LORD: Leveraging Open-Set Recognition with Unknown Data

Tobias Koch*, †, Christian Riess†, and Thomas Köhler*
*e.solutions GmbH, Erlangen, Germany
†Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
{tobias.koch, thomas.koehler}@esolutions.de, christian.riess@fau.de

Abstract

Handling entirely unknown data is a challenge for any deployed classifier. Classification models are typically trained on a static pre-defined dataset and are kept in the dark for the open unassigned feature space. As a result, they struggle to deal with out-of-distribution data during inference. Addressing this task on the class-level is termed open-set recognition (OSR). However, most OSR methods are inherently limited, as they train closed-set classifiers and only adapt the downstream predictions to OSR.

This work presents LORD, a framework to Leverage Open-set Recognition by exploiting unknown data. LORD explicitly models open space during classifier training and provides a systematic evaluation for such approaches. We identify three model-agnostic training strategies that exploit background data and applied them to well-established classifiers. Due to LORD’s extensive evaluation protocol, we consistently demonstrate improved recognition of unknown data. The benchmarks facilitate in-depth analysis across various requirement levels. To mitigate dependency on extensive and costly background datasets, we explore mixup as an off-the-shelf data generation technique. Our experiments highlight mixup’s effectiveness as a substitute for background datasets. Lightweight constraints on mixup synthesis further improve OSR performance.

1. Introduction

Most classification algorithms are designed for closed-set environments, where all classes are known prior to deployment of the classifier. However, real-world applications may expose classifiers to unseen classes. Recognizing that an input belongs to an unseen class constitutes the open-set recognition (OSR) task. Scheirer et al. [50] distinguish two subtasks: 1) Recognizing samples as known or unknown. 2) Classify knowns to its specific class. An example is face recognition of few known subjects among many others [20].

We distinguish three types of classes [51] as depicted in Fig. 1: 1) Known classes (KCs) are uniquely labeled examples detected and classified. They are essential in both the training and test set. 2) Known unknown classes (KUCs) is background data and comprises samples not necessarily grouped into meaningful categories, but assumed to be unrelated to the KCs. They are part of the training and test set. We differentiate genuine KUCs, a dataset subset, from synthesized KUCs. 3) Unknown unknown classes (UUCs) are unseen during training. They are only part of the test set.

In recent years, OSR has gained increasing attention [17, 49, 61]. Most approaches adopt closed-set training with open-set inference, as exemplified in various methods, including distance-based classifiers [4, 29, 38, 47], Support Vector Machines (SVMs) [3, 19, 27, 51], Extreme Value Machines (EVMs) [23, 30, 48, 58], losses and calibrations for Deep Neural Networks (DNNs) [5, 39, 54], uncertainty quantification [6, 34, 35], novelty detection [7], and autoencoder architectures [41, 42, 63]. For inference, a threshold is introduced to either reject or classify a test sample based on a confidence value [56].
Let $T_K = \{(x_i, y_i)\}_{i=1}^T$ be a labeled training dataset with features $x \in \mathbb{R}^D$ and class labels $y \in \mathcal{C}_K$, where $\mathcal{C}_K$ denotes the set of KCs. Given $T_K$, the goal is to infer a likelihood function $f(x; T_K) = z$ with $z \in \mathbb{R}^{|\mathcal{C}_K|}$ that maps features to a confidence $z_y \in \mathbb{R}$ for each class in $\mathcal{C}_K$. For instance, $z$ could describe normalized posterior probabilities or distances. Function $f$ is used to decide for class $\hat{y}$ with the highest confidence: $\hat{y} = \arg\max_{y \in \mathcal{C}_K} f(x; T_K)^y$. A reject option converts this prediction into OSR according to:

$$g(x, \delta; T_K) = \begin{cases} \hat{y} & \text{if } \max_{y \in \mathcal{C}_K} f(x; T_K)^y > \delta, \\ u & \text{otherwise}, \end{cases}$$

(1)

where $\delta$ denotes the decision threshold. However, training exclusively on a closed set can benefit the classification task but inherently limits the OSR capabilities.

To the best of our knowledge, only a few works consider background samples [13, 20, 43]. While genuine KUCs undoubtedly simplify OSR, they may not be available. Recent approaches show that synthesizing out-of-distribution (OOD) data with generative adversarial networks (GANs) can suffice for training DNNs [16, 40, 52]. However, GAN training can be unstable [31] and is designed for in-distribution data [18], necessitating additional methods to handle outliers [16]. The mentioned works offer valuable insights into end-to-end OSR with KUCs. However, most are trained in a $K+1$ fashion, where the background is modeled in a single class. This approach is limited by the background class’s complexity [13]. An exception is the work by Günther et al. [20] that studies OSR via linear discriminant analysis and the EVM. We extend their work by exploiting KUCs in diverse ways and examine whether synthetic data is a suitable proxy for genuine KUCs.

Our framework LORD provides guidance for researchers to explicitly model open space during classifier training. To verify the efficacy of this modeled space on unknown recognition, a benchmark protocol is introduced. The protocol requires, besides KCs, access to genuine KUCs and UUCs. LORD distinguishes between two assessments: 1) biased assessment, involving training with genuine KUCs and testing with genuine KUCs and UUCs. 2) Unbiased assessment excludes KUCs during testing. This identifies optimal learning conditions for different models and application-specific requirements. In the context of face recognition, a biased application could be a system at an airport likely to encounter genuine KUCs, while an unbiased application might be the authentication on a smartphone.

LORD encompasses a comprehensive range of OSR metrics that facilitate in-depth analysis across various requirement levels. These levels may place a particular emphasis on secure open-set performance or prioritize strong closed-set performance, catering a more user-friendly behavior. In particular, this extensive evaluation highlights a fundamental trade-off between open- and closed-set performance across models.

To model open space, LORD deploys three model-agnostic training strategies on six classifiers spanning four distinct categories. These strategies improve biased OSR performance by up to 30%, while unbiased performance generally demonstrates a more modest improvement.

Admitting the scarcity and cost of domain-specific background data in real-world applications, LORD offers a solution in the form of mixup as an off-the-shelf data generation technique. Mixup samples are convex combinations of distinct KCs and act as substitutes for genuine KUCs. While naïve mixups constitute the occupation problem, LORD proposes effective constraints to refine the generation process. Our experiments confirm mixup as an excellent substitute for genuine KUCs. This approach proves beneficial in assessing models before investing in resource-intensive collection of background data.

To summarize, LORD provides a systematic framework to assess the efficacy of background data exploitation to enhance classifiers within particular application contexts. When background data is not available, mixup poses an effective solution. If this proves beneficial, researchers might find it valuable to gather additional background data.

This work is organized as follows: Section 2 reviews related works. Section 3 introduces our OSR benchmarking protocol. Section 4 outlines the training strategies and their use for 6 OSR models. Section 5 explores mixup and the constrained generation. Section 6 summarizes the main findings and concludes this work.

2. Related work

We explore incorporating KUCs during training and reformulate Eq. (1) accordingly. Let $T = \{(x_i, y_i)\}_{i=1}^T$ be a partially labeled training dataset with class labels $y \in \mathcal{C}$, where $\mathcal{C} = \mathcal{C}_K \cup u$ and $u$ indicates a missing label. Set $T$ contains samples of KCs and KUCs, leading to a change in function $g$ where $T_K$ is replaced by $T$, $h(x, \delta; T) = g(x, \delta; T)$. The class decision remains unchanged, but the inclusion of KUCs impacts the likelihood function $f$, categorizing different approaches in OSR.

2.1. Semi-supervised model training

KUCs samples can also be considered as a special instance of unlabeled data. Exploiting both labeled and unlabeled data is a common focus in semi-supervised learning. Notable techniques in this field comprise manifold regularization [37] and the generation of pseudo-labels [2, 24], integrating unlabeled data into a supervised learning regime. Using unlabeled samples is considerably cheaper than labeled samples [59]. Jointly exploiting labeled and unlabeled data can substantially enhance the underlying models compared to purely supervised learning [65].

4387
However, unlike OSR, semi-supervised learning tackles a complementary scenario: predictions cover only KCs. The approach assumes that unlabeled instances belong to one or more of the KCs, without considering the open space. We follow concepts similar to semi-supervised learning and incorporate unlabeled KUCs to improve OSR learning.

2.2. Open-set training with open-set inference

The vast body of previous works use unlabeled KUCs in a $K+1$ fashion, i.e., $K$ KCs and one unknown class [16, 40, 52]. In function $h$, $C_K$ is substituted by $C$, treating all samples with the label $u$ as one large class. This allows the model to predict the background class without a threshold $\delta$. However, we show that modeling unknowns in a single class can be suboptimal and is surpassed by alternative strategies.

Others use external datasets to define special loss functions to calibrate filters [43] or probabilities [13], particularly tailored to train DNNs. Günther et al. [20] take an initial step towards employing genuine KUCs with EVMs. They do not cluster KUCs but consider them to learn probabilistic models of KCs. This can improve the detection of UUCs compared to simple baselines like distance-based detection. These findings indicate the potential of leveraging KUCs to enhance the predictive power of OSR models.

In this paper, we generalize model-specific approaches into three different model-agnostic training strategies that exploit KUCs. Specifically, we extend multiple OSR models using these training strategies and determine the optimal strategy based on the underlying model and task.

2.3. Generating out-of-distribution samples

Recent works addressed the question of whether KUCs could be synthesized when external data is unavailable. For instance, Du et al. [14] model KUCs by estimating Gaussian distributions per KC and sampling background data near class boundaries. However, the Gaussian distribution assumption is simplistic, resulting in uniform restrictions of the decision boundary. Zhou et al. [67] regularizes DNN training using manifold mixup [57] to learn placeholders.

In-distribution data is synthesized through generative models [11, 18, 45, 53] or augmentation [10, 33, 44, 57, 64, 66]. Conversely, OOD data generation is more challenging. Ge et al. [16] propose a complex pipeline: 1) Train a DNN and conditional GAN using KCs. 2) Generate GAN samples and classify them with the DNN. 3) Store misclassified samples as KUCs. 4) Train another DNN in a $K+1$ way with KUCs as background class. This generation is guided by the assumption that regions where both models disagree do not belong to any KC. Unfortunately, achieving convergence of one GAN is challenging, leading to the training of several GANs on different class subsets. Also, control over the generated random samples is very limited, and the required number of KUCs remains an open question. Neal et al. [40] train an encoder-decoder GAN and a classifier. To generate KUCs, they alter KC samples in the latent space of the GAN to make the DNN uncertain about the prediction. The DNN is finetuned in a $K+1$ manner. Also here, there is a mutual dependence between the GAN and DNN and the generation process lacks control. Generating images is challenging due to their high dimensionality and the need to match the data context. Moreover, DNNs are vulnerable to small deviations as introduced by adversarial attacks [1, 15].

Kong and Ramanan [31] avoid mutual dependencies using DNN penultimate layer features to train a GAN. The GAN’s discriminator detects unknowns, while the DNN classifies knowns. We adopt a similar two-stage approach, but unlike GANs operating in image space, we operate solely in the feature space. Specifically, we use manifold mixup [57] to generate KUCs, a lightweight alternative to complex GANs. However, manifold mixup may generate unwanted KUCs that overlap with a KC, referred to as occupation problem. To mitigate this, we introduce an effective filter to retain only meaningful KUCs.

3. How to benchmark with known unknowns?

Common OSR benchmarks [13, 40, 50] ignore KUCs for training and testing. In contrast, we extend the work of Günther et al. [20] to explore the impact of KUCs on model training. This serves as a reference under conditions when dataset-related background data is available.

3.1. Defining protocols with known unknowns

For our experiments, we define evaluation protocols using well-established datasets with varying characteristics.

CIFAR-100 [32]. This image dataset consists of 20 superclasses divided into 100 equally balanced subclasses. We split the superclasses into 12 KCs, 4 KUCs, and 4 UUCs. Adopting the subclass labels, this results in 60 KCs, 20 KUCs, and 20 UUCs. The training set contains 500 images per class, leading to a KUC to KC sample ratio of 0.33. For feature extraction, we use EfficientNet-B4 [55], pre-trained on ImageNet [12] and finetuned on the 60 KCs. The embedding dimension is 1792. Unless otherwise stated, we repeat all experiments 3 times with varying superclass splits and report averaged results. While a pre-trained ImageNet feature extractor already saw all CIFAR-100 classes, this setup remains open-set for 3 reasons: 1) Image resolutions significantly differ. 2) Finetuning is subject to catastrophic forgetting [36, 46]. 3) We use the features to train a downstream classifier, which is never exposed to ImageNet.

Labeled Faces in the Wild (LFW) [25, 26]. For this face recognition dataset, we follow the protocol of Günther et al. [20] and evaluate the $o_3$ probe set. The training set consists of 3 samples for each of the 610 KCs and 1 sample for
each of the 1070 KUCs which is a ratio of 0.58. We extract 128-dimensional features using ResNet-50 [22], pre-trained on MS-Celeb-1M [21], and finetuned on VGGFace2 [9].

CASIA-WebFace (C-WF) [62]. This is a face recognition dataset with imbalanced distributions. We split it into 6345 KCs, 2115 KUCs, and 2115 UUCs. Unless otherwise stated, all experiments are repeated 3 times with varying KC and KUC splits and averaged results are reported. On average, the least represented KC has 13 samples and the largest 627 samples. The KUC to KC sample ratio is 0.33. Feature extraction uses the same model as for LFW but with a feature dimension of 2048. We also employ a reduced version (Tiny C-WF) to address classifier scaling issues with the full dataset. Tiny C-WF retains 20 % of KCs, KUCs, and UUCs. The number of samples per class is reduced by 50 %.

3.2. Evaluating open-set recognition models

First, we assess the ability to distinguish knowns from unknowns in a binary problem. This is supported by a Receiver Operating Characteristic (ROC) depicting the true positive rate (TPR) vs. the false positive rate (FPR). To obtain a performance measure independent of the threshold δ, we use the Area Under the ROC Curve (AUC-ROC).

Second, we employ the Open-Set Classification Rate (OSCR) [13], which measures the correct classification rate (CCR) and FPR. Let \( g(x, \delta; T) \) be a decision function as formalized in Eq. (1). The test set comprises three types of classes, denoted as \( E = E_K \cup E_u \) with \( E_u = E_{KU} \cup E_{UU} \). Then these measures are expressed as follows:

\[
\begin{align*}
\text{CCR}(\delta) &= |E_K|^{-1} \{ x \in E_K \cap g(x, \delta; T) = y \}, \\
\text{FPR}(\delta) &= |E_u|^{-1} \{ x \in E_u \cap g(x, \delta; T) \neq u \}.
\end{align*}
\]

For most open-set applications, e.g. face recognition, a high CCR at low FPR is preferable [20]. Therefore, we report the CCRs in the open-set relevant FPR range of 0 to 10 %.

3.3. Biased vs. unbiased evaluation

The biased protocol in Section 3.1 with genuine KUCs in the training set simplifies the detection of KUCs in the test set, resulting in significant improvements in OSR measures.

To ensure unbiased evaluation, we compute the metrics solely on a test subset comprising KCs and UUCs, excluding KUCs. Thus, \( E_u \) in Eq. (3) is substituted by \( E_{UU} \). Biased evaluation considers metrics for both KUCs and UUCs, while unbiased evaluation focuses solely on UUCs. The biased assessment tends to paint an overly optimistic picture for applications that primarily deal with UUCs.

4. How to learn models with known unknowns?

This section outlines three open-set training strategies and their deployment to six classifiers of four categories.

4.1. Open-set training strategies

Incorporating KUCs into the training process of classifiers can be achieved using various strategies. Figure 2 illustrates examples for these approaches, including the EVM.

Single Pseudo Label (SPL). Samples without labels are assigned the pseudo label \( u \) and treated as a single class. This method aligns with the \( K+1 \) training strategy in previous works [16, 40, 52] with function \( h \), but with \( C_K \) substituted by \( C \). This also implies \( f(x; T) = z \) with \( z \in \mathbb{R}^{|C|} \).

The model can now predict label \( u \) directly independent of the threshold \( \delta \). This results in steep decision boundaries as demonstrated in Fig. 2b. While this approach is universally applicable, the complex distribution of the large background class presents a challenge. Depending on the underlying OSR model, finding suitable representations for the background class formed by the pseudo label is difficult.

Multi Pseudo Label (MPL). This strategy assigns an individual pseudo label to each KUC sample. The number of classes in \( C \) increases by \( |T_u| \), the amount of KUC samples in \( T \). The remaining strategy follows the SPL, but now each former KUC contains only one sample. If one of the newly declared classes is predicted, it must be mapped to label \( u \).

Similar to SPL, this approach enables direct predictions of unknown labels, resulting in sharp decision boundaries, cf. Fig. 2c. We hypothesize that multiple pseudo labels stabilize OSR learning, especially when each KUC sample belongs to a distinct category with high inter-category distances in feature space. A notable drawback is the significant increase in classes, making it impracticable for One vs. Rest (OvR) models. Additionally, MPL may induce label noise, leading to inconsistent decision boundaries.

Known vs. Rest (KvR). This mode is akin to the common OvR multiclass strategy, often used for binary classifiers like SVMs to handle multiclass problems. In KvR, each KUC never acts as a positive class, avoiding the need for its own model representation. Instead, it serves as negative in the rest-class of other binary models. This approach resolves the issue of complex background class representation. Other classes can still adjust their decision boundaries.
to the negative samples, observed in Fig. 2d where KCs’ boundaries are tightened by KUCs. Implementing KvR can vary, but we intend for the background data not to be explicitly modeled as a pseudo-class. It serves as a tool to determine better decision boundaries for KUCs concerning the open space. Experiments in Section 4.3 show that this approach enhances unknown recognition.

4.2. Deployed strategies for open-set models

In this section, we briefly outline how to deploy the strategies to 6 OSR models representing 4 distinct categories. More details are provided in the supplements.

The Open-Set Nearest Neighbor (OSNN) [29] exploits the ratio between the two nearest samples from distinct classes as confidence value. SPL includes all KUCs in distance ratio computation and class prediction. MPL treats each KUC as a separate class, impacting the reject option only. For KvR, only KCs are used for label prediction, while both KCs and KUCs determine distance ratios. The strategies primarily vary in the confidence computation, leading to nearly identical results.

Given the vast possibilities of training Deep Neural Networks (DNNs) with KUCs by tailoring losses, we opt for the classical cross-entropy loss. For SPL, we expand the number of output units by one class. MPL does not scale to large KUC sets. Entropic open-set loss [13] is adopted for KvR.

The Extreme Value Machine (EVM) [48], C-EVM [23], and both Support Vector Machine (SVM) variants, Weibull SVM (W-SVM) [51] and Probability of Inclusion SVM ($P_I$-SVM) [27], implement an OvR scheme. For each one-class the other rest-classes serve as counterpart. For these models SPL and MPL differ only in the number of pseudo-classes. The SVMs omit MPL due to its limited scalability to large number of classes and their requirement of at least three samples per class. The KvR strategy can be deployed by not representing the KUCs as a positive one-class. Instead, they are always considered part of the rest-class.

4.3. Performance of training strategies

We benchmark the training strategies using the protocols in Section 3.1. Hyperparameters are determined through a grid search based on 5-fold stratified cross-validation. For LFW, standard 3-fold cross-validation is applied. The grid search is conducted for the baseline models and the parameters are reused for SPL, MPL, and KvR. This approach may favour the baseline, where no KUCs are used.

Biased evaluation. In Fig. 3, we report average OSCR measures for the three datasets. Results for the C-EVM and $P_I$-SVM are reported in the supplement. KUC exploitation improves OSR performance compared to baselines without KUCs. The exact behavior is model-dependent.

For OSNN, cf. Figs. 3a – 3c, we observe consistent improvements by SPL, MPL, and KvR. The highest gain of about 21% is observed in Fig. 3b for FPRs below 0.1%. This result is crucial for safety-relevant applications that require low FPRs. For C-WF, we see a gain in the FPR range of 0.1 to 1%, and for CIFAR-100 from 0.1% upwards. The
exploitation of KUCs does not negatively impact the CCR.

For DNN, cf. Figs. 3d–3f, KvR consistently outperforms the baseline. SPL performs similarly to KvR for CIFAR-100 and C-WF but is outperformed by KvR and the baseline on LFW, except for FPRs below 0.1%. We explain this behaviour by the few samples per KCs and KUC in case of LFW, causing two problems: 1) Each KUC is represented by one sample, making it challenging to model a pseudo-class. 2) The pseudo-class is large compared to the KCs. After examining SPL, we found that the DNN mainly predicts unknowns, a common issue of DNNs with unbalanced classes [5]. This is not the case for C-WF and CIFAR-100, where all classes have sufficient samples.

For the EVM in Figs. 3g–3i, all training strategies outperform the baseline. For CIFAR-100, we observe a gain of 17.6% in the CCR@FPR=10%, for LFW a gain of 30.6% at an FPR of 0.05%, and for C-WF 20% at an FPR of 1%.

The W-SVM results for CIFAR-100 and Tiny C-WF are in Figs. 3j and 3k. As indicated in Section 4.2, we do not apply the MPL strategy and also do not evaluate LFW due to the model’s limitations. By exploiting KUCs, OSR performance in the selected open-set range is improved by up to 6% and 4% for CIFAR-100 and Tiny C-WF, respectively.

Unbiased evaluation. The question remains whether KUCs also improve the detection of UUCs. In Fig. 4, we compare the biased and unbiased evaluation using the CCR@FPR=1% gain in relation to the baseline. The performances expressed by the unbiased evaluation fall below the biased measures. This indicates that UUCs are more difficult to detect and the biased evaluations draws a too optimistic picture. We found that this discrepancy is model-dependent. The OSNN exhibits a loss for CIFAR-100 in the unbiased measure and only marginal gains for LFW and C-WF. The DNN with KvR outperforms the baseline in both evaluations. This supports the finding that the entropic open-set loss is more suitable for OSR than the simple $K+1$ (SPL) strategy [13]. The EVMs improves over the baselines in both biased and unbiased cases, except in for LFW. Unlike the OSNN, the EVMs take advantage of a tail of KC and KUC samples to refine the decision boundary of KCs.

In contrast to the W-SVM, the $P_I$-SVM does not always benefit from KUCs in the unbiased case.

5. How to generate known unknowns?

We showed that genuine KUCs can improve OSR. However, in certain domains, genuine KUC data may be unavailable. This raises the question of how to synthesize them. In contrast to GANs that are not easily usable as standalone OOD generator for arbitrary OSR models, cf. Section 2.2, our proposed framework answers this question using the tools at hand, namely the data and the feature extractor.

5.1. Mixup – known unknowns straight off the shelf

Mixup [66] is a lightweight augmentation technique for DNNs that generates new data by convex combinations of samples. Jian et al. [28] show that standard mixup increases open-space risk by smoothing decision boundaries between classes. However, we further constrain mixup to use samples from distinct classes and assign to the mixed sample $\tilde{x}$ the unknown label $u$, leading to steep decision boundaries. Here, $(x_i, y_i)$ and $(x_j, y_j)$ are randomly drawn from $T_K$:

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j , \text{ with } y_i \neq y_j \text{ and } \tilde{y} = u . \quad (4)$$

We use manifold mixup [57], as mixing in feature space offers several advantages: 1) It is more efficient, as the combinations no longer need to be computed in the high-dimensional image space. 2) It is independent of peculiarities of the feature extractor, eliminating the need to pass the mixed image through that extractor.

We use CIFAR-100 and replace genuine KUCs with mixups of KCs. Mixing factor $\lambda$ in Eq. (4) is set to $\lambda \sim \text{Beta}(2, 2)$ with $\lambda \in [0.4, 0.6]$, ensuring that most mixups lie between two classes. While training with synthetic and testing with genuine KUCs is not biased, we present the metrics calculated on the test set with excluded KUCs for a fair comparison with models using genuine KUCs, and we continue to refer to this as unbiased. We also assess OSR performance by the number of generated mixup samples, represented by the mixup-to-known ratio, indicating how many mixups are generated per one known sample.
5.2. Augmenting models by manifold mixup

We exemplary investigate manifold mixup for the EVM in Tab. 1. With more mixups, the gaps between classes are filled and the open space is extended. The strategies have varying degrees of invasiveness, with SPL favoring open space the most, followed by MPL and then KvR. KvR requires finding a suitable threshold to form open space.

The AUC-ROC results for naively generated off-the-shelf mixups are shown in Fig. 5. Other metrics like the OSCR are anticipated here and can be found in the supplement for a more detailed analysis, this includes also the results for the C-EVM and P_{f}-SVM.

Similar to the findings in the first unbiased experiment in Fig. 4, the DNN (SPL) in this experiment also does not benefit from the additional background data, cf. Fig. 5a. The AUC-ROC and CCR\(\text{@FPR=10\%}\) of KvR show a steady increase. Interestingly, this does not apply to the entire FPR range, as the CCR\(\text{@FPR=1\%}\) temporarily deteriorates. This suggests that mixups may offer advantages at medium FPRs while providing no benefit at low FPRs, or vice versa. This makes it difficult to make a universal statement, such as mixup always improves OSR performance.

The EVM in Fig. 5b shows remarkable benefits from all strategies and mixup. The AUC-ROC corresponds perfectly with Tab. 1. SPL constricts the space for the KCs more than MPL, resulting in higher recognition rates of unknowns at the expense of knowns. MPL and KvR loosen this constriction. Not only the baseline, but also EVM (SPL) trained with genuine KUCs is outperformed. This is an intriguing observation, showing that mixup improves UUCs recognition to the same extent or even more than genuine dataset-related data, which is often expensive to obtain in real-world scenarios [59]. Although SPL is slightly better in the first experiment, cf. Fig. 4, we recommend KvR as preferred strategy due to its consistently stable results across all FPRs.

The result of the W-SVM is in Fig. 5c. Training with KvR yields a comparable AUC-ROC gain to genuine KUCs. With a mixup-to-known ratio of 0.1, it exceeds the baseline by about 1.2%.

Improving OSR with mixup is not universally effective. The OSNN and P_{f} -SVM, cf. supplement, decline in performance when mixup is applied. A nearest neighbor based approach is particularly prone to unfavorable sample placement, which is exacerbated by naïve mixups. We address this occupation problem in the next section.

5.3. Solving the occupation problem

Zhou et al. [67] mention the issue of already-occupied space between two KCs by a third. Since mixups are labeled as unknown, this can degrade the adjacent KCs. As augmentation technique or in feature learning, the occupation problem is not approached due to two reasons: 1) Mixups are generated in mini-batches and checking on their location in the feature space in relation to all other samples is infeasible. 2) The occupation might be favorable as mixups push away the in-between class, increasing the space between all affected classes. However, in case of pre-trained features, this can indeed impact OSR performance. We hypothesize that it is the main reason for the OSNN failure case.

We propose an effective constraint for an on-the-fly mixup generation. Let \(\bar{x}_{y}\) be a KC’s centroid, calculated as mean of all samples within a class. Using Euclidean distance \(d(\cdot, \cdot)\), we determine the mean distance \(d\) among all class centroids. A mixup sample is kept if its distance to all class centroids exceeds the scaled \(d\). Scale \(\alpha = 0\) represents unconstrained generation and \(\alpha > 0\) promotes filtering:

\[
d(\bar{x}, \bar{x}_{y}) > \alpha \cdot d \quad \forall y \in C_{K}.
\]

We conduct the CIFAR-100 experiment again, varying \(\alpha \in \{0, 0.6, 0.8, 1\}\). Results are presented in Fig. 6 and we recommend to view the supplement for additional details.
The OSNN in Fig. 6a benefits from stronger constraints ($\alpha \geq 0.8$), resulting in consistent AUC-ROC improvements of up to 3.4% compared to the baseline as the number of mixups increases. The CCR@FPR=1% is erratic, but with FPR=10%, it improves by almost 2%. We note three observations: 1) OSNN is prone to naïve mixups. 2) Highly constrained mixups improve OSR, but not across the entire FPR range. 3) Performance deterioration is unlikely and can be mitigated by monitoring a held-out set.

The DNN (KvR) in Fig. 6b has little benefit with stronger restrictions, for $\alpha = 1$ it even deteriorates after a mixup-to-known ratio of 1. Mixups lead to an improvement of up to 4%, especially for FPRs >1%. We conclude: 1) Naïve mixups enhance OSR for medium and larger FPRs. 2) DNNs usually benefit from diverse data, constraining mixups might be counterproductive.

The EVM (KvR) in Fig. 6c shows that no or lower constraints ($\alpha < 0.8$) increase more with the number of mixup samples in the CCR than the more restricted ones. The magnitude of the constraints affects different areas of the FPR. Our conclusions are: 1) Due to the variable behavior of different constraints, the EVM can be optimized for a desired FPR. 2) Finding the optimal configuration is complex and may require a hyperparameter search.

While the AUC-ROC of the W-SVM in Fig. 6d appears promising, the OSCR is irregular with a downward trend. This behaviour is underlined by the toy example in the supplement, where different mixup-to-known ratios lead to erratic changes at low confidences.

6. Discussion and Conclusion

This paper presents LORD to explicitly model open space by exploiting KUCs for open-set learning and to boost OSRs models. Our key findings are as follows:

**Known unknowns improve OSR learning.** Exploiting KUCs to train OSR models improves over baselines ignoring such samples. Considering KUCs improves the detection of KUCs and UUCs. The performance for UUCs is usually lower. Practitioners should carefully select the evaluation strategy and only benchmark with KUCs if this reflects the conditions in the addressed use case.

**The training strategy matters.** SPL can be considered as a baseline due to its simplicity. When used with synthetic KUCs, SPL is outperformed by competing methods. MPL shows comparable results but is not tractable for every model, treating each KUC sample as a separate class. KvR outperforms the other strategies in most benchmarks and requires fewer parameters to learn.

**The OSR evaluation measure matters.** AUC-ROC measures the ability to distinguish knowns and unknowns, but relying solely on this metric can create false expectations. The relevant FPR range for OSR is usually <10%, occupying a small part of the AUC and being outweighed by the range >10%. AUC-ROC improvements might be due to enhancements at high FPRs, which have limited relevance for OSR. It is crucial to report TPR or CCR at low FPRs. For future work, the OpenAUC [60] could also be a helpful alternative to measure OSR performance.

**Mixups can surrogate genuine known unknowns.** Mixups serve as a lightweight and effective surrogate for KUCs. DNN, EVM, and OSNN benefit the most from mixups. OSNN requires extra filtering to mitigate the occupation problem. Filtering does not consistently improve all models but increases the CCR in certain FPR ranges. The aim of solving the occupation problem is to maintain a high KC classification. A suitable compromise between unknown detection and known recognition has to be found.

**LORD is extendable to open-world learning.** Future work refines the generation of known unknowns. The constraints to address the occupation problem shall be enhanced. The highly efficient mixup synthesis proposed here is suitable for open-world learning and future experiments should involve open-world capable classifiers.
References


[26] Gary B. Huang, Marwan Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environ-


