Teacher–Explorer–Student Learning: A Novel Learning Method for Open Set Recognition

Jaeyeon Jang and Chang Ouk Kim

Abstract—When an unknown example, one that was not seen during training, appears, most recognition systems usually produce overgeneralized results and determine that the example belongs to one of the known classes. To address this problem, teacher–explorer–student (T/E/S) learning, which adopts the concept of open set recognition (OSR) to reject unknown samples while minimizing the loss of classification performance on known samples, is proposed in this study. In this novel learning method, the overgeneralization of deep-learning classifiers is significantly reduced by exploring various possibilities for unknowns. The teacher network extracts hints about unknowns by distilling the pretrained knowledge about knowns and delivers this distilled knowledge to the student network. After learning the distilled knowledge, the student network shares its learned information with the explorer network. Next, the explorer network shares its exploration results by generating unknown-like samples and feeding those samples to the student network. As this alternating learning process is repeated, the student network experiences a variety of synthetic unknowns, reducing overgeneralization. The results of extensive experiments show that each component proposed in this article significantly contributes to improving OSR performance. It is found that the proposed T/E/S learning method outperforms current state-of-the-art methods.

Index Terms—Exploration, generative adversarial learning, knowledge distillation, open set recognition (OSR), overgeneralization.

I. INTRODUCTION

RECOGNITION systems have greatly improved in performance due to recent advancements in deep learning [1], [2]. However, many challenges remain in applying deep-learning techniques to real-world problems. One of the main challenges is that most recognition systems have been designed under closed-world assumptions in which all categories are known a priori. However, samples that are unknown from the training phase may be fed into these systems during the testing phase. In this article, unknown refers to samples belonging to “unknown unknown” classes as defined in [3], namely, classes without either any representative examples or any side information available during training. When such an unknown sample appears, traditional recognition systems will incorrectly identify the sample as belonging to one of the classes learned during training. To overcome this problem, the concept of open set recognition (OSR), which aims to correctly classify samples of known classes while rejecting unknown samples, has been proposed [4]. In addition, OSR has been introduced in many application areas, including autonomous driving [5], network intrusion detection [6], [7], defect classification [8], [9], and social media forensics [10].

Most existing discriminative models, including deep neural networks (DNNs), suffer from the problem of overgeneralization in open set scenarios [11]. Here, overgeneralization refers to the situation in which a discriminative model determines with high confidence that unknown samples belong to known classes. Many studies have tried to mitigate the overgeneralization problem in OSR [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. Despite the above performance improvements, many OSR methods remain affected by overgeneralization. The reason is that learning from only given known samples, regardless of the type of model used, has limitations in reducing overgeneralization.

Given the infinite diversity of unknowns, the learning process in OSR systems must be able to explore various possibilities for unknowns. In this article, we propose a teacher–explorer–student (T/E/S) learning method for this purpose. Let us suppose that there exists a student network without any knowledge and a teacher network that has been pretrained and is assumed to have knowledge of the known classes. This teacher network is also assumed to consider various possibilities for unknowns. To derive such possibilities, the teacher must teach not only the original class information of a given example but also the uncertainty that may be inherent in the example. Here, uncertainty is defined as the possibility of belonging to an unknown class. Thus, the teacher distills the available information while extracting uncertainty from the example.

The teacher can provide slight hints about unknowns. However, such hints are not sufficient for learning the various possibilities for unknowns. Therefore, we introduce an explorer, a generative adversarial network (GAN) that performs exploration to produce unknown-like examples based on the shared information that the student has learned. Finally, the student receives distilled information on knowns and explored knowledge of unknowns, and both are used to reduce
overgeneralization. By repeating this alternating learning process, the student experiences various possibilities for unknowns. In addition, we use an architecture in which a set of one-versus-rest networks (OVRNs) follow a convolutional neural network (CNN) feature extractor to enable the student network to establish more sophisticated decision boundaries for OSR [14].

Extensive experiments were performed to evaluate the proposed T/E/S learning model. The experimental results showed that the teacher’s distilled knowledge reduced overgeneralization. In addition, the explorer generated realistic but unknown-like synthetic samples and provided them to the student network. As a result, the proposed method outperformed state-of-the-art methods for OSR in various open-set scenarios.

II. BACKGROUND AND RELATED STUDIES

A. Open Set Recognition

The OSR problem was formalized in [4] as the problem of finding a measurable recognition function that minimizes the open set risk, consisting of open space risk and empirical risk. Here, the open space risk is a relative measure of positively labeled open space, which is far from any known training samples, compared to the overall measure of positively labeled space, while the empirical risk represents the loss of classification performance on known samples. In the early days of OSR research, some shallow machine-learning models were redesigned to introduce open-set risk minimization in modeling. For example, Scheirer et al. [4] proposed a one-versus-set machine, a variant of a support vector machine (SVM), by introducing an open set risk minimization term into linear kernel SVM modeling. Similarly, Cevikalp [22] applied the intuitive idea of training a classwise hyperplane to be as close as possible to the target class samples and as far as possible from samples of other classes. Scheirer et al. [23] introduced a statistical extreme value theory (EVT) [24] to calibrate the decision scores of a radial basis function SVM based on the distribution of extreme scores. In addition, these researchers developed a compact abating probability model based on a one-class SVM to manage open space risk. Zhang and Patel [25] proposed a sparse representation-based OSR method based on their observation that discriminative information is hidden in the tails of matched and nonmatched reconstruction error distributions.

Over the past few years, deep-learning techniques have led to advancements in OSR systems. Most methods in this category have focused on mitigating the overgeneralization of general discriminative DNNs, which usually generate excessive open space [11]. The first deep model introduced for OSR was OpenMax, which models a class-specific representation distribution in the penultimate layer of a CNN and computes a regularized confidence score by applying an EVT-based calibration strategy [12]. Shu et al. [13] replaced a softmax layer with a sigmoid layer, such that the output nodes of the latter could make their own class-specific determinations. These scholars additionally used Gaussian fitting to obtain class-specific rejection/acceptance thresholds to tighten the decision boundaries. Jang and Kim [14] showed that the collective decisions of the OVRNs can be used to establish more sophisticated decision boundaries to reduce redundant open space. Recently, Vaze et al. [26] demonstrated that the performance of existing closed-set classifiers in OSR can be significantly enhanced by leveraging techniques such as a deeper network, longer training, better augmentations, and label smoothing.

In the field of prototype learning, Yang et al. [27] proposed a convolutional prototype network (CPN) that learned several prototypes for all known classes while leaving room for unknown samples in an embedding space. Chen et al. [28] further enhanced this concept by leveraging generative adversarial learning to address the problem that the unknown space is not well modeled during training using existing prototype learning techniques.

Several researchers have used reconstructive or generative models to calibrate the confidence scores of discriminative DNNs. For instance, Yoshihashi et al. [15] proposed a deep hierarchical reconstruction network (DHRNet) that combines classification and reconstruction networks. These researchers expanded the OpenMax model by additionally utilizing the hierarchical latent representations of DHRNet. Perera et al. [29] recently incorporated a self-supervision technique into a model composed of reconstructive and classification networks to improve the separation of classes from each other and from unknown classes. Ge et al. [16] further enhanced the OpenMax model by utilizing synthetic samples generated by a conditional GAN. Neal et al. [17] proposed an encoder–decoder GAN to generate counterfactual samples and retrained a pretrained CNN to classify the generated samples as unknown samples. Lee et al. [18] introduced a new term into the GAN loss function with the aim of generating synthetic unknown samples. This term compels the generated samples to not belong to any known classes by producing the same probability for each known class. Nevertheless, although generators have been used in a few existing studies to produce synthetic samples, the models in these studies were built based on a certain restrictive assumption for synthetic sample generation: instead of assuming the infinity of the unknowns, the generators in the existing studies were able only to produce samples limited to a small portion of the open space [3].

In a preliminary study, we compared our proposed T/E/S method with other GAN-based OSR methods, as shown in Fig. 1. We found that compared to our method, G-OpenMax generates less diverse samples, some of which closely resemble known samples. This occurs because G-OpenMax does not require that the generated samples be distinct from known classes, nor does it demand diversity. Although training confidence-calibrated classifiers (TCCC) produces more diverse samples than G-OpenMax, a considerable number remain close to known samples and do not extend beyond the bounds of known classes into the edge area. This arises from the authors’ assumption that unknown samples are those sharing features with multiple known classes. In contrast, our proposed learning method presumes that the generated samples should be distinct from all known classes and that unknown samples should also be distinct from each other.
To address this issue, Kong and Ramanan [30] recently proposed OpenGAN, which utilizes the hidden representations generated by a closed-set classifier for discriminator training. They demonstrated that the proposed model outperforms state-of-the-art OSR models. However, OpenGAN generates and uses synthesized adversarial samples by leveraging outliers, which are generally assumed to belong to unknown classes in an open set setting.

Several two-stage methods that sequentially perform the tasks of unknown detection and closed-set classification have recently been developed. Oza and Patel [19] proposed a network configuration in which a decoder and a classifier follow a shared encoder for reconstruction and classification. These scholars modeled the tail of the reconstruction error distribution with EVT to compute the unknown detection score. Finally, the classifier assigns one class among the known classes to each sample determined to be known. In a subsequent study [20], these scholars extended the decoder to a class-conditioned decoder and named their model class-conditional auto-encoder (C2AE). Sun et al. [21] proposed a conditional Gaussian distribution learning (CGDL) method in which a class-conditional posterior distribution is generated in the latent space using a variational autoencoder that follows classwise multivariate Gaussian models. The learned features are fed into two models: an unknown detector and a closed-set classifier. The unknown detector identifies unknowns based on the set of classwise Gaussian cumulative probabilities and the reconstruction errors of the variational autoencoder.

**B. One-Versus-Rest Networks**

The softmax function is the de facto standard activation function used for multiclass classification; it measures the relative likelihood of a known class compared to the other known classes. Due to this characteristic, when an unknown sample is analyzed, a network with a softmax output layer is trained to choose the best-matching class instead of rejecting the sample [31]. In other words, a network with softmax activation is at high risk of assigning a high confidence score to an unknown by selecting the most similar class among all known classes. However, when the sigmoid activation function is used in the output layer instead, each sigmoid output is not conditioned on the other outputs. Rather, each sigmoid output is trained to discriminate a dissimilar example from matching examples, allowing the output nodes for all classes to independently reject unknown examples. Thus, by combining multiple class-specific determinations into the collective decision, more sophisticated decision boundaries for rejection can be established. In addition, the overgeneralization problem can be further mitigated by using a set of OVRNs as the output layer instead of a single sigmoid layer [14]. Hence, in our student network, we apply a structure in which OVRNs follow a CNN feature extractor.

**C. Teacher–Student Learning**

In this article, teacher–student (T/S) learning [32], [33] is extended to OSR. Therefore, in this section, we briefly introduce the original concept of T/S learning. Recent top-performing DNNs usually involve very wide and deep (heavy) structures with numerous parameters. T/S learning, often called knowledge distillation, was proposed to reduce the computational burden of inference imposed by such a heavy structure.

In the original T/S learning approach, the knowledge of a heavy teacher network is transferred to a relatively light student network. The student network is penalized to learn a softened version of the teacher’s output. Learning this soft target guides the student to capture the finer structure learned by the teacher [32]. Generally, through softmax activation, a neural network produces posterior class probabilities for a sample \( x \), as follows:

\[
P(y|x_\tau, \theta_T) = \frac{\exp(l'_{y_\tau})}{\sum_{l_\in \mathcal{Y}} \exp(l'_{l})} \quad \forall y \in \mathcal{Y}
\]

where \( \theta_T \) is the set of parameters of the teacher network \( T \), \( l'_{y_\tau} \) is the logit of class \( y \), and \( \mathcal{Y} \) is the set of known classes. To produce soft targets, the class probabilities are scaled by a temperature \( \tau \) as follows:

\[
P^\tau(y|x_\tau, \theta_T) = \frac{\exp(l'_{y_\tau}/\tau)}{\sum_{l_\in \mathcal{Y}} \exp(l'_{l}/\tau)} \quad \forall y \in \mathcal{Y}.
\]

In T/S learning, additional semantic information is provided to the student network by increasing the probabilities of nontarget classes. Interestingly, the student can then recognize samples of unseen classes only by learning softened probabilities of examples of the seen classes as long as the...
The teacher has knowledge about the unseen classes [33]. The reason is that the teacher assigns the seen examples a small possibility of belonging to unseen classes, and the student can infer knowledge about the unseen classes from that small possibility. Without loss of generality, the student network can thus obtain some hints for recognizing unknowns if the teacher network can discover the uncertainties inherent in the known samples.

III. PROPOSED METHOD

Fig. 2 shows an overview of the proposed T/E/S learning method. First, the teacher network is pretrained to provide $[P(y|x_i, \theta_T)\forall y \in \mathcal{Y}]$, the posterior probabilities of the known classes. Next, the probabilities are calibrated to assign softened probabilities for the known classes and hints for $U$, which represents all unknown classes. Here, $U$ is used to indicate an additional “unknown” class in the student network. For this calibration, a novel hint-extracting knowledge distillation (HE-KD) method is proposed. Intuitively, the student network should be able to recognize an unknown sample well after learning sufficient diverse possibilities for unknowns. However, the HE-KD method provides only small hints about unknowns. To address this problem, an explorer network that explores open space and generates unknown-like open set examples is also proposed. The role of the explorer is to support the student by discovering new open set examples based on the student’s current knowledge about $U$. Thus, the student and explorer are trained together alternately.

Through T/E/S learning, the student learns not only the information about “knowns” distilled by the teacher but also the information about “unknowns” explored by the explorer. In every iteration, real known samples and generated unknown samples are fed into the student network. The student network is trained to produce a soft probability vector $[P^d(y|x_i, \theta_T)\forall y \in \mathcal{Y} \cup \{U\}]$ via HE-KD for the known training samples by minimizing $\mathcal{L}_{TS}$, which is the loss between $[P(y|x_i, \theta_T)\forall y \in \mathcal{Y} \cup \{U\}]$ and $[P^d(y|x_i, \theta_T)\forall y \in \mathcal{Y} \cup \{U\}]$. In addition, for the generated unknown samples, the student network is trained to minimize $\mathcal{L}_{DS}$, which is the loss between $[P(y|x_i, \theta_D)\forall y \in \mathcal{Y} \cup \{U\}]$ and the hard label for $U$.

A. Teacher Network: HE-KD

Let $x_i \in \mathcal{X}$ be an input sample from any of the known classes, where $\mathcal{X}$ is the input space, and let $t_i \in \mathcal{Y}$ be its corresponding ground-truth label. Then, the teacher network is first trained to minimize the following categorical cross-entropy loss:

$$\mathcal{L}_T(\theta_T) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{y \in \mathcal{Y}} \mathbb{I}(t_i = y) \log P(y|x_i, \theta_T) \quad (3)$$

where $N$ is the batch size, $\mathbb{I}$ is an indicator function, and $q_{i,y}$ is the posterior probability of sample $x_i$ for class $y$.

After the teacher network has been trained, it extracts uncertainty from the training samples. The teacher considers a training sample to be more uncertain if that sample has a lower probability for the target class. However, the trained teacher network provides a very high target class score for most training samples, leaving minimal differences among the samples. For example, if no temperature scaling is used, the average confidence scores for the target class are over 0.998 for the “0,” “1,” and “2” classes of MNIST, as shown in Fig. 3. Thus, we produce a scaled probability vector $[P^t(y|x_i, \theta_T)\forall y \in \mathcal{Y}]$ for $x_i$ by applying temperature scaling as expressed in (2). Afterward, a more relaxed probabilistic interpretation becomes possible. For example, the “0” class can be interpreted as having a probability of 87.3% for class “0,” 3.9% for the second highest class, 2.1% for the third highest class, and 6.7% for the remaining seven-digit classes after being scaled with $\tau = 5$. This probabilistic interpretation can also support the student network in inferring non-target classes without being trained on those classes, as described in Section II-C. However, the aforementioned probabilistic interpretation is no longer valid if we assume an open-set scenario. The reason is that the probability of not belonging to the target class encompasses not only the probability of belonging to one of the other known classes but also the uncertainty. It is impossible to obtain exact information about unknowns from the probabilistic interpretation; however, hints for inferring unknowns can be extracted.

It is unnecessary to guide the student to learn the scaled scores for all known classes, including nontarget classes, following the original goal of T/S learning. Rather, it is better to utilize the scaled scores to produce softened target class scores for closed-set classification and hints for unknown detection. To this end, through HE-KD, the distilled target class probability $P^t(t|x_i, \theta_T)$ and the uncertainty $P^d(U|x_i, \theta_T) = 1 - P^t(t|x_i, \theta_T)$ are regressed based on $P^t(t|x_i, \theta_T)$, where $t_i$ is the ground-truth label. The elements of the known nontarget classes are set to zero. Let the training data be split into $D_c$ and $D_u$, which are the set of examples correctly classified by the teacher and the set of misclassified examples, respectively. Let $S_D$ be the set of scaled target probabilities $P^t(t|x_i, \theta_T)$ for $x_i \in D_c$. Then, $P^d(t|x_i, \theta_T)$ is computed as follows:

$$P^d(t|x_i, \theta_T) = \begin{cases} 
\min(d_{\text{min}} + (1 - d_{\text{min}})N(i|S_D), \quad & \text{if } x_i \in D_c \\
\min(d_{\text{min}}, \quad & \text{otherwise}
\end{cases} \quad (4)$$

where $d_{\text{min}}$ is the minimum distilled probability for the target class and $N(i|S_D) = (P^t(t|x_i, \theta_T) - \min(S_T))/(\max(S_T) - \min(S_T))$. Here, $d_{\text{min}}$ is a parameter used to prevent the student network from learning too small a probability for the target class and losing its discriminative capability.

B. Explorer Network: Open Set Example Generation

The explorer network has a general GAN structure containing a generator and a discriminator. In the original form of GAN training, the goal of the generator is to produce fake samples such that the discriminator will be deceived into identifying the fake samples as real. In addition to this original goal, the generator of the explorer network is trained to generate unknown samples in open space that the student network will determine to be unknown samples, as shown
Fig. 2. Overview of T/E/S learning. Single arrows indicate the direction of information flow, while double arrows connect two elements of the loss function. Solid lines and dashed lines indicate the flows of real known samples and generated unknown samples, respectively. $\theta_T$ and $\theta_S$ are the sets of parameters of the teacher network (T) and student network (S), respectively.

C. Student Network: Learning From Known and Unknown-Like Samples

In T/E/S learning, the student network learns from real known samples and fake unknown samples generated by the explorer. For a real known sample $x_i$, the student is trained to predict $\mathbb{P}(y_i|x_i, \theta_T)$ based on the following binary cross-entropy loss function:

$$
\min_{\theta_S} \mathbb{E}_{x\sim\mathcal{X}} [\log(1 - D(G(z))) + \lambda L_{BCE}(y_U, S(G(z)))]
$$

where $\theta_G$ is the generator’s parameter set, $L_{BCE}(\cdot, \cdot)$ is the binary cross-entropy, $y_U = [0, 0, \ldots, 1]^T$ is the hard label of an unknown sample, and $\lambda$ is a balancing parameter.

The discriminator of the explorer is trained to discriminate between real known and generated unknown samples by updating $\theta_D$, the discriminator’s parameter set, based on (6). Because the discriminator learns the synthetic samples as generated unknowns and the training samples as real knowns, whenever synthetic examples are generated, the generator consistently tries to produce new realistic open set examples in every iteration. The objective function of this alternating training process is given in (7)

$$
\max_{\theta_D} \mathbb{E}_{x\sim\mathcal{X}} [\log(D(x))] + \mathbb{E}_{z\sim p_{\text{pri}}(z)} [\log(1 - D(G(z)))] + \mathbb{E}_{x\sim\mathcal{X}} [\log(D(G(z))] + \mathbb{E}_{z\sim p_{\text{pri}}(z)} [\log(1 - D(G(z)))] + \lambda L_{BCE}(y_U, S(G(z))]
$$

$$
\min_{\theta_G} \mathbb{E}_{x\sim\mathcal{X}} [\log(1 - D(G(z))) + \lambda L_{BCE}(y_U, S(G(z))]
$$

C. Student Network: Learning From Known and Unknown-Like Samples

In T/E/S learning, the student network learns from real known samples and fake unknown samples generated by the explorer. For a real known sample $x_i$, the student is trained to predict $[P^d(y|x_i, \theta_T)|_{y \in \mathcal{Y} \cup \{U\}}$ based on the following binary cross-entropy loss function:

$$
L_{T/S}(\theta_S) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{y \in \mathcal{Y}\cup\{U\}} [P^d(y|x_i, \theta_T)\log(P(y|x_i, \theta_S) + (1 - P^d(y|x_i, \theta_T))\times(1 - \log(P(y|x_i, \theta_S))]
$$

where $\theta_S$ denotes the student’s parameter set.

The student also learns from fake unknown samples $\tilde{x}_k = G(z_k)$, $z_k \sim p_{\text{pri}}(z)$. However, it is dangerous to train the student network on all $\tilde{x}_k$ with a hard label $y_U$. The reason is...
that during the competitive learning process of the explorer, the generator will sometimes produce known-like samples. Training on such samples as unknown samples can reduce performance in both closed-set classification and unknown detection. Thus, only unknown-like samples, which we call active unknown samples, should be used during the training of the student network. Active unknown samples are selected by means of an indicator function $A$, given as follows:

$$A(\tilde{x}_k) = \begin{cases} 
1, & \text{if } \max_{y \in \mathcal{Y}} P(y|\tilde{x}_k, \theta_S) < 1 - d_{\text{min}} \\
0, & \text{otherwise.}
\end{cases} \quad (9)$$

Here, since $d_{\text{min}}$ is the minimum distilled probability for known classes, the quantity $1 - d_{\text{min}}$ is the maximum uncertainty for known training samples. Once the generator has produced a number of synthetic samples equal to the size of an input batch of training samples, the student is trained to minimize the following loss:

$$\mathcal{L}_S(\theta_S) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{y \in \mathcal{Y} \cup \{U\}} \left[ P^d(y|x_i, \theta_T) \log P(y|x_i, \theta_S) + (1 - P^d(y|x_i, \theta_T)) \times (1 - \log P(y|x_i, \theta_S)) \right]$$

$$- \frac{1}{N_{\text{e}}} \sum_{k=1}^{N_{\text{e}}} \left[ A(\tilde{x}_k) \left( \log P(U|\tilde{x}_k, \theta_S) + \sum_{y \in \mathcal{Y}} (1 - \log P(y|\tilde{x}_k, \theta_S)) \right) \right] \quad (10)$$

where $N_{\text{e}}$ is the batch size for the explorer, representing the number of samples generated in each iteration.

The quantity of generated data from which the student network learns can be significant for OSR performance. In the proposed model, the quantity of generated data provided to the student network is parameterized by $d_{\text{min}}$. In general, if we use a low $d_{\text{min}}$, more generated data are supplied to the student network. The reason is that it is easier for the generated samples to satisfy the requirements of the student network by attaining lower confidence scores than $1 - d_{\text{min}}$ for known classes. In contrast, the higher $d_{\text{min}}$ is, the harder it is for the student network to learn from synthesized unknowns.

The student network is jointly trained with the explorer while sharing the learned information. The joint training process is summarized in Algorithm 1.

### D. OSR Rule

In this section, we propose a recognition rule based on the collective decisions of the OVRNs in the student network. A sample is more likely to belong to the target class if that sample has a high output score for the target class and low scores for the other classes. Furthermore, since nontarget OVRNs usually produce probabilities of zero for a sample, we compute the collective decision score based on the logits of the OVRNs as suggested in [14]. Let $l^j_i$ be the logit value of example $x_i$ for class $y$. Then, $cds^j_i$, the collective decision score for class $y$, is computed using the following simple function:

$$cds^j_i = l^j_i - \frac{1}{|\mathcal{Y}|} \sum_{j \neq y \in \mathcal{Y} \cup \{U\}} l^j_i \quad \forall y \in \mathcal{Y} \cup \{U\}. \quad (11)$$

In addition, the unknown probability score $P(U|x_i, \theta_S)$ can be used individually to supplement unknown detection because the OVRN for the unknown class $U$ is trained to discriminate between known samples and unknown-like samples. Thus, we propose the following OSR rule for both closed-set classification and unknown detection:

$$y^* = \begin{cases} 
\arg\max_{y \in \mathcal{Y} \cup \{U\}} cds^j_i, & \text{if } cds^j_i > e^y \text{ and } P(U|x_i, \theta_S) < \eta \text{ (optional)} \\
U, & \text{otherwise}
\end{cases} \quad (12)$$

where $e^y$ is the collective decision score threshold for class $y$ and $\eta$ is an uncertainty threshold. Empirically, it is not recommended to apply the condition $P(U|x_i, \theta_S) < \eta$ if it is not expected that there will be many unknown samples during the testing phase.

### IV. EXPERIMENTS

To evaluate the performance of the T/E/S learning method, extensive experiments were conducted on multiple benchmark datasets. First, we analyzed how the generated open-set examples affect learning in open-set scenarios. Next,
various ablation experiments were performed to validate the contribution of each component of the proposed learning method. Finally, we compared the proposed method with other state-of-the-art OSR methods.

A. Implementation Details

In this study, we utilized multiple small-scale benchmark image datasets as well as the large-scale ImageNet dataset. When using the small-scale benchmark datasets, we employed the two CNN configurations suggested in [14], one consisting of a set of OVRNs added to a plain CNN and the other being the redesigned VGNNet defined in [15], for the student networks. For the teacher networks, the original configurations suggested in [15] were used. Specifically, a plain CNN was used for training on the MNIST dataset, and the redesigned VGNNet was used for training on the other small-scale datasets. For ImageNet, a ResNet-50 [2] was employed as the backbone for both the teacher and student networks to ensure that they could be trained within a reasonable time. Finally, we used the architectures shown in Table I for the explorer networks. We used the Adam optimizer [34], with learning rates of 0.002 for the small-scale datasets and 0.00001 for ImageNet.

A class-specific threshold \( \epsilon \) was obtained by ensuring that 95% of the training data of class \( y \) were classified as belonging to class \( y \). The value of \( \epsilon \) was set to zero. \( \eta \) was set to ensure that 95% of the training data were recognized as known. A 95% threshold is commonly used in OSR studies [21]. Because the number of known classes ranged from 4 to 20 when using the small-scale benchmark datasets, the expected uncertainty from random guessing was between 0.05 and 0.2. For instance, if we have four known classes and additionally include an “unknown” class, the expected uncertainty is 0.2. The maximum uncertainty \((1 - d_{\text{min}})\) should be higher than these values to provide hints for the “unknown” class. However, it should not be set too high to avoid losing the capability of closed-set classification. Therefore, the minimum distilled probability \(d_{\text{min}}\) in (4) and (9) was empirically set to 0.7 for the small-scale benchmark datasets. For the same reason, this hyperparameter was set to 0.99 when using the ImageNet dataset, for which the number of known classes was 1000. To set the temperature \(\tau\) in (2) and the balancing parameter \(\lambda\) in (5), we applied a cross-class validation framework [35] in which performance on the validation set is measured while leaving out a randomly selected subset of known classes as “unknown.” The results of sensitivity analyses for \(d_{\text{min}}\) and \(\tau\) can be found in Section IV-D. We conducted experiments using the Python TensorFlow library on a personal computer equipped with an Intel Core i9 processor running at 3.19 GHz, 64.0 GB of RAM, and an NVIDIA GeForce RTX 4090.

B. Effects of Open Set Example Generation

In this section, we discuss the impact of the explorer network by analyzing the synthetically generated unknown samples. To this end, we present samples generated by the explorer. Moreover, we examine how both real and fake unknown samples are identified by a trained student network.

The T/E/S networks were trained on the MNIST dataset, and Fig. 5 shows the corresponding changes in the examples generated by the explorer. These examples were produced by feeding 15 fixed latent vectors into the explorer network after every training epoch. Interestingly, after only a few epochs, the generator produced digit-like images consisting of an unknown character and a black background. In addition, the generator continuously changed the pattern of the synthetic samples throughout the training period. Thus, in T/E/S learning, the explorer can provide the student with a variety of unknown-like examples that change slightly with each iteration. It was also confirmed that the T/E/S learning process converged even when learning from open set examples that changed continuously. Fig. 6 shows that the explorer easily reached a learning equilibrium state in which the generator and discriminator were competing with almost equal strength. Similarly, Fig. 7 shows that the loss of the student network converged well during training on both real known and fake unknown samples and that the generator could easily produce unknown-like samples to satisfy the student’s needs.

An experiment was designed to analyze whether the generated samples could effectively represent unknowns in

| TABLE I | EXPLORER NETWORK ARCHITECTURES |
|------------------|------------------|
| **Generator** | **Discriminator** |
| MNIST | MNIST |
| Input: 100 | Input: (28, 28, 1) |
| FC(7 \times 7 \times 128) | C(64, 3, 2) |
| R(7, 7, 128) | C(64, 3, 2) |
| TC(128, 4, 2) | FC(1) |
| TC(128, 4, 2) | Output: 1 |
| C(1, 7, 1) | |
| Output: (28, 28, 1) | |

| Other small-scale datasets | | |
|------------------|------------------|
| **Generator** | **Discriminator** |
| Input: 100 | Input: (32, 32, 3) |
| FC(4 \times 4 \times 256) | C(64, 3, 2) |
| R(4, 4, 256) | C(128, 3, 2) |
| TC(128, 4, 2) | C(128, 3, 2) |
| TC(128, 4, 2) | C(256, 3, 2) |
| TC(128, 4, 2) | FC(1) |
| C(3, 3, 1) | |
| Output: (32, 32, 3) | |

| ImageNet | | |
|------------------|------------------|
| **Generator** | **Discriminator** |
| Input: 1000 | Input: (224, 224, 3) |
| FC(14 \times 14 \times 64) | C(64, 4, 2) |
| R(14, 14, 64) | C(64, 4, 2) |
| TC(64, 4, 2) | C(128, 4, 2) |
| TC(128, 4, 2) | C(256, 4, 2) |
| TC(256, 4, 2) | C(512, 4, 1) |
| TC(512, 4, 2) | C(1, 4, 1) |
| C(3, 3, 1) | FC(1024) |
| Output: (32, 32, 3) | FC(1) |
| Output: 1 | |

**FC(x)** is a fully connected layer with \( x \) nodes. \( R \) is a reshape layer. \( C(x, y, z) \) and \( TC(x, y, z) \) are a convolutional layer and a transposed convolutional layer, respectively, each with \( x \) filters, a \( y \times y \) kernel, and a stride of \( z \). Sigmoid activation is used in the output layer, and leaky ReLU activation is used in the other layers.
the learned latent space of the student network. In this experiment, T/E/S learning was applied to the MNIST dataset. For the unknown class, we used two datasets of grayscale images, Omniglot [36] and MNIST-Noise (see Fig. 8). The latter dataset was constructed by superimposing the test samples of MNIST onto a set of noise images synthesized by independently sampling each pixel value from a uniform distribution on [0, 1]. We randomly selected 1000 samples from each known class and each unknown dataset. In addition, 1000 samples were generated by the explorer.

The results of visualizing the learned latent representations of the known samples, the real unknown samples, and the generated samples with t-distributed stochastic neighbor embedding (t-SNE) [37] are shown in Fig. 9. The generated samples are clearly separated from the known classes, showing that the generator of the explorer mostly produces samples in open space. For Omniglot, most of the unknown samples are very close to the generated ones, resulting in overlapping regions, whereas only a few samples are close to samples of known classes. In addition, the MNIST-noise samples are closer to the cluster of generated samples than to any of the known classes even though they look like MNIST samples. The reason is that T/E/S learning builds very tight class-specific decision boundaries to discriminate fake samples with similar appearances. The results indicate that the explorer can effectively generate unknown-like samples.

The student network was trained to classify known classes and “unknowns,” where a single class was assigned to represent “unknowns.” Hence, it is natural that the generated samples form a single cluster. In addition, there is usually insufficient space remaining for unknown samples in a trained feature space because the trained network has not considered...
Fig. 9. t-SNE visualizations of known, generated, and unknown samples. We employed a set of OVRNs as the output layer. The output features from the backbone network preceding the OVRNs were used to generate the t-SNE visualizations. (a) Omniglot and (b) MNIST-Noise were used as sources of real unknown samples.

real unknowns. Thus, real unknown samples may be forced to lie close to each other within a small subspace even if they are quite different from each other. Nonetheless, the unknown samples are clustered in a small subspace that is far from any of the known classes and closest to the generated samples. This shows that learning from the generated samples can help in identifying unknown samples.

C. Ablation Study

We first conducted a qualitative analysis. The MNIST dataset was partitioned into six known classes (0–5) and four unknown classes (6–9). We trained a CNN with OVRNs (CNN-OVRN), equivalent to a student network only (without teacher and explorer networks), and our proposed model (T/E/S networks) on the training dataset for the known classes. A comparison between Fig. 10(a) and (b) shows that T/E/S learning can significantly reduce overgeneralization by assigning low confidence scores to unknowns. In addition, Fig. 10(c) indicates that most unknown samples were assigned significantly higher uncertainty scores than those of known samples even though the networks were never trained to have high uncertainty on real unknown samples. Specifically, approximately 14.2% of unknown samples scored higher than 0.9. This reveals that T/E/S learning can be applied to infer some information about unknowns without direct training on real unknown samples.

In a quantitative analysis, the effects of network composition were examined. In all the baseline models in this section except T, which consisted only of a CNN classifier with softmax activation, the collective decision scores were used for OSR. Henceforth, the CD is used to denote the collective decision score. Specifically, the following ten baselines were compared.

1) $T$: In this baseline, only the teacher network, a conventional softmax CNN, is used. A sample is rejected if it has a confidence score of less than the rejection threshold for every known class. We set the rejection threshold to 0.5, the most commonly used value.

2) $OVRN$-$CD$: This baseline is a CNN-OVRN with the collective decision method.

3) $E/T$-$CD$: To obtain this baseline, the student network is removed from the proposed model. In other words, the teacher network learns from real known samples and fake unknown samples while minimizing the categorical cross-entropy loss. Because only the teacher network learns from known samples, knowledge distillation is not applied.

4) $E/T$-$CDU$: In this baseline, the network composition and the training process are the same as in baseline 3, but the collective decision thresholds and an uncertainty threshold are used jointly in the decision rule.

5) $T/S$-$CD$: A pretrained teacher network is added to baseline 2 to provide the student network with hints regarding unknown samples by applying HE-KD.

6) $T/S$-$CDU$: An uncertainty threshold is used in addition to the collective decision thresholds in baseline 5.

7) $E/S$-$CD$: This baseline is different from baseline 5 in that an explorer network is used instead of a teacher network to support the student network by generating synthetic open set examples.

8) $E/S$-$CDU$: An uncertainty threshold is used in addition to the collective decision thresholds in baseline 7.

9) $T/E/S$-$CD$ (Proposed Method): In this baseline, the proposed T/E/S learning method is applied without an uncertainty threshold.

10) $T/E/S$-$CDU$ (Proposed Method): This baseline additionally introduces an uncertainty threshold into baseline 9.

The performance was evaluated in terms of the macroaverage $F_1$ score ($F_1$) for the known classes and the “unknown” class. We adopted the experimental settings suggested in [14] for a quantitative ablation study. In particular, we used four nonanimal classes from the CIFAR-10 dataset as known classes and 100 natural classes from the CIFAR-100 dataset as unknown classes. OSR performance is significantly affected by the ratio of unknown classes to known classes. Thus, we set various openness values and measured the corresponding performance. Here, openness is a measure of how open the problem setting is and is defined as follows [4]:

$$\text{openness} = 1 - \sqrt{\frac{2C_T}{C_E + C_R}}$$

where $C_T$ is the number of classes used in training, $C_E$ is the number of classes used in evaluation, and $C_R$ is the number of
In this work, the hyperparameter $d_{\text{min}}$ (the minimum distilled probability) and the temperature $\tau$ can both significantly affect the classification performance of the student network. Thus, in this section, we analyze the effects of these two hyperparameters. For these sensitivity analyses, we adopted the same open set scenario based on CIFAR-10 and CIFAR-100 that was used for the quantitative ablation study in Section IV-C, and no uncertainty threshold was applied.

Fig. 12 shows the F1 scores with different values of $d_{\text{min}}$. The best result was obtained when a $d_{\text{min}}$ of 0.7 was used in T/E/S learning. The results show that a deviation of $d_{\text{min}}$ from 0.7 caused performance deterioration, but it was not significant in most cases. However, when $d_{\text{min}}$ was set to 0.5, there was a very large decrease in performance. This indicates that the teacher network must provide a probability of higher than 50% for the target class when distilling the posterior probability. As seen in Fig. 13, a $\tau$ of 3 resulted in the best performance, while the worst performance was observed when no temperature scaling was performed ($\tau = 1$). At other values of $\tau$, the F1 scores were slightly worse than that at $\tau = 3$ under the lowest openness. However, the difference became insignificant for $2 \leq \tau \leq 4$ as the openness increased.

E. Comparison With State-of-the-Art Methods

In this section, the proposed methods (T/E/S-CD and T/E/S-CDU) are compared with other state-of-the-art methods. We considered two different experimental setups. In the first, the performance of unknown detection, which considers only the classification between “known” and “unknown,” was measured in terms of the area under the receiver operating characteristic curve (AUROC). In the second, the OSR performance, which reflects closed-set classification following unknown detection, was measured in terms of the macroaverage F1 score.

For the unknown detection performance comparison, we followed the protocol defined in [17] with four image datasets: MNIST, SVHN, CIFAR-10, and Tiny-ImageNet. The MNIST, SVHN, and CIFAR-10 datasets were randomly partitioned into six known classes and four unknown classes. In addition, the model was trained on four nonanimal classes from CIFAR-10, and ten animal classes were randomly selected from the CIFAR-100 dataset and added as unknown samples during the testing phase. This task is referred to as CIFAR + 10. Similarly, 50 unknown classes were randomly selected from CIFAR-100; we refer to this task as CIFAR + 50. Finally, 20 classes were randomly chosen from the Tiny-ImageNet dataset as known classes, and the
remaining 180 classes were designated as unknown. For all datasets used for unknown detection, a random class split was repeated five times, and the average AUROC was used for evaluation. The comparison results are shown in Table II. Since the AUROC is a calibration-free measure, T/E/S-CD and T/E/S-CDU performed equally well. Therefore, we report the performance for both methods under the label T/E/S-CD(U).

These experimental results show that the proposed method achieved the best performance in two of the data settings. However, in the remaining data settings, CGDL showed slightly better performance than our method. The main difference between CGDL and our method lies in the model used during the testing phase. We utilize a single classifier, the student network, for testing. However, in CGDL, a class-conditional variational autoencoder is primarily used for unknown detection, with a greater emphasis on unsupervised learning through a reconstruction loss. We believe this distinction is the primary reason why our method achieved the second-best performance in unknown detection. However, despite this slight decrease in unknown detection performance, the proposed method significantly outperformed CGDL in terms of OSR performance, which encompasses both unknown detection and closed-set classification, as shown by further experiments described below.

Fig. 11. (a) $F_1$ scores for nine baselines. $T$ was excluded because its performance was too low. Specifically, it achieved $F_1$ scores of less than 0.75 when the openness was 21.6% and less than 0.40 for an openness of 72.8%. (b) and (c) Each component’s contribution when classes from CIFAR-10 and CIFAR-100 were used as the known classes and unknown classes, respectively. Specifically, (b) and (c) show the performance improvements attained by introducing additional networks into the OVRN-CD baseline and by introducing an uncertainty threshold, respectively.

**TABLE II**

RESULTS OF UNKNOWN DETECTION

| Method          | MNIST | SVHN  | CIFAR-10 | CIFAR+10 | CIFAR+50 | Tiny-ImageNet | Avg.  |
|-----------------|-------|-------|----------|----------|----------|---------------|-------|
| Softmax         | 0.978 | 0.886 | 0.677    | 0.816    | 0.805    | 0.577         | 0.790 |
| OpenMax [12]    | 0.981 | 0.894 | 0.695    | 0.817    | 0.796    | 0.576         | 0.793 |
| G-OpenMax [16]  | 0.984 | 0.896 | 0.675    | 0.827    | 0.819    | 0.580         | 0.797 |
| OSRCI [17]      | 0.988 | 0.910 | 0.699    | 0.838    | 0.827    | 0.586         | 0.808 |
| DOC [13]        | 0.986 | 0.937 | 0.885    | 0.880    | 0.882    | 0.725         | 0.883 |
| OVRN-CD [14]    | 0.989 | 0.941 | 0.903    | 0.907    | 0.902    | 0.730         | 0.895 |
| CPN [27]        | 0.990 | 0.926 | 0.828    | 0.881    | 0.879    | 0.639         | 0.857 |
| MLOSSR [19]     | 0.989 | 0.921 | 0.845    | 0.895    | 0.877    | 0.718         | 0.874 |
| CAIE [20]       | 0.989 | 0.922 | 0.895    | 0.943    | 0.937    | 0.774         | 0.908 |
| CGDL [21]       | 0.994 | 0.935 | 0.903    | 0.959    | 0.950    | 0.762         | 0.917 |
| T/E/S-CD(U)     | 0.992 | 0.953 | 0.905    | 0.951    | 0.924    | 0.733         | 0.910 |

The result for the best-performing method is highlighted in bold, and the result for the second-best method is underlined.

**TABLE III**

COMPARISON OF OSR PERFORMANCE ON CIFAR-10 WITH VARIOUS “UNKNOWN” DATASETS

| Method           | ImageNet-crop | ImageNet-resize | LSUN-crop | LSUN-resize | Avg.  |
|------------------|---------------|-----------------|-----------|-------------|-------|
| Softmax          | 0.639         | 0.653           | 0.642     | 0.647       | 0.645 |
| OpenMax [12]     | 0.660         | 0.684           | 0.657     | 0.668       | 0.667 |
| LadderNet+Softmax [15] | 0.640       | 0.646           | 0.644     | 0.647       | 0.644 |
| LadderNet+OpenMax [15] | 0.653       | 0.670           | 0.652     | 0.659       | 0.659 |
| DHPRNet+Softmax [15] | 0.645       | 0.649           | 0.650     | 0.649       | 0.648 |
| DHPRNet+OpenMax [15] | 0.655       | 0.675           | 0.656     | 0.664       | 0.663 |
| CROSAR [15]      | 0.721         | 0.735           | 0.720     | 0.749       | 0.731 |
| DOC [13]         | 0.760         | 0.753           | 0.748     | 0.764       | 0.756 |
| TCC [18]         | 0.798         | 0.779           | 0.758     | 0.806       | 0.785 |
| OVRN-CD [14]     | 0.835         | 0.823           | 0.846     | 0.839       | 0.836 |
| GPORSR [29]      | 0.821         | 0.777           | 0.843     | 0.784       | 0.806 |
| MLOSSR [19]      | 0.837         | 0.826           | 0.783     | 0.801       | 0.812 |
| CGDL [21]        | 0.840         | 0.832           | 0.806     | 0.812       | 0.823 |
| T/E/S-CD (Ours)  | **0.852**     | **0.816**       | **0.851** | **0.837**   | **0.839** |
| T/E/S-CDU (Ours) | 0.843         | 0.808           | 0.845     | 0.828       | 0.831 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Finally, the proposed T/E/S learning paradigm was validated through comparison with state-of-the-art methods in terms of OSR performance. OSR models were trained on all training samples of either MNIST or CIFAR-10. However, in testing, we additionally used another unknown dataset in combination with the test samples from MNIST or CIFAR-10. Specifically, samples from Omniglot, MNIST-Noise, and Noise were considered unknowns when MNIST was used as the training dataset. Here, noise is a set of synthesized images in which each pixel value was independently sampled from a uniform distribution on [0, 1]. When the CIFAR-10 dataset was used for training, samples from ImageNet and LSUN were used as unknown samples. Images from the ImageNet and LSUN datasets were resized or cropped to make the unknown samples the same size as the known samples, following the protocol suggested in [15]. The known-to-unknown ratio was set to 1:1 for all cases.

The results are compared in Tables III and IV. T/E/S-CD performed the best on average, scoring the highest for the two unknown datasets when CIFAR-10 was used for training. When MNIST was the training dataset, the proposed T/E/S-CD method also achieved the best results on all unknown datasets. Contrary to the results of the unknown detection experiments, the proposed method outperformed CGDL (the best performer in unknown detection) as well as other state-of-the-art OSR methods.

Despite its superiority in OSR performance, the proposed method must also be able to minimize the loss of closed-set classification to ensure generalizability. Therefore, we compared our method with existing OSR methods in terms of closed-set classification accuracy. Specifically, classification accuracy was measured over the selected known classes of MNIST, SVHN, and CIFAR-10 using the same five random class splits as in the unknown detection experiment. Considering the overwhelmingly higher OSR performance shown in Tables III–V indicates that the degradation in closed-set classification performance due to T/E/S learning is not significant compared to that of state-of-the-art OSR methods. Moreover, T/E/S-CD achieved the best performance on CIFAR-10.

Finally, as shown in Table VI, we compared the proposed method with GAN-based OSR methods (G-OpenMax [16] and TCCC [18]) in terms of training cost to evaluate the time cost increase incurred by our method, which requires both a teacher network and an explorer network (a GAN) in addition to the student network. In contrast, other GAN-based methods need only a classifier and a GAN. The training time was measured on the MNIST and CIFAR-10 datasets, considering all classes as known during the training phase.
First, we found that training a GAN accounted for the majority of the training time, as the vanilla CNN (denoted by Softmax) required considerably less training time than the other methods. Moreover, our method required the longest time, although the time gap between our method and other GAN-based methods was only a few minutes. Nevertheless, we were able to achieve significant improvements in terms of OSR performance. However, the issue of high-time complexity should be addressed in future research to facilitate broader applications.

### F. Experiments on ImageNet

To validate whether the proposed method can scale and generalize to large-scale datasets, we conducted comparative experiments on the ImageNet dataset following the protocol described in [28]. Specifically, we used the ImageNet-1K dataset [38] as the known class dataset. The ImageNet-1K dataset consists of 1000 classes, with over 1.2 million training images and 50,000 images for testing. For testing, we introduced ImageNet-O [39], which includes 2000 samples distinct from those in ImageNet-1K, as the set of unknown samples. It would be quite challenging to train all networks of the proposed method on ImageNet from scratch using a personal computer. Instead, we opted to initialize both the student and teacher networks with pretrained ResNet-50. After that, the fine-tuning process required 424815.6 s (equivalent to 4.92 days) to complete 10 epochs, with a batch size of 32.

We compared the proposed method with several state-of-the-art OSR methods, including a vanilla CNN with softmax output activation (also referred to as “Softmax” in Table VII). This latter method can be regarded as the pretrained model used for the initialization of the proposed method. Our comparison was based on the closed-set accuracy, AUROC, and open-set classification rate (OSCR) [40]. Here, the OSCR is a measure that combines the accuracy for known classes and the unknown detection performance. Specifically, the OSCR is calculated as the area under the curve of the correct classification rate (CCR) versus the false positive rate (FPR). The CCR represents the fraction of samples for which the target known class has the highest probability and that probability exceeds a threshold δ, while the FPR represents the fraction of unknown samples classified as any known class with a probability greater than δ.

| Method      | Accuracy | AUROC | OSCR  |
|-------------|----------|-------|-------|
| Softmax     | 0.696    | 0.482 | 0.424 |
| GCPL [41]   | 0.630    | 0.560 | 0.430 |
| RPL [42]    | 0.698    | 0.594 | 0.486 |
| ARPL [28]   | 0.702    | 0.600 | 0.489 |
| T/E/S-CD    | **0.726**| **0.603**| **0.506**|

The proposed method achieved the best performance not only in terms of the AUROC for the unknown detection task but also in terms of the closed-set classification accuracy. Consequently, the proposed method improved the OSCR, which considers both tasks, by approximately 17%. These experimental results demonstrate the excellent scalability of the proposed method to challenging large-scale image datasets while also showing its strong generalization capabilities.

### V. Conclusion

In this article, we developed a T/E/S learning method for OSR based on our intuition that the overgeneralization problem of deep-learning classifiers can be significantly mitigated by exploring various possibilities for unknowns. We first extended traditional T/S learning by introducing HE-KD to not only soften the posterior probabilities of the teacher network for known classes but also extract uncertainty. The softened probabilities prevent an unknown sample from scoring high, and the uncertainty can be used to glean hints that can guide an explorer network to generate unknown-like examples. In addition, for the generation of unknown-like open set examples, we introduced a new objective and training procedure for a GAN. The developed explorer network explores a wide range of unknown possibilities. Experimental results showed that each component proposed in this article contributes to improving OSR performance. As a result, the proposed T/E/S learning method can outperform current state-of-the-art methods in OSR. As evidenced in [26], by employing different network structures and image recognition techniques, both our proposed method and other existing methods may be further improved. Therefore, it is imperative to exploit general image recognition techniques when developing OSR models. However, under identical conditions and training settings, our proposed method demonstrated superior performance.

Discriminating between known and unknown samples is considered a key capability of intelligent self-learning systems [31]. However, if a given system cannot subsequently learn from an identified unknown sample, then that system cannot be called self-learning. Thus, the proposed T/E/S learning paradigm should be extended to incorporate class-incremental learning, in which incoming unknown samples are used to continually train the learning model on new unknown classes. This will be considered one of our future research directions. In addition, we have proposed an OSR recognition rule that employs logits without providing a solid theoretical foundation for this proposal. We acknowledge that we could potentially further enhance OSR performance by incorporating a more formalized uncertainty estimation function, such as the energy function [43]. This is another issue we plan to address in our future work.

### References

[1] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–12.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[3] C. Geng, S.-J. Huang, and S. Chen, “Recent advances in open set recognition: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3614–3631, Oct. 2021.

[4] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, “Toward open set recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, Jul. 2013.
C. Pires, M. Barandas, L. Fernandes, D. Folgado, and H. Gamboa, “Towards knowledge uncertainty estimation for open set recognition,” Mach. Learn. Knowl. Extraction, vol. 2, no. 4, pp. 505–532, Oct. 2020.

J. Henrydoss, S. Cruz, E. M. Rudd, M. Gunther, and T. E. Boul, “Incremental open set intrusion recognition using extreme value machine,” in Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2017, pp. 1089–1093.

S. Cruz, C. Coleman, E. M. Rudd, and T. E. Boul, “Open set intrusion recognition for fine-grained attack categorization,” in Proc. IEEE Int. Symp. Technol. Homeland Secur. (HST), Apr. 2017, pp. 1–6.

J. Jang, M. Seo, and C. O. Kim, “Support weighted ensemble model for open set recognition of wafer map defects,” IEEE Trans. Semicond. Manuf., vol. 33, no. 4, pp. 635–643, Nov. 2020.

J. Jang and G. T. Lee, “Decision fusion approach for detecting unknown wafer bin map defects based on a deep multitask learning model,” Expert Syst. Appl., vol. 215, Apr. 2022, Art. no. 119363.

A. Rocha et al., “Authorship attribution for social media forensics,” IEEE Trans. Inf. Forensics Secur., vol. 12, no. 1, pp. 5–33, Jan. 2017.

G. Spigler, “Denoising autoencoders for overgeneralization in neural networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 4, pp. 998–1004, Apr. 2020.

A. Bendale and T. E. Boul, “Towards open set deep networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1563–1572.

L. Shu, H. Xu, and B. Liu, “DOC: Deep open classification of text documents,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2017, pp. 2911–2916.

J. Jang and C. O. Kim, “Collective decision of one-vs-rest networks for open-set recognition,” IEEE Trans. Neural Netw. Learn. Syst., early access, Jul. 14, 2022, doi: 10.1109/TNNLS.2022.3189996.

R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, “Generative OpenMax for multi-class open set classification,” arXiv:1903.03161.

A. Bendale and T. E. Boul, “Towards open world recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1893–1902.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” Science, vol. 350, no. 6266, pp. 1332–1338, Dec. 2015.

L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, pp. 2579–2605, Nov. 2008.

O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, Dec. 2015.

D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, “Natural adversarial examples,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 15257–15266.

A. R. Dhamija, M. Günther, and T. E. Boul, “Reducing network agnostophobia,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 9157–9168.

H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, “Robust classification with convolutional prototype learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 3474–3482.

G. Chen et al., “Learning open set network with discriminative reciprocal points,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 507–522.

W. Liu, X. Wang, J. Owens, and Y. Li, “Energy-based out-of-distribution detection,” in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 21464–21475.

JaeYeon Jang received the Ph.D. degree in industrial engineering from Yonsei University, Seoul, South Korea, in 2021.

From 2021 to 2022, he was a Post-Doctoral Scholar with the Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA. In 2022, he joined the Department of Data Science, Catholic University of Korea, Bucheon, South Korea, where he is currently an Assistant Professor. His current research interests include industrial data analytics, deep learning, machine learning, and reinforcement learning.

Chang Ouk Kim received the Ph.D. degree in industrial engineering from Purdue University, West Lafayette, IN, USA, in 1996.

He is currently a Professor with the Department of Industrial Engineering, Yonsei University, Seoul, South Korea. He has published more than 100 articles in journals and conference proceedings. His current research interests include pattern recognition, machine learning, and data science for manufacturing and defense analysis.