A GENERAL FRAMEWORK FOR UNSUPERVISED ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper we present *GenAD*, a simple and generic framework for detecting examples that lie out-of-distribution for a given training set. The approach is based on first learning a semantic neighbourhood for each example in the training set and then training a binary discriminator to classify whether two examples come from the same or different semantic neighbourhoods. The framework is applicable to a wide range of anomaly detection problems, including visual, audio, and textual data. We are able to outperform previous approaches for anomaly, novelty, or out-of-distribution detection in the visual domain by a large margin. In particular we obtain AUROC > 0.999 for the challenging task of detecting examples from SVHN as out-of-distribution given CIFAR-10 as in-distribution, without making use of label information.

1 INTRODUCTION

Anomaly detection or novelty detection aims at identifying patterns in data that are significantly different to what is expected. This problem is inherently a binary classification problem that classifies examples either as in-distribution or out-of-distribution (OOD), given a sufficiently large sample from the in-distribution (training set). A natural approach to OOD detection is to learn a density model from the training data and compute the likelihood ratio of OOD examples. However, in practice this approach frequently fails for high-dimensional data (Nalisnick et al. (2019)), where it has been shown that deep generative models can assign higher likelihood to OOD examples than to in-distribution examples. This surprising result is likely the consequence of how existing deep generative models generalise. For example, Variational Autoencoders (Kingma & Welling (2014)) generalise by superposition of examples, which is a consequence of the stochastic nature of the posterior that can map different examples to the same point in latent space. As superposition is an averaging process that reduces the information content it can be expected that examples of lower complexity than training examples can map to high likelihood regions in latent space. Note that it is possible for a datapoint to have high likelihood under a distribution yet be nearly impossible to be sampled, a property known as asymptotic equipartition property in information theory. For autoregressive generative models, such as PixelCNN (van den Oord et al. (2016)), it has been shown that the pixel-by-pixel generation process is strongly determined by the local surrounding of pixels (Chen et al. (2018)), where the fact that nearby pixels of training examples frequently share the same color can explain why mono-chromatic images are assigned a high likelihood (Nalisnick et al. (2019)).

As a consequence, most of the current OOD detection approaches make use of a score function $s(x)$ to classify test examples as in-distribution or OOD. In the case that the examples of the training set are labelled, a simple score can be given by $s(x) = \max_y p(y|x)$, with $p(y|x)$ the softmax probability for predicting class labels, $y \in \{1, .., K\}$ (Hendrycks & Gimpel (2017)). If $s(x)$ is below a threshold the test example is classified as OOD. Labelled data allows to learn representations that are associated with the semantic information shared by the examples in the training set, which can be used for OOD detection. However, the approach suffers from the problem that the score for in-distribution examples can be widely distributed across the interval of possible score values, $s(x) \in [1/K, 1]$, especially if the number of labels are low and the classification task is hard, which strongly increases the false-positive rate. Intuitively, the OOD detection problem should be independent of the hardness of an in-distribution classification task. Consequently, better performance was found for approaches that use labeled data for learning a higher dimensional

representation that encodes for semantic information (Lee et al. (2018b)). In this representation space the in-distribution occupies just a small volume and a random feature vector would be most likely classified as OOD. Another simplification of the OOD detection problem arises if some OOD examples labelled as such and are part of the training set. In this case the OOD detection problem boils down to an unbalanced classification problem (Chalapathy & Chawla (2019)).

In this work, we introduce a general framework for OOD detection problems that does not require label information. Our framework can be widely applied to OOD detection tasks, including visual, audio, and textual data with the only limitation that transformations must be *a priori* known that conserve the semantics of training examples, such as geometric transformations for images, proximity of time intervals for audio recordings (van den Oord et al. (2018)), or randomly masking a small fraction of words in a sentence or paragraph (Devlin et al. (2019)). For visual data we show new state-of-the OOD classification accuracies for standard benchmark data sets, surpassing even the accuracies that include labels as additional information. The key contributions of this work are

- We propose a new OOD detecting framework that is applicable to a wide range of OOD detection problems without need of having access to labeled in-distribution data or to OOD examples that are labeled as such.
- We show that our approach strongly improves OOD detection for challenging tasks in the visual domain
- We find that identifying semantically close examples in the training set is central for reliable OOD detection

2 METHOD

An intuitive approach for OOD detection is to learn a representation that densely maps the in-distribution to a small region within a lower dimensional space (latent space), with the consequence that OOD examples will be found outside this region with high probability. The representation should include all the semantic information of the training set, to ensure that test examples from the in-distribution are not misclassified as OOD, but disregard irrelevant pixel level information that would prevent dense mapping. As learning this mapping by Autoencoders is difficult – as explained in the introduction – we split the OOD detection task into finding a semantically dense mapping of in-distribution onto d-dimensional unit-hypersphere by contrastive learning, followed by classifying examples as semantically close or distant.

2.1 LEARNING SEMANTIC SIMILARITY

The contrastive objective aligns feature vectors $h = h(x)$ that are semantically similar and at the same time distributes examples of the training set almost uniformly over the unit-hypersphere (Wang & Isola (2020)). This representation allows to identify for any test example semantically close example from the training set. The mapping $h(x) = f(x)/||f(x)||$ can be learned from training a deep neural network $f(x)$ to minimise the contrastive loss

$$\mathcal{L}[h] = -\mathbb{E}_{(x,x')\sim\mathcal{T}(x,x')}\left[\log \frac{e^{h(x)^T h(x')/\tau}}{\mathbb{E}_{x_{neg}\sim\mathcal{T}(x)}\left[e^{h(x)^T h(x_{neg})/\tau}\right]}\right] \tag{1}$$

where τ denotes a temperature parameter. Here, each positive pair (x, x') is the result of sampling from a distribution of transformations $\mathcal{T}(x, x')$ that conserve semantics between x and x', with $\mathcal{T}(x)$ the marginal of $\mathcal{T}(x, x')$. For datasets used to benchmark object recognition tasks, this can be realised by picking a single example from the training set and randomly apply e.g. geometric transformations, colour distortions, or cropping. The negative pairs can be generated by applying random transformations to different training examples. We emphasise that the types of transformations and their strengths essentially define the semantics we want to encode and thus determine if, for example, the image of a black swan is classified as OOD for an in-distribution that contains only white swans. The design of transformations that capture the underlying semantics of the training dataset requires either higher level understanding of the data or extensive sampling of different combinations of transformations with evaluation on an in-distribution validation set.

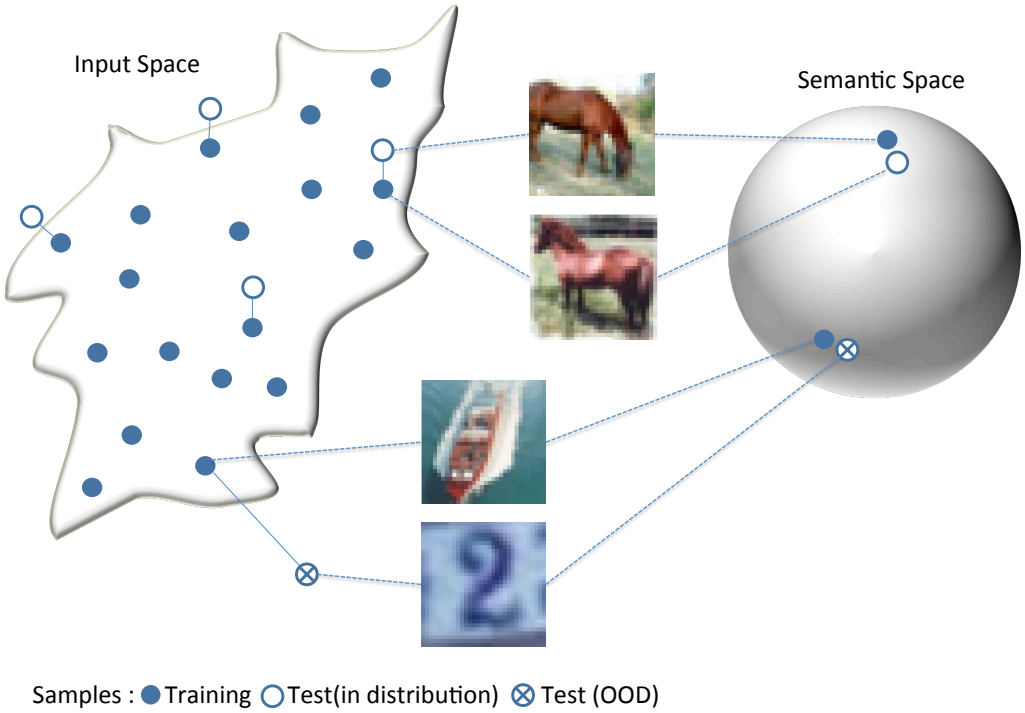

Figure 1: Illustration of mapping examples from the in-distribution onto a unit-hypersphere. In this representation, feature vectors from the in-distribution are semantically similar if they approximately align and semantically diverse if they are separated by a large angle. If OOD examples are mapped onto the unit-hypersphere, they can align with training examples without being semantically similar. A discriminator trained to classify pairs of feature vectors from the training set as semantically close or distant, with sharing or not sharing semantic neighbourhood as target, can then be used for detecting OOD examples.

2.2 Learning Semantic Differences

As the encoder $h(x)$ maps any example x on the unit-hypersphere, including OOD examples, we have the situation that OOD examples can be close to training examples without sharing semantic information. The obvious reason is that the contrastive objective homogeneously distributes training examples on unit-hypersphere, giving no preferred direction for OOD examples to cluster. As a consequence, the representation $h(x)$ is unsuitable for OOD detection. The idea is now to make use of the fact that nearby examples on unit-hypersphere share semantic information if both come from the in-distribution but don't share semantic information if one of the two examples is OOD. We therefore train a score function $s(x, x')$ to detect semantic differences between nearby pairs on the unit-hypersphere. A statically meaningful score can be given by the likelihood $\mathcal{P}_{pos}$ of a test example x_{test} and its next nearest neighbour training example $x_{next} = \arg\max_x h^T(x)h(x_{test})$ to be semantically close in relation to the likelihood $\mathcal{P}_{neg}$ that the same two examples are semantically distant

$$s^*(x_{next}, x_{test}) = \log \frac{\mathcal{P}_{pos}(x_{next}, x_{test})}{\mathcal{P}_{neg}(x_{next}, x_{test})} \tag{2}$$

For the distribution of positive examples we either use $\mathcal{P}_{pos}(x, x') = \mathcal{T}(x, x')$, or sample from the semantic neighbourhood $\mathbb{S}(x)$ on the unit-hypersphere $\mathcal{P}_{pos}(x, x') = \mathbf{1}_{\mathrm{x}' \in \mathbb{S}(\mathrm{x})}/|\mathbb{S}(x)|$, or use a combination both. The semantic neighbourhood $\mathbb{S}(x)$ is defined by the k-nearest neighbours of x using cosine similarity $h^T(x)h(x')$ as distance measure. For negative examples we take $\mathcal{P}_{neg}(x, x') = \mathcal{P}_{pos}(x)\mathcal{P}_{pos}(x')$. In principle negative examples can be constructed that are harder

to classify, such as showing two examples with semantic details removed, e.g. showing a training example and the same example with strong blurring (Choi & Chung (2020)). As shown in Appendix A, the score s^*maximises the training objective $J[s;\gamma]$, given by

$$J[s;\gamma] \quad = \quad \gamma_{pos}\mathbb{E}_{(x,x')\sim\mathcal{P}_{pos}}\Big[\log\sigma(a)\Big] + \gamma_{neg}\mathbb{E}_{(x,x')\sim\mathcal{P}_{neg}}\Big[\log\big(1-\sigma(a)\big)\Big] \tag{3}$$

Here, we defined $\sigma(a) = 1/(1+e^{-a})$, $a = s(x,x')+\log(\gamma_{pos}/\gamma_{neg}))$ and the score function $s(x,x')$ is realised by a deep neural network. It can be further shown that the optimal solution s^* is invariant to any variation in $\gamma_{pos}, \gamma_{neg} > 0$. We introduced $\gamma_{pos}, \gamma_{neg}$ as it is notoriously hard to learn ratios of probability densities in high dimensional spaces, which is a central problem of generative adversarial networks (Azadi et al. (2019)). In general, $s(x,x')$ learned by objective Eq. 3 can deviate significantly from the optimal generalising likelihood ratio $s^*(x,x')$. This deviation is most apparent if $P_{pos}(x,x')$ is zero where $P_{neg}(x,x')$ is non-zero and vice versa. In this case the objective is maximised by any $s(x,x')$ that realises a decision boundary between the distributions $P_{pos}(x,x')$ and $P_{neg}(x,x')$. To smoothen the score function $s(x,x')$ we sample $\gamma_{pos}, \gamma_{neg}$ at each iteration of the learning process and thereby effectively sample over an ensemble of gradients (Appendix B). If we belief in the lottery hypothesis that training of a deep neural networks under constant objective mainly affects the weights of a small subnetwork Frankle & Carbin (2019), we can reason that sampling over γ_{neg} affects the weights for an ensemble of subnetworks. As consequence $s(x,x')$ is the prediction from an ensemble of models, which typically results in higher prediction accuracies, less variance with respect to weight initialisation, and higher robustness to overfitting. It can be shown that for the extreme cases $\gamma_{pos} = 1, \gamma_{neg} \to \infty$ and $\gamma_{neg} = 1, \gamma_{pos} \to \infty$ the score function learns the optimal weights to realise importance sampling for P_{pos} by sampling from P_{neg} and vice versa (Appendix B).

3 Training and Evaluation Protocols

3.1 Training

For all experiments we use a relatively small convolutional network architecture consisting of four convolutional layers followed by a 1-layer MLP projection head – similar to the network used in Ji et al. (2019) – for both the encoder $f(x)$ and the discriminator $s(x,x')$. For contrastive learning the MLP head projects to a 128-dimensional feature vector, whereas for the discriminator the MLP head projects to a scalar output. We train at batch sizes of either 1024 or 2048 using ADAM optimiser. For transformations that conserve semantics we use *random cropping* followed by resizing to the original size, *random horizontal flipping*, *random color jitter*, and *random grey scaling*, sampled with same strength as reported in (Chen et al. (2020)). We set the temperature parameter to $\tau = 1$, which is the value that showed best performance for the datasets used in this work (Winkens et al. (2020)). Unless otherwise specified, the semantic neighbourhood consists of 5 nearest neighbours for each example in the training set as determined by $x_{next} = \arg\max_{x'} h^T(x)h(x')$. We regularise all networks with weight decay of 10^{-6} and either set $\gamma_{pos} = 1, \gamma_{neg} = 1$ or uniformly sample $\gamma_{neg} \sim \mathcal{U}(1,N)$, with $N = 10$, at each iteration of the learning process, with $\gamma_{pos} = 1$. Unless not otherwise specified we set $\gamma_{pos} = 1, \gamma_{neg} = 1$ for experiments. Details about network structures and hyperparameters for training can be found in Appendix C.

3.2 Evaluation

We evaluate the results using Area Under the Receiver Operating Characteristic curve (AUROC), which has the advantage to be scale-invariant – measures how well predictions are ranked, rather than their absolute values – and classification-threshold-invariant – it measures how well OOD samples are separated from the training set. However, for any practical setting of OOD detection a classification threshold is needed and can be chosen such that the false positive rate of an in-distribution test set is close to some threshold, e.g. $\alpha = 0.05$. We do not report values for Area Under the Precision-Recall curve (AUPR) as in this work we have no class imbalance between the OOD test set and the in-distribution test-set.

Table 1: Out-of-distribution detection performance (% AUROC).

Method	$\mathcal{D}_{in}$: CIFAR10 $\mathcal{D}_{out}$: SVHN	CIFAR10 CIFAR100	CIFAR100 CIFAR10
Methods making use of label information			
Softmax probs.(Hendrycks & Gimpel, 2017)	89.9	86.4	77.1
ODIN(Liang et al., 2018)	96.7	85.8	77.2
Mahalanobis(Lee et al., 2018b)	99.1	88.2	77.5
Residual flows(Zisselman & Tamar, 2020)	99.1	89.4	77.1
Outlier exposure(Hendrycks et al., 2019a)	98.4	93.3	75.7
Rotation pred.(Hendrycks et al., 2019b)	98.9	90.9	-
Gram matrix(Sastry & Oore, 2020)	99.5	79.0	67.9
Contrastive Aug.(Winkens et al., 2020)	99.5	92.9	78.3
OpenHybrid(Zhang et al., 2020)	99.8	95.1	85.6
Unsupervised OOD			
Likelihood Regret (Xiao et al., 2020)	86.6	-	-
SVD-RND(Choi & Chung, 2020)	96.9	-	-
GenAD (Ours)	**99.9**	**99.9**	**99.9**

Table 2: Ablation study for different discriminator input representations

	DISCRIMINATOR INPUT			
$\mathcal{D}_{in}/\mathcal{D}_{out}$	$h(x)$	$\mathcal{T}(x)$	Sem. neighbourhood	$\mathcal{T}(x)$ + Sem. neighbourhood
CIFAR100/CIFAR10	57.8	89.3	98.3	99.9

4 EXPERIMENTAL RESULTS

In our experiments, we focus on difficult OOD detection problems in the visual domain (Nalisnick et al. (2019)), in particular CIFAR-10/SVHN, CIFAR-10/CIFAR-100, and CIFAR-100/CIFAR-10. Our main results are summarised in Table 1. We compare our unsupervised method (GenAD) with supervised methods that use label information and/or OOD examples that are labeled as such. Surprisingly, we find that GenAD outperforms not only all unsupervised methods on these problems but also all supervised methods we are aware of. This result is especially striking as supervised methods typically outperform unsupervised methods, as semantic representations learned on label information are typically of advantage. Our interpretation is that learning a representation that includes all semantic information shared between any two examples in the training set helps OOD detection. In contrast, learning from label information results in representations that include mainly the semantics that help predicting labels. Comparison of the SVHN as OOD test set with CIFAR-10 as in-distribution test set, shows that both distributions are unimodal (Fig. 2), which is in contrast to other OOD detection scores (Choi & Chung (2020); Grathwohl et al. (2020)).

In our approach we define semantics by the information that is conserved under a given set of transformations. An important difference between our approach and others is that our network model is comparatively small, and with $\sim 10^6$ parameters about two orders of magnitude smaller than one of the best supervised models (Winkens et al. (2020)). That a small model can compete with larger models is surprising, especially as it has been shown that Contrastive Representation Learning strongly benefits from large model size (Chen et al. (2020)). We expect to see a significant increase in OOD detection performance upon increasing network size, which left to future work.

The performance gain of our method is strong for all OOD detection problems considered in this work but most apparent for CIFAR-100 as in-distribution and OOD examples from CIFAR-10, with increase in state-of -the-art AUROC from 0.783 to > 0.999. Note that the classes of CIFAR-10 and CIFAR-100 are mutually exclusive. Also for the other OOD problems investigated in this work we set new state-of-the-art AUROC values. To see the impact of how positive pairs $(x, x') \sim \mathcal{P}_{pos}$ affect the predictive power of the discriminator we carried out an ablation study. The results show that training the discriminator using the learned feature vectors $h(x)$ as input give the worst result for CIFAR-100/CIFAR-10, whereas training on both transformations and semantic neighbourhoods

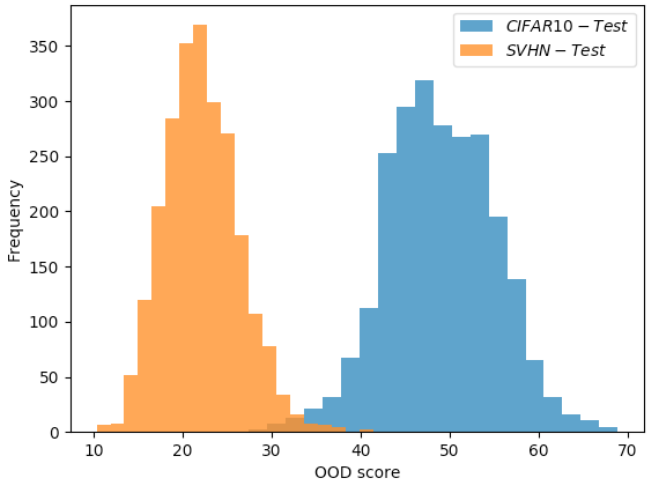

Figure 2: Distribution of our OOD detection score $s(x_{next}, x_{test})$ trained on CIFAR-10 for the SVHN test set (out-distribution) in comparison to the CIFAR-10 test set.

gives the best result (Table. 2). Note that $h(x)$ encodes for features of semantic similarity but not necessarily feature that allow to score semantic dissimilarity. Taking examples from the same semantic neighbourhood as positive example in discriminator training, we effectively make the discrimination task harder, as examples that arise from transforming the same image are semantically closer. In fact, we observe for CIFAR-100 that examples from the same semantic neighbourhood do not always share the same label.

We next test the performance gain of a stochastic training objective for the discriminator that is applied at each iteration in the learning process. As high AUROC values limit statistical interpretation of the results, we focus on the problem CIFAR-100 as in-distribution and CIFAR-10 as OOD, using only transformations for generating positive examples, which results in AUROC of 0.893 for $\gamma_{pos} = 1, \gamma_{neg} = 1$ (Table 2). Setting $\gamma_{pos} = 1$ and sampling from a uniform distribution $\gamma_{neg} \sim \mathcal{U}(1, 10)$ increases the AUROC value of the problem by about 6% to 0.959.

Algorithm 1 Algorithm for training discriminator s

input: γ_{neg} ,s structure
for sampled minibatch $\{(\boldsymbol{x}_k, \boldsymbol{x}_k^p)\}_{k=1}^N$ from paired dataset {for each x uniformly select among the top K best matches } **do**
  sample minibatch $\{\boldsymbol{x}_k^r\}_{k=1}^N$
  for all $k \in \{1, ..., N\}$ **do**
    sample three transformation functions $t_0 \sim \mathcal{T}^{positive}$, $t_1 \sim \mathcal{T}^{positive}$, $t_2 \sim \mathcal{T}^{negative}$
    $\tilde{\boldsymbol{x}}_k = t_0(\boldsymbol{x}_k)$
    $\tilde{\boldsymbol{x}}_k^p = t_1(\boldsymbol{x}_k^p)$
    $\tilde{\boldsymbol{x}}_k^r = t_2(\boldsymbol{x}_k^r)$
    $\boldsymbol{s}_k^{pos} = s(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}_k^p)$
    $\boldsymbol{s}_k^{neg} = s(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}_k^r)$
    $a_k^{pos} = \boldsymbol{u}^T \boldsymbol{s}_k^{pos} - log(\gamma_{neg})$
    $a_k^{neg} = \boldsymbol{u}^T \boldsymbol{s}_k^{neg} - log(\gamma_{neg})$
    $l_k = -(log(\frac{1}{1+\exp(-a_k^{pos})}) + \gamma_{neg} * log(\frac{1}{1+\exp(a_k^{neg})}))$
  end for
  $L = \frac{1}{N} \sum_{k=1}^{k=N} l_k$
  update the s to minimise L
end for

Algorithm 2 Generate dataset of pairs of top K best matches

input: h structure, K {K as number of top best matches}
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  $topKPaired = []$
  $scores = []$
  for $i = 0 : to : i = M$ **do**
    sample minibatch $\{\boldsymbol{x}'_k\}_{k=1}^{N}$
    for all $k \in \{1, ..., N\}$ **do**
      $scores_k += \exp\left(\frac{h(\boldsymbol{x}_k)^T h(\boldsymbol{x}'_k)}{\|h(\boldsymbol{x}_k)\| \|h(\boldsymbol{x}'_k)\|}\right)$
    end for
  end for
  $topK = FindKBestMatch(scores, K)$
  $topKPaired += [\boldsymbol{x}, topK]$
end for
return $topKPaired$

5 Related Work

Methods using labelled data. Most OOD detection methods make use of labels and generate scores that are either based on class prediction probabilities or on intermediate representations that arise from the classification task. For example, Hendrycks & Gimpel (Hendrycks & Gimpel (2017)) used the maximum of the softmax probabilities (MSP) to discriminate between OOD and in-distribution. More recent approaches Lee et al. (2018a); Winkens et al. (2020); Zhang et al. (2020) use labels to learn an intermediate representation on which a density distribution (e.g. multivariate normal distribution or deep generative network) can be fitted, which then can be used to compute the likelihood of OOD examples. As labels implicitly provide information about the semantic relation of examples in the training set, approaches using label information typically show higher accuracy than unsupervised methods. These approaches are improved by introducing additional parameters or training strategies. For example, MSP was improved by introducing a temperature parameter (Liang et al. (2018)), alternative losses (Lee et al. (2018a); Vyas et al. (2018)), auxiliary objectives (Devries & Taylor (2018); Hendrycks et al. (2019b); Mohseni et al. (2020)), or outlier exposure (Hendrycks et al. (2019a)). Intermediate representations were improved using a multi-head network architecture (Shalev et al. (2018), contrastive learning Winkens et al. (2020), metric learning Masana et al. (2018)).

Unsupervised Methods. Most OOD detection problems have neither access to label information for examples in the in-distribution nor access to a sufficiently large dataset that is *a priori* classified as OOD. If label information is absent, unsupervised OOD detection methods must be used that can be separated in methods that learn semantic proximity by transformations that conserve semantics (GenAD), transformations that destroy semantics Choi & Chung (2020), or from a generative model Xiao et al. (2020). Except GenAD, unsupervised methods have so far underperformed in comparison to supervised methods.

6 Conclusion

In this work we proposed GenAD – a new OOD detection approach that is based on learning semantic neighbourhoods for examples of the in-distribution. We showed that our method outperforms supervised and unsupervised methods on challenging OOD detection tasks. The central idea of our approach is to first learn overlapping neighbourhoods of examples that are semantically close and train a discriminator to classify if pairs of examples share the same neighbourhood or likely come from different neighbourhoods. Our approach requires a definition of semantic similarity, which is the central limitation of our approach. Although semantic similarity can be broadly defined by 'everything that is not noise', high predictive power can only be obtained if the semantics similarity catches the higher order features that are essential for understanding the dataset. In practice, there are problems where the definition of semantic similarity is very difficult. For example, if genome sequence data is the input and the effect on the phenotype of the organism is the true underlying se-

mantics. In this case, it is unclear how transformations of the genome sequence should look like that lead to the same phenotype. In contrast, for the important problem of protein folding, such transformations can be inferred from multiple sequence alignments of sequences that likely conserved the function of a protein.

AUTHOR CONTRIBUTIONS

ACKNOWLEDGMENTS

REFERENCES

Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian J. Goodfellow, and Augustus Odena. Discriminator rejection sampling. In *International Conference on Learning Representations*, 2019.

Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *CoRR*, abs/1901.03407, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning*, pp. 863–871, 2018.

Sung-Ik Choi and Sae-Young Chung. Novelty detection via blurring. In *International Conference on Learning Representations*, 2020.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

Terrance Devries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *ArXiv*, abs/1802.04865, 2018.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019a.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pp. 15637–15648, 2019b.

Xu Ji, Andrea Vedaldi, and João F. Henriques. Invariant information clustering for unsupervised image classification and segmentation. In *International Conference on Computer Vision, ICCV*, pp. 9864–9873, 2019.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018a.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pp. 7167–7177, 2018b.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

Marc Masana, Idoia Ruiz, Joan Serrat, Joost van de Weijer, and Antonio M. López. Metric learning for novelty and anomaly detection. In *British Machine Vision Conference*, 2018.

Sina Mohseni, Mandar Pitale, Jbs Yadawa, and Zhangyang Wang. Self-supervised learning for generalizable out-of-distribution detection. In *AAAI*, 2020.

Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, 2020.

Gabi Shalev, Yossi Adi, and Joseph Keshet. Out-of-distribution detection using multiple semantic label representations. In *Advances in Neural Information Processing Systems*, pp. 7386–7396, 2018.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 2016.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L. Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision*, pp. 560–574, 2018.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *CoRR*, abs/2005.10242, 2020.

J. Winkens, R. Bunel, Abhijit Guha Roy, Robert Stanforth, V. Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, T. cemgil, S. Eslami, and O. Ronneberger. Contrastive training for improved out-of-distribution detection. *ArXiv*, abs/2007.05566, 2020.

Zhisheng Xiao, Q. Yan, and Y. Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *ArXiv*, abs/2003.02977, 2020.

Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. *CoRR*, abs/2003.12506, 2020.

Ev Zisselman and Aviv Tamar. Deep residual flow for novelty detection. *CoRR*, abs/2001.05419, 2020.

A APPENDIX

The objective

$$J[s;\gamma] \quad = \quad \gamma_{pos}\mathbb{E}_{(x,x')\sim\mathcal{P}_{pos}}\Big[\log\sigma(a)\Big] + \gamma_{neg}\mathbb{E}_{(x,x')\sim\mathcal{P}_{neg}}\Big[\log\big(1-\sigma(a)\big)\Big] \tag{4}$$

with $\sigma(a) = 1/(1+e^{-a})$, $a = s(x,x') + \log(\gamma_{pos}/\gamma_{neg}))$, and $\gamma = (\gamma_{pos}, \gamma_{neg})$, has the upper bound $J[s^*;\gamma] \geq J[s;\gamma]$ for all $\gamma_{pos}, \gamma_{neg} > 0$, where s^* is given by

$$s^*(x,x') = \log\frac{\mathcal{P}_{pos}(x,x')}{\mathcal{P}_{neg}(x,x')} \tag{5}$$

under the condition that P_{pos} and P_{neg} have the same support – that is where P_{pos} is non-zero also P_{neg} is non-zero and vice versa.

To proof that assertion we make use of variational calculus (see e.g. C. Bishop, Patter Recognition and ML, Appendix D) to compute the functional derivative $\delta J/\delta s$, which is defined by the integral over a arbitrary test function, $\eta(x,x')$,

$$\frac{J[s+\epsilon\eta;\gamma]}{d\epsilon}\bigg|_{\epsilon=0} \quad = \quad \int \frac{\delta J[s;\gamma]}{\delta s(y,y')}\eta(y,y')dydy' \tag{6}$$

$$= \quad \int \Big[\gamma_{pos}\mathcal{P}_{pos}(x,x')\big(1-\sigma(a)\big) - \gamma_{neg}\mathcal{P}_{neg}(x,x')\sigma(a)\Big]\eta(x,x')dxdx' \tag{7}$$

where we have used that $d\sigma(a)/ds = \sigma(a)\big(1-\sigma(a)\big)$. The optimum can be computed from $\delta J/\delta s|_{s=s^*} = 0$, which results in

$$\gamma_{pos}\mathcal{P}_{pos}(x,x')\big(1-\sigma(a)\big) - \gamma_{neg}\mathcal{P}_{neg}(x,x')\sigma(a) = 0 \tag{8}$$

$$\Rightarrow \quad \frac{\gamma_{pos}\mathcal{P}_{pos}(x,x')}{\gamma_{pos}\mathcal{P}_{pos}(x,x') + \gamma_{neg}\mathcal{P}_{neg}(x,x')} = \frac{1}{1+e^{-s^*(x,x')-\log(\gamma_{pos}/\gamma_{neg})}} \tag{9}$$

$$\Rightarrow \quad s^*(x,x') = \log\frac{\mathcal{P}_{pos}(x,x')}{\mathcal{P}_{neg}(x,x')} \qquad \forall\gamma_{pos},\gamma_{neg} > 0 \tag{10}$$

Note that $J[s;\gamma]$ is not bounded from below, so the optimum is a maximum.

B APPENDIX

We show that optimising the objective $J[s;\gamma_{pos},\gamma_{neg}]$ by gradient ascent, with $\gamma_{pos},\gamma_{neg} > 0$ randomly sampled in each optimisation step, leads to averaging over an ensemble of gradients. A gradient based optimisation method in its simplest form updates the parameters θ, which determine the function $s(x,x')$, by the rule

$$\theta \leftarrow \theta + \alpha \nabla_\theta J[s;\gamma] \tag{11}$$

with α the learning rate and

$$\begin{aligned}
\nabla_\theta J[s;\gamma] &= \gamma_{pos}\mathbb{E}_{(x,x')\sim\mathcal{P}_{pos}}\Big[\big(1-\sigma(a)\big)\nabla_\theta s(x,x')\Big] - \gamma_{neg}\mathbb{E}_{(x,x')\sim\mathcal{P}_{neg}}\Big[\sigma(a)\nabla_\theta s(x,x')\Big] \\
&= \mathbb{E}_{(x,x')\sim\mathcal{P}_{pos}}\left[\frac{\gamma_{pos}}{1+\frac{\gamma_{pos}}{\gamma_{neg}}e^{s(x,x')}}\nabla_\theta s(x,x')\right] \\
&\quad - \mathbb{E}_{(x,x')\sim\mathcal{P}_{neg}}\left[\frac{\gamma_{neg}}{1+\frac{\gamma_{neg}}{\gamma_{pos}}e^{-s(x,x')}}\nabla_\theta s(x,x')\right]
\end{aligned} \tag{12}$$

This result shows that for given $s(x,x')$, random values of $\gamma_{pos},\gamma_{neg} > 0$ weight the expected gradients for the positive examples and for the negative examples differently. As a consquence, $\nabla_\theta J[s;\gamma]$ takes different directions for each parameter update, given fixed (mini-)batch and fixed initial conditions.

If we consider the following limiting cases for $s \approx s^*$

$$\lim_{\substack{\gamma_{pos}\to 1\\ \gamma_{neg}\to\infty}} \nabla_\theta J[s;\gamma] = \mathbb{E}_{(x,x')\sim\mathcal{P}_{pos}}\left[\nabla_\theta s(x,x')\right] - \mathbb{E}_{(x,x')\sim\mathcal{P}_{neg}}\left[e^{s(x,x')}\nabla_\theta s(x,x')\right] \tag{13}$$

$$\approx \mathbb{E}_{(x,x')\sim\mathcal{P}_{pos}}\left[\nabla_\theta s(x,x')\right] - \mathbb{E}_{(x,x')\sim\mathcal{P}_{neg}}\left[\frac{\mathcal{P}_{pos}(x,x')}{\mathcal{P}_{neg}(x,x')}\nabla_\theta s(x,x')\right] \tag{14}$$

and

$$\lim_{\substack{\gamma_{pos}\to\infty\\ \gamma_{neg}\to 1}} \nabla_\theta J[s;\gamma] = \mathbb{E}_{(x,x')\sim\mathcal{P}_{pos}}\left[e^{-s(x,x')}\nabla_\theta s(x,x')\right] - \mathbb{E}_{(x,x')\sim\mathcal{P}_{neg}}\left[\nabla_\theta s(x,x')\right] \tag{15}$$

$$\approx \mathbb{E}_{(x,x')\sim\mathcal{P}_{pos}}\left[\frac{\mathcal{P}_{neg}(x,x')}{\mathcal{P}_{pos}(x,x')}\nabla_\theta s(x,x')\right] - \mathbb{E}_{(x,x')\sim\mathcal{P}_{neg}}\left[\nabla_\theta s(x,x')\right] \tag{16}$$

we see that the objective learns the optimal importance weights of importance sampling. As in this work we have the situation that $\mathcal{P}_{pos}(x,x') = 0$ for some some negative examples, the case $\gamma_{pos} \to \infty, \gamma_{neg} \to 1$ should not be applied, which is why we set $\gamma_{pos} = 1$ and sample uniformly from $\gamma_{neg} \in [1, N]$, with $N > 1$.

C APPENDIX

C.1 TRAINING CONTRASTIVE ENCODER

We use the same contrastive encoder structure for both cifar10 and cifar100 experiments. The structure is similar to Ji et al. (2019), including a stack of convolution layers, batch normalisation and ELU activation followed by MLP head projecting to a 128-dimensional latent space. For the details of the network please refer to Table 3. We use learning rate of 10^{-3} and ADAM optimiser with weight decay of 10^{-6} trained with batch size 1024.

C.2 TRAINING DISCRIMINATOR FOR CIFAR10

To train the discriminator $s(x, x')$, we use almost the same network structure as our contrastive encoder but with smaller width and the MLP layer projects to a scalar output. The discriminator s is trained with learning rate of 10^{-6} using Adam optimiser with weight decay of 10^{-6} on batch size of 2048. Details for this discriminator can be found in Table 3

C.3 TRAINING DISCRIMINATOR FOR CIFAR100

For cifar100 we use a wider discriminator structure than cifar10, but the MLP layer projects to latent space of same dimension as cifar10. the discriminator $s(x, x')$ is trained with learning rate of 10^{-4} using ADAM optimiser with weight decay of 10^{-6} and batch size 2048. Details for this discriminator can be found in Table 3

C.4 GEOMETRICAL TRANSFORMATIONS

As data augmentation, we use the same transformations as in Chen et al. (2020). including, randmoly chosen geometrical transformation from *Cropping, Horizontal Flip, Color Jiter, GrayScale, Gaussian Bluring*

Table 3: Network structures

h	**s (Cifar100)**	**s (Cifar10)**
ConvLayer(5x5x64), BN(),ELU()	ConvLayer(5x5x32),BN(),ELU()	ConvLayer(5x5x16),BN(),ELU()
MaxPool()	MaxPool()	MaxPool()
ConvLayer(5x5x128),BN(),ELU()	ConvLayer(5x5x64),BN(),ELU()	ConvLayer(5x5x32),BN(),ELU()
MaxPool()	MaxPool()	MaxPool()
ConvLayer(5x5x256),BN(),ELU()	ConvLayer(5x5x128),BN(),ELU()	ConvLayer(5x5x64),BN(),ELU()
MaxPool()	MaxPool()	MaxPool()
ConvLayer(5x5x512),BN(),ELU()	ConvLayer(5x5x256),BN(),ELU()	ConvLayer(5x5x128),BN(),ELU()
FullyConnected()	FullyConnected()	FullyConnected()