Joint Classification and Unknown Detection using Class Conditional Probability Calibration

Daniel Brignac*, Adam Cuellar†, Banafsheh Latibari*, Abhijit Mahalanobis*
*University of Arizona, Department of Electrical and Computer Engineering
Tucson, AZ
Email: {dbrignac, banafsheh, amahalan}@arizona.edu
†University of Central Florida, Center for Research in Computer Vision
Orlando, FL
Email: acuellar@crcv.ucf.edu

***Abstract*— The *closed set assumption*, where training classes are fixed at inference, is often impractical as deployed models face *open-set conditions* with unknown classes. This challenge drives the field of *Open-Set Recognition* (OSR), which aims to identify unknown samples during inference. A common approach to OSR involves training on exemplars of unknown objects (also referred to as *known unknowns*), which are examples that do not belong to the closed set of known classes. However, this is infeasible for methods that rely solely on the training samples of the known classes. For such cases, we show that the OSR problem can be effectively tackled by combining the decision confidences of two networks: one trained with softmax cross-entropy and the other with tuplet loss using class anchors. We show that the proposed approach outperforms individual methods across OSR benchmarks, maintaining correct classification and high confidence for known samples while effectively rejecting unknowns.**

I. Introduction

It is well known that deep learners in recent years have shown the capacity to achieve or even surpass that of human level performance [1]. This performance however is typically achieved under the *closed set assumption* in which the classes used for training are fixed and presumed to be the same classes encountered during testing. In many practical applications however, models are deployed under *open-set conditions* where the classes used for training are only a small subset of the infinite surrounding world. In safety critical applications (e.g., autonomous driving) the model must have the ability to separate the encountered *unknown* classes from the *known* classes present in the training data.

Conventionally, deep learners struggle when operating in open-set conditions as they tend to confidently map unknown classes to the known class decision space [2]. This motivates the study of *Open-Set Recognition* [3] in which we seek to extend autonomous systems to operate under these open-set conditions by giving them the ability to recognize the known classes and identity all other classes as unknown.

Early solutions to the open-set recognition problem rely on the use of *known unknowns* [3], [4] during training in which a small set of unknowns or general background classes is used during training to map non-relevant classes to a generic "other" category [5]. The use of such known-unknowns may not always be feasible however, or lead to an poor representation of the unknown space as it is impossible to fully encapsulate the infinite unknown world via a small representative subset. Thus, we must seek solutions that rely only on known classes during training while maintaining robust performance for unknown detection and rejection.

Such solutions that only make use of known classes during training can generally be grouped in one of two categories: softmax produced probability analysis [6], [3], [4], [7] or prototype-based methods [8], [9], [10], [11]. The softmax-based methods make the distinction between known and unknown by either manipulating the logits produced by a network before the softmax function or modifying the softmax probabilities directly such that known samples have significantly higher probability than unknown samples. Protype-based methods rely on representing the known classes by a single point in the latent space known as a prototype (or anchor) and proceed to make the declaration of known or unknown based on distance to respective prototypes. While each category is successful in their own regard, the union of both methodologies into a single solution has remained unstudied and it is unknown if the combination of each methodology yields superior results.

In this work, we combine softmax-based and prototype-based methods to perform joint probability estimation for de- termining if a sample is known or unknown. We take inspiration from [12] and train one net- work with cross-entropy for softmax probabilities and a sec- ond to minimize a sample's distance to its latent-space anchor. Using a Bayesian framework, we estimate the joint probabil- ity of an input belonging to a declared class and the closed set of knowns. This joint probability is high for known classes and approaches zero for unknowns. Our method outperforms the individual methods of softmax and prototype method, as well as modern state-of-the-art solutions on standard open-set recognition benchmarks.

II. Methodology

Our method approach consists of two neural networks: a main classifier whose output is treated as $p(y|x)$, and a secondary network whose output is a proxy for $p(K|y, x)$. We can then estimate the joint probability that the main classifier predicts class y and the secondary classifier agrees with the

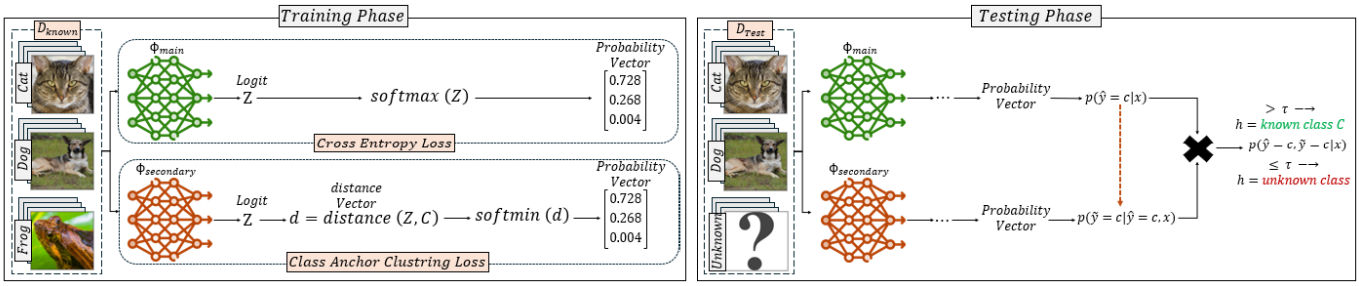


Fig. 1: We train two networks: the first ϕ_{main} to output the logit $\mathbf{z}$ that is then converted to a softmax probability, and the second $\phi_{secondary}$ to output distance $\mathbf{d}$ to class protypes in the latent space that is then converted to a softmin probability. After ϕ_{main} obtains the max prediction probability for class c, we condition $\phi_{secondary}$ on the prediction of ϕ_{main} by analyzing $\phi_{secondary}$'s probability prediction for class c. A joint estimation is then performed and passed to discriminator h to make the declartion of known or unknown.

assignment. From this joint probability, we can i) determine whether a sample is known or unknown and ii) if it is known, its corresponding class. An overview of our approach is shown in Figure 1.

A. The Main Classifier

Consider a neural network $\phi_{main} : x \rightarrow \mathbf{z}^{(1)} \in \mathbb{R}^N$ that maps the input image x to some logit $\mathbf{z}^{(1)}$ in N-dimensional space. This logit $\mathbf{z}^{(1)}$ is then passed through the softmax function to obtain a probability vector as follows

$$softmax(\mathbf{z}_i^{(1)}) = \frac{e^{\mathbf{z}_i^{(1)}}}{\sum_{j=1}^{N} e^{\mathbf{z}_j^{(1)}}} = p(\hat{y}_i|x) \tag{1}$$

where $\hat{y}_i$ is the prediction of class i from ϕ_{main} with corresponding probability value $p(\hat{y}_i|x)$. This formulation of the main classifier is the typical scenario when training with the cross-entropy loss function

$$\mathcal{L}_{CE}(x, y) = -\sum_{j=1}^{N} y_j \log p(\hat{y}_j|x). \tag{2}$$

where y is the one-hot vector encoding of the ground truth class.

Previous works [13], [6] argue that using these probabilities produced by cross-entropy trained networks is satisfactory for unknown detection, however more recent works [14], [9] suggest that a logit's distance to class anchors in the latent space not formed by the cross-entropy loss tend to outperform these purely cross-entropy based methods. Thus, we seek to leverage the power of these prototype methods in conjunction with the standard cross-entropy training described above.

B. The Secondary Network

The secondary network $\phi_{secondary}$ is trained differently from the main classifier, following the Class Anchor Clustering (CAC) approach from [8]. It maps input x to a logit vector $\mathbf{z}^{(2)}$, which is compared to fixed class anchor points $\mathbf{c}_i$ in N-dimensional space to compute a distance vector $\mathbf{d}$. These anchor points are scaled standard basis vectors (not learned), and the training encourages $\mathbf{z}^{(2)}$ to be close to its corresponding class anchor while far from others. This is achieved using a modified Tuplet loss:

$$\mathcal{L}_T(x, y) = \log\left(1 + \sum_{j \neq y}^{N} e^{d_y - d_j}\right) \tag{3}$$

and a penalty term

$$\mathcal{L}_A(x, y) = ||\mathbf{z}^{(2)} - \mathbf{c}_y||_2 = d_y \tag{4}$$

combined into the CAC loss: $\mathcal{L}_{\text{CAC}} = \mathcal{L}_T(x, y) + \lambda \mathcal{L}_A(x, y)$, where λ is a hyperparameter.

To convert the outputs of $\phi_{secondary}$ into class probabilities, we apply a softmin over the distance vector $\mathbf{d}$, assigning higher probability to the closest anchor:

$$softmin(d_i) = \frac{e^{-d_i}}{\sum_{j=1}^{N} e^{-d_j}} = p(\tilde{y}_i|x) \tag{5}$$

This probability is used alongside the main classifier's confidence to compute a joint estimate, allowing us to decide whether to accept the classification or reject x as unknown.

C. Joint Probability Estimation for Unknown Rejection

To decide whether to accept the classifiers' assignment of input x to class c_i or reject it as unknown, we establish a class-conditional relationship between the outputs of ϕ_{main} and $\phi_{secondary}$. Specifically, for ϕ_{main}'s prediction of class i, we use the associated confidence score $p(\hat{y}i|x)$ and combine it with $\phi_{secondary}$'s output probability $p(\tilde{y}_i|\hat{y}_i, x)$ to compute the joint probability that x truly belongs to class i:

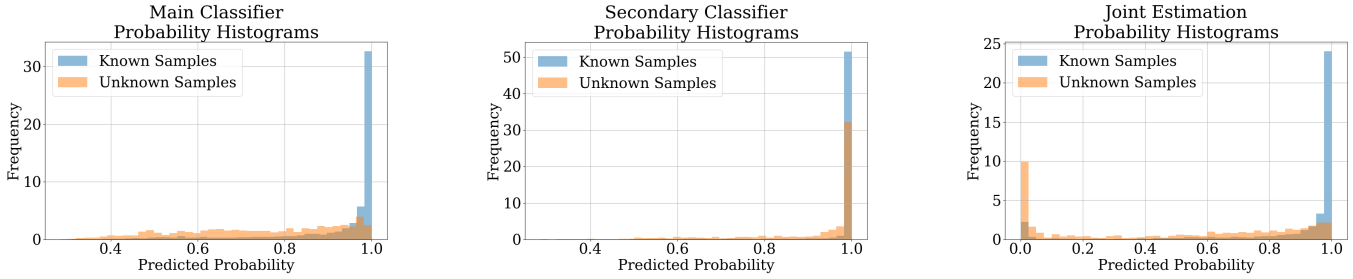
$$p(\hat{y}_i, \tilde{y}_i|x) = p(\tilde{y}_i|\hat{y}_i, x)p(\hat{y}_i|x) \tag{6}$$

This joint probability is then compared to a threshold τ to make the final classification decision: if the probability exceeds τ, x is assigned to class i; otherwise, it is labeled as unknown.

This approach can be interpreted as measuring the agreement between the two networks. High joint probability implies that both networks are confident and aligned in their

Method	MNIST	SVHN	CIFAR10	CIFAR+10	CIFAR+50	Tiny-Imagenet
Softmax	0.7855 ± 0.012	0.8719 ± 0.010	0.7194 ± 0.009	0.7354 ± 0.016	0.7309 ± 0.018	0.5908 ± 0.014
CAC	0.8187 ± 0.011	0.9038 ± 0.015	0.7156 ± 0.002	0.7425 ± 0.013	0.7721 ± 0.002	0.5452 ± 0.036
Good Classifier	0.9894 ± 0.001	0.9058 ± 0.012	0.7479 ± 0.008	0.7734 ± 0.014	0.7720 ± 0.002	0.6291 ± 0.016
ARPL+CS	$\mathbf{0.9900 \pm 0.001}$	$\mathbf{0.9342 \pm 0.005}$	0.7813 ± 0.002	0.8346 ± 0.005	0.8241 ± 0.004	0.6402 ± 0.023
SLCPL [15]	0.8751 ± 0.019	0.8931 ± 0.031	0.7741 ± 0.11	0.8394 ± 0.017	0.8249 ± 0.021	0.6732 ± 0.014
Ours	0.9854 ± 0.001	0.9279 ± 0.001	$\mathbf{0.7906 \pm 0.0004}$	$\mathbf{0.8436 \pm 0.004}$	$\mathbf{0.8427 \pm 0.001}$	0.6848 ± 0.001
Ours (Reversed)	0.9777 ± 0.012	0.9267 ± 0.005	0.7891 ± 0.0009	0.8267 ± 0.015	0.8262 ± 0.015	$\mathbf{0.6941 \pm 0.009}$

TABLE I: Reported AUROC score means and standard deviations averaged over 3 runs.



(a) (b) (c)

Fig. 2: CIFAR+10 probability histograms for (a) main classifier, (b) secondary classifier, and (c) joint estimation.

prediction, favoring acceptance of x as a known class. Conversely, disagreement or low confidence leads to a low joint probability, prompting rejection as unknown. Viewed this way, the method also serves as a form of probabilistic calibration between the classifiers.

III. Experiments and Results

A. Experimental Setup

Datasets. We test our method on six commonly used datasets in open-set recognition literature. In the MNIST and SVHN datasets, we randomly choose 6 classes as known and the remaining 4 classes are treated as unknown. In the CIFAR10 experiments, we treat the 6 non-vehicle classes as known classes the the remaining 4 vehicle classes as unknown. CIFAR+M experiments consider the 4 vehicle classes from CIFAR10 as known and randomly samples M disjoint classes from the CIFAR100 dataset. Lastly, for the Tiny-Imagenet experiments we randomly select 20 classes as the known classes and consider the remaining 180 classes as unknown.

Metrics. We use the standard area under the ROC curve (AUROC) to evaluate the performance of all compared methods. The AUROC performance evaluation lends itself as a threshold independent metric plotting the true positive rate against the false positive rate by varying a threshold. It may be interpreted as the probability a positive (known) sample is assigned a higher detection score than a negative (unknown) sample.

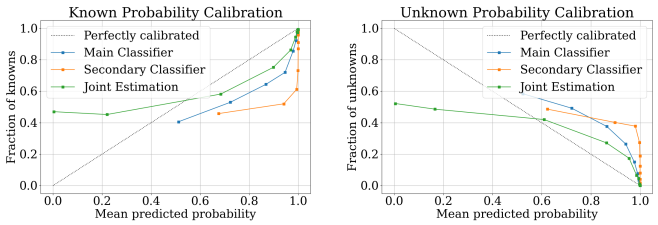
Compared Methods. We compare our method to four open-set recognition approaches with similar methodological foundations. The softmax baseline [6] sets a threshold on class probabilities after applying the softmax function. Class Anchor Clustering (CAC) [8] introduces a loss that encourages known classes to cluster around anchor points in the latent space, leaving unknowns to occupy the remaining regions. ARPL+CS [9] enhances this by training reciprocal points for each class and generating confusing samples to promote separation in latent space, using distance to reciprocal points to assess class membership. Lastly, [7] argues that a strong closed-set classifier alone can suffice, using the maximum logit score to identify unknowns. Importantly, we exclude comparisons to methods that rely on known-unknown samples during training, focusing instead on approaches applicable to purely open-set scenarios.

All methods are trained on the same dataset split using SGD with standard L2 regularization. We use ResNet18 for ϕ_{main} and retain the original architecture and hyperparameters from [8] for $\phi_{secondary}$ to ensure a fair comparison. For consistency, ResNet18 is used across all other methods unless otherwise specified.

B. Results Comparison

We first evaluate the performance of our method vs. all other compared methods from an AUROC standpoint. Table I shows the AUROC results averages across 3 runs for all methods. We observe that our method either outperforms or is very competitive compared to all other methods. We take particular note in the performance gains when testing on the Tiny-ImageNet case. Our method clearly outperforms all others on this much more difficult dataset.

Of note is our method compared directly to the softmax baseline and CAC. The solution proposed in Section II is actually a combination of each method as ϕ_{main} outputs its probabilities using the softmax function and $\phi_{secondary}$ is trained using the CAC loss as proposed in [8]. In all cases our method handily outperforms both the isolated trials of the



(a) (b)

Fig. 3: CIFAR+10 probability calibration curves of main classifier, secondary classifier, and joint estimation for (a) known test samples, (b) unknown test samples.

softmax baseline and CAC demonstrating that the probability joint estimation using both softmax and CAC yields superior known recognition and unknown detection performance.

We additionally report the results of our method in the reverse order. The secondary network could be used as the main classifier simply by assigning x to the class corresponding with largest probability value. Thereafter, we select the confidence score of the first network for the same class. Using the notation in Section II-C, this takes the form of

$$p(\hat{y}_i, \tilde{y}_i|x) = p(\hat{y}_i|\tilde{y}_i, x)p(\tilde{y}_i|x) \tag{7}$$

where we emphasize that ϕ_{mains}'s prediction $\hat{y}_i$ is now conditioned on $\phi_{secondary}$'s prediction of $\tilde{y}_i$. If we compare this form to Equation 6 we see that the joint probability calculation has essentially been reversed. The results of these experiments are reflected in Table I by **Ours (Reversed)**. Even in the reverse case, we can once again infer that our method either outperforms or is very competitive with all compared methods, and once again handily outperforms the individual methods of the softmax baseline and CAC.

C. Effect of Joint Probability Estimation

We assess whether our joint probability estimation effectively distinguishes known from unknown samples by analyzing predicted probabilities on the CIFAR+10 dataset. Figure 2 shows histograms for ϕ_{main}, $\phi_{secondary}$, and our joint method. While ϕ_{main} gives high confidence for knowns, its unknown predictions are widely spread, making separation difficult. $\phi_{secondary}$ shows similar trends for knowns but is overconfident for unknowns, with a peak near 1. In contrast, our joint estimation pushes unknown probabilities toward 0, creating clearer separation—though a small number of knowns are also misclassified as unknowns.

Figure 3 presents probability calibration diagrams to further evaluate prediction reliability. Our joint estimation demonstrates the best calibration, aligning predicted probabilities more closely with true class likelihoods. Unlike ϕ_{main} and $\phi_{secondary}$, which often assign high probabilities to unknowns, our method consistently lowers confidence for unknown inputs, supporting improved open-set recognition.

IV. Conclusion

We propose a joint probability estimation method to classify inputs and identify unknown categories without using known-unknowns during training. Our approach outperforms individual solutions like softmax analysis and CAC's prototype method while remaining competitive with or surpassing modern state-of-the-art methods. It effectively drives unknown sample probabilities to zero while maintaining high probabilities for known samples, as shown in probability histograms and calibration curves.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[2] Anh Nguyen, Jason Yosinski, and Jeff Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[3] Walter J. Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult, "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013.

[4] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult, "Reducing network agnostophobia," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[5] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen, "Recent advances in open set recognition: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3614–3631, 2020.

[6] 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.

[7] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman, "Open-set recognition: A good closed-set classifier is all you need," in *International Conference on Learning Representations*, 2022.

[8] Dimity Miller, Niko Sünderhauf, Michael Milford, and Feras Dayoub, "Class anchor clustering: A loss for distance-based open set recognition," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[9] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian, "Adversarial reciprocal points learning for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8065–8081, 2021.

[10] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian, "Learning open set network with discriminative reciprocal points," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 507–522.

[11] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu, "Robust classification with convolutional prototype learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3474–3482.

[12] Adam Cuellar, Daniel Brignac, Abhijit Mahalanobis, and Wasfy Mikhael, "Simultaneous classification of objects with unknown rejection (scour) using infra-red sensor imagery," *Sensors*, vol. 25, no. 2, pp. 492, 2025.

[13] 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.

[14] Yu Shu, Yemin Shi, Yaowei Wang, Tiejun Huang, and Yonghong Tian, "P-odn: Prototype-based open deep network for open set recognition," *Scientific reports*, vol. 10, no. 1, pp. 7146, 2020.

[15] Ziheng Xia, Penghui Wang, Ganggang Dong, and Hongwei Liu, "Spatial location constraint prototype loss for open set recognition," *Computer Vision and Image Understanding*, vol. 229, pp. 103651, 2023.