

Collaborative Intelligence Orchestration: Inconsistency-Based Fusion of Semi-Supervised Learning and Active Learning

Jiannan Guo
Zhejiang University
jiannan@zju.edu.cn

Xiaozhong Liu
Indiana University Bloomington
liu237@indiana.edu

Kun Kuang
Zhejiang University
kunkuang@zju.edu.cn

Yangyang Kang
Alibaba Group
yangyang.kangyy@alibaba-inc.com

Siliang Tang*
Zhejiang University
siliang@zju.edu.cn

Changlong Sun
Alibaba Group
changlong.scl@taobao.com

Yu Duan
Alibaba Group
derrick.dy@alibaba-inc.com

Wenqiao Zhang
Zhejiang University
wenqiaozhang@zju.edu.cn

Fei Wu
Zhejiang University
wufei@zju.edu.cn

ABSTRACT

While annotating decent amounts of data to satisfy sophisticated learning models can be cost-prohibitive for many real-world applications. Active learning (AL) and semi-supervised learning (SSL) are two effective, but often isolated, means to alleviate the data-hungry problem. Some recent studies explored the potential of combining AL and SSL to better probe the unlabeled data. However, almost all these contemporary SSL-AL works use a simple combination strategy, ignoring SSL and AL's inherent relation. Further, other methods suffer from high computational costs when dealing with large-scale, high-dimensional datasets. Motivated by the industry practice of labeling data, we propose an innovative Inconsistency-based virtual aDversarial Active Learning (IDEAL) algorithm to further investigate SSL-AL's potential superiority and achieve mutual enhancement of AL and SSL, *i.e.*, SSL propagates label information to unlabeled samples and provides smoothed embeddings for AL, while AL excludes samples with inconsistent predictions and considerable uncertainty for SSL. We estimate unlabeled samples' inconsistency by augmentation strategies of different granularities, including fine-grained continuous perturbation exploration and coarse-grained data transformations. Extensive experiments, in both text and image domains, validate the effectiveness of the proposed algorithm, comparing it against state-of-the-art baselines. Two real-world case studies visualize the practical industrial value of applying and deploying the proposed data sampling algorithm.

CCS CONCEPTS

- Computing methodologies → Active learning settings; Semi-supervised learning settings.

*Corresponding Author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '22, August 14–18, 2022, Washington, DC, USA

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

KEYWORDS

active learning, semi-supervised learning, adversarial learning

ACM Reference Format:

Jiannan Guo, Yangyang Kang, Yu Duan, Xiaozhong Liu, Siliang Tang, Wenqiao Zhang, Kun Kuang, Changlong Sun, and Fei Wu. 2022. Collaborative Intelligence Orchestration: Inconsistency-Based Fusion of Semi-Supervised Learning and Active Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA*. ACM, New York, NY, USA, 11 pages.

1 INTRODUCTION

The utilization of large training datasets is an essential ingredient to endorse the success of deep learning [18, 19, 33, 34] and its applications (*e.g.*, image classification [17], text classification [5, 7]). However, in real-world systems, the acquisition of decent labeled data is cost-prohibitive and challenging (*e.g.*, legal text annotation [30] can be expensive because of domain expert scarcity). Therefore, initiating deep learning with sparse [20] and ever-growing training data, with/without human involvement, can be a vital task to explore.

To address this problem, prior studies embraced two directions: 1) *active learning*, (AL) [10, 22, 31] aims to select the most informative samples for human annotation with the least labeling cost (Figure 1(a)); 2) *semi-supervised learning*, (SSL) [2, 4, 15] also faces such challenges with labeled data sparseness and unlabeled data abundance. Along this line of research, the potential of combining SSL and AL (*i.e.*, SSL-AL) can be nontrivial to further improve model performance. However, recent SSL-AL investigations are limited by the following two drawbacks: 1). Most current SSL-AL methods [14, 27, 32] with VAE-GAN structure suffer from the mismatch problem (Figure 1(b)), as learned representations of samples does not facilitate (or even do harm to) the classification. 2). Current methods [14, 27, 32] employ SSL and AL as two isolated modules, while great potentials can be unleashed by enabling the sophisticated interaction between SSL and AL (Figure 1(c)).

Indeed, AL and SSL can be complementary when they share the same goal (*i.e.*, utilizes unlabeled samples), and they can work collaboratively to achieve mutual enhancement: 1) (SSL→AL), SSL can provide enhanced embeddings to assist AL to exploit unlabeled samples' underlying distribution and evaluate unlabeled samples' attributes; 2) (AL→SSL), AL can exclude some complex examples

from the unlabeled pool and help SSL reduce the uncertainty of model’s prediction. Utilizing the potential relatedness among various learning methods can be regarded as a form of inductive transfer, e.g., *inductive bias* [1], to equip the combined learning model with the correct hypothesis and performance improvement. SSL-AL reciprocal optimization is a novel but critical problem for human-in-the-loop AI.

In this work, we utilize ‘sample inconsistency’ as the bridge to enable the interaction between SSL and AL, which is inspired by the fact that the sample inconsistency is an effective means for both SSL [2] and AL [11]. The robustness against the sample inconsistency (e.g., the translation invariance of CNNs) can provide essential information to enhance models’ generalization ability. Furthermore, the prior study [23] has shown that not only human-perceivable but also human-unperceivable sample variance can lead to a significant impact on models’ prediction.

To address these challenges, we propose a novel Inconsistency-based virtual aDvErsarial Active Learning (IDEAL) framework to enhance human-in-the-loop AI by leveraging effective SSL-AL fusion. IDEAL carries three modules: **the SSL label propagator** propagates the label information from the sparse training data to the unlabeled samples, which is a mixture of coarse-grained (human-perceivable) and fine-grained (human-unperceivable) augmentations. It yields the task model with high consistency and low entropy. Then, **the virtual inconsistency ranker** ranks all the unlabeled samples in terms of their coarse-grained and fine-grained inconsistency and selects top samples (*i.e.*, poor consistency) as the rough annotation candidates. Finally, **the density-aware uncertainty re-ranker** further filters the selected samples (*i.e.*, high entropy), and provides the final samples for human annotation.

To validate the performance of IDEAL, we provide a comprehensive evaluation on four benchmark datasets and two real-world datasets. Experiments demonstrate that IDEAL outperforms state-of-the-art approaches and makes noticeable economic benefits.

Our major contribution can be summarized as follows:

- We propose a novel Inconsistency-based virtual aDvErsarial Active Learning (IDEAL) framework where SSL and AL form a closed-loop structure for reciprocal optimization and enhancement.
- We develop novel coarse-grained and fine-grained data augmentation strategies and couple them to both SSL and AL. Specially, we leverage the virtual adversarial technique to explore the continuous local distribution of unlabeled samples and further measure the inconsistency of samples.
- We evaluate IDEAL over diverse text and image datasets. The results validate IDEAL is superior than previous approaches and it is cost effective in various industrial applications.

2 RELATED WORK

In practice, it is easy to acquire abundant unlabeled samples. Thus pool-based AL [25, 27, 31, 32] is more popular than the other two scenarios: steam-based [6] and membership query synthesis [29].

• **Pool-based active learning.** Uncertainty-based sampling and distribution-based sampling are common methods in the pool-based scenario. Our method considers both uncertainty and distribution. For uncertainty-based methods, the framework [13] uses Gaussian

processes to evaluate uncertainty, while [8] uses a Bayesian network. MC dropout [10] is used as an approximation of the Bayesian network. However, these methods have a high computational cost and can not handle large-scale datasets. In the era of deep-learning-based AL, [36] and [31] adopt similar metrics to choose informative samples. [36] selects samples leading to the greatest change to the model’s gradient during the training process, while [31] selects samples with the biggest training loss. However, the task model’s loss and gradient information are unstable in the early training stage, and they may influence the quality of the final selected samples.

The distribution-based method [25] selects samples of a subset whose feature distribution covers the entire feature space as much as possible. However, subset selection (NP-hard problem) becomes computationally infeasible for large-scale datasets or high-dimension input data. Thus, the algorithms of selecting subsets will suffer from inefficient computing.

• **Semi-supervised active learning.** Recently, [14, 27, 32] leverage VAE-GAN structure to learn the representation of both labeled and unlabeled samples in latent space. However, these methods misuse the relation between labeling states and class labels. Consequently, the learning process may harm the semantic distribution of samples in latent space. Annotation information (labeling states) and the feature distribution (class labels) are orthogonal, while the feature distribution is highly correlated with semantic information of different classes. Compared to these VAE-GAN based methods, our method considers inconsistency of different granularity to combine AL and SSL. The works [11, 28] combine AL and SSL based on prediction consistency given a set of data augmentations. However, these methods only use a limited number of ways of human-perceivable data augmentation to estimate inconsistency. In contrast, our method further leverages human-unperceivable inconsistency to obtain more abundant information of unlabeled samples for model estimation and optimization. Besides, we also consider the feature distribution of samples.

Our method is related to the latest work [12] in terms of hierarchical sample selection. Sadly, [12] suffers from high computational and spatial costs caused by its critical step (builds a KNN graph in every selection cycle). Consequently, deploying [12] in real-world scenarios to handle large-scale datasets faces enormous challenges. In addition, [12] does not integrate adversarial perturbation generation into the training process. Thus, the selection criterion (adversarial perturbation) can not adapt dynamically to the model’s optimization. Our method is free of the above two deficiencies. Firstly, we replace graph SSL with the Mixup technique to handle large-scale datasets. Then we couple selection criterion (fine-grained inconsistency) to model optimization for optimal perturbation generation.

Unlike our work, all of these methods are specially designed for image classification tasks, which can not be directly transferred to tasks of other media types (*e.g.*, text type).

3 METHOD

In this section, we formulate the proposed Inconsistency-based virtual aDvErsarial Active Learning (IDEAL) algorithm. We first provide a brief overview of our whole AL framework, and then

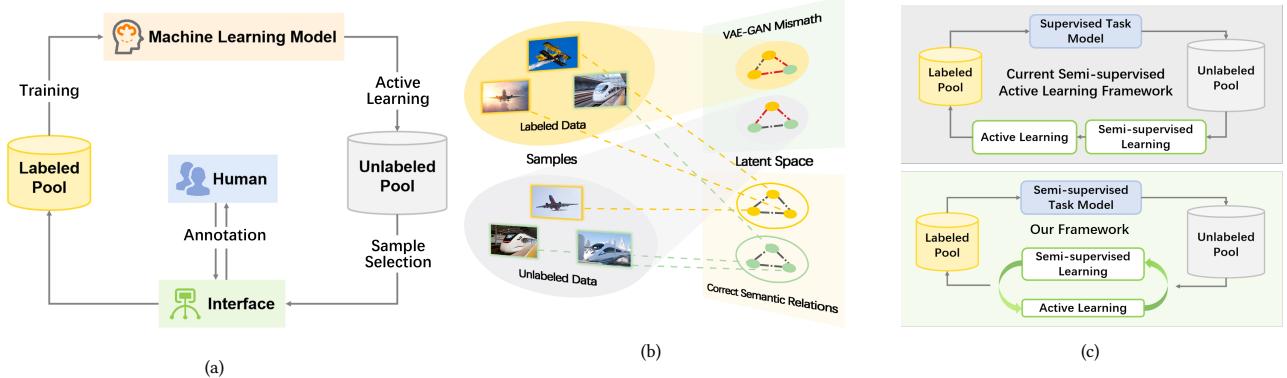


Figure 1: (a) Active learning illustration. (b) Since samples’ labeling states and class labels are uncorrelated, these methods tend to project the representation of samples with different class labels (e.g., train and plane) to the same class (i.e., labeled or unlabeled), and vice versa. (c) Comparison of current SSL-AL methods and our approach.

introduce three main components of IDEAL: a SSL label propagator (section 3.2), a virtual inconsistency ranker (section 3.3) and a density-aware uncertainty re-ranker (section 3.4).

3.1 Overview

In this section, we formally describe the pool-based AL loop with our proposed IDEAL demonstrated in Figure 2. We define the labeled pool as $\mathcal{D}^l = \{(\mathbf{x}_1^l, y_1^l), \dots, (\mathbf{x}_{N_l}^l, y_{N_l}^l)\}$ and the unlabeled pool as $\mathcal{D}^u = \{\mathbf{x}_1^u, \dots, \mathbf{x}_{N_u}^u\}$ (N_l and N_u are the numbers of labeled samples and unlabeled samples respectively). (1) For samples in the pool, IDEAL first feeds them into the SSL label propagator, which propagates the label information from labeled samples to unlabeled samples by mixing up samples. (2) Based on the inconsistency signals provided by the SSL label propagator, our virtual inconsistency ranker calculates the total inconsistency (coarse-grained and fine-grained inconsistency) for each unlabeled sample in the pool \mathcal{D}^u , and selects the top- \mathcal{M} samples with the largest inconsistency as initial annotation candidates. (3) The density-aware uncertainty re-ranker further selects top- \mathcal{K} candidates with the largest density-aware uncertainty from initial top- \mathcal{M} candidates for human annotation, finally excluding unsmoothed and unstable samples for SSL to perform better label propagation. As a consequence, the sizes of the labeled pool and the unlabeled pool will be updated to $N_l + \mathcal{K}$ and $N_u - \mathcal{K}$ respectively. The loop will be repeated until the annotation budget is run out.

3.2 SSL Label Propagator

We leverage the SSL label propagator to propagate the label information from labeled samples to unlabeled samples by smoothing local inconsistency distribution of unlabeled samples, and obtain enhanced embeddings for following AL.

Formally speaking, given unlabeled dataset $\mathcal{D}^u = \{\mathbf{x}_1^u, \dots, \mathbf{x}_{N_u}^u\}$ and the labeled dataset $\mathcal{D}^l = \{(\mathbf{x}_1^l, y_1^l), \dots, (\mathbf{x}_{N_l}^l, y_{N_l}^l)\}$, we firstly generate K augmented unlabeled samples $\mathbf{X}_n^u = \{\mathbf{x}_{n,1}^u, \dots, \mathbf{x}_{n,K}^u\}$ for each unlabeled sample $\mathbf{x}_n^u \in \mathcal{D}^u$ through coarse-grained augmentations. Specifically, the coarse-grained augmentation is a family of data augmentation functions T that s.t. $\mathbf{x}_{aug} = T(\mathbf{x})$ and

$\max(\mathbf{x}_{aug} - \mathbf{x}) > \delta$ (where δ is a threshold discriminating human-perceivable and human-unperceivable difference). The corresponding inconsistency estimation is called to be coarse-grained, and vice versa. Moreover, we adopt sentence-level transformations, rewriting sentences with similar meanings (back translation) as the coarse-grained augmentation strategy for text data. We use image-level transformations (e.g., rotation and clipping of the whole image) to obtain coarse-grained augmented image samples.

In addition to the above discrete coarse-grained data transformations, we introduce pixel-level or embedding-level fine-grained augmentations, adding local perturbation to inputs’ representations in feature space (described in section 3.3), to force the model to explore continuous local distribution.

After applying these augmentation strategies of two different granularity, we can obtain the final augmentation samples $\tilde{\mathbf{X}}_n^u = \{\tilde{\mathbf{x}}_{n,1}^u, \dots, \tilde{\mathbf{x}}_{n,K}^u\}$ for samples in \mathbf{X}_n^u .

Then, we feed each unlabeled sample \mathbf{x}_n^u and its augmented samples $\{\tilde{\mathbf{x}}_{n,1}^u, \dots, \tilde{\mathbf{x}}_{n,K}^u\}$ into the task model to get predicted labels \hat{y}_n^u and $\{\hat{y}_{n,1}^u, \dots, \hat{y}_{n,K}^u\}$. We generate shared guessed label for augmentation samples by averaging all of their predicted labels:

$$\bar{y}_n^a = \frac{1}{w_u + \sum_k w_k} (w_u \hat{y}_n^u + \sum_k w_k \hat{y}_{n,k}^u). \quad (1)$$

where weights w_u and w_k mean the contribution ratios of different augmentation samples to the final guessed label. We perform a weighted average operation to guarantee the task model to output consistent predictions for different augmentation samples. We set weights parameters for original samples and their augmented samples based on translation qualities of different languages in text classification. For images, we set all weights parameters to 1.

Next, we obtain the labeled set \mathcal{D}^l , the unlabeled set \mathcal{D}^u and the augmented set $\mathcal{D}^a = \{(\tilde{\mathbf{X}}_1^u, \hat{y}_1^a), \dots, (\tilde{\mathbf{X}}_{N_u}^u, \hat{y}_{N_u}^a)\}$ (\mathcal{D}^u and \mathcal{D}^a share the same labels). We can obtain training samples by randomly mixing two batches of samples’ hidden states from these three sets as well as their respective labels (or guessed labels) together.

We can obtain three types of mixed samples: (1) mixing two batches of labeled samples; (2) mixing a batch of labeled and a batch

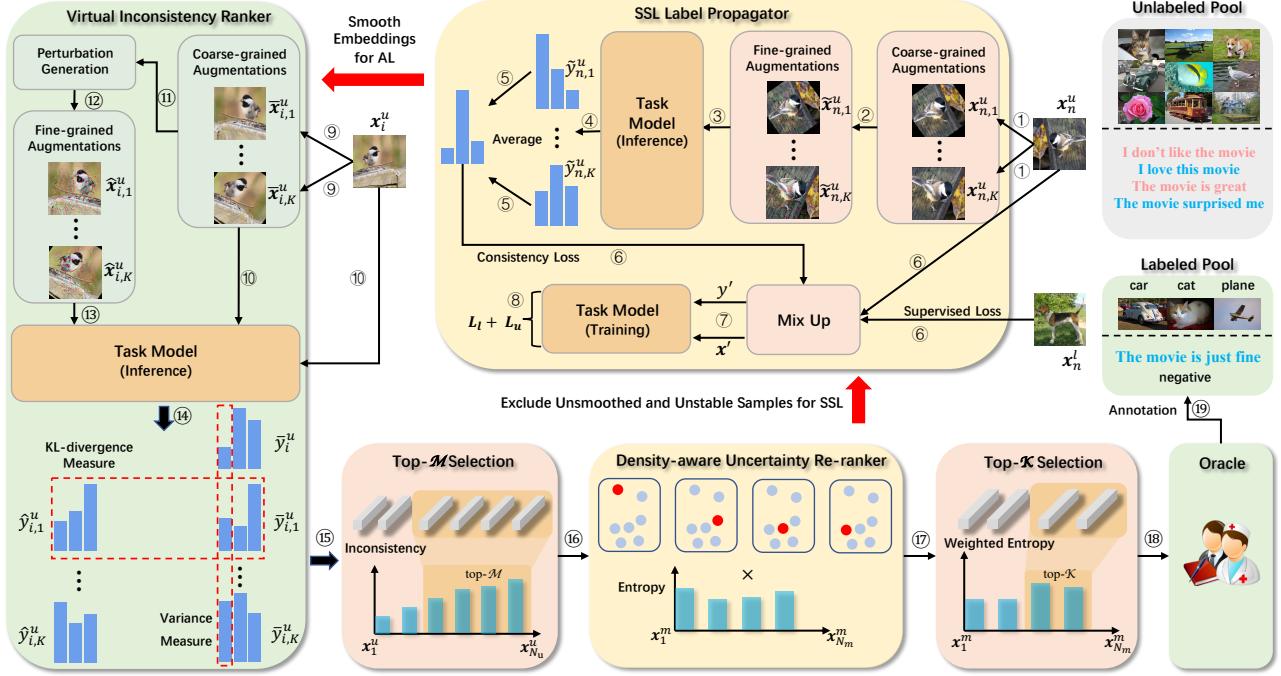


Figure 2: An overview of our proposed AL framework IDEAL which consists of three modules: (a) *SSL Label Propagator* propagates label information from labeled samples to unlabeled samples and smooths local distribution for AL; (b) *Virtual Inconsistency Ranker* estimates inconsistency by calculating coarse-grained and fine-grained inconsistency respectively; (c) *Density-aware Uncertainty Re-ranker* further screens the samples with high density-aware uncertainty.

of unlabeled samples; (3) and mixing two batches of unlabeled samples. This can be defined as:

$$\mathbf{x}' = \lambda \mathbf{h}_1 + (1 - \lambda) \mathbf{h}_2. \quad (2)$$

$$y' = \lambda y_1 + (1 - \lambda) y_2. \quad (3)$$

where $\mathbf{h}_1, \mathbf{h}_2$ are the hidden states corresponding to the samples \mathbf{x}_1 and \mathbf{x}_2 . In detail, for text tasks, we select the hidden states of the middle layer of the task model for mixing up. For image tasks, we mix up the raw input features directly. Besides, the weight factor λ is sampled from *Beta Distribution*, and it can be further defined as:

$$\lambda' = Beta(\alpha, \alpha). \quad (4)$$

$$\lambda = max(\lambda', 1 - \lambda'). \quad (5)$$

where α is the hyper-parameter of the *Beta* distribution λ' . After the mixing process, the mixed feature \mathbf{x}' aggregates both labeled samples' and unlabeled samples' features. Similarly, the mixed label y' aggregates supervised label information from labeled samples and distribution information of unlabeled samples.

Finally, we define the loss function as $L_l + L_u$: cross-entropy loss term L_l is minimized through supervised label information. Consistency loss term L_u (e.g., KL-divergence or L2 norm) is applied to constrain different augmented samples to have the same label with their original unlabeled samples.

Trained with mixed samples, the label propagator will propagate rich label information from labeled samples to unlabeled samples. The propagator smoothes the local distribution of unlabeled samples and makes the task model insensitive to both coarse-grained data transformation and fine-grained perturbation, providing enhanced embeddings for AL. Mixup mines and optimizes local and global samples distribution for SSL and AL. Apart from the local distribution mining based on the smoothing of point-wise augmentation, Mixup explores convex combinations of pairs of labeled, unlabeled, and augmented samples to mine the global distribution. Since enumerating all the convex combinations of pairs of samples to precisely model the whole picture is intractable, we utilize the random sampling method to estimate the distribution and optimize the corresponding objectives to learn the desired distribution. Transformed/untransformed samples with similar classes are close.

3.3 Virtual Inconsistency Ranker

Above SSL label propagator can smooth the local distribution around unlabeled samples, which means that the task model's predictions for unlabeled samples tend to be consistent before and after perturbation. Our goal is to select the samples with poor consistency for further annotation from \mathcal{D}^u . Because these non-smooth samples fail to acquire enough label information, the task model has a vague understanding of the local distribution of these samples. In this sense, selecting non-smooth samples for further annotation are more valuable, which can help the task model smooth unlabeled samples better and output more stable predictions.

To select these non-smooth samples, the virtual inconsistency ranker estimates the inconsistency of the task model's prediction on unlabeled samples under augmentation strategies of two different granularity. Specifically, for each unlabeled sample \mathbf{x}_i^u in the same unlabeled pool as mentioned in section 3.2 (this section, we use a different symbol \mathbf{x}_i^u to distinguish between the selection stage and the SSL training stage): (1) we first obtain its coarse-grained augmentation set $\bar{\mathbf{X}}_i^u = \{\bar{\mathbf{x}}_{i,1}^u, \dots, \bar{\mathbf{x}}_{i,K}^u\}$ by aforementioned augmentations, and feed \mathbf{x}_i^u and $\bar{\mathbf{X}}_i^u$ into the task model to get their predictions \bar{y}_i^u and $\bar{\mathbf{Y}}_i^u = \{\bar{y}_{i,1}^u, \dots, \bar{y}_{i,K}^u\}$. (2) Then we feed $\bar{\mathbf{X}}_i^u$ and $\bar{\mathbf{Y}}_i^u$ simultaneously into the ranker to get adversarial perturbation $\mathbf{R}_i^{adv} = \{\mathbf{r}_{i,1}^{adv}, \dots, \mathbf{r}_{i,K}^{adv}\}$ for each augmented sample $\bar{\mathbf{x}}_{i,k}^u \in \bar{\mathbf{X}}_i^u$. (3) After that, we feed each fine-grained augmented sample $\hat{\mathbf{x}}_{i,k}^u = \bar{\mathbf{x}}_{i,k}^u + \mathbf{r}_{i,k}^{adv}$ into the task model again to get their fine-grained augmented predictions $\hat{\mathbf{Y}}_i^u = \{\hat{y}_{i,1}^u, \dots, \hat{y}_{i,K}^u\}$. Besides, we conduct the fine-grained augmentations on texts' embedding vectors extracted from the middle layer of the task model, as texts' input space is discrete. The process of generating adversarial perturbation is formulated as:

$$\mathbf{r}^{adv} = \arg \max_{\Delta \mathbf{r}, \|\Delta \mathbf{r}\| \leq \epsilon} KL(p(\bar{y}^u | \bar{\mathbf{x}}^u, \theta), p(\hat{y}^u | \bar{\mathbf{x}}^u + \Delta \mathbf{r}, \theta)). \quad (6)$$

where $p(y|\mathbf{x}, \theta)$ represents the posterior probability of the task model. Under perturbation of the same norm ϵ , there is a higher probability for adversarial samples of unlabeled samples with unstable predictions to change their original label and obtain predictions of other classes. Therefore, the ranker computes the KL-divergence of the posterior probability of samples and their adversarial samples to measure the inconsistency of unlabeled samples.

Since the computation of \mathbf{r}^{adv} is intractable for many neural networks, [23] proposed to approximate \mathbf{r}^{adv} using the second-order Taylor approximation and solved the \mathbf{r}^{adv} via the power iteration method. Specifically, we can approximate \mathbf{r}^{adv} by applying the following update:

$$\mathbf{r}^{adv} \leftarrow \epsilon \nabla_{\Delta \mathbf{r}} \overline{KL(p(\bar{y}^u | \bar{\mathbf{x}}^u), p(\hat{y}^u | \bar{\mathbf{x}}^u + \Delta \mathbf{r}))}. \quad (7)$$

where $\Delta \mathbf{r}$ is a randomly sampled unit vector, $p(\bar{y}^u | \bar{\mathbf{x}}^u)$ is the label for coarse-grained augmented samples, $p(\hat{y}^u | \bar{\mathbf{x}}^u + \Delta \mathbf{r})$ is the fine-grained augmented prediction, and the sign $\bar{\mathbf{v}}$ means the unit vector of \mathbf{v} . The computation of $\nabla_{\Delta \mathbf{r}} \overline{KL}$ can be performed with one set of backpropagation for neural networks.

Once the adversarial perturbation \mathbf{r}^{adv} is solved, we can estimate the total inconsistency [11] of unlabeled samples under coarse-grained and fine-grained augmentations. Coarse-grained inconsistency can be formulated as:

$$In_{coa}(\mathbf{x}_i^u) = \sum_{c=1}^C Var[p(\bar{y}_i^u = c | \mathbf{x}_i^u, \theta), p(\bar{y}_{i,1}^u = c | \mathbf{x}_{i,1}^u, \theta), \dots, p(\bar{y}_{i,K}^u = c | \mathbf{x}_{i,K}^u, \theta)], \quad (8)$$

where C is the number of classes, and Var means the variance. Furthermore, fine-grained inconsistency can be formulated as:

$$In_{fin}(\mathbf{x}_i^u) = \sum_{k=1}^K KL(\bar{y}_{i,k}^u, \hat{y}_{i,k}^u). \quad (9)$$

where KL means the KL-divergence. To combine the above two selection criteria together, we normalize and transform them into percentiles. Specially, we denote $\Phi\varphi(\mathbf{x}_i^u, \mathcal{D}^u)$ as the percentile of the criteria φ of a given sample among the dataset \mathcal{D}^u . For instance, $\Phi\varphi(\mathbf{x}_i^u, \mathcal{D}^u) = 75\%$ indicates that 75% of the samples in \mathcal{D}^u are smaller than \mathbf{x}_i^u under the criteria φ . We can calculate the total inconsistency of each unlabeled sample as:

$$In(\mathbf{x}_i^u) = \gamma \cdot \Phi_{In_{coa}}(\mathbf{x}_i^u, \mathcal{D}^u) + (1 - \gamma) \cdot \Phi_{In_{fin}}(\mathbf{x}_i^u, \mathcal{D}^u). \quad (10)$$

where γ is the weight coefficient to balance two criteria. Finally, we select top- \mathcal{M} samples with the largest inconsistency $In(\mathbf{x}_i^u)$ from unlabeled samples as the initial annotation candidates.

3.4 Density-aware Uncertainty Re-ranker

Based on the goal described in section 3.3, the virtual inconsistency ranker roughly selects annotation candidates with large inconsistency, which can be regarded as an initial recall set of final potential samples. However, there is still a gap between the goal of the virtual inconsistency ranker (*i.e.* select samples with large inconsistency) and our final AL goal (*i.e.* select samples with high uncertainty). Here, we propose a density-aware uncertainty re-ranker to rank the annotation candidates further, guaranteeing that the selected samples are with high uncertainty and large inconsistency. In this way, the top samples in the candidates set can bring great model uncertainty reduction.

In practice, we use the entropy of predicted labels from the task model to estimate the uncertainty of samples and select top- \mathcal{K} samples from former top- \mathcal{M} annotation candidates. The entropy of i -th unlabeled sample \mathbf{x}_i^m from the set \mathcal{M} can be calculated with the following formulation:

$$En'(\mathbf{x}_i^m) = - \sum_c^C P(y_c | \mathbf{x}_i^m) \log P(y_c | \mathbf{x}_i^m). \quad (11)$$

where $P(y_c | \mathbf{x}_i^m)$ is the probability of the i -th unlabeled sample in the set \mathcal{M} belonging to the c -th class.

However, the above entropy-based formula only estimates the uncertainty information of each sample and fails to take the distribution relations among samples into account. As a result, the metric may run the risk of selecting some outliers or unrepresentative samples in the distribution space. To alleviate this issue, we re-weight the uncertainty metric with a representativeness factor and explicitly consider the distribution structure of samples. We denote this density-aware uncertainty as:

$$En(\mathbf{x}_i^m) = En'(\mathbf{x}_i^m) \times \left(\frac{1}{M} \sum_{\mathbf{x}' \in \mathcal{M}} sim(\mathbf{x}_i^m, \mathbf{x}') \right). \quad (12)$$

where M is the size of the candidates set \mathcal{M} , and the additional second term in the formula is the similarity of each sample \mathbf{x}_i^m relative to other samples in the distribution space. We can obtain the similarity term through cosine similarity, Euclidean distance, Spearman's rank correlation, or any other metrics. Considering that cosine similarity is common to compute the similarity of texts' embedding and also effective for images' similarity computation, we use cosine similarity in our paper:

$$sim(\mathbf{x}_i^m, \mathbf{x}') = \frac{\mathbf{x}_i^{mT} \cdot \mathbf{x}'}{||\mathbf{x}_i^m|| \times ||\mathbf{x}'||}. \quad (13)$$

and this density-aware indicator can help us select samples with high uncertainty as well as spatial representativeness.

Accordingly, SSL and AL can work collaboratively for reciprocal enhancement in this framework. On the one hand, the SSL label propagator can propagate label information of labeled samples and smooth the local distribution around unlabeled samples through a mixture of labeled and unlabeled samples. Then SSL can provide a consistent signal to help AL decide the annotation candidates. On the other hand, the virtual inconsistency ranker and the density-aware uncertainty re-ranker enable human to annotate the samples with the largest inconsistency plus the highest uncertainty weighted by representativeness. AL assists the SSL label propagator exclude unstable and unsmoothed augmentation samples for better label propagation. We detail IDEAL in Algorithm 1 in Appendix A.1.

4 EXPERIMENTS

To validate the proposed method, we conduct a comprehensive evaluation in both text and image domains, with four benchmark datasets and two new real-world datasets. We initiate each task with a small (randomly sampled) labeled training set, and the remaining unlabeled data are used as candidates for IDEAL/baseline’s AL/SSL components. Once labeled by human, samples will be added to the training set. For each cycle, we train the task model with the newly updated training set. For a fair comparison, all baselines share the same initial training set and model parameters. All the figures in this study depict the results over five experiment trials.¹

4.1 Datasets

We choose two text classification benchmarks: AG News [35] and IMDB [21], and two image classification benchmarks: CIFAR-10 [16] and CIFAR-100 [16]. Furthermore, we collect two industrial text classification datasets. The Legal Text dataset is collected from Chinese court trial cases, used to classify and retrieve similar cases. We define 12 fact labels (elements of judgment basis) for each case. The Bidding dataset is used to facilitate the users to filter the procurement methods (different companies prefer different procurement methods) and help them locate the most suitable bidding opportunities. We define 22 labels of purchase and sale for each announcement document. Unlike benchmarks, Legal Text and Bidding datasets come from industrial applications, and their annotations need experts with domain knowledge. The average annotation price of each sample for these two datasets is \$0.91 and \$0.92 respectively.

4.2 Baselines

For text classification tasks, we employ four baselines for comparison: (1) EGL [36]; (2) Core-set [25]; (3) MC-dropout [26]; (4) Random sampling [9]. For supervised image classification tasks, we adopt seven baselines including: (1) Core-set [25]; (2) Random sampling [9]; (3) ICAL [11]; (4) SRAAL [32]; (5) VAAL [27]; (6) LLAL [31]; (7) REVIVAL [12]. Since some baselines under supervised learning can not be applied or transferred to SSL scenarios

¹To facilitate other scholars reproduce the experiment outcomes, we will make all the data code available via <https://github.com/AmazingJeff/IDEAL>.

directly, we use different baselines in the SSL scenario. The baseline details can be found in Appendix A.3.

4.3 Evaluation Settings

We adopt Wide ResNet-28 [24] as the backbone of task models for image classification, while we use a BERT model along with a two-layer MLP architecture as text tasks’ backbone. The initial training set is uniformly distributed over classes, and we set up the initial set by randomly sampling. Under the supervised learning scenario, we only use the labeled data to train the task model. Thus, we leave out the operations of the SSL propagator. Under the SSL scenario, SSL hyper-parameters have been well-explored in MixText [4] and Mixmatch [2]. They found in practice that hyper-parameters can be fixed and do not need to be tuned on a per-experiment or per-dataset basis. Thus, we follow these empirical settings. For text, we set K to 2 (back translation from German and Russian). Based on translation qualities, the corresponding weight parameters (w_u , w_{Ger} , w_{Ru}) for original samples and their augmented samples are (1, 1, 0.5). Over real-world datasets, we set back translation (English and German) weight parameters (w_u , w_{En} , w_{Ger}) to (1, 0.8, 0.2). For images, we set K to 5 (5 augmentations obtained by horizontally flipping and random cropping) and all weight parameters w to 1. We set α to 16 and 0.75 for text and image datasets respectively. Further, we refer to the work [23] to fine-tune the hyper-parameter ϵ , and we obtain ϵ as 1e-2 and 10 for text and image data respectively.

4.4 Performance Analysis

- **Performance for image classification.** Figures 3(a) and 3(b) depict the supervised learning performance of IDEAL on image benchmarks. For CIFAR-10 (Figure 3(a)), we can observe that IDEAL outperforms evidently against state-of-the-art methods in all selection cycles. In Figure 3(b), IDEAL beats all baselines with a **margin up to 1.37%**. Figures 3(c) and 3(d) show that, with the SSL setting, IDEAL is superior to baselines throughout the entire sample selection process. Compared to AL, SSL brings essential improvements early in the selection process, so it is necessary to have different selection batch sizes for better visualization of the performance increase under the SSL scenario. In Figure 3(c), the accuracy of methods under the SSL scenario (with 150 labeled samples) is higher than that under the supervised learning scenario (with 4000 labeled samples). IDEAL reaches 94.6% accuracy with 2000 labeled samples, while previous SOTA requires more than 3500 (IDEAL significantly **reduces the annotation cost by at least 42.8%**). Compared to mere SSL (random selection), IDEAL brings a huge reduction in annotation cost of 75% (4000 → 1000) to reach 93.69% accuracy. In Figure 3(d), when using 12500 labeled samples, semi-supervised IDEAL achieves 8.57% more accuracy than supervised IDEAL.

- **Performance for text classification.** Figures 4(a) and 4(b) show the results of supervised IDEAL for text classification. One can witness that IDEAL **outperforms all baselines** throughout the whole sample selection stage. On the ‘BERT scale’, IDEAL leads to decent performance improvement, which validates the effectiveness of the proposed method. Figures 4(c) and 4(d) show the results of semi-supervised IDEAL for the text classification task, and the Random represents the text SSL method MixText (w/o AL). From the figures, we can see that IDEAL **consistently demonstrates**

the best performances across different labeled data numbers in both datasets. The graph propagator limits REVIVAL’s performance by failing to capture complex semantic relations among text samples and build a high-quality relation graph. In contrast, IDEAL leverages the Mixup technique to propagate label information for subsequent inconsistency estimation. Further, IDEAL can not only measure the prediction inconsistency of augmented samples but estimates the robustness of samples’ embedding to adversarial perturbation as well. Thus, IDEAL significantly improves the naive SSL method (the Random) by excluding unsmoothed and unstable samples, compared to ICAL and REVIVAL.

Table 1: Annotation cost comparison with similar accuracy on Legal and Bidding datasets. Superscript \dagger indicates the SOTA SSL-AL model. \downarrow suggest that the small value is better.

Methods	Dataset	Setting	N	\$/L	Cost \$ \downarrow	Accuracy %
Random [9]		SSL	9,000	0.91	8,190	87.71
ICAL [11] \dagger	Legal	SSL-AL	7,000	0.91	6,370	87.77
IDEAL		SSL-AL	5,000	0.91	4,550	88.36
Random [9]	Bidding	SSL	2,500	0.92	2,300	85.35
IDEAL		SSL-AL	1,000	0.92	920	86.25
ICAL [11] \dagger	Bidding	SSL-AL	2,500	0.92	2,300	87.60
IDEAL		SSL-AL	2,000	0.92	1,840	87.70

• **Cost saving analysis of industrial application.** To address IDEAL’s potential advancement on industrial applications, we perform an in-depth performance plus cost-saving analysis on two real-world industrial text datasets. As shown in Figures 5(a), 5(c) and Figures 5(b), 5(d), IDEAL maintains consistent superiority over other SOTAs in all experiment settings. In real scenarios, AL is used to improve the system’s performance to a certain usable standard at the cost of minimal annotations. From this perspective, a more practical way to measure AL’s performance is to compare the annotation cost saving (Labeling cost per label (\$/L) \times Number of labels (N)) demanded to reach a specific system performance. Table 1 summarizes the statistics of annotation costs: 1) **Legal dataset.** IDEAL reaches 88.36% accuracy with 5,000 labeled samples, reducing 44.44% (**IDEAL vs Random sampling, saving \$3640**) and 28.57% (**IDEAL vs ICAL, saving \$1820**) annotations, respectively. 2) **Bidding dataset.** IDEAL achieves 86.25% and 87.70% accuracy with 1,000 and 2,000 labeled samples, respectively. With the better performance, IDEAL reduces 60.0% (**IDEAL vs Random sampling, saving \$1380**) and 20.0% (**IDEAL vs ICAL, saving \$460**) of annotation cost. In sum, IDEAL is promising to reduce the annotation cost significantly for various industrial AI efforts. Notably, the analysis also shows that the data magnitude and cost-cutting scale are positively correlated. In other words, as the data required from industrial tasks grows, IDEAL can investigate unlabeled data deeply, and the annotation cost saving can rise in lockstep with the data scale. For instance, the number of unlabeled samples in a legal fact-finding dataset exceeds 10 million; IDEAL can expedite the data annotation process and save a few hundred thousand dollars in the best-case scenario.

5 MODEL ANALYSIS

To further validate the proposed IDEAL’s effectiveness, we select the most commonly used dataset (CIFAR-100) and perform an in-depth analysis of our method on the dataset.

5.1 Robustness to Hyperparameters

- **Hyperparameter \mathcal{M} .** We conduct experiments to analyze the proposed method’s robustness to hyper-parameter \mathcal{M} against the CIFAR-100 dataset under supervised learning. The results are shown in Figure 6(a). In each selection stage, the primary selection is made by inconsistency, and then the further selection is made by entropy. Their impacts on performance vary with \mathcal{M} . Besides, top- \mathcal{K} is not a hyperparameter as it is equal to the budget size (2500) in each cycle. When $\mathcal{M} = 2500$ (the number of samples in primary selection equals the budget), only inconsistency affects the performance. When \mathcal{M} equals the number of all unlabeled samples (equivalent to no primary selection), only entropy affects the performance. When \mathcal{M} is in between, inconsistency and entropy work together. From $\mathcal{M} = 6500$ to $\mathcal{M} = 17500$, the impact of inconsistency and entropy on performance is relatively stable (IDEAL has strong robustness to \mathcal{M}). After $\mathcal{M} = 17500$, the impact of inconsistency begins to decrease, and correspondingly, the effect of entropy begins to increase. We tend to choose smaller \mathcal{M} to save computational overhead while ensuring accurate performance. Thus, we set \mathcal{M} to 6500. We can obtain similar results for other datasets.

- **Hyperparameter γ .** We conduct an exploratory analysis to systematically infer a proper weight coefficient γ for balancing coarse-grained and fine-grained inconsistency. On the one hand, we can observe from Figure 6(b) that IDEAL obtains the best performance (accuracy of 64.74%) when $\gamma=0.4$, i.e., the weight of coarse-grained and fine-grained inconsistency are 0.4 and 0.6, respectively. These results suggest that the fine-grained inconsistency has a higher weight to select informative AL samples. This phenomenon is reasonable because the fine-grained perturbation (pixel-level or embedding-level) can guide the model to deeply explore continuous local distribution non-smooth deeply and select the non-smooth samples. On the other hand, using the coarse-grained or fine-grained inconsistency as the major guidance ($\gamma=0.1$ or 0.9) to choose AL samples will yield relatively unsatisfactory results. The findings indicate the importance of balance between coarse-grained or fine-grained inconsistency. In other words, the two granularities of inconsistency in IDEAL can work together to pick the informative unlabeled samples as AL annotation candidates for superior SSL in the following learning cycle.

5.2 Ablation Study

We present ablation studies to evaluate the contribution of critical modules of IDEAL on the CIFAR-100 dataset in Table 2. For convenience, we use Ranker and Re-ranker to represent virtual inconsistency ranker and density-aware uncertainty re-ranker respectively. Without the ranker, the model accuracy yields a significant drop (72.23% \rightarrow 70.19%). Diving into the ranker, coarse-grained inconsistency estimation helps SSL exclude unstable samples (not robust to augmentation methods), while fine-grained inconsistency estimation helps SSL exclude unsmoothed samples (sensitive to adversarial perturbation). Besides, the continuous exploration of local

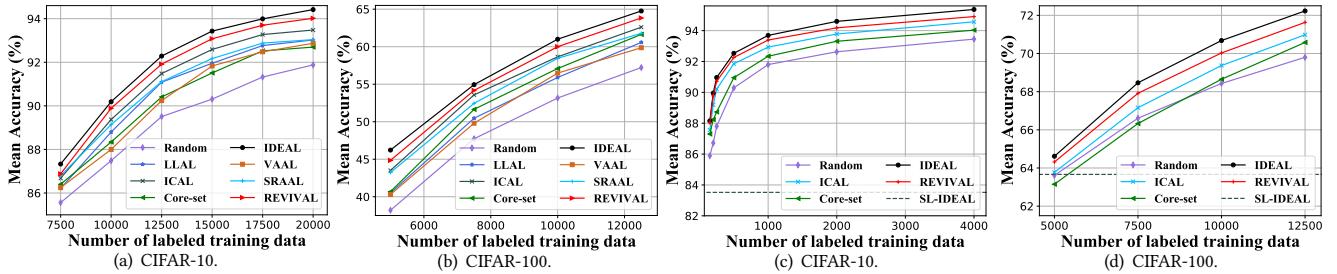


Figure 3: Performance over image benchmarks. Results in (a) and (b) are obtained under supervised learning. Results in (c) and (d) are obtained under semi-supervised learning. The dotted lines (SL-IDEAL) at the bottom in (c) and (d) represent the performance of IDEAL under supervised learning with 4000 and 12500 labeled data respectively.

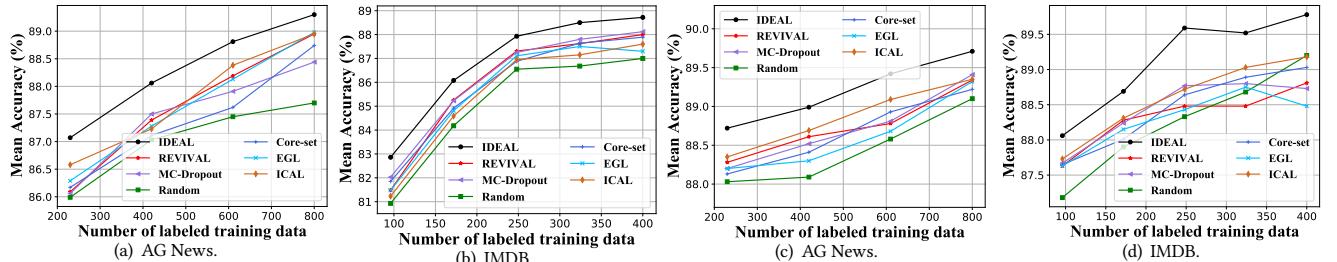


Figure 4: Performance over text benchmarks. Results in (a) and (b) are obtained under supervised learning. Results in (c) and (d) are obtained under semi-supervised learning.

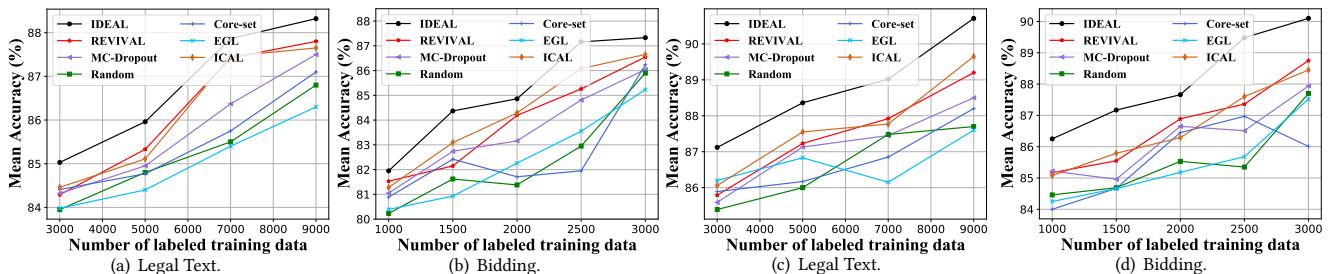


Figure 5: Algorithm performance in real-world scenarios. Results in (a) and (b) are obtained under supervised learning. Results in (c) and (d) are obtained under semi-supervised learning.

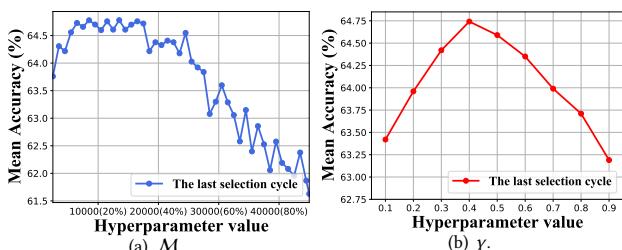


Figure 6: Performance under different hyper-parameters of \mathcal{M} and γ on CIFAR-100 dataset.

distribution is more beneficial than limited and discrete data augmentations ($71.38\% \rightarrow 71.09\%$). Moreover, the re-ranker can further constrain the entropy of initially selected samples and take samples' feature distribution into account. Based on our proposed density-aware uncertainty, the re-ranker further boosts model performance from 71.68% to 72.23%.

Table 2: Ablation study under semi-supervised learning over CIFAR-100. Symbol - indicates that IDEAL excludes the corresponding module.

Methods	Number of labeled samples	
	7,500	12,500
IDEAL	68.46 ± 0.17	72.23 ± 0.20
- density-aware module	68.24 ± 0.21	71.93 ± 0.19
- Re-ranker	68.03 ± 0.18	71.68 ± 0.18
- coarse-grained inconsistency	67.71 ± 0.18	71.38 ± 0.19
- fine-grained inconsistency	67.34 ± 0.20	71.09 ± 0.19
- Ranker	66.93 ± 0.19	70.19 ± 0.17
Random	66.61 ± 0.22	69.82 ± 0.22

6 CONCLUSION

Annotated data is the backbone of vast ML algorithms, and low-resource learning, w/wo human engagement, becomes a vital task. In this study, we propose a novel inconsistency-based virtual adversarial active learning (IDEAL) framework where SSL and AL

formed a unique closed-loop structure for reciprocal enhancement. IDEAL can locate optimal unlabeled samples for human annotation based on innovative coarse-grained and fine-grained inconsistency estimation and density-aware uncertainty. To validate the effectiveness and practical advantages of IDEAL, we conducted experiments over several benchmarks and real-world datasets. Experimental results across text and image domains witness that IDEAL could make significant improvements over existing SOTA methods. In the future, we will investigate more sophisticated models to enable enhanced collaborations between human and algorithm, e.g., using explainable SSL to maximize human annotators' contribution.

ACKNOWLEDGMENTS

This work has been supported in part by National Key Research and Development Program of China (2018AAA0101900), Zhejiang NSF (LR21F020004), Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, Alibaba Group through Alibaba Research Intern Program, Key Research and Development Program of Zhejiang Province, China (No.2021C01013), Key Research and Development Plan of Zhejiang Province (Grant No.2021C03140), Chinese Knowledge Center of Engineering Science and Technology (CKCEST).

The author Guo would also like to thank his girlfriend, Miss Earth (Xiaolin Li), for the considerable support in plotting figures.

REFERENCES

- [1] Jonathan Baxter. 2000. A model of inductive bias learning. *Journal of artificial intelligence research* 12 (2000), 149–198.
- [2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*. 5049–5059.
- [3] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of Semantic Representation: Dataless Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 2. 830–835.
- [4] Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2147–2157.
- [5] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*. 452–461.
- [6] Ido Dagan and Sean P Engelson. 1995. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*. Elsevier, 150–157.
- [7] Pedro Henrique Luz de Araujo, Teófilo Emídio de Campos, Fabricio Ataides Braz, and Nilton Correia da Silva. 2020. VICTOR: a dataset for Brazilian legal documents classification. In *Proceedings of The 12th Language Resources and Evaluation Conference*. 1449–1458.
- [8] Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. 2020. Uncertainty-guided Continual Learning with Bayesian Neural Networks. In *International Conference on Learning Representations*.
- [9] Rosa L Figueiroa, Qing Zeng-Treitler, Long H Ngo, Sergey Goryachev, and Eduardo P Wiechmann. 2012. Active learning for clinical text classification: is it better than random sampling? *Journal of the American Medical Informatics Association* 19, 5 (2012), 809–816.
- [10] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian Active Learning with Image Data. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. 1183–1192.
- [11] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö Arik, Larry S Davis, and Tomas Pfister. 2020. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *European Conference on Computer Vision*. Springer, 510–526.
- [12] Jiannan Guo, Haochen Shi, Yangyang Kang, Kun Kuang, Siliang Tang, Zhuoren Jiang, Changlong Sun, Fei Wu, and Yueteng Zhuang. 2021. Semi-supervised active learning for semi-supervised models: exploit adversarial examples with graph-based virtual labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2896–2905.
- [13] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. 2007. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 1–8.
- [14] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. 2021. Task-Aware Variational Adversarial Active Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*. 8166–8175.
- [15] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [16] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
- [17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.
- [18] Juncheng Li, Siliang Tang, Linchao Zhu, Haochen Shi, Xuanwen Huang, Fei Wu, Yi Yang, and Yueteng Zhuang. 2021. Adaptive hierarchical graph reasoning with semantic coherence for video-and-language inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1867–1877.
- [19] Juncheng Li, Junlin Xie, Long Qian, Linchao Zhu, Siliang Tang, Fei Wu, Yi Yang, Yueteng Zhuang, and Xin Eri Wang. 2022. Compositional Temporal Grounding with Structured Variational Cross-Graph Correspondence Learning. *arXiv preprint arXiv:2203.13049* (2022).
- [20] Mengze Li, Tianbiao Wang, Haoyu Zhang, Shengyu Zhang, Zhou Zhao, Jiaxu Miao, Wenqiao Zhang, Wenming Tan, Jin Wang, Peng Wang, et al. 2022. End-to-End Modeling via Information Tree for One-Shot Natural Language Spatial Video Grounding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 8707–8717.
- [21] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 142–150.
- [22] Christoph Mayer and Radu Timofte. 2020. Adversarial sampling for active learning. In *The IEEE Winter Conference on Applications of Computer Vision*. 3071–3079.
- [23] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* 41, 8 (2018), 1979–1993.
- [24] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems* 31 (2018).
- [25] Ozan Sener and Silvio Savarese. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations*.
- [26] Aditya Siddhant and Zachary C Lipton. 2018. Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2904–2909.
- [27] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. 2019. Variational adversarial active learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 5972–5981.
- [28] Shuang Song, David Berthelot, and Afshin Rostamizadeh. 2019. Combining mixmatch and active learning for better accuracy with fewer labels. *arXiv preprint arXiv:1912.00594* (2019).
- [29] Toan Tran, Thanh-Toan Do, Ian Reid, and Gustavo Carneiro. 2019. Bayesian generative active deep learning. In *International Conference on Machine Learning*. PMLR, 6295–6304.
- [30] Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguishing Confusing Law Articles for Legal Judgment Prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3086–3095.
- [31] Donggeun Yoo and In So Kweon. 2019. Learning loss for active learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 93–102.
- [32] Beichen Zhang, Liang Li, Shijie Yang, Shuhui Wang, Zheng-Jun Zha, and Qingming Huang. 2020. State-Relabeling Adversarial Active Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8756–8765.
- [33] Wenqiao Zhang, Haochen Shi, Jiannan Guo, Shengyu Zhang, Qingpeng Cai, Juncheng Li, Siliang Tang, and Yueteng Zhuang. 2021. MAGIC: Multimodal relational Graph adversarial inferenCe for Diverse and Unpaired Text-based Image Captioning. *arXiv preprint arXiv:2112.06558* (2021).
- [34] Wenqiao Zhang, Haochen Shi, Siliang Tang, Jun Xiao, Qiang Yu, and Yueteng Zhuang. 2021. Consensus Graph Representation Learning for Better Grounded Image Captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 3394–3402.
- [35] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems* 28 (2015), 649–657.
- [36] Ye Zhang, Matthew Lease, and Byron Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

A APPENDIX

A.1 Algorithm Details

Algorithm 1 Inconsistency-based virtual aDvErsarial Active Learning (IDEAL)

Input: Labeled pool \mathcal{D}^l , Unlabeled pool \mathcal{D}^u , Task model’s parameters $p(y|x, \theta)$, Initial candidates set’s size \mathcal{M} , Budget \mathcal{K} , Selection epochs T

Output: Task model trained with \mathcal{D}^l and \mathcal{D}^u

```

1: Initialize  $\mathcal{D}^l$  by randomly sampling
2: for t = 0 to T-1 do
3:    $\mathbf{X}^u \leftarrow x^u$                                 ▷ Coarse-grained augmentation
4:    $\tilde{\mathbf{X}}^u \leftarrow \mathbf{X}^u$                       ▷ Fine-grained augmentation
5:    $(\tilde{\mathbf{Y}}^u, \tilde{y}^u) \leftarrow p(\tilde{\mathbf{X}}^u \cup x^u, \theta)$       ▷ Infer labels
6:    $\tilde{y}^a \leftarrow \text{ave}(\tilde{\mathbf{Y}}^u, \tilde{y}^u)$           ▷ Average labels
7:    $(y', x') \leftarrow \text{mix}((x^l, y^l), (x^u, \tilde{y}^a), (\tilde{\mathbf{X}}^u, \tilde{y}^a))$     ▷ Mix up
8:    $p(y|x, \theta) \leftarrow (y', x')$                   ▷ Train the task model
9:    $\mathbf{X}^u \leftarrow x^u$                                 ▷ Coarse-grained augmentation
10:   $(\bar{\mathbf{Y}}^u, \bar{y}^u) \leftarrow p(\bar{\mathbf{X}}^u \cup x^u, \theta)$         ▷ Infer labels
11:   $\mathbf{R}_{adv} \leftarrow (\bar{\mathbf{X}}^u, \bar{\mathbf{Y}}^u)$                 ▷ Compute perturbation
12:   $\hat{\mathbf{X}}^u \leftarrow \bar{\mathbf{X}}^u + \mathbf{R}_{adv}$           ▷ Fine-grained augmentation
13:   $\hat{\mathbf{Y}}^u \leftarrow p(\hat{\mathbf{X}}^u, \theta)$                   ▷ Infer labels
14:   $In(x^u) \leftarrow (KL(\bar{\mathbf{Y}}^u, \hat{\mathbf{Y}}^u), Var(\bar{\mathbf{Y}}^u \cup \bar{y}^u))$  ▷ Inconsistency
15:  Select top  $\mathcal{M}$  candidates with the largest  $In(x^u)$ 
16:  Compute  $En(x_i^m)$  for each  $x_i^m$  in  $\mathcal{M}$  ▷ weighted entropy
17:  Select top  $\mathcal{K}$  samples as finally selected set  $\mathcal{D}^s$  with the
     largest density-aware entropy for human annotation
18:  Update  $\mathcal{D}^l = \mathcal{D}^l \cup \mathcal{D}^s$ ,  $\mathcal{D}^u = \mathcal{D}^u / \mathcal{D}^s$ 
19: end for
20: return Task model trained with  $\mathcal{D}^l$  and  $\mathcal{D}^u$ 

```

A.2 Additional Details on the Dataset

Table 3 summarizes the statistics of datasets used in experiments. The legal dataset consists of the fact-finding portion of the public adjudication documents. The case type of the dataset is private lending disputes (PLD). We collect the bidding dataset from public bid notifications to classify enterprises’ purchase and sale documents. We leverage the bidding dataset to help customers search for desired bidding information.

Table 3: Datasets statistics.

Datasets	Type	Classes	Train	Val	Test
CIFAR-10	Image	10	45,000	5,000	10,000
CIFAR-100	Image	100	45,000	5,000	10,000
AG News	Text	4	112,000	8,000	7,600
IMDB	Text	2	20,000	5,000	25,000
Legal	Text	12	27,432	1,523	1,524
Bidding	Text	22	23,000	2,000	2,514

A.3 Additional Details on the Baselines

The baselines used in our experiments can be summarized as follows:

- EGL [36] selects samples with the largest expected gradient change, as they are expected to impose a large impact on the model. The expectation is computed over the posterior distribution of labels for the sample according to the trained model.
- Core-set [25] is a distribution-based sampling algorithm that selects a subset to cover the whole set’s distribution. The relation between the subset and the whole set is shown in [25] intuitively from the perspective of geometry, and the NP-hard problem of subset selection can be solved efficiently by a greedy algorithm.
- MC-dropout [26] selects samples with the largest uncertainty that is calculated by Monte Carlo Dropout on multiple inference cycles. It uses the max-entropy acquisition function.
- Random [9] sampling randomly selects samples and often serves as the lower bound of active learning algorithms.
- VAAL [27] learns representations of both labeled and unlabeled data in latent space. Afterwards, the VAE-GAN module uses extracted representations to estimate the label state information and pick the most informative samples through a min-max adversarial training process.
- SRAAL [32] deeply explores the label state information and maps discrete label states into a continuous variable with a state relabeling method, compared to VAAL.
- ICAL [11] chooses samples with the highest inconsistency of predictions over a set of data augmentations.
- LLAL [31] annotates samples with the largest training loss.
- REVIVAL [12] utilizes a graph to boost AL with an effective prior by inferring and passing samples’ relationships through the graph. It annotates samples near the boundary of clusters in the graph, which cloud refine the graph faster as a better prior for SSL.

A.4 Computational Costs Analysis

In this subsection, we conduct experiments on computational cost for IDEAL and its contemporary hierarchical sampling method REVIVAL [12]. The experiments are conducted on a Linux server equipped with an Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz, 512GB RAM and two NVIDIA Titan V GPU. We use the Yahoo! Answers dataset (1,400,000 training data) [3] to exhaustively test the computational cost of methods by changing the size of the unlabeled pool. We fine-tune the hyper-parameters of these two methods to reach their peak performance for a fair comparison. The results (average over five experiment trials) of actual running time and additional memory cost are shown in Table 4. We can observe that REVIVAL consumes abundant computing resources, including actual running time and additional memory space, to build a KNN graph. At the same time, IDEAL is able to deal with a huge unlabeled pool at a higher speed and memory-efficient. Thus, IDEAL is more advantageous when facing large-scale, high-dimensional datasets and deployed in practical industry scenarios.

Table 4: Computational costs comparison on the Yahoo! Answers dataset.

Data size	IDEAL		REVIVAL	
	Running time (s)	Memory cost (MB)	Running time (s)	Memory cost (MB)
20,000	14.1	35	191.2	1376
60,000	41.1	91	499.3	3782
100,000	65.2	138	923.0	6022
150,000	101.1	204	1722.9	8965
200,000	131.2	283	1538.5	11652
400,000	267.3	574	3175.1	22806
600,000	404.2	862	4913.0	34576
800,000	479.3	1069	6374.5	45927
900,000	545.7	1175	7296.8	51642