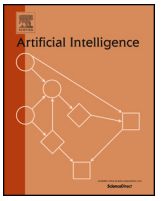
Artificial Intelligence 327 (2024) 104059

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Exploratory machine learning with unknown unknowns ☆

Peng Zhao [a,b], Jia-Wei Shan [a,b], Yu-Jie Zhang [a,b], Zhi-Hua Zhou [a,b,*]

[a] National Key Laboratory for Novel Software Technology, Nanjing University, China
[b] School of Artificial Intelligence, Nanjing University, China

ARTICLE INFO

Keywords:
Exploratory machine learning
Unknown unknowns
Robust AI
Robustness

ABSTRACT

In conventional supervised learning, a training dataset is given with ground-truth labels from a known label set, and the learned model will classify unseen instances to known labels. This paper studies a new problem setting in which there are unknown classes in the training data misperceived as other labels, and thus their existence appears unknown from the given supervision. We attribute the *unknown unknowns* to the fact that the training dataset is badly advised by the incompletely perceived label space due to the insufficient feature information. To this end, we propose the *exploratory machine learning*, which examines and investigates training data by actively augmenting the feature space to discover potentially hidden classes. Our method consists of three ingredients including rejection model, feature exploration, and model cascade. We provide theoretical analysis to justify its superiority, and validate the effectiveness on both synthetic and real datasets.

1. Introduction

In this paper, we study the task in which there are unknown labels hidden in the training dataset, namely some training instances belonging to a certain class are wrongly perceived as others, and thus appear unknown to the learned model. This is always the case when the label space is misspecified due to the insufficient feature information. Consider the task of medical diagnosis, where we need to train a machine learning model for the community healthcare centers based on their patient records, to help diagnose the cause of a patient with cough and dyspnea. As shown in Fig. 1, there are actually three causes: two common ones (*asthma* and *pneumonia*), as well as an unusual one (*lung cancer*). Note that the diagnosis of lung cancer crucially relies on the computerized tomography (CT) scan device, yet is too expensive to purchase. Thus, the community healthcare centers are not likely to diagnose patients with dyspepsia as cancer, resulting in that the class of "lung cancer" becomes invisible and hidden in the collected training dataset. As a result, the learned model will be unaware of this unobserved class, hence facing the unknown unknowns.

Similar phenomena occur in many other applications. For instance, the trace of a new-type aircraft was mislabeled as old-type aircrafts until performance of the aircraft detector is found poor (i.e., the capability of collected signals is inadequate), and the officer suspects that there are new-type aircrafts unknown previously. When the feature information is insufficient, there is a high risk to misperceive some classes of training data as others, leading to the existence of hidden unknown classes. More importantly, the hidden classes are sometimes of more interest, like in the above two examples. It is therefore crucial for the learned model to

☆ This article belongs to Special Issue: Open-World AI.

* Corresponding author.
E-mail address: zhouzh@lamda.nju.edu.cn (Z.-H. Zhou).

https://doi.org/10.1016/j.artint.2023.104059
Received 9 January 2023; Received in revised form 3 December 2023; Accepted 10 December 2023

Available online 19 December 2023
0004-3702/

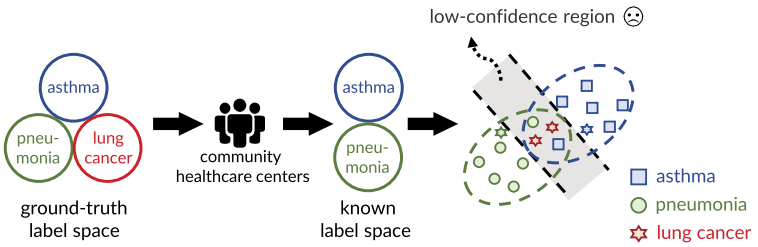


Fig. 1. Unknown unknowns in the task of medical diagnosis. Patients with *lung cancer* are misdiagnosed as *asthma* or *pneumonia* due to the lack of CT scan devices, and thus appear as unknown to the learned model.

discover hidden classes and classify known classes well simultaneously, and this is also one of the key requirements of robust and open-world/open environment artificial intelligence [1–3].

The *conventional supervised learning (SL)*, where a predictive model is trained on a given labeled dataset and then deployed to classify unseen instances into known labels, crucially relies on a high-quality training dataset. Thus, when the aforementioned *unknown unknowns* emerged in the training data, the conventional supervised learning cannot obtain a satisfied learned model. *Open category learning* (also known as *learning with new classes*), which focuses on handling unknown classes appearing only in the testing phase [4–8], assumes that the unknown classes only appear in the testing stage, while in above examples there exist unknown classes in training data (see Section 5 for more details). Neither of the learning frameworks could deal with the raised scenarios. As a result, it is necessary to develop new learning framework to handle such unknown unknowns that might emerge in the training data.

2. ExML: a new learning framework

The problem we are concerned with is essentially a class of *unknown unknowns*. In fact, how to deal with unknown unknowns is the fundamental question of robust artificial intelligence [2] and open-environment machine learning [3,9], and many studies have been devoted to addressing various aspects including changing distributions [10–12], evolvable features [13–15], open categories [4,16,8], etc. Different from them, we study a new problem setting ignored previously, that is, the training dataset is badly advised by the *incompletely perceived label space* due to the *insufficient feature information*. This problem turns out to be quite challenging, since feature space and label space are entangled and *both* of them are unreliable.

Notably, it is infeasible to merely pick out instances with low predictive confidence as hidden classes, because we can hardly distinguish: (i) instances from hidden classes that suffer from low-confidence predictions owing to the incomplete label space; (ii) instances from known classes that suffer from low-confidence predictions because of insufficient feature information. This characteristic reflects intrinsic hardness of learning with unknown unknowns due to feature deficiency, and it is therefore necessary to ask for external feature information.

There are lines of works sharing similar spirits, that is, asking for external feature information to enhance model performance, such as *detecting high-confidence false predictions* [17–19], *avoiding negative side effects* [20,21] and *active learning* [22]. However, our setting and developed methodologies are significantly different from theirs; see Section 5 for more details. In fact, these studies as well as our work both align with the *human-in-the-loop learning* principle, which leverages human knowledge to advance machine learning [23,24]. We believe there is potential for mutual benefit between ExML and other human-in-the-loop learning techniques, such as large language models (LLM) trained through *reinforcement learning from human feedback (RLHF) [25]*.

2.1. Exploratory machine learning

To handle unknown unknowns caused by the feature deficiency, we resort to the human in the learning loop to interact with environments for enhancing the data collection, more specifically, actively augmenting the feature space. The idea is that when a learned model remains performing poorly even fed with much more data, the learner will suspect existence of hidden classes and subsequently seek several candidate features to augment. Fig. 2 shows a straightforward example that the learner receives a dataset and observes that there are two classes with poor separability, resulting in a noticeable low-confidence region. After a proper feature augmentation, the learner will then realize that there exists an additional class hidden in the training data previously due to the feature deficiency.

Enlightened by the above example, we introduce a new learning framework called *exploratory machine learning* (ExML), which explores more feature information to deal with unknown unknowns caused by feature deficiency. The terminology of exploratory learning is originally raised in the area of education, defined as an approach to teaching and learning that encourages learners to examine and investigate new material with the purpose of discovering relationships between existing background knowledge and unfamiliar content and concepts [26,27]. In the context of machine learning, our proposed framework encourages learners to *examine and investigate the training dataset via exploring new feature information, with the purpose of classifying known classes and discovering potentially hidden classes*. Our proposed framework is also inspired by recent advances in cognitive science. For instance, when facing uncertain and constantly changing environments, the prefrontal cortex continuously constructs new strategies through exploration and evaluates their reliability [28,29]. Fig. 3 compares the proposed ExML to conventional supervised learning (SL). Conventional SL views the training dataset as an observable representation of environments and exploits it to train a model to predict the label.

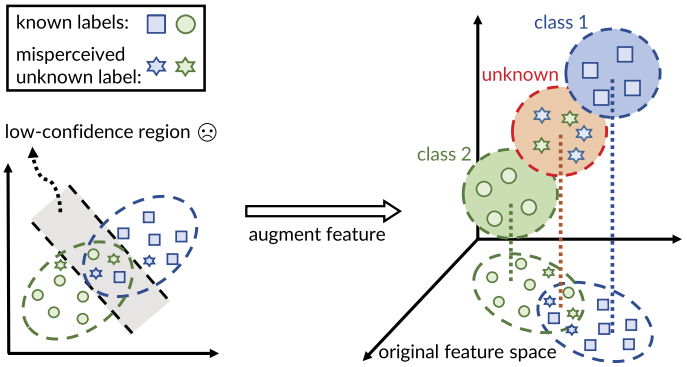


Fig. 2. An example illustrates that an informative feature can substantially improve separability of low-confidence samples and make the hidden class distinguishable. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

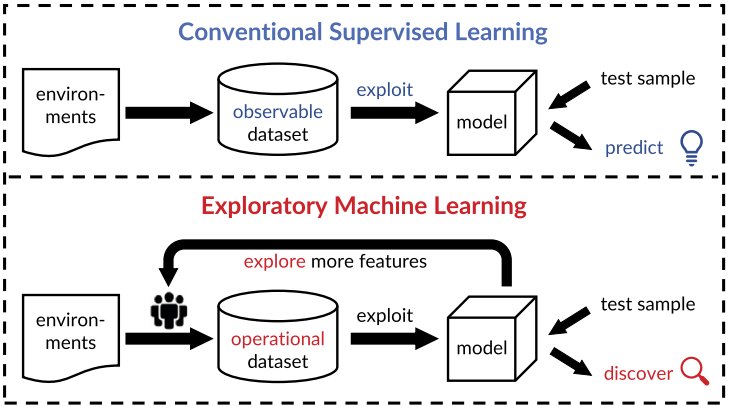


Fig. 3. Comparison of two learning frameworks. Conventional supervised learning exploits the observable dataset for prediction. Exploratory machine learning explores more features based on the operational dataset for both prediction and discovery of the hidden classes.

By contrast, ExML considers the training dataset is *operational*, where learners can examine and investigate the dataset by *exploring* more feature information, and thereby *discover* unknown unknowns due to feature deficiency. Note that we do not assume that the new class necessary exists. When there is no unknown classes, our approach still offers a powerful tool to present feature exploration to help refine the performance of conventional supervised learning.

We further develop an approach to implement the principle of ExML, consisting of three important ingredients: rejection model, feature exploration, and model cascade. The rejection model identifies suspicious instances that potentially belong to the hidden classes. Feature exploration guides which feature should be explored among the candidates, and then retrains the model on the augmented feature space. Model cascade allows a layer-by-layer processing to refine the selection of suspicious instances. Theoretical analysis is provided to justify the superiority of our proposed framework. Besides, we present empirical evaluations on synthetic data to illustrate the idea and further validate the effectiveness on real datasets.

2.2. Problem formulation

Training dataset The learner receives a training dataset $\widehat{D}_{tr} = \{(\widehat{\mathbf{x}}_i, \widehat{\mathbf{y}}_i)\}_{i=1}^{m}$, where the feature $\widehat{\mathbf{x}}_i \in \widehat{\mathcal{X}} \subseteq \mathbb{R}^d$ is from the *observed* feature space and the label $\widehat{\mathbf{y}}_i \in \widehat{\mathcal{Y}}$ is from the *incomplete* label space with N known classes. Throughout the paper, we focus on the binary case for simplicity. We remind that in our concerned unknown unknowns setting there exist training samples that are actually from hidden classes yet wrongly labeled as known classes due to feature deficiency.

Candidate features and cost budget Besides the training dataset, the learner can access a set of candidate features $\mathcal{A} = \{a_1, \ldots, a_K\}$, whose values are *unknown* before acquisition. Moreover, a certain cost c_i will be incurred to acquire an observation on the candidate feature a_i for any sample. The learner aims to identify top k informative features from the pool under the given budget $B > 0$ such that she will then augment the dataset on those top informative features in the testing stage. For convenience, we focus on the case that the learner desires to find the best feature, i.e., $k = 1$.

We expand the two examples in the introduction to demonstrate the rationality of our formulation. In the first example, suppose a patient's physical examination results suggest that he might have pneumonia, but the diagnosis is at a low confidence. At this point, the doctor may recommend the patient to do further examinations (i.e., the pulmonary histopathology examination, the CT scans,

etc.) which can be regarded as candidate features in our setting. The cost of doing these examinations varies, and the assistance they may provide for a more accurate diagnosis also differs. In the second example, when the performance of the aircraft is found poor, the detector may ask for more sources of signals (i.e., optical sensors, aviation sonar, etc.). The signals generated by the new equipment can be regarded as candidate features in our setting. The cost of deploying these devices varies, and the effectiveness of the signals also differs.

Testing stage Suppose the learner identifies the best feature as a_i, she will then augment the testing sample with this particular feature in the feature space, leading to an augmented feature space denoted by $\mathcal{X}_i = (\widehat{\mathcal{X}} \cup \mathcal{X}^i) \subseteq \mathbb{R}^{d+1}$ where $\mathcal{X}^i$ is the feature space of a_i and recall that $\widehat{\mathcal{X}} \in \mathbb{R}^d$ is the original feature space. The learned model requires predicting the label of the augmented testing sample, either classified to one of known classes or discovered as hidden classes (abbrev. hc).

Remark 1 *(Possible relaxations of some assumptions).* We have made several modeling assumptions are introduced in the above problem formulation for simplicity, with the aim of avoiding distractions of an over-complicated setting and better understanding the essence of this new problem setup. Indeed, our proposed principle can still work when relaxing these assumptions by borrowing more advanced techniques. For example, we can leverage multi-class rejection techniques [30,31] to generalize our framework into multi-class problems, and use top-k best arm identification [32,33] to select multiple augmented features. We leave those potential extensions as future works. ¶

Remark 2 *(Training-time and test-time feature cost).* Our problem formulation captures the training-time feature cost, which means the learner is required to pay for acquiring new features for the training samples. Note that in the testing stage, augmenting the testing sample with candidate features may also incur a certain cost. Our paper focuses on the training-time feature cost and designs budget allocation strategies for feature exploration, while it is also possible to extend our framework to further accommodate test-time feature cost by modifying the goal of feature exploration, for example, to encourage the algorithm to identify the feature with highest quality-cost ratio [34]. We leave the extension to test-time feature cost as future work. ¶

3. A practical approach

Due to the feature deficiency, the learner might be even unaware of the existence of hidden classes based on the observed training data. It is thus necessary to introduce the assumption that *instances with high predictive confidence are safe, i.e., they will be correctly predicted as one of the known classes*. The learner will suspect the existence of hidden classes (which are the unknown unknowns to the learner at the beginning) when the learned model performs badly.

We justify the necessity of the assumption. Actually, there are some previous works studying the problem of high-confidence false predictions without considering the issue of feature deficiency [17,18], in which there exist some instances that are wrongly predicted with high confidence. Since the model's performance is highly unreliable, to rectify that, they assume the existence of an oracle providing ground-truth labels for the given queries. However, in the presence of the feature deficiency as in our scenario, the problem would not be tractable unless there is an oracle able to provide ground-truth labels based on the insufficient feature representation, which turns out to be an even stronger assumption that does not hold in reality generally. So this paper focuses on the aforementioned case to trust the high-confidence predictions and we leave high-confidence unknown unknowns due to the insufficient feature as the future work to explore.

We further clarify and emphasize that the introduced assumption does not trivialize the problem setup, because notice that the low-predictive instances are *not* necessarily from hidden classes (as explained at the beginning of Section 2), which necessitates more efforts in discovering and identifying unknown unknowns. Following the methodology of ExML (examining the training dataset via exploring new feature information), we design a novel approach, which consists of three components: rejection model, feature exploration, and model cascade. Fig. 4 illustrates the main procedures, and we will describe the details of each component subsequently.

3.1. Rejection model

As shown in Fig. 4(a), at the beginning, the learner requires to train an initial model on the original dataset, with capability of identifying low-confidence instances. As emphasized previously (cf. the beginning of Section 2), these low-confidence instances could come from both known and hidden classes, so they are only detected as suspicious and will be refined in the further procedures.

In order to obtain such models, we leverage the techniques of learning with rejection [35], where the learned model will abstain from predicting instances whose maximum conditional probabilities are lower than a given threshold. More precisely, we learn a function pair $f = (h, g)$, where $h : \widehat{\mathcal{X}} \mapsto \mathbb{R}$ is the *predictive* function for the known classes and $g : \widehat{\mathcal{X}} \mapsto \mathbb{R}$ is the gate function to *reject* the hidden class. The sample $\widehat{\mathbf{x}}$ is classified to the hidden class if $g(\widehat{\mathbf{x}}) < 0$, and otherwise to the class of $\mathtt{sign}(h(\widehat{\mathbf{x}}))$. Such rejection models can be trained via optimizing the following objective:

$$\min_f \ \mathbb{E}_{(\widehat{\mathbf{x}},\widehat{\mathbf{y}})\sim\widehat{D}}[\ell_{0/1}(f,\widehat{\mathbf{x}},\widehat{\mathbf{y}};\theta)], \tag{1}$$

where

$$\ell_{0/1}(f,\widehat{\mathbf{x}},\widehat{\mathbf{y}};\theta) = \mathbb{1}_{\widehat{\mathbf{y}}\cdot h(\widehat{\mathbf{x}})<0} \cdot \mathbb{1}_{g(\widehat{\mathbf{x}})>0} + \theta \cdot \mathbb{1}_{g(\widehat{\mathbf{x}})\leq 0}$$

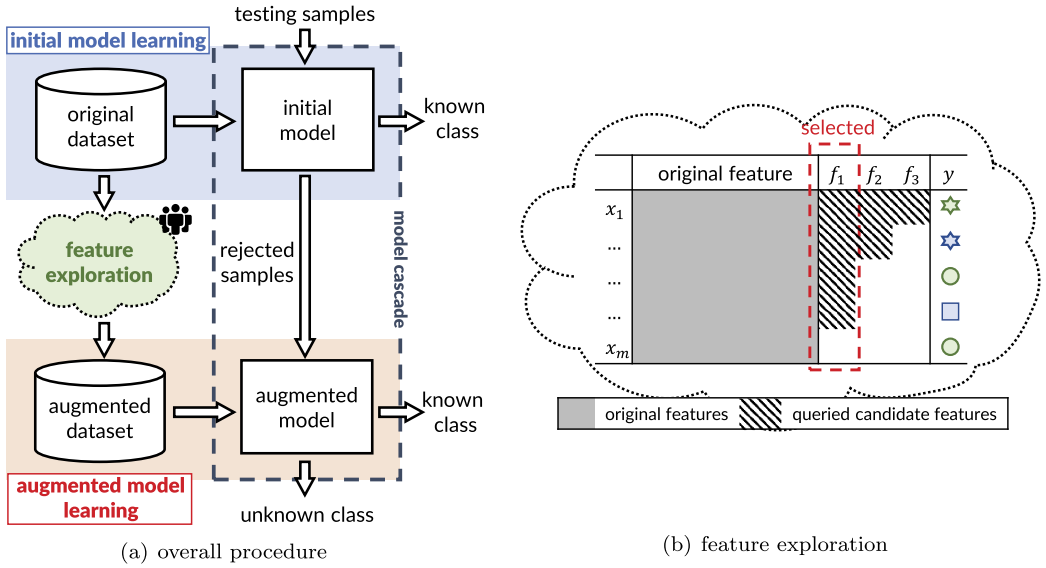


Fig. 4. The left figure shows the overall procedure of ExML. Our approach begins with an initial model (blue part), followed by exploring the best candidate feature among the candidates (green part). Afterwards, a learned model is retrained based on the augmented dataset, and finally is cascaded with the initial model to discover the hidden class (red part). The right figure describes the procedure of the feature exploration in ExML.

is the 0-1 loss of the rejection model f parameterized by the threshold $\theta \in (0, 1)$ and $\widehat{D}$ is the data distribution over $\widehat{\mathcal{X}} \times \widehat{\mathcal{Y}}$. To tackle the difficulty of non-convex optimization arising from the indicator function, Cortes et al. [35] introduce the following surrogate loss function

$$\ell_{surr}(f, \widehat{\mathbf{x}}, \widehat{\mathbf{y}}; \theta) = \max\left\{1 + \frac{1}{2}\left(g(\widehat{\mathbf{x}}) - \widehat{\mathbf{y}} \cdot h(\widehat{\mathbf{x}})\right), \theta \cdot \left(1 - \frac{g(\widehat{\mathbf{x}})}{1 - 2\theta}\right), 0\right\} \tag{2}$$

to approximate the original $\ell_{0/1}$ loss. Since the distribution is unknown and we cannot directly measure the risk, we choose the model that minimizes the empirical risk:

$$\min_{f \in \mathbb{H} \times \mathbb{H}} \frac{1}{m} \sum_{i=1}^{m} \ell_{surr}(f, \widehat{\mathbf{x}}_i, \widehat{\mathbf{y}}_i; \theta) + C_h \|h\|_{\mathbb{H}}^2 + C_g \|g\|_{\mathbb{H}}^2, \tag{3}$$

where C_h and C_g are regularization parameters, and $\mathbb{H}$ is the RKHS induced by the kernel $K : \widehat{\mathcal{X}} \times \widehat{\mathcal{X}} \mapsto \mathbb{R}$. By the representer theorem [36], the optimizer of (3) is in the form of $h(\widehat{\mathbf{x}}) = \sum_{i=1}^{m} u_i K(\widehat{\mathbf{x}}, \widehat{\mathbf{x}}_i)$ and $g(\widehat{\mathbf{x}}) = \sum_{i=1}^{m} w_i K(\widehat{\mathbf{x}}, \widehat{\mathbf{x}}_i)$, where u_i and w_i are coefficients to learn. So (3) can be reformulated as quadratic programming and solved efficiently.

Remark 3 *(Reliability of the initial model).* The reliability of the initial model is crucial to make ExML effective. Fortunately, we have many methods to enhance the reliability of the initial model. Since the training of the initial model goes as a standard process of conventional supervised learning, we can make use of any standard supervised learning techniques (e.g., data enhancement, feature engineering) to make the initial model more reliable. Besides, we can also adjust the rejection model (e.g., reduce the rejection cost θ) to make it easier to meet the assumption we made at the beginning of Section 3 (instances with high predictive confidence are safe), at a cost of rejecting more samples and passing them to the subsequent models. ¶

3.2. Feature exploration

If the initial model is unqualified (for instance, it rejects too many samples for achieving the desired accuracy), the learner will suspect the existence of hidden classes and explore new features to augment. In our setting, the learner requires to select the best feature from K candidates and retrain a model based on the augmented data, as shown in Fig. 4(b).

We emphasize that conventional feature selection is not feasible here, because it requires to know the values of candidate features, while these values are unknown before acquisitions. To address the challenge, we propose a novel procedure—*feature exploration*—to adaptively identify the most informative feature under the cost budget, *without* requiring feature values in advance. To address the issue, there are two fundamental questions to answer:

- how to measure the quality of candidate features?
- how to allocate the budget to identify the best feature?

In the following, we will answer these two questions and then describe our strategy for the feature exploration in ExML.

Algorithm 1 Median Elimination for Feature Exploration.

Input: Feature exploration budget B, original dataset $\widehat{D}_{tr} = \{(\widehat{\mathbf{x}}_i, \widehat{\mathbf{y}}_i)\}_{i=1}^m$, candidate feature pool $\mathcal{A} = \{a_1, \ldots, a_K\}$, threshold $\theta \in (0, 1)$.

Output: Selected feature $c_{i_s} \in \mathcal{A}$ and corresponding augmented model $\widehat{f}_{i_s}$.

Initialize: dataset $D_i = \emptyset$ for each feature $a_i \in \mathcal{A}$, set of active features $\mathcal{A}_1 = \mathcal{A}$, $T = \lceil \log_2 K \rceil$.
for $t = 1, \ldots, T$ **do**
  Randomly select $n_t = \lfloor B/(T|\mathcal{A}_t|) \rfloor$ samples from $\widehat{D}_{tr}$ and query active features $a_i \in \mathcal{A}_t$;
  Update D_i with selected samples and train a model $\widehat{f}_{t,i}$ on D_i by ERM (3), for all $a_i \in \mathcal{A}_t$;
  Compute $\widehat{R}_{t,i}^{surr}$ according to (5), for all $a_i \in \mathcal{A}_t$;
  Update $\mathcal{A}_{t+1}$ as half of features in $\mathcal{A}_t$ with lower $\widehat{R}_{t,i}^{surr}$;
end for

Feature quality measure. Denote by D_i the data distribution over $\mathcal{X}_i \times \widehat{\mathcal{Y}}$, where $\mathcal{X}_i$ is the augmented feature space of the i-th candidate feature. Recall that the augmented feature space is defined as $\mathcal{X}_i = (\widehat{\mathcal{X}} \cup \mathcal{X}^i) \subseteq \mathbb{R}^{d+1}$ where $\mathcal{X}^i$ is the feature space of a_i, see more notation details in Section 2.2. Then, we use the *Bayes risk* on D_i as the feature quality measure, defined as

$$R_i^* = R_i(f_i^*) = \min_f \ \mathbb{E}_{(\mathbf{x},\widehat{\mathbf{y}})\sim D_i}\left[\ell_{0/1}(f, \mathbf{x}, \widehat{\mathbf{y}}; \theta)\right], \tag{4}$$

where $R_i(f)$ is the expected $0/1$ risk of function f over D_i, and f_i^* minimizes $R_i(f)$ over all measurable functions. The Bayes risk essentially reflects the minimal error of any rejection model that can attain on the augmented data distribution. The value will be smaller when the selected augmented feature improves the separability more significantly, and thus the associated feature is believed more informative.

Due to the inaccessibility of the underlying distribution D_i, we approximate the Bayes risk by its empirical version evaluated on surrogate loss over the augmented data $D_i = \{(\mathbf{x}_j, \widehat{\mathbf{y}}_j)\}_{j=1}^{n_i}$,

$$\widehat{R}_i^{surr}(\widehat{f}_i) = \frac{1}{n_i}\sum_{j=1}^{n_i} \ell_{surr}(\widehat{f}_i, \mathbf{x}_j, \widehat{\mathbf{y}}_j; \theta), \tag{5}$$

where $\mathbf{x}_j \in \mathcal{X}_i, \widehat{\mathbf{y}}_j \in \widehat{\mathcal{Y}}$, and $\widehat{f}_i$ is the rejection model learned by ERM over the surrogate loss (3) on augmented dataset D_i. We prove that the approximation by surrogate loss almost does no harm to the theoretical guarantees on the performance of the proposed algorithm (in Section 4), and even better, we verify in experiments that the empirical surrogate loss is easy to be well-optimized to obtain an augmented feature with high quality (in Section 6).

Based on the feature quality measure (4) and its empirical version (5), we now introduce the budget allocation strategy to identify the best candidate feature.

Budget allocation strategy. The goal of the feature exploration is to identify the best feature within the limited budget, and meanwhile the model retrained on augmented data should have good generalization ability. Note that the feature quality is definitely unknown to the learner.

We first consider the simplified case of uniform cost, namely, $c_1 = c_2 = \cdots = c_K = 1$. For this setting, we propose two feature exploration strategies: uniform allocation and median elimination. Below, we describe the details.

Uniform Allocation. We have the uniform allocation strategy as follows, under the guidance of criterion (4). For each candidate feature a_i, $i \in [K]$, learner allocates $\lfloor B/K \rfloor$ budget and obtains an augmented dataset D_i. We can thus compute the empirical feature measure by (5), and select the feature with the smallest risk. The above strategy is simple yet effective. We prove that ExML equipped with uniform allocation as the feature exploration strategy can achieve a low excess risk with high probability, as demonstrated in Theorem 1 of Section 4.

Median Elimination. We further propose another variant inspired by the bandit theory [37] to improve the budget allocation efficiency. Specifically, we adopt the technique of *median elimination* (ME) [38], which removes one half of poor candidate features after every iteration and only the best one remains in the end, and proposed Algorithm 1 which can avoid allocating too many budgets on poor features. More specifically, the elimination proceeds in $T = \lceil \log_2 K \rceil$ episodes, in each episode, $\lfloor B/T \rfloor$ budget is allocated uniformly to all remaining candidate features, and the learner could query their values for updating the corresponding augmented datasets D_i. Then, the score $\widehat{R}_i^{surr}$ is calculated on the current augmented datasets D_i and the half features with high $\widehat{R}_i^{surr}$ are eliminated. In the last, only one candidate feature a_{i_s} will be left and its augmented dataset D_{i_s} contains around $\lfloor B/\log K \rfloor$ samples, which is the largest among all the candidate features.

As shown in Fig. 4(b), poor features are eliminated earlier, the budget left for the selected feature is thus improved from $\lfloor B/K \rfloor$ to $\lfloor B/\log K \rfloor$ by Algorithm 1, which ensures better generalization ability of the learned model. The behavior is formally justified in Theorem 2. In a nutshell, we find that median elimination shows its advantage in exploring the best candidate feature more efficiently than uniform allocation despite its higher probability that fails to identify the best candidate feature than uniform allocation, since both strategies enjoy an exponentially-decayed failing probability. We finally remark that our paper currently focuses on identifying the best feature, and our framework is ready for top k features identification ($k > 1$) by introducing more sophisticated techniques [32,33]. Feature exploration in our approach also shares similar ideas with a recent line of works called *feature budget learning* [39–41] (see Section 5 for more discussions). We believe that further leveraging the techniques from feature budget learning could be beneficial to our feature exploration problem.

Non-uniform Query Cost. We have assumed so far that the query of different features shares the same cost (unit-cost setting, i.e., $c_1 = c_2 = \cdots = c_K = 1$), and now we relax this assumption by considering the more general *non-uniform* cost for different candidate

features, i.e., $c_1, c_2, \ldots, c_K$ can be distinct. While our goal remains as identifying the best feature within the limited budget and meanwhile obtaining good generalization ability, new consideration appears after the non-uniform cost nature that the feature exploration algorithm should balance between querying good but expensive features and querying cheap but low-quality features. As a consequence, the feature exploration phase aims to identify the best candidate feature (namely, feature a_1) and meanwhile to ensure that there are a large number of queries in this returned feature. To this end, we propose two principles for adapting strategies to the non-uniform case.

- *Sample Alignment.* The first one is the *sample alignment principle*, where at each time we are allocating budget, the budget allocated to each active feature is aligned to ensure that a same number of samples is queried for each active feature. Specifically, when a total budget b is to be allocated to a set A of active features, the learner allocates to each feature $a_i \in A$ a total budget of $\left\lfloor \frac{c_i b}{\sum_{a_j \in A} c_j} \right\rfloor$ to ensure that a total number of $\left\lfloor \frac{b}{\sum_{a_j \in A} c_j} \right\rfloor$ samples are queried for each active feature.
- *Budget Alignment.* We further have another variant to improve the budget allocation efficiency, which is called the *budget alignment principle*. Specifically, when a total budget b is to be allocated to a set A of active features, the learner equally allocates to each feature a total budget of $\lfloor b/|A| \rfloor$, and thus $\lfloor b/(|A|c_i) \rfloor$ samples are queried for any active feature $a_i \in A$. Intuitively, we can have more training samples augmented with cheaper candidate feature, which possibly leads to a better generalization ability if the candidate feature has a relatively high quality. Therefore, the budget alignment principle may provide a better performance, since cheap features with relatively high quality are more sufficiently explored.

3.3. Model cascade

After the feature exploration, the learner will retrain a model on the augmented data. Considering that the augmented model might not always be better than the initial model, particularly when the budget is not enough or candidate features are not quite informative, we employ the ensemble method by proposing the *model cascade* mechanism to cascade the augmented model with the initial one. Concretely, high-confidence predictions are accepted in the initial model, the rest suspicious are passed to the next layer for feature exploration, those augmented samples with high confidence will be accepted by the augmented model, and the remaining suspicious continue to the next layer for further refinements. At the final layer, those samples with multiple refinements but are still suspicious will be classified into the unknown new class.

Essentially, our approach can be regarded as *a layer-by-layer processing for identifying instances of hidden classes*, and the procedures can be stopped until human discovers remaining suspicious are indeed with certain hidden structures. For simplicity, we only implement a two-layer architecture, that is, the suspicious samples in the second layer will be classified into the unknown new class.

Our proposed multi-layer model cascade provides a way of hierarchical refinements, but at a cost of error composition or overfitting during the learning process. Note that our model cascade strategy can be regarded as a sequential cascaded ensemble, thus, the aforementioned issues can be potentially alleviated by the techniques from *ensemble learning* [42]. Several interesting observations can be made from the view of ensemble learning. For example, since *diversity* is crucial for the success of ensemble learning [42,43], our proposed ExML framework may further benefit from diversity encouragement among multiple base learners (i.e., different models in the multi-layer cascade structure), such as bagging [44] and selective ensemble [45], etc. Moreover, it would be also useful to introduce diversity in the feature exploration, which is left as an interesting future work.

4. Theoretical analysis

In this section, we present theoretical analysis for our proposed exploratory machine learning (ExML) framework. Specifically, we first investigate the attainable excess risk of supervised learning, supposing that the best feature were *known* in advance. Next, we analyze the excess risk of ExML, demonstrating its effectiveness in terms of both the selection criterion and budget allocation strategies. In the following, we first present the theoretical result for supervised learning with known best feature (Section 4.1), and then provide the guarantee for ExML with unknown best feature (Section 4.2). The proofs are deferred to Appendix A.

Throughout the section, for each candidate feature a_i, we denote the corresponding hypothesis space as $\mathcal{H}_i, \mathcal{G}_i = \{\mathbf{x} \mapsto \langle \mathbf{w}, \Phi_i(\mathbf{x}) \rangle \mid \|\mathbf{w}\|_{\mathbb{H}_i} \leq \Lambda_i\}$, where Φ_i and $\mathbb{H}_i$ are induced feature mapping and RKHS of kernel K_i in the augmented feature space, and we also define $\kappa_i^2 = \sup_{\mathbf{x} \in \mathcal{X}_i} K_i(\mathbf{x}, \mathbf{x})$. For simplicity and without loss of generality, we assume that the feature indices are sorted in ascending order based on their associated feature quality, i.e., $R_1^* \leq \cdots \leq R_K^*$.

4.1. Supervised learning with known best feature

Suppose the best feature were known in advance. Given a budget B and the unit uniform cost of different features, evidently we could obtain B samples augmented with this particular (best) feature a_1. Let f_{SL} be the model learned by supervised learning via minimizing the objective (3). According to the standard learning theory literature [35,46], we know that for any $\delta > 0$, with probability at least $1 - \delta$, the excess risk is bounded by

$$R_1(f_{\mathrm{SL}}) - R_1^* \leq \mathcal{O}\left(\sqrt{\frac{(\kappa_1 \Lambda_1)^2}{B}} + \sqrt{\frac{\log(1/\delta)}{2B}} \right) + R_{ap}, \tag{6}$$

where $R_{ap} = \inf_{f \in \mathcal{H}_1 \times \mathcal{G}_1} R_1^{surr}(f) - \inf_f R_1(f)$ is the approximation error. Note that the definition of R_{ap} here is slightly more than the classical definition of approximation error that measures how well hypothesis spaces $\mathcal{H}_1$, $\mathcal{G}_1$ approach the target in terms of the expected risk $R_1(f) = \mathbb{E}_{(\mathbf{x},\hat{\mathbf{y}})\sim\mathcal{D}_1}[\ell_{0/1}(f,\mathbf{x},\hat{\mathbf{y}};\theta)]$ in the statistical learning literature [47], since our definition additionally counts the approximation error owing to optimizing the surrogate loss $\ell_{surr}(f)$ during the learning process instead of $\ell_{0/1}(f)$ due to the hardness of its non-convexity. Thus, if the best feature were *known* in advance, the excess risk of supervised learning would converge to the inevitable approximate error in the rate of $\mathcal{O}(1/\sqrt{B})$, with a given budget B.

4.2. Exploratory learning with unknown best feature

In reality, however, *the best feature is unfortunately unknown ahead of time*. More importantly, since the values of K candidate features are unavailable, it is *infeasible* to perform the feature selection. We show that by means of ExML (feature exploration), the excess risk also converges in a favorable rate, yet *without* requiring to know the best feature in advance. Below, we first introduce a key decomposition of excess risk in generic ExML (Section 4.2.1), then present the theoretical result of ExML equipped with the uniform allocation (Section 4.2.2) and ExML with median elimination (Section 4.2.3), respectively.

4.2.1. Key decomposition and exploratory regret

We first introduce the notations and an assumption used throughout the theoretical analysis in ExML, then present the key decomposition which demonstrates the different challenges in ExML comparing with conventional SL.

Notations and assumption We use $\widehat{D}_{tr,i}$ to denote the entire training dataset augmented with feature a_i and use $\widehat{R}_{tr,i}^{surr}(f)$ to denote the averaged surrogate risk on $\widehat{D}_{tr,i}$. Let $\widehat{f}_i^* \in \mathcal{H}_i \times \mathcal{G}_i$ be the minimizer of $\widehat{R}_{tr,i}^{surr}(f)$, namely, $\widehat{f}_i^* \in \arg\min_{f\in\mathcal{H}_i\times\mathcal{G}_i} \widehat{R}_{tr,i}^{surr}(f)$. To facilitate the theoretical analysis, we introduce the assumption that *the most informative feature leads to the smallest loss on the entire augmented training dataset*, more specifically, $\widehat{R}_{tr,1}^{surr}(\widehat{f}_1^*) = \min_{i\in[K]} \widehat{R}_{tr,i}^{surr}(\widehat{f}_i^*)$, noting that as mentioned earlier the features are supposed to be sorted according to the quality, $R_1^* \le \cdots \le R_K^*$, without loss of generality.

The assumption is natural in the sense that when deploying ExML framework to tackle unknown unknowns, one should already have tried collecting a relatively large training dataset (but without feature augmentation), so evaluating on the empirical data should be able to reflect the underlying feature quality. Moreover, the assumption is also necessary to the best of our understanding, because suppose otherwise, the most informative feature cannot be identified through the empirical data even with an unlimited feature budget, then obviously any algorithm can hardly approach a desired excess risk.

Remark 4 *(Most informative feature assumption over 0/1 loss).* One can notice that the assumption is made on the surrogate loss, while the feature quality is measured via the 0/1 loss. In fact, the assumption is to guarantee the performance of feature exploration, which includes feature quality evaluations on surrogate loss by ERM. Therefore, the loss function in the assumption should be aligned with the loss function used in the feature exploration algorithm. However, due to the difficulty of non-convex optimization, it is generally hard to proceed ERM on the 0/1 loss, thus it remains unclear whether we can obtain the same guarantees when making such an assumption over 0/1 loss, which is an interesting future issue to explore. ¶

We measure the performance of ExML by the excess risk $R_{i_s}(\widehat{f}_{i_s}) - R_1^*$, which is the difference between the expected risk of the hypothesis $\widehat{f}_{i_s}$ returned by ExML evaluated over the augmented feature space $\mathcal{X}_{i_s}$ and the Bayes risk R_1^* over the best augmented feature space $\mathcal{X}_1$. To proceed the theoretical analysis, we introduce an important quantity used in analyzing the behavior of ExML algorithms, defined as

$$\Delta_i = \widehat{R}_{tr,i}^{surr}(\widehat{f}_i^*) - \widehat{R}_{tr,1}^{surr}(\widehat{f}_1^*), \tag{7}$$

which qualifies the empirical difference of feature quality between feature i and that of the best feature. Let $\Delta = \min_{i\in[K],\Delta_i>0} \Delta_i$ be the smallest one in the candidate features, which we call as *optimality gap* measuring the hardness of feature exploration in ExML.

The key step in the analysis of generic ExML which demonstrates the different challenges comparing to SL is to decompose the excess risk of the learned model $\widehat{f}_{i_s}$ into five parts,

$$R_{i_s}(\widehat{f}_{i_s}) - R_1^* = \underbrace{R_{i_s}(\widehat{f}_{i_s}) - \widehat{R}_{tr,i_s}^{surr}(\widehat{f}_{i_s})}_{\texttt{term (a)}} + \underbrace{\widehat{R}_{tr,i_s}^{surr}(\widehat{f}_{i_s}) - \widehat{R}_{tr,1}^{surr}(\widehat{f}_1^*)}_{\texttt{term (b)}} + \underbrace{\widehat{R}_{tr,1}^{surr}(\widehat{f}_1^*) - \widehat{R}_{tr,1}^{surr}(f_1^*)}_{\texttt{term (c)}} + \underbrace{\widehat{R}_{tr,1}^{surr}(f_1^*) - R_1^{surr}(f_1^*)}_{\texttt{term (d)}} + \underbrace{R_{ap}}_{\texttt{term (e)}}. \tag{8}$$

The decomposition categorizes the error according to the sources they are incurred: `term (a)` and `term (d)` are the *generalization error* due to the inaccessibility of the true data distribution, and `term (b)` is the *exploratory regret*, which not only includes the generalization error due to the limited budget to query the candidate features of the entire training dataset, but also includes the *optimization error* due to the unknown best candidate feature in advance. The `term (b)` of exploratory regret thus reflects the main difference between ExML and supervised learning. Besides, `term (c)` is a negative term, and `term (e)` is the unavoidable approximation error. This key decomposition shows that ExML not only requires to control the generalization error as SL does, but also needs to have a low exploratory regret, which has not been considered in previous study. In the remaining of this section, we will show the power of our feature exploration algorithms in lemmas, and verify the effectiveness of our proposed ExML approach in theorems.

4.2.2. Exploratory learning with uniform allocation

According to the assumption on most informative feature which is introduced at the beginning of Section 4.2, we succeed in identifying the best feature a_1 as long as we succeed to identify a_1 as the best feature in the entire training dataset. For ExML with feature exploration by uniform allocation (see details in Section 3.2), we have the following lemma that bounds the exploratory regret as shown in `term` (b) of Eq. (8),

Lemma 1 *(Exploratory regret of uniform allocation). Let a_{i_s} be the feature identified by uniform allocation, then uniform allocation identifies the best feature (i.e., $i_s = 1$) with probability at least $1 - \delta_{fail}$, where*

$$\delta_{fail} = 4(K-1)\exp\left(-\frac{2}{9}\lfloor B/K\rfloor\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K\rfloor}}\right)^2\right), \tag{9}$$

providing that the identification condition $\lfloor B/K\rfloor > \frac{16((1-\theta)\kappa\Lambda)^2}{((1-2\theta)\Delta)^2}$ holds, with θ the threshold of rejection model defined in (2), $\Delta = \min_{i\in[K],\Delta_i>0}\Delta_i$ is the optimality gap defined in (7), $\Lambda = \sup_{i\in[K]}\Lambda_i$ and $\kappa = \sup_{i\in[K]}\kappa_i$.

Further more, for any $\delta > 0$, with probability at least $1 - \delta - \delta_{fail}$, we have

$$\widehat{R}^{surr}_{tr,i_s}(\widehat{f}_{i_s}) - \widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1) \le \frac{4-4\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K\rfloor}} + 2\sqrt{\frac{\log(2/\delta)}{2\lfloor B/K\rfloor}}.$$

Remark 5 *(Launch budget in feature exploration).* Lemma 1 bounds the exploratory regret induced by uniform allocation with high probability. We would notice that the identification condition introduces a "launch budget" for uniform allocation to be theoretically effective, and there is an extra probability δ_{fail} that uniform allocation would fail. These come from the statistical limit to differentiate features of different qualities with finite samples, and this statistical limit finally results in the difference between the excess risk bounds of ExML and supervised learning. ¶

Lemma 1 directly yields a bound on `term` (b) of Eq. (8), thus we can achieve the following theorem that validates the effectiveness of ExML equipped with uniform allocation:

Theorem 1 *(Excess risk of ExML with uniform allocation). Let a_{i_s} be the identified feature and $\widehat{f}_{i_s}$ be the augmented model returned by* ExML *with uniform allocation. Then, for any $\delta > 0$, with probability at least $1 - \delta - \delta_{fail}$, we have the following excess risk bound:*

$$R_{i_s}(\widehat{f}_{i_s}) - R^*_1 \le \mathcal{O}\left(\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K\rfloor}} + \sqrt{\frac{\log(6/\delta)}{2\lfloor B/K\rfloor}}\right) + R_{ap}, \tag{10}$$

with the failure probability $\delta_{fail} = \mathcal{O}(\exp(-\lfloor B/K\rfloor))$ that decays exponentially with respect to the total budget B (the formal definition can be found in (9) of Lemma 1), providing that the identification condition $\lfloor B/K\rfloor > \frac{64((1-\theta)\kappa\Lambda)^2}{((1-2\theta)\Delta)^2}$ holds, where θ is the threshold of rejection model defined in (2), $\Lambda = \sup_{i\in[K]}\Lambda_i$, $\kappa = \sup_{i\in[K]}\kappa_i$, and R_{ap} is the approximation error introduced in (6).

Remark 6 *(Comparison between excess risk of SL and ExML).* We have the following comparison between the theoretical results of SL and ExML. Comparing the excess risk bounds of (6) and (10), we can observe that ExML exhibits a similar convergence tendency to SL with *known* best feature yet *without* requiring to know the best feature in advance, which is realized at the expense of an extra $\sqrt{K}$ times factor for the best feature exploration as well as an extra failure probability δ_{fail}. ¶

4.2.3. Exploratory learning with median elimination

For ExML with feature exploration by median elimination (Algorithm 1 in Section 3.2), we have the following lemma that bounds the exploratory regret as shown in `term` (b) of Eq. (8),

Lemma 2 *(Exploratory regret of median elimination). Let a_{i_s} be the feature identified by median elimination, then median elimination identifies the best feature (i.e., $i_s = 1$) with probability at least $1 - \delta_{fail}$, where*

$$\delta_{fail} = \frac{8\exp\left(-\frac{2}{9}\lfloor B/(K\log_2 K)\rfloor\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/(K\log_2 K)\rfloor}}\right)^2\right)}{1-\exp\left(-\frac{2}{9}\lfloor B/(K\log_2 K)\rfloor\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/(K\log_2 K)\rfloor}}\right)^2\right)}, \tag{11}$$

providing that the identification condition $\lfloor B/(K\log_2 K)\rfloor > \frac{16((1-\theta)\kappa\Lambda)^2}{((1-2\theta)\Delta)^2}$ holds, with θ the threshold of rejection model defined in (2), $\Delta = \min_{i\in[K],\Delta_i>0}\Delta_i$ is the optimality gap defined in (7), $\Lambda = \sup_{i\in[K]}\Lambda_i$ and $\kappa = \sup_{i\in[K]}\kappa_i$.

Further more, with probability at least $1-\delta-\delta_{fail}$*, we have*

$$\widehat{R}^{surr}_{tr,i_s}(\widehat{f}_{i_s}) - \widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1) \leq \frac{4-4\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/\log_2 K\rfloor}} + 2\sqrt{\frac{\log(2/\delta)}{2\lfloor B/\log_2 K\rfloor}}\,.$$

Lemma 2 directly yields a bound on `term` (b) of Eq. (8), thus we can achieve the following theorem that validates the effectiveness of ExML equipped with median elimination (Algorithm 1):

Theorem 2 *(Excess risk of ExML with median elimination). Let* a_{i_s} *be the identified feature and* $\widehat{f}_{i_s}$ *be the augmented model returned by* ExML *with median elimination. Then, for any* $\delta > 0$*, with probability at least* $1-\delta-\delta_{fail}$*, we have the following excess risk bound:*

$$R_{i_s}(\widehat{f}_{i_s}) - R^*_1 \leq \mathcal{O}\left(\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/(\log_2 K)\rfloor}} + \sqrt{\frac{\log(6/\delta)}{2\lfloor B/(\log_2 K)\rfloor}}\right) + R_{ap}, \tag{12}$$

with the failure probability $\delta_{fail} = \mathcal{O}\left(\exp\left(-\lfloor B/(K\log_2 K)\rfloor\right)\right)$ *which decays exponentially with respect to the total budget* B *(the formal definition can be found in* (11) *in Lemma 2), providing that the identification condition* $\lfloor B/(K\log_2 K)\rfloor > \frac{64((1-\theta)\kappa\Lambda)^2}{((1-2\theta)\Delta)^2}$ *holds, where* θ *is the threshold of rejection model defined in* (2)*,* $\Lambda = \sup_{i\in[K]}\Lambda_i$*,* $\kappa = \sup_{i\in[K]}\kappa_i$*, and* R_{ap} *is the approximation error in* (6)*.*

The proof of Theorem 2 mostly parallels with that of Theorem 1, which includes a decomposition of excess risk as shown in Section 4.2.1 and a key lemma that bounds the exploratory regret induced by median elimination as shown in Lemma 2.

Remark 7 *(Comparison between uniform allocation and median elimination).* Comparing Theorem 1 and Theorem 2, we can see that median elimination improves the $\sqrt{K}$ times factor paid for the feature exploration by uniform allocation to $\sqrt{\log_2 K}$ in the excess risk bound, as poor candidate features have been removed in the earlier episodes. By contrast, median elimination requires a larger "launch budget" in the identification condition compared to uniform allocation, and have a higher failure probability δ_{fail}, because only a partial budget is used in early stages and so the best feature has a larger probability to be mistakenly discarded in the earlier episodes. Nevertheless, the failure probability in both results decays exponentially with respect to the total budget, which is thus low-order term and can be ignored in many situations. ¶

5. Related work

In this section, we briefly discuss some topics related to our proposed ExML framework.

Open category learning Open category learning is also named as learning with new classes, which focuses on handling unknown classes appearing only in the testing phase [4–8], see the recent survey [16] for a thorough overview of literature. Although these studies also care about the unknown classes detection, they differ from us significantly and thus cannot apply to our more challenging scenario: on one hand, they do not consider the issue of feature deficiency in the training data, which leads to great challenge in our problem; on the other hand, there exist unknown classes in the training data in our setting, while for open category learning the unknown classes only appear in the testing stage.

Learning with unknown unknowns How to deal with unknown unknowns is a fundamental problem of robust artificial intelligence [2] and open-environment machine learning [3,9]. A line of works deals with *high-confidence false predictions* appear due to model's unawareness of such kind of mistake, which are also referred to as a kind of "unknown unknowns" [17–19]. Existing studies typically ask for external human expert to help identifying high-confidence false predictions and then retrain the model with the guidance. Although these works also consider unknown unknowns and resort to external human knowledge, their setting and methodology differ from ours: our unknown unknowns are caused due to feature deficiency, so the learner requires to augment features rather than querying labels. Another kind of related works considers to *avoid negative side effects*, which means that the reward functions in the prediction/decision process may be misleading due to the incomplete knowledge of the environments. There are emerging works that aim to detect and avoid the problem of negative side effects [48–51]. These works and ours both aim to enhance the robustness of AI systems in the face of unknown unknowns, while the specific problem modeling and developed methodologies are significantly different.

Active learning Active learning aims to achieve greater accuracy with fewer labels by asking queries of unlabeled data to be labeled by the human expert [22]. Active learning bares certain similarities with our exploratory learning in the spirit — instead of learning in a purely passive way, we both resort to some additional information sources to help the learning process. Interestingly, there are also some works querying features [52–54] to improve learning with missing features via as few as possible queries of entry values (feature of an instance). However, unlike their setting, we augment new features to help the identification of the unknown classes rather than querying missing values of the given feature to improve the performance of known classes classification.

Learning with rejection Learning with rejection gives the classifier an option to reject an instance instead of providing a low-confidence prediction [55]. Plenty of works are proposed to design effective algorithms [56–60] and establish theoretical foundations [30,31,35,61–63]. As aforementioned, methods of learning with rejection cannot be directly applied in exploratory machine learning since it will result in inaccurate rejections of instances from known classes, and meanwhile, it cannot exploit new features.

Feature budget learning Feature budget learning considers a variant of supervised learning where an access of each feature on each sample is attached a cost, and the goal is to minimize the error within a given budget. This subject is initiated in [39]. Hazan and Koren [40] pioneered the study of this area in linear regression considering uniform costs, and Kukliansky and Shamir [41] generalizes the results into the cases with non-uniform cost. The feature exploration module in our proposed approach is related to the setting in feature budget learning, while their results are restricted to specific choices of loss functions in order to get strong theoretical guarantees. Nevertheless, we believe it is possible to integrate the techniques of feature budget learning to develop more adaptive mechanisms in identifying top-k features.

6. Experiments

In this section, we conduct experiments to examine empirical performance of the proposed exploratory machine learning (ExML). Specifically, we provide evaluations on synthetic data for visualizing the superiority of ExML to conventional supervised learning in handling unknown unknowns. Then, we report results on real-world datasets to demonstrate the effectiveness of the overall method, as well as the usefulness of feature exploration and model cascade modules.

The rejection models are learned with Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\gamma)$, where the bandwidth γ is set as $\gamma = \text{median}_{\mathbf{x}_i,\mathbf{x}_j \in D}(\|\mathbf{x}_i - \mathbf{x}_j\|_2^2)$. Besides, parameters C_h, C_g are set as 1. We select the best rejection threshold θ of augmented model from the pool $[0.1, 0.2, 0.3, 0.4]$ for each algorithm, and threshold of the initial model is selected by cross validation to ensure 95% accuracy on high-confidence predictions. Feature exploration budget is set as $B = b \cdot mK$, where m is number of training samples, K is number of candidate features, $b \in [0, 1]$ is the budget ratio.

Remark 8 *(Automatic parameter tuning).* We repeated the experiments by running ExML with each parameter settings in the pool, and the reported performance of each algorithm is the performance under their individual optimal parameters in hindsight. In fact, we can also perform automatic parameter tuning on the augmented model. For example, we can firstly use the best parameters of initial model to spend a proportion of budget on feature exploration to build a validation dataset, then select the best parameters by cross-validation on this dataset. ¶

6.1. Synthetic data for illustration

We first illustrate the advantage of exploratory machine learning over the conventional supervised learning in discovery of the hidden classes on the synthetic data.

Setting Following the illustrative example in Fig. 1, we generate data with 3-dim feature and 3 classes, each class has 100 samples. Fig. 5(a) presents the ground-truth distribution. However, as shown in Fig. 5(b), the third-dim feature is unobservable in training data, resulting in a hidden class (hc) located in the intersection area of known classes (kc1 and kc2). Samples from hc are mislabeled as kc1 or kc2 randomly. In detail, instances from each class are generated from a 3-dim Gaussian distributions. The means and variances are $[-a, 0, -z]$ and $\sigma \cdot \mathbf{I}_{3\times3}$ for class 1, $[a, 0, z]$ and $\sigma \cdot \mathbf{I}_{3\times3}$ for class 2 as well as $[0, 0, 0]$ and $\sigma/2 \cdot \mathbf{I}_{3\times3}$ for class 3, where $\mathbf{I}_{3\times3}$ is a 3×3 identity matrix. We fix $\sigma = 3a$ and set $z = 5a$. In the training stage, the third-dim is unobservable and the third class is randomly labeled as another two. There are 100 instances for each class in the training data.

Besides, we generate 9 candidate features in various qualities, whose angle to the horizon varies from 10° to 90°, the larger the better. Fig. 5(c) plots the augmented feature space via t-SNE. The budget ratio is $b = 20\%$. In the testing stage, the learner requires to predict on the 3-dim data, where the third dimension is the selected candidate features.

Contenders There are two contenders for the synthetic experiments, namely SL and ExML. For all the rejection model used in the experiments, we employ the Gaussian kernel with the bandwidth $\gamma = \text{median}_{\mathbf{x}_i,\mathbf{x}_j \in D}(\|\mathbf{x}_i - \mathbf{x}_j\|_2^2)$, and parameters C_h, C_g are set to 1.

- **SL**: the rejection model [35] trained on the 2-dim labeled training data, following the paradigm of conventional supervised learning. The threshold θ is chosen as one achieving best accuracy on the testing data from the pool $[0.1, 0.2, 0.3, 0.4]$.
- **ExML**: our proposal with cascade models and using median elimination for feature exploration. The threshold for the initial rejection model is selected by cross validation to ensure 95% accuracy on high-confidence samples. The threshold for the augmented rejection model is chosen as one achieving best accuracy on the testing data from the pool $[0.1, 0.2, 0.3, 0.4]$. The budget ratio is 20%.

Results We first conduct SL to train a rejection model based on the 2-dim training data, and then perform ExML to actively augment the feature within the budget to discover unknown unknowns. Figs. 6(a) and 6(b) plot the results, demonstrating a substantial advantage of ExML over SL in discovering the hidden class and predicting known classes. Furthermore, Fig. 6(c) reports budget allocation of each candidate feature over 50 times repetition. We can see that the allocation clearly concentrates to more informative features (with larger angles), which validates the effectiveness of median elimination for the best feature exploration.

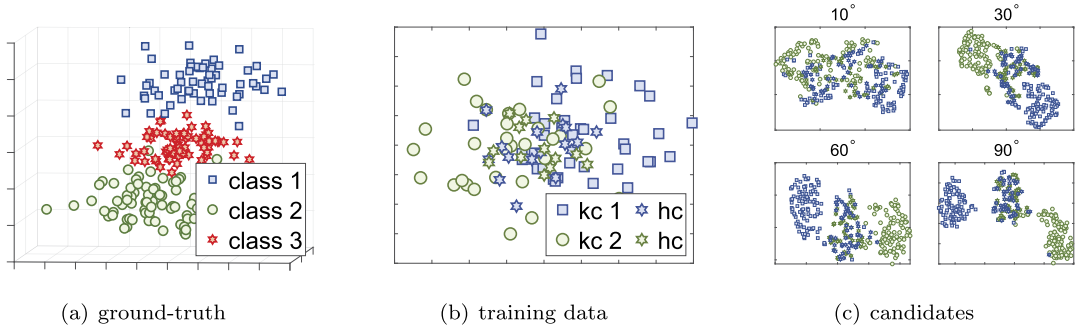


Fig. 5. Visualization of synthetic data. (a): ground-truth distribution; (b): training data (only first two dims are observable); (c): *t*-SNE of candidate features with various qualities (larger angles imply better features).

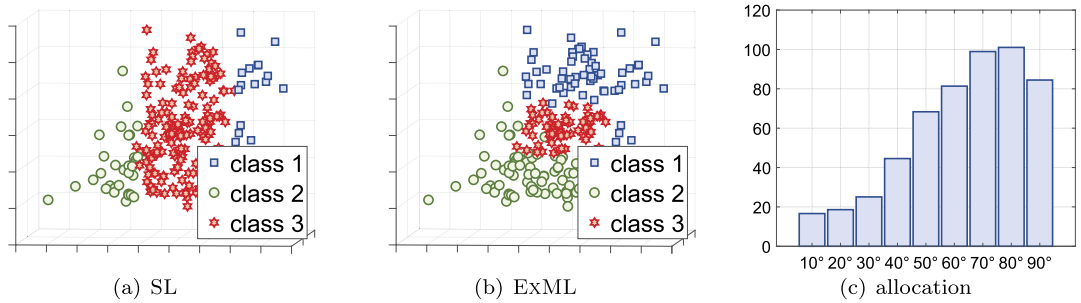


Fig. 6. Visualization of results. (a)/(b): SL/ExML; (c): budget allocation of ExML with median elimination.

6.2. *Benchmark data for evaluation*

We further evaluate on a UCI benchmark dataset *Mfeat* [64].

Dataset Mfeat is a multi-view dataset[1] containing 2000 samples and 6 views of features extracted by various methods, whose brief semantic and statistical information are listed as follows.

- **Fac**: profile correlations, 216-dim;
- **Pix**: pixel averages in 2×3 windows, 240-dim;
- **Kar**: Karhunen-Love coefficients, 64-dim;
- **Zer**: Zernike moments, 47-dim;
- **Fou**: Fourier coefficients of the character shapes, 76-dim;
- **Mor**: morphological features, 6-dim.

According to the domain knowledge, we can sort the six features by their feature quality as: Fac > Pix > Kar > Zer > Fou > Mor, in a descending order.

Since *Mfeat* is a multi-class dataset, we randomly sample 5 configurations to convert it into the binary classification task, where each known class and hidden class contain three original classes (and so each configuration includes an amount of 1800 samples), and the instances from the hidden class are randomly mislabeled as one of known classes. There are in total 50 random configurations for training. As for the candidate features, we take one as original and the rest are prepared in the candidate set. Before training, we normalize all the features to the range $[0, 1]$. In the training stage, 600 instances are randomly samples from the whole dataset for 10 times to form the labeled training data. In the testing stage, the rest 1200 instances are used for measuring the performance of compared algorithms.

Setting We randomly sample 600 instances as the training data for 10 times, and the rest are used for testing. As for the candidate features, each one of six views (features) is taken as original feature and the rest are prepared as candidate features. The budget ratio varies from 10% to 30%.

Contenders Apart from SL, we include two ExML variants: $\text{ExML}_{\text{csd}}^{\text{UA}}$ and $\text{ExML}_{\text{aug}}^{\text{ME}}$ for ablation studies. Here aug/csd denotes the final model is only the augmented or cascaded with the initial model; UA/ME refers to feature exploration by uniform allocation or median elimination.

[1] The dataset can be downloaded from http://archive.ics.uci.edu/ml/datasets/Multiple+Features.

Table 1
Evaluation on *Mfeat* dataset. Features are sorted by descending feature qualities. Bold font indicates algorithms that significantly outperform others (paired t-test at 95% significance level).

Feature	Description	Budget	SL	$\text{ExML}_{\text{aug}}^{\text{ME}}$	$\text{ExML}_{\text{csd}}^{\text{UA}}$	ExML ($= \text{ExML}_{\text{csd}}^{\text{ME}}$)	Recall
Fac	Profile correlations	10%	**93.39 ± 1.66**	71.80 ± 9.55	92.39 ± 2.79	92.40 ± 2.78	48%
		20%	**93.39 ± 1.66**	82.26 ± 7.52	91.95 ± 3.32	92.00 ± 3.27	46%
		30%	**93.39 ± 1.66**	89.29 ± 4.72	92.20 ± 3.33	92.50 ± 2.86	44%
Pix	Pixel averages in 2 × 3 windows	10%	**92.19 ± 2.47**	70.53 ± 8.27	90.54 ± 6.27	90.55 ± 6.31	58%
		20%	**92.19 ± 2.47**	81.70 ± 7.16	90.84 ± 6.17	90.87 ± 6.09	54%
		30%	**92.19 ± 2.47**	88.67 ± 4.14	90.45 ± 5.74	**91.82 ± 4.26**	68%
Kar	Karhunen-Love coefficients	10%	**86.87 ± 3.43**	70.25 ± 10.2	85.55 ± 4.94	85.90 ± 4.85	56%
		20%	**86.87 ± 3.43**	81.46 ± 6.88	85.21 ± 5.46	**86.49 ± 4.81**	54%
		30%	86.87 ± 3.43	86.01 ± 5.41	86.52 ± 4.71	**88.18 ± 3.57**	56%
Zer	Zernike moments	10%	73.82 ± 8.82	69.61 ± 10.7	72.96 ± 10.4	**76.17 ± 8.52**	82%
		20%	73.82 ± 8.82	**80.86 ± 8.02**	77.31 ± 7.89	**81.72 ± 7.33**	82%
		30%	73.82 ± 8.82	**86.07 ± 5.51**	81.11 ± 6.79	**86.33 ± 5.04**	86%
Fou	Fourier coefficients	10%	68.73 ± 9.07	69.42 ± 9.68	68.88 ± 11.8	**75.92 ± 8.81**	82%
		20%	68.73 ± 9.07	82.11 ± 6.48	77.93 ± 8.27	**85.03 ± 4.39**	88%
		30%	68.73 ± 9.07	**89.90 ± 3.69**	82.45 ± 5.20	**89.35 ± 3.89**	92%
Mor	Morphological features	10%	57.47 ± 15.3	69.09 ± 11.3	66.58 ± 13.5	**71.07 ± 11.1**	80%
		20%	57.47 ± 15.3	**79.60 ± 10.1**	73.61 ± 8.86	**79.74 ± 9.92**	84%
		30%	57.47 ± 15.2	**87.44 ± 7.34**	78.31 ± 9.00	**86.98 ± 7.07**	90%

• $\textbf{ExML}_{\textbf{csd}}^{\textbf{UA}}$: our proposal with cascade model and using *uniform allocation* for feature exploration, sharing the same parameters setting as ExML.

• $\textbf{ExML}_{\textbf{aug}}^{\textbf{ME}}$: our proposal *without* cascade model and using median elimination for feature exploration, sharing the same parameters setting as ExML.

For all ExML methods, the budget ratio b varies from 10% to 30%. The parameter settings of SL and ExML are the same as those in the synthetic experiments (Section 6.1).

Measure We measure the performance of all the methods by the classification. Additionally, we introduce the *recall* to measure the effectiveness of feature exploration, defined as the ratio of the number of cases when identified feature is one of its top 2 features to the total number.

• **Accuracy**: the mean and standard deviation of the predictive accuracy on testing dataset over 50 configurations, where the true label of hidden classes is observable.

• **Recall**: the ratio of the number of cases when identified feature is one of its top 2 features to the total number, where the quality of features is measured by the attainable accuracy of the augmented model trained on the whole dataset with this particular feature.

Results Table 1 reports mean and std of the predictive accuracy, and all features are sorted in descending order by their quality. We first compare the conventional supervised learning (SL) to (variants of) ExML. When the original features are in high quality (Kar, Pix, Fac), SL could achieve favorable performance and there is no need to explore new features. However, in the case where uninformative original features are provided, which is of more interest for ExML, SL degenerates severely and $\text{ExML}_{\text{aug}}^{\text{ME}}$ (the single ExML model without model cascade) achieves better performance even with the limited budget. Besides, from the last column, we can see that informative candidates (top 2) are selected to strengthen the poor original features, which validates the efficacy of the proposed budget allocation strategy.

Since the $\text{ExML}_{\text{aug}}^{\text{ME}}$ is not guaranteed to outperform SL, particularly with the limited budget on poor candidate features, we propose the cascade structure. Actually, ExML approach (aka, $\text{ExML}_{\text{csd}}^{\text{ME}}$) achieves roughly *best-of-two-worlds* performance, in the sense that it is basically no worse or even better than the best of SL and $\text{ExML}_{\text{aug}}^{\text{ME}}$. It turns out that even $\text{ExML}_{\text{csd}}^{\text{UA}}$ could behave better than $\text{ExML}_{\text{aug}}^{\text{ME}}$. These results validate the effectiveness of the model cascade component.

Notice that there are also some cases that the augmented model ($\text{ExML}_{\text{aug}}^{\text{ME}}$) outperforms the cascade model ($\text{ExML}_{\text{aug}}^{\text{ME}}$). Indeed, since the rejection model at the first layer is trained on the original dataset, the performance of the cascaded model will be affected by the rejection model to some extent. When feature exploration is of high quality, the performance of the second layer itself already becomes good enough, the model cascade may slightly affect the overall performance. Nevertheless, the cascading structure can still prevent the impact of low-quality feature exploration on the overall performance, enhancing the robustness of our method.

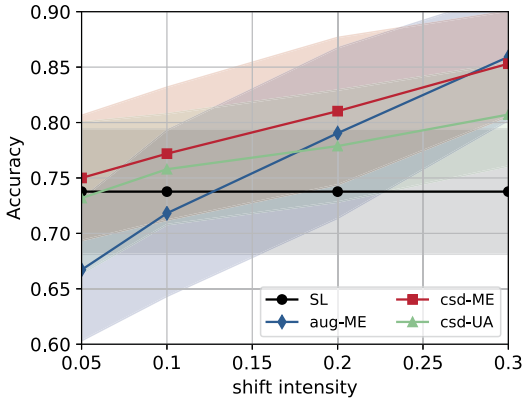


Fig. 7. Performance comparisons of all the contenders.

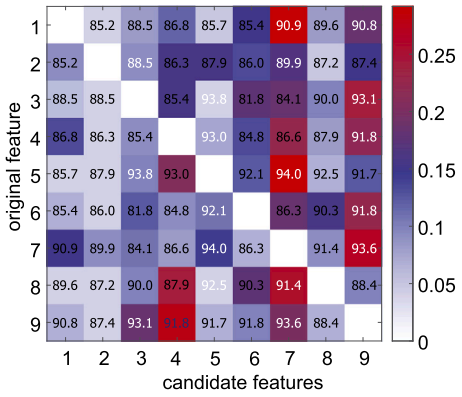


Fig. 8. Illustration of budget allocation with median elimination.

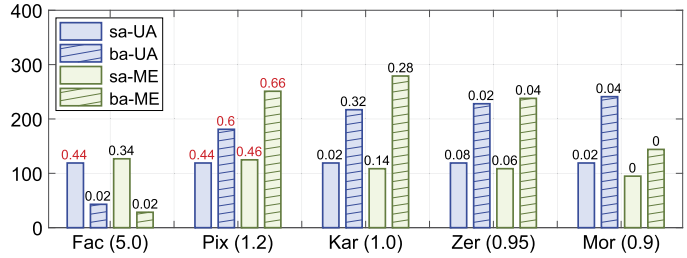
6.3. Real data of activities recognition

We additionally examine the effectiveness of our proposed algorithm on a real-world dataset called *RealDisp*,[2] which is an activities recognition task [65]. Specifically, there are 9 on-body sensors used to capture various actions of participants. Each sensor is placed on different parts of the body and provides 13-dimensional features including 3-dim from acceleration, 3-dim from gyro, 3-dim from magnetic field orientation and another 4-dim from quaternions. Hence, in this dataset we have 117 features in total.

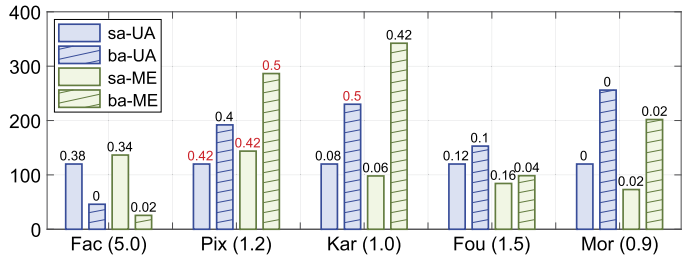
Dataset In our experiments, three types of actions (*walking*, *running*, and *jogging*) of the first subject under the ideal-placement are included to form the dataset containing 2000 instances, where 30% of them are used for training and 70% for testing. In the training data, one sensor is deployed and the class of jogging is misperceived as walking or running randomly. The learner would explore the rest eight candidate features to discover the unknown unknowns. Thus, there are 9 partitions, and each is repeated for 10 times by sampling the training instances randomly.

Results Fig. 7 shows the mean and std of accuracy, our approach ExML (aka, $\text{ExML}^{\text{ME}}_{\text{csd}}$) outperforms others, validating the efficacy of our proposal. In addition, Fig. 8 illustrates the budget allocation when the budget ratio $b = 30\%$. The i-th row denotes the scenario when the i-th sensor is the original feature, and patches with colors indicate the fraction of budget allocated to each candidate feature. The number above a patch means the attainable accuracy of the model trained on the whole training dataset with the particular feature. We highlight the top two candidate features of each row in white, and use blue color to indicate selected feature is not in top two. The results show that ExML with median elimination can select the top two informative features to augment for all the original sensors. The only exception is the 9-th sensor, but quality of the selected feature (91.8%) does not deviate too much from the best one (93.6%). These results reflect the effectiveness of our feature exploration strategy.

[2] http://archive.ics.uci.edu/ml/datasets/REALDISP+Activity+Recognition+Dataset.



(a) Sample allocation with original feature Fou



(b) Sample allocation with original feature Zer

Fig. 9. Sample allocation with different original features. The x-axis denotes the candidate features sorted by descending qualities and the values in brackets are their costs. The y-axis denotes the number of queries on each feature. The values over the bars are the ratios of the number of cases that the algorithm identifies the corresponding candidate as the best feature, with the reds indicating the most frequent ones.

6.4. Non-uniform cost

We finally examine the effectiveness of our proposed principle for non-uniform cost on the previous *Mfeat* dataset with each group of features attached with a different cost.

Dataset *Mfeat* is a multi-view dataset containing 2000 samples and 6 views of features extracted by various methods, whose brief semantic and statistical information are listed as follows. Additionally, we attach each feature a cost which are shown in the brackets below, in order to simulate the non-uniform cost scenario.

- **Fac** (5.0): profile correlations, 216-dim;
- **Pix** (1.2): pixel averages in 2×3 windows, 240-dim;
- **Kar** (1.0): Karhunen-Love coefficients, 64-dim;
- **Zer** (0.95): Zernike moments, 47-dim;
- **Fou** (1.5): Fourier coefficients of the character shapes, 76-dim;
- **Mor** (0.9): morphological features, 6-dim.

According to the domain knowledge, we can sort the six features by their feature quality as: Fac > Pix > Kar > Zer > Fou > Mor, in a descending order.

We remark that our cost attachment is rational, as it includes a feature with highest quality but is expensive (Fac), features with relatively high quality and are cheap (Pix, Kar), features that are cheap but with low quality (Zer, Mor), and a feature with low quality and is expensive (Fou). Intuitively, the expected behavior of the algorithm is to query more samples on the relatively good and cheap features (Pix, Kar), rather than to spend a lot on expensive features which leads to poor generalization ability, nor to query a lot on inherently poor features.

Setting Same as the experimental setup in Section 6.2, we randomly sample 600 instances as the training data for 10 times, and the rest are used for testing. As for the candidate features, each one of six views (features) is taken as original feature and the rest are prepared as candidate features. The budget ratio varies from 10% to 30%.

Contenders We include four ExML variants: $\text{ExML}_{\text{SA}}^{\text{UA}}$, $\text{ExML}_{\text{BA}}^{\text{UA}}$, $\text{ExML}_{\text{SA}}^{\text{ME}}$ and $\text{ExML}_{\text{BA}}^{\text{ME}}$ for ablation studies. Here SA/BA denotes the principle of non-uniform cost adaptation, where SA means sample alignment and BA means budget alignment; UA/ME refers to feature exploration by uniform allocation or median elimination. For all ExML methods, the parameter settings are the same as those in the synthetic experiments (Section 6.1).

Results Table 2 reports mean and std of the predictive accuracy, and all features are sorted in descending order by their quality. As verified in previous experiments, when the original features are in high quality (Pix, Fac), all contenders mostly rely on the prediction of the first layer, which lead to similar performance. However, in the case where uninformative original features are provided, ExML achieves better performance with limited budget since features with better quality are explored, and within the four

Table 2
Evaluation on *Mfeat* dataset attached with non-uniform costs. Features are sorted by descending feature qualities. Bold font indicates algorithms that significantly outperform others (paired t-test at 95% significance level).

Feature	Description	Budget	$\text{ExML}_{\text{SA}}^{\text{UA}}$	$\text{ExML}_{\text{BA}}^{\text{UA}}$	$\text{ExML}_{\text{SA}}^{\text{ME}}$	$\text{ExML}_{\text{BA}}^{\text{ME}}$
Fac	Profile correlations	10%	91.81 ± 3.10	91.81 ± 3.10	91.58 ± 5.74	91.58 ± 5.74
		20%	92.30 ± 2.45	92.31 ± 2.44	91.74 ± 3.32	91.73 ± 3.25
		30%	92.13 ± 3.32	92.14 ± 3.32	92.33 ± 2.82	92.26 ± 2.83
Pix	Pixel averages in 2 × 3 windows	10%	90.90 ± 3.77	90.90 ± 3.77	91.22 ± 3.61	91.22 ± 3.62
		20%	90.33 ± 6.14	90.42 ± 5.27	90.65 ± 4.52	90.86 ± 4.28
		30%	90.96 ± 4.11	90.78 ± 4.23	91.57 ± 3.06	**92.72 ± 2.51**
Kar	Karhunen-Love coefficients	10%	84.27 ± 5.84	84.14 ± 6.09	84.11 ± 5.80	84.95 ± 6.02
		20%	84.64 ± 5.85	84.78 ± 6.55	84.67 ± 5.62	**88.84 ± 3.74**
		30%	84.95 ± 5.36	86.35 ± 4.96	87.27 ± 4.09	**90.65 ± 3.08**
Zer	Zernike moments	10%	70.99 ± 9.77	70.72 ± 8.76	74.16 ± 9.54	**79.42 ± 6.20**
		20%	76.02 ± 8.19	76.92 ± 7.78	81.33 ± 7.98	**86.32 ± 6.77**
		30%	80.60 ± 6.88	81.10 ± 6.60	85.93 ± 6.24	**90.59 ± 3.83**
Fou	Fourier coefficients	10%	69.67 ± 9.52	69.02 ± 10.3	74.45 ± 7.72	**76.93 ± 7.65**
		20%	77.43 ± 6.96	75.26 ± 7.22	81.39 ± 7.48	**88.04 ± 3.12**
		30%	84.08 ± 4.30	83.21 ± 5.13	87.57 ± 3.86	**90.17 ± 3.21**
Mor	Morphological features	10%	63.61 ± 13.9	65.59 ± 10.4	68.95 ± 10.7	**74.31 ± 7.44**
		20%	73.04 ± 8.46	72.85 ± 11.2	77.41 ± 9.90	**86.14 ± 6.72**
		30%	79.61 ± 8.84	82.38 ± 7.44	85.32 ± 7.74	**90.28 ± 5.90**

contenders, $\text{ExML}_{\text{BA}}^{\text{ME}}$ outperforms the other three algorithms because $\text{ExML}_{\text{BA}}^{\text{ME}}$ allocates more budget to relatively good but much cheaper features (Pix, Kar).

Moreover, Fig. 9 shows the sample allocation of each candidate feature over 50 random configurations with the budget ratio $b = 20\%$. The colors of the bars indicate the basic budget allocation strategy, i.e., blue for uniform allocation and green for median elimination, and the shades in the bars indicate the non-uniform adaptation principles, i.e., the empty shade for sample allocation and the dot for budget allocation. Besides, the values over the bars are the ratios of the number of cases that the algorithm identifies the corresponding candidate as the best feature, with the red texts are the most frequent ones. We can see that the budget alignment principle ('ba-') avoids allocating too much budget to query expensive feature (Fac) comparing to the sample alignment principle ('sa-' with the empty shade). Besides, median elimination ('ME') shows a clearer concentration on relatively good but much cheaper features (Pix, Kar) comparing to uniform allocation ('UA'), which results in a better generalization ability. This verifies our interpretation of Table 2, and validates the effectiveness of our median elimination strategy as well as the budget alignment principle proposed in Section 3.2.

7. Conclusion

This paper studies the task of learning with unknown unknowns, where there exist some instances in training datasets belonging to an unknown hidden class but are wrongly perceived as known classes, due to the insufficient feature information. To address this issue, we propose the *exploratory machine learning* (ExML) to encourage the learner to examine and investigate the training dataset by exploring more features to discover potentially hidden classes. Following this principle, we design an approach consisting of three procedures: rejection model, feature exploration, and model cascade. By leveraging techniques from bandit theory, we prove the rationale and efficacy of the feature exploration procedure. Experiments validate the effectiveness of our approach.

There remain many interesting directions to further push forward the study of exploratory machine learning. First, as mentioned, one may borrow more advanced techniques to relax some current modeling assumptions such as binary known classes, best feature exploration, etc. Second, the method proposed in this paper is merely one implementation of the ExML framework, and exploring other effective mechanisms of feature exploration and hierarchical processing is also left as an interesting future work. Third, since in the environments with unknown unknowns, it would be difficult to expect passive learning can do well and the algorithm should explore necessary additional information from the environments, we believe the methodology behind our proposed ExML framework can serve as a principled way to handle unknown unknowns even beyond the scope of our concerned one due to feature deficiency.

Furthermore, unknown unknowns not only appear in the tasks of prediction, but also in the field of decision making. There are paradigms that models the sequential decision-making processes, such as reinforcement learning and rehearsal learning. In reinforcement learning, an agent learns to make decisions by performing actions in an environment to achieve maximum cumulative reward [66]. As for rehearsal learning, the learner tries to act proactively to prevent undesirable outcomes, which is a promising domain for further exploration [67]. Evidently, the unknown unknowns issue becomes even more severe in decision-making tasks compared to the prediction tasks, because the effect of unknown unknowns at current decision stage may entangle with the effect of unknown unknowns in the past stages. We believe that the methodology behind our proposed ExML framework, especially the principle of interactively exploring more information from environments, can be extended to decision-making scenarios.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This research was supported by NSFC (U23A20382, 61921006), JiangsuSF (BK20220776), National Postdoctoral Program for Innovative Talent, China Postdoctoral Science Foundation (2023M731597), and the Collaborative Innovation Center of Novel Software Technology and Industrialization. We are grateful for the anonymous reviewers from AAAI and AIJ for their valuable comments.

Appendix A. Omitted proofs

In this section, we present the proofs of the main results introduced in Section 4. We first introduce some useful lemmas in A.1, then prove the excess risk bounds given by Theorem 1 and Theorem 2 in A.2 and A.3. After that, we prove the exploratory regret bounds given by Lemma 1 and Lemma 2 in A.4 and A.5, and finally we give the proof of one of the useful lemmas originally proposed in this paper in A.6.

A.1. Useful lemmas

We introduce two useful lemmas before the proof of main results.

We first have the following lemma on the generalization error of the rejection model, which can be regarded as a counterpart of [35, Theorem 5].

Lemma 3. *Let $\mathcal{H}$ and $\mathcal{G}$ be the kernel-based hypotheses $\mathcal{H}$, $\mathcal{G} = \{\mathbf{x} \mapsto \langle \mathbf{w}, \Phi(\mathbf{x})\rangle \mid \|\mathbf{w}\|_{\mathbb{H}} \leq \Lambda\}$. Then for any $\delta > 0$, with probability of $1-\delta$ over the draw of a sample D of size m from $\mathcal{D}$, the following holds for all $f \in \mathcal{H} \times \mathcal{G}$:*

$$R(f) - \widehat{R}_D^{surr}(f) \leq \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}}, \tag{A.1}$$

where $\kappa^2 = \sup_{\mathbf{x}\in\mathcal{X}} K(\mathbf{x},\mathbf{x})$ and $K : \mathcal{X}\times\mathcal{X} \mapsto \mathbb{R}$ is the kernel function associated with $\mathbb{H}$.

We then have the following lemma, which bounds the probability that a sub-optimal candidate feature is considered better than the optimal feature in a single empirical evaluation, which is the basic step in analyzing the effectiveness of feature exploration. The proof of Lemma 4 can be found in Appendix A.6.

Lemma 4. *For any $i \in [K]$ with $\Delta_i > 0$, if $\widehat{f}_i$ is trained by ERM $\widehat{R}_i^{surr}(f)$ on n samples i.i.d. chosen in $\widehat{D}_{tr,i}$, and $\widehat{f}_1$ is trained by ERM $\widehat{R}_1^{surr}(f)$ on n samples i.i.d. chosen in $\widehat{D}_{tr,1}$, then*

$$\Pr\left[\widehat{R}_i^{surr}(\widehat{f}_i) < \widehat{R}_1^{surr}(\widehat{f}_1)\right] \leq 4\exp\left(-\frac{2}{9}n\left(\frac{\Delta_i}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n}}\right)^2\right),$$

providing that the identification condition $n > \frac{16((1-\theta)\kappa\Lambda)^2}{((1-2\theta)\Delta)^2}$ holds, where $\Lambda = \sup_{i\in[K]}\Lambda_i$ and $\kappa = \sup_{i\in[K]}\sup_{\mathbf{x}\in\mathcal{X}_i} K_i(\mathbf{x},\mathbf{x})$.

A.2. Proof of Theorem 1

Proof. According to Eq. (8), the excess risk of learned model $\widehat{f}_{i_s}$ can be decomposed into five parts,

$$R_{i_s}(\widehat{f}_{i_s}) - R_1^* = \underbrace{R_{i_s}(\widehat{f}_{i_s}) - \widehat{R}_{tr,i_s}^{surr}(\widehat{f}_{i_s})}_{\texttt{term (a)}} + \underbrace{\widehat{R}_{tr,i_s}^{surr}(\widehat{f}_{i_s}) - \widehat{R}_{tr,1}^{surr}(\widehat{f}_1^*)}_{\texttt{term (b)}} + \underbrace{\widehat{R}_{tr,1}^{surr}(\widehat{f}_1^*) - \widehat{R}_{tr,1}^{surr}(f_1^*)}_{\texttt{term (c)}} + \underbrace{\widehat{R}_{tr,1}^{surr}(f_1^*) - R_1^{surr}(f_1^*)}_{\texttt{term (d)}} + \underbrace{R_{ap}}_{\texttt{term (e)}},$$

where `term (a)` is the gap between the expected risk of the learned model $\widehat{f}_{i_s}$ evaluated by 0/1 loss and the empirical risk evaluated by surrogate loss, and `term (b)` is the difference between empirical criterion of the selected feature and that of the best feature, where $\widehat{f}_1^*$ refers to the best empirical model on the full training dataset augmented with best feature. Besides, `term (c)` captures the difference between the empirical risk of $\widehat{f}_1^*$ and that of the best hypothesis evaluated by surrogate loss $f_1^* = \arg\min_{f\in\mathcal{H}_1\times\mathcal{G}_1} R_1^{surr}(f)$, `term (d)` is the generalization error of f_1^* evaluated by surrogate loss, and `term (e)` is the unavoidable approximation error. Notice

that term (c) ≤ 0 by the definition of $\widehat{f}_1^*$. Thus, to prove the theorem, it is sufficient to bound term (a), term (b) and term (d) respectively.

According to Lemma 3, for any $\delta_1 > 0$, term (a) can be directly bounded by

$$\texttt{term (a)} \leq \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{m}} + \sqrt{\frac{\log(1/\delta_1)}{2m}}, \tag{A.2}$$

with probability at least $1-\delta_1$.

By classical derivation of generalization bound based on Rademacher complexity, for any $\delta_2 > 0$, we have the following bound of term (d) with probability at least $1-\delta_2$,

$$\widehat{R}^{surr}_{tr,1}(f_1^*) - R_1^{surr}(f_1^*) \leq 2\mathfrak{R}_m(\widetilde{\mathcal{F}}_1) + \sqrt{\frac{\log(1/\delta_2)}{2m}},$$

where $\widetilde{\mathcal{F}}_1 = \{\ell_{surr} \circ f \mid f \in \mathcal{H}_1 \times \mathcal{G}_1\}$. According to [35, Theorem 5] we further have

$$\mathfrak{R}_m(\widetilde{\mathcal{F}}_1) \leq \frac{1-\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{m}},$$

thus for any $\delta_2 > 0$ we obtain with probability at least $1-\delta_2$,

$$\texttt{term (d)} \leq \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{m}} + \sqrt{\frac{\log(1/\delta_2)}{2m}}. \tag{A.3}$$

We then bound term (b). By Lemma 1, for any $\delta_3 > 0$, we directly obtain with probability at least $1-\delta_3-\delta_{\text{fail}}$,

$$\texttt{term (b)} \leq \frac{4-4\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K \rfloor}} + 2\sqrt{\frac{\log(2/\delta_3)}{2\lfloor B/K \rfloor}}.$$

For any $\delta > 0$, let $\delta_1 = \delta_2 = \delta_3 = \delta/3$ and apply the union bound inequality, we have with probability at least $1-\delta-\delta_{\text{fail}}$,

$$\begin{aligned} R_{i_s}(\widehat{f}_{i_s}) - R_1^* &\leq \frac{4-4\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{m}} + \frac{4-4\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K \rfloor}} + 2\sqrt{\frac{\log(3/\delta)}{2m}} + 2\sqrt{\frac{\log(6/\delta)}{2\lfloor B/K \rfloor}} + R_{ap} \\ &= \mathcal{O}\left(\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K \rfloor}} + \sqrt{\frac{\log(6/\delta)}{2\lfloor B/K \rfloor}}\right) + R_{ap}. \end{aligned}$$

Finally, since $\lfloor B/K \rfloor > \frac{64((1-\theta)\kappa\Lambda)^2}{((1-2\theta)\Delta)^2}$, we have $\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K \rfloor}} \geq \frac{\Delta}{4}$, which is strictly greater than an absolute constant, so we can obtain an upper bound on fail probability as

$$\begin{aligned} \delta_{\text{fail}} &= 4(K-1)\exp\left(-\frac{2}{9}\lfloor B/K \rfloor\left(\frac{\Delta}{2} - \frac{4-4\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K \rfloor}}\right)^2\right) \\ &\leq 4(K-1)\exp\left(-\frac{2}{9}\lfloor B/K \rfloor\frac{\Delta^2}{16}\right) \\ &= 4(K-1)\exp\left(-\frac{\Delta^2}{72}\lfloor B/K \rfloor\right) \\ &= \mathcal{O}(\exp(-\lfloor B/K \rfloor)), \end{aligned}$$

and the proof is finished. □

A.3. Proof of Theorem 2

Proof. We first apply the same excess risk decomposition as shown in Eq. (8),

$$R_{i_s}(\widehat{f}_{i_s}) - R_1^* = \underbrace{R_{i_s}(\widehat{f}_{i_s}) - \widehat{R}^{surr}_{tr,i_s}(\widehat{f}_{i_s})}_{\texttt{term (a)}} + \underbrace{\widehat{R}^{surr}_{tr,i_s}(\widehat{f}_{i_s}) - \widehat{R}^{surr}_{tr,1}(\widehat{f}_1^*)}_{\texttt{term (b)}} + \underbrace{\widehat{R}^{surr}_{tr,1}(\widehat{f}_1^*) - \widehat{R}^{surr}_{tr,1}(f_1^*)}_{\texttt{term (c)}} + \underbrace{\widehat{R}^{surr}_{tr,1}(f_1^*) - R_1^{surr}(f_1^*)}_{\texttt{term (d)}} + \underbrace{R_{ap}}_{\texttt{term (e)}},$$

and term (a), term (c) and term (d) can be bounded following the same derivation as in Theorem 1. According to Lemma 2, we also have an upper bound of term (b) with probability at least $1-\delta_3-\delta_{\text{fail}}$,

$$\widehat{R}^{surr}_{tr,i_s}(\widehat{f}_{i_s}) - \widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1) \le \frac{4-4\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/\log_2 K\rfloor}} + 2\sqrt{\frac{\log(2/\delta_3)}{2\lfloor B/\log_2 K\rfloor}},$$

where

$$\delta_{\text{fail}} = \frac{8\exp\left(-\frac{2}{9}\lfloor B/(K\log_2 K)\rfloor\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/(K\log_2 K)\rfloor}}\right)^2\right)}{1-\exp\left(-\frac{2}{9}\lfloor B/(K\log_2 K)\rfloor\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/(K\log_2 K)\rfloor}}\right)^2\right)}.$$

We then proceed to estimate the order of δ_{fail}. Since $\lfloor B/(K\log_2 K)\rfloor > \frac{64((1-\theta)\kappa\Lambda)^2}{((1-2\theta)\Delta)^2}$ we have $\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K\rfloor}} \ge \frac{\Delta}{4}$, and so

$$\begin{aligned}
&\exp\left(-\frac{2}{9}\lfloor B/(K\log_2 K)\rfloor\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/(K\log_2 K)\rfloor}}\right)^2\right)\\
&\le \exp\left(-\frac{2}{9}\lfloor B/(K\log_2 K)\rfloor\frac{\Delta^2}{16}\right)\\
&= \exp\left(-\frac{\Delta^2}{72}\lfloor B/(K\log_2 K)\rfloor\right),
\end{aligned}$$

which upper-bounds the numerator of δ_{fail}. Further let $C = \frac{64((1-\theta)\kappa\Lambda)^2}{((1-2\theta)\Delta)^2}$ for simplicity as it appears to be a constant independent of B and K. We conclude that

$$\begin{aligned}
&1-\exp\left(-\frac{2}{9}\lfloor B/(K\log_2 K)\rfloor\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/(K\log_2 K)\rfloor}}\right)^2\right)\\
&\ge 1-\exp\left(-\frac{\Delta^2}{72}\lfloor B/(K\log_2 K)\rfloor\right)\\
&\ge 1-\exp\left(-\frac{\Delta^2 C}{72}\right),
\end{aligned}$$

which shows that the denominator of δ_{fail} is greater than an absolute constant independent of B, and so we have $\delta_{\text{fail}} = \mathcal{O}(\exp(-\lfloor B/(K\log_2 K)\rfloor))$. Again follow the derivation in the proof of Theorem 1, combine the results and set $\delta_1 = \delta_2 = \delta_3 = \delta/3$ finishes the proof. □

A.4. Proof of Lemma 1

Proof. If uniform allocation does not return the empirically best feature, then there must exists $i \in [K]$ s.t. a_i is not the best feature, while its estimated risk is lower than the estimated risk of the best feature, i.e. $\widehat{R}^{surr}_i(\widehat{f}_i) < \widehat{R}^{surr}_1(\widehat{f}_1)$. Therefore, the algorithm returns the best feature with probability at least $1-\delta_{\text{fail}}$, where

$$\begin{aligned}
\delta_{\text{fail}} &= \Pr\left[\exists\, i\in[K],\ i\neq 1 \wedge \widehat{R}^{surr}_i(\widehat{f}_i) < \widehat{R}^{surr}_1(\widehat{f}_1)\right]\\
&\le \sum_{i\in[K],i\neq 1}\Pr\left[\widehat{R}^{surr}_i(\widehat{f}_i) < \widehat{R}^{surr}_1(\widehat{f}_1)\right]\\
&\le 4\sum_{i\in[K],i\neq 1}\exp\left(-\frac{2}{9}\lfloor B/K\rfloor\left(\frac{\Delta_i}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K\rfloor}}\right)^2\right)\\
&\le 4(K-1)\exp\left(-\frac{2}{9}\lfloor B/K\rfloor\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K\rfloor}}\right)^2\right),
\end{aligned}$$

which proves the first part of the lemma. Specifically, the first inequality is because the union bound inequality, and the second inequality is according to Lemma 4.

To prove the second part, we firstly condition on the event that the algorithm has already identified an empirically best feature a_1. Define distribution $\mathcal{P}$ to be the uniform distribution on $\widehat{D}_{tr,1}$, we have

$$\widehat{R}^{surr}_{tr,i_s}(\widehat{f}_{i_s}) - \widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1) = \widehat{R}^{surr}_{tr,1}(\widehat{f}_1) - \widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1) = \underbrace{\widehat{R}^{surr}_{tr,1}(\widehat{f}_1) - \widehat{R}^{surr}_1(\widehat{f}_1)}_{\texttt{term (a)}} + \underbrace{\widehat{R}^{surr}_1(\widehat{f}_1) - \widehat{R}^{surr}_1(\widehat{f}^*_1)}_{\texttt{term (b)}} + \underbrace{\widehat{R}^{surr}_1(\widehat{f}^*_1) - \widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1)}_{\texttt{term (c)}}$$

where term (a) is the generalization error of $\widehat{f}_1$ on $\mathcal{P}$, term (b) is the difference between the empirical error of the empirically best hypothesis $\widehat{f}_1$ and the best hypothesis $\widehat{f}^*_1$ on $\mathcal{P}$, and term (c) is the generalization error of $\widehat{f}^*_1$ on $\mathcal{P}$. Notice that term (b) ≤ 0 since $\widehat{f}_1$ minimizes $\widehat{R}^{surr}_1$ by the ERM criterion. Thus, to prove the second part, it suffices to bound term (a) and term (c). By the standard analysis of generalization error based on the Rademacher complexity [47], with probability at least $1 - \delta/2$, we have

$$\texttt{term (a)} \leq \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K \rfloor}} + \sqrt{\frac{\log(2/\delta)}{2\lfloor B/K \rfloor}},$$

and

$$\texttt{term (c)} \leq \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K \rfloor}} + \sqrt{\frac{\log(2/\delta)}{2\lfloor B/K \rfloor}}.$$

Therefore, conditioning on the event that the algorithm returns a best feature, then with probability at least $1-\delta$, the uniform allocation algorithm satisfies

$$\widehat{R}^{surr}_{tr,i_s}(\widehat{f}_{i_s}) - \widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1) \leq \frac{4-4\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\lfloor B/K \rfloor}} + 2\sqrt{\frac{\log(2/\delta)}{2\lfloor B/K \rfloor}}.$$

Since the event occurs with probability at least $1 - \delta_{\text{fail}}$, we conclude the second part of the proof by the union bound inequality. □

A.5. Proof of Lemma 2

Proof. Without loss of generality, assume throughout the proof that $K = 2^c$ for some positive integer c, and that B is a multiplier of $K \log_2 K$. Let n_t be the number of samples collected at round t, and $\widehat{R}^{surr}_{t,i}(f)$ be the empirical surrogate risk on the samples collected at round t. Suppose $a_1 \in \mathcal{A}_t$, and consider the probability $\delta^{(t)}_{\text{fail}}$ that a_1 is discarded at round t. For any $a_i \in \mathcal{A}_t$ s.t. $\Delta_i > 0$, let $p^{(t)}_i$ be the probability that $\widehat{R}^{surr}_{t,i}(\widehat{f}_{t,i}) < \widehat{R}^{surr}_{t,1}(\widehat{f}_{t,1})$ with $\widehat{f}_{t,j}$ the models trained at round t. By Lemma 4 we have

$$\begin{aligned} p^{(t)}_i &\leq 4\exp\left(-\frac{2}{9}n_t\left(\frac{\Delta_i}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_t}}\right)^2\right) \\ &\leq 4\exp\left(-\frac{2}{9}n_t\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_1}}\right)^2\right). \end{aligned}$$

If a_1 is discarded at round t, then there must be at least $\frac{|\mathcal{A}_t|}{2}$ features a_i such that a_i is not the best feature but $\widehat{f}_{t,i}$ has lower empirical risk on D_i than that of $\widehat{f}_{t,1}$ on D_1. Let X be the random variable indicating the number of features satisfying the above property, it is easy to verify that

$$\mathbb{E}[X] = \sum_{a_i \in \mathcal{A}_t} p^{(t)}_i \leq 4|\mathcal{A}_t|\exp\left(-\frac{2}{9}n_t\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_1}}\right)^2\right).$$

By Markov's inequality we have

$$\delta^{(t)}_{\text{fail}} = \Pr\left[X \geq \frac{|\mathcal{A}_t|}{2}\right] \leq \frac{\mathbb{E}[X]}{|\mathcal{A}_t|/2} \leq 8\exp\left(-\frac{2}{9}n_t\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_1}}\right)^2\right).$$

So we can conclude that

$$\delta_{\text{fail}} = \sum_{t=1}^{T}\delta^{(t)}_{\text{fail}} \leq 8\sum_{t=1}^{T}\exp\left(-\frac{2}{9}n_t\left(\frac{\Delta}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_1}}\right)^2\right).$$

Since $|\mathcal{A}_{t+1}| = \frac{|\mathcal{A}_t|}{2}$, we have $n_{t+1} = 2n_t = \cdots = 2^t n_1$. Therefore,

$$
\begin{aligned}
\delta_{\text{fail}} &\le 8\sum_{t=1}^{T}\exp\left(-\frac{2}{9}n_t\left(\frac{\Delta}{2}-\frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_1}}\right)^2\right)\\
&\le 8\sum_{t=1}^{T}\exp\left(-\frac{2}{9}tn_1\left(\frac{\Delta}{2}-\frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_1}}\right)^2\right)\\
&\le 8\sum_{t=1}^{\infty}\exp\left(-\frac{2}{9}tn_1\left(\frac{\Delta}{2}-\frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_1}}\right)^2\right)\\
&= \frac{8\exp\left(-\frac{2}{9}n_1\left(\frac{\Delta}{2}-\frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_1}}\right)^2\right)}{1-\exp\left(-\frac{2}{9}n_1\left(\frac{\Delta}{2}-\frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n_1}}\right)^2\right)},
\end{aligned}
$$

which proves the first part of the lemma.

The second part shares the same derivation as that of Lemma 1, by which we have for any $\delta>0$, with probability at least $1-\delta-\delta_{\text{fail}}$,

$$
\widehat{R}^{surr}_{tr,i_s}(\widehat{f}_{i_s})-\widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1)\le\frac{4-4\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{\sum_{t=1}^{T}n_t}}+2\sqrt{\frac{\log(2/\delta)}{2\sum_{t=1}^{T}n_t}}.
$$

Finally, notice the fact that

$$
\sum_{t=1}^{T}n_t=n_1\sum_{t=1}^{T}2^t=(2^{\lceil\log_2 K\rceil+1}-1)\left\lfloor\frac{B}{K\log_2 K}\right\rfloor\ge K\left\lfloor\frac{B}{K\log_2 K}\right\rfloor=\mathcal{O}\left(\left\lfloor\frac{B}{\log_2 K}\right\rfloor\right),
$$

which finishes the proof. □

A.6. Proof of Lemma 4

Proof of Lemma 4. If $\widehat{R}^{surr}_i(\widehat{f}_i)<\widehat{R}^{surr}_1(\widehat{f}_1)$, then it must be the case that either the estimation $\widehat{R}^{surr}_i(\widehat{f}_i)$ is over-optimistically, or the estimation $\widehat{R}^{surr}_1(\widehat{f}_1)$ is over-pessimistically. Let p_i be the probability that $\widehat{R}^{surr}_i(\widehat{f}_i)<\widehat{R}^{surr}_1(\widehat{f}_1)$, then p_i can be bounded by

$$
\begin{aligned}
p_i &\le \Pr\left[\left(\widehat{R}^{surr}_i(\widehat{f}_i)<\widehat{R}^{surr}_{tr,i}(\widehat{f}^*_i)-\frac{\Delta_i}{2}\right)\vee\left(\widehat{R}^{surr}_1(\widehat{f}_1)>\widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1)+\frac{\Delta_i}{2}\right)\right]\\
&\le \underbrace{\Pr\left[\widehat{R}^{surr}_i(\widehat{f}_i)<\widehat{R}^{surr}_{tr,i}(\widehat{f}^*_i)-\frac{\Delta_i}{2}\right]}_{\texttt{term (a)}}+\underbrace{\Pr\left[\widehat{R}^{surr}_1(\widehat{f}_1)>\widehat{R}^{surr}_{tr,1}(\widehat{f}^*_1)+\frac{\Delta_i}{2}\right]}_{\texttt{term (b)}},
\end{aligned}
$$

where $\Delta_i=\widehat{R}^{surr}_{tr,i}(\widehat{f}^*_i)-\min_{j\in[K]}\widehat{R}^{surr}_{tr,j}(\widehat{f}^*_j)$ is defined in (7). We next explain how to upper bound `term (a)`, and the bound on `term (b)` follows a similar derivation. First notice that

$$
\begin{aligned}
\widehat{R}^{surr}_{tr,i}(\widehat{f}^*_i)-\widehat{R}^{surr}_i(\widehat{f}_i)&=\left(\widehat{R}^{surr}_{tr,i}(\widehat{f}^*_i)-\widehat{R}^{surr}_{tr,i}(\widehat{f}_i)\right)+\left(\widehat{R}^{surr}_{tr,i}(\widehat{f}_i)-\widehat{R}^{surr}_i(\widehat{f}_i)\right)\\
&\le \widehat{R}^{surr}_{tr,i}(\widehat{f}_i)-\widehat{R}^{surr}_i(\widehat{f}_i),
\end{aligned}
$$

which is exactly a margin-based generalization bound. Define $\widetilde{\mathcal{F}}_i=\left\{\ell_{surr}\circ f\mid f\in\mathcal{H}_i\times\mathcal{G}_i\right\}$, standard generalization bound based on Rademacher complexity shows that for any $\delta>0$, with probability at least $1-\delta$,

$$
\begin{aligned}
\widehat{R}^{surr}_{tr,i}(\widehat{f}_i)-\widehat{R}^{surr}_i(\widehat{f}_i)&\le 2\mathfrak{R}_m(\widetilde{\mathcal{F}})+\sqrt{\frac{\log(1/\delta)}{2n}}\\
&\le \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n}}+\sqrt{\frac{\log(1/\delta)}{2n}},
\end{aligned}
$$

where the second inequality is due to [35, Theorem 5]. Since $n>\frac{16((1-\theta)\kappa\Lambda)^2}{((1-2\theta)\Delta)^2}$, we have $\frac{\Delta_i}{2}\ge\frac{\Delta}{2}>\frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n}}$, so the generalization error bound above can be translated as

$$\Pr\left[\widehat{R}^{surr}_{tr,i}(\widehat{f}_i) - \widehat{R}^{surr}_{i}(\widehat{f}_i) > \frac{\Delta_i}{2}\right] \leq 2\exp\left(-\frac{2}{9}n\left(\frac{\Delta_i}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n}}\right)^2\right).$$

Since $\widehat{R}^{surr}_{tr,i}(\widehat{f}^*_i) - \widehat{R}^{surr}_{i}(\widehat{f}_i) \leq \widehat{R}^{surr}_{tr,i}(\widehat{f}_i) - \widehat{R}^{surr}_{i}(\widehat{f}_i)$, we conclude that

$$\texttt{term (a)} \leq 2\exp\left(-\frac{2}{9}n\left(\frac{\Delta_i}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n}}\right)^2\right).$$

Similarly, we have

$$\texttt{term (b)} \leq 2\exp\left(-\frac{2}{9}n\left(\frac{\Delta_i}{2} - \frac{2-2\theta}{1-2\theta}\sqrt{\frac{(\kappa\Lambda)^2}{n}}\right)^2\right),$$

and the proof is finished. □

References

[1] E. Horvitz, Artificial intelligence in the open world, in: AAAI 2008 Presidential Address, 2008.
[2] T.G. Dietterich, Steps toward robust artificial intelligence, AI Mag. (2017) 3–24.
[3] Z.-H. Zhou, Open-environment machine learning, Nat. Sci. Rev. 9 (2022) nwac123.
[4] W.J. Scheirer, A. de Rezende Rocha, A. Sapkota, T.E. Boult, Toward open set recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2013) 1757–1772.
[5] W.J. Scheirer, L.P. Jain, T.E. Boult, Probability models for open set recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2014) 2317–2324.
[6] Q. Da, Y. Yu, Z.-H. Zhou, Learning with augmented class by exploiting unlabeled data, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI), 2014, pp. 1760–1766.
[7] S. Liu, R. Garrepalli, T.G. Dietterich, A. Fern, D. Hendrycks, Open category detection with PAC guarantees, in: Proceedings of the 35th International Conference on Machine Learning (ICML), 2018, pp. 3175–3184.
[8] Y.-J. Zhang, P. Zhao, L. Ma, Z.-H. Zhou, An unbiased risk estimator for learning with augmented classes, in: Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 10247–10258.
[9] P. Langley, Open-world learning for radically autonomous agents, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI), 2020, pp. 13539–13543.
[10] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. (2010) 1345–1359.
[11] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (2014) 44.
[12] P. Zhao, Y.-J. Zhang, L. Zhang, Z.-H. Zhou, Adaptivity and non-stationarity: problem-dependent dynamic regret for online convex optimization, ArXiv preprint, arXiv:2112.14368, 2021.
[13] B.-J. Hou, L. Zhang, Z.-H. Zhou, Learning with feature evolvable streams, in: Advances in Neural Information Processing Systems (NIPS), vol. 30, 2017, pp. 1417–1427.
[14] C. Hou, Z.-H. Zhou, One-pass learning with incremental and decremental features, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 2776–2792.
[15] Z.-Y. Zhang, P. Zhao, Y. Jiang, Z.-H. Zhou, Learning with feature and distribution evolvable streams, in: Proceedings of the 37th International Conference on Machine Learning (ICML), 2020, pp. 11317–11327.
[16] C. Geng, S.-J. Huang, S. Chen, Recent advances in open set recognition: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2020) 3614–3631.
[17] J. Attenberg, P. Ipeirotis, F. Provost, Beat the machine: challenging humans to find a predictive model's unknown unknowns, ACM J. Data Inf. Qual. (2015) 1–17.
[18] H. Lakkaraju, E. Kamar, R. Caruana, E. Horvitz, Identifying unknown unknowns in the open world: representations and policies for guided exploration, in: Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI), 2017, pp. 2124–2132.
[19] G. Bansal, D.S. Weld, A coverage-based utility model for identifying unknown unknowns, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), 2018, pp. 1463–1470.
[20] S. Saisubramanian, E. Kamar, S. Zilberstein, A multi-objective approach to mitigate negative side effects, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI), 2020, pp. 354–361.
[21] S. Saisubramanian, S. Zilberstein, E. Kamar, Avoiding negative side effects due to incomplete knowledge of AI systems, AI Mag. 42 (2021) 62–71.
[22] B. Settles, Active Learning, Morgan & Claypool Publishers, 2012.
[23] F.M. Zanzotto, Human-in-the-loop artificial intelligence, J. Artif. Intell. Res. 64 (2019) 243–252.
[24] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, L. He, A survey of human-in-the-loop for machine learning, Future Gener. Comput. Syst. 135 (2022) 364–381.
[25] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Adv. Neural Inf. Process. Syst. 35 (2022) 27730–27744.
[26] M. Njoo, T. De Jong, Exploratory learning with a computer simulation for control theory: learning processes and instructional support, J. Res. Sci. Teach. (1993) 821–844.
[27] J.M. Spector, M.D. Merrill, J. Elen, M. Bishop, Handbook of Research on Educational Communications and Technology, fourth ed., Springer, 2014.
[28] M. Donoso, A.G. Collins, E. Koechlin, Foundations of human reasoning in the prefrontal cortex, Science 344 (2014) 1481–1486.
[29] A.E. Stahl, L. Feigenson, Observing the unexpected enhances infants' learning and exploration, Science 348 (2015) 91–94.
[30] C. Zhang, W. Wang, X. Qiao, On reject and refine options in multicategory classification, J. Am. Stat. Assoc. (2018) 730–745.
[31] C. Ni, N. Charoenphakdee, J. Honda, M. Sugiyama, On the calibration of multiclass classification with rejection, in: Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 2582–2592.
[32] S. Kalyanakrishnan, A. Tewari, P. Auer, P. Stone, PAC subset selection in stochastic multi-armed bandits, in: Proceedings of the 29th International Conference on Machine Learning (ICML), 2012, pp. 655–662.
[33] L. Chen, J. Li, M. Qiao, Nearly instance optimal sample complexity bounds for top-k arm selection, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017, pp. 101–110.
[34] Z. Qin, X. Gan, J. Liu, H. Wu, H. Jin, L. Fu, Exploring best arm with top reward-cost ratio in stochastic bandits, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 159–168.
[35] C. Cortes, G. DeSalvo, M. Mohri, Learning with rejection, in: Proceedings of International Conference on Algorithmic Learning Theory (ALT), 2016, pp. 67–82.

[36] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, Adaptive Computation and Machine Learning Series, MIT Press, 2002.
[37] T. Lattimore, C. Szepesvári, Bandit Algorithms, Cambridge University Press, 2019.
[38] E. Even-Dar, S. Mannor, Y. Mansour, Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems, J. Mach. Learn. Res. 7 (2006) 1079–1105.
[39] S. Ben-David, E. Dichterman, Learning with restricted focus of attention, in: Proceedings of the Sixth Annual Conference on Computational Learning Theory (COLT), 1993, pp. 287–296.
[40] E. Hazan, T. Koren, Linear regression with limited observation, in: Proceedings of the 29th International Conference on International Conference on Machine Learning (ICML), 2012, pp. 1865–1872.
[41] D. Kukliansky, O. Shamir, Attribute efficient linear regression with distribution-dependent sampling, in: Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015, pp. 153–161.
[42] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012.
[43] T. Sun, Z.-H. Zhou, Structural diversity for decision tree ensemble learning, Front. Comput. Sci. (2018) 560–570.
[44] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[45] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artif. Intell. 137 (2002) 239–263.
[46] O. Bousquet, S. Boucheron, G. Lugosi, Introduction to statistical learning theory, in: Advanced Lectures on Machine Learning (Machine Learning Summer Schools 2003), 2003, pp. 169–207.
[47] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, second ed., The MIT Press, 2018.
[48] B. Hibbard, Avoiding unintended AI behaviors, in: Artificial General Intelligence: 5th International Conference, AGI 2012, Proceedings 5, Oxford, UK, December 8-11, 2012, Springer, 2012, pp. 107–116.
[49] D. Hadfield-Menell, S. Milli, P. Abbeel, S.J. Russell, A. Dragan, Inverse reward design, Adv. Neural Inf. Process. Syst. 30 (2017).
[50] S. Zhang, E.H. Durfee, S. Singh, Minimax-regret querying on side effects for safe optimality in factored Markov decision processes, in: IJCAI, 2018, pp. 4867–4873.
[51] A. Turner, N. Ratzlaff, P. Tadepalli, Avoiding side effects in complex environments, Adv. Neural Inf. Process. Syst. 33 (2020) 21406–21415.
[52] P. Melville, F.J. Provost, R.J. Mooney, An expected utility approach to active feature-value acquisition, in: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM), 2005, pp. 745–748.
[53] A. Dhurandhar, K. Sankaranarayanan, Improving classification performance through selective instance completion, Mach. Learn. (2015) 425–447.
[54] S. Huang, M. Xu, M. Xie, M. Sugiyama, G. Niu, S. Chen, Active feature acquisition with supervised matrix completion, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), 2018, pp. 1571–1579.
[55] C.K. Chow, On optimum recognition error and reject tradeoff, IEEE Trans. Inf. Theory (1970) 41–46.
[56] M. Yuan, M.H. Wegkamp, Classification methods with reject option based on convex risk minimization, J. Mach. Learn. Res. (2010) 111–130.
[57] C. Cortes, G. DeSalvo, M. Mohri, Boosting with abstention, in: Advances in Neural Information Processing Systems (NIPS), vol. 29, 2016, pp. 1660–1668.
[58] W. Wang, X. Qiao, Learning confidence sets using support vector machines, in: Advances in Neural Information Processing Systems (NeurIPS), vol. 31, 2018, pp. 4934–4943.
[59] H. Shim, S.J. Hwang, E. Yang, Joint active feature acquisition and classification with variable-size set encoding, in: Advances in Neural Information Processing Systems (NeurIPS), vol. 31, 2018, pp. 1375–1385.
[60] S. Kothawade, S. Chopra, S. Ghosh, R. Iyer, Active data discovery: mining unknown data using submodular information measures, arXiv preprint, arXiv: 2206.08566, 2022.
[61] R. Herbei, M.H. Wegkamp, Classification with reject option, Can. J. Stat. (2006) 709–721.
[62] P.L. Bartlett, M.H. Wegkamp, Classification with a reject option using a hinge loss, J. Mach. Learn. Res. (2008) 1823–1840.
[63] O. Bousquet, N. Zhivotovskiy, Fast classification rates without standard margin assumptions, Inf. Inference 10 (2021) 1389–1421.
[64] M. van Breukelen, R.P.W. Duin, D.M.J. Tax, J.E. den Hartog, Handwritten digit recognition by combined classifiers, Kybernetika (1998) 381–386.
[65] O. Baños, M. Damas, H. Pomares, I. Rojas, M.A. Tóth, O. Amft, A benchmark dataset to evaluate sensor displacement in activity recognition, in: Proceedings of the 12th ACM Conference on Ubiquitous Computing, 2012, pp. 1026–1035.
[66] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[67] Z.-H. Zhou, Rehearsal: learning from prediction to decision, Front. Comput. Sci. 16 (2022) 164352.