On the Learning with Augmented Class via Forests

Fan Xu , **Wuyang Chen** and **Wei Gao**
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
School of Artificial Intelligence, Nanjing University, Nanjing, China
{xuf, chenwy, gaow}@lamda.nju.edu.cn

Abstract

Decision trees and forests have achieved successes in various real applications, most working with all testing classes known in training data. In this work, we focus on learning with augmented class via forests, where an augmented class may appear in testing data yet not in training data. We incorporate information of augmented class into trees' splitting, that is, *augmented Gini impurity*, a new splitting criterion is introduced to exploit some unlabeled data from testing distribution. We then develop the Learning with Augmented Class via Forests (short for LACForest) approach, which constructs shallow forests according to the augmented Gini impurity and then splits forests with pseudo-labeled augmented instances for better performance. We also develop deep neural forests via an optimization objective based on our augmented Gini impurity, which essentially utilizes the representation power of neural networks for forests. Theoretically, we present the convergence analysis for our augmented Gini impurity, and we finally conduct experiments to evaluate our approaches. The code is available at https://github.com/nju-xuf/LACForest.

1 Introduction

How to handle distribution changes has become an important problem in a non-stationary learning environment [Zhou, 2022], and recent years have witnessed increasing attentions with various applications [Gama *et al.*, 2014; Geng *et al.*, 2021; Wang *et al.*, 2024]. This work focuses on learning with augmented class, that is, the class distribution changes and an augmented class unseen in the training data may emerge during the testing process [Da *et al.*, 2014]. Here, we take object recognition in autonomous driving for example: new objects may appear on roads yet not in the historical labeled data, and a reliable learning system should make good predictions over both known classes and augmented class.

Various approaches have been developed for learning with augmented class. Da *et al.* [2014] studied decision boundaries for augmented class under the low-density assumption. Bendale and Boult [2016] and Rudd *et al.* [2017] estimated the probability of an instance belonging to augmented class on the basis of extreme value theory. Mendes Júnior *et al.* [2017] detected augmented class by a nearest neighbor method, and Liu *et al.* [2018] presented PAC guarantees for the detection of augmented class. Some generative networks have also been applied to learn augmented class [Ge *et al.*, 2017; Neal *et al.*, 2018; Chen *et al.*, 2021]. Zhang *et al.* [2020] gave an unbiased risk estimation by exploiting unlabeled data, and Shu *et al.* [2023] generalized such approach to arbitrary loss functions.

Decision trees and forests have achieved great successes in various applications with strong generalization in handling discrete features and exploring local regions [Cutler *et al.*, 2007; Qi, 2012; Grinsztajn *et al.*, 2022; Costa and Pedreira, 2023]. Most previous studies worked on the same distribution between training and testing data. For augmented class, Mu *et al.* [2017] and Liu *et al.* [2018] only studied its detection from labeled data of known classes, whereas it remains open on how to construct trees for learning with augmented class by exploiting unlabeled data from testing data.

This work aims to incorporate some useful information of augmented class into the construction of forests, and our main contributions can be summarized as follows:

- We introduce a new splitting criterion, i.e., *augmented Gini impurity*, to incorporate information of augmented class from unlabeled data during the trees' splitting. We develop the *LACForest* approach for learning with augmented class, which constructs shallow forests by augmented Gini impurity and considers pseudo-labeled augmented instances for better performance.
- For complex data with intrinsic structures (e.g., images), we develop deep neural forests for learning with augmented class, because of their powerful representations with an end-to-end training manner. We propose a new optimization objective for deep neural forests to learn augmented class, and the basic idea is to extend our augmented Gini impurity into a differentiable form by considering the mechanism of deep neural trees.
- From a theoretical view, we present the convergence analysis of our augmented Gini impurity with respect to both decision trees and deep neural trees. We finally conduct extensive experiments to validate the effectiveness of our proposed approaches and perform some parameter influence analysis.

The rest of this work is organized as follows: Section 2 gives some preliminaries. Section 3 introduces augmented Gini impurity and LACForest approach. Section 4 presents our deep neural approach. Section 5 conducts some extensive experiments. Section 6 concludes with future works.

2 Preliminaries

Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{1, \ldots, \kappa, \kappa+1\}$ be the instance and class spaces, respectively. Suppose that $\mathcal{D}$ is an underlying (unknown) distribution over $\mathcal{X} \times \mathcal{Y}$, and denote by $\mathcal{D}_{\mathcal{X}}$ its marginal distribution over $\mathcal{X}$. We focus on learning with augmented class [Da *et al.*, 2014], where the first κ classes can be observed in training data, while the $(\kappa+1)$-th class will merely emerge in testing data, which is known as an augmented class in training phase.

For known classes and augmented class, let $\mathcal{D}_{\text{kc}}$ and $\mathcal{D}_{\text{ac}}$ be the marginal distributions over space $\mathcal{X} \times \{1, \ldots, \kappa\}$ and $\mathcal{X} \times \{\kappa+1\}$ from distribution $\mathcal{D}$, respectively. We introduce the *class shift assumption* as follows:

Assumption 1. *We say that distribution $\mathcal{D}$ and its marginal distributions $\mathcal{D}_{kc}$ and $\mathcal{D}_{ac}$ satisfy class shift assumption if*

$$\mathcal{D} = (1-\theta)\mathcal{D}_{kc} + \theta\mathcal{D}_{ac} \text{ for some constant } \theta \in (0,1) . \quad (1)$$

This assumption correlates data distributions for known classes and augmented class, which has been well-studied for learning with augmented class [Zhang *et al.*, 2020; Shu *et al.*, 2023] and open-set recognition [Scheirer *et al.*, 2013].

We focus on the semi-supervised setting for learning with augmented class [Da *et al.*, 2014; Liu *et al.*, 2018; Zhang *et al.*, 2020; Shu *et al.*, 2023], and the goal is to learn a function $f\colon \mathcal{X} \to \mathcal{Y}$ from labeled data S_l and unlabeled data S_u with

$$S_l = \{(\boldsymbol{x}_1^l, y_1), \ldots, (\boldsymbol{x}_{n_l}^l, y_{n_l})\} \text{ and } S_u = \{\boldsymbol{x}_1^u, \ldots, \boldsymbol{x}_{n_u}^u\} .$$

Here, each labeled example $(\boldsymbol{x}_i^l, y_i)$ is drawn i.i.d. from $\mathcal{D}_{\text{kc}}$, and each unlabeled instance $\boldsymbol{x}_j^u$ is sampled i.i.d. from $\mathcal{D}_{\mathcal{X}}$.

Let $\mathbb{I}[\cdot]$ be the indicator function, which returns 1 if the argument is true and 0 otherwise. For integer $k > 0$, we write $[k] = \{1, 2, \cdots, k\}$, and denote by Δ_k the k-dimensional simplex. For $z \in \mathbb{R}$, we denote by $\lfloor z \rfloor$ the largest integer no more than z, and define $(z)_+ = \max(0, z)$. Let $|A|$ be the cardinality of set A.

3 Our LACForest Approach

This section proposes the *LACForest* approach for learning with augmented class based on random forests, and the core idea is to introduce a new splitting criterion, *augmented Gini impurity*, to incorporate potential information of augmented instances during the construction of decision trees.

3.1 Augmented Gini Impurity

For instance space $\mathcal{C} \subseteq \mathcal{X}$, unlabeled data S_u and labeled data S_l, we introduce

$$S_{\mathcal{C},u} = S_u \cap \mathcal{C} \text{ and } S_{\mathcal{C},l} = \{(\boldsymbol{x}, y)\colon (\boldsymbol{x}, y) \in S_l \text{ and } \boldsymbol{x} \in \mathcal{C}\}.$$

Denote by $n_{\mathcal{C},l} = |S_{\mathcal{C},l}|$ and $n_{\mathcal{C},u} = |S_{\mathcal{C},u}|$. We could define the *augmented Gini impurity* as a splitting criterion.

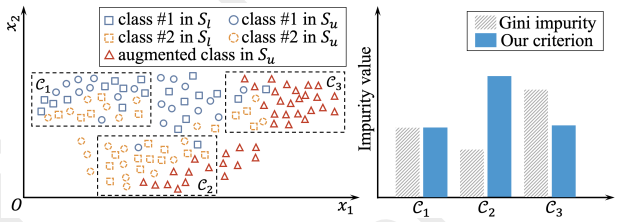

Figure 1: An intuitive illustration on the difference between our criterion and original Gini impurity over a 2-dimensional dataset, by considering augmented class.

Definition 1. *For an instance space $\mathcal{C} \subseteq \mathcal{X}$, we define the augmented Gini impurity $\mathcal{G}_{\mathcal{C}}(S_l, S_u)$ w.r.t. S_l and S_u as*

$$\mathcal{G}_{\mathcal{C}}(S_l, S_u) = 1 - \sum_{k \in [\kappa+1]} \vartheta_{\mathcal{C},k}^2(S_l, S_u) . \quad (2)$$

Here, $\vartheta_{\mathcal{C},\kappa+1}(S_l, S_u)$ is defined as

$$\vartheta_{\mathcal{C},\kappa+1}(S_l, S_u) = \mathbb{I}[n_{\mathcal{C},u} > 0]\Big(1 - \frac{(1-\theta)n_u n_{\mathcal{C},l}}{n_l \max(1, n_{\mathcal{C},u})}\Big)_+ ,$$

where θ is given in Eqn. (1)*; and for $k \in [\kappa]$, we define*

$$\vartheta_{\mathcal{C},k}(S_l, S_u) = \sum_{(\boldsymbol{x},y) \in S_l} (1 - \vartheta_{\mathcal{C},\kappa+1}(S_l, S_u)) \frac{\mathbb{I}[\boldsymbol{x} \in \mathcal{C}, y = k]}{\max(1, n_{\mathcal{C},l})} .$$

This definition essentially follows the original Gini impurity [Breiman, 1984], except with an additional term $\vartheta_{\mathcal{C},\kappa+1}(S_l, S_u)$, which aims to incorporate some information of augmented class into impurity measure.

Figure 1 presents an intuitive illustration on the difference between our augmented and original Gini impurity over a 2-dimensional synthetic dataset. It is observable that our augmented Gini impurity could properly take augmented data into consideration compared with Gini impurity.

For $\vartheta_{\mathcal{C},\kappa+1}(S_l, S_u)$, we also have

Lemma 1. *For $\delta \in (0,1)$ and instance space $\mathcal{C} \subseteq \mathcal{X}$, the following holds with probability at least $1-\delta$ over S_l and S_u*

$$\Big|\Pr_{(\boldsymbol{x},y)\in\mathcal{D}}[y = \kappa+1 | \boldsymbol{x} \in \mathcal{C}] - \vartheta_{\mathcal{C},\kappa+1}(S_l, S_u)\Big| \leq O\left(\frac{\sqrt{\ln(1/\delta)}}{\gamma}\left(\frac{1}{\sqrt{n_l}} + \frac{\sqrt{\gamma}+1}{\sqrt{n_u}}\right)\right) ,$$

if $n_{\mathcal{C},u}/n_u \geq \gamma$ for some constant $\gamma \in (0,1)$.

Let $\mathcal{L}_{\mathcal{C}}^*$ be the optimal squared loss over distribution $\mathcal{D}$ under the condition $\boldsymbol{x} \in \mathcal{C}$ as follows:

$$\mathcal{L}_{\mathcal{C}}^* = \min_{\boldsymbol{w} \in \Delta_{\kappa+1}} E_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\|\boldsymbol{w} - \tilde{y}\|_2^2 | \boldsymbol{x} \in \mathcal{C}\right] , \quad (3)$$

where $\tilde{\boldsymbol{y}} \in \mathbb{R}^{\kappa+1}$ is the one-hot encoding of $y \in [\kappa+1]$. Intuitively, $\mathcal{L}_{\mathcal{C}}^*$ shows the minimum expected squared loss under the same prediction for all instances in space $\mathcal{C}$.

Based on Lemma 1 and Assumption 1, we have

Theorem 1. *For $\delta \in (0,1)$ and instance space $\mathcal{C} \subseteq \mathcal{X}$, the following holds with probability at least $1-\delta$ over S_l and S_u*

$$|\mathcal{L}_{\mathcal{C}}^* - \mathcal{G}_{\mathcal{C}}(S_l, S_u)| \leq O\left(\frac{\kappa\sqrt{\ln(\kappa/\delta)}}{\gamma/(1+\sqrt{\gamma})}\left(\frac{1}{\sqrt{n_l}} + \frac{1}{\sqrt{n_u}}\right)\right)$$

if $n_{\mathcal{C},u}/n_u \geq \gamma$ and $n_{\mathcal{C},l}/n_l \geq \gamma$ for constant $\gamma \in (0,1)$.

This theorem shows that our augmented Gini impurity can be viewed as a good estimation of $\mathcal{L}^*_{\mathcal{C}}$ with the convergence rate $O(1/\sqrt{\min\{n_l, n_u\}})$. The detailed proof is given in our full work [Xu *et al.*, 2025], which involves the equivalence between splitting criterions and loss functions, as well as the convergence of $\vartheta_{\mathcal{C},\kappa+1}(S_l, S_u)$ in Lemma 1.

3.2 Construction of LACForest

A tree model can be constructed by partitioning instance space into disjoint rectangular nodes recursively. Given node $\mathcal{C} \subseteq \mathcal{X}$, splitting feature $j \in [d]$ and splitting point $a \in \mathbb{R}$, we denote by the left and right children as

$$\mathcal{C}^l_{j,a} = \{\boldsymbol{x} \in \mathcal{C} \colon x_j \leq a\} \text{ and } \mathcal{C}^r_{j,a} = \{\boldsymbol{x} \in \mathcal{C} \colon x_j > a\} ,$$

respectively. We then solve the optimal splitting feature j^* and splitting point a^* from the following optimization

$$(j^*, a^*) \in \arg\max_{j,a} \{R_{\mathcal{C}}(S_l, S_u, j, a)\} , \tag{4}$$

where the reduction of augmented Gini impurity is given by

$$R_{\mathcal{C}}(S_l, S_u, j, a) = \mathcal{G}_{\mathcal{C}}(S_l, S_u) - \frac{n_{\mathcal{C}^l_{j,a},u}}{n_{\mathcal{C},u}} \mathcal{G}_{\mathcal{C}^l_{j,a}}(S_l, S_u) - \frac{n_{\mathcal{C}^r_{j,a},u}}{n_{\mathcal{C},u}} \mathcal{G}_{\mathcal{C}^r_{j,a}}(S_l, S_u) .$$

From Theorem 1, we can distinguish augmented class from known classes in node $\mathcal{C}$ by solving Eqn. (4), as it converges to optimizing the squared loss as training data increases. From a theoretical view, we require sufficient training data in each node $\mathcal{C}$, i.e., $n_{\mathcal{C},l} \geq \gamma n_l$ and $n_{\mathcal{C},u} \geq \gamma n_u$ for some $\gamma \in (0, 1)$ in Theorem 1; therefore, we should construct relatively shallow tree models.

On the other hand, we should take deeper tree models for better performance, and it is well-known that better predictions could be obtained by deepening tree models sufficiently [Quinlan, 1986; Hastie *et al.*, 2009]. Hence, our algorithm includes two main steps as follows:

Step-I: Exploration of Augmented Instances

We preliminarily construct m random trees $\mathcal{T}'_1, \cdots, \mathcal{T}'_m$ for the exploration of augmented instances from Eqn. (4). For simplicity, we present the detailed construction of tree $\mathcal{T}'_1$, and consider other trees similarly. We initialize $\mathcal{T}'_1$ with one root node of instance space $\mathcal{X}$, and recursively repeat a procedure for each leaf node as follows:

- Select a τ-subset $\mathcal{S}$ from d available features randomly without replacement;
- Solve the optimal splitting feature $j^* \in \mathcal{S}$ and point a^* according to Eqn. (4) under the constraints

$$\min(n_{\mathcal{C}^l_{j,a},l}, n_{\mathcal{C}^r_{j,a},l}) \geq \gamma n_l , \tag{5}$$

$$\min(n_{\mathcal{C}^l_{j,a},u}, n_{\mathcal{C}^r_{j,a},u}) \geq \gamma n_u . \tag{6}$$

  This ensures sufficient instances in each splitting node from Theorem 1;
- Split current node into left and right children via the optimal splitting feature j^* and position a^*.

Algorithm 1 Our LACForest approach

Input: Labeled data S_l, unlabeled data S_u, proportion θ, hyper-parameters m, τ, γ

% Step-I: Exploration of Augmented Instances.
for $i \in \{1, \cdots, m\}$ **do**
  Grow random tree $\mathcal{T}'_i$ based on S_l and S_u according. to the reduction of augmented Gini impurity
end for
% Step-II: Improvement of Prediction Performance
Calculate $\mathcal{LF}'_{\kappa+1}(\boldsymbol{x})$ for $\boldsymbol{x} \in S_u$ by Eqn. (7).
Label top $n_p = \lfloor \theta n_u \rfloor$ instances in S_u of the highest $\mathcal{LF}'_{\kappa+1}(\boldsymbol{x})$ and obtain $\tilde{S}$.
for $i \in \{1, \cdots, m\}$ **do**
  Split random tree $\mathcal{T}'_i$ based on $S_l \cup \tilde{S}$ according to the reduction of Gini impurity and obtain $\mathcal{T}_i$.
end for
return $\mathcal{LF}(\boldsymbol{x}) = \arg\max_{k \in [\kappa+1]} \sum_{i \in [m]} \mathcal{T}_{i,k}(\boldsymbol{x})$.

The above procedure is stopped if there is no feasible solution for Eqn. (4) under constraints in Eqns. (5) and (6).

Given random tree $\mathcal{T}'_1$ with t' leaves $\mathcal{C}'_1, \mathcal{C}'_2, \cdots, \mathcal{C}'_{t'}$, we present the probability of augmented class for $\boldsymbol{x} \in \mathcal{X}$ as

$$\mathcal{T}'_{1,\kappa+1}(\boldsymbol{x}) = \sum_{j \in [t']} \vartheta_{\mathcal{C}'_j,\kappa+1}(S_l, S_u) \mathbb{I}[\boldsymbol{x} \in \mathcal{C}'_j] ,$$

where $\vartheta_{\mathcal{C}'_j,\kappa+1}(S_l, S_u)$ is an unbiased estimator for the proportion of augmented instances in $\mathcal{C}'_j$, given by Definition 1.

Step-II: Improvement of Prediction Performance

Given m trees $\mathcal{T}'_1, \cdots, \mathcal{T}'_m$, we calculate the average augmented score of instance $\boldsymbol{x} \in \mathcal{X}$ by

$$\mathcal{LF}'_{\kappa+1}(\boldsymbol{x}) = \frac{1}{m} \sum_{i \in [m]} \mathcal{T}'_{i,\kappa+1}(\boldsymbol{x}) . \tag{7}$$

We then select $\lfloor \theta n_u \rfloor$ instances in S_u of the highest average augmented scores, and label them with pseudo-labels of augmented class. Without loss of generality, we denote by

$$\tilde{S} = \{(\boldsymbol{x}^u_1, \kappa+1), (\boldsymbol{x}^u_2, \kappa+1), \cdots, (\boldsymbol{x}^u_{\lfloor \theta n_u \rfloor}, \kappa+1)\} .$$

Intuitively, $\tilde{S}$ retains the most likely augmented instances for further partition, and similar approaches have been studied by [Liu *et al.*, 2002; Tanha *et al.*, 2017].

We further partition $\mathcal{T}'_1, \cdots, \mathcal{T}'_m$ for better predictions based on $\tilde{S} \cup S_l$. We also repeat the following procedure recursively for each leaf $\mathcal{C}$ of $\mathcal{T}'_1$:

- Select a τ-subset $\mathcal{S}$ from d available features randomly without replacement;
- Solve the optimal splitting feature $j^* \in \mathcal{S}$ and position a^* w.r.t. Gini impurity and instances in $\tilde{S} \cup S_l$;
- Split the current node into left and right children via the optimal splitting feature j^* and position a^*.

The above procedure is stopped if all instances have the same label in a leaf, and get the final random tree $\mathcal{T}_1$. Similarly, we repeat the above procedure for $\mathcal{T}'_2, \cdots, \mathcal{T}'_m$.

Given m random trees $\mathcal{T}_1, \cdots, \mathcal{T}_m$, we predict the final label of instance $\boldsymbol{x} \in \mathcal{X}$ as

$$\mathcal{LF}(\boldsymbol{x}) = \arg\max_{k \in [\kappa+1]} \sum_{i \in [m]} \mathcal{T}_{i,k}(\boldsymbol{x}) \ ,$$

where $\mathcal{T}_{i,k}(\boldsymbol{x})$ is the probability of the k-th category of tree $\mathcal{T}_i$ with leaves $\mathcal{C}_{i,1}, \mathcal{C}_{i,2}, \cdots, \mathcal{C}_{i,t_i}$, i.e.,

$$\mathcal{T}_{i,k}(\boldsymbol{x}) = \sum_{j \in [t_i]} \frac{\mathbb{I}[\boldsymbol{x} \in \mathcal{C}_{i,j}] \sum_{(\boldsymbol{x}',y) \in S_l \cup \tilde{S}} \mathbb{I}[\boldsymbol{x}' \in \mathcal{C}_{i,j}, y = k]}{\max(1, \sum_{(\boldsymbol{x}',y) \in S_l \cup \tilde{S}} \mathbb{I}[\boldsymbol{x}' \in \mathcal{C}_{i,j}])} \ .$$

Algorithm 1 gives detailed descriptions of our LACForest approach. Notice that our method requires the proportion θ of augmented class, which is usually unknown in practice. One feasible solution is to consider the kernel mean embedding method from [Ramaswamy *et al.*, 2016], and this is similar to previous studies on learning with augmented class [Zhang *et al.*, 2020; Shu *et al.*, 2023].

4 Deep Neural LACForest

This section explores deep neural forests via augmented Gini impurity for learning with augmented class, which presents powerful representations with an end-to-end training manner for complex data such as images and texts. For neural forests, another advantage is to take soft splits rather than hard splits of decision trees, which could yield smoother decision boundaries and avoid overfitting [Kontschieder *et al.*, 2015]. In this work ,we take one step on learning with augmented class via deep neural forests, while previous works mostly focused on all known classes [Kontschieder *et al.*, 2015; Tanno *et al.*, 2019; Ji *et al.*, 2020; Li *et al.*, 2024].

4.1 Augmented Gini Impurity for Neural Trees

A deep neural forest is constructed with a DNN encoder h and m differentiable neural trees $\mathcal{DT}_1, \mathcal{DT}_2, \cdots, \mathcal{DT}_m$. The key point is how to learn differentiable neural trees by incorporating information of augmented class, and our idea is to consider augmented Gini impurity as an optimization objective for neural trees.

We take neural tree $\mathcal{DT}_1$ as an example, and it is essentially an l-layer complete binary tree with $l \geq 2$. We denote by $\{\mathcal{B}_1, \cdots, \mathcal{B}_t\}$ and $\{\mathcal{B}_{t+1}, \cdots, \mathcal{B}_{2t+2}\}$ all internal and leaf nodes of $\mathcal{DT}_1$, respectively, where $t = 2^{l-1} - 1$.

For every internal node $\mathcal{B}_i$ ($i \in [t]$), we associate with a function $f_i \colon \mathcal{X} \to [0,1]$ to represent the probability of being assigned to the left child, and hence the probability of the right child is $1 - f_i(\boldsymbol{x})$. Given encoder h, we define

$$f_i(\boldsymbol{x}) = \text{sigmoid}(\boldsymbol{w}_i^T h(\boldsymbol{x}) + b_i) \ ,$$

where $\text{sigmoid}(z) = (1+\exp(-z))^{-1}$ and $\boldsymbol{w}_i, b_i$ are learned parameters, as in [Kontschieder *et al.*, 2015].

For every leaf node $\mathcal{B}_i$ ($i \in [2t+2] \setminus [t]$), let $\mathcal{I}(\boldsymbol{x} \to \mathcal{B}_i)$ be the event that $\boldsymbol{x}$ is assigned to leaf $\mathcal{B}_i$, and denote by

$$\begin{aligned} \mu_{\mathcal{B}_i}(\boldsymbol{x}) &= \Pr\left[\mathcal{I}(\boldsymbol{x} \to \mathcal{B}_i)\right] \\ &= \prod_{j \in [t]} f_j(\boldsymbol{x})^{\mathbb{I}[\mathcal{B}_i \swarrow \mathcal{B}_j]} (1 - f_j(\boldsymbol{x}))^{\mathbb{I}[\mathcal{B}_j \searrow \mathcal{B}_i]} \ , \end{aligned}$$

Algorithm 2 Deep Neural LACForest

Input: Labeled data S_l, unlabeled data S_u, proportion θ, hyper-parameters $m, l, T, \lambda_{\text{ce}}$

```
Randomly initialize h and DT_1, ..., DT_m.
for j ∈ {1, ..., T} do
    Break S_l and S_u into random mini-batches.
    for mini-batches S_l' and S_u' from S_l and S_u do
        Compute L by Eqn. (10) and compute gradients.
        Update h and DT_1, ..., DT_m by SGD.
    end for
end for
return DF(x) = argmax_{k∈[κ+1]} Σ_{i∈[m]} DT_{i,k}(x).
```

where $\mathcal{B}_i \swarrow \mathcal{B}_j$ and $\mathcal{B}_j \searrow \mathcal{B}_i$ represent that $\mathcal{B}_i$ belongs to the left and right subtrees of $\mathcal{B}_j$, respectively. We also denote by

$$n_{\mathcal{B}_i,u} = \sum_{\boldsymbol{x} \in S_u} \mu_{\mathcal{B}_i}(\boldsymbol{x}) \quad \text{and} \quad n_{\mathcal{B}_i,l} = \sum_{(\boldsymbol{x},y) \in S_l} \mu_{\mathcal{B}_i}(\boldsymbol{x}) \ ,$$

for unlabeled data S_u and labeled data S_l, respectively.

Definition 2. *For $i \in [2t+2] \setminus [t]$, we define the augmented Gini impurity w.r.t. S_l and S_u as*

$$\mathcal{G}_{\mathcal{B}_i}(S_l, S_u) = 1 - \sum_{k \in [\kappa+1]} \vartheta^2_{\mathcal{B}_i,k}(S_l, S_u) \ , \tag{8}$$

where $\vartheta_{\mathcal{B}_i,\kappa+1}(S_l, S_u)$ is defined as

$$\vartheta_{\mathcal{B}_i,\kappa+1}(S_l, S_u) = \left(1 - \frac{(1-\theta) n_u n_{\mathcal{B}_i,l}}{n_l n_{\mathcal{B}_i,u}}\right)_+ \ ,$$

and for $k \in [\kappa]$, $\vartheta_{\mathcal{B}_i,k}(S_l, S_u)$ is given by

$$(1 - \vartheta_{\mathcal{B}_i,\kappa+1}(S_l, S_u)) \sum_{(\boldsymbol{x},y) \in S_l} \frac{\mu_{\mathcal{B}_i}(\boldsymbol{x})}{n_{\mathcal{B}_i,l}} \mathbb{I}[y = k] \ .$$

This definition follows the augmented Gini impurity of decision trees in Definition 1, whereas deep neural trees consider probabilistic assignment of instances to nodes while the latter focuses on deterministic assignment. Essentially, $\vartheta_{\mathcal{B}_i,k}(S_l, S_u)$ estimates the probability of the k-th class under the condition $\mathcal{I}(\boldsymbol{x} \to \mathcal{B}_i)$. Based on Assumption 1, we have

Lemma 2. *For $k \in [\kappa+1]$, $\delta \in (0,1)$ and $i \in [2t+2] \setminus [t]$, we have, with probability at least $1 - \delta$ over S_l and S_u,*

$$\begin{aligned} &\left| \Pr_{(\boldsymbol{x},y) \sim \mathcal{D}}[y = k | \mathcal{I}(\boldsymbol{x} \to \mathcal{B}_i)] - \vartheta_{\mathcal{B}_i,k}(S_l, S_u) \right| \\ &\leq \ O\left(\frac{\sqrt{\ln(1/\delta)}}{\gamma^2/(\gamma+1)} \left(\frac{1}{\sqrt{n_l}} + \frac{1}{\sqrt{n_u}} \right) \right) \ , \end{aligned}$$

if $\mathbb{E}_{\mathcal{D}}[\mu_{\mathcal{B}_i}(\boldsymbol{x})] \geq \gamma$, $\mathbb{E}_{\mathcal{D}_{kc}}[\mu_{\mathcal{B}_i}(\boldsymbol{x})] \geq \gamma$, $n_{\mathcal{B}_i,u} \geq \gamma n_u$ and $n_{\mathcal{B}_i,l} \geq \gamma n_l$ for some constant $\gamma \in (0,1)$.

Denote by $\mathcal{L}^*_{\mathcal{B}_i}$ the optimal squared loss under $\mathcal{I}(\boldsymbol{x} \to \mathcal{B}_i)$:

$$\mathcal{L}^*_{\mathcal{B}_i} = \min_{\boldsymbol{w} \in \Delta_{\kappa+1}} E_{(\boldsymbol{x},y) \sim \mathcal{D}} \left[\|\boldsymbol{w} - \tilde{\boldsymbol{y}}\|_2^2 | \mathcal{I}(\boldsymbol{x} \to \mathcal{B}_i) \right] \ ,$$

where $\tilde{\boldsymbol{y}} \in \mathbb{R}^{\kappa+1}$ is the one-hot encoding of $y \in [\kappa+1]$. Based on Assumption 1 and Lemma 2, we have

Datasets	#inst	#feat	#class	Datasets	#inst	#feat	#class	Datasets	#inst	#feat	#class	Datasets	#inst	#feat	#class
segment	2,310	19	7	mfcc	7,195	22	4	drybean	13,661	16	7	mnist	70,000	784	10
texture	5,500	40	11	usps	9,298	256	10	letter	20,000	16	26	fmnist	70,000	784	10
optdigits	5,620	64	10	har	10,299	562	6	shuttle	58,000	9	7	kuzushiji	70,000	784	10
satimage	6,435	36	6	mapping	10,845	28	6	drive	58,509	48	11	svhn	99,289	3072	10
landset	6,435	73	6	pendigits	10,992	16	10	senseveh	61,581	100	3	cifar10	60,000	3072	10

Table 1: Details of datasets.

Theorem 2. *For $\delta \in (0,1)$ and $i \in [2t+2]\setminus[t]$, the following holds with probability at least $1-\delta$ over S_l and S_u*

$$\left|\mathcal{L}^*_{\mathcal{B}_i} - \mathcal{G}_{\mathcal{B}_i}(S_l, S_u)\right| \leq O\left(\frac{\kappa\sqrt{\ln(\kappa/\delta)}}{\gamma^2/(\gamma+1)}\left(\frac{1}{\sqrt{n_l}} + \frac{1}{\sqrt{n_u}}\right)\right),$$

if $\mathbb{E}_{\mathcal{D}}[\mu_{\mathcal{B}_i}(\boldsymbol{x})] \geq \gamma$, $\mathbb{E}_{\mathcal{D}_k}[\mu_{\mathcal{B}_i}(\boldsymbol{x})] \geq \gamma$, $n_{\mathcal{B}_i,u} \geq \gamma n_u$ and $n_{\mathcal{B}_i,l} \geq \gamma n_l$ for some constant $\gamma \in (0,1)$.

This theorem shows that $\mathcal{G}_{\mathcal{B}_i}(S_l, S_u)$ could be seen as an unbiased estimation of $\mathcal{L}^*_{\mathcal{B}_i}$. The detailed proof is presented in [Xu *et al.*, 2025], which is similar to that of Theorem 1, but takes different analysis on the convergence of $\vartheta_{\mathcal{B}_i,k}(S_l, S_u)$ by considering the probabilistic assignment of neural trees.

4.2 Deep Neural LACForest

In the training phase, we randomly initialize encoder h and neural trees $\mathcal{DT}_1, \cdots, \mathcal{DT}_m$, and update the encoder and neural trees iteratively. In each iteration, we receive two mini-batches of labeled data S'_l and unlabeled data S'_u sampled randomly from S_l and S_u, respectively. We introduce

$$\mathcal{L}_{\text{ag}}(\mathcal{DT}_i) = \sum_{j\in[2t+2]\setminus[t]} \omega_{i,j}(S'_u)\mathcal{G}_{\mathcal{B}_{i,j}}(S'_l, S'_u)\,, \qquad (9)$$

where $\{\mathcal{B}_{i,j}\}_{j\in[2t+2]\setminus[t]}$ denote all leaf nodes of $\mathcal{DT}_i$, and $\omega_{i,j}(S'_u)$ is the proportion of unlabeled data in $\mathcal{B}_{i,j}$, i.e.,

$$\omega_{i,j}(S'_u) = \frac{1}{|S'_u|}\sum_{\boldsymbol{x}\in S'_u} \mu_{\mathcal{B}_{i,j}}(\boldsymbol{x})\,.$$

The augmented Gini loss in Eqn. (9) is a weighted average augmented Gini impurity for all leaf nodes of $\mathcal{DT}_i$.

For better representation, we introduce an auxiliary linear predictor $g\colon \mathbb{R}^{d'} \to \Delta_\kappa$ as done in [Yan *et al.*, 2021], and take the cross entropy loss on known classes w.r.t. S'_l as

$$\mathcal{L}_{\text{ce}} = -\frac{1}{|S'_l|}\sum_{(\boldsymbol{x},y)\in S'_l}\sum_{t\in[\kappa]} \tilde{y}_t \log\left(g_t\left(h\left(\boldsymbol{x}\right)\right)\right)\,.$$

We finally get the optimization objective as follows:

$$\mathcal{L} = \sum_{i\in[m]} \frac{\mathcal{L}_{\text{ag}}(\mathcal{DT}_i)}{m} + \lambda_{\text{ce}}\mathcal{L}_{\text{ce}}\,, \qquad (10)$$

where $\lambda_{\text{ce}} > 0$ is a trade-off parameter. Intuitively, the cross entropy loss is beneficial to learning a basic representation of target data, and the augmented gini loss could improve the ability of neural trees to distinguish augmented class from known classes, as shown by Theorem 2.

Notice that the optimization objective $\mathcal{L}$ in Eqn. (10) is differentiable w.r.t. the parameters of feature encoder h and neural trees $\mathcal{DT}_1, \cdots, \mathcal{DT}_m$, and we could update the entire model with stochastic gradient descent directly.

Given feature encoder h and neural trees $\mathcal{DT}_1, \cdots \mathcal{DT}_m$, we could present the prediction for instance $\boldsymbol{x} \in \mathcal{X}$ as

$$\mathcal{DF}(\boldsymbol{x}) = \operatorname*{arg\,max}_{k\in[\kappa+1]} \sum_{i\in[m]} \mathcal{DT}_{i,k}(\boldsymbol{x})\,,$$

where $\mathcal{DT}_{i,k}(\boldsymbol{x})$ shows the probability of the k-th class w.r.t. neural tree $\mathcal{DT}_i$, i.e.,

$$\mathcal{DT}_{i,k}(\boldsymbol{x}) = \sum_{j\in[2t]\setminus[t-1]} \mu_{\mathcal{B}_{i,j}}(\boldsymbol{x})\vartheta_{\mathcal{B}_{i,j},k}(S_l, S_u)\,,$$

with $\vartheta_{\mathcal{B}_{i,j},k}(S_l, S_u)$ given by Definition 2.

Algorithm 2 presents the detailed description of our deep neural LACForest approach. In experiments, we should take relatively large batch sizes for mini-batches S'_l and S'_u, since $\mathcal{G}_{\mathcal{B}_{i,j}}(S'_l, S'_u)$ in Eqn. (9) converges to the optimal squared loss in the rate of $O(1/\sqrt{\min\{|S'_l|, |S'_u|\}})$ from Theorem 2.

4.3 Related Works

Zhou and Chen [2002] introduced the problem of *class-incremental learning* via a few labeled augmented instances. Fink *et al.* [2006] learned multiple binary classifiers for this problem and Topalis and Polikar [2008] considered voting classifiers. Recent years have witnessed increasing attention on the design of practical algorithms for this problem [Li and Hoiem, 2017; Rebuffi *et al.*, 2017; Yan *et al.*, 2021; Zou *et al.*, 2022; Zhou *et al.*, 2024]. These methods can not be applied to our learning scenario directly, since we have no access to any labeled augmented instances.

Another relevant problem is *open-set recognition* in computer vision. Scheirer *et al.* [2013] introduced open space risk to penalize predictions outside the support of training data. Along this line, various approaches have been developed based on open space risk [Scheirer *et al.*, 2014], extreme value theory [Bendale and Boult, 2016; Rudd *et al.*, 2017], nearest neighbors [Mendes Júnior *et al.*, 2017] and generative neural networks [Ge *et al.*, 2017; Neal *et al.*, 2018; Chen *et al.*, 2021]. Those studies are strongly based on some geometric assumptions.

5 Experiments

We conduct experiments on 15 benchmark datasets and 5 image datasets, and the details are summarized in Table 1. Most datasets have been well-studied in previous works on learning with augmented class.

Datasets	Our LACForest	GLAC	EULAC	EVM	PAC-iForest	OSNN	LACU-SVM	OVR-SVM
segment	.9436±.0186	.8838±.0390•	.9256±.0276•	.8612±.0746•	.6678±.0763•	.5467±.0384•	.4918±.0796•	.6109±.0901•
texture	.9151±.0176	.9130±.0328	.9138±.0208	.8724±.0270•	.7107±.0518•	.5956±.0405•	.5829±.0478•	.6078±.0587•
optdigits	.9260±.0203	.8842±.0384•	.9243±.0186	.9069±.0234•	.7202±.0415•	.6572±.0282•	.7385±.0326•	.7547±.0552•
satimage	.8791±.0322	.8215±.0526•	.8644±.0371•	.7238±.0664•	.7460±.0566•	.5101±.0380•	.6324±.0165•	.5228±.0568•
landset	.9243±.0206	.8627±.0262•	.8857±.0236•	.8120±.0464•	.7572±.0671•	.5302±.0257•	.6289±.0124•	.5366±.0316•
mfcc	.9418±.0144	.8937±.0194•	.9506±.0137○	.8669±.0888•	.7901±.0710•	.5329±.0311•	.7751±.0362•	.6428±.0215•
usps	.8931±.0198	.8645±.0382•	.8955±.0250	.7959±.0697•	.5695±.0894•	.6413±.0317•	.7642±.0379•	.7488±.0464•
har	.9020±.0319	.8772±.0460•	.8922±.0269•	.5016±.0558•	.5963±.0699•	.5048±.0386•	.5570±.0408•	.5046±.0251•
mapping	.8612±.0199	.7964±.0350•	.8515±.0309•	.7111±.1320•	.6712±.1046•	.6218±.0605•	.6538±.0559•	.5210±.0596•
pendigits	.9281±.0215	.8872±.0241•	.9276±.0245	.9266±.0330	.7709±.0742•	.5822±.0243•	.7352±.0513•	.6424±.0469•
drybean	.8932±.0208	.9026±.0528○	.9038±.0202○	.7834±.0428•	.7555±.0693•	.5661±.0580•	.6384±.0397•	.5471±.0478•
letter	.7402±.0335	.6331±.0394•	.6057±.0387•	.7004±.0460•	.4875±.0373•	.5980±.0231•	.5547±.0538•	.6238±.0399•
shuttle	.9750±.0197	.9444±.0506•	.9770±.0116	.6245±.0347•	.6042±.0326•	.5464±.0649•	.6734±.0218•	.4862±.0249•
drive	.8529±.0532	.6114±.0824•	.7844±.0460•	.7708±.0476•	.4296±.0578•	.5474±.0246•	.6032±.0783•	.5287±.1040•
senseveh	.7977±.0295	.7545±.0406•	.7925±.0196•	.5685±.0572•	.5580±.1089•	.5331±.0499•	.5973±.0425•	.5726±.0688•
mnist	.8540±.0299	.7947±.0408•	.8390±.0380•	.6181±.0475•	.5369±.0827•	.6504±.0386•	.6481±.0496•	.7234±.0505•
fmnist	.8008±.0269	.7592±.0343•	.7672±.0382•	.6508±.0368•	.6090±.0577•	.5152±.0534•	.5674±.0629•	.5499±.0727•
average	.8840±.0591	.8285±.0916	.8648±.0862	.7468±.1204	.6459±.1042	.5694±.0486	.6378±.0774	.5955±.0822
win/tie/loss		**15/1/1**	**10/5/2**	**16/1/0**	**17/0/0**	**17/0/0**	**17/0/0**	**17/0/0**

Table 2: Experimental comparisons of accuracy (mean±std) for compared methods, and •/○ indicates that our approach is significantly better/worse than the corresponding method (paired t-test at 95% significance level).

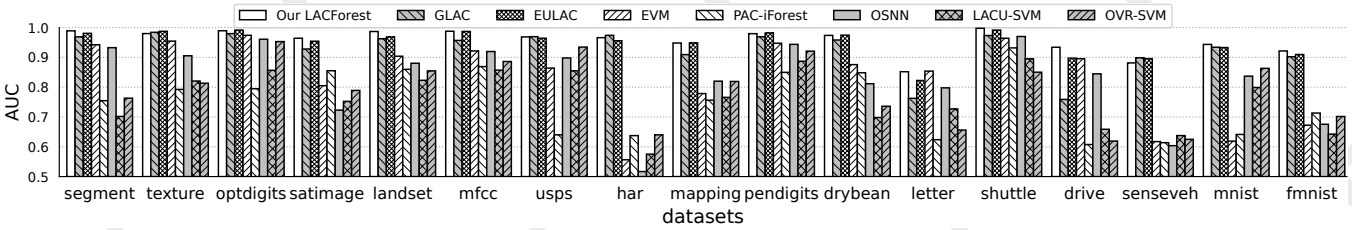

Figure 2: Experimental comparisons of AUC on the detection of augmented class.

5.1 Evaluation of Our LACForest Approach

For our LACForest approach, we compare with seven state-of-the-art approaches: GLAC [Shu *et al.*, 2023], EULAC [Zhang *et al.*, 2020], PAC-iForest [Liu *et al.*, 2018], EVM [Rudd *et al.*, 2017], OSNN [Mendes Júnior *et al.*, 2017], LACU-SVM [Da *et al.*, 2014] and OVR-SVM [Rifkin and Klautau, 2004]. More details on these approaches could be found in our full work [Xu *et al.*, 2025].

For each dataset, we randomly select half of classes as the augmented class with the rest as known classes, following [Zhang *et al.*, 2020]. We then randomly sample 500 examples of known classes as labeled data S_l, and 1000 instances as unlabeled data S_u and 100 instances as testing data. We take $\theta = 0.5$ in Eqn. (1), and more experimental settings could be found in [Xu *et al.*, 2025]. The performance is evaluated by 10 trials of random selections of augmented class, and with 10 times of random data sampling. The average test accuracies are obtained over these 100 runs, as shown in Table 2.

It is clear that our LACForest method achieves significantly better performance than previous EVM, OSNN, PAC-iForest and OVR-SVM, as our LACForest wins in most times and never loses. This is because those methods mainly focus on labeled data from known classes, but without exploring information from unlabeled data.

Our LACForest also outperforms LACU-SVM, which is heavily dependent on the low-separation assumption over data. In comparison to GLAC and EULAC, our LACForest achieves better and comparable performance in most times, except for datasets mfcc and drybean, partially because of class imbalance in these datasets, which makes it difficult to accurately estimate proportions of augmented class in nodes.

We also take the average AUC to show the performance on the detection of augmented class in Figure 2. It is obvious that our LACForest takes better and comparable performance on the detection of augmented class over most datasets, since our method could explore augmented class effectively in each leaf node, as shown by Lemma 1.

5.2 Evaluation of Our Deep Neural LACForest

For our deep neural LACForest, we compare with seven deep learning methods for augmented class: Deep-GLAC [Shu *et al.*, 2023], ARPL [Chen *et al.*, 2021], Deep-EULAC [Zhang *et al.*, 2020], OSRCI [Neal *et al.*, 2018], G-Openmax [Ge *et al.*, 2017], Openmax [Bendale and Boult, 2016] and Softmax-T [Hendrycks and Gimpel, 2016]. More details on these approaches are presented in our full work [Xu *et al.*, 2025].

We take a three-layer convolutional neural network as the backbone neural network on mnist, fmnist and kuzushiji, and consider VGG16 [Simonyan and Zisserman, 2015] on svhn and cifar10, as done in [Shu *et al.*, 2023]. We randomly select four classes as the augmented class and take the rest as known classes, and set $\theta = 0.4$ similarly to [Shu *et al.*, 2023].

Datasets	Our approach	Deep-GLAC	Deep-EULAC	ARPL	G-Openmax	OSRCI	Openmax	Softmax-T
mnist	**.9844±.0021**	.9778±.0039	.9596±.0033	.9304±.0203	.8934±.0064	.9114±.0047	.8876±.0042	.8834±.0029
fmnist	**.9024±.0114**	.9010±.0148	.8464±.0085	.7682±.0102	.6820±.0148	.6912±.0070	.6672±.0139	.5878±.0054
kuzushiji	**.9636±.0044**	.9516±.0032	.8872±.0032	.9002±.0143	.8570±.0055	.8602±.0046	.8426±.0070	.8282±.0069
svhn	**.9238±.0134**	.8926±.0168	.8330±.0087	.8060±.0069	.7868±.0161	.7912±.0072	.7888±.0057	.7252±.0047
cifar10	**.8008±.0331**	.7840±.0426	.7168±.0231	.7208±.0032	.6560±.0125	.6788±.0084	.6612±.0293	.6350±.0203
average	**.9150±.0639**	.9016±.0669	.8486±.0793	.8251±.0790	.7750±.0933	.7866±.0914	.7695±.0915	.7319±.1117

Table 3: Experimental comparisons of accuracy (mean±std) over 5 image datasets, and the best performance is highlighted in bold.

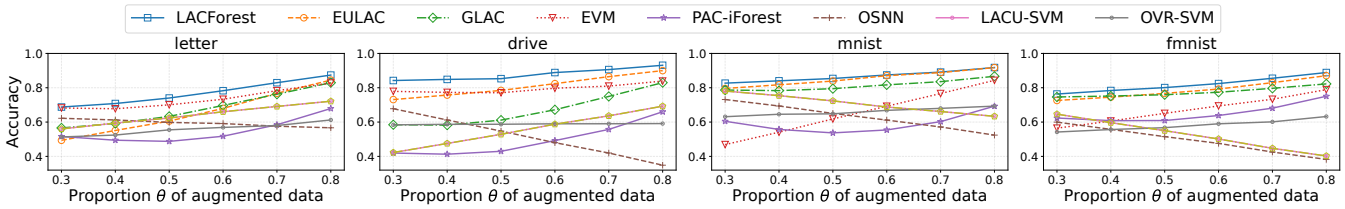

Figure 3: Evaluations of LACForest over different proportions of augmented data. The larger the curve, the better the performance.

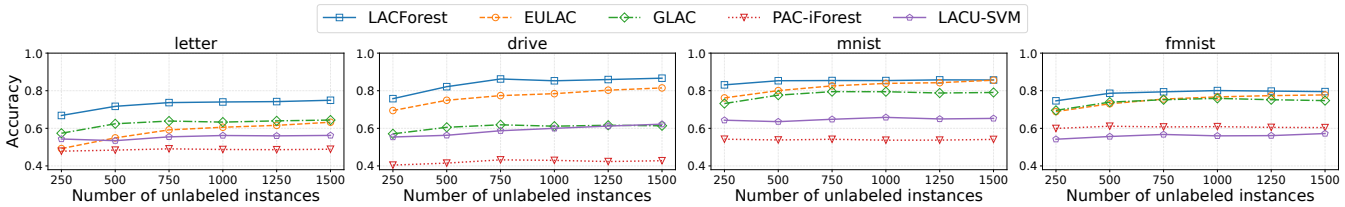

Figure 4: Evaluations of LACForest under different sizes of unlabeled data S_u. The larger the curve, the better the performance.

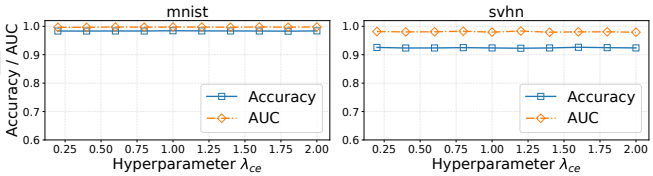

Figure 5: Influence of parameter λ_{ce}.

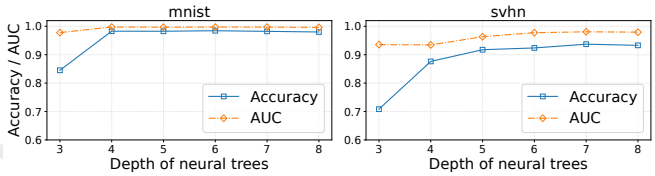

Figure 6: Influence of neural trees' depth.

We take average test accuracies over 5 random selections of augmented class as our performance measure, and the experimental comparisons are shown in Table 3. It is clear that our approach takes better performance than ARPL, G-Openmax, OSRCI, Openmax and Softmax-T, due to some additional geometric assumptions over those methods. For Deep-GLAC and Deep-EULAC, our method achieves better and comparable performance in most times, and an intuitive explanation is that our approach could effectively explore augmented class from each local region during tree partitions.

Parameter Influence. We analyze the influence of various parameters on several datasets, and the trends are similar for other datasets. Figure 3 shows that our LACForest gets better performance under different proportions $\theta \in [0.3, 0.8]$ for augmented data. Figure 4 shows the performance with different sizes of unlabeled data, where our LACForest achieves better and stable performance with the increase of unlabeled data, in consistency with Theorem 1. For deep neural LACForest, Figure 5 shows that our approach is insensitive to parameter λ_{ce} and generally works well for $\lambda_{\text{ce}} \in [0.2, 2]$. Figure 6 shows the influence of the depth of neural trees, and our method takes stable results when the tree depth $l \geq 5$.

6 Conclusion

This work studies learning with augmented class via forests, where an augmented class may appear in testing data yet not in training data. We introduce the augmented Gini impurity as a new splitting criterion by incorporating information of augmented class during tree construction. We develop the approach on *Learning with Augmented Class via Forests*, which constructs shallow forests based on augmented Gini impurity and further splits forests with pseudo-labeled augmented instances. We also explore deep neural forests with a new optimization objective via our augmented Gini impurity. We validate the effectiveness of our methods both empirically and theoretically. An interesting future direction is to exploit our methods under other learning settings such as streaming datasets and multiple augmented classes.

Acknowledgments

The authors want to thank the reviewers for their helpful comments and suggestions. This research was supported by National Key R&D Program of China (2021ZD0112802) and NSFC (62376119). Wei Gao is the corresponding author.

References

[Bendale and Boult, 2016] A. Bendale and T. E. Boult. Towards open set deep networks. In *Proceedings of the 29th IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016.

[Breiman, 1984] L. Breiman. *Classification and Regression Trees*. CRC Press, 1984.

[Chen *et al.*, 2021] G. Chen, P. Peng, X. Wang, and Y. Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8065–8081, 2021.

[Costa and Pedreira, 2023] V.-G. Costa and C.-E. Pedreira. Recent advances in decision trees: An updated survey. *Artificial Intelligence Review*, 56(5):4765–4800, 2023.

[Cutler *et al.*, 2007] D.-R. Cutler, T.-C. Edwards Jr., K.-H. Beard, A. Cutler, K.-T. Hess, J. Gibson, and J.-J. Lawler. Random forests for classification in ecology. *Ecology*, 88(11):2783–2792, 2007.

[Da *et al.*, 2014] Q. Da, Y. Yu, and Z.-H. Zhou. Learning with augmented class by exploiting unlabeled data. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1760–1766, 2014.

[Fink *et al.*, 2006] M. Fink, S. Shalev-Shwartz, Y. Singer, and S. Ullman. Online multiclass learning by interclass hypothesis sharing. In *Proceedings of the 23rd International Conference on Machine learning*, pages 313–320, 2006.

[Gama *et al.*, 2014] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014.

[Ge *et al.*, 2017] Z. Ge, S. Demyanov, and R. Garnavi. Generative openmax for multi-class open set classification. In *Proceedings of the 28th British Machine Vision Conference*, pages 42.1–42.12, 2017.

[Geng *et al.*, 2021] C. Geng, S.-J. Huang, and S. Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3614–3631, 2021.

[Grinsztajn *et al.*, 2022] L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems 35*, pages 507–520, 2022.

[Hastie *et al.*, 2009] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[Hendrycks and Gimpel, 2016] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR/Abstract*, 2001.04295, 2016.

[Ji *et al.*, 2020] R. Ji, L. Wen, L. Zhang, D. Du, Y. Wu, C. Zhao, X. Liu, and F. Huang. Attention convolutional binary neural tree for fine-grained visual categorization. In *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10468–10477, 2020.

[Kontschieder *et al.*, 2015] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulo. Deep neural decision forests. In *Proceedings of the 15th IEEE/CVF International Conference on Computer Vision*, pages 1467–1475, 2015.

[Li and Hoiem, 2017] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

[Li *et al.*, 2024] J. Li, R. Cai, and Y. Yan. Combinatorial routing for neural trees. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*, pages 4407–4415, 2024.

[Liu *et al.*, 2002] B. Liu, W.-S. Lee, P. S. Yu, and X. Li. Partially supervised classification of text documents. In *Proceedings of 19th International Conference on Machine Learning*, pages 387–394, 2002.

[Liu *et al.*, 2018] S. Liu, R. Garrepalli, T. Dietterich, A. Fern, and D. Hendrycks. Open category detection with PAC guarantees. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3169–3178, 2018.

[Mendes Júnior *et al.*, 2017] P. R. Mendes Júnior, R. Medeiros de Souza, R. de Oliveira Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, Otávio A. B. Penatti, R. da Silva Torres, and A. Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, 2017.

[Mu *et al.*, 2017] X. Mu, K.-M. Ting, and Z.-H. Zhou. Classification under streaming emerging new classes: A solution using completely-random trees. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1605–1618, 2017.

[Neal *et al.*, 2018] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li. Open set learning with counterfactual images. In *Proceedings of the 15th European Conference on Computer Vision*, pages 613–628, 2018.

[Qi, 2012] Y. Qi. Random forest for bioinformatics. In *Ensemble Machine Learning*, pages 307–323. Springer, 2012.

[Quinlan, 1986] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[Ramaswamy *et al.*, 2016] H. Ramaswamy, C. Scott, and A. Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2052–2060, 2016.

[Rebuffi *et al.*, 2017] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[Rifkin and Klautau, 2004] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141, 2004.

[Rudd *et al.*, 2017] E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult. The extreme value machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):762–768, 2017.

[Scheirer *et al.*, 2013] W.-J. Scheirer, A. Rocha, A. Sapkota, and T.-E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2013.

[Scheirer *et al.*, 2014] W.-J. Scheirer, L.-P. Jain, and T.-E. Boult. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2317–2324, 2014.

[Shu *et al.*, 2023] S. Shu, S. He, H. Wang, H. Wei, T. Xian, and L. Feng. A generalized unbiased risk estimator for learning with augmented classes. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 9829–9836, 2023.

[Simonyan and Zisserman, 2015] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[Tanha *et al.*, 2017] J. Tanha, M. Van Someren, and H. Afsarmanesh. Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics*, 8:355–370, 2017.

[Tanno *et al.*, 2019] R. Tanno, K. Arulkumaran, D. Alexander, A. Criminisi, and A. Nori. Adaptive neural trees. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6166–6175, 2019.

[Topalis and Polikar, 2008] M. D. Muhlbaierand A. Topalis and R. Polikar. Learn++.NC: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes. *IEEE Transactions on Neural Networks*, 20(1):152–168, 2008.

[Wang *et al.*, 2024] L. Wang, X. Zhang, H. Su, and J. Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024.

[Xu *et al.*, 2025] F. Xu, W. Chen, and W. Gao. On the learning with augmented class via forests. *CoRR/Abstract*, 2505.09294, 2025.

[Yan *et al.*, 2021] S. Yan, J. Xie, and X. He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the 34th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021.

[Zhang *et al.*, 2020] Y.-J. Zhang, P. Zhao, L. Ma, and Z.-H. Zhou. An unbiased risk estimator for learning with augmented classes. In *Advances in Neural Information Processing Systems 33*, pages 10247–10258, 2020.

[Zhou and Chen, 2002] Z.-H. Zhou and Z.-Q. Chen. Hybrid decision tree. *Knowledge-Based Systems*, 15(8):515–528, 2002.

[Zhou *et al.*, 2024] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9851–9873, 2024.

[Zhou, 2022] Z.-H. Zhou. Open-environment machine learning. *National Science Review*, 9(8):nwac123, 2022.

[Zou *et al.*, 2022] Y. Zou, S. Zhang, Y. Li, and R. Li. Margin-based few-shot class-incremental learning with class-level overfitting mitigation. In *Advances in Neural Information Processing Systems 35*, pages 27267–27279, 2022.