suppose f_θ is locally linear around $\{X_i\} \cup \{X_j : j \in N_i\}$. Then, for the mixed input $\hat{X}_i$ with weight $\lambda \in (0,1)$,

$$\gamma(\hat{X}_i) \approx \lambda\, \gamma(X_i) + (1-\lambda)\, \Delta_i, \tag{20}$$

where Δ_i is the top-two logit gap of the neighbor-averaged prediction $\bar{P}_i = \sum_{j \in N_i} P(X_j)$. If N_i spans multiple classes so that $\bar{P}_i$ is class-ambiguous, then Δ_i is small, and hence $\gamma(\hat{X}_i) \leq \lambda\, \gamma(X_i)$, yielding C_i reduced relative to $C(X_i)$.

Proof. Local linearity yields $P(\hat{X}_i) \approx \lambda P(X_i) + (1-\lambda)\bar{P}_i$. Let $a(\cdot)$ denote the top-two logit gap (an affine functional restricted to the two dominant coordinates). Then $a(\lambda p + (1-\lambda)\bar{p}) = \lambda a(p) + (1-\lambda)a(\bar{p})$ for any logits $p, \bar{p}$ sharing the same top-two ordering; otherwise the gap cannot increase beyond the convex combination by triangle inequality. Hence $\gamma(\hat{X}_i) \leq \lambda\gamma(X_i) + (1-\lambda)\Delta_i$. If neighbors are heterogeneous, Δ_i is small due to averaging conflicting logits, which lowers $\gamma(\hat{X}_i)$ and therefore C_i by softmax monotonicity in the gap. □

Samples whose neighbor sets N_i cross decision boundaries are systematically assigned lower confidence after mixing and are thus prioritized by DM2. This aligns selection with true boundary regions where labels are most informative for reducing model uncertainty.

5.4. Diversity Preservation via Distance Coupling

Let $\mathcal{G}$ be the k-NN graph on $\{U_i\}$ under d_c. DM2 forms mixes anchored at many distinct nodes with their local neighborhoods. If the acquisition selects the n lowest-confidence anchors after mixing, these anchors tend to be located on edges or cuts of $\mathcal{G}$ that cross clusters. Under mild clusterability:

(A5) The embedding decomposes into r well-separated clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_r\}$ with inter-cluster distances larger than intra-cluster distances under d_c.

Then boundary regions appear around each cut $(\mathcal{C}_a, \mathcal{C}_b)$; mixed inputs that pool neighbors from both $\mathcal{C}_a$ and $\mathcal{C}_b$ reduce confidence within each cut. Consequently, the n lowest-confidence anchors are spread across multiple cuts, promoting diversity without explicit diversity regularizers.

5.5. Stability of Mixed Confidence Under Neighbor Noise

Consider neighbor noise: a fraction ρ of N_i are erroneous neighbors (e.g., misembedded or outliers). Let P_i^{true} be the average logits over true semantic neighbors and P_i^{noise} over noisy neighbors. Then

$$P(\hat{X}_i) \approx \lambda P(X_i) + (1-\lambda)\big((1-\rho)P_i^{\text{true}} + \rho P_i^{\text{noise}}\big). \tag{21}$$

If $\|P_i^{\text{noise}} - P_i^{\text{true}}\| \leq \delta$ (bounded contamination), then the perturbation to logits is at most $(1-\lambda)\rho\delta$, so the induced confidence change satisfies

$$|C(\hat{X}_i) - \tilde{C}(\hat{X}_i)| \leq L_{\text{sm}}(1-\lambda)\rho\delta, \tag{22}$$

where L_{sm} is the Lipschitz constant of the softmax-max operator. Thus DM2 confidence is robust to small neighbor noise for moderate λ.

5.6. Choice of the Combined Distance d_c

The combined metric $d_c = \frac{1}{2}(d_{\text{E}} + d_{\text{M}})$ inherits the following:

(i) Metric property: since d_{E} and d_{M} are metrics on $\mathbb{R}^d$, any positive weighted sum is a metric. Hence d_c satisfies non-negativity, symmetry, and the triangle inequality.

(ii) Sensitivity balance: d_{E} is sensitive to dense directions, while d_{M} is robust to sparse, axis-aligned deviations. Averaging thus mitigates anisotropy and promotes stable neighbor sets in heterogeneous embeddings.

d_c defined by (6) is a metric on $\mathbb{R}^d$.

 doi:10.20944/preprints202510.0404.v1

Proof. For all $x, y, z \in \mathbb{R}^d$: non-negativity and identity of indiscernibles follow from those of d_{E} and d_{M}. Symmetry is immediate. For the triangle inequality,

$$\begin{aligned} d_c(x,z) &= \tfrac{1}{2}\big(d_{\mathrm{E}}(x,z) + d_{\mathrm{M}}(x,z)\big) \\ &\le \tfrac{1}{2}\big(d_{\mathrm{E}}(x,y) + d_{\mathrm{E}}(y,z) + d_{\mathrm{M}}(x,y) + d_{\mathrm{M}}(y,z)\big) \\ &= d_c(x,y) + d_c(y,z). \end{aligned} \tag{23}$$

□

5.7. Acquisition Optimality Under a Localized Fisher Criterion

Let $\Sigma(x)$ denote the conditional Fisher information of f_θ at input x with respect to parameters θ (under the model distribution). For classification with softmax outputs, points near the decision boundary tend to have larger Fisher trace $\operatorname{tr}\Sigma(x)$, which correlates with higher expected gradient magnitude.

Define the mixed-point Fisher score

$$\Phi_i(\lambda) \triangleq \mathbb{E}\big[\|\nabla_\theta \ell(Y, f_\theta(\hat{X}_i))\|^2 \,\big|\, X_i, N_i\big] \propto \operatorname{tr}\Sigma(\hat{X}_i). \tag{24}$$

Under (A1)–(A4) and local linearization, if N_i is heterogeneous, $\hat{X}_i$ approaches the boundary and $\operatorname{tr}\Sigma(\hat{X}_i)$ increases. Therefore selecting minimum-confidence $\hat{X}_i$ approximately maximizes $\Phi_i(\lambda)$ among anchors, aligning DM2 with a proxy of information gain.

Theorem1: Informative selection under DM2. Suppose (A1)–(A5) hold and that f_θ is locally linear in a neighborhood containing $\{X_i\} \cup \{X_j : j \in N_i\}$. Then, for any fixed $\lambda \in (0,1)$, ranking anchors X_i by ascending confidence C_i on mixed inputs $\hat{X}_i$ is equivalent to ranking by a non-increasing function of the margin $\gamma(\hat{X}_i)$ and thus, up to a monotone transform, by $\operatorname{tr}\Sigma(\hat{X}_i)$. Consequently, the DM2 acquisition set approximates a maximizer of the localized Fisher score among anchors, favoring boundary-spanning, diverse regions of the data manifold.

Proof sketch. Softmax confidence is a monotone function of the logit gap $\gamma(\hat{X}_i)$; hence ordering by C_i equals ordering by $\gamma(\hat{X}_i)$. Under local linearization and (A4), smaller $\gamma(\hat{X}_i)$ implies proximity to the decision boundary, where the Fisher information increases for multinomial logistic models. Thus ranking by C_i approximates ranking by $\operatorname{tr}\Sigma(\hat{X}_i)$. Cluster separation (A5) ensures that anchors selected across different cuts yield coverage of multiple boundary regions (diversity). □

5.8. On the Mixing Coefficient λ

The coefficient λ trades off anchor faithfulness and boundary probing.

- If $\lambda \to 1$, $\hat{X}_i \to X_i$, recovering standard uncertainty sampling. - If $\lambda \to 0$, $\hat{X}_i$ collapses to neighbor averages, which may over-smooth and obscure fine boundaries. - Under (A3), there exists an interval $\Lambda \subset (0,1)$ such that for $\lambda \in \Lambda$, heterogeneous neighborhoods strictly reduce $\gamma(\hat{X}_i)$ relative to $\gamma(X_i)$ while homogeneous neighborhoods preserve or increase it. Therefore, DM2 self-selects anchors with heterogeneous N_i.

Existence of a beneficial mixing range. Assume there exist anchors with $\gamma(X_i) > 0$ and heterogeneous N_i such that $\Delta_i < \gamma(X_i)$ in Lemma 1. Then for any $\lambda \in (0,1)$,

$$\gamma(\hat{X}_i) \le \lambda\,\gamma(X_i) + (1-\lambda)\Delta_i < \gamma(X_i). \tag{25}$$

Thus confidence strictly decreases for such anchors; conversely, if N_i is homogeneous with large margin, confidence is non-decreasing for λ near 1.

Proof. Immediate from Lemma 1 and the strict inequality $\Delta_i < \gamma(X_i)$. □

 doi:10.20944/preprints202510.0404.v1

5.9. Considerations and Summary

Let $m = |U|$ and d the feature dimension. Computing pairwise d_c naively is $O(m^2 d)$; approximate k-NN reduces this to near-linear time in m. Mixing and forward passes scale as $O(mn \cdot C_f)$ where $n = |N_i|$ and C_f is model inference cost. Hence, with approximate neighbors and mini-batched evaluation, DM2 scales to large pools.

Mixing within d_c-based neighborhoods yields mixed inputs whose logits approximate convex combinations of neighbor logits. Heterogeneous neighborhoods reduce the logit margin and thus confidence, preferentially surfacing boundary samples for labeling. The acquisition is robust to moderate neighbor noise and approximates selection by localized Fisher information. The combined distance d_c is a proper metric that balances Euclidean and Manhattan sensitivities, stabilizing neighbor selection.

6. Experimental Results

We evaluate our method against state-of-the-art and baseline active learning approaches, including random selection, least confidence selection, and entropy sampling. Our approach is validated on both balanced and unbalanced image classification tasks using MobileNet [22] architectures. We conduct experiments on MNIST [35], CIFAR-10 [21], SVHN [36], and CIFAR-10 mixed with CIFAR-100 noise. Experiments employ 7 to 10 active learning cycles with labeling budgets ranging from 20 to over 2500 samples per cycle. Data selections follow standard active learning practices without replacement, and all results are averaged over 3 runs. All experiments are implemented using PyTorch [37].

For MNIST, we employ a CNN classification model with the Adam optimizer [38] at learning rate 10^{-3} and batch size 96, training for 50 epochs per cycle. For CIFAR-10, CIFAR-10s, and SVHN, we use MobileNet [22] with SGD optimizer [39], initial learning rate 0.1, batch size 128, momentum 0.9, and weight decay 5×10^{-4}. Training proceeds for 200 epochs with learning rate decay to 0.01 at epoch 160.

We compare against established baselines including Random Selection[3], Entropy[23], Least Confidence[5], Margin[6], BALD[4], CoreSet[12], EntropyBayesian[7], UncertainGCN[40], BADGE[19], ProbCover[29], Alpha-Mix[1], and NoiseStability[20], using identical parameters for fair comparison.

6.1. Efficiency for Distance-Measure Data Mixing

Table 1 demonstrates that our method outperforms all other active learning approaches across most datasets. The BfFus method achieves superior performance compared to all baseline active learning methods on all datasets except MNIST. For the MNIST dataset, the simplicity and limited data volume result in the boundary-based selection model failing to learn useful information more rapidly than simpler selection strategies.

 doi:10.20944/preprints202510.0404.v1

Table 1. Accuracy (%) of experimental results with different datasets.

Methods	MNIST	CIFAR-10	CIFAR-10s	SVHN	Avg
Entropy[23] (IJCNN14)	92.78±0.14	84.00±0.13	64.00±1.42	70.91±9.01	77.92
Margin[6] (WIREs14)	**93.42±0.13**	83.62±0.07	63.47±0.37	69.64±4.84	77.53
Least Confidence[5]	92.98±0.13	83.51±0.04	63.89±1.79	70.52±2.25	77.73
Random[3]	87.78±0.32	81.62±0.19	62.94±0.01	69.82±3.42	75.54
EntropyBayesian[7][7] (ICML16)	92.05±0.65	83.62±0.25	61.82±0.56	71.21±4.13	77.18
CoreSet[12]	89.45±1.12	83.04±0.14	63.78±0.72	71.41±1.94	76.85
UncertainGCN[40] (NIPS20)	87.56±2.47	83.51±0.10	61.40±4.43	70.65±0.07	75.78
ProbCover[29] (NIPS22)	88.75±0.68	81.63±0.17	64.13±1.88	69.04±12.58	75.89
BALD[4] (ICML17)	92.55±1.22	82.03±0.22	63.62±1.81	70.36±3.12	77.14
BADGE[19]	92.35±0.06	83.83±0.05	62.46±1.69	71.94±2.35	77.65
Alpha-Mix[1] (CVPR22)	93.10±0.46	84.22±0.05	62.58±0.72	69.43±0.28	77.25
NoiseStability[20] (IJCAI24)	92.63±0.37	83.87±0.04	62.55±0.36	69.87±19.85	77.23
DM2(Ours)	93.11±0.34	**84.29±0.01**	**64.69±3.92**	**72.29±2.69**	**78.60**

The excellent performance on the SVHN dataset demonstrates that the BfFus method exhibits strong stability when handling unbalanced datasets. This robustness likely stems from the method's concise design, which avoids data imbalance issues that can negatively impact auxiliary model training and subsequently degrade task model performance. In experiments with CIFAR-10s containing noisy data, the BfFus method successfully identifies samples conducive to model learning and demonstrates superior noise stability compared to competing approaches.

To further validate our method's robustness, we conducted additional experiments using CIFAR-10 and SVHN datasets with ResNet18 [41] and VGG16 [**?**] architectures. Standard data augmentation techniques were applied during training, including random horizontal flipping and cropping. As shown in Table 2, our method consistently outperforms all other active learning approaches across these more complex architectures.

Table 2. Accuracy (%) of experimental results under large scale models.

Methods	SVHN (ResNet18)	CIFAR-10 (vgg16)	Avg
Entropy[23]	92.94±0.04	75.97±5.56	84.46
Margin[6]	92.99±0.01	79.51±0.24	86.25
Least Confidence[5]	93.02±0.01	79.57±0.18	86.3
Random[3]	91.32±0.09	77.84±0.08	84.58
EntropyBayesian[7]	91.38±0.09	77.12±3.34	84.27
CoreSet[12]	92.48±0.01	77.10±0.08	84.79
UncertainGCN[40]	91.57±0.06	78.82±0.05	85.20
ProbCover[29]	90.88±0.19	77.62±0.17	84.25
BALD[4]	92.67±0.02	75.51±1.2	84.09
BADGE[19]	93.04±0.06	79.20±0.76	86.12
Alpha-Mix[1]	92.69±0.03	79.06±0.34	85.88
NoiseStability[20]	92.91±0.04	79.10±0.42	86.01
DM2(Ours)	**93.05±0.05**	**79.67±0.07**	**86.36**

6.2. Robustness for Adversarial Training

The comparative experimental results for different active learning strategies are presented in Table 3. The results demonstrate that our method incorporating adversarial training achieves higher model accuracy compared to other active learning approaches. This indicates that the boundary data feature fusion algorithm with adversarial training proposed in this work effectively improves model performance.

 doi:10.20944/preprints202510.0404.v1

Table 3. Comparison of Accuracy Results of Different Methods on Different Models (%).

Methods	MNIST-C (CNN)	SVHN-C (MobileNet)	CIFAR10-C (MobileNet)	SVHN-C (ResNet18)	CIFAR10-C (ResNet34)
Entropy[23]	84.74±2.54	72.51±1.25	67.21±0.54	79.53±0.89	74.25±2.56
Least Confidence[5]	85.62±1.67	73.66±0.72	68.35±1.85	81.20±2.56	75.61±1.87
Margin[6]	84.83±1.34	73.24±1.54	67.04±1.78	79.98±2.28	74.32±1.43
Random[3]	82.61±3.78	70.28±2.66	63.27±0.28	78.61±3.95	73.01±1.22
EntropyBayesian[7]	83.34±1.32	71.64±2.84	63.59±2.72	78.91±0.31	73.20±0.18
CoreSet[12]	84.52±2.54	72.91±1.52	62.18±3.36	79.01±1.02	73.61±2.45
UncertainGCN[40]	83.11±1.98	71.37±1.73	64.36±1.84	78.23±2.43	72.93±1.95
ProbCover[29]	80.13±2.75	70.01±3.43	65.23±0.76	79.54±3.61	73.81±3.68
BALD[4]	84.55±1.43	72.29±2.85	64.87±2.05	79.02±0.88	72.77±2.84
BADGE[19]	85.78±0.74	73.35±1.94	65.79±1.96	80.11±1.96	74.32±1.29
Alpha-Mix[1]	85.91±0.32	73.59±1.22	67.91±2.67	81.32±1.61	75.29±2.71
NoiseStability[20]	85.26±1.22	73.03±2.51	66.52±1.43	80.89±2.27	73.04±1.43
DM2-AT(Ours)	**86.62±0.89**	**74.58±2.13**	**69.04±1.69**	**82.03±2.44**	**76.61±1.76**

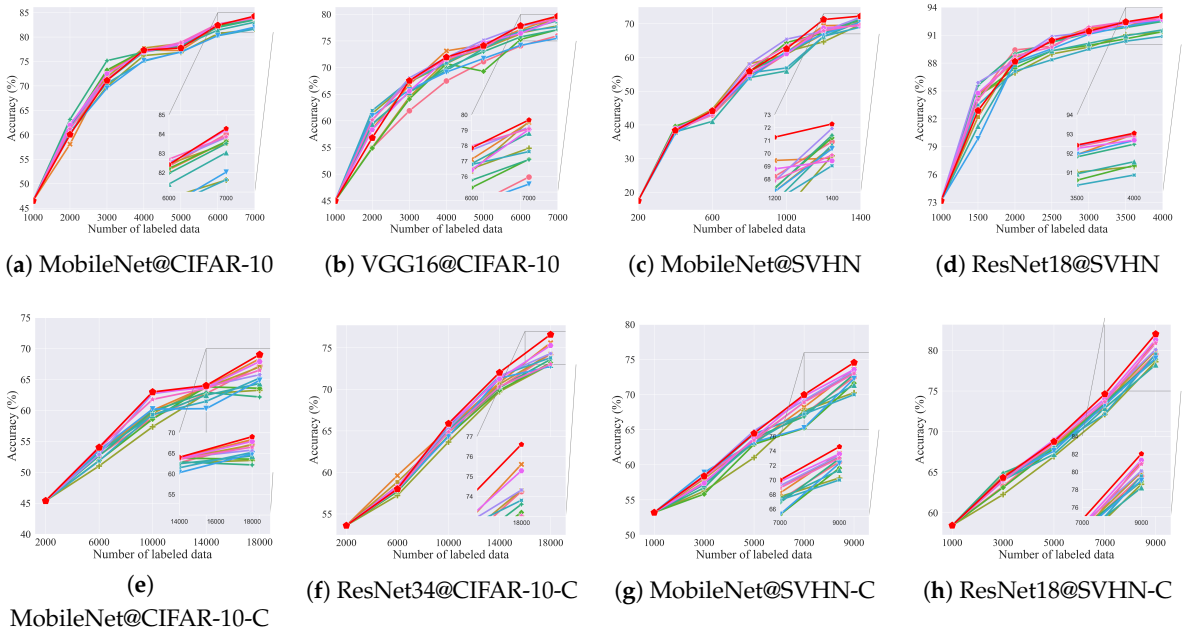
The experiments also reveal that conventional active learning methods often struggle to enhance performance on test sets in complex environments. Traditional active learning approaches can only learn from complete datasets and exhibit reduced effectiveness when recognizing data in challenging conditions. By integrating adversarial training with the method from Chapter 4, our approach generates adversarial samples based on selected data, enhancing the model's ability to learn from difficult-to-classify samples. This enables more effective integration and learning from data in complex environments, thereby improving overall performance.

The experimental results in Table 3 show that the LlBfFus method proposed in this work significantly outperforms competing methods across different models and datasets, confirming the effectiveness of our approach. The method increases learning challenges by generating adversarial samples through adversarial training on selected samples. When facing real-world scenarios with complex environments, this approach demonstrates stronger capability in identifying samples under noise interference, efficiently reducing the data requirements for building machine learning models while achieving superior performance results.

6.3. Convergence Analysis

To gain deeper insight into the performance trends of our method across different data selection cycles, we plotted convergence graphs based on the results from Tables 1 and 2. Figure 4 presents trend plots for six different datasets and model combinations. To ensure fair comparison, all experimental values are averaged over three runs using consistent learning rates and parameters across all methods.

 doi:10.20944/preprints202510.0404.v1



(**a**) MobileNet@CIFAR-10 (**b**) VGG16@CIFAR-10 (**c**) MobileNet@SVHN (**d**) ResNet18@SVHN

(**e**) MobileNet@CIFAR-10-C (**f**) ResNet34@CIFAR-10-C (**g**) MobileNet@SVHN-C (**h**) ResNet18@SVHN-C

Figure 4. Classification performance of different methods across various datasets and models. The graphs are divided into two groups: experiments on datasets for DM2 are shown at the upper panels, while experiments for DM2-AT are at the lower panels. Our method shows significant advantages across multiple datasets and models, proving its wide applicability.

Figure 4 is organized into two groups: the upper panels show results for DM2, while the lower panels display results for DM2-AT. The method demonstrates gradual improvement in subsequent epochs, particularly evident in CIFAR-10 experiments using MobileNet and VGG16 architectures, where our approach surpasses competing methods. For CIFAR-10s, the model quickly learns to select more relevant data, with our method ultimately outperforming all alternatives. On SVHN, our method demonstrates clear superiority by the 6th round, consistently outperforming other approaches throughout the remaining cycles.

Our method exhibits significant advantages across multiple datasets and model architectures, demonstrating broad applicability. In contrast, competing methods show less adaptability and inconsistent performance across different datasets. Panel (e) reveals that the CIFAR-10-C dataset presents certain challenges, with the trend chart showing some fluctuations in data selection performance. In panel (f), experiments using the CIFAR-10-C dataset with ResNet34 show smoother progression and higher performance compared to the earlier MobileNet results.

The ResNet18 model trained on SVHN-C in panel (h) exhibits favorable convergence trends. Except for the 3000th iteration where LlBfFus did not surpass several competing methods, it achieved excellent results across all other iterations. In panel (g), SVHN-C initially presents challenges during early training phases. However, after processing 5,000 data points, the model rapidly identifies samples with stronger feature information, leading to sharp performance improvements in later stages and ultimately achieving superior results. These findings demonstrate the effectiveness of our active learning approach that fuses features from adversarial training boundary data.

 doi:10.20944/preprints202510.0404.v1

6.4. Time Efficiency

The computational efficiency of active learning methods depends on the cost of sample distance calculations and subset selection procedures, as presented in Table 4. We evaluated the time efficiency of several effective methods using identical hyperparameters from our experiments. Our methods demonstrate computational efficiency comparable to state-of-the-art algorithms and exhibit favorable scalability as the annotation budget or number of categories increases.

Table 4. Time effect result.

Method	Mnist (/s)	CIFAR-10 (/m)
BADGE	2	2.07
EntropyBayesian	4	3.25
BALD	16	4.58
NoiseStability	20	4.21
Our	3	**2.07**

6.5. Ablation Experiment

The active learning method based on boundary data feature fusion with adversarial training proposed in this work incorporates two key components: boundary data feature fusion and adversarial training. Since removing adversarial training yields a method similar to that in Chapter 3, we conduct ablation experiments focusing on sample distances, fusion ratios, and perturbation values, where the perturbation value determines adversarial sample effectiveness.

To further assess method validity, we replace components in the ablation study: using Euclidean distance instead of the combined Euclidean and Manhattan distance approach, employing equal-weight fusion instead of adaptive fusion ratios, and using a fixed perturbation value of 0.05 instead of computed adaptive values. For fair comparison, identical models are used across all ablation variants. Table 5 presents the ablation results, demonstrating that adversarial training constitutes an integral component of the overall method, significantly enhancing fault tolerance and robustness.

Table 5. Ablation Study: Accuracy Comparison Results (%).

Datasets	-Distance	-Fusion Ratio	- Perturbation	DM2-AT
MNIST-C (CNN)	83.48 ± 0.96	84.31 ± 1.43	85.62 ± 0.56	86.62 ± 0.89
CIFAR10-C (MobileNet)	73.31 ± 1.37	72.22 ± 2.27	73.58 ± 1.59	74.58 ± 2.13
SVHN-C (MobileNet)	66.48 ± 1.42	66.21 ± 0.86	67.04 ± 1.43	69.04 ± 1.69
SVHN-C (ResNet18)	78.90 ± 1.55	80.34 ± 2.92	81.03 ± 0.86	82.03 ± 2.44
CIFAR10-C (ResNet34)	74.91 ± 0.68	75.43 ± 1.38	75.61 ± 2.34	76.61 ± 1.76

According to the experimental results in Table 5, replacing the combined Euclidean and Manhattan distances with single Euclidean distance in the adversarial training boundary data feature fusion algorithm demonstrates that this simplified approach struggles to accurately capture data distributions across different datasets. This frequently leads to model confusion and prevents optimal performance achievement. Ablation experiments using equal-weight fusion as the control variable show that uniform fusion causes merged features to lose distinctiveness, resulting in deteriorated model recognition performance.

The ablation study confirms that the adversarial training-based boundary data feature fusion algorithm enhances sample efficiency in active learning while improving the model's generalization capability across diverse datasets and challenging conditions.

7. Conclusion

Active learning represents a prominent research direction for deep neural networks, enabling efficient model training with reduced sample requirements. We propose a simple yet stable method

 doi:10.20944/preprints202510.0404.v1

that exploits inter-sample relationships and data distribution characteristics. Through uncertainty prediction based on similarity measures and weighted mixing strategies, our approach demonstrates superior performance in both theoretical analysis and experimental evaluation across multiple tasks. The integration of adversarial training with boundary data feature fusion further enhances model robustness and generalization capability in complex environments.

Future work will focus on addressing more challenging scenarios where the computational efficiency and cost reduction benefits of active learning become increasingly significant. We aim to extend our approach to handle larger-scale datasets and more complex domain adaptation problems, where traditional supervised learning approaches face substantial annotation costs and computational constraints.

Acknowledgments: This paper was supported by Natural Science Foundation of Jilin Province under Grant YDZJ202401610ZYTS.

References

1. Parvaneh, A.; Abbasnejad, E.; Teney, D.; Haffari, G.R.; Van Den Hengel, A.; Shi, J.Q. Active learning by feature mixing. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 12237–12246.
2. Munjal, P.; Hayat, N.; Hayat, M.; Sourati, J.; Khan, S. Towards robust and reproducible active learning using neural networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 223–232.
3. Settles, B. Active learning literature survey. Technical report, 2009.
4. Gal, Y.; Islam, R.; Ghahramani, Z. Deep bayesian active learning with image data. In Proceedings of the International conference on machine learning. PMLR, 2017, pp. 1183–1192.
5. Lewis, D.D. A sequential algorithm for training text classifiers: Corrigendum and additional data. In Proceedings of the Acm Sigir Forum. ACM New York, NY, USA, 1995, Vol. 29, pp. 13–19.
6. Kremer, J.; Steenstrup Pedersen, K.; Igel, C. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2014**, *4*, 313–326.
7. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the international conference on machine learning. PMLR, 2016, pp. 1050–1059.
8. Freund, Y.; Seung, H.S.; Shamir, E.; Tishby, N. Selective sampling using the query by committee algorithm. *Machine learning* **1997**, *28*, 133–168.
9. Gorriz, M.; Carlier, A.; Faure, E.; Giro-i Nieto, X. Cost-effective active learning for melanoma segmentation. *arXiv preprint arXiv:1711.09168* **2017**.
10. Ducoffe, M.; Precioso, F. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841* **2018**.
11. Mayer, C.; Timofte, R. Adversarial sampling for active learning. In Proceedings of the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 3071–3079.
12. Sener, O.; Savarese, S. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489* **2017**.
13. Caramalau, R.; Bhattarai, B.; Kim, T.K. Sequential graph convolutional network for active learning. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 9583–9592.
14. Yoo, D.; Kweon, I.S. Learning loss for active learning. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 93–102.
15. Liu, Z.; Ding, H.; Zhong, H.; Li, W.; Dai, J.; He, C. Influence selection for active learning. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 9274–9283.
16. Wang, T.; Li, X.; Yang, P.; Hu, G.; Zeng, X.; Huang, S.; Xu, C.Z.; Xu, M. Boosting active learning via improving test performance. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2022, Vol. 36, pp. 8566–8574.
17. Koh, P.W.; Liang, P. Understanding black-box predictions via influence functions. In Proceedings of the International conference on machine learning. PMLR, 2017, pp. 1885–1894.
18. Sinha, S.; Ebrahimi, S.; Darrell, T. Variational adversarial active learning. In Proceedings of the Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 5972–5981.

 doi:10.20944/preprints202510.0404.v1

19. Ash, J.T.; Zhang, C.; Krishnamurthy, A.; Langford, J.; Agarwal, A. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671* **2019**.
20. Li, X.; Yang, P.; Gu, Y.; Zhan, X.; Wang, T.; Xu, M.; Xu, C. Deep active learning with noise stability. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 13655–13663.
21. Krizhevsky, A.; Hinton, G.; et al. Learning multiple layers of features from tiny images. Toronto, ON, Canada, 2009.
22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* **2017**.
23. Wang, D.; Shang, Y. A new active labeling method for deep learning. In Proceedings of the 2014 International joint conference on neural networks (IJCNN). IEEE, 2014, pp. 112–119.
24. Kirsch, A.; Van Amersfoort, J.; Gal, Y. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems* **2019**, *32*.
25. Houlsby, N.; Huszár, F.; Ghahramani, Z.; Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745* **2011**.
26. Loquercio, A.; Segu, M.; Scaramuzza, D. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters* **2020**, *5*, 3153–3160.
27. Kuo, W.; Häne, C.; Yuh, E.; Mukherjee, P.; Malik, J. Cost-sensitive active learning for intracranial hemorrhage detection. In Proceedings of the Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part III 11. Springer, 2018, pp. 715–723.
28. Beluch, W.H.; Genewein, T.; Nürnberger, A.; Köhler, J.M. The power of ensembles for active learning in image classification. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 9368–9377.
29. Yehuda, O.; Dekel, A.; Hacohen, G.; Weinshall, D. Active learning through a covering lens. *Advances in Neural Information Processing Systems* **2022**, *35*, 22354–22367.
30. Elhamifar, E.; Sapiro, G.; Yang, A.; Sasrty, S.S. A convex optimization framework for active learning. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 209–216.
31. Hasan, M.; Roy-Chowdhury, A.K. Context aware active learning of activity recognition models. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4543–4551.
32. Citovsky, G.; DeSalvo, G.; Gentile, C.; Karydas, L.; Rajagopalan, A.; Rostamizadeh, A.; Kumar, S. Batch active learning at scale. *Advances in Neural Information Processing Systems* **2021**, *34*, 11933–11944.
33. Yang, C.; Wu, Q.; Li, H.; Chen, Y. Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340* **2017**.
34. Guo, R.; Chen, Q.; Liu, H.; Wang, W. Adversarial robustness enhancement for deep learning-based soft sensors: An adversarial training strategy using historical gradients and domain adaptation. *Sensors* **2024**, *24*, 3909.
35. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.
36. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y.; et al. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS workshop on deep learning and unsupervised feature learning. Granada, 2011, Vol. 2011, p. 4.
37. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch, 2017.
38. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
39. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers. Springer, 2010, pp. 177–186.
40. Zhao, X.; Chen, F.; Hu, S.; Cho, J.H. Uncertainty aware semi-supervised learning on graph data. *Advances in Neural Information Processing Systems* **2020**, *33*, 12827–12836.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

 doi:10.20944/preprints202510.0404.v1