

Learning Deep Features for Limited-data Visual Problems

Jie Hong

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

April 2024

© Jie Hong 2024
All rights reserved.

Except where otherwise indicated, this thesis is my original work. The nature and extent of collaboration have been outlined in this thesis.

Jie Hong
23 April 2024

To my family.

Declaration

I have conducted my doctoral research under the expert guidance and supervision of Dr. Lars Petersson, Dr. Mehrtash Harandi, Prof. Nick Barnes, and Dr. Jing Zhang. I hereby declare that this thesis is original and contains no material previously accepted for awarding any degree or diploma in any other university.

Most of the findings presented in this thesis have been published in peer-reviewed international journals or conferences. Some of these findings were achieved through collaborative efforts with other researchers. Below is a list of co-authored publications that have emerged as a part of my thesis research.

0.1 Published Papers

- **Hong, J.**, Fang, P., Li, W., Zhang, T., Simon, C., Harandi, M., & Petersson, L. (2021). Reinforced attention for few-shot learning and beyond. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 913-923).
- **Hong, J.**, Hayder, Z., Han, J., Fang, P., Harandi, M., & Petersson, L. (2023). Hyperbolic Audio-visual Zero-shot Learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 7873-7883).
- **Hong, J.**, Fang, P., Li, W., Han, J., Petersson, L., & Harandi, M. (2023). Curved geometric networks for visual anomaly recognition. IEEE Transactions on Neural Networks and Learning Systems.
- **Hong, J.**, Li, W., Han, J., Zheng, J., Fang, P., Harandi, M., & Petersson, L. (2023). Goss: Towards generalized open-set semantic segmentation. The Visual Computer, 1-14.

0.2 Under-review Papers

- **Hong, J.**, Qiu, S., Li, W., Anwar, S., Harandi, M., Barnes, N., & Petersson, L. (2023). PointCaM: Cut-and-Mix for Open-Set Point Cloud Learning. Under review by Pattern Recognition.

0.3 Non-lead Author Papers

- Zheng, J., Li, W., **Hong, J.**, Petersson, L., & Barnes, N. (2022). Towards open-set object detection and discovery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (pp. 3961-3970).
- Han, J., Fang, P., Li, W., **Hong, J.**, Armin, M. A., Reid, I., Petersson, L., & Li, H. (2022, June). You only cut once: Boosting data augmentation with a single cut. In International Conference on Machine Learning (ICML) (pp. 8196-8212). PMLR.
- Han, J., Li, W., Fang, P., Sun, C., **Hong, J.**, Armin, M. A., Petersson, L., & Li, H. (2022, October). Blind image decomposition. In European Conference on Computer Vision (ECCV) (pp. 218-237). Cham: Springer Nature Switzerland.
- Zheng, Q., **Hong, J.**, & Farazi, M. (2023, June). A generative approach to audio-visual generalized zero-shot learning: Combining contrastive and discriminative techniques. In 2023 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
- Li, W., Farazi, M., **Hong, J.**, & Petersson, L. (2023, November). REIN: Reusing ImageNet to Improve Open-set Object Detection. In 2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA) (pp. 523-530). IEEE.
- Wang, Y., **Hong, J.**, Cheraghian, A., Rahman, S., Ahmedt-Aristizabal, D., Petersson, L., & Harandi, M. (2024). Continual Test-time Domain Adaptation via Dynamic Sample Selection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (pp. 1701-1710).
- Wang, Y., Cheraghian, A., Hayder, Z., **Hong, J.**, Rahman, S., Ramasinghe, S., Ahmedt-Aristizabal, D., Li, X., Petersson, L., & Harandi, M. (2024). Backpropagation-free Network for 3D Test-time Adaptation, In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
- Han, P., Ye, C., Zhou, J., Zhang, J., **Hong, J.** & Li, X., (2024). Latent-based Diffusion Model for Long-tailed Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

Acknowledgments

In the journey of academia, support and guidance are fundamental to the success of a Ph.D. candidate. Reflecting on this intense yet rewarding journey, I find myself overwhelmed with gratitude towards those who have paved the way for my academic and personal growth.

First, I like to express my heartfelt thanks to Dr. Lars Petersson, who is my primary supervisor. His consistent encouragement and intelligence are important to shaping my Ph.D. experience into a productive and successful venture. The investment of his time, profound ideas, and patience throughout my candidature are invaluable.

Equally deserving of recognition are my supervisory panels, Dr. Mehrtash Harandi, Prof. Nick Barnes, and Dr. Jing Zhang. Their guidance illuminates the complex path of research, providing critical insights and invaluable feedback at various stages of my doctoral work. The opportunity to work under their supervision has been a significant highlight of my academic journey.

The scholarly communities at the Australian National University (ANU) and Data61-CSIRO provide a ground for my research. I extend my appreciation to all my friends and colleagues there for creating a warm and collaborative working environment. A special mention goes to my dear friends, Weihao Li, Pengfei Fang, Shi Qiu, Ali Cheraghian, and Zeeshan Hayder, whose insightful discussions have great inspirations for our collaborations.

Finally, the most profound and sincere of my gratitude is reserved for my family and Ms. Guichun Lin. Their unconditional love, boundless patience, and unwavering support have been the bedrock of my existence. The sacrifices they made and the faith they had in me fueled my perseverance. This thesis, along with all my achievements, is a tribute to them.

As I stand in the place of a new beginning, I carry forward the lessons learned, the friendships made, and the support received with utmost gratitude and humility. To all who have been part of this journey, I extend my deepest thanks. Your contribution to my scholarly pursuit has been indispensable and will forever be cherished.

Abstract

In the realm of deep computer vision, especially when faced with the challenge of limited data, leveraging innovative approaches for learning deep features becomes critical. This thesis presents five works to tackle limited-data visual problems, including low-shot and open-set scenarios.

In the first work, few-shot recognition is addressed by introducing an attention mechanism to the backbone network. This mechanism, trained via reinforcement learning, adaptively locates representative regions on feature maps of the backbone. The policy gradient algorithm aids this process, focusing the mechanism for better generalization across unknown classes of data. The results demonstrate that progressively discriminative feature maps are learned, which is evident in the performance of few-shot image classification.

The problem of audio-visual zero-shot learning is studied through a curved geometric learning approach in the second work. With a recognized ability to discover hierarchical structure in data, the adaptive hyperbolic projection facilitates learning discriminative feature embeddings for audio-visual samples. Moreover, the cross-modality alignment in the hyperbolic space is key in this approach, leading to more effective feature fusion across modalities.

Moving to the challenge of open-set recognition, curved geometric learning continues to take the spotlight in the third work. The previous work demonstrates that hyperbolic geometry can encode richer structural information. For the open-set recognition task, we extend the approach using more types of curved geometries, including spherical and mixed geometries. These geometric constraints are interchangeably employed depending on the task. Two innovative geometric modules are proposed, replacing the original Euclidean space for learning feature embeddings and improving performance in many visual open-set recognition tasks.

From the third work, we find that open-set semantic segmentation sometimes leaves big regions for unknown objects on the image unprocessed. To this end, in the fourth work, we extend open-set semantic segmentation to the more encompassing task, *i.e.*, generalized open-set semantic segmentation (GOSS). GOSS not only detects unknown regions but also enhances the agent’s decision-making ability by further predicting more segments inside the unknown region. By integrating both pixel classification and clustering mechanisms, GOSS offers a more comprehensive open-set segmentation solution.

Finally, in the fifth work, the thesis sheds light on the practical challenges of point cloud learning. A novel Point Cut-and-mix (PointCaM) mechanism is introduced to address open-set point cloud learning scenarios. PointCaM mechanism simulates out-of-distribution data and distinguishes between known and unknown data, achieving the more discriminative features of unknown point cloud data and showcasing promising results in its experimental evaluation.

In conclusion, throughout this thesis, five innovative techniques or strategies are presented to optimize the learning of deep features, particularly in scenarios marked by data scarcity.

Contents

Declaration	vii
0.1 Published Papers	vii
0.2 Under-review Papers	vii
0.3 Non-lead Author Papers	viii
Acknowledgments	ix
Abstract	xi
1 Introduction	1
1.1 Introduction	1
1.2 Thesis Outline	6
2 Background	7
2.1 Curved Geometry	7
2.1.1 Notation	7
2.1.2 Spherical Geometry	7
2.1.3 Hyperbolic Geometry	9
2.2 Low-shot Learning	10
2.3 Open-set Recognition	11
I Limited-data Visual Problem: Low-shot Learning	13
3 Reinforced Attention for Few-shot Learning	15
3.1 Introduction	15
3.2 Related Work	17
3.3 Methodology	19
3.3.1 Problem Statement	19
3.3.2 Policy Design	21
3.3.3 Policy Training	22
3.4 Experiments	23
3.4.1 Datasets	23
3.4.2 Few-shot Image Classification	24
3.4.2.1 <i>mini</i> ImageNet	25
3.4.2.2 CUB-200-2011	25
3.4.3 Image Classification	26

3.4.4	Ablation Study	27
3.4.5	Discussion	28
3.5	Summary	29
4	Curved Geometry for Audio-visual Zero-shot Learning	31
4.1	Introduction	31
4.2	Related Work	34
4.2.1	Audio-visual Zero-shot Learning	34
4.2.2	Hyperbolic Geometry	34
4.3	Methodology	35
4.3.1	Baseline	35
4.3.2	Hyper-alignment	36
4.3.3	Hyper-single	37
4.3.4	Hyper-multiple	38
4.4	Experiments	38
4.4.1	Result Analysis	39
4.4.2	Ablation Study	40
4.4.3	Discussion	41
4.5	Summary	43
II	Limited-data Visual Problem: Open-set Recognition	45
5	Curved Geometric Learning for Open-set Recognition	47
5.1	Introduction	48
5.2	Related Work	50
5.2.1	Open-set Image Recognition	50
5.2.2	Geometric Learning	51
5.3	Methodology	52
5.3.1	Geometric-in-One Model	53
5.3.2	Geometric-in-Two Model	55
5.3.3	Model Training	56
5.4	Experiments	57
5.4.1	Multi-Class OOD Detection	57
5.4.2	One-Class Open-set Detection	58
5.4.3	Multi-Class Open-set Segmentation	60
5.4.4	One-Class Anomaly Segmentation	60
5.4.5	Abalation Study	62
5.4.6	Discussion	62
5.4.7	Visualization	65
5.5	Summary	66

6	Generalized Open-set Semantic Segmentation	69
6.1	Introduction	69
6.2	Related Work	72
6.3	Format and Metric	73
6.3.1	Task Format	73
6.3.2	Evaluation Metrics	73
6.4	Methodology	75
6.4.1	Baseline	75
6.4.1.1	Pixel Classification	75
6.4.1.2	Pixel Identification	76
6.4.1.3	Pixel Clustering	77
6.4.1.4	Fusion Module	77
6.4.2	GOSS Segmentor	77
6.4.2.1	Confidence Adjustment	78
6.4.2.2	Pixel Contrastive Learning	78
6.5	Benchmark	79
6.5.1	COCO-Stuff-GOSS	79
6.5.2	Cityscapes-GOSS	80
6.6	Experiments	80
6.6.1	Implementation	80
6.6.2	Results	81
6.6.3	Ablation Study	82
6.6.4	Discussion	83
6.7	Summary	84
7	Cut-and-Mix for Open-set Point Cloud Learning	85
7.1	Introduction	85
7.2	Related Work	88
7.2.1	Point Cloud Learning	88
7.2.2	3D Open-set Settings	88
7.2.3	Cut-Mix Augmentation	89
7.3	Problem Statement	90
7.3.1	Open-set Point Cloud Learning	90
7.3.2	Benchmarks	90
7.3.3	Evaluations	91
7.4	Methodology	91
7.4.1	Baseline Method	91
7.4.2	PointCaM Mechanism	91
7.4.2.1	Unknown-Point Simulator	92
7.4.2.2	Unknown-Point Estimator	93
7.5	Experiments	96
7.5.1	Implementation	96
7.5.2	Open-set 3D Semantic Segmentation	96
7.5.3	Open-set 3D Classification	98

7.5.4 Ablation Study	99
7.5.5 Discussion	101
7.6 Summary	103
8 Conclusion	105

List of Figures

1.1	The limited-data visual problem. Here is the case from open-set object detection Dhamija et al. [2020a]. The data on the “elephant” is never known to the visual model during the model training stage. Consequently, the “elephant” is wrongly detected as a “train” or ignored during the evaluation. The image of “elephants” is copied and modified from Dhamija et al. [2020a].	2
2.1	An illustration of Euclidean space and curved spaces (<i>i.e.</i> , spherical and hyperbolic spaces). (a) The Euclidean space $\mathcal{M}_E$ represents a flat space without curvature. (b) In contrast, the spherical space $\mathcal{M}_S$ restricts all points to lie on the surface of a hypersphere. (c) The hyperbolic space $\mathcal{M}_H$, represented by the Poincaré ball model Nickel and Kiela [2017]; Khruikov et al. [2020], makes points exist inside the ball, demonstrating its negatively curved nature. The transformation from Euclidean space to curved spaces can be achieved through respective spherical or hyperbolic transformations, which project points onto the corresponding curved space.	8
2.2	Low-shot learning. Here is the case from the few-shot image classification Liu et al. [2020a]. The t-SNE visualization of the feature embeddings and the corresponding 1-shot accuracy in the unknown classes of the <i>mini</i> ImageNet dataset using three different algorithms. the best algorithm achieves the 1-shot accuracy of 63.08% on 4 unknown classes, which surpasses the worst one by 10.48%. The image is copied from Liu et al. [2020a].	11
2.3	Open-set recognition. Here is the case from open-set detection Zheng et al. [2022]. The model has the training data from “cat”, “dog” and “bird”. However, the model does not access the data from the “vase.” The open-set recognition requires the model to detect the objects of the “vase” behind the “dog” and “bird”. The image is copied from Zheng et al. [2022].	12
3.1	Few-shot learning processes. (a) Vanilla few-shot learning. (b) Few-shot learning with RAP. The learning process has been formulated as a Markov Decision Process (MDP).	16

3.2	The frameworks of baseline model and RAP model. (a) Baseline model. Given the input image I_o , the backbone network generates the embedding $\mathbf{e}$. (b) RAP model. The recurrent formulation is built where each input image is processed over multiple time steps. The designed policy module is combined with the backbone network to receive the state $\mathbf{s}_t$ and then compute the action $\mathbf{a}_t$. The embedding computed after the final time step $\mathbf{e}_T$ is the resulting embedding.	20
3.3	The framework of the policy module. The module maps the state $\mathbf{s}_t$ to the action $\mathbf{a}_t$. $\sigma(\cdot)$ is Sigmoid function.	21
3.4	The recorded accuracy of image classification of LeNet on CIFAR-10 and VGG-16 on STL-10.	30
4.1	We introduce the Hyperbolic Alignment Module, represented by the block with the blue line. The input data features from each modality are encoded by two consecutive networks (encoder and projector). A cross-attention module in between is used to explore the natural correspondence between visual and audio features. Before the cross-attention module processes the features, our Hyperbolic Alignment Module computes a hyperbolic alignment loss, which aims to explore more hierarchy in audio-visual data. For example, the model may discriminate embeddings of "Playing piano" and "Walking the dog" when it finds that these embeddings belong to different superclasses: "Playing musical instruments" and "Walking/exercising/playing with animals".	32
4.2	The frameworks of the proposed Hyperbolic Alignment Module: (a) Hyper-alignment, which utilizes the hyperbolic space to learn feature alignment; (b) Hyper-single, which automatically computes the curvature using the Hyper-adaptive module after concatenating visual and audio features as input, and (c) Hyper-multiple, which performs alignment using multiple learnable curvatures. While Hyper-alignment uses a fixed curvature, Hyper-single and Hyper-multiple enable flexible exploration of intrinsic data structures by adapting the curvature. Hyper-multiple is particularly suited when diverse manifold structures coexist, improving the generality of the learned representations.	36
4.3	Visualization examples on UCF-GZSL ^{cls} . We give t-SNE visualizations of $\mathbf{e}_v$ from six known classes: "Archery", "CricketShot", "FieldHockeyPenalty", "HandstandPushups", "HandstandWalking" and "PlayingCello". They can be categorized into three superclasses: "Sports", "Acrobatics" and "Playinginstruments". Hyper-multiple, which learns a hyperbolic alignment loss, pushes away features from different superclasses.	43

-
- 4.4 Visualization examples on ActivityNet-GZSL^{cls}. We give t-SNE visualizations of $\mathbf{e}_v$. The features $\mathbf{e}_v$ of three unknown classes, which are from different parent classes, are visualized in the upper figure. The lower figure depicts the feature distribution of three unknown classes from the same parent class. Comparing two figures, Hyper-multiple pushes away features more from different superclasses than the same superclass. 44
- 5.1 An illustration of the discrimination ability of curved spaces on unknown data. We train two models using Euclidean and hyperbolic geometry on *miniImageNet* dataset Ravi and Larochelle [2017], following the few-shot learning protocol. We randomly choose four unknown classes and plot the class embeddings using t-SNE Van der Maaten and Hinton [2008]. Each unknown class is represented by one color. From this visualization, we can directly observe that within each unknown class, embeddings in hyperbolic space (See the figure on the right) show a more compact distribution than embeddings in Euclidean space (See the figure on left). Moreover, it can be observed that hyperbolic embeddings exhibit improved separations based on the discovered hierarchies. For instance, when compared to the distribution of Euclidean embeddings, hyperbolic embeddings show closer proximity between classes (*i.e.*, the yellow-colored “African hunting dog” and the green-colored “lion”) that belong to the same superclass (*i.e.*, ‘animal’). Conversely, hyperbolic embeddings of classes (*i.e.*, the olive-colored “ant” and the purple-colored “bookshop”), which belong to different superclasses (*i.e.*, “insect” and “building”), are pulled further apart. 49
- 5.2 The general frameworks of the models and their training processes are depicted. (a) The baseline model follows an “encoder-classifier” structure. The image encoder takes the input image I and encodes it into a feature embedding $\mathbf{e}_E$. The classifier, primarily composed of a fully-connected (FC) layer, generates the confidence vector for loss computation. We propose two models in our work: Geometric-in-One (GiO) and Geometric-in-Two (GiT). GiO and GiT differ from the baseline model in integrating the geometric transformation (GT) and geometric FC layer into the classifier. In essence, while the baseline model follows an “encoder-classifier” structure, GiO and GiT adopt an “encoder-geometric-classifier” structure. (b) In the GiO model, the original Euclidean FC layer is replaced with a geometric FC layer based on a curved geometry. (c) GiT model adopts the geometric FC layer as the additional branch. In the training stage, we follow the setting that the training data does not include unknown samples. 53

5.3	The general frameworks of the models and their evaluation processes are depicted. In the evaluation stage, the model processes known and unknown inputs. The geometric score z_G is extracted from the curved embedding for open-set score computation in the GiO model. For the GiT model, the geometric score z_{EG} is obtained via Euclidean and curved embeddings. During the evaluation process, GiO and GiT models use fewer parameters than the baseline.	54
5.4	Visualization: multi-class OOD detection. Density distributions of $\max(\mathbf{c}_E)$, $\tanh(z_S)$, $\tanh(z_{ES})$, $\tanh(z_H)$ and $\tanh(z_{EH})$ of the baseline, SiO, SiT, HiO and HiT models with Dense-BC on CIFAR10→TINc are provided. The obtained AUROC for these five models are 94.8%, 94.7%, 98.1%, 98.7% and 96.0%. More corresponding results can be referred to in Table 5.3.	65
5.5	Visualization: one-class open-set detection. Density distribution of open-set score $OS(z_{EH})$ of HiT based on RotNet* on CIFAR-10 Krizhevsky et al. [2009] is visualized. We choose cases where class₁ , class₅ or class₉ is taken as the normal class. The distribution of one case is shown in one column. More corresponding results can be referred to in Table 5.6.	66
5.6	Visualization: one-class open-set detection. Examples with open-set scores OS computed by RotNet*, HiT, and MiT are provided (see upper figures). In addition, AUROC curves along the training epoch are plotted in the lower figure.	67
5.7	Visualization: one-class anomaly segmentation (anomaly localization). The heatmaps of open-set score $OS(\mathbf{c}_E)$, $OS(z_{ES})$, $OS(z_{EH})$ and $OS(z_{EM})$ of Patch-SVDD [†] , SiT, HiT and MiT on MVTecAD Bergmann et al. [2019] are visualized. We provide examples from “bottle”, “metal nut”, “tile”, and “grid.” More corresponding results can be found in Table 5.8. In addition, for the category “bottle”, AUROC curves of different models along the training epoch are plotted in the lower figure.	68
6.1	Different tasks of image segmentation. For a given input image (a) that contains both known (“person”, “dog” and “vegetation”) and unknown objects (“sheep”, “rail” and “grass”), we show: (b) open-set semantic segmentation (OSS) by pixel identification, (c) generic segmentation (GS) by pixel clustering, and (d) generalized open-set semantic segmentation (GOSS).	70
6.2	Toy model of ground truth and predicted GOSS of an image. The predicted segments for “unknown” are partitioned into true positives TP^{uk} , false positives FP^{uk} , and false negatives FN^{uk}	74

6.3	The framework of the baseline and GOSS Segmentor. (a) Baseline. The input image is fed into the encoder for feature extraction. The dual-branch heads are jointly trained for pixel classification and clustering. Furthermore, pixel-wise contrastive learning is leveraged to learn discriminative feature embeddings. The pixel identification module is designed to recognize sets of pixels of the unknown class from the semantic prediction. The final GOSS output is generated using the identified semantic and grouping predictions. (b) GOSS Segmentor (GST). Confidence adjustment and pixel contrastive learning modules are included.	76
6.4	Visualized segmentation results from GST (N+1-model+CA+CL) on COCO-Stuff-GOSS. The GOSS prediction (f) merges the OSS prediction (c) and the grouping prediction (d). Hence, within GOSS prediction (f), “objects” inside identified unknown regions can be segmented. For example, unknown objects, “paper” in the 1st row, “dog/grass” in the 2nd row, and “bottle” in the 3rd row are correctly outlined even though their classes are unknown. We also have some zoom-in images to show the effectiveness of the GST.	83
7.1	The PointCaM mechanism consists of two main modules: the Unknown-Point Simulator (UPS) simulates the out-of-distribution (OOD) data during the training, and the Unknown-Point Estimator (UPE) is responsible for computing the unknown-class scores.	87
7.2	The detailed structure of the Unknown-Point Estimator (UPE). The figure illustrates the operations of the UPE module during the training process. For simplicity, we only demonstrate an example of fusing three intermediate feature maps. j is an arbitrary integer satisfying $1 \leq j \leq M$	94

7.3	The visualization examples: open-set point cloud semantic segmentation of PointNet on S3DIS- <i>Split</i> with the “Manual-10-3” split. Under the “Manual-10-3” split, “table,” “chair” and “sofa” are taken as unknown classes. We visualize heatmaps of estimated unknown-class scores using different open-set methods, <i>i.e.</i> , MSP Hendrycks and Gimpel [2017], UPS+MSP, and our PointCaM (UPS+UPE). The “blue” and “red” points represent points from known and unknown classes. As we can observe in the second-column figures, the baseline MSP has the essential ability to recognize unknown objects. Using UPS, the model is endowed with superior perception ability for unknown points. For example, in the first-row figure, UPS+MSP identifies more parts of the “unknown table” than MSP. However, the UPS module also causes a side effect on the models’ recognition confidence in known classes. To address this issue, the UPE module exploits the point cloud’s feature context by adaptively fusing the multi-scale feature maps. With the help of both the UPS and UPE modules, our proposed PointCaM mechanism well distinguishes the known and unknown points, as shown in the fourth-column figures.	100
7.4	The visualization examples: open-set point cloud semantic segmentation of PointNet on S3DIS- <i>Split</i> with the “Manual-10-3” split. We visualize the density distributions of the unknown-class score of MSP and UPS+MSP.	101

List of Tables

3.1	Few-shot image classification accuracy on <i>mini</i> ImageNet Ravi and Larochelle [2017]. Published results of MAML and ProtoNet with Conv4 are provided in Finn et al. [2017] and Snell et al. [2017], respectively, while their outcomes under Conv-6 and ResNet-10 are reported in Chen et al. [2019a]. "*" indicates the result is obtained using networks implemented by us.	24
3.2	Comparison with other attention designs applied to few-shot image classification on <i>mini</i> ImageNet Ravi and Larochelle [2017]. Different attention models are compared under the same backbone.	25
3.3	Few-shot image classification accuracy on CUB-200-2011 Wah et al. [2011]. The results of MAML and ProtoNet are provided from Chen et al. [2019a].	26
3.4	Image classification accuracy of all models on CIFAR-10/CIFAR-100 and STL-10 Coates et al. [2011]. LeNet LeCun et al. [1998], VGG-16 Simonyan and Zisserman [2014], ResNet-18, ResNet34 and ResNet-50 are chosen as backbones. Baseline models are trained on both the train set and the validation set. "*" indicates the result is obtained using networks implemented by us.	27
3.5	Fair comparisons on CUB-200-2011 Wah et al. [2011] where baseline models are trained on both the train and validation sets like RAP models. "*" indicates the result is obtained using networks implemented by us.	28
3.6	The results of 5-way classification of RAP-ProtoNet on CUB-200-2011 Wah et al. [2011] using different α	28
3.7	Few-shot image classification accuracy using ProtoNet as a baseline with different regular attention designs on CUB-200-2011 Wah et al. [2011]. "*" indicates the result is obtained using networks implemented by us.	29
3.8	Comparisons among RAP models with different time step T on CUB-200-2011 Wah et al. [2011].	29

4.1	Experimental results of audio-visual zero-shot learning on three datasets (main feature). AVCA Mercea et al. [2022b] is adopted as the baseline for the proposed Hyper-alignment, Hyper-single, and Hyper-multiple modules. The best results in HM and ZSL are in bold . The curvatures of Hyper-alignment on VGGSound-GZSL, UCF-GZSL, and ActivityNet-GZSL are set as -0.5 , -0.2 , and -0.2 . The numbers of adaptive curvatures N_c for Hyper-multiple are 2, 3 and 2.	39
4.2	Experimental results of audio-visual zero-shot learning on three datasets (cls feature). AVCA Mercea et al. [2022b] is adopted as the baseline for the proposed Hyper-alignment, Hyper-single, and Hyper-multiple modules. The best results in HM and ZSL are in bold . The curvatures of Hyper-alignment on VGGSound-GZSL ^{cls} , UCF-GZSL ^{cls} , and ActivityNet-GZSL ^{cls} are set as -0.1 , -0.2 and -0.2 . The numbers of adaptive curvatures N_c for Hyper-multiple are 3, 2 and 3.	40
4.3	Ablation study: curvature. Different curvature values of Hyper-alignment are tested on UCF-GZSL ^{cls} and ActivityNet-GZSL ^{cls} . Besides Hyper-alignment, Euclidean-alignment and Sphere-alignment are also considered. From the table, Hyper-alignment shows advantages over Euclidean-alignment and Sphere-alignment.	41
4.4	Ablation study: mixed curvature learning. Different variants of mixed-curvature geometries are tested on UCF-GZSL ^{cls} and ActivityNet-GZSL ^{cls} . H, S and E indicate hyperbolic, spherical and Euclidean alignments. From the table, Geometry-multiple, which incorporates two hyperbolic geometries, gains the most performance improvements.	41
4.5	Ablation study: effectiveness of multiple curvatures. Different curvature numbers of Hyper-multiple are evaluated on VGGSound-GZSL and VGGSound-GZSL ^{cls}	42
4.6	Ablation study: audio-visual feature alignment. Different locations of aligned features by Hyper-single are tested on datasets UCF-GZSL and ActivityNet-GZSL.	42
4.7	Discussion: δ -Hyperbolicity. We compute δ_{rel} on the feature $\mathbf{e}_v$. The smaller value of δ_{rel} indicates the stronger hyperbolicity inside $\mathbf{e}_v$. Hyper-alignment gets a lower value of δ_{rel} than ACVA, which indicates the representations via the hyperbolic alignment loss learn more hierarchical structures.	42
5.1	The proposed geometric networks in the Geometric-in-One (GiO) Model and the Geometric-in-Two (GiT) Model. "Abb.", "Geo. Components" and "Geo. Score" are "abbreviation", "geometry components" and "geometry score" respectively.	55
5.2	Visual open-set recognition tasks where we evaluate the proposed geometric models.	57

-
- 5.3 **Multi-class OOD detection** on CIFAR-10/CIFAR-100 Krizhevsky et al. [2009] with DenseNet/WRN-28-10 image encoder. Mixed-geometry embedding in the MiO/MiT model includes spherical and hyperbolic embedding. We provide averaging results over 5 OOD datasets: TINC, TINr, LSUNC, LSUNr, and iSUN. "*" indicates results are obtained via a self-implemented network. The curvatures of SiO, SiT, HiO, HiT, MiO, and MiT are set to 1.0, 1.0, -0.01 , -1.0 , $(1.0, -0.01)$, $(1.0, -1.0)$ 58
- 5.4 **Multi-class OOD detection** on CIFAR-10/CIFAR-100 Krizhevsky et al. [2009] with DenseNet/WRN-28-10 image encoder. We provide averaging results over 5 OOD datasets: TINC, TINr, LSUNC, LSUNr, and iSUN. "*" indicates results are obtained via self-implemented networks. The curvature of HiO is set to -0.01 59
- 5.5 **Multi-class OOD detection** on CIFAR-100 Krizhevsky et al. [2009] with DenseNet Huang et al. [2017] and WRN-28-10 Zagoruyko and Komodakis [2016] image encoders. Mixed-geometry embedding in the MiT model combines a spherical embedding and hyperbolic embedding. We provide averaging results over 4 multiple OOD datasets: TINC, TINr, LSUNC, and LSUNr. The curvatures of SiT, HiT, and MiT are set to 1.0, -0.001 , and $(1.0, -0.001)$ 59
- 5.6 **One-class open-set detection** on CIFAR-10 Krizhevsky et al. [2009]. Image-level AUROC in % is given. **class_i** indicates the *i*th class. "Extra" indicates utilizing extra data for training (e.g., using pre-trained models on ImageNetDeng et al. [2009]). "*" indicates results are obtained via a self-implemented network. Mixed-geometry embedding in the MiT model incorporates both the spherical embedding and hyperbolic embedding. RotNet*, SiT, HiT, and MiT adopt WRN-28-10 Zagoruyko and Komodakis [2016] as image encoder while CSI, SiT, HiT, and MiT utilize ResNet-18 He et al. [2016]. The curvatures of SiT, HiT and MiT based on RotNet* are set to 1.0, -0.005 and $(1.0, -0.005)$. The curvatures of SiT, HiT and MiT based on CSI are 1.0, -0.01 , and $(1.0, -0.01)$ 60
- 5.7 **Multi-class open-set segmentation** on StreetHazards Hendrycks et al. [2022]. FPR (95% TPR), pixel-level AUROC, and AUPR in % are given. The metric std over 5 runs. The results of AE, Dropout, and MSP are provided in Hendrycks et al. [2022]. The curvatures of SiT, HiT, and MiT based on MSP are set to 1.0, -0.01 , and $(1.0, -0.01)$ 61

5.8	One-class anomaly segmentation (anomaly localization) on MVTecAD. Pixel-level AUROC in % is given. “Extra” indicates utilizing extra data for training (e.g., using pre-trained models on ImageNetDeng et al. [2009]). “Cpr” indicates using the comparison with the training data for open-set score computation during evaluation. The results of AnoGAN/AE and AVID are provided in Salehi et al. [2021] and Venkataramanan et al. [2020], respectively. Mixed-geometry embedding in the MiO/MiT model includes spherical and hyperbolic embedding. The curvatures of SiO, HiO, MiO, SiT, HiT, and MiT are set to 1.0, -0.01 , $(1.0, -0.01)$, 1.0, -1.0 and $(1.0, -1.0)$. The model sizes of Patch-SVDD [†] , SiO, HiO, MiO, SiT, HiT, and MiT are 1.72M, 1.72M, 1.72M, 1.82M, 1.82M, 1.83M and 1.93M.	62
5.9	One-class anomaly segmentation (anomaly localization) on MVTecAD. Pixel-level AUROC in % is given. “Extra” indicates utilizing extra-data for training (e.g., using pre-trained models on ImageNetDeng et al. [2009]). “Cpr” indicates using the comparison with the training data for open-set score computation during evaluation. The results of LSA are provided in Venkataramanan et al. [2020]. Mixed-geometry embedding in the MiT model includes spherical embedding and hyperbolic embedding. The curvatures of SiT, HiT, and MiT are set to 1.0, -1.0 , and $(1.0, -1.0)$	63
5.10	Multi-class anomaly segmentation on MVTecAD Bergmann et al. [2019]. Pixel-level AUROC in % is given. Mixed-geometry embedding in the MiT model includes spherical embedding and hyperbolic embedding. The curvatures of SiT, HiT, and MiT are set to 1.0, -1.0 , and $(1.0, -1.0)$	63
5.11	Ablation study: multi-class OOD detection. Image-level AUROC in % is given. Different cases are evaluated on CIFAR-10 Krizhevsky et al. [2009] to verify the interactions between hyperbolic or spherical and Euclidean spaces.	64
5.12	Ablation study: one-class anomaly segmentation (anomaly localization). Pixel-level AUROC in % is given. Different fixed curvatures on MVTecAD Bergmann et al. [2019] are evaluated. For each category, different curvatures bring about different performances. Also, the optimal curvature choice is not consistent among different categories.	64
6.1	Comparisons of different image segmentation tasks. Compared to traditional segmentation tasks, GOSS takes better care of unknown objects.	72
6.2	Details of different splits of the COCO-Stuff-GOSS/Cityscapes-GOSS benchmarks. The numbers in the table indicate how many known (or unknown) classes are selected for each data split and how many training (or testing) images are kept. We also provide details of the original COCO-Stuff Caesar et al. [2018] and Cityscapes Cordts et al. [2016] for comparison.	80

6.3	GOSS results of GST (N+1-model+CA+CL) on COCO-Stuff-GOSS under three splits. “CA” is the confidence adjustment, and “CL” is the cross-pixel contrastive learning. The best results of GQ are in bold	81
6.4	GOSS results of GST (N-model+CL) on Cityscapes-GOSS under “Manual-16/3” and “Manual-13/6” splits. “CL” is cross-pixel contrastive learning. The best results of GQ are in bold	82
6.5	Ablation study on COCO-Stuff-GOSS under “Random-111/60” split: clustering method. Two clustering models of GST, super-BPD in a supervised setting and DFC Kim et al. [2020] in an unsupervised setting are compared.	82
6.6	GOSS results of GST (N+1-model+CA+CL) on COCO-Stuff-GOSS under “Manual-20/60” split. “CA” is the confidence adjustment and “CL” is the cross-pixel contrastive learning. The best results of GQ are in bold	84
6.7	Ablation study on COCO-Stuff-GOSS under “Random-111/60” split: training strategy. Two strategies are compared.	84
7.1	Open-set Point Cloud Semantic Segmentation on S3DIS- <i>Split</i> with PointNet, PointNet++ and PointTransformer. FPR (95% TPR), pixel-level AUPR, and AUROC in % are given. “PointCaM” is equal to “UPS+UPE”. β_{max} and α are set to 0.6 and 5.0, respectively.	97
7.2	Open-set Point Cloud Classification on ModelNet40- <i>Split</i> . The closed-set metrics, including classification accuracies over sample/class, and open-set metrics, including FPR (95% TPR), detection error, and image-level AUROC in %, are given. “PointCaM” is equal to “USS+USE”. ModelNet40- <i>Split2</i> is taken as the training dataset, while ModelNet40- <i>Split2</i> plus - <i>Split1</i> is the evaluation dataset.	98
7.3	Open-set Point Cloud Classification on ScanObjectNN- <i>Split</i> . The closed-set metrics, including classification accuracies over sample/class, and open-set metrics, including FPR (95% TPR), detection error, and image-level AUROC in %, are given. “PointCaM” is equal to “USS+USE”. ScanObjectNN- <i>Split2</i> is taken as the training dataset, while ScanObjectNN- <i>Split2</i> plus - <i>Split1</i> is the evaluation dataset.	99
7.4	The ablation study: Unknown point generator. PointNet is performed in Open-set Point Cloud Semantic Segmentation on S3DIS- <i>Split</i> . The metric AUROC in % is given.	101
7.5	The ablation study: β_{max} . PointTransformer is performed in Open-set Point Cloud Semantic Segmentation on S3DIS- <i>Split</i> . The metric AUROC in % is given.	102
7.6	The ablation study: α . PointNet++ is performed in Open-set Point Cloud Semantic Segmentation on S3DIS- <i>Split</i> . The metric AUROC in % is given.	102

7.7	The ablation study: Fusion layers in UPE. PointNet++ is performed in Open-set Point Cloud Semantic Segmentation on S3DIS- <i>Split</i> . The metric AUROC in % is given.	103
7.8	Training time: PointNet is performed in Open-set Point Cloud Semantic Segmentation on S3DIS- <i>Split</i> . The training time for each iteration and model size are provided.	103
7.9	Performance drop: Open-set Point Cloud Classification of PointNet++ on ModelNet40-Split with different ratios of using samples as unknown ones.	104

Introduction

1.1 Introduction

The technology of deep learning revolutionizes the field of computer vision, achieving high accuracy in visual tasks like image classification Krizhevsky et al. [2012]; He et al. [2016] and object detection Girshick et al. [2014]; Girshick [2015]. However, the success of deep-learning visual models relies significantly on the availability of large-scale datasets Deng et al. [2009]; Lin et al. [2014]. This dependency presents a significant challenge, as acquiring such extensive data is often labor-intensive, expensive, and sometimes impractical. The requirement of addressing the data scarcity leads to the emergence of the definitions of limited-data visual problems Scheirer et al. [2012]; Vinyals et al. [2016], which focus on deep-learning visual models that can be trained or adapted effectively with insufficient data.

Limited-data visual problems in deep computer vision refer to scenarios encountered when training visual models with insufficient data Scheirer et al. [2012]; Vinyals et al. [2016]. Unlike scenarios where abundant and diverse data are available, limited-data environments pose difficulties that significantly affect the performance and reliability of deep-learning visual models. For example, Figure 1.1 illustrates a specific case of limited-data visual problems, *i.e.*, open-set detection Dhamija et al. [2020a]. Since the visual model is not exposed to "elephant" data during the training phase, it incorrectly detects the "elephant" belonging to other classes during the evaluation process. The worst case is that the model does not even give the prediction to the other "elephant". In this thesis, we define the limited-data visual problem as the problem where some classes of data are lacking during the visual model training but appear in the evaluation. Hence, throughout this thesis, we define the class of data that can (or cannot) be accessed during the visual model training as the known (or unknown) class. The limited-data visual problem includes well-defined sub-problems like few-shot learning Vinyals et al. [2016]; Finn et al. [2017], zero-shot learning Xian et al. [2018]; Liu et al. [2020b], and open-set recognition Scheirer et al. [2012]; Bendale and Boulton [2016], each presenting unique challenges and requiring innovative solutions.

Addressing limited-data visual problems in deep computer vision presents several challenges. These challenges stem from the inherent complexities of the nature of deep-learning visual models. Understanding these challenges is crucial for devel-

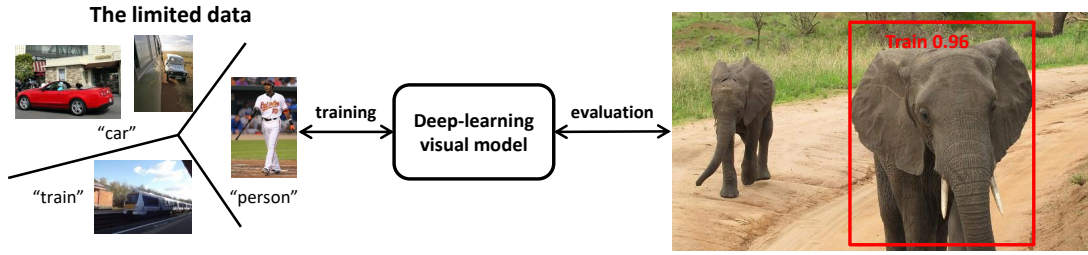


Figure 1.1: The limited-data visual problem. Here is the case from open-set object detection Dhamija et al. [2020a]. The data on the “elephant” is never known to the visual model during the model training stage. Consequently, the “elephant” is wrongly detected as a “train” or ignored during the evaluation. The image of “elephants” is copied and modified from Dhamija et al. [2020a].

oping effective solutions to the problem. Here are the key challenges associated with limited-data visual problems:

- **Overfitting and generalization.** With small training data, deep-learning visual models, which typically require large datasets, are prone to overfitting Finn et al. [2017]. They tend to learn to memorize the known training samples rather than generalize from them Scheirer et al. [2012]. Balancing the complexity of a deep-learning visual model with the limited amount of available data is also challenging since complex models more easily overfit the data Srivastava et al. [2014].
- **Low data diversity.** Limited training data can not capture the full diversity of real-world scenarios, leading to poor performance on data of unknown classes. Visual models trained on limited data tend to develop biases as they are not exposed to a sufficiently varied range of samples Torralba and Efros [2011]; Scheirer et al. [2012]. Moreover, these models trained on limited data are more sensitive to noise in the real-world data, affecting their robustness and reliability Salehi et al. [2022].

Addressing the above core challenges of the limited-data visual problem requires innovative approaches in model architecture, training strategies, data augmentation, or task definition. In particular, designing effective model architectures for limited-data scenarios is important, as standard ones under sufficient data may not be applicable. In this thesis, we develop approaches specifically for five limited-data visual sub-problems, respectively. The explored sub-problems are few-shot learning, audio-visual zero-shot learning, open-set recognition, generalized open-set semantic segmentation, and point cloud 3D open-set learning.

1. Few-shot learning. In the first work, the thesis addresses few-shot learning Vinyals et al. [2016]; Finn et al. [2017], which involves classifying or clustering unknown image samples with a few labeled ones.

- **Motivation.** Traditional approaches, like model-initialization Finn et al. [2017] and metric-learning methods Snell et al. [2017], laid the groundwork for the early progress of few-shot learning. However, they do not consider spatial information within feature maps of the backbone network, leading to the model's sensitivity to the background clutter on image samples Yan et al. [2019].
- **Methodological approach.** In this work, to address the problem, few-shot learning is tackled by integrating a Reinforced-attention policy (RAP) module, using reinforcement learning to adaptively focus on informative regions in feature maps of the model backbone. Specifically, an attention mechanism is implemented along with the backbone network. This mechanism is trained using the policy gradient algorithm Williams [1992] to localize representative regions on feature maps adaptively. Without complex meta-learner designs, this approach enhances the discriminative power of feature embeddings by accurately attending to the model's feature maps.

2. Audio-visual zero-shot learning. The second work of this thesis addresses audio-visual zero-shot learning Mercea et al. [2022b,a], which involves classifying unknown audio and video sequences with the help of semantic attributes of classes.

- **Motivation.** The rich hierarchy inside the audio-visual data has not been leveraged for audio-visual zero-shot learning. Coincidentally, the property of one of the curved geometries, *i.e.*, hyperbolic geometry, is good at capturing the complex hierarchical structure in data Khrulkov et al. [2020]; Fang et al. [2021].
- **Methodological approach.** The proposed approach in this work involves a hyperbolic alignment loss function for cross-modality alignment and introduces multiple adaptive curvatures for hyperbolic projections, aiming to learn more discriminative feature embeddings in this challenging task. Particularly, we first project audio and visual features inside the model to hyperbolic space and align them across modalities. Then, we design a loss function that facilitates the alignment of video and audio features in the hyperbolic space. Furthermore, we explore using multiple adaptive curvatures for hyperbolic projections, enabling a more flexible exploration of data hierarchies.

3. Open-set recognition. The previous two works of the thesis address low-shot learning. From the third work, the thesis starts to explore open-set recognition, which focuses on recognizing unknown objects in data Scheirer et al. [2012]; Bendale and Boulton [2016].

- **Motivation.** The power of curved geometry in achieving more discriminative features of unknown samples has been demonstrated in the second work. Naturally, in the third work, we intend to test curved geometry in another limited-data visual sub-problem, *i.e.*, open-set recognition, which requires unknown features to be more distinct from known features Scheirer et al. [2012]; Bendale and Boulton [2016].

- **Methodological approach.** In this work, the designed curvature-aware geometric networks project Euclidean embeddings to curved space embeddings, including hyperbolic, spherical, or mixed feature embeddings, to represent better and recognize objects of unknown class. This work also formulates functions to compute open-set scores based on curved embeddings.

4. Generalized open-set semantic segmentation (GOSS). In the fourth work, a new task, generalized open-set semantic segmentation (GOSS), is defined not only to identify but also to process unknown regions on the image. GOSS outputs a more comprehensive prediction compared to well-defined open-set semantic segmentation Lis et al. [2019]; Hendrycks et al. [2022].

- **Motivation.** We hope to expand the scope of open-set semantic segmentation (OSS) Lis et al. [2019]; Hendrycks et al. [2022] to not only recognize but also process and analyze unknown regions in images, providing a more comprehensive understanding of the visual data. Using GOSS, we hope to improve the capability of intelligent systems to make informed decisions when encountering unknown objects or scenarios.
- **Methodological approach.** As an expansion of OSS, GOSS aims to classify known pixels and cluster unknown ones, thereby providing a more comprehensive understanding of images containing unknown objects. This newly-defined task combines OSS with the ability to process unknown regions, offering richer scene perception. To achieve this, we propose a unified deep-learning framework with a dual-branch architecture, where one branch focuses on pixel classification and the other on pixel clustering. Moreover, the framework integrates pixel-wise contrastive learning Wang et al. [2021c] to enhance the discriminative power of the learned features, particularly for processing unknown regions of the image. The work also presents new metrics and benchmarks to facilitate the task definition.

5. Open-set point cloud learning. Finally, the fifth work of the thesis addresses open-set recognition in the context of point cloud learning Qi et al. [2017a,b].

- **Motivation.** Developing methods for effectively identifying unknown objects in 3D point cloud data is a relatively unexplored area in open-set recognition Ma et al. [2018]; Bhardwaj et al. [2021]. Recognizing unknown 3D data is important to practical applications in robotics, augmented reality, and other fields where interpreting and interacting with 3D environments are needed Pomerleau et al. [2015]; Chen et al. [2019b].
- **Methodological approach.** The proposed model architecture in this work is the Point Cut-and-Mix (PointCaM) mechanism for the purpose of addressing open-set point cloud learning, comprising the Unknown-Point Simulator (UPS) and Unknown-Point Estimator (UPE) modules. This designed model aims to enhance the model’s ability to distinguish between known and unknown objects in 3D point cloud data. Specifically, the UPS module first manipulates

the known point cloud data to approximate out-of-distribution objects. Then, the UPE module leverages multi-level feature contexts of point clouds to differentiate between known and simulated unknown data. As such, PointCaM enhances the ability to identify unknown objects in point cloud data.

The overall goal of this thesis is to develop innovative models or tasks that effectively address the challenges of limited training data availability in various visual sub-problems. By exploring and combining different approaches, from the reinforced attention mechanism to geometric learning, this thesis aims to push the boundaries of what deep-learning visual models can achieve in scenarios where the training data is incomplete. We conclude the main contributions regarding five works as follows:

- **Advancements in few-shot learning.** The first work of the thesis presents how attention can improve the model’s ability to learn from limited data, leading to better generalization in few-shot learning scenarios. By developing an attention mechanism trained by reinforcement learning, the study contributes to learning more informative feature maps and, thus, more discriminative feature embeddings for few-shot image classification.
- **Innovations in audio-visual zero-shot learning.** The second work of this thesis emphasizes the successful application of hyperbolic geometry in learning features for the sub-problem of audio-visual zero-shot learning. Hyperbolic projection and alignment enable more effective exploration of complex hierarchical structures in audio-visual data, enhancing the discriminative power of feature embeddings for audio-visual zero-shot learning.
- **Contributions to open-set recognition.** Following the second work, where the adoption of hyperbolic geometry is verified, the third work discusses how various curved geometries, *i.e.*, hyperbolic, spherical, and mixed geometries, facilitate better recognition of unknown objects. The feature embeddings are mapped from Euclidean to curved space for discriminative feature embeddings of unknown visual inputs. This work expands the scope and capabilities of open-set recognition visual models via geometric solutions.
- **Development of generalized open-set semantic segmentation (GOSS).** The thesis builds the task of GOSS in the fourth work and designs the GOSS framework, which unifies the identification and clustering of unknown regions in images, providing a more comprehensive scene understanding solution than open-set semantic segmentation. Defining GOSS, we hope intelligent systems can enhance their ability to make accurate decisions when faced with unfamiliar objects or scenarios.
- **Point cloud data learning in open-set settings.** In the fifth work, the thesis proposes the novel PointCaM mechanism, including UPS and UPE modules, and demonstrates its impact on enhancing open-set feature learning in 3D point cloud data. In particular, the UPS module simulates out-of-distribution objects via Cut-and-Mix data augmentation for the model’s training stage, while

the UPE module learns the discriminations between unknown and unknown objects using the context of feature maps and point clouds. In conclusion, PointCaM provides an effective solution for addressing the significant gap in the area of open-set point cloud learning.

We also conclude three key points from this thesis regarding the methodology as follows:

- **Advancements in the feature learning.** This thesis studies how features in the part of the model can be exploited to represent unknown objects better. The first work of the thesis attends to the feature maps of the model backbone, and the second aligns multimodal feature embeddings inside the model in the hyperbolic space. Moreover, the fifth work employs feature maps to compute open-set scores.
- **Application of curved geometry.** The thesis also brings about interdisciplinary contributions by employing curved geometry. The second and third works incorporate curved geometric learning in deep-learning visual models. It has been demonstrated that the interdisciplinary approach using curved geometry leads to performance improvements in several limited-data visual sub-problems.
- **Expanding the scope of open-set recognition.** By extending open-set recognition to a new task, GOSS, in the fourth work and studying open-set recognition in the context of the 3D point cloud in the fifth work, the thesis broadens the application of the concept of open-set recognition. These expansions might help the theoretical development of deep computer vision algorithms capable of handling unknown data in diverse scenarios.

1.2 Thesis Outline

In addition to the introduction in this chapter, the background knowledge of this thesis is introduced in Chapter 2. Five innovative works addressing limited-data visual sub-problems, *i.e.*, reinforced attention for few-shot learning, hyperbolic geometry for audio-visual zero-shot Learning, curved geometric learning for open-set recognition, generalized open-set semantic segmentation, and cut-and-mix for open-set point cloud learning, are presented in Chapters 3-7, respectively. Finally, the conclusion of this thesis is given in Chapter 8.

Background

In this chapter, we introduce the background knowledge we use throughout this thesis. First, we provide the preliminaries of curved geometry. Then, we describe low-shot learning and open-set recognition. Note that specific related works will be introduced in corresponding chapters.

2.1 Curved Geometry

Curved geometry, which is often referred to as non-Euclidean geometry, deals with properties and applications of spaces that are not flat Lee [2006]. It will be used in Chapters 4 and 5. Here, we provide its preliminary knowledge.

2.1.1 Notation

Firstly, we use κ to denote the curvature of a space. In general, a vectorized representation or a feature embedding can be embedded in three types of manifolds: the Euclidean space $\mathcal{M}_E$, the spherical space $\mathcal{M}_S$ and the hyperbolic space $\mathcal{M}_H$, corresponding to $\kappa = 0$, $\kappa > 0$ and $\kappa < 0$, respectively. Three types of spaces are depicted in Figure 2.1 (a), (b), and (c), respectively. Throughout the thesis, we call any space with $\kappa \neq 0$ a curvature-aware space or a curved space. A mixed-curvature manifold $\mathcal{M}_M$ is a product space consisting of a set of different spaces Gu et al. [2019]; Skopek et al. [2020]. In our work, the mixed-curvature manifold is defined as $\mathcal{M}_M = \mathcal{M}_1 \times \mathcal{M}_2 \times \mathcal{M}_3 \times \dots \times \mathcal{M}_N$, in which we mix N different manifolds. For example, the mixed-curvature manifold $\mathcal{M}_M = \mathcal{M}_E \times \mathcal{M}_S \times \mathcal{M}_H$ includes a Euclidean space, a spherical space, and a hyperbolic space.

2.1.2 Spherical Geometry

The spherical space $\mathcal{M}_S$, *i.e.*, the n -sphere with curvature $\kappa > 0$, is defined as

$$\mathcal{S}_\kappa^{n-1} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|^2 = 1/\kappa\} . \quad (2.1)$$

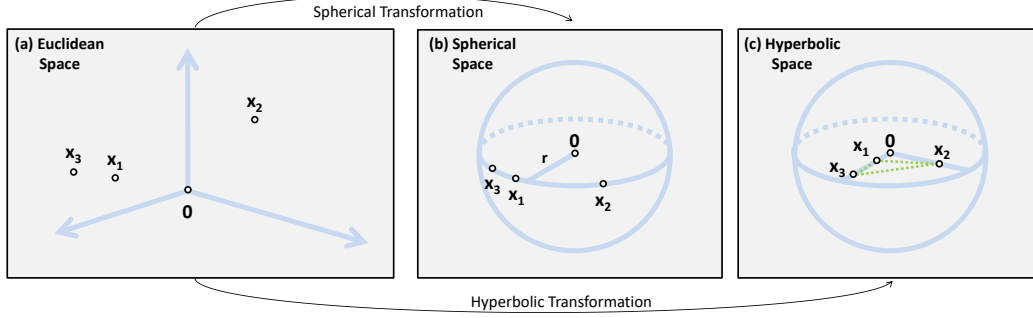


Figure 2.1: An illustration of Euclidean space and curved spaces (*i.e.*, spherical and hyperbolic spaces). (a) The Euclidean space $\mathcal{M}_E$ represents a flat space without curvature. (b) In contrast, the spherical space $\mathcal{M}_S$ restricts all points to lie on the surface of a hypersphere. (c) The hyperbolic space $\mathcal{M}_H$, represented by the Poincaré ball model Nickel and Kiela [2017]; Khrulkov et al. [2020], makes points exist inside the ball, demonstrating its negatively curved nature. The transformation from Euclidean space to curved spaces can be achieved through respective spherical or hyperbolic transformations, which project points onto the corresponding curved space.

The mapping $\Gamma_S : \mathbb{R}^n \rightarrow \mathbb{S}_\kappa^{n-1}$ projects an embedding $\mathbf{x} \in \mathbb{R}^n$ generated by an image backbone to n -sphere (see Figure 2.1 (b)) as:

$$\mathbf{x}_S = \Gamma_S(\mathbf{x}) = \frac{\mathbf{x}}{\sqrt{\kappa} \|\mathbf{x}\|}. \quad (2.2)$$

The embedding $\mathbf{x}_S$ indicates a point in the n -sphere, satisfying the constraint in Equation (2.1). In practice, the angular mapping in the n -sphere, analogous to the linear mapping in the Euclidean space, can be realized by a fully-connected (FC) layer with weight $\mathbf{W}$. Let $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_j, \dots, \mathbf{w}_C]$ where $\mathbf{w}_j \in \mathbb{S}_\kappa^{n-1}$ is the corresponding column, representing the parameters of the classifier. The prediction associated with the j th class for an embedding $\mathbf{x}$ is determined by $\mathbf{w}_j$. We note that for $\mathbf{x}, \mathbf{w}_j \in \mathbb{S}_\kappa^{n-1}$, the term for the j th class, $l_S(\mathbf{x}, \mathbf{w}_j) = \langle \mathbf{x}, \mathbf{w}_j \rangle$, is indeed related to the geodesic distance on $\mathbb{S}_\kappa^{n-1}$, hence one can understand this term as a form of the distance-based value. Here, we use the notation $l_S(\mathbf{x}, \mathbf{W})$ to show a vector obtained by applying the columns of $\mathbf{W}$ to $\mathbf{x} \in \mathbb{S}_\kappa^{n-1}$.

For the spherical network, we compute the angular loss ℓ_S based on B samples in one batch:

$$\ell_S = -\frac{1}{B} \sum_i \log \frac{\exp(l_{S,y_i})}{\sum_j \exp(l_{S,j})} \quad (2.3)$$

where $l_{S,j}$ is the j th element in $l_S(\mathbf{x}, \mathbf{W})$ under the i th input sample and $\mathbf{x} \in \mathbb{S}_\kappa^{n-1}$. Accordingly, l_{S,y_i} is the y_i th element in $l_S(\mathbf{x}, \mathbf{W})$ and y_i indicates the label class to the i th input sample.

2.1.3 Hyperbolic Geometry

Different from the spherical space $\mathcal{M}_S$, the hyperbolic space $\mathcal{M}_H$ is a curved space with a constant negative curvature (*i.e.*, $\kappa < 0$). The hyperbolic space offers an appealing property for limited-data visual problems. Specifically, the volume in the hyperbolic space increases exponentially. This allows the algorithm to incorporate more embeddings of unknown objects, particularly those closer to the origin, as the space expands from the origin Khrulkov et al. [2020].

In this thesis, we employ the Poincaré ball Nickel and Kiela [2017]; Khrulkov et al. [2020] to model and work with the hyperbolic space (see Figure 2.1 (c)). The n -dimensional Poincaré ball, with curvature κ , is defined by the manifold:

$$\mathbb{H}_\kappa^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < -1/\kappa\}^1.$$

To embed $\mathbf{x} \in \mathbb{R}^n$, obtained by a backbone to the Poincaré ball $\mathbb{H}_\kappa^n$, we use the following transformation:

$$\mathbf{x}_H = \Gamma_H(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\| \leq \frac{1}{|\kappa|} \\ \frac{1-\xi}{|\kappa|} \frac{\mathbf{x}}{\|\mathbf{x}\|} & \text{else,} \end{cases} \quad (2.4)$$

where $\xi > 0$ is a small value to ensure numerical stability. The embedding $\mathbf{x}_H$ is a point in the Poincaré ball. To enable the vector operations in the Poincaré ball, we make use of the Möbius addition for $\mathbf{x}, \mathbf{y} \in \mathbb{H}_\kappa^n$ as:

$$\mathbf{x} \oplus_\kappa \mathbf{y} = \frac{(1 + 2|\kappa|\langle \mathbf{x}, \mathbf{y} \rangle + |\kappa|\|\mathbf{y}\|^2)\mathbf{x} + (1 - |\kappa|\|\mathbf{x}\|^2)\mathbf{y}}{1 + 2|\kappa|\langle \mathbf{x}, \mathbf{y} \rangle + |\kappa|^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}, \quad (2.5)$$

where $\langle, \rangle$ is the inner product. The geodesic distance between $\mathbf{x}, \mathbf{y} \in \mathbb{H}_\kappa^n$ is defined as:

$$d_{Geo}(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{|\kappa|}} \tanh^{-1}(\sqrt{|\kappa|} \|\mathbf{x} \oplus_\kappa \mathbf{y}\|). \quad (2.6)$$

One can also generalize the hyperbolic linear operation, parameterized by $\mathbf{W}$ (*e.g.*, the hyperbolic linear layer), in the Poincaré ball Khrulkov et al. [2020]:

$$\mathbf{W}_{\oplus_\kappa}(\mathbf{x}) := \frac{1}{\sqrt{|\kappa|}} \tanh\left(\frac{\|\mathbf{W}\mathbf{x}\|}{\|\mathbf{x}\|} \tanh^{-1}(\sqrt{|\kappa|} \|\mathbf{x}\|)\right) \frac{\mathbf{W}\mathbf{x}}{\|\mathbf{W}\mathbf{x}\|}. \quad (2.7)$$

The proposed network contains the multi-class classification layer. We employ the generalization of multi-class logistic regression (MLR) Khrulkov et al. [2020] to the hyperbolic spaces. Following the work in Khrulkov et al. [2020], the formulation

¹In this case, the Riemannian metric is defined as $g_\kappa^H(\mathbf{x}) = \lambda_\kappa^2(\mathbf{x}) \cdot g^E$, where $\lambda_\kappa(\mathbf{x}) = \frac{1}{1+\kappa\|\mathbf{x}\|^2}$, and g^E is the Euclidean metric.

of the hyperbolic MLR for C classes is given by:

$$l_H(y = j|\mathbf{x}) \propto \exp\left(\frac{\lambda^\kappa(\mathbf{x})\|\mathbf{W}_j\|}{\sqrt{|\kappa|}} \sinh^{-1}\left(\frac{2\sqrt{|\kappa|}\langle -\mathbf{p}_j \oplus_\kappa \mathbf{x}, \mathbf{W}_j \rangle}{(1 - |\kappa| \cdot \|\mathbf{p}_j \oplus_\kappa \mathbf{x}\|^2)\|\mathbf{W}_j\|}\right)\right) \quad (2.8)$$

where $j \in \{1, 2, \dots, C\}$. Here, $\mathbf{x} \in \mathbb{H}_\kappa^n$ is an embedding in the hyperbolic space, and $\mathbf{p}_j \in \mathbb{H}_\kappa^n$, $\mathbf{W}_j \in T_{\mathbf{p}_j} \mathbb{H}_\kappa^n \setminus \{\mathbf{0}\}$ are learnable weights.

The hyperbolic tangent space is a Euclidean space that locally approximates the hyperbolic space Gao et al. [2022]. By selecting a tangent point $\mathbf{z}_H \in \mathbb{H}_\kappa^n$, we can generate a tangent plane $T_{\mathbf{z}_H} \mathbb{H}_\kappa^n$ at $\mathbf{z}_H$. The process of taking the logarithm is used to map a point $\mathbf{x}_H \in \mathbb{H}_\kappa^n$ onto the tangent space $T_{\mathbf{z}_H} \mathbb{H}_\kappa^n$ of $\mathbf{z}_H$, which is given as follows:

$$\mathbf{x}_{Tg} = \frac{2}{\sqrt{|\kappa|}\lambda_\kappa(\mathbf{z}_H)} \tanh^{-1}\left(\sqrt{|\kappa|}\|\mathbf{z}_H \oplus_\kappa \mathbf{x}_H\|\right) \frac{-\mathbf{z}_H \oplus_\kappa \mathbf{x}_H}{\|\mathbf{z}_H \oplus_\kappa \mathbf{x}_H\|} \quad (2.9)$$

where $\mathbf{x}_{Tg} \in T_{\mathbf{z}_H} \mathbb{H}_\kappa^n$ is the transformed point in the tangent space. In this thesis, we consider the identity tangent space of the Poincaré ball at the origin $\mathbf{0}_H \in \mathbb{H}_\kappa^n$, so we have $\mathbf{z}_H = \mathbf{0}_H$.

2.2 Low-shot Learning

In this thesis, the limited-data visual problems we study are categorized into two specific parts, *i.e.*, low-shot learning and open-set recognition. We start with introducing the low-shot learning problem, which is illustrated in Figure 2.2. The main target of the low-shot learning task is to classify or cluster the input samples from unknown classes. As shown in Figure 2.2, using the low-shot learning approaches, we hope to achieve a more separate distribution of features from different unknown classes. The low-shot learning is particularly useful in scenarios where collecting or labeling large datasets is impractical or too expensive. It is known that low-shot learning is mainly divided into two sub-problems, *i.e.*, few-shot and zero-shot learning.

Few-shot learning. Few-shot learning is a task designed to enable models to learn and generalize from a very limited amount of samples Vinyals et al. [2016]; Finn et al. [2017]. That is to say, the model trained on the data of known classes, is required to classify the data of unknown classes only using a few labeled samples. These labeled samples from unknown classes are typically referred to as “shots.” Practically, few-shot learning aims to leverage prior knowledge and intelligent algorithms to understand new tasks with minimal samples. By focusing on extracting and applying features from small datasets, few-shot learning models can adapt to new tasks with high efficiency.

Zero-shot learning. Zero-shot learning enables models to correctly make predictions on objects they have never known during training Xian et al. [2018]; Liu et al. [2020b]. Unlike few-shot learning, zero-shot learning does not require any labeled samples of the unknown classes. Instead, it relies on a rich semantic relationship between known

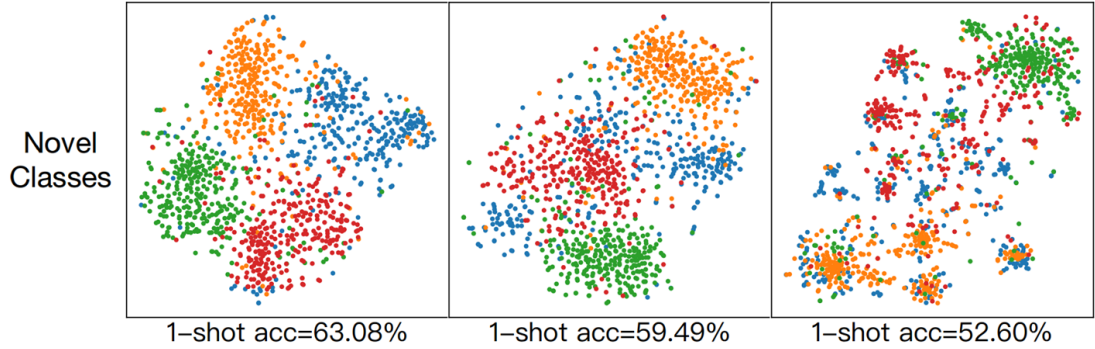


Figure 2.2: Low-shot learning. Here is the case from the few-shot image classification Liu et al. [2020a]. The t-SNE visualization of the feature embeddings and the corresponding 1-shot accuracy in the unknown classes of the *miniImageNet* dataset using three different algorithms. the best algorithm achieves the 1-shot accuracy of 63.08% on 4 unknown classes, which surpasses the worst one by 10.48%. The image is copied from Liu et al. [2020a].

and unknown categories, often using auxiliary information such as class descriptions or attributes. The zero-shot learning allows models to generalize from known to unknown categories based on the understanding that the learned knowledge can be transferred across related tasks.

2.3 Open-set Recognition

The second part of the limited-data visual problems we study in this thesis is open-set recognition. Although modern deep learning has demonstrated great success in closed-set visual recognition tasks, the challenge of open-set recognition has gained attention Scheirer et al. [2012]; Bendale and Boulton [2016]. In a closed-set recognition scenario, the classifier aims to identify a set of categories that remain consistent during both the training and testing phases. On the other hand, in an open-set recognition scenario, the classifier must differentiate between the training categories and determine if an object belongs to an unknown class (see Figure 2.3).

One case of open-set recognition, *i.e.*, open-set detection Dhamija et al. [2020b]; Joseph et al. [2021]; Zheng et al. [2022]; Li et al. [2023] is designed to detect unknown objects from open-set data that includes both known and unknown objects. Similarly, in open-set semantic segmentation Hendrycks et al. [2022], the goal is to segment an image containing known and unknown object classes. In addition to the task settings above, researchers have also been actively exploring the solutions in open-set recognition, and several studies have proposed methods like density estimation Zhang et al. [2020b], uncertainty modeling Liang et al. [2018], and input reconstruction Pidhorskyi et al. [2018] to identify open-set samples. For instance, Zhang et al. [2020b] uses a backbone to encode the input data into a joint embedding space, a classifier

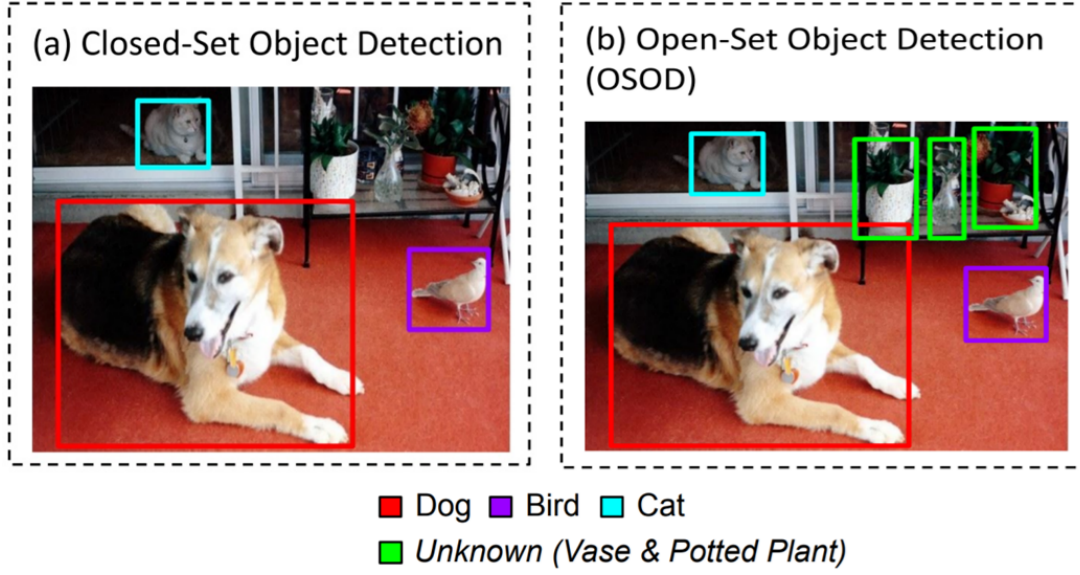


Figure 2.3: Open-set recognition. Here is the case from open-set detection Zheng et al. [2022]. The model has the training data from “cat”, “dog” and “bird”. However, the model does not access the data from the “vase.” The open-set recognition requires the model to detect the objects of the “vase” behind the “dog” and “bird”. The image is copied from Zheng et al. [2022].

to classify samples into known classes, and a flow-based density estimator to detect whether a sample belongs to the unknown class. Pidhorskyi et al. [2018] introduces a network architecture for open-set recognition based on learning reconstruction of input data and detects unknown samples by evaluating its inlier probability distribution.

According to different motivations, there are other problems, including out-of-distribution (OOD) detection Liang et al. [2018] and open-vocabulary recognition Radford et al. [2021]; Pham et al. [2021], which are closely related to open-set recognition. OOD detection focuses on detecting samples significantly different from the training data distribution Liang et al. [2018]; Veeramacheneni and Valdenegro-Toro [2022]; Salehi et al. [2022]. For instance, the solution in Veeramacheneni and Valdenegro-Toro [2022] detects indoor objects, *i.e.*, OOD dataset, from outdoor 3D scenes. Open-vocabulary recognition classifies unknown objects by associating them with textual descriptions in a zero-shot manner using vision-language pretraining models, such as CLIP Radford et al. [2021].

Part I

Limited-data Visual Problem: Low-shot Learning

Reinforced Attention for Few-shot Learning

In this chapter, we focus on one of the low-shot learning problems, *i.e.*, few-shot learning. As introduced in the previous chapter, few-shot learning aims to learn a model to classify unlabeled query samples of unknown classes, utilizing a very small number of labeled support samples. In this chapter, we propose to equip the backbone network with an attention mechanism (or agent), which is trained by reinforcement learning. The policy gradient algorithm trains the agent towards adaptively localizing the representative regions on feature maps of the backbone over time. We further design a reward function based on the prediction of the held-out data, thus helping the attention mechanism to generalize better across the unknown classes. The extensive experiments show, with the help of reinforced attention, that our backbone network can progressively generate discriminative feature embeddings in few-shot image classification. Moreover, experiments on the image classification task also show the proposed design’s effectiveness. This chapter is based on our published work Hong et al. [2021].

3.1 Introduction

To address few-shot learning, one prominent category of methods is the model-initialization-based approach Ravi and Larochelle [2017]; Finn et al. [2017]. It temporarily updates the model parameter using support samples via gradient descent steps for the training tasks and seeks a representation that generalizes well in the testing phase. Another line of work, the metric-learning-based methods Koch et al. [2015]; Snell et al. [2017]; Sung et al. [2018], is based on the complex manipulation of global embeddings learned by the backbone network.

Even though the traditional approaches work well for the task of few-shot learning, they are likely to ignore the spatial information encoded within feature maps of the backbone, which makes the model very sensitive to the background clutter on image samples Yan et al. [2019]. To make full use of the available spatial information, attention-oriented designs have been developed for few-shot learning. Using word embeddings as auxiliary data, semantics-guided attention modules are proposed to

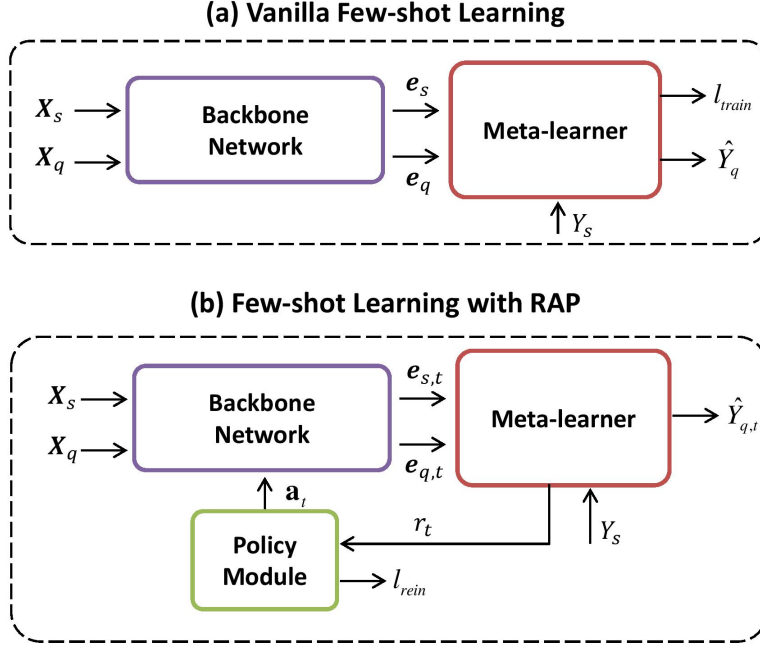


Figure 3.1: Few-shot learning processes. (a) Vanilla few-shot learning. (b) Few-shot learning with RAP. The learning process has been formulated as a Markov Decision Process (MDP).

capture the relevant visual features among query samples Wang et al. [2017a]; Chu and Wang [2018]; Yan et al. [2019]. In addition to the semantics-guided attention, the sample-guided attention designs can further explore the feature relevance between support samples and query samples Vinyals et al. [2016]; Ren et al. [2019b]; Hou et al. [2019]. While these attention models effectively make class features more representative, they tend to focus too much on designing a complex meta-learner.

To address the aforementioned weakness, in this chapter, we propose a reinforced-attention policy (RAP) model for few-shot learning, an attention mechanism trained by reinforcement learning (see Figure 3.1). Specifically, an auxiliary agent is designed to equip the backbone network for computing a series of attention maps, which recurrently decide where to enforce or ignore the feature maps in the middle of the backbone.

Our designed RAP enables the backbone network to identify informative parts of the feature maps of a sample and thus make generated embeddings more discriminative for the few-shot meta-learner. We formulate the feature extraction of samples as a Markov Decision Process (MDP) and optimize RAP in a reinforcement learning setting. Given the image samples, the agent that progressively refines the attention upon feature maps over time is optimized according to the online feedback, *i.e.*, the computed reward. The specific reward function, which incorporates the performance of the meta-learner on the held-out data, is designed to guide the agent toward being more generic.

As illustrated in Figure 3.1 (a), the vanilla few-shot learning process can be viewed as a two-component paradigm: the backbone network and the meta-learner. Given several query samples $\mathbf{X}_q = \{\mathbf{X}_{q,1}, \mathbf{X}_{q,2}, \dots, \mathbf{X}_{q,i}, \dots\}$ and support samples $\mathbf{X}_s = \{\mathbf{X}_{s,1}, \mathbf{X}_{s,2}, \dots, \mathbf{X}_{s,j}, \dots\}$ with labels $Y_s = \{Y_{s,1}, Y_{s,2}, \dots, Y_{s,j}, \dots\}$, the meta-learner works as a classifier to identify the category $\hat{Y}_q = \{\hat{Y}_{q,1}, \hat{Y}_{q,2}, \dots, \hat{Y}_{q,i}, \dots\}$ of query samples based on feature embeddings, $\mathbf{e}_q = \{\mathbf{e}_{q,1}, \mathbf{e}_{q,2}, \dots, \mathbf{e}_{q,i}, \dots\}$ and $\mathbf{e}_s = \{\mathbf{e}_{s,1}, \mathbf{e}_{s,2}, \dots, \mathbf{e}_{s,j}, \dots\}$, computed by the backbone network. How RAP works on a few-shot learning task is shown in Figure 3.1 (b). Using RAP to equip the backbone network, we convert the few-shot learning into a MDP. The policy module continuously receives the reward r_t as feedback from the meta-learner and gives the action $\mathbf{a}_t$ towards the larger total reward. The embedding $\mathbf{e}_T$ from the last time step T is viewed as the resulting embedding. Hence, instead of $\hat{Y}_q$, we take $\hat{Y}_{q,T}$ as the final prediction. The modification to only the backbone network makes RAP skip the further design of the meta-learner, such that RAP can be embedded in most existing few-shot learning baselines. The contributions of this work can be summarized as follows:

- Our proposed RAP can attend to informative regions of feature maps while avoiding the extra cumbersome meta-learner design. Additionally, most of the few-shot learning baselines can be equipped with RAP since RAP is essentially a flexible extension specific to the backbone network.
- We provide a novel solution to train the attention mechanism using reinforcement learning. Intuitively, the recurrent formulation in a reinforcement learning manner can help the attention mechanism to incrementally locate useful parts of the features due to the characteristic that reinforcement learning can learn from experience substantially.

In the experimental part of our few-shot image classification, we select several baselines for which RAP agents are trained. In effect, our embedded design pushes the backbone network to produce embeddings that become more discriminative. Aside from few-shot image classification, our design applies to image classification. The effectiveness is demonstrated via experiments on multiple benchmark datasets.

3.2 Related Work

Attention Design. The attention mechanism, aimed at attending to the discriminative areas adaptively, has been studied extensively for visual tasks Hu et al. [2018a]; Woo et al. [2018]; Wang et al. [2017b]; Mnih et al. [2014]; Zoran et al. [2020]. Those attention blocks either focus on the channel encoding Hu et al. [2018a] or spatial context connection Wang et al. [2017b]. Hu *et al.* first propose the Squeeze-and-Excitation Network (SENet) to weigh each slice of the feature map Hu et al. [2018a]. Later work, termed Convolutional Block Attention Module (CBAM) Woo et al. [2018], further employs hybrid spatial and channel features for attention design. In Wang et al. [2017b], the spatial context connection is established by the visual similarity

between the query features and the key. Unlike these works, our attention design selects the informative areas by recurrently attending to the feature maps in a reinforcement learning manner. The work in Mnih et al. [2014]; Zoran et al. [2020] also uses the recurrent model to learn an attention mask. In our work, RAP is expected to boost its generalization power by the learning experience on the held-out data.

Attention in Few-shot Learning. There are two main categories of attention mechanisms applied in few-shot learning. The first one is semantics-guided attention Wang et al. [2017a]; Chu and Wang [2018]; Yan et al. [2019]. Guided by word embeddings, Multi-attention Network Wang et al. [2017a] generates attention maps over the visual features of samples; hence, the representative parts of a sample can be precisely captured. Like a Multi-attention Network, an attention generator is developed in Chu and Wang [2018] to localize the relevant regions in an image with the help of word embeddings. Yan *et al.* Yan et al. [2019] develop a dual attention network (STANet) trained with a semantic-aware loss. The other category is sample-guided attention Vinyals et al. [2016]; Ren et al. [2019b]; Hou et al. [2019]. Matching Network (MatchingNet) Vinyals et al. [2016] uses attention based on the softmax function to fully specify the prediction of the meta-learner classifier. Similar to the Matching Network, the Cross-Attention Network proposed by Hou *et al.* Hou et al. [2019] aims at modeling the semantic dependency between support samples and query samples. The relevant regions on the query samples are adaptively localized such that the discrimination of feature embeddings benefits. MatchingNet provides attention to embeddings, while Cross-Attention manipulates feature maps. Under incremental few-shot learning, an attention attractor model Ren et al. [2019b] is presented to regulate the meta-learning of unknown classes by attending to known classes. In contrast to these works, our developed RAP attempts to sequentially capture the important information within the feature maps of the backbone network. Its simple extension to the backbone network avoids the complicated meta-learner structure design. In addition, the external semantic data resource is not needed.

Reinforcement Learning on Visual Tasks. Deep learning is widely investigated in the domain of reinforcement learning. By introducing the deep neural network to build the value function or the policy function, the reinforcement learning agent has a powerful capacity to learn the dynamics of the environment with which it interacts. Deep Q Network (DQN) Mnih et al. [2015] and Deep Deterministic Policy Gradient (DDPG) Lillicrap et al. [2016] are the very first attempts to apply the deep neural network in a reinforcement learning setting. Several works have introduced deep reinforcement learning to the computer vision community. Reinforcement learning is easily utilized in sequential visual tasks, like tracking Yun et al. [2017]; Chen et al. [2018a]; Ren et al. [2018], action recognition Yeung et al. [2016]; Tang et al. [2018]; Chen et al. [2018b], visual navigation Wang et al. [2019a]; Zhu et al. [2019], video summarization Li et al. [2018] and person re-id Lan et al. [2017]. The applications are also studied on more classic tasks, such as classification Xu et al. [2019], object detection Caicedo and Lazebnik [2015]; Mathe et al. [2016], segmentation Araslanov et al. [2019]; Zhou et al. [2019] and pose estimation Krull et al. [2017]; Shao et al. [2020]. However, few works employing reinforcement learning have been done in

the domain of few-shot learning.

3.3 Methodology

The overviews of relevant frameworks are depicted in Figure 3.2. As shown in Figure 3.2 (b), the baseline model (see Figure 3.2 (a)) is equipped with a policy module that receives recurrent signals and reinforces the backbone network to attend to the discriminative areas within the feature maps. Conditioned on an image sample, the RAP model sequentially seeks the scoring tensor for the corresponding feature maps generated by the backbone network. During this process, we leverage reinforcement learning to train a reward-directed agent, *i.e.*, the policy module (see Figure 3.3).

3.3.1 Problem Statement

As described in Figure 3.2 (b), one execution of RAP can be understood as a MDP. RAP agent, *i.e.*, the policy module interacts with the backbone network via multiple time steps. Its behavior can be represented as state-action pairs. At time step t , the agent observes the state $\mathbf{s}_t$, executes the action $\mathbf{a}_t$, and receives the reward r_t towards optimizing the policy. The action at current time step $\mathbf{a}_t$ only depends on the current state $\mathbf{s}_t$ and the next state $\mathbf{s}_{t+1}$ is conditioned on both $\mathbf{s}_t$ and $\mathbf{a}_t$.

Specifically, given feature maps from the backbone network, our RAP module will recurrently attend to informative areas within feature maps, as the RAP agent will determine attention maps until more informative areas in feature maps are found. In our setting, the action $\mathbf{a}_t$ equals the attention maps. The varying factors, such as the input image $\mathbf{I}_0$ and the computed embeddings $\mathbf{e}_t$, are modeled as states $\mathbf{s}_t$ in RL. In the following, we will introduce more details about state, action, and reward.

State. Under reinforcement learning settings, we normally encode the observed environment dynamics as states. As depicted in Figure 3.2 (b), the dataset and backbone network are modeled as the environment to the agent. The image itself $\mathbf{I}_0 \in \mathbb{R}^{H \times W \times 3}$ and the feature embedding $\mathbf{e}_t \in \mathbb{R}^{N_{SE} \times 1}$ generated from the last convolutional block are aggregated as the state $\mathbf{s}_t = \{\mathbf{s}^I = \mathbf{I}_0, \mathbf{s}^E = \mathbf{e}_t\}$ at time step $t \in \{0, 1, 2, \dots, T\}$. T is the total execution times for each image. It is noted that $\mathbf{s}^I$ is fixed in one processing sequence. We need to address the issue that the high variance exists among tremendous images Mnih et al. [2014]; Mao et al. [2018]. For this purpose, we treat the image $\mathbf{I}_0$ as one of the states $\mathbf{s}^I$. Hence, RAP is capable of learning the variance among input images. In addition, the setting of the state enables the agent to fuse both the low-level information and the high-level information. The high-level information from $\mathbf{s}^E$ indicates how the action affects the backbone network. The low-level information encoded within $\mathbf{s}^I$ reflects the image variance.

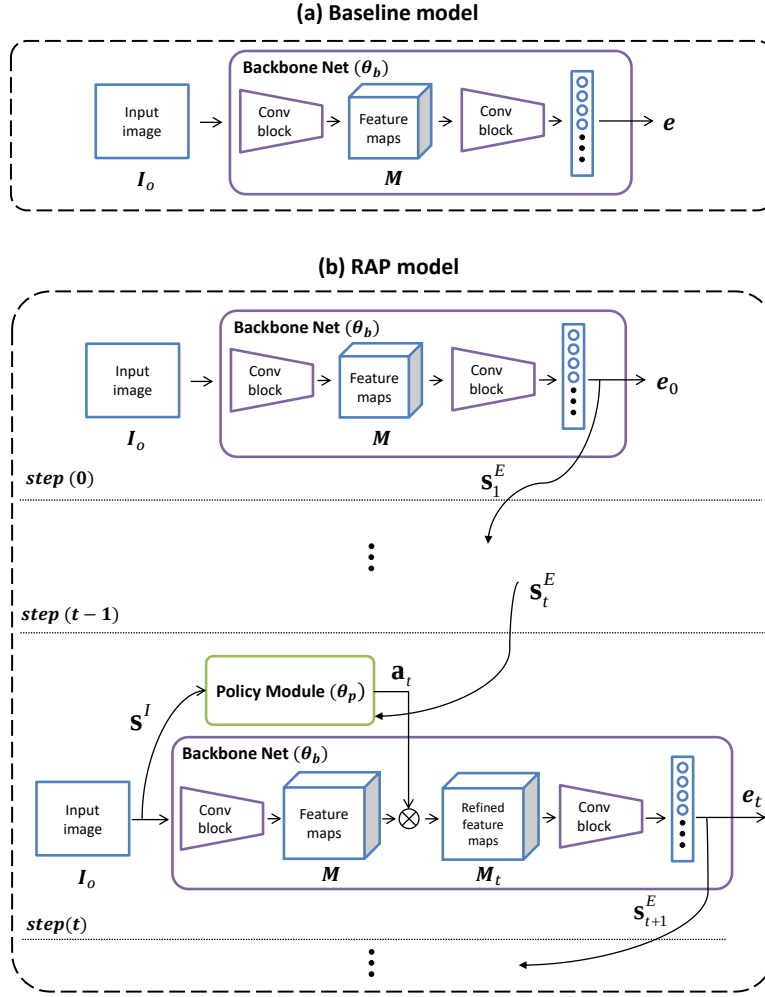


Figure 3.2: The frameworks of baseline model and RAP model. (a) Baseline model. Given the input image I_o , the backbone network generates the embedding $\mathbf{e}$. (b) RAP model. The recurrent formulation is built where each input image is processed over multiple time steps. The designed policy module is combined with the backbone network to receive the state $\mathbf{s}_t$ and then compute the action $\mathbf{a}_t$. The embedding computed after the final time step $\mathbf{e}_T$ is the resulting embedding.

Action. Attention computed at time t for weighting each pixel on the feature map is defined as the action $\mathbf{a}_t \in \mathbb{R}^{h \times w \times c}$ where h , w and c are its height, width, and length. As illustrated in Figure 3.3, the action before reshaping operation $\mathbf{a}_t^v \in \mathbb{R}^{hw \times 1}$ is drawn from the distribution π : $\mathbf{a}_t^v \sim \pi(\cdot | g(\mathbf{s}_t | \theta_p))$ where π is the distribution function parameterized by the computation from policy function g . Then, $\mathbf{a}_t^v \in \mathbb{R}^{hw \times 1}$ is reshaped and duplicated to $\mathbf{a}_t \in \mathbb{R}^{h \times w \times c}$ which has the same size as the operated feature map. Then, as shown in Figure 3.2 (b), the element-wise multiplication is

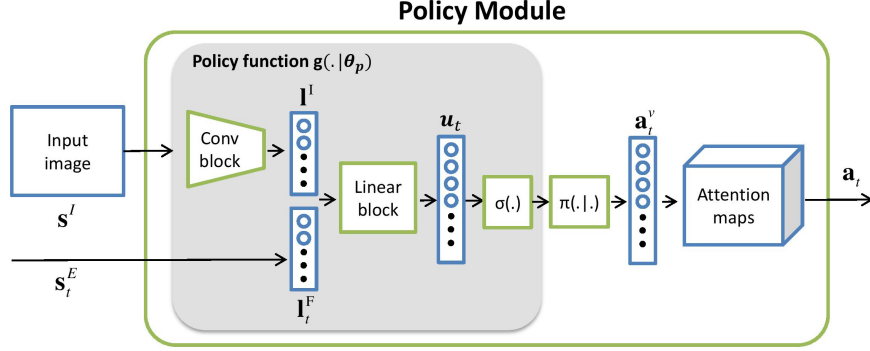


Figure 3.3: The framework of the policy module. The module maps the state $\mathbf{s}_t$ to the action $\mathbf{a}_t$. $\sigma(\cdot)$ is Sigmoid function.

performed to give the refined feature map as follows:

$$\mathbf{M}_t = \mathbf{a}_t \otimes \mathbf{M}. \quad (3.1)$$

where $\mathbf{M} \in \mathbb{R}^{h \times w \times c}$ ($\mathbf{M}_t \in \mathbb{R}^{h \times w \times c}$) is the feature map before (after) the action being taken at time t . Given the state $\mathbf{s}_t$, the policy function g can decide which pixel along the channel in the feature map should be heightened or weakened at time t .

Reward. After the execution of the action $\mathbf{a}_t$ over feature maps at step t , the agent receives the reward r_t which evaluates $\mathbf{a}_t$. The reward function matters in the sense that it criticizes the direction that the policy is optimized towards. One goal of introducing RAP is to adaptively allocate the informative areas over feature maps via a sequential process. In doing so, one option is to closely monitor and react to the model performance on the held-out data. This can be achieved with the help of the validation data. Hence, the reward function is proposed as

$$\mathbf{r}_t = -\alpha \mathcal{L}_{val,t} \quad (3.2)$$

where $\mathcal{L}_{val,t}$ is the loss built on the validation data at time t and α is the coefficient. In our case, the reward design aims to direct the policy to achieve a higher prediction on the validation data. By explicitly pursuing better performance on the validation data, RAP can have a better generalization ability and, therefore, pay attention to more useful information on feature maps.

3.3.2 Policy Design

In this case, we exploit reinforcement learning to learn the policy. The architecture of the policy module is illustrated in Figure 3.3. Let $g(\mathbf{s}_t|\theta_p)$ be the policy function which outputs the vector $\mathbf{u}_t \in \mathbb{R}^{hw \times 1}$, representing the parameter value of the distribution function $\pi(\cdot|g(\cdot|\theta_p))$. Under this distribution, $\mathbf{a}_t^v \in \mathbb{R}^{hw \times 1}$ can be chosen.

As shown in Figure 3.3, given the image $\mathbf{s}^I$, the conv block, which contains standard conv operations followed by the global average pooling along each channel, is

used to compute a N_{I1} -dimensional representation $\mathbf{l}^I \in \mathbb{R}^{N_{I1} \times 1}$. The resulting vector $\mathbf{l}^I$ is then concatenated with another state vector $\mathbf{l}_t^E = \mathbf{s}_t^E \in \mathbb{R}^{N_{IE} \times 1}$ passed from the last time step to form the vector $\mathbf{l}_t = \{\mathbf{l}^I, \mathbf{l}_t^E\}$ which encodes the low-level spatial information from the given image sample and the high-level semantic information from the backbone network. Taking $\mathbf{l}_t$ as input, the linear block follows to calculate $\mathbf{u}_t$. The configuration of the linear block is constructed with fully connected (FC) layers. The computed action vector $\mathbf{a}_t^v$ should be reshaped and extended c times to $\mathbf{a}_t \in \mathbb{R}^{h \times w \times c}$ where c indicates the channel number of the feature map to be operated on.

3.3.3 Policy Training

The learnable parameter $\theta = \{\theta_b, \theta_p\}$ contains the parameter of the backbone network θ_b and the policy module θ_p . The objective of policy training is to maximize the sum of the reward $J^\pi = \sum_t^T r_t$ under the distribution π given the input image $\mathbf{s}^I$. We directly write the reinforcement loss $\mathcal{L}_{rein}$ as follows:

$$\begin{aligned} \mathcal{L}_{rein} &= -\frac{1}{NT} \sum_{i=1}^N J_i^\pi \\ &= -\frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T \log(\pi(\mathbf{u}_{i,t} | \mathbf{s}_{i,t}, \theta_p)) r_{i,t} \end{aligned} \quad (3.3)$$

where $\mathbf{u}_t = g(\mathbf{s}_t | \theta_p)$ on the current state $\mathbf{s}_t$ is the predicted parameter (e.g., mean) to the distribution function π in choosing $\mathbf{a}_t^v$ and $\log \pi$ is the log-probability. N is the batch size of images that the model processes in each time step. Using the REINFORCE algorithm Williams [1992], the gradient can be expressed as:

$$\nabla_\theta J^\pi = \sum_{t=1}^T \nabla_\theta \log(\pi(\mathbf{u}_t | \mathbf{s}_t; \theta_p)) r_t, \quad (3.4)$$

The gradient $\nabla_\theta J^\pi$ in Equation (3.4) encourages the optimal parameter to produce a larger reward. The parameter θ_p determines the sensitivity of the policy to the reward r_t . Its motivation is to seek a suitable scale acting on the feedback from the validation. Suppose $r_{i,t}$ becomes zero, the gradient $\nabla_\theta J^\pi$ likewise becomes zero, which would not push the weight to be optimized. One possible concern is that only a reinforcement learning loss $\mathcal{L}_{rein}$ that does not include the training data would make the model more firmly rooted in the validation data and thus negatively degrade the accuracy. Incorporating the loss based on the train set can explicitly avoid this problem. It is, therefore, worth considering a composite form of the total loss as:

$$\mathcal{L}_{total} = \mathcal{L}_{rein} + \mathcal{L}_{train} \quad (3.5)$$

where $\mathcal{L}_{train}$ is the loss built on the training data. When the training is under progress with $\mathcal{L}_{total}$, θ is free to be updated. The training loss $\mathcal{L}_{train}$ based on the training data

guarantees the basic performance of the backbone network, whereas the reinforce loss $\mathcal{L}_{rein}$ based on the validation data enables the policy module to actively identify the informative areas on the feature map of the backbone. Regarding the inference stage, θ becomes frozen.

3.4 Experiments

The proposed design is mainly evaluated on the task of few-shot image classification. We assess our RAP model over the *miniImageNet* Ravi and Larochelle [2017] and Caltech-UCSD Birds (CUB-200-2011) Wah et al. [2011] datasets where it consistently outperforms the corresponding baseline model. Then, to verify that the feasibility of our design is not restricted to few-shot learning alone, experiments on image classification are performed on CIFAR-10/100 and STL-10. The standard data augmentations, *i.e.*, random crop and random horizontal flips, are applied to the training samples throughout all experiments. The Adam optimizer Kingma and Ba [2014] is employed to optimize the model weight. Moreover, the trained model that achieves the best validation accuracy is selected for the inference. All deep models are implemented using PyTorch Paszke et al. [2019]. We opt for the Gaussian function as a distribution function π whose mean value is computed by policy function g .

3.4.1 Datasets

***miniImageNet*.** For few-shot image classification, *miniImageNet* Ravi and Larochelle [2017] is the most popular benchmark (60000 images of size 84×84 selected from ImageNet), which is comprised of 64 training (or known) classes, 16 validation classes, and 20 test (or unknown) classes of images. Every class has 600 images. The size ratio of the train set and validation set is 4:1.

CUB-200-2011. The CUB-200-2011 dataset Wah et al. [2011] consists of 11788 images of 84×84 from 200 classes. Following the data preparation from Hilliard et al. [2018], we randomly divide it into 100 train (or known) classes, 50 validation classes, and 50 test (or unknown) classes.

CIFAR-10/100. CIFAR-10/100 Krizhevsky et al. [2009] comprises 60000 images from 10 classes/100 classes. The size of each image is 32×32 . There are originally 50000 train images and 10000 test images. Our method needs a validation set. Thus, we randomly divide the original train set into 40000 train samples and 10000 validation samples.

STL-10. STL-10 Coates et al. [2011] contains 10 classes where each class has 500 train images and 800 test images of 96×96 . Like CIFAR, the original training data is proportionately split into train and validation data by a ratio 4:1.

Table 3.1: Few-shot image classification accuracy on *miniImageNet* Ravi and Larochelle [2017]. Published results of MAML and ProtoNet with Conv4 are provided in Finn et al. [2017] and Snell et al. [2017], respectively, while their outcomes under Conv-6 and ResNet-10 are reported in Chen et al. [2019a]. "*" indicates the result is obtained using networks implemented by us.

Model	Backbone	5-way Acc.	
		1-shot	5-shot
MAML Finn et al. [2017]	Conv-4	48.70±1.84	63.15±0.92
	Conv-6	50.96±0.92	66.09±0.71
	ResNet-10	54.69±0.89	66.62±0.83
RAP-MAML	Conv-6	52.57±0.61	66.96±0.52
	ResNet-10	56.13±0.62	68.74±0.54
ProtoNet Snell et al. [2017]	Conv-4	49.42±0.78	68.20±0.66
	Conv-6	50.37±0.83	67.33±0.67
	ResNet-10	51.98±0.84	72.64±0.64
RAP-ProtoNet	Conv-6	51.72±0.72	69.18±0.45
	ResNet-10	53.64±0.60	74.54±0.45
TPN Liu et al. [2018b]	ResNet-12	59.46	75.65
LTS Li et al. [2019]	ResNet-12	70.1±1.9	78.7±0.8
MetaOptNet-SVM Lee et al. [2019]	ResNet-12	64.09±0.62	80.00±0.45
Neg-Margin Liu et al. [2020a]	ResNet-12	63.85±0.81	81.57±0.56
DSN-MR Simon et al. [2020]	ResNet-12	67.09±0.68	81.65±0.69
Su <i>et al.</i> Su et al. [2019]	ResNet-18	-	76.60±0.70
Hyperbolic ProtoNet Khurlov et al. [2020]	ResNet-18	59.47±0.20	76.84±0.14
LwoF Gidaris and Komodakis [2018]	WRN	60.06±0.14	76.39±0.11
wDAE-GNN Gidaris and Komodakis [2019]	WRN	61.07±0.15	76.75±0.11
EPNet Rodríguez et al. [2020]	WRN	70.74±0.85	84.34±0.53
LaplacianShot Ziko et al. [2020]	ResNet-10*	69.47±0.20	79.78±0.15
	ResNet-12*	72.29±0.20	82.85±0.14
	DenseNet-121	75.57±0.19	84.72±0.13
RAP-LaplacianShot	ResNet-10	71.34±0.19	81.98±0.14
	ResNet-12	74.29±0.20	84.51±0.13
	DenseNet-121	75.58±0.20	85.63±0.13

3.4.2 Few-shot Image Classification

Training a few-shot learning machine is characterized as episode training. An episode, which is randomly sampled from training data, is a simple N -way K -shot Q -query classification task. The training process is to classify $Q \times N$ query samples from N classes correctly given K samples per class (*e.g.*, $K = 1$ or 5).

The 1-shot and 5-shot scenarios are examined. Our entire set of experiments is performed using the settings of 5-way and 16-query. Moreover, we arrange 1200 episodes at random for MAML Finn et al. [2017] and ProtoNet Snell et al. [2017] (10000 and 600 episodes for LaplacianShot Ziko et al. [2020] and Neg-Margin Liu et al. [2020a], respectively). The final performance is provided, averaging the classification accuracy over episodes. For the design details of the policy module, the conv

block includes three convolutional layers with 3×3 convolutions, where each layer is followed by batch normalization, ReLU, and max pooling. One-layer FC is followed as a linear block. The total time step T is kept fixed as 5. The coefficient α in Equation (3.2) is set to $1e-4$. Different architectures of backbone network are assessed including Conv-4 Vinyals et al. [2016], Conv6, ResNet-10, ResNet-12, ResNet-18 He et al. [2016] and DenseNet-121 Huang et al. [2017]. We insert our attention module after the second (or third) conv block of Conv-4 and Conv-6 (or ResNet-10, ResNet-12, ResNet-18, and DenseNet-121).

3.4.2.1 *miniImageNet*

In the first part of our few-shot image classification experiment, we compare the performance on *miniImageNet* Ravi and Larochelle [2017]. The results are presented in Table 3.1. We equip the classic baselines, including MAML Finn et al. [2017] and ProtoNet Snell et al. [2017], with our proposed RAP. LaplacianShot Ziko et al. [2020], an excellent baseline, is also considered. In most cases, our RAP models outperform their corresponding baseline counterparts except for RAP-LaplacianShot with DenseNet-121 in the 5-way 1-shot setting. Another observation is that the choice of baseline affects performance improvement against baselines. RAP-MAML and RAP-ProtoNet yield a greater performance improvement than RAP-LaplacianShot. Comparisons with other few-shot learning baselines that exploit attention are provided in Table 3.2. As we can see from the table, our performance is better than others, given the same backbone at the time of publication of our work.

Table 3.2: Comparison with other attention designs applied to few-shot image classification on *miniImageNet* Ravi and Larochelle [2017]. Different attention models are compared under the same backbone.

Model	5-way Acc.		
	Backbone	1-shot	5-shot
Attention Attractor Ren et al. [2019b]	ResNet-10	54.95 ± 0.30	63.04 ± 0.30
RAP-MAML	ResNet-10	56.13 ± 0.62	68.74 ± 0.54
RAP-ProtoNet	ResNet-10	53.64 ± 0.60	74.54 ± 0.45
RAP-LaplacianShot	ResNet-10	71.34 ± 0.19	81.98 ± 0.14
STANet Yan et al. [2019]	ResNet-12	58.35 ± 0.57	71.07 ± 0.39
Cross-Attention Hou et al. [2019]	ResNet-12	67.19 ± 0.55	80.64 ± 0.35
RAP-LaplacianShot	ResNet-12	74.29 ± 0.20	84.51 ± 0.13

3.4.2.2 CUB-200-2011

Below, experiments are performed on CUB-200-2011 Wah et al. [2011]. In this case, besides MAML, ProtoNet, and LaplacianShot, we test another strong baseline, Neg-Margin Liu et al. [2020a]. The obtained accuracy is reported in Table 3.3. It is obvious that using MAML, RAP models have an accuracy increase of around 2% to 6% against the baseline models regardless of the backbone. Similarly, it is also the case

Table 3.3: Few-shot image classification accuracy on CUB-200-2011 Wah et al. [2011]. The results of MAML and ProtoNet are provided from Chen et al. [2019a].

Model	5-way Acc.		
	Backbone	1-shot	5-shot
MAML Finn et al. [2017]	Conv-4	54.73 $\pm$ 0.97	75.75 $\pm$ 0.76
	Conv-6	66.26 $\pm$ 1.05	78.82 $\pm$ 0.70
	ResNet-10	70.32 $\pm$ 0.99	80.93 $\pm$ 0.71
	ResNet-18	68.42 $\pm$ 1.07	83.47 $\pm$ 0.62
RAP-MAML	Conv-4	61.49 $\pm$ 0.70	77.15 $\pm$ 0.50
	Conv-6	69.95 $\pm$ 0.68	81.48 $\pm$ 0.44
	ResNet-10	74.33 $\pm$ 0.65	83.29 $\pm$ 0.42
	ResNet-18	75.04$\pm$0.70	86.07$\pm$0.43
ProtoNet Snell et al. [2017]	Conv-4	50.46 $\pm$ 0.88	76.39 $\pm$ 0.64
	Conv-6	66.36 $\pm$ 1.00	82.03 $\pm$ 0.59
	ResNet-10	73.22 $\pm$ 0.92	85.01 $\pm$ 0.52
	ResNet-18	72.99 $\pm$ 0.88	86.64 $\pm$ 0.51
RAP-ProtoNet	Conv-4	56.71 $\pm$ 0.66	78.70 $\pm$ 0.44
	Conv-6	67.79 $\pm$ 0.66	83.78 $\pm$ 0.41
	ResNet-10	75.17$\pm$0.63	88.29 $\pm$ 0.34
	ResNet-18	74.09 $\pm$ 0.60	89.23$\pm$0.31
DeepEMD Zhang et al. [2020a]	ResNet-12	76.65 $\pm$ 0.83	88.69 $\pm$ 0.50
MatchingNet Vinyals et al. [2016]	ResNet-18	72.36 $\pm$ 0.90	83.64 $\pm$ 0.60
RelationNet Sung et al. [2018]	ResNet-18	67.59 $\pm$ 1.02	82.75 $\pm$ 0.58
Chen <i>et al.</i> Chen et al. [2019a]	ResNet-18	67.02	83.58
SimpleShot Wang et al. [2019b]	ResNet-18	70.28	86.37
Manifold Mangla et al. [2020]	WRN	80.68 $\pm$ 0.81	90.85 $\pm$ 0.44
EPNet Rodríguez et al. [2020]	WRN	87.75 $\pm$ 0.70	94.03 $\pm$ 0.33
Neg-Margin Liu et al. [2020a]	ResNet-18	72.66 $\pm$ 0.85	89.40 $\pm$ 0.43
RAP-Neg-Margin	ResNet-18	75.37$\pm$0.81	90.61$\pm$0.39
LaplacianShot Ziko et al. [2020]	ResNet-18	80.96	88.68
RAP-LaplacianShot	ResNet-18	83.59$\pm$0.18	90.77$\pm$0.10

for ProtoNet, LaplacianShot, and Neg-Margin, that the performance of RAP models is significantly superior to the corresponding baselines. The competitive outcomes prove that the learned policy by reinforcement learning helps the backbone network to localize the informative areas of the feature map and thus generate discriminative embeddings. It should also be emphasized, again, that our RAP is compatible with several various few-shot learning baselines.

3.4.3 Image Classification

For image classification, all models are trained with a mini-batch size of 128 for 4000 epochs (2000 epochs on STL-10). Pre-trained backbones are used. The learning rate is set to 1e-6. The RAP models execute actions for $T=5$ times. The results of the classification accuracy on three datasets are listed in Table 3.4, where RAP models surpass the corresponding baseline models with a healthy margin. We can

Table 3.4: Image classification accuracy of all models on CIFAR-10/CIFAR-100 and STL-10 Coates et al. [2011]. LeNet LeCun et al. [1998], VGG-16 Simonyan and Zisserman [2014], ResNet-18, ResNet34 and ResNet-50 are chosen as backbones. Baseline models are trained on both the train set and the validation set. "*" indicates the result is obtained using networks implemented by us.

Dataset	Model	Backbone	Accuracy (%)
CIFAR-10	Baseline	LeNet*	75.94
		VGG-16*	91.48
		ResNet-18*	92.68
		ResNet-50*	93.02
	RAP	LeNet	77.15 (↑ 1.21)
		VGG-16	92.05 (↑ 0.57)
		ResNet-18	93.25 (↑ 0.57)
		ResNet-50	93.56 (↑ 0.54)
CIFAR-100	Baseline	LeNet*	41.63
		VGG-16*	67.23
		ResNet-18*	72.56
		ResNet-50*	74.22
	RAP	LeNet	43.02 (↑ 1.39)
		VGG16	68.45 (↑ 1.22)
		ResNet-18	73.76 (↑ 1.20)
		ResNet-50	75.36 (↑ 1.14)
STL-10	Baseline	VGG-16*	74.43
		ResNet-18*	78.21
		ResNet-34*	76.31
	RAP	VGG-16	76.16 (↑ 1.73)
		ResNet-18	80.15 (↑ 1.94)
		ResNet-34	77.74 (↑ 1.43)

observe that on each dataset, RAP models have a similar accuracy improvement given different backbone architectures.

3.4.4 Ablation Study

Validation Set. RAP models trained by $\mathcal{L}_{total} = \mathcal{L}_{train} + \mathcal{L}_{rein}$ work under the specific setting that $\mathcal{L}_{train}$ and $\mathcal{L}_{rein}$ are built on training data and validation data, respectively. Any batch is formed by images from either the training set or the validation set. The reward r_t of RAP incorporates the feedback from the validation loss $\mathcal{L}_{val}$. Hence, to justify the comparison between RAP models and baseline models, it is necessary to train the baseline models by using both the train set and the validation set. Thus, the loss $\mathcal{L}_{total} = \mathcal{L}_{train}$ to the baseline model is replaced by $\mathcal{L}_{total} = \mathcal{L}_{train} + \mathcal{L}_{val}$. As reported in Table 3.5, even under the same data settings, RAP models still beat baseline models with different backbones. Another fact is that RAP models are sensitive to α . $\mathcal{L}_{rein}$ does contribute even if α is set to 1e-4. Thus, we choose α carefully since α decides what extent validation data should be leveraged (see Table 3.6).

Table 3.5: Fair comparisons on CUB-200-2011 Wah et al. [2011] where baseline models are trained on both the train and validation sets like RAP models. "*" indicates the result is obtained using networks implemented by us.

Model	Backbone	5-way Acc.	
		1-shot	5-shot
MAML Finn et al. [2017]	Conv-4*	57.55 $\pm$ 0.67	75.61 $\pm$ 0.49
	Conv-6*	65.07 $\pm$ 0.72	79.89 $\pm$ 0.48
RAP-MAML	Conv-4	61.49 $\pm$ 0.70	77.15 $\pm$ 0.50
	Conv-6	69.95 $\pm$ 0.68	81.48 $\pm$ 0.44
ProtoNet Snell et al. [2017]	Conv-4*	51.61 $\pm$ 0.65	75.29 $\pm$ 0.48
	Conv-6*	64.72 $\pm$ 0.72	81.35 $\pm$ 0.43
RAP-ProtoNet	Conv-4	56.71 $\pm$ 0.66	78.70 $\pm$ 0.44
	Conv-6	67.79 $\pm$ 0.66	83.78 $\pm$ 0.41

Table 3.6: The results of 5-way classification of RAP-ProtoNet on CUB-200-2011 Wah et al. [2011] using different α .

Model	α	Backbone	5-way Acc.	
			1-shot	5-shot
RAP-ProtoNet	0	Conv-4	53.83 $\pm$ 0.66	77.10 $\pm$ 0.46
	1e-4	Conv-4	56.71 $\pm$ 0.66	78.70 $\pm$ 0.44
	1e-2	Conv-4	44.68 $\pm$ 0.61	71.02 $\pm$ 0.49
	1	Conv-4	44.90 $\pm$ 0.63	65.07 $\pm$ 0.50
RAP-ProtoNet	0	ResNet-10	72.71 $\pm$ 0.61	86.68 $\pm$ 0.33
	1e-4	ResNet-10	75.17 $\pm$ 0.63	88.29 $\pm$ 0.34
	1e-2	ResNet-10	72.61 $\pm$ 0.64	85.95 $\pm$ 0.37
	1	ResNet-10	66.33 $\pm$ 0.64	80.47 $\pm$ 0.42

Attention Design. To further verify the advantage of our method, we compare RAP against two conventional attention designs, SENet Hu et al. [2018a] and CBAM Woo et al. [2018]. Different attention mechanisms substitute the policy module of RAP, but the remaining settings are kept the same. As can be seen in Table 3.7, models equipped with SENet and CBAM are shown to outperform the corresponding baseline models. However, their performances do not exceed our RAP models except that they have a nearly identical accuracy in the 5-way 1-shot classification using the Conv-4 backbone.

3.4.5 Discussion

Recurrent Process. A natural question is whether the recurrent modeling of reinforcement learning contributes to increased performance. To provide evidence that the behavior of the sequential process has a positive effect, we perform the experiments by varying the total time step T . Under the same few-shot learning scenario, a 2-step model and a 5-step model are compared. We can see from Table 3.8 that the

Table 3.7: Few-shot image classification accuracy using ProtoNet as a baseline with different regular attention designs on CUB-200-2011 Wah et al. [2011]. "*" indicates the result is obtained using networks implemented by us.

	5-way Acc.		
	Backbone	1-shot	5-shot
SENet Hu et al. [2018a]	Conv-4*	55.18±0.61	77.07±0.46
	ResNet-10*	73.89±0.57	87.32±0.38
CBAM Woo et al. [2018]	Conv-4*	55.52±0.55	76.80±0.34
	ResNet-10*	72.43±0.65	86.70±0.48
RAP	Conv-4	56.71±0.66	78.70±0.44
	ResNet-10	75.17±0.63	88.29±0.34

Table 3.8: Comparisons among RAP models with different time step T on CUB-200-2011 Wah et al. [2011].

Model	T	5-way Acc.		
		Backbone	1-shot	5-shot
RAP-MAML	2	Conv-4	60.15±0.71	75.76±0.50
	5	Conv-4	61.49±0.70	77.15±0.50
	2	Conv-6	68.33±0.70	80.43±0.45
	5	Conv-6	69.95±0.68	81.48±0.44
RAP-ProtoNet	2	Conv-4	54.01±0.65	78.12±0.45
	5	Conv-4	56.71±0.66	78.70±0.44
	2	Conv-6	66.27±0.68	82.58±0.42
	5	Conv-6	67.79±0.66	83.78±0.41

performance has increased nearly 1% to 2% with the greater number of time steps. The results suggest that our reinforced attention continuously adapts to important information in the feature maps over time. The fact that classification accuracy gains as more time steps of actions are executed reinforces the contribution and improvement attained by the learned policy. For image classification, we take LeNet on CIFAR-10 and VGG-16 on STL-10 as instantiations. The curves of the test accuracy are plotted in Figure 3.4. Interestingly, the accuracy of the 2-step RAP model increases as quickly as the 5-step model in the initial epochs, but the 5-step model gradually surpasses it. In our setting, the reward is closely linked to the classification accuracy. Figure 3.4 also suggests that the learned policy maximizes the reward. The experiments strongly indicate that reinforcement learning can effectively learn from experience in this context and that it benefits from a greater number of iterations.

3.5 Summary

In this chapter, we propose RAP, an attention design that can fit nicely into most few-shot learning baselines to refine the feature map of the backbone towards a more

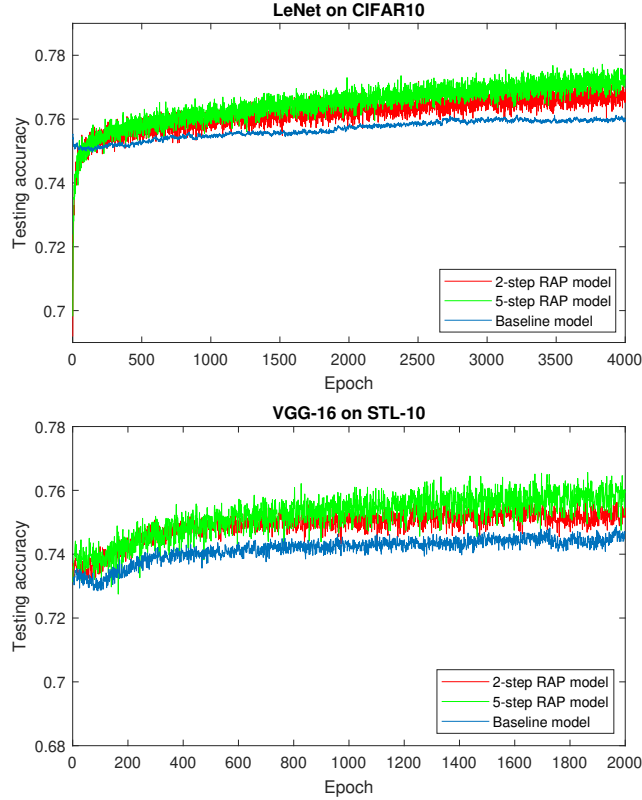


Figure 3.4: The recorded accuracy of image classification of LeNet on CIFAR-10 and VGG-16 on STL-10.

discriminative representation. At the time of publication of this work, our approach was the first attempt to address few-shot learning by applying an attention mechanism (or agent) trained by reinforcement learning. We show that the reward design based on validation data enables the model to learn a more general distribution. The key idea underlying our method is that the recurrent formulation of reinforcement learning has the nature of locating the available information from experience over time, which enables the agent to make more accurate decisions.

In the upcoming chapter, we continue to delve into low-shot learning. While low-shot learning on images has received extensive attention, the territory of audio-visual data in this context remains relatively unexplored. We focus specifically on the audio-visual zero-shot learning problem, incorporating the approach of hyperbolic geometry, one of curved geometries. Our objective is to leverage the inherent property of hyperbolic constraints, which is to encode hierarchical structures within data for generating more discriminative features.

Curved Geometry for Audio-visual Zero-shot Learning

In the previous chapter, we focus on a specific aspect of low-shot problems, namely, few-shot learning. Building upon that foundation, in this chapter, we continue working on low-shot learning, tackling a more challenging sub-problem of low-shot problems: audio-visual zero-shot learning.

Unlike few-shot learning introduced in the previous chapter, which processes unknown image samples, audio-visual zero-shot learning aims to classify unknown samples of corresponding audio and video sequences. The audio-visual data reveals a large degree of hyperbolicity, motivating the potential solution of using one of the curved geometries, *i.e.*, the hyperbolic geometry, to explore more complex hierarchical data structures and thus achieve more discriminative features. In this chapter, we propose an approach that employs a designed loss function incorporating cross-modality alignment between video and audio feature embeddings in the hyperbolic space. Additionally, we explore the use of multiple adaptive curvatures for hyperbolic projections. The experimental results on this challenging task demonstrate that our proposed hyperbolic approach for zero-shot learning outperformed the SOTA method at the time of publication of our work on three datasets: VGGSound-GZSL, UCF-GZSL, and ActivityNet-GZSL, achieving a harmonic mean (HM) improvement of around 3.0%, 7.0%, and 5.3%, respectively. This chapter is based on our published work Hong et al. [2023b].

4.1 Introduction

Visual and audio signals frequently co-occur. For example, movies combine visual and auditory signals, providing an immersive experience to viewers. Humans perceive multiple sensory inputs jointly and make decisions accordingly. When a vehicle approaches from behind and honks, the driver sees the vehicle in the rear-view mirror, hears the sound, and decides to give way. The integration of joint visual and audio signals benefits numerous applications. For instance, acoustic features can help localize objects that emit sound in videos Lin et al. [2021]; Zhou et al. [2021b]; Qian et al. [2020]; Ding et al. [2020]; Senocak et al. [2018]; Pu et al. [2017]. Fur-

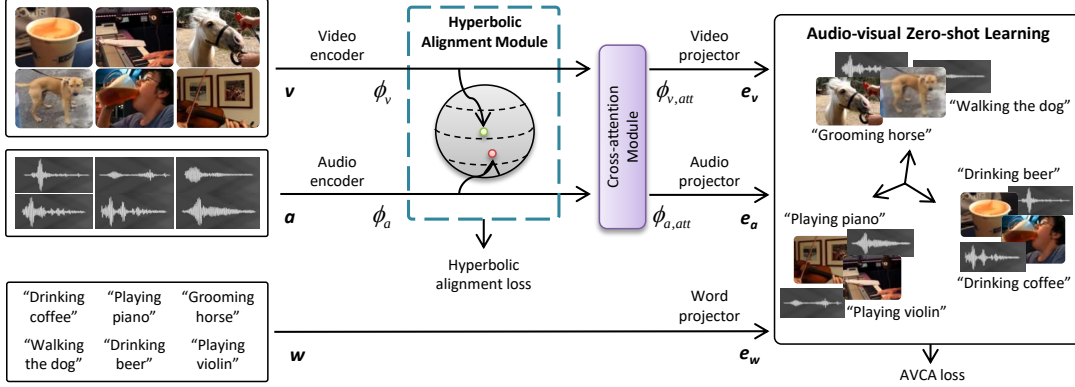


Figure 4.1: We introduce the Hyperbolic Alignment Module, represented by the block with the blue line. The input data features from each modality are encoded by two consecutive networks (encoder and projector). A cross-attention module in between is used to explore the natural correspondence between visual and audio features. Before the cross-attention module processes the features, our Hyperbolic Alignment Module computes a hyperbolic alignment loss, which aims to explore more hierarchy in audio-visual data. For example, the model may discriminate embeddings of “Playing piano” and “Walking the dog” when it finds that these embeddings belong to different superclasses: “Playing musical instruments” and “Walking/exercising/-playing with animals”.

thermore, the natural correlations between visual and audio signals provide strong supervision for learning video representations. As such, there has been growing interest in learning informative representations from audio signals for video classification Mittal et al. [2022]; Morgado et al. [2021]; Asano et al. [2020]; Afouras et al. [2020]; Cramer et al. [2019]; Korbar et al. [2018]; Owens et al. [2016]. The benefits of audio-visual multi-modality learning have also been demonstrated in other tasks, such as robotic navigation Chen et al. [2020a]; Gan et al. [2020]; Dean et al. [2020], action recognition Kazakos et al. [2019]; Gao et al. [2020], highlight detection Ye et al. [2021]; Badamdorj et al. [2021], violence detection Wu et al. [2020], aerial scene recognition Hu et al. [2020a] and speech recognition Xu et al. [2020]; Afouras et al. [2018].

Collecting vast amounts of audio-visual data needed for training deep neural networks is time-intensive, expensive, and, in some applications, impractical. Additionally, deep models may have difficulty making accurate predictions when presented with objects from unknown classes in real-world scenarios. They lack the knowledge and context to make well-informed decisions about unfamiliar objects. Low-shot audio-visual tasks Parida et al. [2020] have emerged to address the problem of insufficient data, intending to generalize models effectively from known to unknown data. These tasks include one-shot learning Wang et al. [2021b], few-shot learning Majumder et al. [2022], and audio-visual zero-shot learning Mercea et al. [2022b,a]; Mazumder et al. [2021]; Parida et al. [2020]. One-shot and few-shot audio-visual learning deals with classification from a few examples of unknown classes.

However, audio-visual zero-shot classification poses a more challenging scenario as the model cannot access audio-visual data from unknown classes during training.

In this chapter, we focus on audio-visual zero-shot learning. We investigate curvature-aware geometric learning for the audio-visual feature alignment. Our inspiration for using hyperbolic geometry, one of curved geometries, comes from the following observations:

- **Data hierarchy.** Audio-visual datasets exhibit a hierarchy. For instance, in VGGSound Chen et al. [2020b], all 309 classes can be categorized into 9 parent classes (or superclasses): “people”, “animals”, “music”, “sports”, “nature”, “vehicle”, “home”, “tools” and “others”. Similarly, ActivityNet Caba Heilbron et al. [2015] provides a rich hierarchy with at least four levels. For example, the class “Hand washing clothes” belongs to “Laundry” (the 2nd level), “Housework” (the 3rd level), and “Household Activities” (the 4th level). Most existing audio-visual works have not adequately leveraged the hierarchical structure in the audio-visual data.
- **Hyperbolic geometric properties.** Hyperbolic methods have been shown to be effective in addressing low-shot visual problems Khrulkov et al. [2020]; Liu et al. [2020b]; Fang et al. [2021]. The learned hyperbolic feature embeddings have the ability to capture the hierarchical structure within the data. This is attributed to the tree-like nature of the underlying space, as shown in Guo et al. [2022]; Chami et al. [2021]. One of the benefits of using a hyperbolic space is that it facilitates the distribution of embeddings in a tree-shaped structure since its volume expands exponentially.

Based on the observations above, we conjecture that the unique properties of hyperbolic spaces can be leveraged to capture the hierarchical structures in audio-visual data, leading to learning more discriminative embeddings for audio-visual samples. The existing audio-visual zero-shot learning methods Mercea et al. [2022b,a]; Mazumder et al. [2021]; Parida et al. [2020]; Zheng et al. [2023] operate in the non-curved Euclidean space without considering the data hierarchy. Therefore, there is a need for curvature-aware geometric solutions that can embed the data hierarchy to improve the performance of audio-visual zero-shot learning. The contributions of this work can be summarized as:

- Our work provides a new perspective on using one of the curved geometries, *i.e.* the hyperbolic geometry, for cross-modality, as shown in Figure 4.1. We propose a hyperbolic alignment loss that learns features in curved space to improve audio-visual zero-shot learning. Specifically, we use alignment between visual and audio features in the hyperbolic space as an auxiliary learning method for feature fusion across modalities. At the time of publication of our work, we were the first to apply curvature-aware geometric solutions to this task.
- Furthermore, we introduce various frameworks for using the hyperbolic embeddings: 1) Hyper-alignment, 2) Hyper-single, and 3) Hyper-multiple. The

Hyper-alignment module maps audio and visual features from Euclidean to hyperbolic space with a fixed negative curvature and compares them using intra-modal similarities. Based on Hyper-alignment, Hyper-single adapts the curvature to the model for flexible data structure exploration. Hyper-multiple generates a set of adaptive curvatures for alignments, enabling more generic embeddings.

- Extensive experiments demonstrate that, in most cases, the proposed modules outperform existing models. Moreover, using the δ_{rel} metric Khrulkov et al. [2020]; Fournier et al. [2015], we show that hyperbolic alignment enables the learned features to exhibit stronger hierarchical properties. Additionally, we observe from t-SNE Van der Maaten and Hinton [2008] visualizations that the audio-visual feature embeddings from different superclasses become more distinct.
- Ablation studies are provided further to investigate the properties of our hyperbolic alignment module. In addition to the curvature-negative hyperbolic projection approach, we also test Euclidean and Spherical approaches, which have curvature-zero and curvature-positive properties, respectively.

4.2 Related Work

4.2.1 Audio-visual Zero-shot Learning

The audio-visual zero-shot learning settings are first proposed in Parida et al. [2020], where the Coordinated Joint Multimodal Embedding (CJME) model is introduced to map video, audio, and text into the same feature space and compare them. The triplet loss was used to push video or audio features closer to their corresponding class features. Subsequently, Mazumder *et al.* Mazumder et al. [2021] develop the Audio-Visual Generalized Zero-shot Learning Network (AVGZSLNet) to address audio-visual zero-shot learning, which includes a module that reconstructs text features from visual and audio features. The Audio-Visual Cross-Attention (AVCA) framework Mercea et al. [2022b] is also designed to exchange information between video and audio representations, enabling informative representations to help achieve comparable performance in audio-visual zero-shot classification. A similar design to AVCA for handling temporal data features is presented in Mercea et al. [2022a].

4.2.2 Hyperbolic Geometry

Hyperbolic geometry has gained considerable interest in the machine learning community since its property of the negative curvature can encode the inherent hierarchical structure of the data. It benefits from a family of learning scenarios Peng et al. [2021], including sentiment analysis, recommendation systems, social network studies, *etc.* Ganea *et al.* first study the necessary to integrate hyperbolic geometry in the learning community due to its intriguing property to encode the hierarchical

structure of the data and develop some necessary neural operators in the Poincaré ball Ganea et al. [2018]. Its superior property is also leveraged to analyze and understand the irregular graph data Chami et al. [2019]. Joint efforts are made to develop hyperbolic learning algorithms in the NLP and graph neural network field, *e.g.*, hyperbolic attention network (HAN) Gulcehre et al. [2019], hyperbolic graph attention network (HGAN) Zhang et al. [2021b] *etc.* Khrulkov *et al.* analyze that the hierarchical structure also exists in the visual data and successfully integrates the hyperbolic space in the visual domain Khrulkov et al. [2020]. The concurrent work in Liu et al. [2020b] also shows that hyperbolic space is a good alternative for aligning the visual and text domains. The following works prove that hyperbolic geometry can benefit a series of visual tasks, like semantic segmentation Atigh et al. [2022], medical image recognition Yu et al. [2022], and action recognition Long et al. [2020].

In this chapter, we first learn hyperbolic embeddings for both visual and audio modalities. We then introduce a novel hyperbolic alignment loss that minimizes the differences between different modalities. This is different from existing approaches Mercea et al. [2022b,a]; Mazumder et al. [2021]; Parida et al. [2020]; Zheng et al. [2023] that focus solely on cross-modal correspondence between visual and audio streams. In contrast, our module considers cross-modal alignment in curvature-negative (or hyperbolic) space for mining hierarchical structures in audio-visual data.

4.3 Methodology

Audio-visual zero-shot learning aims to recognize audio-visual samples $\mathbf{u} = (\mathbf{v}_u, \mathbf{a}_u)$ from unknown classes during the evaluation stage Parida et al. [2020]. In this task, the training data is restricted to the known classes. In line with Mercea et al. [2022b], we adopt the generalized setting where both unknown and unknown class samples, $\mathbf{s} = (\mathbf{v}_s, \mathbf{a}_s)$ and $\mathbf{u}$, appear at the test stage. As illustrated in Figure 4.1, the corresponding semantic words $\mathbf{w}$ are provided along with the audio-visual sample $(\mathbf{v}, \mathbf{a})$ to serve as textual labels.

In this section, we first introduce the baseline AVCA Mercea et al. [2022b] and then provide details of our approach using a hyperbolic alignment loss for audio-visual zero-shot learning. As depicted in Figure 4.1, we embed the network with a Hyperbolic Alignment Module for computing the hyperbolic alignment loss. We propose three designs for this module: Hyper-alignment, Hyper-single, and Hyper-multiple, as shown in Figure 4.2 (a), (b), and (c). The subsequent subsections provide more details on each of the proposed designs.

4.3.1 Baseline

Our method is based on AVCA, which serves as the baseline. AVCA is a two-branch network that operates on synchronized visual and audio inputs, as shown in Figure 4.1. These inputs, denoted by $\mathbf{v}, \mathbf{a} \in \mathbb{R}^{512}$ respectively, are represented by feature embeddings extracted from pre-trained feature extractors, with both vectors having

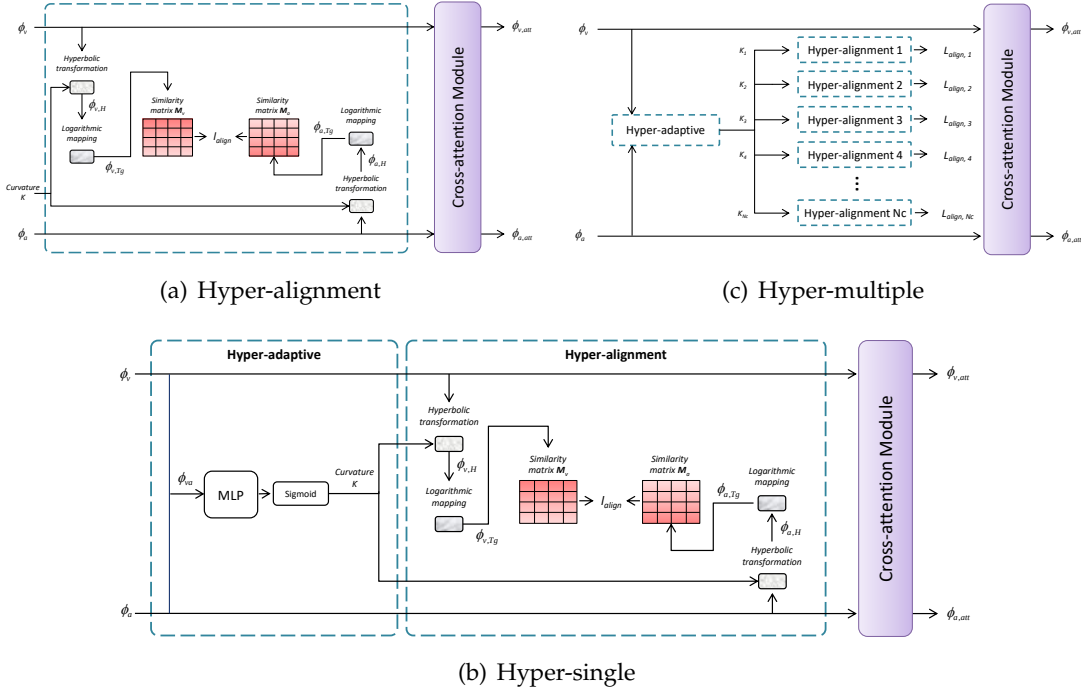


Figure 4.2: The frameworks of the proposed Hyperbolic Alignment Module: (a) Hyper-alignment, which utilizes the hyperbolic space to learn feature alignment; (b) Hyper-single, which automatically computes the curvature using the Hyper-adaptive module after concatenating visual and audio features as input, and (c) Hyper-multiple, which performs alignment using multiple learnable curvatures. While Hyper-alignment uses a fixed curvature, Hyper-single and Hyper-multiple enable flexible exploration of intrinsic data structures by adapting the curvature. Hyper-multiple is particularly suited when diverse manifold structures coexist, improving the generality of the learned representations.

a dimension of 512. In addition to the video clip feature $\mathbf{v}$ and corresponding audio feature $\mathbf{a}$, the baseline model takes the text feature $\mathbf{w} \in \mathbb{R}^{512}$ as input. The features $\phi_v, \phi_a \in \mathbb{R}^{300}$ are obtained after passing $\mathbf{v}$ and $\mathbf{a}$ through the video and audio encoder, respectively. With ϕ_v and ϕ_a , a cross-attention module calculates the cross-modality representations $\phi_{v,att}, \phi_{a,att} \in \mathbb{R}^{300}$. These features are then passed through the video and audio projectors to compute the feature embeddings $\mathbf{e}_v, \mathbf{e}_a \in \mathbb{R}^{64}$ in the common space for classifying audio-visual inputs $(\mathbf{v}, \mathbf{a})$. The text feature $\mathbf{w}$ is processed by the word projector to give $\mathbf{e}_w \in \mathbb{R}^{64}$ in the common space. The loss of AVCA is represented by $\mathcal{L}_{avca}$.

4.3.2 Hyper-alignment

We propose a minimalist design called Hyper-alignment, which learns feature alignment in a hyperbolic manifold using a fixed curvature, as illustrated in Figure 4.2 (a).

To construct the loss, both modality features ϕ_v and ϕ_a are first mapped to points $\phi_{v,H}$ and $\phi_{a,H}$ in the Poincaré ball space using Equation (2.4). Then, we obtain $\phi_{v,Tg}$ and $\phi_{a,Tg}$ in $T_z\mathbb{H}_c^{300}$ by projecting $\phi_{v,H}$ and $\phi_{a,H}$ from the hyperbolic to tangent space, using Equation (2.9).

The alignment loss is crucial in guiding the network to learn informative modal representations. Using this loss, we hope to encourage the audio-visual features ϕ_v and ϕ_a to reflect more hierarchical structures. To achieve this, we transform these features into a curvature-negative space before performing the alignment operation. Additionally, since it is unclear which modality contributes more to hierarchical exploration, our model ensures alignment between features by minimizing the differences in similarities within each modality. Motivated by the approach of Tung and Mori [2019], we design the loss using pairwise feature similarities within each mini-batch. The hyperbolic alignment loss is written as follows:

$$\mathcal{L}_{align} = \frac{1}{N^2} \|\mathbf{M}_{v,norm} - \mathbf{M}_{a,norm}\|_F^2 \quad (4.1)$$

where $\mathbf{M}_{v,norm}, \mathbf{M}_{a,norm} \in \mathbb{R}^{N \times N}$ are the normalized matrices of intra-modal similarity matrices, $\mathbf{M}_v, \mathbf{M}_a \in \mathbb{R}^{N \times N}$, for a batch of video or audio features, $\phi_{v,Tg}$ or $\phi_{a,Tg}$, in the hyperbolic tangent space. Hence, we have $\mathbf{M}_{norm} = \frac{\mathbf{M}}{\|\mathbf{M}\|_F}$. The batch size is N and $\|\cdot\|_F$ calculates the Frobenius norm.

Each element in $\mathbf{M}$ is computed via the cosine similarity between the i^{th} and j^{th} feature on the tangent space in a batch: $m_{i,j} = \cos(\phi_{Tg,i}, \phi_{Tg,j}), i, j \in \{1, 2, \dots, N\}$. Essentially, the loss $\mathcal{L}_{align}$ in Equation (4.1) computes the mean element-wise squared difference between two similarity matrices. It aligns features from two modalities by minimizing the distance between two within-modal similarities. Finally, the overall loss of Hyper-alignment is comprised of two losses:

$$\mathcal{L}_{total} = \mathcal{L}_{avca} + \mathcal{L}_{align} \quad (4.2)$$

where $\mathcal{L}_{avca}$ is the loss of the baseline. It is noted that Hyper-alignment does not bring any extra weight to the network.

4.3.3 Hyper-single

We propose Hyper-alignment to align features in a constant-curvature space. However, the representation ability of audio-visual features might suffer as the fixed curvature may not be suitable for the complex audio-visual data structure Gao et al. [2022]. To address this, we devise Hyper-single, which utilizes an adaptive curvature. Unlike Hyper-alignment, which conforms to a fixed hyperbolic structure, Hyper-single produces a learnable curvature that enables flexible exploration of intrinsic data structures.

Particularly, as illustrated in Figure 4.2 (b), the modality features, ϕ_v and ϕ_a are first concatenated into one vector $\phi_{va} \in \mathbb{R}^{600}$, and ϕ_{va} passes through one multiple-layer perceptron (MLP) to compute the curvature. Then, we have the curvature

adaptively learned as follows:

$$\kappa = \kappa_0 \cdot \text{sigmoid}(\text{MLP}(\phi_{va})) \quad (4.3)$$

where $\text{MLP}(\cdot) : \mathbb{R}^{600} \rightarrow \mathbb{R}$ is one fully-connected layer and $\kappa_0 < 0$ is the initial curvature. Hyper-single shares the same loss function in Equation (4.2) as Hyper-alignment. Compared to Hyper-alignment, Hyper-single introduces the extra weights in MLP to the network.

4.3.4 Hyper-multiple

For the design of Hyper-multiple, as depicted in Figure 4.2 (c), we first attain several curvatures. Then, multiple Hyper-alignment modules are used to learn the alignment losses via the obtained curvatures. The multiple-curvature method leads to more generic spaces since diverse manifold structures may coexist in audio-visual data Gao et al. [2021].

Hyper-multiple generates a set of adaptive curvatures $\mathbf{c} = \{c_1, c_2, \dots, c_{N_c}\}$ and synchronously map ϕ_v and ϕ_a into N_c hyperbolic tangent spaces with these curvatures. Similar to Hyper-single, we make use of $\text{MLP}(\cdot) : \mathbb{R}^{600} \rightarrow \mathbb{R}^{N_c}$ to output curvatures $\mathbf{c}$. The total loss in Equation (4.2) applies Hyper-multiple, where the alignment loss $\mathcal{L}_{align}$ becomes:

$$\mathcal{L}_{align} = \frac{1}{N_c} \sum_i^{N_c} \mathcal{L}_{align,i} \quad (4.4)$$

where $\mathcal{L}_{align,i}$ is the hyperbolic alignment loss given in Equation (4.1) with the i^{th} curvature, $i \in \{1, 2, \dots, N_c\}$. Instead of restricting to only a single curvature, Hyper-multiple computes multiple appropriate curvatures to learn more generic embeddings.

4.4 Experiments

In this section, we provide extensive experiments to verify the effectiveness of our method. The proposed Hyperbolic Alignment Module is evaluated in audio-visual zero-shot classification on three datasets: VGGSound-GZSL, UCF-GZSL, and ActivityNet-GZSL. The details of each dataset are as follows. **VGGSound-GZSL** Mercea et al. [2022b] is a modified version of the audio-visual dataset VGGSound Chen et al. [2020b]. **UCF-GZSL** Mercea et al. [2022b] is a subset of the action video recognition dataset UCF101 Soomro et al. [2012] that includes audio information. **ActivityNet-GZSL** Mercea et al. [2022b] is based on the action recognition dataset ActivityNet Caba Heilbron et al. [2015]. The statistics of the class split for three datasets are provided in Table 1 of Mercea et al. [2022b] and Table 2 of the supplementary material of Mercea et al. [2022b].

We use classical metrics S and U for evaluating the performances of known and unknown classes, respectively. We also use the metric ZSL on unknown classes. In

Table 4.1: Experimental results of audio-visual zero-shot learning on three datasets (main feature). AVCA Mercea et al. [2022b] is adopted as the baseline for the proposed Hyper-alignment, Hyper-single, and Hyper-multiple modules. The best results in HM and ZSL are in **bold**. The curvatures of Hyper-alignment on VGGSound-GZSL, UCF-GZSL, and ActivityNet-GZSL are set as -0.5 , -0.2 , and -0.2 . The numbers of adaptive curvatures N_c for Hyper-multiple are 2, 3 and 2.

Method	VGGSound-GZSL				UCF-GZSL				ActivityNet-GZSL			
	S $\uparrow$	U $\uparrow$	HM $\uparrow$	ZSL $\uparrow$	S $\uparrow$	U $\uparrow$	HM $\uparrow$	ZSL $\uparrow$	S $\uparrow$	U $\uparrow$	HM $\uparrow$	ZSL $\uparrow$
CJME Parida et al. [2020]	8.69	4.78	6.17	5.16	26.04	8.21	12.48	8.29	5.55	4.75	5.12	5.84
AVGZSLNet Mazumder et al. [2021]	18.05	3.48	5.83	5.28	52.52	10.90	18.05	13.65	8.93	5.04	6.44	5.40
AVCA Mercea et al. [2022b]	14.90	4.00	6.31	6.00	51.53	18.43	27.15	20.01	24.86	8.02	12.13	9.13
Hyper-alignment	13.22	5.01	7.27	6.14	57.28	17.83	27.19	19.02	23.50	8.47	12.46	9.83
Hyper-single	9.79	6.23	7.62	6.46	52.67	19.04	27.97	22.09	23.60	10.13	14.18	10.80
Hyper-multiple	15.02	6.75	9.32	7.97	63.08	19.10	29.32	22.24	23.38	8.67	12.65	9.50

addition to ZSL, which solely evaluates the performance for unknown classes, the metric harmonic mean (HM) is used. It considers both known and unknown classes' performances, S and U, in the test stage: $HM = \frac{2US}{U+S}$.

All these metrics are evaluated on both the "main feature" and "cls feature," which indicate the pre-trained models used to extract data features $\mathbf{v}$, $\mathbf{a}$, and $\mathbf{w}$ Mercea et al. [2022b]. To fairly compare the performances of the Hyperbolic Alignment Module and the baseline, we keep all settings of hyperparameters in AVCA Mercea et al. [2022b], including the learning rate, training epoch, *etc*. In this case, κ_0 is set as -0.4 . In the ablation study, we analyze the effects of different parameters: curvature κ of Hyper-alignment and the number of curvatures N_c of Hyper-multiple.

4.4.1 Result Analysis

The main results of the proposed modules for audio-visual zero-shot learning are presented in Table 4.1 and 4.2. In general, Hyper-alignment outperforms the baseline in most cases. For instance, on UCF-GZSL^{cls}, Hyper-alignment achieves an accuracy of 42.52%/39.80% in HM/ZSL, which is higher than the baseline's accuracy of 41.34%/37.72%. Hyper-single, which computes an adaptive curvature for flexible exploration of audio-visual data structures, performs better than Hyper-alignment, except in the case of VGGSound-GZSL^{cls}. For example, on ActivityNet-GZSL, Hyper-single surpasses Hyper-alignment in HM/ZSL by 2.05%/0.97%. Meanwhile, Hyper-multiple, which learns multiple adaptive curvatures, clearly outperforms Hyper-single, except in the case of ActivityNet-GZSL.

As evidenced by our results on UCF-GZSL, Hyper-multiple outperforms all other methods, achieving a performance of 29.32%/22.24%. Moreover, the classification accuracy for known classes (S) using Hyper-multiple is typically higher than those of other approaches. This suggests that the Hyper-multiple module generates more generic representations that are better suited for audio-visual zero-shot learning. In light of these results, we conclude that Hyper-multiple is the preferred design.

In conclusion, our proposed Hyperbolic Alignment Module improves audio-visual zero-shot learning performance over the baseline, as demonstrated in Table

Table 4.2: Experimental results of audio-visual zero-shot learning on three datasets (cls feature). AVCA Mercea et al. [2022b] is adopted as the baseline for the proposed Hyper-alignment, Hyper-single, and Hyper-multiple modules. The best results in HM and ZSL are in **bold**. The curvatures of Hyper-alignment on VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} are set as -0.1 , -0.2 and -0.2 . The numbers of adaptive curvatures N_c for Hyper-multiple are 3, 2 and 3.

Method	VGGSound-GZSL ^{cls}				UCF-GZSL ^{cls}				ActivityNet-GZSL ^{cls}			
	S $\uparrow$	U $\uparrow$	HM $\uparrow$	ZSL $\uparrow$	S $\uparrow$	U $\uparrow$	HM $\uparrow$	ZSL $\uparrow$	S $\uparrow$	U $\uparrow$	HM $\uparrow$	ZSL $\uparrow$
CJME Parida et al. [2020]	10.86	2.22	3.68	3.72	33.89	24.82	28.65	29.01	10.75	5.55	7.32	6.29
AVGZSLNet Mazumder et al. [2021]	15.02	3.19	5.26	4.81	74.79	24.15	36.51	31.51	13.70	5.96	8.30	6.39
AVCA Mercea et al. [2022b]	12.63	6.19	8.31	6.91	63.15	30.72	41.34	37.72	16.77	7.04	9.92	7.58
Hyper-alignment	12.50	6.44	8.50	7.25	57.13	33.86	42.52	39.80	29.77	8.77	13.55	9.13
Hyper-single	12.56	5.03	7.18	5.47	63.47	34.85	44.99	39.86	24.61	10.10	14.32	10.37
Hyper-multiple	15.62	6.00	8.67	7.31	74.26	35.79	48.30	52.11	36.98	9.60	15.25	10.39

4.1 and 4.2, with effective geometric alignments.

4.4.2 Ablation Study

Curvature. Experiments of Hyper-alignment are conducted under varying curvatures to understand how the degree to which the Euclidean space is distorted by hyperbolic geometry affects the model’s performance. In addition to the hyperbolic space ($\kappa < 0$), feature alignments are also performed in Euclidean space ($\kappa = 0$) and spherical space ($\kappa > 0$). The results of these experiments are shown in Table 4.3, which indicate that the performance of Hyper-alignment on UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} peaks at a curvature of -0.2 . This suggests that there is an optimal curvature for Hyper-alignment that achieves the best performance. Moreover, we observe that the model is robust against the curvature of the hyperbolic space, as performances with different curvatures are better than the baseline in most cases.

Mixed curvature learning. In this experiment, variants of multiple geometric curvatures are evaluated. Hyper-multiple combines two hyperbolic alignments when the number of curvatures N_c is set to 2. Other combinations, such as Euclidean and spherical alignments, spherical and hyperbolic alignments, are also tested. The results in Table 4.4 show that the combination of two hyperbolic alignments (H + H) achieves the best performance among all the combinations. This observation is consistent with the results in Table 4.3, which demonstrate the advantage of hyperbolic geometry over other geometries in audio-visual zero-shot learning.

Effectiveness of multiple curvatures. Here, we test the performance of Hyper-multiple with different numbers of adaptive curvatures, N_c , on VGGSound-GZSL and VGGSound-GZSL^{cls}. The results in Table 4.5 demonstrate that Hyper-multiple with 2 or 3 adaptive curvatures achieves the best performance. This suggests that, for Hyper-multiple, having more curvatures does not necessarily lead to better performance in audio-visual zero-shot learning.

Audio-visual feature alignment. In this chapter, we present the hyperbolic alignment modules that align the visual and audio features ϕ_v and ϕ_a before the cross-

Table 4.3: Ablation study: curvature. Different curvature values of Hyper-alignment are tested on UCF-GZSL^{cls} and ActivityNet-GZSL^{cls}. Besides Hyper-alignment, Euclidean-alignment and Sphere-alignment are also considered. From the table, Hyper-alignment shows advantages over Euclidean-alignment and Sphere-alignment.

Method	κ	UCF-GZSL ^{cls}		ActivityNet-GZSL ^{cls}	
		HM $\uparrow$	ZSL $\uparrow$	HM $\uparrow$	ZSL $\uparrow$
AVCA Mercea et al. [2022b]	-	41.34	37.72	9.92	7.58
Sphere-alignment	1.0	38.62	31.64	9.26	6.23
	0.8	35.33	31.58	10.24	7.01
	0.6	38.80	38.62	10.28	7.01
	0.4	38.69	38.22	10.55	7.58
	0.2	38.56	34.45	10.56	6.95
	0.1	34.14	31.72	10.13	6.74
Euclidean-alignment	0.0	38.82	36.01	10.47	8.00
Hyper-alignment	-0.1	40.64	38.09	12.51	8.67
	-0.2	42.52	39.80	13.55	9.13
	-0.4	41.62	36.83	13.15	9.08
	-0.6	37.69	36.56	12.86	9.77
	-0.8	35.48	36.16	12.63	9.62
	-1.0	38.25	34.11	10.95	7.30

Table 4.4: Ablation study: mixed curvature learning. Different variants of mixed-curvature geometries are tested on UCF-GZSL^{cls} and ActivityNet-GZSL^{cls}. H, S and E indicate hyperbolic, spherical and Euclidean alignments. From the table, Geometry-multiple, which incorporates two hyperbolic geometries, gains the most performance improvements.

Method	κ	UCF-GZSL ^{cls}		ActivityNet-GZSL ^{cls}	
		HM $\uparrow$	ZSL $\uparrow$	HM $\uparrow$	ZSL $\uparrow$
AVCA Mercea et al. [2022b]	-	41.34	37.72	9.92	7.58
Mixed-curvature learning	E + E	39.26	37.05	9.83	7.35
	E + S	38.31	36.71	11.93	8.03
	E + H	42.73	37.74	7.49	5.70
	S + S	23.97	33.02	10.02	6.32
	S + H	40.37	38.16	12.25	7.78
	H + H	48.30	52.11	13.70	8.95

attention module of the baseline. In this experiment, we investigate aligning the features after the cross-attention and compare the results in Table 4.6. As seen from the table, aligning ϕ_v and ϕ_a before the cross-attention achieves the best result. On the other hand, aligning $\phi_{v,att}$ and $\phi_{a,att}$ harms the performance. These results suggest that the cross-modal fusion performed by the cross-attention module may alter the hierarchy of the data’s original features from each modality.

4.4.3 Discussion

δ -Hyperbolicity. The underlying structures of audio-visual datasets may exhibit high non-Euclidean latent geometry since there is a rich hierarchy in audio-visual data. In this subsection, we demonstrate that the proposed hyperbolic alignment modules can uncover some hierarchical knowledge. To measure the degree of the hierarchy

Table 4.5: Ablation study: effectiveness of multiple curvatures. Different curvature numbers of Hyper-multiple are evaluated on VGGSound-GZSL and VGGSound-GZSL^{cls}.

Method	N_c	VGGSound-GZSL		VGGSound-GZSL ^{cls}	
		HM $\uparrow$	ZSL $\uparrow$	HM $\uparrow$	ZSL $\uparrow$
AVCA Mercea et al. [2022b]	-	6.31	6.00	8.31	6.91
Hyper-multiple	1	7.62	6.46	7.18	5.47
	2	9.32	7.97	8.97	6.72
	3	8.91	7.12	8.67	7.31

Table 4.6: Ablation study: audio-visual feature alignment. Different locations of aligned features by Hyper-single are tested on datasets UCF-GZSL and ActivityNet-GZSL.

Method	aligned feature	UCF-GZSL		ActivityNet-GZSL	
		HM $\uparrow$	ZSL $\uparrow$	HM $\uparrow$	ZSL $\uparrow$
AVCA Mercea et al. [2022b]	-	27.15	20.01	12.13	9.13
Hyper-single	ϕ_v / ϕ_a	27.97	22.09	14.18	10.80
	$\phi_{v,att} / \phi_{a,att}$	13.78	11.89	8.71	6.59
	$\mathbf{e}_v / \mathbf{e}_a$	25.58	18.77	12.96	10.00

inside the features, we use the δ -Hyperbolicity metric, which effectively detects hierarchical structure Khrulkov et al. [2020]; Fournier et al. [2015]. In our case, we compute δ_{rel} Khrulkov et al. [2020]; Fournier et al. [2015], a metric overall testing sample features that indicate the degree to which the feature distribution resembles a tree-like structure. Smaller values of δ_{rel} reflect stronger hyperbolicity within the feature space. As we can see from Table 4.7, Hyper-alignment achieves lower δ_{rel} values than AVCA on VGGSound-GZSL and ActivityNet-GZSL^{cls}. The results support our hypothesis that aligned representations using hyperbolic alignment loss convey more hierarchical information.

Table 4.7: Discussion: δ -Hyperbolicity. We compute δ_{rel} on the feature $\mathbf{e}_v$. The smaller value of δ_{rel} indicates the stronger hyperbolicity inside $\mathbf{e}_v$. Hyper-alignment gets a lower value of δ_{rel} than ACVA, which indicates the representations via the hyperbolic alignment loss learn more hierarchical structures.

Method	VGGSound-GZSL	ActivityNet-GZSL ^{cls}
	δ_{rel}	δ_{rel}
AVCA Mercea et al. [2022b]	0.30	0.37
Hyper-alignment	0.27	0.32

t-SNE Visualizations. In addition to measuring δ -Hyperbolicity, we provide t-SNE visualizations that reveal the capability of hyperbolic spaces in exploring data hierarchy. As shown in Figure 4.3, the hyperbolic alignment loss facilitates pulling together features from the same parent class while pushing away features belonging to different parent classes on UCF-GZSL^{cls}. For instance, features of "CricketShot" and "FieldHockeyPenalty," which belong to the same parent class "Sports," are pulled together, while features of "CricketShot" and "HandstandWalking," which belong to different parent classes, are pushed away. To some extent, this explains how hyper-

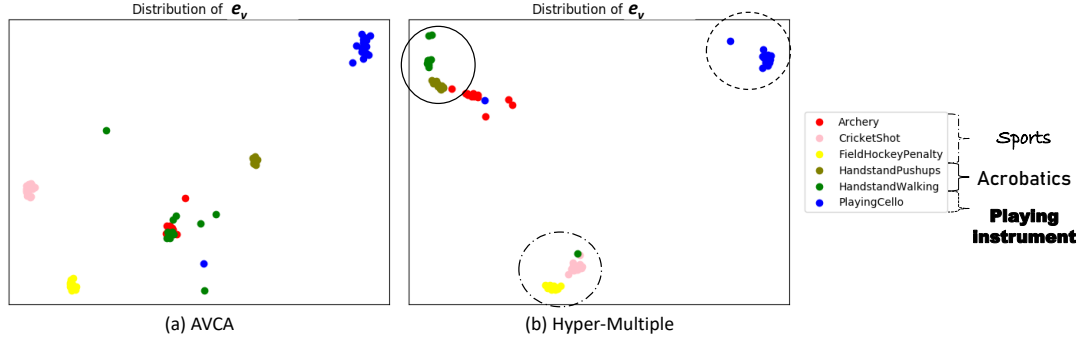


Figure 4.3: Visualization examples on UCF-GZSL^{cls}. We give t-SNE visualizations of $\mathbf{e}_v$ from six known classes: “Archery”, “CricketShot”, “FieldHockeyPenalty”, “HandstandPushups”, “HandstandWalking” and “PlayingCello”. They can be categorized into three superclasses: “Sports”, “Acrobatics” and “Playinginstruments”. Hyper-multiple, which learns a hyperbolic alignment loss, pushes away features from different superclasses.

bolic alignment between visual and audio features in hyperbolic spaces enables more accurate audio-visual zero-shot classification. Similarly, the visualization in Figure 4.4 suggests that on ActivityNet^{cls}, features belonging to different superclasses are more discriminately distributed than features from the same superclass. These visualizations somewhat describe how our approach uncovers the hierarchical knowledge among classes.

4.5 Summary

In this chapter, we introduce a novel hyperbolic alignment module that improves cross-modality features in audio-visual zero-shot learning. This is achieved by leveraging the properties of hyperbolic spaces to explore more hierarchical structures within the audio-visual data and generate more distinctive features. Our approach includes three modules, Hyper-alignment, Hyper-single, and Hyper-multiple, for computing the hyperbolic cross-modality loss, and we demonstrate their effectiveness through empirical evaluations.

In the first part of the thesis, we discuss how to address low-shot learning challenges across two chapters. They are the explorations of low-shot learning, with the first chapter focusing on few-shot learning (see Chapter 3) and the second on audio-visual zero-shot learning (see Chapter 4). These two chapters share a common point: both chapters aim to enhance mid-level features within the backbone (or encoder) network. However, they differ in data types and feature learning approaches. Chapter 3 processes image data, concentrating on attending feature maps by reinforcement learning, while Chapter 4 deals with audio-visual data, focusing on aligning feature embeddings from multiple modalities in the hyperbolic space.

Subsequently, the thesis will transition from the part of low-shot learning to an-

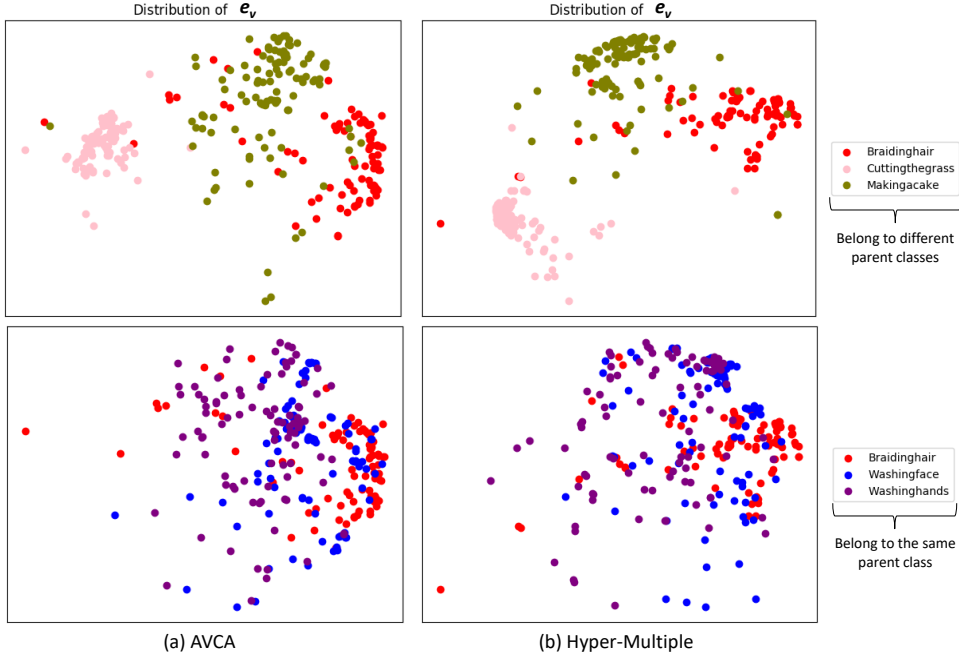


Figure 4.4: Visualization examples on ActivityNet-GZSL^{cls}. We give t-SNE visualizations of $\mathbf{e}_v$. The features $\mathbf{e}_v$ of three unknown classes, which are from different parent classes, are visualized in the upper figure. The lower figure depicts the feature distribution of three unknown classes from the same parent class. Comparing two figures, Hyper-multiple pushes away features more from different superclasses than the same superclass.

other part, *i.e.*, open-set recognition. This part will address recognizing unknown objects from known data, which are more closely related to real-world applications. Following this chapter, which studies the geometric solution, in the forthcoming chapter, we will explore the application of curved geometries in feature learning of open-set recognition. The investigation aims to understand the advantages curved geometries offer in enhancing feature learning of open-set recognition.

Part II

Limited-data Visual Problem: Open-set Recognition

Curved Geometric Learning for Open-set Recognition

The previous part of the thesis, specifically the last two chapters, has focused on addressing a key aspect of limited-data visual problems: low-shot learning. This includes exploring few-shot learning in Chapter 3 and audio-visual zero-shot learning in Chapter 4. Commencing with this part, our discussion shifts toward another critical aspect of limited-data visual problems: open-set recognition.

Learning feature embeddings to understand the underlying nature of data distribution is often formulated in Euclidean spaces with zero curvature. However, in the previous chapter, we demonstrate the success of the geometry constraints posed in the embedding space. The experimental outcomes indicate that curved space might encode more structural information, leading to better discriminative power and, hence, richer representations of feature embeddings. In this chapter, we investigate the benefits of the curved space for recognizing unknown (or open-set/anomalous/out-of-distribution) objects in data. Compared to the previous chapter, where we only study hyperbolic geometry with negative curvature, in this chapter, we consider embeddings via another two geometry constraints, namely, spherical geometry with positive curvature and mixed geometry with both positive and negative curvatures.

Three geometric constraints can be chosen interchangeably in a unified design, projecting the Euclidean embeddings to the corresponding curved space. Two types of geometric modules (*i.e.*, Geometric-in-One and Geometric-in-Two) are proposed to plug in to construct the geometric classifier. Tailored for the embeddings in the curved space, we also formulate functions to compute the open-set score. We evaluate the resulting designs under a diverse set of visual open-set recognition scenarios, including image detection (multi-class out-of-distribution detection and one-class open-set detection) and segmentation (multi-class open-set segmentation and one-class anomaly segmentation). The empirical results show the effectiveness of our proposal through consistent improvement over various scenarios. This chapter is based on our published work Hong et al. [2023a].

5.1 Introduction

In this chapter, we aim to leverage the curved geometry for learning feature embeddings, allowing us to analyze and identify unknown, open-set, anomalous, or out-of-distribution (OOD)¹ objects from known, closed-set, normal, or in-distribution (ID)² input data. Non-flat geometry has gained an increasing amount of interest in various machine learning approaches due to its intriguing properties in encoding the inherent geometry or hidden structure information of the data Nickel and Kiela [2017]; Tifrea et al. [2018]; Dhingra et al. [2018]; Khrulkov et al. [2020]; Park et al. [2021]; Zhang et al. [2021a]. For example, spherical spaces with constant positive curvature show appealing properties along with deep neural networks (DNNs) to encode samples resembling the sphere Liu et al. [2017a]; Fan et al. [2019]. Hyperbolic spaces, featured with a constant negative curvature, are shown to be rich in encoding the underlying hierarchical structure in the data. Such a property enables hyperbolic spaces to discriminate input samples better Khrulkov et al. [2020]. Figure 2.1 conceptualizes the distinctive characteristics of Euclidean, spherical, and hyperbolic spaces and highlights their disparities. Euclidean spaces with zero curvature are familiar faces in DNNs (see Figure 2.1 (a)). Spherical geometry, as shown in Figure 2.1 (b), has been successfully employed to encode directional data, *i.e.*, samples whose magnitude does not carry important information. The Poincaré ball model for the hyperbolic spaces, as shown in Figure 2.1 (c), provides a structure with constant negative curvature to encode data.

Previous studies such as Liu et al. [2020b]; Khrulkov et al. [2020]; Shen et al. [2021] show that curved spaces can attain a superior performance gain over the Euclidean space, especially for tasks relying on image feature embeddings (*e.g.*, zero/few-shot learning or metric learning). For example, in Shen et al. [2021], by employing the similarity metric in spherical embedding spaces, the model enhances its discriminative ability in zero-shot classification for unknown classes. In Khrulkov et al. [2020]; Fang et al. [2021], hyperbolic spaces are shown to have a better distribution across unknown classes, improving few-shot learning performance. Based on the interpretations in Figure 2.1, we infer that the curved geometric embeddings exhibit enhanced discrimination due to the following reasons:

- In the spherical space, all points are constrained to lie on the surface of a hypersphere, as shown in (see Figure 2.1 (b)). On the surface of the hypersphere, the embedding space is enforced to have a uniform distribution across multiple classes, promoting a more unbiased classification. Furthermore, scale-free analysis, the inclusion of spherical spaces, *e.g.*, via ℓ_2 normalization in a DNN, provides useful constraints for learning and has been used with success in the majority of modern neural structures Liu et al. [2017a]; Fan et al. [2019]; Skopek et al. [2020].

¹We mainly use the terms “known” and “open-set” throughout this chapter.

²We mainly use the terms “unknown” and “closed-set” throughout this chapter.

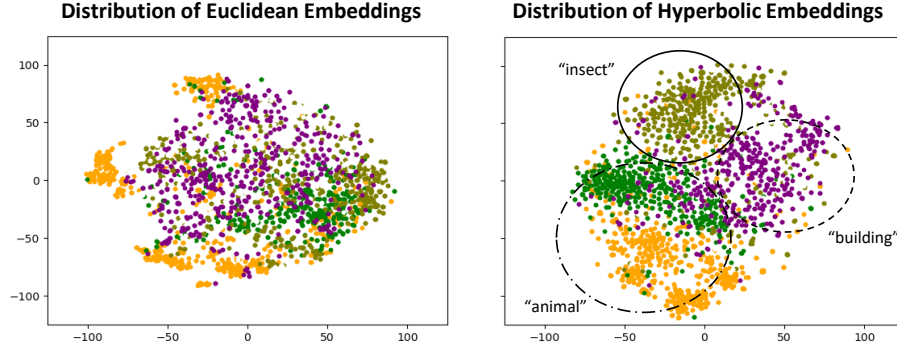


Figure 5.1: An illustration of the discrimination ability of curved spaces on unknown data. We train two models using Euclidean and hyperbolic geometry on *miniImageNet* dataset Ravi and Larochelle [2017], following the few-shot learning protocol. We randomly choose four unknown classes and plot the class embeddings using t-SNE Van der Maaten and Hinton [2008]. Each unknown class is represented by one color. From this visualization, we can directly observe that within each unknown class, embeddings in hyperbolic space (See the figure on the right) show a more compact distribution than embeddings in Euclidean space (See the figure on left). Moreover, it can be observed that hyperbolic embeddings exhibit improved separations based on the discovered hierarchies. For instance, when compared to the distribution of Euclidean embeddings, hyperbolic embeddings show closer proximity between classes (*i.e.*, the yellow-colored “African hunting dog” and the green-colored “lion”) that belong to the same superclass (*i.e.*, ‘animal’). Conversely, hyperbolic embeddings of classes (*i.e.*, the olive-colored “ant” and the purple-colored “bookshop”), which belong to different superclasses (*i.e.*, “insect” and “building”), are pulled further apart.

- As shown in Figure 2.1 (c), points in the hyperbolic space lie inside the ball. The volume in a hyperbolic space expands exponentially with dimensions. This enables us to embed tree-like structures with less distortion (see the green-colored triangle) and aids the discovery of hierarchical relationships among samples. For instance, as illustrated in Figure 5.1 and compared to Euclidean embeddings, the hyperbolic embeddings of “African hunting dog” (yellow) and “lion” (green) are closer, emphasizing their similarity at the superclass level (*i.e.*, “animals”). Moreover, the hyperbolic embeddings of “ant” (olive) and “bookshop” (purple) exhibit better separation than Euclidean embeddings as they originate from distinct superclasses, “insect” and “building.” Learning such hierarchies will lead to success at the inference.

With the appealing observations shown above, in this chapter, we investigate the practice of using geometries with fixed non-zero curvature in open-set recognition tasks. For the purpose of realizing our idea, a natural solution is to simply replace the existing Euclidean classifier with one based on a curved geometry. This idea is

behind the design of our Geometric-in-One model. In addition, we also find that the ‘divergence’ between Euclidean embeddings and curved embeddings can provide a reliable indicator useful for open-set recognition. To benefit from this interesting observation, we further develop a Geometric-in-Two model. Having multiple geometry-aware networks at our disposal, we further present approaches for getting the open-set (or anomaly/OOD) score³ to identify unknown objects. Our main objective is to show that geometry is essential in identifying unknown objects. As such, we develop a generic solution and incorporate it into various baselines in our empirical study. The contributions of this work can be summarized as:

- We propose two lightweight curvature-aware geometric networks for the visual open-set recognition. To our knowledge, this was the first attempt to adopt curved manifolds as embedding spaces to distinguish known and unknown data by the time of the publication of this work. Additionally, multiple curved spaces, including spherical, hyperbolic, and mixed spaces, are studied.
- Extensive experiments on a wide range of visual open-set recognition tasks (*e.g.*, multi-class OOD detection, one-class open-set detection, multi-class open-set segmentation, and one-class anomaly segmentation) suggest that the proposed technique leads to a substantial performance gain over the Euclidean geometry.

5.2 Related Work

5.2.1 Open-set Image Recognition

Three main approaches are developed for doing open-set (or anomaly/OOD) image recognition: confidence-, generative-, and self-supervised-based methods. However, few works are learning features in non-Euclidean spaces.

Confidence-based method. It is well known that the confidence from the softmax in a classifier helps to detect OOD samples from ID samples since ID samples are more likely to have a greater maximum softmax confidence compared to OOD samples Hendrycks and Gimpel [2017]. OOD detector for Neural Network (ODIN) Liang et al. [2018] applies temperature scaling to the confidence vector and adds small perturbations to input samples for more accurate OOD detection. Additional confidence-based methods which make use of the confidence have been studied in DeVries and Taylor [2018]; Corbière et al. [2019]; Hsu et al. [2020]; Hendrycks et al. [2022]; Jang and Kim [2022].

Generative-based method. One of the generative-based methods is to synthesize effective training samples to avoid the deep neural networks becoming overconfident in their predictions Lee et al. [2018]; Sabokrou et al. [2018]; Lis et al. [2019]; Xia et al. [2020]; Kong and Ramanan [2021]. Another choice is to optimize features in the latent space of an encoder-decoder network towards generating a more general distribution Ren et al. [2019a]; Abati et al. [2019]; Perera et al. [2019]; Gong et al.

³We mainly use the term “open-set score” throughout this chapter.

[2019]; Massoli et al. [2021] or a more representative attention map Venkataramanan et al. [2020]; Liu et al. [2020c]; Zhou et al. [2021c].

Self-supervised-based method. Self-supervised learning techniques have been widely employed in open-set recognition. Ensemble Leave-out Classifier (ELOC) Vyas et al. [2018] trains classifiers in a self-supervised manner by setting a subset of training data as OOD data. One main idea behind the self-learning method is to apply geometric augmentations on the input images and train a multi-class model to discriminate such augmentations. Prediction of image rotation is used in Rotation Network (RotNet) Gidaris et al. [2018]. Jittered patches of an image are classified in Patch-SVDD Yi and Yoon [2020] for anomaly localization. Another idea is to use contrastive learning for learning better visual features Tack et al. [2020]; Zheng et al. [2022]. More works using self-supervised learning are presented in Ruff et al. [2018]; Golan and El-Yaniv [2018]; Hendrycks et al. [2019]; Bergman and Hoshen [2020]; Tack et al. [2020]; Li et al. [2021]; Liu et al. [2021].

Some other facts should be noted. 1) Confidence-based and self-supervised-based methods mainly adopt “encoder-classifier” structures, while generative-based models have “encoder-decoder” architectures. The proposed modules in our work are best applied to an “encoder-classifier”⁴ rather than an “encoder-decoder” structure. 2) Teacher-student structures have been utilized in anomaly recognition problems Yu and Aizawa [2019]; Bergmann et al. [2020]; Salehi et al. [2021]; Wang et al. [2021a]. These approaches demonstrate the effectiveness of leveraging the discrepancy between the teacher and student branches to identify and localize anomalous objects. In Salehi et al. [2021], a pair of networks are employed for anomaly detection, where the teacher network is pre-trained on ImageNet. The discrepancy between multi-level layers is utilized to compute anomaly scores. In the case of the Local-Global Net proposed in Wang et al. [2021a], a teacher-student architecture is trained to compute local and global features, respectively. In the inference, Local and global features are compared to compute the anomaly score. 3) Some approaches compute the anomaly score of test samples by comparing their features with those of training samples Yi and Yoon [2020]; Li et al. [2021]. In the Patch-SVDD method Yi and Yoon [2020], for instance, the anomaly scores of test samples are obtained by calculating the L2 distance between the features of the test and training samples.

5.2.2 Geometric Learning

In Section 4.2, we presented an overview of related works in hyperbolic learning. In this section, we introduce the broader domain of hyperbolic learning, *i.e.*, geometric learning.

Geometric learning has been studied extensively to encode structured representations Bronstein et al. [2017]. For example, the set has been used to model order-invariant data (*e.g.*, 3D point clouds or video data). Orthogonal constraints, *i.e.*, subspaces, are often used to encode set data Cheraghian et al. [2021], for its potential to

⁴“Encoder” and “backbone” are used interchangeably in this chapter.

be robust against illumination variations, background, *etc.* In vectorized representations, spherical or hyperbolic spaces are also very effective for metric learning-related tasks. In the spherical space, the similarity of representations is upper-bounded. Hence, such a space is particularly well-behaved at learning a metric space Bronstein et al. [2005]; Liu et al. [2017a,b]. As opposed to the spherical space, tree-like data can be embedded in the hyperbolic space for its intriguing property to capture the hierarchical structure of the data Liu et al. [2020b]; Khrulkov et al. [2020]. To further increase the discrimination power of the learned embeddings in curved spaces, the kernel methods, which implicitly map the geometric representation to a high or even infinite dimensional feature space, are studied for spherical embeddings Jayasumana et al. [2022], or hyperbolic embeddings Fang et al. [2021]. To fully model the structure of the data, mixed-curved spaces are good candidates as embedding spaces Gu et al. [2019]; Skopek et al. [2020].

5.3 Methodology

Visual open-set recognition aims to identify unknown samples from known samples. During the training process, as illustrated in Figure 5.2, only known data can be accessed. For the evaluation stage, as shown in Figure 5.3, both known and unknown inputs to be recognized exist.

The pipeline of the baseline is illustrated in Figure 5.2 (a) for the training phase and Figure 5.3 (a) for the inference phase. Specifically, an image encoder first maps the input image to a feature embedding $\mathbf{e}_E$, lying in a Euclidean space. The following FC layer and Softmax function further predict the probability belonging to each known class, denoted by $\mathbf{c}_E$ (see Figure 5.2 (a)). In the evaluation process, to identify whether the input image or pixel is unknown, one needs to define the open-set score $OS \in [0, 1]$. In this vanilla model (see Figure 5.3 (a)), we follow the common practice in Hendrycks and Gimpel [2017] to define the open-set score by leveraging the predication $\mathbf{c}_E$ as $OS = 1 - \max(\mathbf{c}_E)$.

A curvature-aware geometric model indicates a model where its classifier operates in a curved space $\mathcal{M}_G$, and we term its classifier geometric classifier. Two curvature-aware geometric models are presented in this section: Geometric-in-One (GiO) and Geometric-in-Two (GiT), as shown in Figure 5.2 (b) and (c). Compared to the baseline model, the curvature-aware geometric model has two geometric layers, namely, geometric transformation and geometric FC layer. The geometric transformation is to transform the Euclidean embedding $\mathbf{e}_E$ computed by an image encoder to the geometric embedding $\mathbf{e}_G$. See Equation (2.2) for the spherical geometry $\mathcal{M}_S$ and Equation (2.4) for the hyperbolic geometry $\mathcal{M}_H$. The geometric FC layer is a generalization of the FC layer in $\mathcal{M}_G$ (*e.g.*, the angular linear layers in $\mathcal{M}_S$ or the hyperbolic linear layers in $\mathcal{M}_H$). As shown in Figure 5.2, we learn geometric classifiers where embeddings extracted from an image encoder are manipulated in the curved spaces. During the inference phase, as illustrated in Figure 5.3 (b) and (c), the geometric score z is initially obtained from the curved embedding. In GiO, the variable

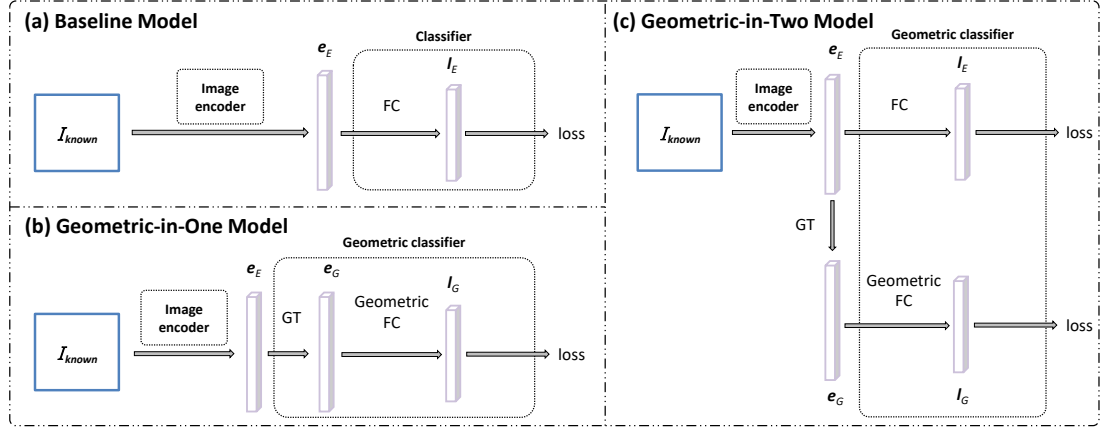


Figure 5.2: The general frameworks of the models and their training processes are depicted. (a) The baseline model follows an “encoder-classifier” structure. The image encoder takes the input image I and encodes it into a feature embedding e_E . The classifier, primarily composed of a fully-connected (FC) layer, generates the confidence vector for loss computation. We propose two models in our work: Geometric-in-One (GiO) and Geometric-in-Two (GiT). GiO and GiT differ from the baseline model in integrating the geometric transformation (GT) and geometric FC layer into the classifier. In essence, while the baseline model follows an “encoder-classifier” structure, GiO and GiT adopt an “encoder-geometric-classifier” structure. (b) In the GiO model, the original Euclidean FC layer is replaced with a geometric FC layer based on a curved geometry. (c) GiT model adopts the geometric FC layer as the additional branch. In the training stage, we follow the setting that the training data does not include unknown samples.

z can be interpreted as the distance value between the curved embedding point and the reference point in the curved space. In GiT, z represents the discrepancy between two embeddings originating from different spaces. Subsequently, the open-set score in our geometric model is computed as $OS = 1 - \tanh(z)$. Since $z \in [0, +\infty)$ is not constrained within the range of 0 to 1, the function $\tanh(\cdot)$ is employed to make sure the open-set score OS falls between 0 and 1.

5.3.1 Geometric-in-One Model

We first introduce a single-branch framework, *i.e.*, the Geometric-in-One (GiO) model. The GiO model (see Figure 5.2 (b)) is a natural modification that replaces the Euclidean embeddings with curved-geometry embeddings. It provides a straightforward comparison to justify the advantages of using curved geometry for anomaly recognition. Specifically, as shown in Figure 5.2 (b), the image or pixel embeddings can be obtained by applying the geometric transformation function (*i.e.*, Equation (2.2) for $\mathcal{M}_S$ and Equation (2.4) for $\mathcal{M}_H$) to the feature embedding encoded by the image encoder. Then, a following geometric classifier, realized by the geometric

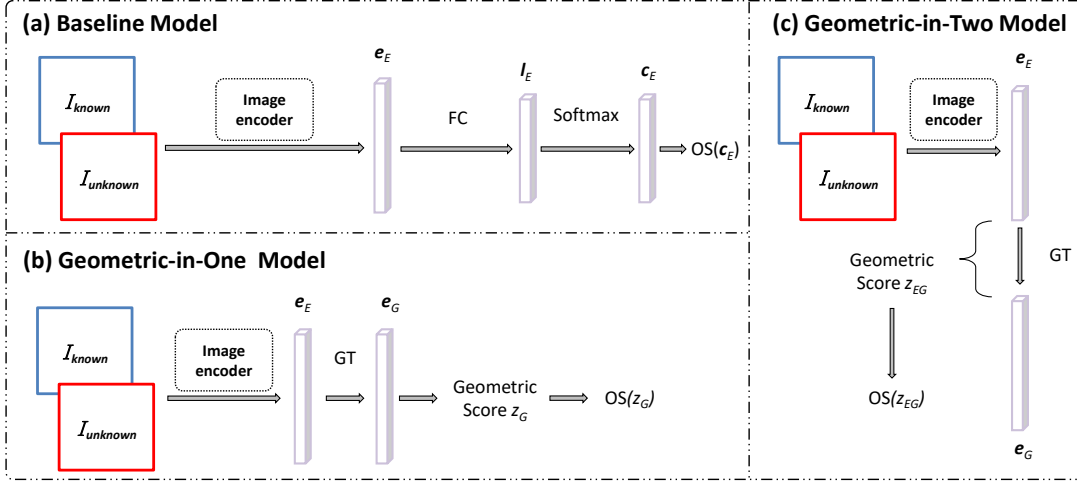


Figure 5.3: The general frameworks of the models and their evaluation processes are depicted. In the evaluation stage, the model processes known and unknown inputs. The geometric score z_G is extracted from the curved embedding for open-set score computation in the GiO model. For the GiT model, the geometric score z_{EG} is obtained via Euclidean and curved embeddings. During the evaluation process, GiO and GiT models use fewer parameters than the baseline.

FC layer, is further used to predict the class of its input. For example, the Hyperbolic-in-One model refers to a network with a geometric classifier in the hyperbolic space. Compared to the baseline model (see Figure 5.2 (a)), the GiO model only modifies the embedding layer and classifier without bringing extra parameters, thereby being a cheap yet flexible solution.

In the GiO model, the geometric score $z_G \in \{z_S, z_H, z_M\}$ can be obtained from embedding $\mathbf{e}_G$. In our work, the geometric score from the spherical manifold $\mathcal{M}_S$ is defined as

$$z_S(\mathbf{e}_S) = \max(l_S(\mathbf{e}_S, \mathbf{W})), \quad (5.1)$$

where $\mathbf{e}_S \in \mathbb{S}_k^{n-1}$. Experiments in Liu et al. [2017a]; Massoli et al. [2021] verified that z_S from $\mathcal{M}_S$ is suitable for visual tasks under the open-set protocol. Hence, we expect z_S to help open-set recognition. For the hyperbolic space $\mathcal{M}_H$, the geometric score is defined as

$$z_H(\mathbf{e}_H) = d_{Geo}(\mathbf{e}_H, \mathbf{0}_H), \quad (5.2)$$

where $d_{Geo}(\mathbf{e}_H, \mathbf{0}_H)$ is also known as the geodesic distance between the point $\mathbf{e}_H$ and the origin $\mathbf{0}_H$, for $\mathbf{e}_H, \mathbf{0}_H \in \mathbb{H}_k^n$. Experiment of image-level OOD detection in Khrulkov et al. [2020] shows the property of $\mathbf{e}_H$, the hyperbolic embedding points of OOD samples are closer to the origin. In the mixed-curvature manifold $\mathcal{M}_M$, the

Table 5.1: **The proposed geometric networks** in the Geometric-in-One (GiO) Model and the Geometric-in-Two (GiT) Model. “Abb.”, “Geo. Components” and “Geo. Score” are “abbreviation”, “geometry components” and “geometry score” respectively.

Method	Abb.	Geo. Components	Geo. Score
Geometric-in-One (GiO) Model			
Spherical-in-One	SiO	$\mathcal{M}_S$	Equation (5.1)
Hyperbolic-in-One	HiO	$\mathcal{M}_H$	Equation (5.2)
Mixed-in-One	MiO	Multiple $\mathcal{M}$	Equation (5.3)
Geometric-in-Two (GiT) Model			
Spherical-in-Two	SiT	$\mathcal{M}_S$	Equation (5.4)
Hyperbolic-in-Two	HiT	$\mathcal{M}_H$	Equation (5.4)
Mixed-in-Two	MiT	Multiple $\mathcal{M}$	Equation (5.5)

geometric score z_M is formulated as:

$$z_M^2 = \sum_i^N z_{M,i}^2, \quad (5.3)$$

where there are N spaces in $\mathcal{M}_M$ and $z_{M,i}$ is the geometric score from the i th component space. For instance, in $\mathcal{M}_M = \mathcal{M}_S \times \mathcal{M}_H$, $z_M^2 = z_S^2 + z_H^2$. The GiO model with $\mathcal{M}_M = \mathcal{M}_S \times \mathcal{M}_H$ is actually a two-branch architecture. We can even incorporate the Euclidean space into the Mixed-in-One model. In $\mathcal{M}_M = \mathcal{M}_E \times \mathcal{M}_S \times \mathcal{M}_H$, we can have $z_M^2 = z_E^2 + z_S^2 + z_H^2$ where the geometric score from Euclidean space is $z_E = \max(c_E)$ Hendrycks and Gimpel [2017]. Table 5.1 lists the proposed geometric networks, in conjunction with the geometry score, in the GiO model.

Having the geometry score z_G at our disposal, the open-set score is defined as $OS = 1 - \tanh(z_G)$. A higher value of OS indicates a higher probability that the input comes from the anomalous distribution. The purpose of $\tanh(\cdot)$ is to normalize z_G to a value that is in the range 0 and 1 (see Figure 5.3 (b)).

5.3.2 Geometric-in-Two Model

Several recent works Yu and Aizawa [2019]; Bergmann et al. [2020]; Salehi et al. [2021]; Wang et al. [2021a] develop the dual-branch architecture and exploit the discrepancy between features in separate classifiers for anomaly, open-set, or OOD recognition. Motivated by this fact, we further introduce our second framework, termed Geometric-in-Two (GiT), where a Euclidean classifier and a geometric classifier are integrated after the image encoder (see Figure 5.2 (c)). In the GiT model, in parallel with a branch of the Euclidean classifier, the other branch learns the feature embedding in the curved space, and a following geometric FC layer is used as a class predictor. The embedding in the curved space $\mathcal{M}_G$ is achieved by transforming $\mathbf{e}_E$ to $\mathbf{e}_G$ via a GT function, as shown in Figure 5.2 (c). In such a pipeline, the geometry-aware score z_{EG} is defined as the discrepancy between distributions of $\mathbf{e}_E$ and $\mathbf{e}_G$,

measured via the Kullback-Leibler (KL) divergence, as follows:

$$z_{EG} = \sum_i p_{E,i} \log \frac{p_{E,i}}{p_{G,i}}, \quad (5.4)$$

where $p_{E,i}$ and $p_{G,i}$ are the i th element in $\mathbf{p}_E$ and $\mathbf{p}_G$. Here, $\mathbf{p}_E = \text{softmax}(\mathbf{e}_E)$ and $\mathbf{p}_G = \text{softmax}(\mathbf{e}_G)$. The geometric score z_{EG} is essentially the distribution discrepancy between the learned embedding $\mathbf{e}_E$ from $\mathcal{M}_E$ and $\mathbf{e}_G$ from $\mathcal{M}_G$. We have three types of GiT models: Spherical-in-Two, Hyperbolic-in-Two and Mixed-in-Two models, thereby $z_{EG} \in \{z_{ES}, z_{EH}, z_{EM}\}$ where z_{ES} , z_{EH} and z_{EM} are from three models, respectively. The values z_{ES} and z_{EH} can be easily calculated by Equation (5.4). Inspired by Equation (5.3), we define z_{EM} in $\mathcal{M}_M$ as follows:

$$z_{EM}^2 = \sum_i^N z_{EM,i}^2, \quad (5.5)$$

where N indicates the number of component spaces in $\mathcal{M}_M$. For example, when $\mathcal{M}_M = \mathcal{M}_S \times \mathcal{M}_H$, the score metric can be obtained by $z_{EM}^2 = z_{ES}^2 + z_{EH}^2$. The GiT model with $\mathcal{M}_M = \mathcal{M}_S \times \mathcal{M}_H$ is actually a three-branch architecture.

Our experiments show that z_{EG} can provide reliable discrimination information for anomaly identification. We find that similar to our GiO models, the open-set score of GiT models should be $\text{OS} = 1 - \tanh(z_{EG})$ (see Figure 5.3 (c)). Table 5.1 lists the networks and the geometry score in the GiT model.

5.3.3 Model Training

In the baseline model, as shown in Figure 5.2 (a), the Euclidean classifier is optimized by a standard cross-entropy loss function, as $\ell = \ell_E(\mathbf{c}_E)$. Similarly, we optimize the GiO model using the confidence vector $\mathbf{c}_G$, predicted in its geometric classifier with its own specific loss $\ell = \ell_G(\mathbf{c}_G)$ (see Figure 5.2 (b)). The loss functions for the spherical and hyperbolic geometric networks are described in Equations (2.3) and (2.8).

The GiT model, as shown in Figure 5.2 (c), is trained in a multi-task learning manner by optimizing a Euclidean classifier and a geometric classifier, as $\ell = \ell_E(\mathbf{c}_E) + \ell_G(\mathbf{c}_G)$. To be more specific, a shared image encoder encodes the input image in a Euclidean space $\mathcal{M}_E$ and the curved spaces $\mathcal{M}_G$. Then, the following Euclidean classifier and geometric classifier are optimized separately. In contrast to the well-studied student-teacher models Bergmann et al. [2020]; Salehi et al. [2021], which aim to transfer the knowledge from the teacher model to the student model, our GiT model learns sub-branches guided by its own spaces and objective functions (see Equations (2.3) and (2.8)).

5.4 Experiments

In this section, we evaluate our models on four visual open-set recognition tasks: (1) multi-class OOD detection (in Tables 5.3, 5.4 and 5.5), (2) one-class open-set detection (in Table 5.6), (3) multi-class open-set segmentation (in Table 5.7) and (4) one-class anomaly segmentation (in Tables 5.8 and 5.9). Table 5.2 shows the difference between each task used in this chapter. For simplicity, we use the following abbreviations for our models: **SiO** (Spherical-in-One), **SiT** (Spherical-in-Two), **HiO** (Hyperbolic-in-One), **HiT** (Hyperbolic-in-Two), **MiO** (Mixed-in-One) and **MiT** (Mixed-in-Two). It is notable that such terms (*e.g.*, SiO and SiT) indicate the curved geometric models apply the geometric classifiers compared to their corresponding baselines. See Table 5.1 for more details. Metrics including the false positive rate at 95% true positive rate (FPR at 95% TPR), the detection error, the area under the receiver operating characteristics (AUROC) Davis and Goadrich [2006]; Fawcett [2006], and the area under the precision-recall (AUPR) Manning and Schutze [1999]; Saito and Rehmsmeier [2015] are measured. All results are averaged over five independent trials.

Table 5.2: **Visual open-set recognition tasks** where we evaluate the proposed geometric models.

Task	Recognition Space	Num. Known or ID Class	Num. Unknown or OOD Class
Multi-class OOD Detection	Image	Multiple	Multiple
One-class Open-set Detection	Image	One	Multiple
Multi-class Open-set Segmentation	Pixel	Multiple	Multiple
One-class Anomaly Segmentation	Pixel	One	Multiple

For the experiments on multi-class OOD detection, we train the geometric models for 300, 300, and 100 epochs using Hendrycks&Gimpel Hendrycks and Gimpel [2017], ODIN Liang et al. [2018], and ELOC Vyas et al. [2018] as the baselines, with a learning rate of 0.05/0.05/0.1. In the experiments of one-class open-set detection, the models based on RotNet Gidaris et al. [2018]/CSI Tack et al. [2020] are trained for 100/1000 epochs with learning rates of 0.01/0.1. For the multi-class segmentation task, the models are trained for 30 epochs with a learning rate of 0.02. In one-class anomaly segmentation, the models are trained for 1200 epochs with a learning rate of $1e-4$. The values of curvatures for the GiO/GiT models are provided in the captions of the respective tables.

5.4.1 Multi-Class OOD Detection

The objective of multi-class OOD detection, traditionally termed OOD detection, is to identify whether a sample is from the given dataset with multiple ID classes. The model is trained on the ID dataset only. In this setting, CIFAR-10 and CIFAR-100 Krizhevsky et al. [2009] are chosen as ID datasets while the cropped TinyImageNet (TiNc), the resized TinyImageNet (TiNr) Deng et al. [2009], the cropped LSUN (LSUNc), the resized LSUN (LSUNr) Yu et al. [2015] and iSUN Xu et al. [2015] are OOD datasets. We first adopt the Hendrycks&Gimpel model Hendrycks and Gimpel [2017] as the baseline network. Both Dense-BC Huang et al. [2017] and WRN-

Table 5.3: **Multi-class OOD detection** on CIFAR-10/CIFAR-100 Krizhevsky et al. [2009] with DenseNet/WRN-28-10 image encoder. Mixed-geometry embedding in the MiO/MiT model includes spherical and hyperbolic embedding. We provide averaging results over 5 OOD datasets: TINC, TINr, LSUNC, LSUNr, and iSUN. “*” indicates results are obtained via a self-implemented network. The curvatures of SiO, SiT, HiO, HiT, MiO, and MiT are set to 1.0, 1.0, -0.01 , -1.0 , $(1.0, -0.01)$, $(1.0, -1.0)$.

Image encoder (on ID dataset)	Method	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑
Dense-BC (on CIFAR-10)	Hendrycks&Gimpel* Hendrycks and Gimpel [2017]	36.9	10.9	94.8
	SiO/HiO/MiO	37.8/6.3/8.3	11.0/5.4/6.2	94.7/98.7/98.5
	SiT/HiT/MiT	10.5/20.8/11.2	6.7/10.0/7.0	98.1/96.0/98.0
Dense-BC (on CIFAR-100)	Hendrycks&Gimpel* Hendrycks and Gimpel [2017]	70.7	26.3	81.5
	SiO/HiO/MiO	69.4/43.0/52.4	26.1/17.1/21.5	81.4/90.7/86.8
	SiT/HiT/MiT	52.6/55.9/66.4	20.3/21.8/25.7	87.5/85.9/81.7
WRN-28-10 (on CIFAR-10)	Hendrycks&Gimpel* Hendrycks and Gimpel [2017]	41.9	12.5	93.5
	SiO/HiO/MiO	43.7/16.4/17.9	12.7/8.5/9.3	93.4/97.1/96.9
	SiT/HiT/MiT	15.5/28.7/20.1	8.1/12.5/9.8	97.1/94.1/96.3
WRN-28-10 (on CIFAR-100)	Hendrycks&Gimpel* Hendrycks and Gimpel [2017]	70.8	26.4	81.5
	SiO/HiO/MiO	69.9/51.1/49.2	26.3/20.3/19.2	81.5/88.1/88.8
	SiT/HiT/MiT	49.3/56.4/57.1	20.0/21.7/21.8	88.2/85.7/86.4

28-10 Zagoruyko and Komodakis [2016] are used as image encoders. As shown in Table 5.3, we can observe that the HiO model attains the overall best accuracy. In addition, our models, except the SiO model, bring the performance gain over the baselines, showing the superiority of curved geometries as embedding spaces. It is also notable that our models are light. For example, the HiT model surpasses the baseline by 4.2% with WRN-28-10 on CIFAR-100, while it only uses an extra 0.02M parameters, *i.e.*, from 146.05M to 146.07M. Moreover, in most cases, the performance of the mixed-curvature model, MiO or MiT, is in between that of hyperbolic and spherical models. Besides the Hendrycks&Gimpel model, we also use ODIN Liang et al. [2018] as the baseline that employs the input pre-processing at the test phase. The results of geometric models that adopt ODIN as the baseline are reported in Table 5.4. From Table 5.3, we identify HiO as the model that obtains the best performance. Hence, we choose and test HiO for ODIN. Except for the experiment of WRN-28-10 on CIFAR-100, we see that HiO boosts the performance against the baseline.

In addition to Hendrycks&Gimpel and ODIN models, we also incorporate the proposed geometric classifier into advanced baselines. In this study, we employ the ELOC Vyas et al. [2018] as the baseline network. As shown in Table 5.5, the HiT model performs the best over two image encoders. Specifically, it surpasses the baseline by 0.77%/1.60% in AUROC under Dense-BC/WRN-28-10. Similar to results shown in Table 5.3, the performance of the MiT model is in between SiT and HiT with Dense-BC on CIFAR-100, but it unexpectedly becomes the worst with WRN-28-10.

5.4.2 One-Class Open-set Detection

In the one-class open-set (or anomaly) detection setting, only one class is set as the known class, while others are used as unknown classes. The common practice of

Table 5.4: **Multi-class OOD detection** on CIFAR-10/CIFAR-100 Krizhevsky et al. [2009] with DenseNet/WRN-28-10 image encoder. We provide averaging results over 5 OOD datasets: TINC, TINr, LSUNC, LSUNr, and iSUN. “*” indicates results are obtained via self-implemented networks. The curvature of HiO is set to -0.01 .

Image encoder (on ID dataset)	Method	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑
Dense-BC (on CIFAR-10)	ODIN* Liang et al. [2018]	11.2	6.8	98.0
	HiO	6.0	5.2	98.8
Dense-BC (on CIFAR-100)	ODIN* Liang et al. [2018]	48.0	17.3	90.3
	HiO	40.9	16.5	91.1
WRN-28-10 (on CIFAR-10)	ODIN* Liang et al. [2018]	22.9	10.1	95.8
	HiO	15.5	8.3	97.3
WRN-28-10 (on CIFAR-100)	ODIN* Liang et al. [2018]	52.2	19.2	88.8
	HiO	49.5	20.0	88.3

Table 5.5: **Multi-class OOD detection** on CIFAR-100 Krizhevsky et al. [2009] with DenseNet Huang et al. [2017] and WRN-28-10 Zagoruyko and Komodakis [2016] image encoders. Mixed-geometry embedding in the MiT model combines a spherical embedding and hyperbolic embedding. We provide averaging results over 4 multiple OOD datasets: TINC, TINr, LSUNC, and LSUNr. The curvatures of SiT, HiT, and MiT are set to 1.0, -0.001 , and $(1.0, -0.001)$.

Image encoder (on ID dataset)	Method	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑
Dense-BC (on CIFAR-100)	ELOC Vyas et al. [2018]	14.93	8.37	97.28
	SiT/HiT/MiT	17.84/10.45/12.84	9.20/6.78/7.75	96.72/98.05/97.63
WRN-28-10 (on CIFAR-100)	ELOC Vyas et al. [2018]	21.40	10.48	95.87
	SiT/HiT/MiT	16.91/13.63/17.73	8.75/7.87/9.10	97.01/97.47/96.78

creating discriminative representations under this setting is modeled as a multi-class classification problem, using the self-supervised learning (SSL) algorithms Gidaris et al. [2018]; Ruff et al. [2018]; Golan and El-Yaniv [2018]; Hendrycks et al. [2019]; Tack et al. [2020]. In this task, we evaluate our models on the one-class CIFAR-10 dataset Krizhevsky et al. [2009].

Our geometric classifier is built on RotNet Gidaris et al. [2018] and CSI Tack et al. [2020]. RotNet predicts the rotation angles as supervision for SSL. Following the setting in Gidaris et al. [2018], we set the rotation angles to 0° , 90° , 180° and 270° . A 4-dimensional classifier that predicts the rotation angle is applied to the input image. CSI adopts the contrastive learning scheme, which contrasts the negative samples from the data augmentation. The results are shown in Table 5.6. We can observe that either of our models can improve the baselines, showing that embedding in curved spaces benefits the discrimination of data embedding. For example, in RotNet, the method with mixed-curvature geometry, MiT, attains the best performance improvement, *e.g.*, 1.53%, and outperforms the SiT and HiT models. It verifies that mixed-curvature geometry indeed benefits from the advantages of both spherical

Table 5.6: **One-class open-set detection** on CIFAR-10 Krizhevsky et al. [2009]. Image-level AUROC in % is given. **class_{*i*}** indicates the *i*th class. “Extra” indicates utilizing extra data for training (*e.g.*, using pre-trained models on ImageNet Deng et al. [2009]). “*” indicates results are obtained via a self-implemented network. Mixed-geometry embedding in the MiT model incorporates both the spherical embedding and hyperbolic embedding. RotNet*, SiT, HiT, and MiT adopt WRN-28-10 Zagoruyko and Komodakis [2016] as image encoder while CSI, SiT, HiT, and MiT utilize ResNet-18 He et al. [2016]. The curvatures of SiT, HiT and MiT based on RotNet* are set to 1.0, -0.005 and $(1.0, -0.005)$. The curvatures of SiT, HiT and MiT based on CSI are 1.0, -0.01 , and $(1.0, -0.01)$.

Method	Extra	class ₀	class ₁	class ₂	class ₃	class ₄	class ₅	class ₆	class ₇	class ₈	class ₉	Avg
RotNet* Gidaris et al. [2018]	✗	72.70	94.25	76.38	69.26	79.30	80.97	76.18	92.78	90.62	88.67	82.11
SiT	✗	73.77	95.21	78.39	70.46	80.92	81.75	77.80	93.25	91.27	89.40	83.22
HiT	✗	72.68	94.54	78.31	69.62	80.80	81.63	77.05	93.06	91.06	88.90	82.77
MiT	✗	73.60	95.18	79.82	70.80	81.76	82.57	78.52	93.52	91.16	89.46	83.64
CSI Tack et al. [2020]	✗	89.9	99.1	93.1	86.4	93.9	93.2	95.1	98.7	97.9	95.5	94.3
SiT	✗	89.26	99.17	94.16	87.76	94.53	93.47	95.44	98.85	97.91	96.03	94.66
HiT	✗	89.40	99.17	94.22	88.06	94.33	93.48	95.58	98.76	97.85	95.93	94.68
MiT	✗	89.46	99.16	94.30	88.04	94.41	93.32	95.58	98.87	97.94	95.99	94.71

geometry and hyperbolic geometry. In CSI Tack et al. [2020], as a strong baseline, our models again bring a performance gain, and the MiT method achieves the best average performance, revealing that our models generalize and are effective.

5.4.3 Multi-Class Open-set Segmentation

In contrast to multi-class OOD detection, which recognizes the OOD samples at the image level, multi-class open-set (or anomaly) segmentation is required to predict unknown objects at the pixel level. Following Hendrycks et al. [2022]; Xia et al. [2020], we evaluate this task using the StreetHazards dataset Hendrycks et al. [2022]. The Hendrycks&Gimpel model Hendrycks and Gimpel [2017] with Maximum Softmax Probability (MSP) is adopted as the baseline. We report the results in Table 5.7. As suggested in Table 5.7, this task also benefits the most from the mixed-curvature geometry (MiT), again showing that multiple geometries are essential to learning discriminative embeddings.

5.4.4 One-Class Anomaly Segmentation

One-class anomaly segmentation, also known as anomaly localization, aims to identify whether the input pixel is an anomalous pixel or not Bergmann et al. [2019]; Liu et al. [2020c]. In contrast to the multi-class segmentation, the training samples in one-class anomaly segmentation are drawn from only one dataset class.

We verify the effectiveness of our models on the MVTecAD dataset Bergmann et al. [2019] and adopt the excellent model Patch-SVDD Yi and Yoon [2020] as a baseline without using extra data. A possible limitation of Patch-SVDD is that the

Table 5.7: **Multi-class open-set segmentation** on StreetHazards Hendrycks et al. [2022]. FPR (95% TPR), pixel-level AUROC, and AUPR in % are given. The metric std over 5 runs. The results of AE, Dropout, and MSP are provided in Hendrycks et al. [2022]. The curvatures of SiT, HiT, and MiT based on MSP are set to 1.0, -0.01 , and $(1.0, -0.01)$.

Method	FPR (95% TPR) ↓	AUROC ↑	AUPR ↑
AE Baur et al. [2018]	91.7	66.1	2.2
Dropout Gal and Ghahramani [2016]	79.4	69.9	7.5
MSP Hendrycks and Gimpel [2017]	33.7	87.7	6.6
SynthCP Xia et al. [2020]	28.4	88.5	9.3
MaxLogit Hendrycks et al. [2022]	26.5	89.3	10.6
MSP Hendrycks and Gimpel [2017]	33.7	87.7	6.6
SiT	29.9	86.4	4.6
HiT	27.7	88.9	6.3
MiT	27.1	89.5	8.6

computation of the open-set scores OS for inference depends to a great extent on the comparison with training samples. To simplify the evaluation process, we calculate the open-set score directly from its normalized classifier’s value without utilizing training data (denoted by Patch-SVDD[†]). We then plug our geometric models on top of Patch-SVDD[†]. The results are reported in Table 5.8. As suggested by Table 5.8, all geometric models, except the SiO model, boost the performance of the baseline, and the mixed-curvature geometric model, MiT, performs the best. It gains 17.55% improvement. In this task, the dual-branch architecture (*i.e.*, SiT/HiT/MiT) consistently outperforms the single-branch model (*i.e.*, SiO/HiO/MiO). Along with a considerable improvement, our proposal is also cheap. For example, the SiT improves the baseline by a margin of 16.41%, while it only brings extra 0.1M parameters, *i.e.*, 1.82M vs. 1.72M, again showing the benefits from curved geometric embeddings.

The idea of Patch-SVDD[†] is to evaluate our method on a toy example to illustrate the advantage of curved geometries. However, after we remove the comparison process with training images from Patch-SVDD, we find its identification performance significantly drops. To show the full potential of our design in conjunction with the original Patch-SVDD, we employ our geometric model over the original Patch-SVDD where training images are considered for calculating the open-set score. As shown in Table 5.9, the accuracy of Patch-SVDD on MVTecAD is boosted from 95.7% to 96.5%/96.5%/96.7% once using SiT/HiT/MiT. We follow the open-set score computation of the original Patch-SVDD except for replacing the patch’s embedding with the geometric score z_{EG} .

As described in Section V.C, the multi-class setting allows training data from multiple classes. We utilize this multi-class data to train PatchSVDD[†] and the proposed geometric models on the MVTecAD dataset Bergmann et al. [2019]. Consistent with the approach outlined in You et al. [2022], we do not employ class label informa-

Table 5.8: **One-class anomaly segmentation (anomaly localization)** on MVTecAD. Pixel-level AUROC in % is given. “Extra” indicates utilizing extra data for training (e.g., using pre-trained models on ImageNetDeng et al. [2009]). “Cpr” indicates using the comparison with the training data for open-set score computation during evaluation. The results of AnoGAN/AE and AVID are provided in Salehi et al. [2021] and Venkataramanan et al. [2020], respectively. Mixed-geometry embedding in the MiO/MiT model includes spherical and hyperbolic embedding. The curvatures of SiO, HiO, MiO, SiT, HiT, and MiT are set to 1.0, -0.01 , $(1.0, -0.01)$, 1.0, -1.0 and $(1.0, -1.0)$. The model sizes of Patch-SVDD[†], SiO, HiO, MiO, SiT, HiT, and MiT are 1.72M, 1.72M, 1.72M, 1.82M, 1.82M, 1.83M and 1.93M.

Method	Extra	Cpr	Bottle	Hazelnut	Capsule	Metal Nut	Leather	Pill	Wood	Carpet	Tile	Grid	Cable	Transistor	Toothbrush	Screw	Zipper	Avg
AnoGAN Schlegl et al. [2017]	X	X	86	87	84	76	64	87	62	54	50	58	78	80	90	80	78	74
AVID Sabokrou et al. [2018]	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	77
AE-L2Bergmann et al. [2018]	X	X	86	95	88	86	75	85	73	59	51	90	86	86	93	96	77	82
AE-SSIMBergmann et al. [2018]	X	X	93	97	94	89	78	91	73	87	59	94	82	90	92	96	88	87
VAE-VE Liu et al. [2020c]	X	X	87	98	74	94	95	83	77	78	80	73	90	93	94	97	78	86
VAE-grad Dehaene et al. [2020]	X	X	92.2	97.6	91.7	90.7	92.5	93	83.8	73.5	65.4	96.1	91.0	91.9	98.5	94.5	86.9	89.3
Patch-SVDD [†] Yi and Yoon [2020]	X	X	62.78	71.03	80.03	68.28	90.03	67.15	65.45	82.60	89.28	82.55	67.60	50.35	88.20	85.95	67.23	74.57
SiO	X	X	62.80	73.53	80.50	68.40	89.70	70.13	65.50	74.10	85.90	82.10	68.35	51.10	86.57	50.00	68.35	71.80
HiO	X	X	68.27	76.43	89.00	50.00	93.65	73.20	79.90	86.40	86.65	78.50	87.45	71.50	87.03	78.65	90.60	79.82
MiO	X	X	68.50	79.73	92.03	65.28	94.80	74.83	79.00	86.20	70.20	66.17	61.93	64.47	87.43	71.10	88.08	76.65
SiT	X	X	95.43	94.60	90.80	96.07	96.30	92.20	82.93	89.20	92.55	89.13	93.67	92.40	93.30	83.40	91.68	91.58
HiT	X	X	95.80	93.60	92.87	88.40	97.03	92.90	81.10	83.13	87.03	92.03	95.20	95.27	94.57	81.63	94.20	90.98
MiT	X	X	96.33	96.05	92.77	96.77	96.63	93.50	82.30	84.30	93.63	93.70	94.93	94.40	95.03	77.33	94.13	92.12

tion. The results, presented in Table 5.10, demonstrate the clear superiority of the geometric approaches over the baseline model.

5.4.5 Abalation Study

Interactions among geometries. The GiT model requires meanwhile learning two embeddings (e.g., a Euclidean $\mathbf{e}_E$, and a hyperbolic $\mathbf{e}_H$ or spherical embedding $\mathbf{e}_S$). We observe that it happens in the interactions between different geometric components. For instance, in multi-class OOD detection (see Table 5.3), $\mathbf{e}_E$ could enrich $\mathbf{e}_S$ (SiO *vs.* SiT), however, for $\mathbf{e}_H$, it has less or even negative impact (HiO *vs.* HiT). To further understand the influence, we analyzed the experiments of WRN-28-10 on CIFAR-10 where we separately test $\mathbf{e}_E$ and $\mathbf{e}_H$ (or $\mathbf{e}_S$) in HiT (or SiT). Results in Table 5.11 suggests the aforementioned point (HiO *vs.* $\mathbf{e}_H$ in HiT and SiO *vs.* $\mathbf{e}_S$ in SiT).

Curvature. The curvature κ is the only hyper-parameter in the proposed curvature-aware geometric networks. The study of one-class anomaly segmentation of HiT based on Patch-SVDD[†] in Table 5.12 suggests that κ clearly impacts open-set recognition performance. The table demonstrates that for each category, there exists an optimal curvature value κ that leads to the best performance. Deviating from this optimal value results in a decrease in performance.

5.4.6 Discussion

Performance. We learn from empirical observations in Subsections 5.4.1, 5.4.2, 5.4.3 and 5.4.4 that the curved spaces can consistently provide reliable information for

Table 5.9: **One-class anomaly segmentation (anomaly localization)** on MVTecAD. Pixel-level AUROC in % is given. “Extra” indicates utilizing extra-data for training (*e.g.*, using pre-trained models on ImageNetDeng et al. [2009]). “Cpr” indicates using the comparison with the training data for open-set score computation during evaluation. The results of LSA are provided in Venkataramanan et al. [2020]. Mixed-geometry embedding in the MiT model includes spherical embedding and hyperbolic embedding. The curvatures of SiT, HiT, and MiT are set to 1.0, -1.0 , and $(1.0, -1.0)$.

Method	Extra	Cpr	Bottle	Hazelnut	Capsule	Metal Nut	Leather	Pill	Wood	Carpet	Tile	Grid	Cable	Transistor	Toothbrush	Screw	Zipper	Avg
LSAAbati et al. [2019]	✓	✗	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	81
CAVGA-Rw Venkataramanan et al. [2020]	✓	✗	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	90
MR-KD Salehi et al. [2021]	✓	✗	96.32	94.62	95.86	86.38	98.05	89.63	84.80	95.64	82.77	91.78	82.40	76.45	96.12	95.96	93.90	90.71
Glancing-at-Patch Wang et al. [2021a]	✓	✗	93	84	90	91	90	93	81	96	80	78	94	100	96	96	99	91
PANDA-OE Reiss et al. [2021]	✓	✗	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	96.2
CutPaste Li et al. [2021]	✗	✓	97.6	97.3	97.4	93.1	99.5	95.7	95.5	98.3	90.5	97.5	90.0	93.0	98.1	96.7	99.3	96.0
Patch-SVDD Yi and Yoon [2020]	✗	✓	98.1	97.5	95.8	98.0	97.4	95.1	90.8	92.6	91.4	96.2	96.8	97.0	98.1	95.7	95.1	95.7
SiT	✗	✓	98.10	98.93	97.00	98.00	97.20	94.83	92.30	91.20	96.30	97.43	96.30	97.83	96.40	97.53	98.45	96.52
HiT	✗	✓	98.23	96.90	96.83	97.60	97.20	94.20	92.73	94.13	96.50	97.65	95.75	97.60	95.95	97.37	98.37	96.47
MiT	✗	✓	98.23	99.00	96.33	98.80	97.20	96.03	92.43	91.70	96.60	97.50	96.80	97.63	96.65	97.73	98.45	96.74

Table 5.10: **Multi-class anomaly segmentation** on MVTecAD Bergmann et al. [2019]. Pixel-level AUROC in % is given. Mixed-geometry embedding in the MiT model includes spherical embedding and hyperbolic embedding. The curvatures of SiT, HiT, and MiT are set to 1.0, -1.0 , and $(1.0, -1.0)$.

Method	Bottle	Hazelnut	Capsule	Metal Nut	Leather	Pill	Wood	Carpet	Tile	Grid	Cable	Transistor	Toothbrush	Screw	Zipper	Avg
Patch-SVDD [†] Yi and Yoon [2020]	67.8	77.1	89.1	79.1	62.8	75.9	93.6	61.9	80.2	84.3	84.7	82.6	74.3	88.2	72.2	78.3
SiT	84.5	92.6	89.8	86.8	97.9	83.0	88.7	88.0	88.7	70.2	77.9	72.3	81.6	77.1	91.3	84.7
HiT	85.1	92.9	88.9	85.0	97.2	85.2	86.6	91.3	91.6	76.6	74.1	84.1	81.2	63.8	91.1	85.0
MiT	83.5	90.7	89.2	88.5	98.0	87.3	86.1	84.6	92.6	70.8	76.3	74.6	82.7	63.8	91.3	84.0

open-set tasks. In most cases, our curvature-aware geometric networks clearly outperform Euclidean networks. One possible explanation is that the geometric representations benefit particular problems. For example, as discussed in Section 5.1, hyperbolic geometry is good at encoding hierarchical structures inside the data. The datasets we test include hierarchical information to some extent. For instance, the CIFAR dataset includes 10 super-classes and 10 sub-classes under each super-class. The 15 classes of MVTecAD can be categorized into two main super-categories, “object” and “texture.” Hence, the hyperbolic space may take effect.

Mixed geometry. Another interesting fact has been observed is that the mixed-curvature geometry beats its component single-curvature geometries in several cases: MiT based on RotNet*/CSI of one-class open-set detection, MiT based on MSP of multi-class open-set segmentation and MiT based on Patch-SVDD[†] of one-class anomaly segmentation in Table 5.6, 5.7 and 5.8, respectively. In some cases, the mixed-curvature geometry has a balanced performance. For example, in the task of multi-class OOD detection, there exists a significant performance gap between hyperbolic and spherical geometries, as evidenced by SiO *vs.* HiO. Thus, the mixed space MiO might be expected to have an average performance.

Method Choice. Our comprehensive empirical study suggests a definitive conclusion cannot be made. This is in line with observations made in recent works. For example, in Skopek et al. [2020], the best geometry choice depends on the task. Our

Table 5.11: **Ablation study: multi-class OOD detection.** Image-level AUROC in % is given. Different cases are evaluated on CIFAR-10 Krizhevsky et al. [2009] to verify the interactions between hyperbolic or spherical and Euclidean spaces.

HiO	HiT	e_E in HiT	e_H in HiT
97.1	94.1	92.1	94.5
SiO	SiT	e_E in SiT	e_S in SiT
93.4	97.1	93.6	96.8

Table 5.12: **Ablation study: one-class anomaly segmentation (anomaly localization).** Pixel-level AUROC in % is given. Different fixed curvatures on MVTecAD Bergmann et al. [2019] are evaluated. For each category, different curvatures bring about different performances. Also, the optimal curvature choice is not consistent among different categories.

Method	κ	Hazelnut	Leather	Wood	Toothbrush
HiT	-1e-5	85.53	96.31	79.24	88.26
HiT	-1e-4	93.77	96.30	84.67	89.90
HiT	-1e-2	95.40	95.77	83.87	88.83
HiT	-1.0	93.60	97.03	81.10	94.57
HiT	-2.0	84.52	96.08	81.79	88.51

study clearly shows that curved geometry is beneficial in capturing the geometry of data, contributing tangibly to identifying unknown objects in data. Specifically, our empirical study suggests that the preferred model for each task is shown below: Multi-class OOD Detection: HiO/HiT; Multi-class Open-set Segmentation: MiT; One-class Open-set Detection: MiT; One-class Anomaly Segmentation: MiT. If one model should be chosen in all instances, we will opt for MiT as the potential model for the visual open-set recognition tasks.

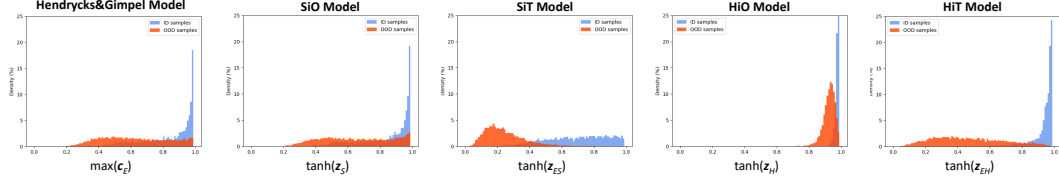


Figure 5.4: **Visualization: multi-class OOD detection.** Density distributions of $\max(\mathbf{c}_E)$, $\tanh(z_S)$, $\tanh(z_{ES})$, $\tanh(z_H)$ and $\tanh(z_{EH})$ of the baseline, SiO, SiT, HiO and HiT models with Dense-BC on CIFAR10→TINc are provided. The obtained AUROC for these five models are 94.8%, 94.7%, 98.1%, 98.7% and 96.0%. More corresponding results can be referred to in Table 5.3.

5.4.7 Visualization

In this subsection, we qualitatively study our method in image-level classification and pixel-level segmentation to understand why our method can bring performance gain over the baseline model.

Image-Level Classification. In this work, we particularly study whether the geometric score z_G and z_{EG} from the curved embedding spaces can provide more useful information than the confidence vector $\mathbf{c}_E$ from the Euclidean space in distinguishing known and unknown objects. We visualize the distribution of $\max(\mathbf{c}_E)$, $\tanh(z_S)$, $\tanh(z_{ES})$, $\tanh(z_H)$ and $\tanh(z_{EH})$ on multi-class OOD detection over CIFAR-10 (CIFAR10→TINc) in Figure 5.4. As shown in Figure 5.4, the curved embedding spaces help to better separate ID and OOD distributions. The distribution of open-set score $OS(z_{EH})$ of one-class open-set detection on one-class CIFAR-10 is visualized in Figure 5.5. The visualization shows z_{EH} provides reliable information for distinguishing the known and unknown classes. We present some examples in Figure 5.6 to verify the performance differences among models, where we compare RotNet*/HiT/MiT on *Class 2* as the known class. The open-set scores OS in the upper figure show different models’ capacities to recognize unknown classes. Also, the AUROC curve suggests substantial improvements led by the curved geometries.

Pixel-Level Segmentation. From experiments, we find that besides image-level tasks, curved spaces as embedding spaces do help in pixel-level tasks. In Figure 5.7, we show examples in which we visualize the open-set score OS of Patch-SVDD[†], SiT, HiT and MiT on MVTecAD Bergmann et al. [2019]. As shown in Figure 5.7, z_{EG} from $\mathcal{M}_G$ outperforms $\mathbf{c}_E$ from $\mathcal{M}_E$ in identifying anomalous pixels.

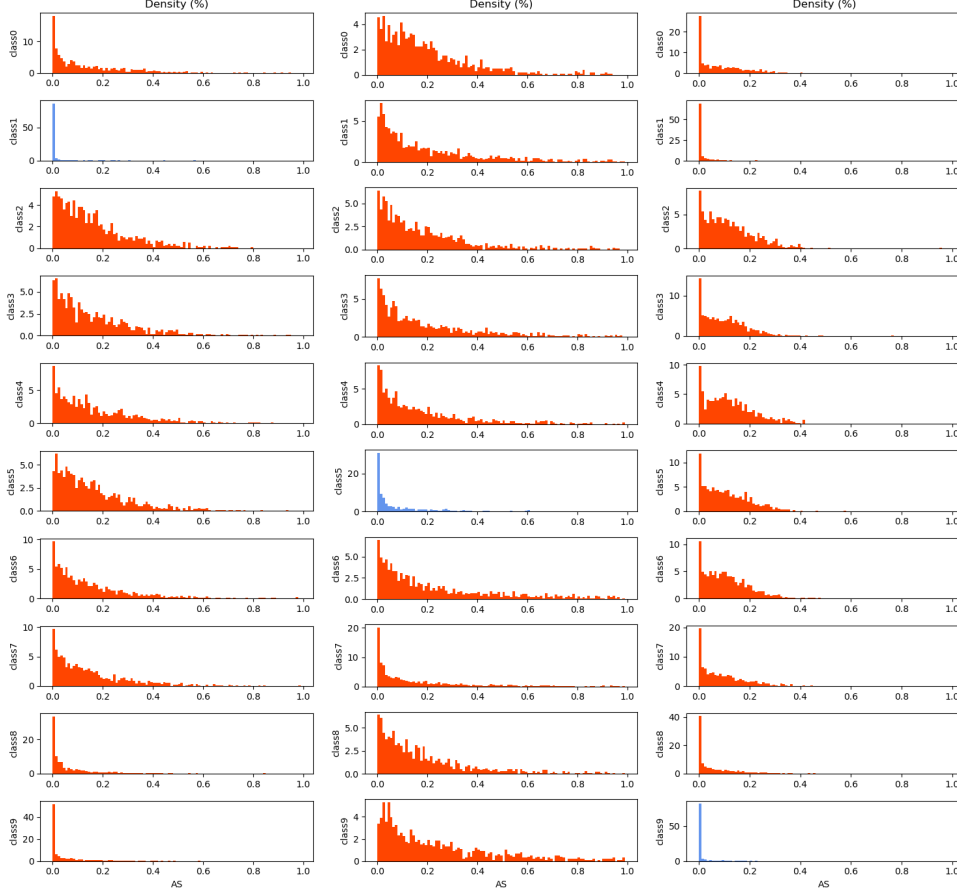


Figure 5.5: **Visualization: one-class open-set detection.** Density distribution of open-set score $OS(z_{EH})$ of HiT based on RotNet* on CIFAR-10 Krizhevsky et al. [2009] is visualized. We choose cases where **class₁**, **class₅** or **class₉** is taken as the normal class. The distribution of one case is shown in one column. More corresponding results can be referred to in Table 5.6.

5.5 Summary

In this chapter, we study the use and benefit of employing curved spaces for the purpose of visual open-set recognition tasks. Our idea is inspired by the observation that curved embedding spaces help better represent “unknown” data in low-shot problems, *e.g.*, the hyperbolic geometry for the audio-visual zero-shot learning in Chapter 4. This chapter proposes two novel geometric networks, GiO and GiT, for the visual open-set recognition tasks. We fully study the potential of different geometry constraints in each geometric model. To our knowledge, our curvature-aware geometric networks were the first attempt to employ curved geometries in open-set recognition by the time of the publication of this work. Based on extensive experiments, we confirm that more distinct feature embeddings between known and un-

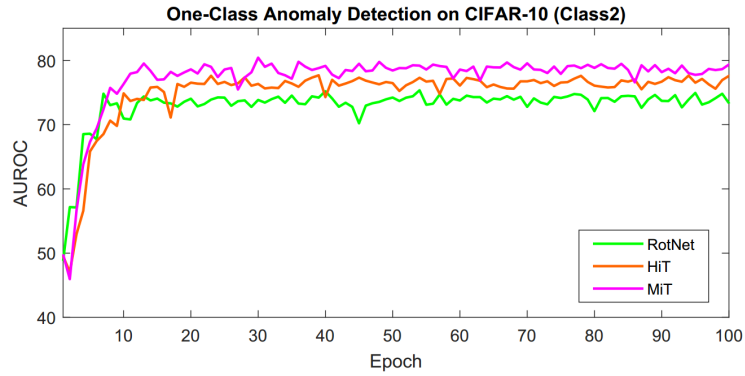
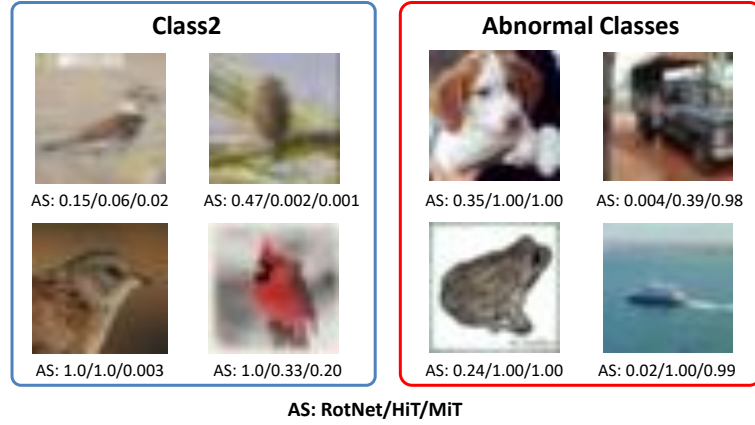


Figure 5.6: **Visualization: one-class open-set detection.** Examples with open-set scores OS computed by RotNet*, HiT, and MiT are provided (see upper figures). In addition, AUROC curves along the training epoch are plotted in the lower figure.

known samples can be learned using curved spaces, clearly showing the benefits of curved geometry.

In examining curved geometric networks for multi-class open-set semantic segmentation, we find a problem where significant portions of the image are occupied by identified unknown areas that remain unprocessed. To tackle this problem, the subsequent chapter will introduce a novel task: generalized open-set semantic segmentation. The newly defined task aims to enhance predictions, identifying more segments within the unknown regions of the image.

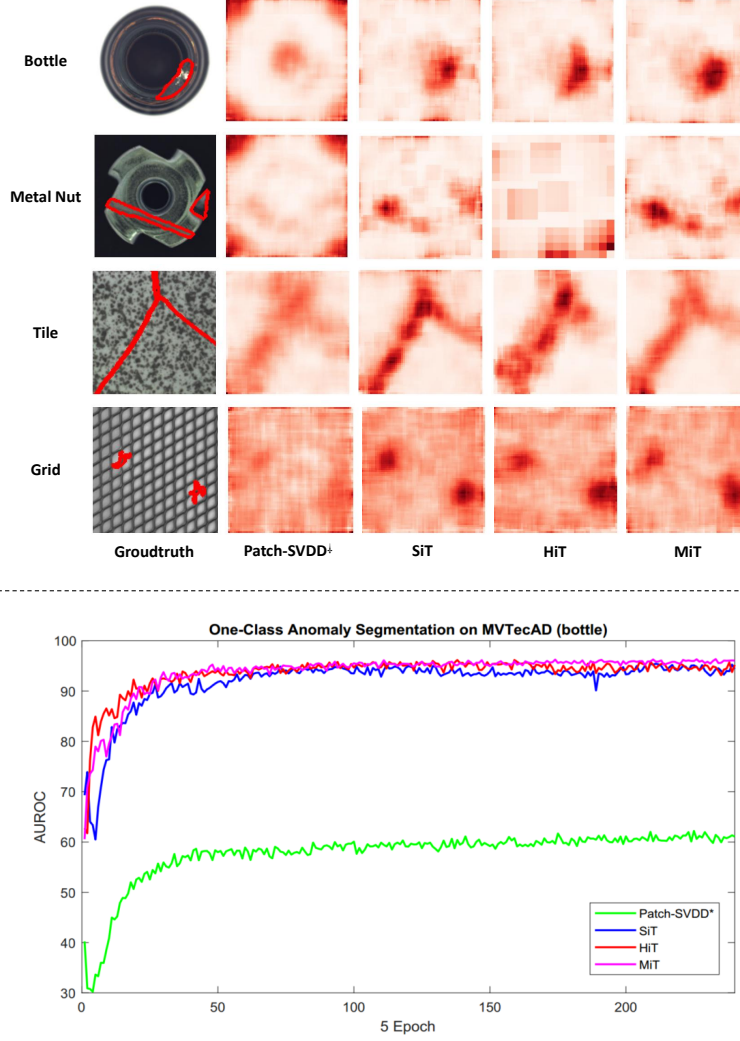


Figure 5.7: **Visualization: one-class anomaly segmentation (anomaly localization).** The heatmaps of open-set score $OS(c_E)$, $OS(z_{ES})$, $OS(z_{EH})$ and $OS(z_{EM})$ of Patch-SVDD⁺, SiT, HiT and MiT on MVTecAD Bergmann et al. [2019] are visualized. We provide examples from “bottle”, “metal nut”, “tile”, and “grid.” More corresponding results can be found in Table 5.8. In addition, for the category “bottle”, AUROC curves of different models along the training epoch are plotted in the lower figure.

Generalized Open-set Semantic Segmentation

In the previous chapter, we explored the application of curved geometry to visual open-set recognition tasks. Upon studying one such task, *i.e.*, open-set semantic segmentation (OSS), we find that a significant portion of the OSS prediction, *i.e.*, identified unknown area, remains unprocessed.

In this chapter, we extend OSS into a new image segmentation task called Generalized open-set semantic segmentation (GOSS). Previously, with OSS, intelligent systems only detect unknown regions without further processing, limiting their perception capacity of the environment. It stands to reason that further analysis of the detected unknown pixels would benefit systems' decision-making. Therefore, we propose GOSS, which holistically unifies the abilities of two well-defined segmentation tasks, *i.e.*, OSS and generic segmentation (GS). Specifically, GOSS classifies pixels as belonging to known classes, and clusters (or groups) of pixels of the unknown class are labeled as such. We also propose a metric that balances the pixel classification and clustering aspects to evaluate this newly expanded task. Moreover, we build benchmark tests on existing datasets and propose neural architectures as baselines. Our experiments on multiple benchmarks demonstrate the effectiveness of our baselines. This chapter is based on our published work Hong et al. [2023c].

6.1 Introduction

Image segmentation has significantly progressed in the deep learning era, especially class-specific semantic segmentation (SS) Long et al. [2015]; Chen et al. [2017a,b, 2018c]; Cheng et al. [2021b]; Zheng et al. [2021]; Cheng et al. [2022]. The goal of the SS task is to predict the class label of each pixel in an image from a set of predefined object classes. Adjacent pixels naturally belong together to form a segment when they share the same object category. Despite the considerable improvement, most SS settings follow a closed-set assumption that training and test data come from the same set of known object classes Shotton et al. [2006]; Cordts et al. [2016]; Zhou et al. [2017]; Caesar et al. [2018]. However, this assumption is rarely the practice case, and it limits the generalization of segmentation models to unknown classes, which

models do not see during training.

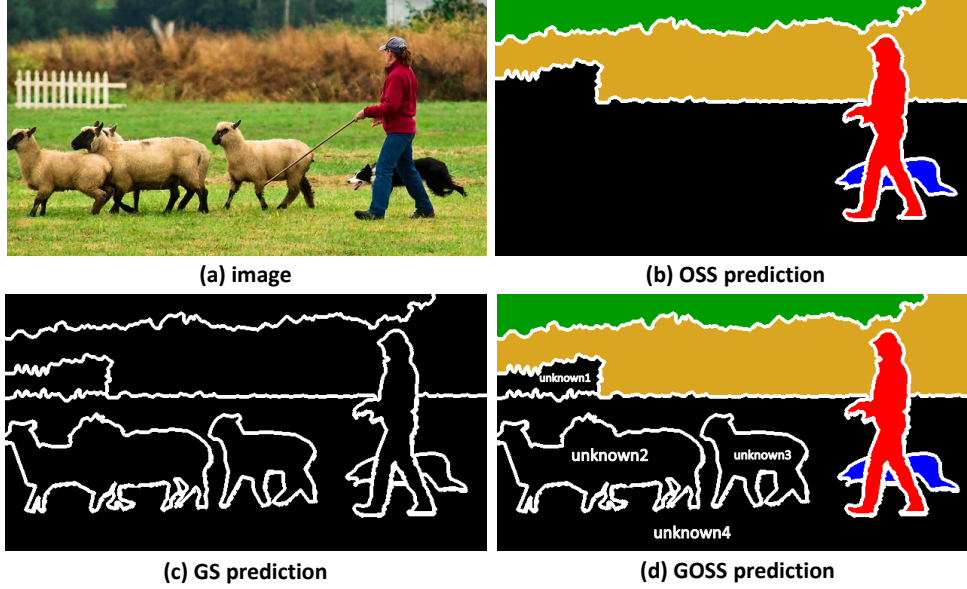


Figure 6.1: Different tasks of image segmentation. For a given input image (a) that contains both known (“person”, “dog” and “vegetation”) and unknown objects (“sheep”, “rail” and “grass”), we show: (b) open-set semantic segmentation (OSS) by pixel identification, (c) generic segmentation (GS) by pixel clustering, and (d) generalized open-set semantic segmentation (GOSS).

Open-set semantic segmentation (OSS) Lis et al. [2019]; Cen et al. [2021a]; Chan et al. [2021a]; Hendrycks et al. [2022]; Bevandić et al. [2022] has recently been proposed to relax the above assumption, which aims to segment an image containing both known and unknown object classes. Unlike SS, OSS identifies the unknown region where pixels belong to unknown classes.

Although OSS aims to identify pixels that do not belong to one of the known classes, it does not provide any further processing or analysis amongst those identified unknown pixels. We argue that such a setting of OSS may limit the broad usage of vision-based intelligent systems when encountering unfamiliar scenes where several unknown object classes are adjacent rather than separated. Consider a scenario where an intelligent system enters a new scene, as shown in Figure 6.1 (a). OSS leaves the whole unknown region as a large segment without further processing (see “black region” in Figure 6.1 (b)). Insufficient information provided by the OSS setting might affect the decision-making of intelligent systems. Hence, we raise fundamental questions: how can we improve OSS to generate richer representations for images in real-world scenes? Moreover, what is a more expressive and versatile image segmentation task beyond OSS? These questions are crucial in real-world applications like autonomous driving and robotics.

Towards the goal of better handling unknown regions in an image, inspired by the perception system of humans that they can jointly recognize the previously known

objects and easily group unknown areas into different segments even though they do not know the categories of those unknown objects, this chapter studies a new type of image segmentation task called Generalized Open-set Semantic Segmentation (GOSS). It aims to classify pixels belonging to known classes and group the unknown pixels (see Figure 6.1 (d) for GOSS prediction). As we can see from Figure 6.1 (d), “unknown rail” (or “unknown 1”) and “unknown sheep” (or “unknown2/3”) are segmented out from “unknown grass” (or “unknown4”). GOSS takes advantage of generic segmentation (GS), which groups pixels into segments sharing similarities Shi and Malik [2000]; Comaniciu and Meer [2002]; Felzenszwalb and Huttenlocher [2004]; Arbelaez et al. [2010]; Isaacs et al. [2020]; Wan et al. [2020]. Compared to OSS, GOSS can detect more “objects” inside unknown regions. We specify two real-world applications where GOSS can help. First, again considering the example in Figure 6.1 (a), self-driving cars may be assisted in avoiding potential obstacles if “unknown sheep” or “unknown rail” inside “unknown grass” can be found in the OSS prediction (see Figure (b)). Another possible practical example is that new detected “objects” inside unknown regions of images could help accelerate the data annotation process, especially when images from unfamiliar scenes are being labeled.

To enable intelligent systems to perform GOSS, we first build benchmarks using existing segmentation datasets, *i.e.*, COCO-Stuff Caesar et al. [2018] and Cityscapes Cordts et al. [2016]. We split the full set of object categories into two sets: known classes and unknown classes. We keep the semantic annotations of known categories. For unknown categories, we use connectivity labeling Wan et al. [2020] to convert their original semantic annotations into clustering ground truths. Along with the available datasets, a valid metric is also required to validate the quality of both OSS and GS. Although many existing metrics for segmentation tasks exist, such metrics are limited to measuring a single setting. In this chapter, we introduce a metric termed GOSS Quality (GQ), which evaluates the segmentation quality of both known and unknown objects. Having the datasets and evaluation metrics at hand, we further establish a trainable framework, namely, GOSS Segmentor (GST). The proposed GST adopts a dual-branch architecture with a shared encoder network. To perform the GOSS task, one branch conducts pixel classification, and the other performs pixel clustering. Moreover, to learn more discriminative embeddings and thus better process unknown objects, we leverage the pixel-wise contrastive learning loss into the training. In summary, our contributions are as follows:

- We present a new image segmentation task called GOSS, which jointly classifies known pixels and groups identified unknown pixels from OSS.
- We propose the GQ metric that measures the quality of both pixel classification and pixel clustering under open-set settings. Moreover, according to the GOSS settings, we build benchmarks by customizing existing datasets.
- We show a simple yet effective baseline and its extended version, GST, to facilitate future research.

6.2 Related Work

Image segmentation is one of the most widely explored tasks in computer vision. Throughout image segmentation research, novel segmentation tasks have driven research directions and innovations. We compare our new setting and relevant older tasks in Table 6.1.

Table 6.1: Comparisons of different image segmentation tasks. Compared to traditional segmentation tasks, GOSS takes better care of unknown objects.

Task	Known classes	Unknown classes
Generic Segmentation	Cluster	Cluster
Open-set Semantic Segmentation	Classify	Identify
Generalized Open-set Semantic Segmentation	Classify	Identify & Cluster

Open-set Semantic Segmentation (OSS). OSS, which is capable of identifying unknown objects, has developed significantly recently. Performing OSS is essential for intelligent systems as they work in open-set settings where many objects are never seen. A natural solution studied in Hendrycks et al. [2022]; Jung et al. [2021]; Chan et al. [2021b] determines the unknown regions by directly computing the anomaly score from logit or confidence vectors provided by the model classifier. Alternatively, synthesis approaches Lis et al. [2019]; Xia et al. [2020]; Di Biase et al. [2021]; Vojir et al. [2021]; Kong and Ramanan [2021] are proposed to detect unexpected anomalies from reconstructed images. In addition, the work Cen et al. [2021a] employs metric learning to learn more discriminative features and incrementally label novel classes using a human-in-the-loop approach. Beyond the existing OSS setting, the proposed GOSS performs holistic segmentation via classifying the known objects and clustering the unknown objects, providing more expressive information about the environment than OSS. With richer information, GOSS could benefit the practical usage in real-world scenarios.

Generic Segmentation (GS). The task of GS is to find groups of pixels that “go together” Szeliski [2022]. In the early days of computer vision, the term “image segmentation” and the bottom-up general (non-semantic) segmentation share the same meaning. Recently, it is often called “generic segmentation” Arbelaez et al. [2010]; Isaacs et al. [2020]; Wan et al. [2020] to distinguish it from other segmentation tasks. The pipeline of early segmentation methods consists of first extracting local pixel features such as brightness, color, or texture and then clustering these features, *e.g.*, mean shift Comaniciu and Meer [2002], normalized cuts Shi and Malik [2000], random walks Grady [2006], graph-based representations Felzenszwalb and Huttenlocher [2004], or oriented watershed transform Arbelaez et al. [2010]. Learning-based image segmentation methods have now also become popular. DEL Liu et al. [2018a] learns a feature embedding corresponding to a similarity measure between two adjacent superpixels. Saacs et al. Isaacs et al. [2020] propose pixel-wise representations that reflect how segments are related. Super-BPD Wan et al. [2020] learns super

boundary-to-pixel direction to provide a direction similarity between adjacent pixels. Comparing the performance of different image segmentation algorithms, public datasets such as BSD3 Martin et al. [2001]; Arbelaez et al. [2010] provide human-labeled class-agnostic ground truth. However, they do not provide any semantic information.

Image Segmentation as a subtask. Image segmentation is often taken as a subtask jointly solved with other vision problems in a single framework Eigen and Fergus [2015]; Jafari et al. [2017]; Kokkinos [2017], Bleyer et al. [2011]; Yamaguchi et al. [2014]; Sun et al. [2012]; Sevilla-Lara et al. [2016]; Hane et al. [2013]; Hu et al. [2019]. Panoptic segmentation Kirillov et al. [2019b,a]; Hwang et al. [2021] has recently become a standard image segmentation task by unifying semantic and instance segmentation.

6.3 Format and Metric

6.3.1 Task Format

Here, the task format for GOSS is formulated at a pixel level. For the i th pixel of an image, the GOSS output is defined as a pair $\mathbf{goss}_i = (s_i, g_i)$, where the classification label s_i indicates the pixel's semantic class and the clustering (or grouping) label g_i represents the cluster id. Suppose that there are N known semantic classes $\mathbf{L}^{kn} \in \mathbb{R}^N$ and an unknown class indicator $L^{uk} \in \mathbb{R}$, we have the semantic label set $\mathbf{L} = \{\mathbf{L}^{kn}, L^{uk}\}$ which is encoded by $\mathbf{L} := \{0, \dots, N-1, N\}$. Each pixel can be predicted in our formulation as one of the classes. In the first case, each pixel must have a semantic label, while the cluster id is unnecessary. This is because once the i th pixel is labeled with $s_i \in \mathbf{L}^{kn}$, its corresponding cluster id g_i is invalid (which is denoted by *void*). When the pixel is predicted as the unknown class, it can be clustered to g_i . Hence, the i th pixel with known classes (or unknown classes) can be assigned with $\mathbf{goss}_i = (s_i, \text{void})$ (or $\mathbf{goss}_i = (N, g_i)$). In practice, a classification model can predict s_i , and g_i can be determined after the unknown pixels are clustered.

6.3.2 Evaluation Metrics

Appropriate evaluation metrics are fundamental in popularizing a new image segmentation task Kirillov et al. [2019b]; Cheng et al. [2021a]; Everingham et al. [2010]. In this subsection, we briefly review some popular existing metrics for relevant segmentation tasks and then introduce a metric tailored for the proposed GOSS.

Previous Metrics. Standard metrics for OSS include the false positive rate at 95% true positive rate (FPR at 95% TPR), the area under the receiver operating characteristics (AUROC) Davis and Goadrich [2006]; Fawcett [2006] and the area under the precision-recall (AUPR) Manning and Schutze [1999]; Saito and Rehmsmeier [2015]. Such metrics assess performance based on the overlap of anomaly score distributions between the known and unknown classes. However, they are not suited for evaluating GOSS since they do not need to classify the input as a known or unknown class, as GOSS requires each input pixel to be explicitly classified as belonging to a known

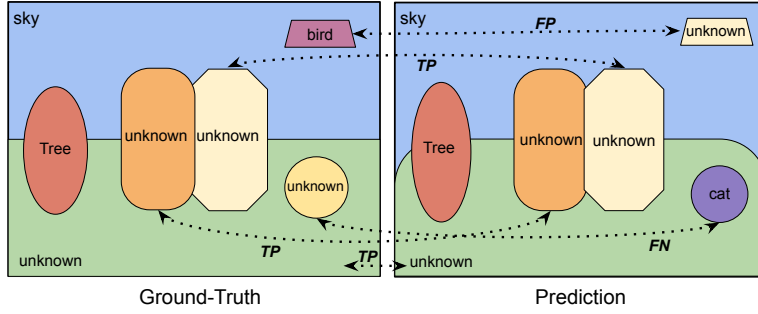


Figure 6.2: Toy model of ground truth and predicted GOSS of an image. The predicted segments for “unknown” are partitioned into true positives TP^{uk} , false positives FP^{uk} , and false negatives FN^{uk} .

or unknown class. Instead, GOSS requires each input pixel to be explicitly classified as belonging to a known or unknown class. Well-known metrics for GS include the variation of information Meila [2005], probabilistic rand index Rand [1971], F-measure Martin et al. [2004], and segmentation covering Arbelaez et al. [2010]. These metrics are initially proposed to evaluate data clustering or edge detection quality. As no multi-class semantic labels are considered, they cannot be directly used to measure the performance of joint GS and OSS.

GOSS Quality. We borrow the idea of the segment matching from the panoptic quality (PQ) Kirillov et al. [2019b] in panoptic segmentation (PS) and adapt the panoptic quality as GOSS quality to be suitable for evaluating GOSS. As shown in Figure 6.1 (d), the GOSS output can be viewed as a set of predicted segments, which is similar to the panoptic output of PS. The primary distinction between GOSS and panoptic predictions lies in the capability of GOSS to predict unknown segments.

We treat the unknown pixels as a new class in addition to N known classes. Thus, there is a total of $N + 1$ classes of segmentation. Utilizing segment matching, a predicted segment from GOSS is matched with the corresponding ground truth segment when their Intersection over Union (IoU) is higher than 0.5. As illustrated in Figure 6.2, this approach enables the identification of true positives (TP), false positives (FP), and false negatives (FN) for the predicted segments generated by GOSS. We let GQ^{kn} be the average GOSS quality over N known classes. Accordingly, GQ^{uk} is the GOSS quality of the unknown class:

$$GQ^{kn} = \frac{1}{N} \sum_{j \in \mathcal{L}^{kn}} \frac{\sum_{(u, \hat{u}) \in TP_j^{kn}} \text{IoU}(u, \hat{u})}{TP_j^{kn} + \frac{1}{2}FP_j^{kn} + \frac{1}{2}FN_j^{kn}} \quad (6.1)$$

$$GQ^{uk} = \frac{\sum_{(u, \hat{u}) \in TP^{uk}} \text{IoU}(u, \hat{u})}{TP^{uk} + \frac{1}{2}FP^{uk} + \frac{1}{2}FN^{uk}} \quad (6.2)$$

where $\text{IoU}(u, \hat{u})$ calculates the Intersection over Union value for the predicted segment u and the ground-truth segment $\hat{u}$. Furthermore, TP_j^{kn} , FP_j^{kn} and FN_j^{kn} denote true positives, false positives and false negatives for the j th known class, respectively.

Similarly, GQ^{uk} is obtained specially for the unknown class with its true positives TP^{uk} , false positives FP^{uk} and false negatives FN^{uk} .

The metrics GQ^{kn} and GQ^{uk} are computed based on the GOSS output (see Figure 6.1 (d)). The known and unknown segments on the GOSS prediction are evaluated separately via GQ^{kn} and GQ^{uk} . However, a unified metric is required to simplify the evaluation. Thus, we define a metric GOSS Quality (GQ) as:

$$GQ = \lambda \cdot GQ^{kn} + (1 - \lambda) \cdot GQ^{uk} \quad (6.3)$$

where we set λ as the most natural number, 0.5, throughout this chapter. If we simply average GQ over $N + 1$ classes, then the ratio between known and unknown would be significantly biased ($N : 1$). In Equation (6.3), GQ takes care of the known and unknown segments equally. We also introduce GQ^{clu} , which only assesses the pixel clustering quality regardless of pixel class (see Figure 6.1(b)).

6.4 Methodology

To effectively perform GOSS, we propose a baseline framework (see Figure 6.3 (a)). The baseline comprises five components: the shared encoder, the pixel classification branch, the pixel clustering branch, the identification module, and the fusion module. Then, we extend the baseline into a more advanced one, GOSS SegmenTor (GST), as shown in Figure 6.3 (b). More details of our design will be described next.

6.4.1 Baseline

GOSS can be modeled as a unified segmentation task incorporating pixel classification and clustering in an open-setting scenario. Given an image $I_o \in \mathbb{R}^{3 \times \hat{h}_o \times \omega_o}$, we expect the proposed baseline to generate semantic and grouping predictions simultaneously. Hence, we adopt a dual-branch architecture, with one branch for pixel classification and another for pixel clustering. As shown in Figure 6.3 (a), two branches share the same encoder as a feature extractor. The branch for pixel classification computes a prediction map $\mathbf{S} \in \mathbb{R}^{\hat{h}_o \times \omega_o}$ while the pixel clustering branch outputs a mask map $\mathbf{G} \in \mathbb{R}^{\hat{h}_o \times \omega_o}$ which includes the grouped class-agnostic segments. The unknown regions in $\mathbf{S}$ are identified, denoted by $\mathbf{S}^{ide}$, which is further fused with $\mathbf{G}$, obtaining the final GOSS output $\mathbf{G}_{oss} \in \mathbb{R}^{2 \times \hat{h}_o \times \omega_o}$.

The baseline model is jointly trained with two losses: the classification loss $\mathcal{L}_{cla}$, and the clustering loss $\mathcal{L}_{clu}$. The total loss is $\mathcal{L}_{ws} = \alpha_{cla} \mathcal{L}_{cla} + \alpha_{clu} \mathcal{L}_{clu}$ where α_{cla} and α_{clu} are positive adjustment weights.

6.4.1.1 Pixel Classification

We train the branch for pixel classification to classify each pixel as one of N classes where N is the number of predefined known classes. DeepLabV3+ Chen et al.

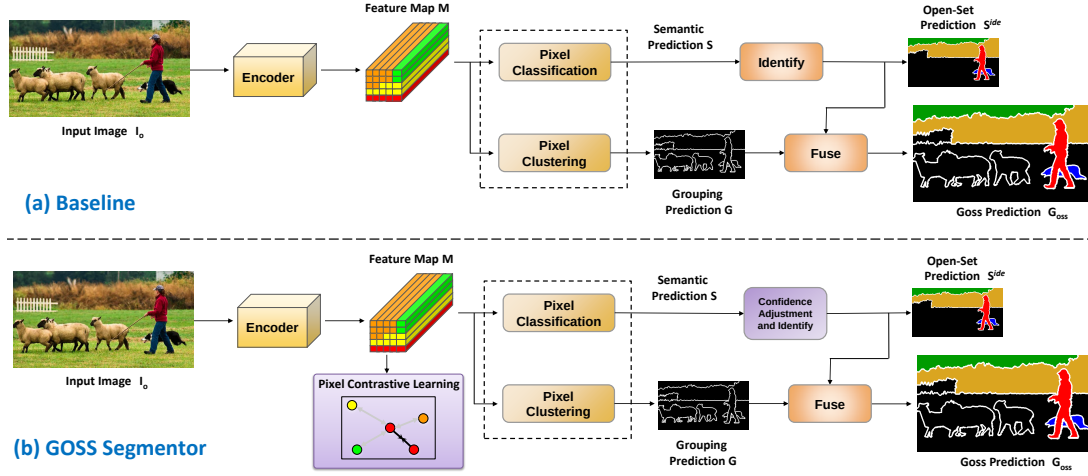


Figure 6.3: **The framework of the baseline and GOSS Segmentor.** (a) Baseline. The input image is fed into the encoder for feature extraction. The dual-branch heads are jointly trained for pixel classification and clustering. Furthermore, pixel-wise contrastive learning is leveraged to learn discriminative feature embeddings. The pixel identification module is designed to recognize sets of pixels of the unknown class from the semantic prediction. The final GOSS output is generated using the identified semantic and grouping predictions. (b) GOSS Segmentor (GST). Confidence adjustment and pixel contrastive learning modules are included.

[2018c], an existing powerful baseline for semantic segmentation, is chosen as the basic architecture for this branch. The branch is updated under $\mathcal{L}_{cla}$, the cross-entropy loss between the predicted semantic map $\mathbf{S}$ and its ground-truth map.

DeepLabV3+ leverages an encoder-decoder architecture that takes a bottom-up pathway network with features at multiple spatial resolutions and appends a top-down pathway with lateral connections. The top-down pathway progressively up-samples features starting from the deepest layer of the network while concatenating or adding them with higher-resolution features from the bottom-up pathway. The Atrous Spatial Pyramid Pooling (ASPP) layer Chen et al. [2017b] is employed in the DeepLabV3+ model to enlarge the receptive field.

6.4.1.2 Pixel Identification

Pixel identification from OSS is executed to identify sets of pixels of unknown classes from the semantic prediction. Hereafter, we study several recipes for pixel identification with which the identified semantic prediction $\mathbf{S}^{ide} \in \mathbb{R}^{h_0 \times w_0}$ is computed after processing $\mathbf{S}$. Common metrics like AUROC and AUPR of OSS assess the distribution overlap between known and unknown classes. Still, pixel identification is required to state if the input pixel is clearly known or not. In other words, binary classification is necessary.

N-model. In the pixel classification branch, it is natural to design the semantic seg-

mentation model with an N -dimensional confidence output $\mathbf{C} \in \mathbb{R}^{N \times h_o \times \omega_o}$. The N -model is restricted to recognizing the set of predetermined known classes. When an unknown region comes up in a test image, it would be erroneously classified as one of the known classes. To identify unknown pixels based on outputs from the N -model, we employ the comparable OSS method, Maximum Softmax Probability (MSP) Hendrycks and Gimpel [2017] or Maximum Unnormalized Logit (MaxLogit) Hendrycks et al. [2022]. Thresholds classify pixels as belonging to a known or unknown class.

N+1-model. As opposed to the N -model, the $N+1$ -model DeVries and Taylor [2018]; Corbière et al. [2019] contains the unknown class in the output $\mathbf{C} \in \mathbb{R}^{(N+1) \times h_o \times \omega_o}$ such that the $N+1$ -model can directly identify the unknown pixels. During the training stage, the $N+1$ -model explicitly takes the “unlabeled” pixels (*i.e.*, the “void” pixels) as the unknown pixels. $N+1$ -model is not valid if no “void” pixels are provided.

6.4.1.3 Pixel Clustering

The pixel clustering branch is built in parallel with the pixel classification branch. The goal of this branch is to partition the whole image into clusters. During training, to generate the corresponding annotations, we convert the semantic labeling of the known classes into connectivity labeling by ignoring the previous semantics of each segment. The top-performing method, super boundary-to-pixel direction (Super-BPD) Wan et al. [2020] is selected to establish the branch. The branch is trained in a supervised manner as well. Super-BPD applies the ground-truth annotation generated by the distance transform algorithm. Using the Super-BPD model, the representation of boundary-to-pixel direction (BPD) for each pixel is learned ($\mathcal{L}_{clu} = \mathcal{L}_{bpd}$). Super-BPDs are extracted based on the initial BPDs using the component-tree computation, followed by graph partitioning to merge super-BPDs into new segments.

6.4.1.4 Fusion Module

The identification module outputs the identified semantic prediction $\mathbf{S}^{ide}$. Based on $\mathbf{S}^{ide}$, the grouping output $\mathbf{G}$ becomes $\mathbf{G}^{uk} \in \mathbb{R}^{h_o \times \omega_o}$ where the element $g_i \rightarrow void$ if the corresponding semantic prediction $s_i \in [0, \dots, N-1]$. Afterward, $\mathbf{S}^{ide}$ is merged with $\mathbf{G}^{uk}$ to form the GOSS output $\mathbf{G}_{oss} = [\mathbf{goss}_1, \mathbf{goss}_2, \dots, \mathbf{goss}_{h_o \omega_o}]$, where $\mathbf{goss}_i = (s_i, g_i) \in [0, \dots, N] \times [1, \dots, g_{max}] \cup void$. For i th pixel in $\mathbf{G}_{oss}$, $\mathbf{goss}_i = (s_i, void)$ if $s_i \in [0, 1, \dots, N-1]$ or $\mathbf{goss}_i = (N, g_i)$ if $s_i = N$. The prediction $\mathbf{G}_{oss}$ can be viewed as a map that is composed of a set of several segments (see “GOSS prediction” in Figure 6.3).

6.4.2 GOSS Segmentor

As shown in Figure 6.3 (b), the baseline model is extended to a new model that we call GOSS Segmentor (GST). Keeping the original five baseline components, we

propose to equip the baseline with a confidence adjustment module and a contrastive learning module.

6.4.2.1 Confidence Adjustment

For the N+1 model, it is hard to accommodate unknown classes of objects since the model is trained without seeing any examples from these classes. Instead of completely trusting the confidence prediction $\mathbf{C}$, specific to the pixel identification of the N+1-model, we propose to modify $\mathbf{C}$ using a confidence adjustment. Particularly, for the i th pixel, its confidence score after softmax, $\mathbf{c}_i = [\mathbf{c}_i^{kn}, c_i^{uk}] \in \mathbb{R}^{N+1}$, is re-scaled as $[\mathbf{c}_i^{kn}, \beta^{uk} c_i^{uk}]$ where $\beta^{uk} \in (1, +\infty)$ is the scale coefficient of the confidence of the unknown class.

6.4.2.2 Pixel Contrastive Learning

To learn more discriminative representations for a better GOSS performance, inspired by Wang et al. [2021c], we adopt a pixel-wise contrastive learning algorithm where we contrast embeddings with different semantic labels. We have i th pixel embedding $\mathbf{e}_i \in \mathbb{R}^{cn}$ in the feature map $\mathbf{M} \in \mathbb{R}^{cn \times h_o \times \omega_o}$ where cn , h_o and ω_o are the channel number, height and width of the feature map. For $\mathbf{e}_i$, the positive pixel embeddings $\mathbf{e}_i^+$ have the same ground truth label to $\mathbf{e}_i$ in the same feature map, while the negative pixel embeddings $\mathbf{e}_i^-$ have different ground truth from $\mathbf{e}_i$. The pixel to pixel contrastive loss Wang et al. [2021c]; Sohn [2016] is then defined as:

$$\mathcal{L}_{pc,i} = \frac{1}{|N_{pst,i}|_{i^+ \in N_{pst,i}}} \sum -\log \frac{\exp(\mathbf{e}_i \cdot \mathbf{e}_i^+ / \tau)}{\exp(\mathbf{e}_i \cdot \mathbf{e}_i^+ / \tau) + \sum_{i^- \in N_{neg,i}} \exp(\mathbf{e}_i \cdot \mathbf{e}_i^- / \tau)} \quad (6.4)$$

where $N_{pst,i}$ and $N_{neg,i}$ are positive and negative embedding sets for pixel embedding $\mathbf{e}_i$. τ is the temperature parameter. We employ the semi-hard example sampling strategy from Wang et al. [2021c] to construct the positive and negative sample sets. Before the $\mathcal{L}_{pc}$ is calculated, we downscale the ground-truth map to make it have the same size as the feature map $\mathbf{M}$.

The total loss $\mathcal{L}_{ws}$ of GST is obtained by merging the pixel contrastive loss $\mathcal{L}_{pc}$ with the classification loss $\mathcal{L}^{cla}$ and clustering loss $\mathcal{L}^{clu}$ as follows: $\mathcal{L}_{ws} = \alpha_{cla} \mathcal{L}^{cla} + \alpha_{clu} \mathcal{L}^{clu} + \alpha_{pc} \mathcal{L}_{pc}$, where α_{pc} is a positive adjustment weight for $\mathcal{L}_{pc}$. Contrastive learning aims to make the representations of pixels in the latent space closer when they belong to the same class and farther apart when they belong to different classes. This metric learning technique has been widely used in segmentation tasks, with many existing works reporting better empirical results Harley et al. [2017]; De Brabandere et al. [2017]; Hwang et al. [2019]; Zhao et al. [2019]. In GOSS, we apply contrastive learning to train our model, intending to generate more representative embeddings for open-set evaluation.

6.5 Benchmark

Most datasets for OSS, like StreetHazards Hendrycks et al. [2022] and Road Anomaly Lis et al. [2019], present separate unknown objects in an image. Ensuring that objects of unknown classes naturally appear together (are adjacent) in the image, in this chapter, we split the full set of labeling categories into known and unknown classes using a proper ratio. We simulate the training and testing of GOSS using existing semantic segmentation datasets, i.e., COCO-Stuff Caesar et al. [2018], and Cityscapes Cordts et al. [2016]. Note that grouping labels of unknown segments are derived from their initial ground-truth semantic labels before the split. Following Wan et al. [2020], we convert the original semantic labeling of unknown areas to GS ground truths using connectivity labeling.

6.5.1 COCO-Stuff-GOSS

COCO-Stuff Caesar et al. [2018] augments the popular COCO Lin et al. [2014] dataset with stuff classes and dense-pixel annotations. It has a large-scale semantic multi-class setting containing the “things” and “stuff” classes. On COCO-Stuff, around 94% of the pixels are labeled with one semantic category, and the remaining are “unlabeled” pixels. We customize COCO-Stuff, creating a new benchmark named COCO-Stuff-GOSS. We strictly divide existing specific classes of COCO-Stuff into known and unknown classes. Training and testing images are selected from “train2017” and “val2017”. Those categories defined as unknown categories will not be represented in the training examples. Every selected testing example comprises objects from the set of known categories and unknown categories (or from only unknown categories). The benchmark statistics on different splits are shown in Table 6.2.

VOC Split. The “VOC Split” is a common category split Pinheiro et al. [2015]; Hu et al. [2018b]; Dhamija et al. [2020a]; Joseph et al. [2021] that provides 20 “thing” classes defined in PASCAL VOC Everingham et al. [2010] as “known thing” classes. The remaining 60 “thing” classes are chosen as “unknown thing” classes.

Manual Split. We divide COCO-Stuff categories according to how frequently each specific class appears. We count the number of occurrences of each class and calculate its ratio over the number of all training images. For example, in the “Manual-20/60” split, following that at least one and at most two classes are chosen from each sub-class, we select 20 of the most popular “thing” classes and treat the remaining “thing” classes as unknown. Besides, all “stuff” classes are set as known classes.

Random Split. We also conduct experiments with a “Random Split”, where all classes are randomly re-defined into known and unknown classes regardless of their super-class and sub-class. The data split of VOC-20/60 and Manual-20/60 do not include “stuff” categories as unknown classes, but Random-111/60 ensures that the known (or unknown) class includes specific classes from both the ‘thing’ and ‘stuff’ super-class. More details can be found in Table 6.2.

Table 6.2: Details of different splits of the COCO-Stuff-GOSS/Cityscapes-GOSS benchmarks. The numbers in the table indicate how many known (or unknown) classes are selected for each data split and how many training (or testing) images are kept. We also provide details of the original COCO-Stuff Caesar et al. [2018] and Cityscapes Cordts et al. [2016] for comparison.

Dataset	Data Split	Known	Unknown	Known (thing)	Unknown (thing)	Known (stuff)	Unknown (stuff)	Train images	Test images
COCO-Stuff Caesar et al. [2018]	-	171	0	80	0	91	0	118287	5000
COCO-Stuff-GOSS	VOC-20/60	111	60	20	60	91	0	21711	4080
	Manual-20/60	111	60	20	60	91	0	17293	4264
	Random-111/60	111	60	51	29	60	31	18707	4156
Cityscapes Cordts et al. [2016]	-	19	0	8	0	11	0	2975	500
Cityscapes-GOSS	Manual-16/3	16	3	5	3	11	0	2975	500
	Manual-13/6	13	6	5	3	8	3	2975	500

6.5.2 Cityscapes-GOSS

The Cityscapes Cordts et al. [2016] dataset consists of 5000 images (2975 train, 500 val, 1525 test) covering urban street scenes in driving scenarios. Dense pixel annotations of 19 classes are provided, that is, 8 “thing” and 11 “stuff” classes. As one goal of the proposed GOSS is to advance self-driving systems, we construct the Cityscapes-GOSS Benchmark. We divide the categories under the “manual split”. Unlike the COCO-Stuff-GOSS Benchmark, all images, regardless of whether they contain unknown categories or not, are kept. We consider pixels from unknown classes as “void” pixels. Table 6.2 presents more details.

Manual Split. We present two versions of the Cityscapes-GOSS Benchmark. Following the split in Cen et al. [2021a], we build the first version, “Manual-16/3”, which includes “car”, “truck”, and “bus” as the “unknown thing”. Based on the first version, we additionally make “building”, “traffic sign”, and “vegetation” as “unknown stuff” to produce a more challenging version, “Manual-13/6”.

6.6 Experiments

Experimental results are presented in this section to demonstrate the rationality and effectiveness of GOSS. Using the baseline and proposed GST, we perform our task on COCO-Stuff-GOSS and Cityscapes-GOSS. The performance is mainly measured via the metric GQ.

6.6.1 Implementation

For all models, ResNet-50 He et al. [2016] pre-trained on ImageNet Deng et al. [2009] is utilized as the encoder. All models are trained for 60K/40K iterations with a batch size of 10/2 on COCO-Stuff-GOSS/Cityscapes-GOSS. The “poly” learning rate policy Liu et al. [2015] is applied with the initial learning rate being set to $5e-5$. GST models are updated using Adam optimization Kingma and Ba [2014] without weight

Table 6.3: GOSS results of GST (N+1-model+CA+CL) on COCO-Stuff-GOSS under three splits. “CA” is the confidence adjustment, and “CL” is the cross-pixel contrastive learning. The best results of GQ are in **bold**.

Data Split	Identification Method	Clustering Method	OSS Metric		GS Metric		GOSS Metric		
			AUROC $\uparrow$	AUPR $\uparrow$	mIoU $\uparrow$	GQ ^{clu} $\uparrow$	GQ ^{kn} $\uparrow$	GQ ^{uk} $\uparrow$	GQ $\uparrow$
VOC-20/60	N-model+MSP Hendrycks and Gimpel [2017]	super-BPD	76.5	26.4	12.7	27.3	14.4	3.0	8.70
	N-model+Maxlogit Hendrycks et al. [2022]		71.0	22.0	12.7	27.3	3.6	4.3	3.93
	N-model+DML Cen et al. [2021a]		71.1	20.9	12.5	26.6	1.6	4.3	2.93
	N+1-model DeVries and Taylor [2018]		75.7	25.7	12.4	25.9	14.3	3.3	8.83
	GST (Ours)		77.0	27.0	13.6	28.7	15.4	4.9	10.15
Manual-20/60	N-model+MSP Hendrycks and Gimpel [2017]	super-BPD	74.7	27.8	12.8	26.1	13.1	2.3	7.70
	N-model+Maxlogit Hendrycks et al. [2022]		71.4	24.1	12.8	26.1	3.7	4.5	4.13
	N-model+DML Cen et al. [2021a]		75.9	28.5	12.8	27.3	13.3	2.3	7.81
	N+1-model DeVries and Taylor [2018]		76.2	29.5	12.9	27.4	13.1	2.2	7.62
	GST (Ours)		77.3	30.7	13.8	28.3	14.3	3.9	9.15
Random-111/60	N-model+MSP Hendrycks and Gimpel [2017]	super-BPD	77.5	43.1	12.6	27.9	16.6	4.1	10.36
	N-model+Maxlogit Hendrycks et al. [2022]		73.9	41.2	12.6	27.9	3.7	8.7	6.22
	N-model+DML Cen et al. [2021a]		78.6	44.4	12.7	28.5	17.0	4.3	10.67
	N+1-model DeVries and Taylor [2018]		77.9	44.9	12.6	27.1	16.1	4.2	10.16
	GST (Ours)		79.8	47.0	13.7	29.0	17.5	4.2	10.87

decay. The weights α_{cla} , α_{clu} and α_{pc} are 1.0, $1e-4$ and $1e-1$. The thresholds in the identification module and the scale β^{uk} (for +CA) are set to 0.5 (0.75 for Cityscapes-GOSS) and 5.0, respectively. Our models are implemented in PyTorch Paszke et al. [2019]. We note that the N+1-model cannot be used for the Cityscapes-GOSS dataset. As we consider labels of unknown classes to be “void” instead of directly filtering out the image, the entropy of void pixels cannot be added to the loss.

6.6.2 Results

The results of GOSS on COCO-Stuff-GOSS and Cityscapes-GOSS using various identification methods are reported in Table 6.3 and 6.4, respectively. In addition to GOSS quality, we also provide metrics for OSS (AUROC and AUPR) and GS (mIoU and GQ^{clu}) tasks to show that the models perform reasonably on these relevant older tasks.

For COCO-Stuff-GOSS in Table 6.3, GST becomes the best-performing model. For example, on the “Manual-20/60 split”, GST attains 9.15% GQ, outperforming the N-model+MSP by a healthy margin of nearly 1.45%. Compared to other baselines, the pixel contrastive learning module assists GST to better discriminate between the known pixels and the unknown pixels in most cases (See “OSS Metric” in Table 6.3). Moreover, it boosts the clustering accuracy in GS. One of the baseline models, N-model+MaxLogit using threshold, sacrifices much of GQ^{kn}, but it achieves a high GQ^{uk}. As expected, MaxLogit identifies more unknown areas. However, it does not simultaneously maintain the pixel classification accuracy Jung et al. [2021]. For Cityscapes-GOSS in Table 6.4, we find a similar performance ranking to COCO-Stuff-GOSS in Table 6.3. For DML Cen et al. [2021a], except for the case on COCO-Stuff-GOSS of “VOC-20/60 split”, it wins MSP and MaxLogit on the other benchmarks.

Several examples from the built benchmark are visualized in Figure 6.4 to illustrate the GOSS setting better. Taking one example from Figure 6.4 (2nd-row figure),

Table 6.4: GOSS results of GST (N-model+CL) on Cityscapes-GOSS under “Manual-16/3” and “Manual-13/6” splits. “CL” is cross-pixel contrastive learning. The best results of GQ are in **bold**.

Data Split	Identification Method	Clustering Method	GOSS Metric		
			GQ ^{kn} ↑	GQ ^{uk} ↑	GQ ↑
Manual-16/3	N-model+MSP Hendrycks and Gimpel [2017]	super-BPD	12.1	1.1	6.60
	N-model+Maxlogit Hendrycks et al. [2022]		6.2	1.7	3.92
	N-model+DML Cen et al. [2021a]		12.3	1.1	6.74
	GST (Ours)		13.4	1.1	7.31
Manual-13/6	N-model+MSP Hendrycks and Gimpel [2017]	super-BPD	7.4	0.1	3.80
	N-model+Maxlogit Hendrycks et al. [2022]		3.4	0.6	2.01
	N-model+DML Cen et al. [2021a]		7.5	0.2	3.87
	GST (Ours)		8.3	0.2	4.26

Table 6.5: Ablation study on COCO-Stuff-GOSS under “Random-111/60” split: clustering method. Two clustering models of GST, super-BPD in a supervised setting and DFC Kim et al. [2020] in an unsupervised setting are compared.

Data Split	Identification Method	Clustering Method	GS Metric		GOSS Metric		
			mIoU ↑	GQ ^{clu} ↑	GQ ^{kn} ↑	GQ ^{uk} ↑	GQ ↑
Random-111/60	N-model+MSP Hendrycks and Gimpel [2017]	super-BPD	12.6	27.9	16.6	4.1	10.36
	N-model+Maxlogit Hendrycks et al. [2022]		12.6	27.9	3.7	8.7	6.22
	N-model+MSP Hendrycks and Gimpel [2017]	DFC	4.8	4.7	16.8	1.8	9.31
	N-model+Maxlogit Hendrycks et al. [2022]		4.8	4.7	3.7	3.6	3.69

GOSS accurately segments out “unknown dogs” from “unknown grass” (see Figure 6.4 (f)). Compared to the prediction of OSS in Figure 6.4 (c), GOSS can provide richer information for intelligent systems to make decisions. With GOSS prediction, robots may avoid the obstacle (“unknown dogs”) when they enter an unfamiliar scene (“unknown grass”). In terms of the GST model, we observe from Figure 6.4 that the confidence adjustment module of GST effectively helps the N+1-model to detect more unknown regions (see Figure 6.4 (b) and (c)).

6.6.3 Ablation Study

Clustering Method. As shown in Figure 6.5, we train the pixel clustering branch for grouping unknown areas in an unsupervised manner by applying differentiable feature clustering (DFC) loss Kim et al. [2020]. DFC in an unsupervised setting has a basic clustering performance, but it is worse than Super-BPD in a supervised setting. **Component Effect.** Here, we discuss the effects of each model component (“CA” or “CL”). The results on COCO-Stuff-GOSS under the “Manual-20/60” split are shown in Table 6.6, which indicates “CA” or “CL” can solely boost the performance by a certain margin.

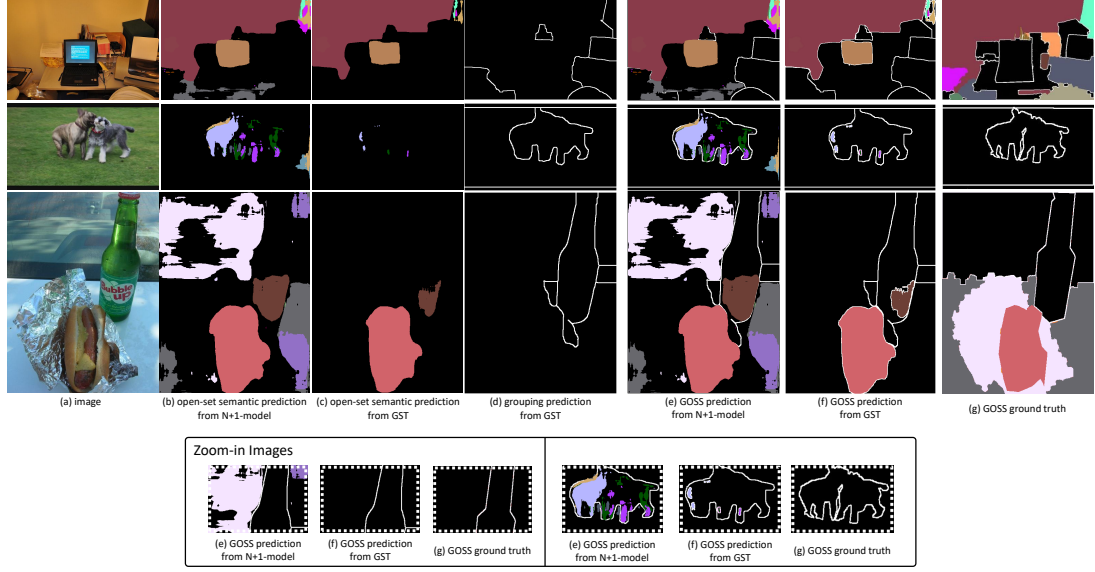


Figure 6.4: Visualized segmentation results from GST (N+1-model+CA+CL) on COCO-Stuff-GOSS. The GOSS prediction (f) merges the OSS prediction (c) and the grouping prediction (d). Hence, within GOSS prediction (f), “objects” inside identified unknown regions can be segmented. For example, unknown objects, “paper” in the 1st row, “dog/grass” in the 2nd row, and “bottle” in the 3rd row are correctly outlined even though their classes are unknown. We also have some zoom-in images to show the effectiveness of the GST.

Training Strategy. For all models in Table 6.3 and 6.4, the pixel classification branch and the pixel clustering branch are trained in a single unified architecture. Here, we study a different training strategy, “Separate,” where two branches are trained separately, and their outputs are merged. We find from Table 6.7 that the performances of “Separate” and our “Single” network are close. We finally choose “Single” since it is fast, light, and easy to implement.

6.6.4 Discussion

Challenging Task. The results in Tables 6.3 and 6.4 verify that GOSS is challenging despite our baseline framework relying on strong encoders and a reasonable architecture. The first main reason is that it is non-trivial to perform accurate pixel identification under the open-set setting. For example, “unknown laptop” has been misclassified as “known tv” in the 1st-row figure in Figure 6.4. Furthermore, the clustering branch suffers a performance drop when the model encounters the unfamiliar appearances of objects from unknown categories at test time. There is significant room for future improvement on the task of GOSS.

Number of Unknown Classes. The increased number of unknown classes makes the GOSS performance significantly drop as we observe the performance of “Manual-16/3” and “Manual-13/6” in Cityscapes GOSS (see Table 6.4).

Table 6.6: GOSS results of GST (N+1-model+CA+CL) on COCO-Stuff-GOSS under “Manual-20/60” split. “CA” is the confidence adjustment and “CL” is the cross-pixel contrastive learning. The best results of GQ are in **bold**.

Data Split	Identification Method	Clustering Method	GOSS Metric		
			$GQ^{kn} \uparrow$	$GQ^{uk} \uparrow$	$GQ \uparrow$
Manual-20/60	N+1-model	super-BPD	13.1	2.2	7.62
	N+1-model+CA		14.0	2.5	8.24
	N+1-model+CL		14.1	3.5	8.82
	GST (Ours)		14.3	3.9	9.15

Table 6.7: Ablation study on COCO-Stuff-GOSS under “Random-111/60” split: training strategy. Two strategies are compared.

Data Split	Identification Method	Clustering Method	Strategy	GOSS Metric		
				$GQ^{kn} \uparrow$	$GQ^{uk} \uparrow$	$GQ \uparrow$
Random-111/60	N-model+MSP Hendrycks and Gimpel [2017]	super-BPD	Single	16.6	4.1	10.36
	N-model+Maxlogit Hendrycks et al. [2022]			3.7	8.7	6.22
	N-model+MSP Hendrycks and Gimpel [2017]		Separate	16.4	4.2	10.17
	N-model+Maxlogit Hendrycks et al. [2022]			3.4	8.9	6.14

6.7 Summary

The proposed new task, referred to as GOSS, is introduced in this chapter. We aim to build upon the well-defined OSS to generate more comprehensive predictions. The task is to semantically classify pixels as one of the known classes or an unknown class and cluster the detected unknown pixels. With more extracted information inside the unknown region, GOSS might benefit intelligent systems in their decision-making process. Specific to the new setting, a metric, two benchmarks, and a corresponding baseline model are presented. We hope the work in this chapter can provide a new alternative to a more comprehensive pixel-level scene understanding.

In previous chapters and this chapter, we have extensively explored various open-set recognition tasks applied to 2D images. Given that open-set 3D tasks remain relatively underexplored, our focus will shift to point cloud data in the next chapter. Specifically, we aim to develop a model that considers both point clouds and feature maps for feature learning within open-set 3D recognition tasks. This model is designed to learn more discriminative visual features for effectively discerning geometric differences between known and unknown points. As such, the unknown point cloud objects are more easily identified.

Cut-and-Mix for Open-set Point Cloud Learning

The preceding two chapters have explored the open-set recognition problem, primarily concentrating on the analysis of 2D images. In this chapter, our focus shifts to the same problem but with an emphasis on handling point cloud data. This allows us to go into a different dimension of the open-set recognition challenge, expanding our understanding and approach to encompass a broader range of data modalities.

Point cloud learning is receiving increasing attention. However, most existing point cloud models lack the practical ability to deal with the unavoidable presence of unknown objects. This chapter mainly discusses point cloud learning under open-set settings, where we train the model without data from unknown classes and identify them in the inference stage. Basically, we propose to solve open-set point cloud learning using a novel Point Cut-and-Mix (PointCaM) mechanism consisting of Unknown-Point Simulator (UPS) and Unknown-Point Estimator (UPE) modules. Specifically, we use the UPS to simulate out-of-distribution data in the training stage by manipulating the geometric context of partially known data. Based on this, the UPE module learns to exploit the point cloud’s feature context to discriminate between known and unknown data. Extensive experiments show the plausibility of open-set point cloud learning and the effectiveness of our proposed solutions. This chapter is based on our under-review work Hong et al. [2023d].

7.1 Introduction

To better illustrate complicated scenes in the real world, 3D data has been widely investigated in computer vision research. As a fundamental representation of 3D data, point clouds that can be collected from 3D scanners Endres et al. [2013]; Jaboyedoff et al. [2012] have shown great potential in AI-related applications, such as robotics Pomerleau et al. [2015]; Li et al. [2020]; Ze et al. [2023]; Li and Guivant [2023], remote sensing Li et al. [2012], and augmented/virtual reality Chen et al. [2019b].

Recent works Qi et al. [2017a,b]; Zhao et al. [2021a]; Guo et al. [2020] leverage data-driven neural networks to analyze the context of 3D point clouds. Particularly, large-scale labeled point cloud data plays a crucial role in impacting the performance

of deep networks. Despite the fast development in 3D sensing technology, point cloud data still requires expensive equipment for collection and huge human resources for annotation. The training datasets are assumed to contain all pre-defined object classes in the scenes; however, it is impractical to cover all possible classes in the real world. Thus, the richness and diversity of point cloud data are usually insufficient, especially for analyzing large-scale scenes Armeni et al. [2017]; Behley et al. [2019] and a wide variety of objects Wu et al. [2015]; Chang et al. [2015]. Moreover, the effectiveness of point cloud data is also limited by its inherent sparsity and irregularity, which cause further challenges in common visual tasks such as point cloud classification Qi et al. [2017a]; Qiu et al. [2021a], segmentation Choy et al. [2019]; Hu et al. [2020b], and detection Qi et al. [2019].

The advanced point cloud learning models Qi et al. [2017a,b]; Hu et al. [2020b]; Wang et al. [2018] mainly focus on closed-set environments. These works often make a strong assumption that the object classes known during the deployment stage are only those provided during the training stage. However, this closed-set assumption is too restrictive since, in real-world environments, the autonomous systems will unavoidably present unknown object classes that are not part of the training data. Currently, open-set recognition in the 2D image domain is active Zhang et al. [2020b]; Dhamija et al. [2020a]; Kong and Ramanan [2021], while the open-set learning of 3D point clouds is not commonly studied. In fact, this study has significant value in real-world applications. A possible outdoor scenario applies in self-driving, where an intelligent system may confront some objects that have not been seen before. By leveraging 3D open-set settings, the intelligent system can effectively identify such unknown objects, making drivers aware of the surroundings. Another potential use case for our 3D open-set settings is in XR (extended reality) systems. When a user is immersed in an XR gaming environment at home, the sudden appearance of a cat or a baby in the guarded area could pose a safety risk. However, traditional alerting mechanisms may not detect these unexpectedly moving creatures as their detection algorithms are trained on regular indoor furniture and objects under closed-set settings. By incorporating our proposed settings and solutions, the XR system can easily recognize such “unknown” creatures, thereby increasing the robustness and effectiveness of the alerting mechanism. Given the promises in academic research and real-world applications, we explore open-set point cloud learning by enhancing the models’ abilities to deal with the inevitable unknown objects in 3D point cloud data.

Recently, a few studies investigate 3D data under open-set settings Ma et al. [2018]; Bhardwaj et al. [2021]. For example, Zhao et al. [2021b] addressed the few-shot 3D semantic segmentation task under an open-set setting, which presents an attention-aware multi-prototype transductive inference method. The target of Wong et al. [2020] is unknown instance recognition under self-driving scenarios, where a feature predictor for anchors is designed to fit the characteristics of LiDAR data. Cen *et al.* Cen et al. [2021b] study open-set LiDAR point cloud detection through metric learning to overcome the problem of massive false positive detection samples. Moreover, Cen et al. [2022] explores open-world LiDAR segmentation, where

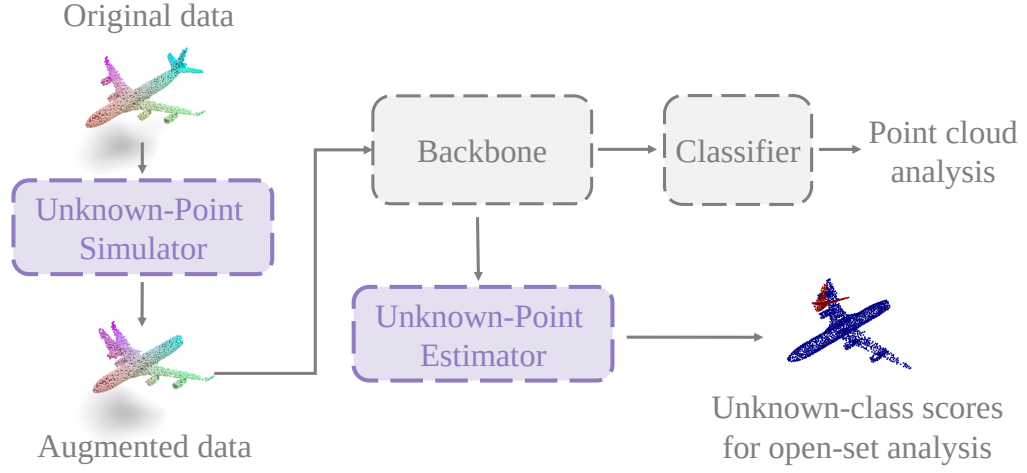


Figure 7.1: The PointCaM mechanism consists of two main modules: the Unknown-Point Simulator (UPS) simulates the out-of-distribution (OOD) data during the training, and the Unknown-Point Estimator (UPE) is responsible for computing the unknown-class scores.

a redundancy classifier framework adapts the model to the open-set setting and incremental learning tasks. However, these works have some limitations that restrict general open-set 3D performance. First, methods Zhao et al. [2021b]; Wong et al. [2020]; Cen et al. [2021b]; Dong et al. [2022] lack simulated data about unknown classes, which impacts the model’s sensitivity to unknown objects. Second, Wong et al. [2020]; Cen et al. [2021b, 2022]; Dong et al. [2022] only focus on the classifier’s features for recognizing unknown objects while neglecting information from the data and learned feature maps.

To address the above limitations, in this chapter, we introduce a novel Point Cut-and-Mix (PointCaM) mechanism for open-set 3D point cloud learning. As illustrated in Figure 7.1, PointCaM incorporates two main modules: Unknown-Point Simulator (UPS)¹ and Unknown-Point Estimator (UPE)². During the training process, we leverage the Unknown-Point Simulator to simulate out-of-distribution (OOD) data by partially manipulating the geometric context of point clouds. To further discriminate the known and unknown objects provided by the UPS module, our UPE is designed to calculate an unknown-class score for each point. In particular, the UPE module estimates the unknown-class scores by adaptively fusing the multi-level feature context of point cloud data to learn the geometric dependencies and semantic correlations between known and unknown data. The main contributions of this chapter are summarized as follows:

- We design a novel mechanism, PointCaM, for open-set point cloud learning.

¹UPS for segmentation and Unknown-Sample Simulator (USS) for classification.

²UPE for segmentation and Unknown-Sample Estimator (USE) for classification. For simplicity, throughout the manuscript, we mainly use the terms: “unknown point,” “UPS,” and “UPE.”

Two-component modules, the Unknown-Point Simulator (UPS) and the Unknown-Point Estimator (UPE) are proposed to simulate OOD data and estimate the unknown-class scores, respectively.

- UPS helps to simulate OOD objects, where the selected point clouds are rotated and translated to exhibit highly distinct geometric distributions from known objects.
- UPE adaptively fuses information from multiple feature encoding layers and computes the unknown scores. The unknown score indicates the probability of whether or not the input is unknown. By utilizing information from both 3D coordinates and high-dimensional feature representations, UPE can leverage rich geometric and semantic contexts for differentiating between known and unknown objects.

7.2 Related Work

7.2.1 Point Cloud Learning

Following the success of Convolutional Neural Networks (CNNs) in image recognition, several papers Guo et al. [2020] invent dedicated CNN-based models to analyze point cloud data. In general, point cloud learning relies highly on the learned point feature representations for accurate visual recognition. Recent networks can be categorized into two main streams: indirect and direct methods. Specifically, the indirect methods project original point clouds into certain intermediate representations, *e.g.*, multi-view images Su et al. [2015] or voxels Maturana and Scherer [2015], and apply regular 2D/3D CNNs to analyze the projected data. More recently, researchers tend to directly analyze given point cloud data using the tools of multi-layer perceptron (MLP) Qi et al. [2017a,b], graph structure Wang et al. [2019c]; Liu et al. [2019], transformer Zhao et al. [2021a]; Guo et al. [2021], *etc.* Compared to indirect methods, direct methods can save complicated data processing, avoid information loss, and benefit practical feasibility. In this chapter, we focus on direct open-set point cloud learning, including the basic visual tasks of semantic segmentation and classification.

7.2.2 3D Open-set Settings

Recently, a few works addressed 3D data under open-set settings Ma et al. [2018]; Bhardwaj et al. [2021]. The few-shot 3D semantic segmentation task under an open-set setting was first studied in Zhao et al. [2021b], where an attention-aware multi-prototype transductive inference method is proposed. In Wong et al. [2020], a feature predictor for anchors is specially designed to fit the characteristics of LiDAR data, targeting unknown instance recognition under self-driving scenarios. Cen *et al.* Cen et al. [2021b] study open-set point cloud detection via metric learning to overcome the problem of massive false positive detection samples in LiDAR-based data. Moreover,

open-world LiDAR segmentation is explored in Cen et al. [2022], where a redundancy classifier framework adapts the model to the open-set setting and incremental learning tasks. However, prior studies are subject to certain limitations: (1) methods in Zhao et al. [2021b]; Wong et al. [2020]; Cen et al. [2021b]; Dong et al. [2022] lack simulated data regarding unknown classes, which may impact the model’s sensitivity to unknown objects; and (2) Wong et al. [2020]; Cen et al. [2021b, 2022]; Dong et al. [2022] only focus on the classifier’s features for recognizing unknown objects, while neglecting information from the data and learned feature maps. To address these limitations, our approach, PointCaM, utilizes parts of the point cloud to approximate the unknown data roughly. Specifically, we rotate and translate these point cloud sections to ensure sufficient geometric pattern differences from known objects. We then leverage the contextual information of the point cloud in both geometric and feature spaces to identify unknown data. Particularly, we adaptively fuse information from feature spaces with guidance from geometric space to compute the unknown score. The rich information concerning differences between known and unknown objects in such spaces enables the model to differentiate between the known and unknown categories more effectively.

7.2.3 Cut-Mix Augmentation

Cut-Mix and its related data augmentations present a simple way to improve visual recognition performance, including object detection Ghiasi et al. [2021], self-supervised representation learning Han et al. [2022], image classification Yun et al. [2019], and anomaly detection Li et al. [2021]. For example, Cut-Paste-and-Learn Dwibedi et al. [2017] proposes to cut foreground object instances and paste them on diverse background images to synthesize object detection training images with bounding box labels. Copy-Paste Ghiasi et al. [2021] shows that applying random scale jittering on two random training images can significantly benefit training instance segmentation models. CutMix Yun et al. [2019] cuts a rectangular image patch from an image and pastes it at a random location of another image. CutPaste Li et al. [2021] cuts an image patch and pastes it at a random location of the same image, producing spatial irregularities to estimate real defects. Inspired by the methods for image domain data augmentation, recent papers Harris et al. [2020]; Lee et al. [2021]; Zhang et al. [2022] have extended CutMix Yun et al. [2019] to the point cloud domain for regular closed-set recognition tasks. In open-set recognition, data augmentations are used to generate samples that simulate the presence of OOD classes during training. This can help the model learn more robust features that are less specific to the known classes and more generalizable to new or unknown classes. For example, Neal et al. [2018] introduces the counterfactual image generation augmentation, which generates examples that are close to the training set examples yet do not belong to any training category using generative adversarial networks. Unlike the existing works mentioned above, we present a Cut-and-Mix mechanism as a pioneering approach targeting open-set point cloud learning. Particularly, our proposed mechanism simulates the OOD data and enhances the models’ capability of

identifying unknown data via a learnable module. Additionally, we propose learning an Unknown-Point Estimator module to exploit the point cloud’s feature context for discriminating the known and unknown data.

7.3 Problem Statement

7.3.1 Open-set Point Cloud Learning

Open-set point cloud learning includes two tasks: open-set point cloud segmentation and classification, where the objective is to assign an unknown score to each point (or sample). The deep model is trained solely on data from N_c known classes, but during evaluation, it must handle both N_c known and unknown classes. Assuming there is one unknown class, we calculate an unknown-class score, OS (or OS_{cls}), for each testing point (or sample) to estimate its probability of belonging to the unknown class. The overall performance of open-set point cloud learning is measured based on the unknown-class scores of all testing points (or samples), denoted as $\{OS_1, OS_2, \dots, OS_i, \dots\}$ (or $\{OS_{cls,1}, OS_{cls,2}, \dots, OS_{cls,j}, \dots\}$). Ideally, if the i -th point (or j -th sample) belongs to the unknown class, we hope that OS_i (or $OS_{cls,j}$) approaches 1; while the point (or sample) belongs to the known classes, the corresponding score is expected to be 0. In summary, the goal of open-set point cloud learning is to train a network that can generate accurate unknown scores for each point (or sample).

7.3.2 Benchmarks

We adopt three widely used point cloud learning datasets, S3DIS Armeni et al. [2017], ModelNet40 Wu et al. [2015], and ScanObjectNN Uy et al. [2019] to conduct our experiments on the proposed 3D open-set tasks. Specifically, S3DIS Armeni et al. [2017] dataset contains 6 large-scale workspace point cloud scans, which can be further divided into 272 different rooms illustrating lobby, office, hallway, conference room, *etc.* In each room, about 0.5 to 2.5 million points are labeled into 13 classes for semantic segmentation. ModelNet40 Wu et al. [2015] consists of 12,311 synthetic 3D object meshes in 40 different categories, where the corresponding point cloud data is sampled from the surface of each object mesh via uniform sampling. Following the common practice in the previous classification works Qi et al. [2017a,b], about 9,843 point clouds are used for training, while the rest 2,468 samples are reserved for testing. As a real-world counterpart of ModelNet40, the ScanObjectNN Uy et al. [2019] dataset incorporates a total of 14,298 manually collected point clouds in 15 object categories. Given the official data split in Uy et al. [2019], the training set comprises 11,416 samples, while the remaining 2,882 samples are utilized for testing purposes. To better evaluate our approaches, we conduct the open-set experiments under a more challenging case using the hardest perturbation variant of ScanObjectNN data Uy et al. [2019].

In this open-set study, we customize each dataset by manually dividing the original data into known and unknown classes based on the given semantic labels. For

open-set point cloud semantic segmentation, we borrow the idea of splitting classes in Cen et al. [2021a] where a few “thing” classes are chosen as unknown classes. Practically, in the “Manual-10-3” split of S3DIS, we choose three “thing” classes of “table,” “chair,” and “sofa” to form the unknown class, while in the “Manual-12-1” split, only “sofa” is selected as the unknown class. For open-set point cloud classification, we divide each original dataset (ModelNet40 Wu et al. [2015] or ScanObjectNN Uy et al. [2019]) into two splits, ModelNet40-*Split1* (or ScanObjectNN-*Split1*) and ModelNet40-*Split2* (or ScanObjectNN-*Split2*). In particular, we purposely separate classes with similar appearances in different splits to increase the task difficulty. Suppose the data chosen as the unknown class has attributes similar to the data pre-defined as the known classes; then, it is more challenging for the models to distinguish between the unknown and the known classes.

7.3.3 Evaluations

Both closed-set and open-set metrics are evaluated in our experiments. We choose mIoU and classification accuracy as the closed-set metrics. The standard open-set metrics for 2D tasks are included as open-set metrics: the false positive rate at 95% true positive rate (FPR at 95% TPR), the area under receiver operating characteristics (AUROC) Davis and Goadrich [2006]; Fawcett [2006] and the area under the precision-recall (AUPR) Manning and Schutze [1999]; Saito and Rehmsmeier [2015]. Such metrics assess the performance based on the overlap of unknown-class score distributions between the known and unknown classes. We hope to boost the models’ open-set performance while maintaining the closed-set classification accuracy. In other words, the model is expected to identify unknown objects and correctly classify known ones.

7.4 Methodology

7.4.1 Baseline Method

We adopt the training process of regular point cloud learning in the baseline models. Regarding the testing process, we borrow ideas from 2D tasks to compute the unknown-class scores. Specifically, we employ Maximum Softmax Probability (MSP) Hendrycks and Gimpel [2017] or Maximum Unnormalized Logit (MaxLogit) Hendrycks et al. [2022] to calculate an unknown-class score OS_i (or $OS_{cls,j}$) for each point (or sample), based on its corresponding confidence or logit vector in the classifier.

7.4.2 PointCaM Mechanism

Given a point cloud containing N points, we describe the 3D coordinates $\mathcal{P} \in \mathbb{R}^{N \times 3}$ as the geometric context, which is explicitly collected by scanners indicating the geometric distribution of points in the original 3D space. In addition, the feature context

$\mathcal{F} \in \mathbb{R}^{N \times C}$ of point cloud can be implicitly learned via CNN-based operations/modules from both local and global perspectives Yan et al. [2020]; Qiu et al. [2021b], representing the latent semantic information in a C -dimensional feature space. In general, the geometric context $\mathcal{P}$ and the feature context $\mathcal{F}$ are considered as two main properties of point cloud data Hu et al. [2020b]; Qiu et al. [2021a], benefiting comprehensive point feature representations for accurate visual analysis. Based on these two properties, we propose a novel self-supervised approach, the PointCaM mechanism, to enhance open-set point cloud learning.

The PointCaM mechanism is illustrated in Figure 7.1. Firstly, the Unknown-Point Simulator (UPS) or USS is applied to manipulate the geometric context partially, simulating unknown objects for training purposes. Secondly, a CNN-based Unknown-Point Estimator (UPE) or USE learns to discriminate between known and unknown objects by exploiting the feature context heavily. Compared to the baseline method mentioned above, our PointCaM not only additionally simulates OOD data for better training perception but also extracts more semantically meaningful feature information from the intermediate layers rather than the output classifiers.

7.4.2.1 Unknown-Point Simulator

Our main concern lies in simulating OOD objects, while only known data are available during the training process. As mentioned above, the geometric context of point cloud data indicates the original distribution of known objects in 3D space; if we alter the geometric context to a certain extent, part of known data would become unfamiliar to a CNN’s perception since the distribution of points partially changes. Using this approach, we propose Algorithm 1 to generate a partial geometric context.

Algorithm 1 Unknown-Point Simulator (UPS)

input: a point cloud $\mathcal{P} \in \mathbb{R}^{N \times 3}$
output: an augmented point cloud $\mathcal{Q} \in \mathbb{R}^{N \times 3}$
 # Randomly select a seed point s
 1. $s = \text{Rand}(\mathcal{P}) \in \mathbb{R}^3$
 # Find the k nearest neighbors of s
 2. $\mathcal{C} = \text{Knn}(s, \mathcal{P}) \in \mathbb{R}^{k \times 3}$
 # Augment $\mathcal{C}$ by rotation and translation
 3. $\mathcal{C}' = \text{Aug}(\mathcal{C}) = T + R \cdot \mathcal{C} \in \mathbb{R}^{k \times 3}$
 # Cut $\mathcal{C}$ from $\mathcal{P}$, and mix with $\mathcal{C}'$
 4. $\mathcal{Q} = \text{CutMix}(\mathcal{C}, \mathcal{P}, \mathcal{C}') \in \mathbb{R}^{N \times 3}$

At first, we randomly select (“**Rand**”) a point $s \in \mathbb{R}^3$ from original point cloud $\mathcal{P} \in \mathbb{R}^{N \times 3}$ as a seed point. Then, we search for the neighbors of s as a subset of points $\mathcal{C} \in \mathbb{R}^{k \times 3}$ that we aim to augment. In practice, we apply the k -nearest-neighbors (“**Knn**”) algorithm Wang et al. [2019c] to find a set of k neighbors based on the point-wise Euclidean-distances in 3D space. Particularly, the number of neighbors follows $k = \beta \cdot N$, where the selection ratio β is randomly drawn from $[\beta_{min}, \beta_{max}]$ for a point cloud of N points. An ablation study about the maximum selection ratio

$\beta_{max} \in (0, 1)$ is provided in Table 7.5. To augment (“**Aug**”) $\mathcal{C}$ as unknown points, we incorporate basic rotation and translation operations, considering two main benefits: (1) Compared to adding noise or scaling, it is a better way of simulating unknown objects as the natural local structures are preserved. This is evidenced by our experimental results in Table 7.4, where different regular transformations are performed to simulate unknown data points. Particularly, we observe that incorporating random rotation and translation can benefit the performance. The scaling operation shows a trivial effect while adding Gaussian noise, which causes inevitable damage to the local structures of an object and thus negatively impacts the performance. (2) Since only a subset of points ($k \ll N$) are modified with regular transformations while a majority of points remain unchanged, we ensure that for each point cloud sample, there is a proper proportion exists between known and unknown data points. This process is important to the 3D open-set settings: on the one hand, a small number of transformed points (*i.e.*, the ones simulated as “OOD data”) can help the model develop an additional capability of differentiating open-set data; on the other hand, a large number of unchanged points (*i.e.*, the ones used as “known data”) can help the model maintain its original capability of recognizing close-set data for regular 3D semantic segmentation and classification tasks. Specifically, $R \in \mathbb{R}^{3 \times 3}$ in Algorithm 1 is a general rotation matrix representing three degree-of-freedom (3-DoF) random rotations, and $T \in \mathbb{R}^3$ is a 3-DoF random vector representing a further translation within the borders of $\mathcal{P}$. Finally, we drop the partial points $\mathcal{C}$ in the original $\mathcal{P}$ and keep the transformed $\mathcal{C}'$ as the simulated OOD points, obtaining the partially altered geometric context $\mathcal{Q} \in \mathbb{R}^{N \times 3}$ that contains both known and unknown points (“**CutMix**”).

Using the UPS module, the network is still trained with a standard segmentation or classification loss, *i.e.*, $\mathcal{L} = \mathcal{L}_{task}$. In open-set point cloud semantic segmentation, the selected points are labeled $N_c + 1$, where N_c is the number of known classes. Similarly, in open-set point cloud classification, two samples of different known classes are partially cut, transformed, and mixed together to form a new sample labeled as $N_c + 1$, which introduces an additional class for unknown data in addition to the N_c known classes during training. The goal of UPS is to enrich the model’s knowledge of unknown objects, enhancing its sensitivity to such data. To achieve this, we generate unknown points in the data by manipulating parts of the original point cloud to ensure that the simulated OOD points have reasonable geometric features. Additionally, the rotation and translation of these points encourage an abnormal geometric distribution from the known points.

Notably, since the UPS module simulates OOD data as a new class for training, one more dimension is added to the classifier’s output accordingly.

7.4.2.2 Unknown-Point Estimator

To discriminate the known and unknown points in $\mathcal{Q}$, a CNN-based Unknown-Point Estimator (UPE) module learns to compute accurate unknown-class scores by exploiting the feature context in the point cloud. As depicted in Figure 7.2, given

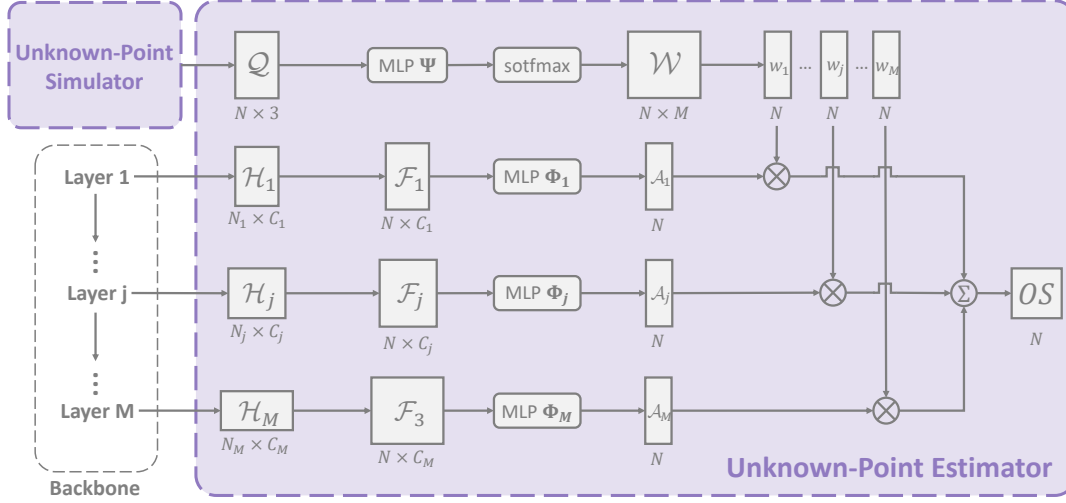


Figure 7.2: The detailed structure of the Unknown-Point Estimator (UPE). The figure illustrates the operations of the UPE module during the training process. For simplicity, we only demonstrate an example of fusing three intermediate feature maps. j is an arbitrary integer satisfying $1 \leq j \leq M$.

a backbone network, we denote the intermediate feature maps extracted from the backbone’s layers as $\{\mathcal{H}_1, \dots, \mathcal{H}_j, \dots, \mathcal{H}_M\}$, where M is the number of intermediate feature maps and $\mathcal{H}_j$ stands for an arbitrary one satisfying $1 \leq j \leq M$. Since a few point cloud networks Qi et al. [2017b]; Choy et al. [2019]; Zhao et al. [2021a] follow a U-Net Ronneberger et al. [2015] architecture to learn multi-level feature maps from lower point cloud resolutions (*i.e.*, $\mathcal{H}_j \in \mathbb{R}^{N_j \times C_j}$, where $N_j < N$), the extracted feature maps usually have different sizes. To restore the full feature context for all N points, we upsample $\{\mathcal{H}_1, \dots, \mathcal{H}_j, \dots, \mathcal{H}_M\}$ via the nearest neighbor interpolation method Hu et al. [2020b]. Then, the corresponding full-size feature context representations are obtained as $\{\mathcal{F}_1, \dots, \mathcal{F}_j, \dots, \mathcal{F}_M\}$, where $\forall \mathcal{F}_j \in \mathbb{R}^{N \times C_j}$.

Based on the feature context of $\{\mathcal{F}_1, \dots, \mathcal{F}_j, \dots, \mathcal{F}_M\}$, we intend to squeeze the latent information and estimate a score OS_i evaluating each point’s probability of whether it belongs to unknown data points. To achieve this, we apply a set of regular MLPs $\{\Phi_1, \dots, \Phi_j, \dots, \Phi_M\}$ to regress $\{\mathcal{F}_1, \dots, \mathcal{F}_j, \dots, \mathcal{F}_M\}$, respectively. For an arbitrary feature context $\mathcal{F}_j$, the estimated score vector is computed as $\mathcal{A}_j = \Phi_j(\mathcal{F}_j)$. Following similar computation, we acquire a set of score vectors $\{\mathcal{A}_1, \dots, \mathcal{A}_j, \dots, \mathcal{A}_M\}$, where $\forall \mathcal{A}_j \in \mathbb{R}^N$. Since the estimated score vectors are derived from different scales/levels of feature context representing a network’s hierarchical semantic information, it is necessary to synthesize them for a comprehensive prediction. Particularly, we leverage the altered geometric context Q to guide an adaptive fusion of $\{\mathcal{A}_1, \dots, \mathcal{A}_j, \dots, \mathcal{A}_M\}$, because the simulated OOD data points are implied by their anomalous geometric distributions in 3D space compared to those given (known) points. To be concrete, we utilize another MLP Ψ followed by a channel-wise soft-

max function to learn the weight vectors:

$$\mathcal{W} = [w_1; \dots; w_j; \dots; w_M] = \text{softmax}(\Psi(\mathcal{Q})), \quad (7.1)$$

where $\mathcal{W} \in \mathbb{R}^{N \times M}$, and $\forall w_j \in \mathbb{R}^N$ represents a weight vector split from the columns of $\mathcal{W}$. Finally, we fuse the score vectors for all points as a weighted sum:

$$\mathcal{OS} = \sum_{j=1}^M w_j \times \mathcal{A}_j, \quad \mathcal{OS} \in \mathbb{R}^N. \quad (7.2)$$

The output vector $\mathcal{OS}$'s i -th scalar, $\mathcal{OS}_i$, corresponds to our computed unknown-class score on the i -th point q_i of $\mathcal{Q}$, indicating its possibility of being a simulated unknown data point. When training the UPE module, in addition to a typical task loss $\mathcal{L}_{task}$, the model is also subject to a mean squared error (MSE) loss $\mathcal{L}_{upe}$ computed upon the UPE's output score vector $\mathcal{OS}$. The computation of $\mathcal{L}_{upe}$ is formulated as:

$$\mathcal{L}_{upe} = \frac{1}{N} \sum_{i=1}^N (\mathcal{OS}_i - \hat{\mathcal{OS}}_i)^2; \quad (7.3)$$

where the reference score $\hat{\mathcal{OS}}_i = 1$ if the point q_i is a simulated OOD data point; and $\hat{\mathcal{OS}}_i = 0$ if the point q_i belongs to one of the known classes. By minimizing the total loss $\mathcal{L} = \mathcal{L}_{task} + \alpha \cdot \mathcal{L}_{upe}$ (α is an empirical parameter studied in Table 7.6) during the training stage, we enable the UPE module to differentiate between the unknown and known data points. In the testing stage, we directly calculate the unknown-class scores $\mathcal{OS}$ following Equation (7.2), where the i -th scalar value $\mathcal{OS}_i$ predicts the possibility of $\mathcal{Q}$'s i -th point q_i belonging to the unknown class. For the semantic segmentation task, the unknown scores are calculated for all points for prediction. The classification task extracts an averaged unknown score over N points of the given sample for prediction.

UPE improves the open-set metric by investigating geometric dependencies and semantic connections between known and unknown points. To accomplish this, UPE employs an adaptive multi-level fusion module to distinguish between known and artificially generated unknown points. By examining point clouds and feature maps, UPE obtains more detailed information about the distinctions between known and unknown points. Compared to traditional multi-level prediction models Ronneberger et al. [2015]; Long et al. [2015]; Lee et al. [2015], which only employ the feature maps, our UPE leverages the 3D geometric context of the point cloud for an adaptive multi-level fusion. This helps to adaptively and effectively integrate the multi-level semantic information in feature space, given the guidance of point cloud geometric structures in 3D space.

7.5 Experiments

Experiments have been carried out on two basic open-set point cloud learning tasks: point cloud semantic segmentation and classification. We test methods on the customized point cloud datasets: *S3DIS-Split* for open-set 3D semantic segmentation; *ModelNet40-Split* and *ScanObjectNN-Split* for open-set 3D classification. In general, we present rich experimental data to indicate the feasibility of our open-set settings for point cloud learning. Coupled with more ablation studies, our PointCaM mechanism is comprehensively verified using different benchmarks and backbone networks.

7.5.1 Implementation

In open-set 3D semantic segmentation experiments, k points are rotated and translated to mimic unknown points. Note $k = \beta \cdot N$ where $\beta \in [\beta_{min}, \beta_{max}]$ is the random selection ratio, and N is the number of total points of the point cloud. PointNet and PointNet++ are trained with 64 epochs using Adam optimizer Kingma and Ba [2014]. The initial learning rate is set to 0.001. PointTransformer is trained with 100 epochs using SGD optimizer. The initial learning rate is 0.5. For open-set 3D classification, we first randomly choose samples to be augmented under an augmentation ratio of 0.1; then, for each chosen sample, we achieve the augmentation process by mixing it with k points that are randomly selected from another sample. Specifically, k points can be selected from another sample of N points ($k = \beta \cdot N$). Then, we augment and mix these k neighbor points with the chosen sample points. A mixed sample will be used as a sample from the unknown class. All classification models are trained with 300 epochs using an SGD optimizer. The number of total points, *i.e.*, N , is set to 4096 and 1024 for open-set point cloud semantic segmentation and classification, respectively. We run all experiments using the GPU of NVIDIA GeForce RTX 3090.

In open-set point cloud semantic segmentation, we establish the selection ratio range $[\beta_{min}, \beta_{max}]$ of UPS at $[0.0, 0.6]$. Unlike classification on the sample level, we set β_{min} as 0.0 in semantic segmentation to facilitate the representation of unknown points at various sizes on the pixel level. The selection ratio range for open-set point cloud classification is set at $[0.4, 0.6]$. We set β_{min} and β_{max} values around 0.5 to ensure that the mixed unknown samples vary significantly from the original point cloud samples. The hyperparameter α , which determines the effect of the UPE module, is set to 5.0 and 1.0 for segmentation and classification tasks, respectively.

7.5.2 Open-set 3D Semantic Segmentation

We first conduct open-set experiments on point cloud semantic segmentation, where the model intends to compute the unknown-class scores $\mathcal{X}$ for all testing points. Here, we test our approaches using three popular backbones: the classical PointNet/PointNet++ Qi et al. [2017a,b], and the advanced PointTransformer Zhao et al. [2021a].

Table 7.1: **Open-set Point Cloud Semantic Segmentation** on S3DIS-*Split* with PointNet, PointNet++ and PointTransformer. FPR (95% TPR), pixel-level AUPR, and AUROC in % are given. “PointCaM” is equal to “UPS+UPE”. β_{max} and α are set to 0.6 and 5.0, respectively.

Data Split	Backbone	Open-set Method	Closed-Set Metric	Open-set Metric		
			mIoU	FPR (95% TPR) ↓	AUPR ↑	AUROC ↑
Manual-12-1	PointNet Qi et al. [2017a]	MSP Hendrycks and Gimpel [2017]	48.8	47.1	0.9	81.3
		MaxLogits Hendrycks et al. [2022]	48.8	45.5	1.3	82.3
		MLUC Cen et al. [2021b]	46.2	42.4	1.2	82.9
		REAL Cen et al. [2022]	46.5	44.7	0.9	81.8
		UPS+MSP	47.1	47.4	1.1	80.5
		UPS+MaxLogits	47.1	45.3	1.0	81.4
		PointCaM (Ours)	46.5	40.9	0.9	83.1
	PointNet++ Qi et al. [2017b]	MSP Hendrycks and Gimpel [2017]	52.8	93.4	0.3	51.3
		MaxLogits Hendrycks et al. [2022]	52.8	92.7	0.3	48.8
		MLUC Cen et al. [2021b]	52.5	93.1	0.4	53.9
		REAL Cen et al. [2022]	53.7	93.1	0.3	51.9
		UPS+MSP	53.7	93.1	0.3	52.4
		UPS+MaxLogits	53.7	93.4	0.3	50.8
		PointCaM (Ours)	54.8	87.4	0.4	55.6
	PointTransformer Zhao et al. [2021a]	MSP Hendrycks and Gimpel [2017]	66.5	100.0	13.3	59.0
		MaxLogits Hendrycks et al. [2022]	66.5	99.8	13.4	59.3
		MLUC Cen et al. [2021b]	65.5	99.9	13.8	59.2
		REAL Cen et al. [2022]	67.5	99.8	14.4	62.0
		UPS+MSP	67.1	99.7	17.0	66.0
		UPS+MaxLogits	67.1	90.1	18.2	65.4
		PointCaM (Ours)	67.1	73.9	18.7	72.0
Manual-10-3	PointNet Qi et al. [2017a]	MSP Hendrycks and Gimpel [2017]	45.4	63.0	9.1	68.1
		MaxLogits Hendrycks et al. [2022]	45.4	63.0	8.5	65.7
		MLUC Cen et al. [2021b]	44.4	46.4	15.8	74.3
		REAL Cen et al. [2022]	43.2	57.9	9.9	71.6
		UPS+MSP	44.1	58.5	10.0	71.7
		UPS+MaxLogits	44.1	56.2	10.4	72.7
		PointCaM (Ours)	43.8	42.0	19.8	82.8
	PointNet++ Qi et al. [2017b]	MSP Hendrycks and Gimpel [2017]	48.7	93.4	6.5	52.6
		MaxLogits Hendrycks et al. [2022]	48.7	94.9	6.4	51.3
		MLUC Cen et al. [2021b]	47.4	93.4	7.7	57.2
		REAL Cen et al. [2022]	49.6	93.4	6.7	53.4
		UPS+MSP	49.3	93.4	6.6	53.3
		UPS+MaxLogits	49.3	93.4	6.9	54.1
		PointCaM (Ours)	49.5	84.7	7.8	60.6
	PointTransformer Zhao et al. [2021a]	MSP Hendrycks and Gimpel [2017]	59.7	100.0	17.0	52.8
		MaxLogits Hendrycks et al. [2022]	59.7	99.5	17.3	53.0
		MLUC Cen et al. [2021b]	59.2	99.9	17.6	54.5
		REAL Cen et al. [2022]	61.3	99.9	19.5	58.9
		UPS+MSP	60.0	99.8	19.5	59.0
		UPS+MaxLogits	60.0	93.9	23.4	59.8
		PointCaM (Ours)	61.6	74.1	26.5	70.7

For most cases in Table 7.1, we observe that the UPS module clearly outperforms the MSP and MaxLogits baselines since the points simulated by the UPS contribute to modeling the unknown point sets during training. Furthermore, the designed UPE module also boosts the open-set metrics by a large margin. For instance, on the “Manual-10-3” split with PointNet, UPS+MSP, and UPS+MaxLogits outperform basic MSP and MaxLogits by 3.6% and 7.0%, respectively, while a full PointCaM (UPS+UPE) mechanism achieves the best performance of 82.8%. When testing with PointTransformer, our PointCaM shows significant advantages over the basic MSP/-MaxLogits and their incorporation with UPS. For example, on the “Manual-10-3” of S3DIS-*Split*, we see PointCaM (UPS+UPE)’s $\sim 18\%$ and $\sim 12\%$ of AUROC improvements over MSP and UPS+MSP. Surprisingly, we notice that the network’s closed-set accuracies (mIoU) are simultaneously increased from 66.5/59.7% to 67.1%/61.6% with the help of the PointCaM. MLUC Cen et al. [2021b] outperforms MSP and MaxLogits with PointNet or PointNet++, but it achieves limited performance improvements using PointTransformer.

Table 7.2: **Open-set Point Cloud Classification** on ModelNet40-*Split*. The closed-set metrics, including classification accuracies over sample/class, and open-set metrics, including FPR (95% TPR), detection error, and image-level AUROC in %, are given. “PointCaM” is equal to “USS+USE”. ModelNet40-*Split2* is taken as the training dataset, while ModelNet40-*Split2* plus -*Split1* is the evaluation dataset.

Backbone	Open-set Method	Closed-Set Metric		Open-set Metric		
		Accuracy (sample)	Accuracy (class)	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑
PointNet Qi et al. [2017a]	MSP Hendrycks and Gimpel [2017]	96.1	94.3	49.9	17.6	89.0
	MLS Vaze et al. [2021]	96.1	94.3	50.7	16.9	88.8
	GradNorm Huang et al. [2021]	96.1	94.3	48.0	17.7	88.0
	USS+MSP	96.1	93.8	47.4	17.1	90.2
	PointCaM (Ours)	96.0	94.1	43.7	17.1	90.2
PointNet++ Qi et al. [2017b]	MSP Hendrycks and Gimpel [2017]	97.8	96.3	33.5	14.8	85.9
	MLS Vaze et al. [2021]	97.8	96.3	35.4	14.5	86.2
	GradNorm Huang et al. [2021]	97.8	96.3	34.3	17.2	82.0
	USS+MSP	97.7	96.2	36.4	15.2	85.9
	PointCaM (Ours)	97.7	96.3	35.4	15.6	86.4
DGCNN Wang et al. [2019c]	MSP Hendrycks and Gimpel [2017]	97.5	95.9	35.9	13.9	91.7
	MLS Vaze et al. [2021]	97.5	95.9	34.6	13.0	91.7
	GradNorm Huang et al. [2021]	97.5	95.9	35.8	13.7	90.9
	USS+MSP	97.4	96.0	35.1	13.6	91.9
	PointCaM (Ours)	97.7	96.3	35.1	13.7	92.5
AdaptConv Zhou et al. [2021a]	MSP Hendrycks and Gimpel [2017]	97.8	96.3	33.0	14.6	90.6
	MLS Vaze et al. [2021]	97.8	96.3	34.7	13.5	89.6
	GradNorm Huang et al. [2021]	97.8	96.3	34.9	14.8	90.6
	USS+MSP	97.9	96.4	33.1	14.4	91.8
	PointCaM (Ours)	97.9	96.6	33.6	14.2	91.9

In addition to the quantitative results above, we have the following observations: (1) Though PointNet++ and PointTransformer show superior performance in closed-set metrics, they have inferior open-set ability. It indicates that, in 3D semantic segmentation, complicated neural networks make overconfident errors and assign high-confidence predictions to unknown inputs. (2) For open-set problems, two key issues are about simulating the negative objects and exploring the spatial/semantic differences between positive and negative data. On the one hand, our UPS and UPE modules are designed to resolve each issue separately; on the other hand, since the UPE module relies on the UPS’s simulated objects to learn the discrepancies between unknown and known data, the proposed two modules also complement each other for accurate 3D open-set learning. (3) Table 7.1 shows that for PointNet on “Manual-12-1”, PointCaM does not outperform the baselines in AUPR. While this phenomenon is also observed in other works Hendrycks et al. [2022]; Cen et al. [2022], where a lower AUPR accompanies a higher AUROC, the underlying reason still requires further investigation. (4) Some visualization examples are compared and discussed in Figures 7.3 and 7.4. The example in Figure 7.4 showcases the density distributions of the scores for unknown-class samples. From the figure, it is evident that the separation between known and unknown points becomes more distinct when UPS is employed. That proves that UPS has the capability to simulate OOD points for the network, leading to an increased network sensitivity in identifying unknown points.

7.5.3 Open-set 3D Classification

Moreover, we transfer our methods to open-set point cloud classification experiments. In addition to PointNet and PointNet++, we utilize DGCNN Wang et al. [2019c] and AdaptConv Zhou et al. [2021a] as testing backbones.

Table 7.3: **Open-set Point Cloud Classification** on ScanObjectNN-*Split*. The closed-set metrics, including classification accuracies over sample/class, and open-set metrics, including FPR (95% TPR), detection error, and image-level AUROC in %, are given. “PointCaM” is equal to “USS+USE”. ScanObjectNN-*Split2* is taken as the training dataset, while ScanObjectNN-*Split2* plus -*Split1* is the evaluation dataset.

Backbone	Open-set Method	Closed-Set Metric		Open-set Metric		
		Accuracy (sample)	Accuracy (class)	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑
PointNet Qi et al. [2017a]	MSP Hendrycks and Gimpel [2017]	80.2	75.7	86.8	39.7	63.6
	MLS Vaze et al. [2021]	80.2	75.7	89.9	42.2	60.4
	GradNorm Huang et al. [2021]	80.2	75.7	87.6	39.4	64.0
	USS+MSP	79.3	75.5	90.5	40.5	63.3
	PointCaM (Ours)	79.6	75.6	86.0	38.6	65.5
PointNet++ Qi et al. [2017b]	MSP Hendrycks and Gimpel [2017]	88.2	87.1	80.6	30.6	72.2
	MLS Vaze et al. [2021]	88.2	87.1	80.2	32.3	71.4
	GradNorm Huang et al. [2021]	88.2	87.1	81.8	35.5	65.1
	USS+MSP	88.8	87.8	81.3	30.9	72.2
	PointCaM (Ours)	88.1	86.8	79.6	30.2	73.1
DGCNN Wang et al. [2019c]	MSP Hendrycks and Gimpel [2017]	88.0	86.2	84.3	29.9	75.3
	MLS Vaze et al. [2021]	88.0	86.2	79.3	30.3	75.5
	GradNorm Huang et al. [2021]	88.0	86.2	82.1	30.2	75.0
	USS+MSP	88.2	86.2	83.7	29.4	75.5
	PointCaM (Ours)	87.6	85.9	82.9	29.7	76.0
AdaptConv Zhou et al. [2021a]	MSP Hendrycks and Gimpel [2017]	87.5	85.5	86.7	31.1	73.4
	MLS Vaze et al. [2021]	87.5	85.5	85.0	30.5	74.4
	GradNorm Huang et al. [2021]	87.5	85.5	85.2	30.4	74.2
	USS+MSP	86.8	84.6	84.6	30.4	74.1
	PointCaM (Ours)	87.1	85.5	84.1	30.1	74.4

The quantitative results of open-set point cloud classification are reported in Tables 7.2 and 7.3, which show that the proposed methods consistently outperform the baseline method MSP in AUROC. For instance, compared to MSP and USS+MSP, the PointCaM (USS+USE) mechanism is marginally superior in AUROC on two datasets. It is worth noting that, in the ScanObjectNN-*Split* results of Table 7.3, the proposed USS+MSP and PointCaM (USS+USE) improve the open-set performance at a small cost to the closed-set accuracy.

7.5.4 Ablation Study

Unknown point generator. Here, we test multiple approaches to simulating unknown points: the single rotation within different angle ranges, the single translation within different ranges of the whole point cloud, the scaling, and the Gaussian noise. Simple rotation and translation can be considered special cases of our UPS when the translation vector is zero or the rotation matrix is identity. Results in Table 7.4 show that UPS can effectively provide a valuable baseline for open-set point cloud learning. Compared to other methods, UPS can generate unknown objects of natural and diverse appearances.

Sensitivity to parameters. There are two main hyper-parameters in the PointCaM mechanism: the maximum selection ratio β_{max} in the UPS module and the coefficient of the MSE loss α in the UPE module. From Tables 7.5 and 7.6, models show sensitivity to these two parameters. The results in Table 7.5 show that with growing β_{max} , the open-set recognition ability of the model increases. Given the testing performances of α shown in Table 7.6, we empirically set the value $\alpha = 5.0$ for our experiments using the PointCaM mechanism.

Fusion layers in UPE. In addition to UPS, which simulates unknown points, we

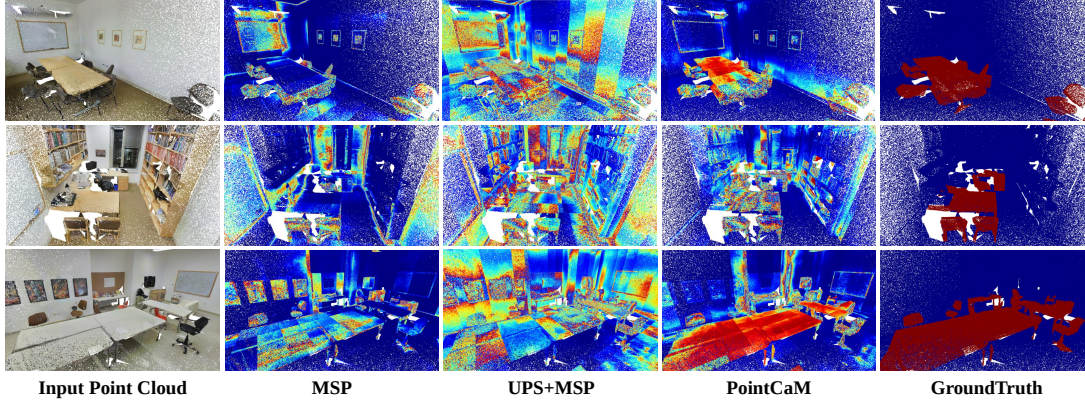


Figure 7.3: **The visualization examples: open-set point cloud semantic segmentation** of PointNet on S3DIS-*Split* with the “Manual-10-3” split. Under the “Manual-10-3” split, “table,” “chair” and “sofa” are taken as unknown classes. We visualize heatmaps of estimated unknown-class scores using different open-set methods, *i.e.*, MSP Hendrycks and Gimpel [2017], UPS+MSP, and our PointCaM (UPS+UPE). The “blue” and “red” points represent points from known and unknown classes. As we can observe in the second-column figures, the baseline MSP has the essential ability to recognize unknown objects. Using UPS, the model is endowed with superior perception ability for unknown points. For example, in the first-row figure, UPS+MSP identifies more parts of the “unknown table” than MSP. However, the UPS module also causes a side effect on the models’ recognition confidence in known classes. To address this issue, the UPE module exploits the point cloud’s feature context by adaptively fusing the multi-scale feature maps. With the help of both the UPS and UPE modules, our proposed PointCaM mechanism well distinguishes the known and unknown points, as shown in the fourth-column figures.

propose UPE to boost the open-set performance by exploiting the feature context in point clouds. In practice, UPE is designed to fuse the feature context from the layers at different levels. We have an ablation study for UPE in Table 7.7, showing that its performance improves when different levels of feature context are fused.

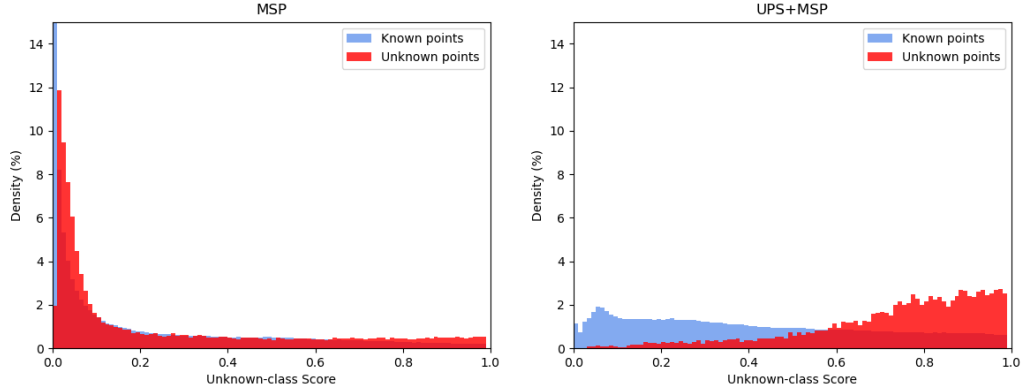


Figure 7.4: **The visualization examples: open-set point cloud semantic segmentation** of PointNet on S3DIS-*Split* with the “Manual-10-3” split. We visualize the density distributions of the unknown-class score of MSP and UPS+MSP.

Table 7.4: **The ablation study:** Unknown point generator. PointNet is performed in Open-set Point Cloud Semantic Segmentation on S3DIS-*Split*. The metric AUROC in % is given.

Data Split	Backbone	Open-set Method	AUROC $\uparrow$
Manual-10-3	PointNet [7]	MSP	68.1
		MSP+single rotation (0.5π)	69.9
		MSP+single rotation (π)	70.4
		MSP+single rotation (2π)	70.9
		MSP+single translation (0.25 range)	63.9
		MSP+single translation (0.5 range)	69.5
		MSP+single translation (whole range)	70.8
		MSP+scaling	67.3
		MSP+Gaussian noise	58.1
		MSP+UPS	71.7

7.5.5 Discussion

Number of unknown classes. The results on S3DIS-*Split* (see Table 7.1) imply the negative impacts of unknown data on closed-set accuracy: increasing the number of unknown classes causes a worse closed-set accuracy. Thus, the mIoU values of the “Manual-10-3” split are lower than the ones reported in the “Manual-12-1” split. This phenomenon is in line with the statements in Ma et al. [2018]; Dhamija et al. [2020a]. **The 2D open-set methods.** In Table 7.1, the incremental improvements produced by MSP Hendrycks and Gimpel [2017] and MaxLogits Hendrycks et al. [2022] demonstrate that the 2D-driven methods also work for 3D point clouds. However, the performances in 2D and 3D cases are not consistent: *e.g.*, MaxLogits usually performs better than MSP in 2D tasks as Hendrycks et al. [2022] verifies; while in our 3D experiments, the performance of MaxLogits is worse than MSP (see Table 7.1). Similar

Table 7.5: **The ablation study: β_{max} .** PointTransformer is performed in Open-set Point Cloud Semantic Segmentation on S3DIS-Split. The metric AUROC in % is given.

Data Split	Backbone	Open-set Method	β_{max}	AUROC $\uparrow$
Manual-12-1	PointTransformer Zhao et al. [2021a]	UPS+MSP	0.2	59.4
		UPS+MaxLogits		62.3
		UPS+MSP	0.3	62.8
		UPS+MaxLogits		64.7
		UPS+MSP	0.4	61.6
		UPS+MaxLogits		64.6
		UPS+MSP	0.5	63.5
		UPS+MaxLogits		65.1
		UPS+MSP	0.6	66.0
		UPS+MaxLogits		65.4
		UPS+MSP	0.7	65.5
		UPS+MaxLogits		65.0

Table 7.6: **The ablation study: α .** PointNet++ is performed in Open-set Point Cloud Semantic Segmentation on S3DIS-Split. The metric AUROC in % is given.

Data Split	Backbone	Open-set Method	α	AUROC $\uparrow$
Manual-10-3	PointNet++ Qi et al. [2017b]	PointCaM	0.2	56.8
			1.0	59.5
			2.5	58.6
			5.0	60.6
			7.5	57.0
			10.0	58.3
			12.5	56.6
			15.0	55.5

observations showing possible incompatibility between 2D and 3D-driven open-set methods can be found in Cen et al. [2022]. Directly adopting some 2D methods to 3D tasks might not be preferable due to point clouds’ unordered and unstructured characteristics. Thus, 3D-specific open-set methods such as our proposed PointCaM are worth further exploration.

Training time. In Table 7.8, we compare the training time and model size of various approaches with PointNet on “Manual-12-1”. The table shows that although there is a slight increase in model complexity, the substantial improvements in the open-set metric of our PointCaM model justify the extra complexity.

Performance drop. We conduct an experiment to investigate the cause of the performance drop, and the results are presented in Table 7.9. From the table, we observed that the open-set performance basically follows a similar trend to the closed-set performance. For instance, when AUROC drops significantly from 86.4% to 84.3%, the classification accuracy decreases by 0.5% from 97.7% to 97.2%. We also test more values of the augmentation ratio hyperparameter within a range of 0.01 to 0.2 and find that an increase in AUROC leads to slightly lower closed-set classification accuracy.

Table 7.7: **The ablation study:** Fusion layers in UPE. PointNet++ is performed in Open-set Point Cloud Semantic Segmentation on S3DIS-*Split*. The metric AUROC in % is given.

Data Split	Backbone	Open-set Method	Fusion Layer	AUROC $\uparrow$
Manual-10-3	PointNet Qi et al. [2017a]	PointCaM	-	68.1
			1st layer	78.8
			1st+2nd layers	81.2
			1st+2nd+3rd layers (w/o point-guided fusion)	79.9
			1st+2nd+3rd layers (with point-guided fusion)	82.8

Table 7.8: **Training time:** PointNet is performed in Open-set Point Cloud Semantic Segmentation on S3DIS-*Split*. The training time for each iteration and model size are provided.

Data Split	Backbone	Open-set Method	Model Complexity		Open-set Metric
			Training Time	Params	AUROC $\uparrow$
Manual-10-3	PointNet Qi et al. [2017a]	MSP Hendrycks and Gimpel [2017]	0.23s	41481K	68.1
		MaxLogits Hendrycks et al. [2022]	0.23s	41481K	65.7
		UPS+MSP	0.19s	41482K	71.7
		UPS+MaxLogits	0.19s	41482K	72.7
		PointCaM (Ours)	0.24s	41501K	82.8

Similar phenomena have been observed in prior studies Chen et al. [2021]; Cen et al. [2022]. We hypothesize that as the model’s ability to distinguish unknown samples improves (indicated by an increase in AUROC), it becomes more cautious and makes more conservative decisions, *i.e.*, misclassifying some known samples as unknown ones. This hypothesis is further supported by the visualization in Figure 7.4, where a more distinct separation between known and unknown samples leads to lower confidence scores for the known samples. Additionally, in this case, the FPR (95% TPR) and Detection Error exhibit a similar trend to the classification accuracy.

7.6 Summary

This chapter discusses point cloud learning under the open-set settings. Two main point cloud visual tasks, semantic segmentation and classification, have been deeply investigated. To address the open-set point cloud problems, we propose a PointCaM mechanism to enhance a model’s capability of identifying unknown objects. Particularly, the UPS module helps to simulate the OOD data in the training stage, while the UPE module aims to exploit the fundamental context of point cloud data for accurate unknown-class score computation. By conducting extensive experiments and comprehensive ablation studies, we expect to inspire more future work on open-set point cloud problems.

Here, we conclude the second part of the thesis, which has been dedicated to exploring solutions for visual open-set recognition problems. We have methodically analyzed data modalities, transitioning from 2D images in Chapters 5 and 6 to 3D

Table 7.9: **Performance drop:** Open-set Point Cloud Classification of PointNet++ on ModelNet40-Split with different ratios of using samples as unknown ones.

Backbone	Ratio	Closed-Set Metric	Open-set Metric		
		Accuracy (sample)	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑
PointNet++ Qi et al. [2017b]	0.01	97.9	33.4	14.7	85.9
	0.05	97.8	35.4	15.5	86.2
	0.1	97.7	35.4	15.6	86.4
	0.2	97.8	34.2	15.0	86.1
	0.3	97.4	42.0	16.2	84.8
	0.5	97.2	43.0	17.1	84.3

point clouds in Chapter 7. Moreover, Chapters 5 and 7 provide various approaches to addressing the problems, whereas Chapter 6 introduces a novel task within the field. Specifically, Chapters 5 and 7 adopt curved geometry, data augmentation, or multi-layer learning for feature learning, while Chapter 6 defines GOSS task upon traditional OSS.

Conclusion

This thesis has done five works toward pursuing deep-learning models or tasks capable of effectively handling limited-data visual problems. Across its chapters, the thesis contributes to the theoretical landscape of deep-learning visual models in scenarios of limited training data. By exploring diverse methodologies, from attention mechanisms to geometric learning, and building new tasks, like generalized open-set semantic segmentation, the thesis offers a comprehensive view of the potential and adaptability of deep-learning models in handling complex limited-data visual problems. We develop algorithms for five sub-problems: few-shot learning, audio-visual zero-shot learning, open-set recognition, generalized open-set semantic segmentation (GOSS), and open-set 3D point cloud learning, in Chapters 3- 7, respectively. The methodologies and insights gained might motivate further innovations in the field of deep computer vision. Here, the main contributions of this thesis are summarized as follows:

- **Advancing low-shot learning tasks.** The thesis commences with a new approach to few-shot learning by introducing the Reinforced-Attention Policy (RAP) module (see Chapter 3). This novel architecture significantly refines feature maps in few-shot learning models, leading to more discriminative feature representations of unknown samples. It demonstrates the utility of reinforcement learning in iteratively focusing on informative features. Progressing to audio-visual zero-shot learning in Chapter 4, the research innovatively applies hyperbolic geometry. The proposed approach, leveraging the unique property of the hyperbolic geometry in encoding hierarchical structures, provides a new pathway to generate more distinctive middle feature embeddings in a multimodal context, demonstrating substantial improvements in cross-modality alignment.
- **Advancing open-set tasks.** The thesis also addresses the complex domain of open-set recognition, effectively utilizing curved spaces to enhance visual feature learning, as introduced in Chapter 5. By introducing novel geometric networks, geometric-in-one (GiO) and geometric-in-two (GiT) networks, the research sets a precedent in employing curved geometries for open-set tasks, showing superior capabilities of curved feature embeddings in distinguishing

between known and unknown visual samples. Another milestone is the development of GOSS, a newly-defined task that extends beyond identifying to processing unknown regions (see Chapter 6). This innovative task of classifying and clustering unknown pixels within images provides a better solution for pixel-level scene understanding.

- **Pioneering in point cloud learning.** The thesis concludes with work into the relatively unexplored territory of open-set 3D point cloud learning in Chapter 7. Introducing the Point Cut-and-Mix (PointCaM) mechanism, including the Unknown-Point Simulator (UPS) and Unknown-Point Estimator (UPE) modules, leads to an improved performance in discerning geometric differences between known and unknown features of point cloud objects. This approach not only enhances model capabilities but also complements the setting for open-set point cloud learning.
- **Practical implications.** In domains where recognizing a broad spectrum of objects, including those not known during training, is crucial, such as autonomous systems Pomerleau et al. [2015], advanced remote sensing Li et al. [2012], and interactive technologies Chen et al. [2019b], the findings of this thesis might provide references for future developments.

In addition to the contributions we have in the thesis, we also discuss the potential future research directions as follows:

- **Adaptive curved space.** In this thesis, we explore the application of curved geometry to address limited-data visual problems across two chapters (see Chapters 4 and 5). Despite the progress made, determining the optimal curvature for the curved space tailored to specific applications is still a challenge. Therefore, future endeavors could focus on refining the algorithms related to curved geometry. The goal could be to develop a method that can automatically search for the most suitable curvature value, thereby significantly improving the learned features. This advancement would hold the potential to not only enhance the overall effectiveness of applying curved geometry in solving limited-data visual problems but also streamline the application process.
- **Foundation model.** Recently, the emergence of foundation models such as Contrastive Language-Image Pre-Training (CLIP) Radford et al. [2021] and Segment Anything Model (SAM) Kirillov et al. [2023] has marked a significant advancement in the field of deep computer vision. These models, renowned for their powerful generalization capabilities and rich prior knowledge, are particularly well-suited to tackle limited-data visual problems. Given their inherent strengths, it is logical to consider these foundation models as valuable tools in addressing challenges associated with limited-data scenarios. Consequently, a promising direction for future research involves exploring the application of the foundation models to solve visual problems under limited-data scenarios.

In summary, this thesis contributes to the field of deep computer vision, particularly in the context of limited-data visual problems. The innovative approaches and tasks developed throughout this thesis not only address several sub-problems but also possibly inspire future advancements. The works through low-shot learning and open-set recognition provide reliable solutions in the face of limited-data challenges. The findings of this thesis are also expected to inspire further research and development, pushing the boundaries of what is possible in the realm of visual problems with limited training data. As more intelligent visual systems are used in complex environments, the methodologies and insights from this thesis might serve for future explorations in the field of deep computer vision.

Bibliography

- ABATI, D.; PORRELLO, A.; CALDERARA, S.; AND CUCCHIARA, R., 2019. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 50 and 63)
- AFOURAS, T.; CHUNG, J. S.; SENIOR, A.; VINYALS, O.; AND ZISSERMAN, A., 2018. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 44, 12 (2018), 8717–8727. (cited on page 32)
- AFOURAS, T.; OWENS, A.; CHUNG, J. S.; AND ZISSERMAN, A., 2020. Self-supervised learning of audio-visual objects from video. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII* 16, 208–224. Springer. (cited on page 32)
- ARASLANOV, N.; ROTHKOPF, C. A.; AND ROTH, S., 2019. Actor-critic instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8237–8246. (cited on page 18)
- ARBELAEZ, P.; MAIRE, M.; FOWLKES, C.; AND MALIK, J., 2010. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33, 5 (2010), 898–916. (cited on pages 71, 72, 73, and 74)
- ARMENI, I.; SAX, S.; ZAMIR, A. R.; AND SAVARESE, S., 2017. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv:1702.01105*, (2017). (cited on pages 86 and 90)
- ASANO, Y.; PATRICK, M.; RUPPRECHT, C.; AND VEDALDI, A., 2020. Labelling unlabelled videos from scratch with multi-modal self-supervision. *Advances in Neural Information Processing Systems*, 33 (2020), 4660–4671. (cited on page 32)
- ATIGH, M. G.; SCHOEP, J.; ACAR, E.; VAN NOORD, N.; AND METTES, P., 2022. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4453–4462. (cited on page 35)
- BADAMDORJ, T.; ROCHAN, M.; WANG, Y.; AND CHENG, L., 2021. Joint visual and audio learning for video highlight detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8127–8137. (cited on page 32)
- BAUR, C.; WIESTLER, B.; ALBARQOUNI, S.; AND NAVAB, N., 2018. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI Brainlesion Workshop*, 161–169. Springer. (cited on page 61)

-
- BEHLEY, J.; GARBADE, M.; MILIOTO, A.; QUENZEL, J.; BEHNKE, S.; STACHNISS, C.; AND GALL, J., 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9297–9307. (cited on page 86)
- BENDALE, A. AND BOULT, T. E., 2016. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 1, 3, and 11)
- BERGMAN, L. AND HOSHEN, Y., 2020. Classification-based anomaly detection for general data. In *ICLR*. (cited on page 51)
- BERGMANN, P.; FAUSER, M.; SATTLEGGGER, D.; AND STEGER, C., 2019. Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9592–9600. (cited on pages xx, xxvi, 60, 61, 63, 64, 65, and 68)
- BERGMANN, P.; FAUSER, M.; SATTLEGGGER, D.; AND STEGER, C., 2020. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4183–4192. (cited on pages 51, 55, and 56)
- BERGMANN, P.; LÖWE, S.; FAUSER, M.; SATTLEGGGER, D.; AND STEGER, C., 2018. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, (2018). (cited on page 62)
- BEVANDIĆ, P.; KREŠO, I.; ORŠIĆ, M.; AND ŠEGVIĆ, S., 2022. Dense open-set recognition based on training with noisy negative images. *Image and Vision Computing*, (2022), 104490. (cited on page 70)
- BHARDWAJ, A.; PIMPALE, S.; KUMAR, S.; AND BANERJEE, B., 2021. Empowering knowledge distillation via open set recognition for robust 3d point cloud classification. *Pattern Recognition Letters*, 151 (2021), 172–179. (cited on pages 4, 86, and 88)
- BLEYER, M.; ROTHER, C.; KOHLI, P.; SCHARSTEIN, D.; AND SINHA, S., 2011. Object stereo—joint stereo matching and object segmentation. In *CVPR 2011*, 3081–3088. (cited on page 73)
- BRONSTEIN, A. M.; BRONSTEIN, M. M.; AND KIMMEL, R., 2005. Expression-invariant face recognition via spherical embedding. In *IEEE International Conference on Image Processing 2005*, vol. 3, III–756. IEEE. (cited on page 52)
- BRONSTEIN, M. M.; BRUNA, J.; LECUN, Y.; SZLAM, A.; AND VANDERGHEYNST, P., 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34, 4 (2017), 18–42. (cited on page 51)
- CABA HEILBRON, F.; ESCORCIA, V.; GHANEM, B.; AND CARLOS NIEBLES, J., 2015. Activitynet: A large-scale video benchmark for human activity understanding. In

-
- Proceedings of the IEEE conference on computer vision and pattern recognition*, 961–970. (cited on pages 33 and 38)
- CAESAR, H.; UIJLINGS, J.; AND FERRARI, V., 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1209–1218. (cited on pages xxvi, 69, 71, 79, and 80)
- CAICEDO, J. C. AND LAZEBNIK, S., 2015. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, 2488–2496. (cited on page 18)
- CEN, J.; YUN, P.; CAI, J.; WANG, M. Y.; AND LIU, M., 2021a. Deep metric learning for open world semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 15333–15342. (cited on pages 70, 72, 80, 81, 82, and 91)
- CEN, J.; YUN, P.; CAI, J.; WANG, M. Y.; AND LIU, M., 2021b. Open-set 3d object detection. In *2021 International Conference on 3D Vision (3DV)*, 869–878. IEEE. (cited on pages 86, 87, 88, 89, and 97)
- CEN, J.; YUN, P.; ZHANG, S.; CAI, J.; LUAN, D.; WANG, M. Y.; LIU, M.; AND TANG, M., 2022. Open-world semantic segmentation for lidar point clouds. *arXiv preprint arXiv:2207.01452*, (2022). (cited on pages 86, 87, 89, 97, 98, 102, and 103)
- CHAMI, I.; GU, A.; NGUYEN, D. P.; AND RÉ, C., 2021. Horopca: Hyperbolic dimensionality reduction via horospherical projections. In *International Conference on Machine Learning*, 1419–1429. PMLR. (cited on page 33)
- CHAMI, I.; YING, Z.; RÉ, C.; AND LESKOVEC, J., 2019. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, 4869–4880. (cited on page 35)
- CHAN, R.; LIS, K.; UHLEMAYER, S.; BLUM, H.; HONARI, S.; SIEGWART, R.; SALZMANN, M.; FUA, P.; AND ROTTMANN, M., 2021a. Segmentmeifyoucan: A benchmark for anomaly segmentation. In *Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*. (cited on page 70)
- CHAN, R.; ROTTMANN, M.; AND GOTTSCHALK, H., 2021b. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5128–5137. (cited on page 72)
- CHANG, A. X.; FUNKHOUSER, T.; GUIBAS, L.; HANRAHAN, P.; HUANG, Q.; LI, Z.; SAVARESE, S.; SAVVA, M.; SONG, S.; SU, H.; ET AL., 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, (2015). (cited on page 86)

- CHEN, B.; WANG, D.; LI, P.; WANG, S.; AND LU, H., 2018a. Real-time'actor-critic'tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 318–334. (cited on page 18)
- CHEN, C.; MAJUMDER, S.; AL-HALAH, Z.; GAO, R.; RAMAKRISHNAN, S. K.; AND GRAUMAN, K., 2020a. Learning to set waypoints for audio-visual navigation. *arXiv preprint arXiv:2008.09622*, (2020). (cited on page 32)
- CHEN, G.; PENG, P.; WANG, X.; AND TIAN, Y., 2021. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 11 (2021), 8065–8081. (cited on page 103)
- CHEN, H.; XIE, W.; VEDALDI, A.; AND ZISSERMAN, A., 2020b. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 721–725. IEEE. (cited on pages 33 and 38)
- CHEN, L.; LU, J.; SONG, Z.; AND ZHOU, J., 2018b. Part-activated deep reinforcement learning for action prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 421–436. (cited on page 18)
- CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K.; AND YUILLE, A. L., 2017a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40, 4 (2017), 834–848. (cited on page 69)
- CHEN, L.-C.; PAPANDREOU, G.; SCHROFF, F.; AND ADAM, H., 2017b. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, (2017). (cited on pages 69 and 76)
- CHEN, L.-C.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F.; AND ADAM, H., 2018c. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818. (cited on pages 69 and 75)
- CHEN, W.-Y.; LIU, Y.-C.; KIRA, Z.; WANG, Y.-C.; AND HUANG, J.-B., 2019a. A closer look at few-shot classification. In *International Conference on Learning Representations*. (cited on pages xxiii, 24, and 26)
- CHEN, Y.; WANG, Q.; CHEN, H.; SONG, X.; TANG, H.; AND TIAN, M., 2019b. An overview of augmented reality technology. In *Journal of Physics: Conference Series*, vol. 1237, 022082. IOP Publishing. (cited on pages 4, 85, and 106)
- CHENG, B.; GIRSHICK, R.; DOLLÁR, P.; BERG, A. C.; AND KIRILLOV, A., 2021a. Boundary iou: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15334–15342. (cited on page 73)

-
- CHENG, B.; MISRA, I.; SCHWING, A. G.; KIRILLOV, A.; AND GIRDHAR, R., 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1290–1299. (cited on page 69)
- CHENG, B.; SCHWING, A.; AND KIRILLOV, A., 2021b. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34 (2021). (cited on page 69)
- CHERAGHIAN, A.; RAHMAN, S.; RAMASINGHE, S.; FANG, P.; SIMON, C.; PETERSSON, L.; AND HARANDI, M., 2021. Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In *ICCV*, 8661–8670. (cited on page 51)
- CHOY, C.; GWAK, J.; AND SAVARESE, S., 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*. (cited on pages 86 and 94)
- CHU, W.-H. AND WANG, Y.-C. F., 2018. Learning semantics-guided visual attention for few-shot image classification. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2979–2983. IEEE. (cited on pages 16 and 18)
- COATES, A.; NG, A.; AND LEE, H., 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 215–223. (cited on pages xxiii, 23, and 27)
- COMANICIU, D. AND MEER, P., 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24, 5 (2002), 603–619. (cited on pages 71 and 72)
- CORBIÈRE, C.; THOME, N.; BAR-HEN, A.; CORD, M.; AND PÉREZ, P., 2019. Addressing failure prediction by learning model confidence. In *NeurIPS*. (cited on pages 50 and 77)
- CORDTS, M.; OMRAN, M.; RAMOS, S.; REHFELD, T.; ENZWEILER, M.; BENENSON, R.; FRANKE, U.; ROTH, S.; AND SCHIELE, B., 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223. (cited on pages xxvi, 69, 71, 79, and 80)
- CRAMER, J.; WU, H.-H.; SALAMON, J.; AND BELLO, J. P., 2019. Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3852–3856. IEEE. (cited on page 32)
- DAVIS, J. AND GOADRICH, M., 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, 233–240. (cited on pages 57, 73, and 91)
- DE BRABANDERE, B.; NEVEN, D.; AND VAN GOOL, L., 2017. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, (2017). (cited on page 78)

- DEAN, V.; TULSIANI, S.; AND GUPTA, A., 2020. See, hear, explore: Curiosity via audio-visual association. *Advances in Neural Information Processing Systems*, 33 (2020), 14961–14972. (cited on page 32)
- DEHAENE, D.; FRIGO, O.; COMBREXELLE, S.; AND ELINE, P., 2020. Iterative energy-based projection on a normal data manifold for anomaly localization. In *ICLR*. (cited on page 62)
- DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; AND FEI-FEI, L., 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. IEEE. (cited on pages xxv, xxvi, 1, 57, 60, 62, 63, and 80)
- DEVRIES, T. AND TAYLOR, G. W., 2018. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, (2018). (cited on pages 50, 77, and 81)
- DHAMIJA, A.; GUNTHER, M.; VENTURA, J.; AND BOULT, T., 2020a. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1021–1030. (cited on pages xvii, 1, 2, 79, 86, and 101)
- DHAMIJA, A.; GUNTHER, M.; VENTURA, J.; AND BOULT, T., 2020b. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. (cited on page 11)
- DHINGRA, B.; SHALLUE, C. J.; NOROUZI, M.; DAI, A. M.; AND DAHL, G. E., 2018. Embedding text in hyperbolic spaces. *arXiv preprint arXiv:1806.04313*, (2018). (cited on page 48)
- DI BIASE, G.; BLUM, H.; SIEGWART, R.; AND CADENA, C., 2021. Pixel-wise anomaly detection in complex driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16918–16927. (cited on page 72)
- DING, Y.; XU, Y.; ZHANG, S.-X.; CONG, Y.; AND WANG, L., 2020. Self-supervised learning for audio-visual speaker diarization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4367–4371. IEEE. (cited on page 31)
- DONG, H.; CHEN, Z.; YUAN, M.; XIE, Y.; ZHAO, J.; YU, F.; DONG, B.; AND ZHANG, L., 2022. Region-aware metric learning for open world semantic segmentation via meta-channel aggregation. In *31th International Joint Conference on Artificial Intelligence (IJCAI-22)*. (cited on pages 87 and 89)
- DWIBEDI, D.; MISRA, I.; AND HEBERT, M., 2017. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on page 89)

-
- EIGEN, D. AND FERGUS, R., 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, 2650–2658. (cited on page 73)
- ENDRES, F.; HESS, J.; STURM, J.; CREMERS, D.; AND BURGARD, W., 2013. 3-d mapping with an rgb-d camera. *Transactions on Robotics*, 30, 1 (2013), 177–187. (cited on page 85)
- EVERINGHAM, M.; VAN GOOL, L.; WILLIAMS, C. K.; WINN, J.; AND ZISSERMAN, A., 2010. The pascal visual object classes (voc) challenge. *IJCV*, (2010). (cited on pages 73 and 79)
- FAN, X.; JIANG, W.; LUO, H.; AND FEI, M., 2019. Spherrereid: Deep hypersphere manifold embedding for person re-identification. *Journal of Visual Communication and Image Representation*, 60 (2019), 51–58. (cited on page 48)
- FANG, P.; HARANDI, M.; AND PETERSSON, L., 2021. Kernel methods in hyperbolic spaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10665–10674. (cited on pages 3, 33, 48, and 52)
- FAWCETT, T., 2006. An introduction to roc analysis. *Pattern recognition letters*, 27, 8 (2006), 861–874. (cited on pages 57, 73, and 91)
- FELZENSZWALB, P. F. AND HUTTENLOCHER, D. P., 2004. Efficient graph-based image segmentation. *International journal of computer vision*, 59, 2 (2004), 167–181. (cited on pages 71 and 72)
- FINN, C.; ABBEEL, P.; AND LEVINE, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1126–1135. JMLR. org. (cited on pages xxiii, 1, 2, 3, 10, 15, 24, 25, 26, and 28)
- FOURNIER, H.; ISMAIL, A.; AND VIGNERON, A., 2015. Computing the gromov hyperbolicity of a discrete metric space. *Information Processing Letters*, 115, 6-8 (2015), 576–579. (cited on pages 34 and 42)
- GAL, Y. AND GHAHRAMANI, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR. (cited on page 61)
- GAN, C.; ZHANG, Y.; WU, J.; GONG, B.; AND TENENBAUM, J. B., 2020. Look, listen, and act: Towards audio-visual embodied navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 9701–9707. IEEE. (cited on page 32)
- GANEVA, O.; BÉCIGNEUL, G.; AND HOFMANN, T., 2018. Hyperbolic neural networks. In *Advances in neural information processing systems*, 5345–5355. (cited on page 35)

- GAO, R.; OH, T.-H.; GRAUMAN, K.; AND TORRESANI, L., 2020. Listen to look: Action recognition by previewing audio. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10457–10467. (cited on page 32)
- GAO, Z.; WU, Y.; HARANDI, M.; AND JIA, Y., 2022. Curvature-adaptive meta-learning for fast adaptation to manifold data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 2 (2022), 1545–1562. (cited on pages 10 and 37)
- GAO, Z.; WU, Y.; JIA, Y.; AND HARANDI, M., 2021. Curvature generation in curved spaces for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8691–8700. (cited on page 38)
- GHIASI, G.; CUI, Y.; SRINIVAS, A.; QIAN, R.; LIN, T.-Y.; CUBUK, E. D.; LE, Q. V.; AND ZOPH, B., 2021. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2918–2928. (cited on page 89)
- GIDARIS, S. AND KOMODAKIS, N., 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4367–4375. (cited on page 24)
- GIDARIS, S. AND KOMODAKIS, N., 2019. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 21–30. (cited on page 24)
- GIDARIS, S.; SINGH, P.; AND KOMODAKIS, N., 2018. Unsupervised representation learning by predicting image rotations. In *ICLR*. (cited on pages 51, 57, 59, and 60)
- GIRSHICK, R., 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448. (cited on page 1)
- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; AND MALIK, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587. (cited on page 1)
- GOLAN, I. AND EL-YANIV, R., 2018. Deep anomaly detection using geometric transformations. In *NeurIPS*. (cited on pages 51 and 59)
- GONG, D.; LIU, L.; LE, V.; SAHA, B.; MANSOUR, M. R.; VENKATESH, S.; AND HENGEL, A. v. D., 2019. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1705–1714. (cited on page 50)
- GRADY, L., 2006. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28, 11 (2006), 1768–1783. (cited on page 72)
- GU, A.; SALA, F.; GUNEL, B.; AND RÉ, C., 2019. Learning mixed-curvature representations in product spaces. In *ICLR*. (cited on pages 7 and 52)

-
- GULCEHRE, C.; DENIL, M.; MALINOWSKI, M.; RAZAVI, A.; PASCANU, R.; HERMANN, K. M.; BATTAGLIA, P.; BAPST, V.; RAPOSO, D.; SANTORO, A.; AND DE FREITAS, N., 2019. Hyperbolic attention networks. In *International Conference on Learning Representations*. (cited on page 35)
- GUO, M.-H.; CAI, J.-X.; LIU, Z.-N.; MU, T.-J.; MARTIN, R. R.; AND HU, S.-M., 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7, 2 (2021), 187–199. (cited on page 88)
- GUO, Y.; GUO, H.; AND YU, S. X., 2022. Co-sne: Dimensionality reduction and visualization for hyperbolic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21–30. (cited on page 33)
- GUO, Y.; WANG, H.; HU, Q.; LIU, H.; LIU, L.; AND BENNAMOUN, M., 2020. Deep learning for 3d point clouds: A survey. *TPAMI*, (2020). (cited on pages 85 and 88)
- HAN, J.; FANG, P.; LI, W.; HONG, J.; ARMIN, M. A.; REID, I.; PETERSSON, L.; AND LI, H., 2022. You only cut once: Boosting data augmentation with a single cut. In *International Conference on Machine Learning*, 8196–8212. PMLR. (cited on page 89)
- HANE, C.; ZACH, C.; COHEN, A.; ANGST, R.; AND POLLEFEYS, M., 2013. Joint 3d scene reconstruction and class segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 97–104. (cited on page 73)
- HARLEY, A. W.; DERPANIS, K. G.; AND KOKKINOS, I., 2017. Segmentation-aware convolutional networks using local attention masks. In *Proceedings of the IEEE International Conference on Computer Vision*, 5038–5047. (cited on page 78)
- HARRIS, E.; MARCU, A.; PAINTER, M.; NIRANJAN, M.; PRÜGEL-BENNETT, A.; AND HARE, J., 2020. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*, (2020). (cited on page 89)
- HE, K.; ZHANG, X.; REN, S.; AND SUN, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778. (cited on pages xxv, 1, 25, 60, and 80)
- HENDRYCKS, D.; BASART, S.; MAZEIKA, M.; ZOU, A.; KWON, J.; MOSTAJABI, M.; STEINHARDT, J.; AND SONG, D., 2022. Scaling out-of-distribution detection for real-world settings. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 8759–8773. PMLR. (cited on pages xxv, 4, 11, 50, 60, 61, 70, 72, 77, 79, 81, 82, 84, 91, 97, 98, 101, and 103)
- HENDRYCKS, D. AND GIMPEL, K., 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*. (cited on pages xxii, 50, 52, 55, 57, 58, 60, 61, 77, 81, 82, 84, 91, 97, 98, 99, 100, 101, and 103)
- HENDRYCKS, D.; MAZEIKA, M.; KADAVATH, S.; AND SONG, D., 2019. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*. (cited on pages 51 and 59)

-
- HILLIARD, N.; PHILLIPS, L.; HOWLAND, S.; YANKOV, A.; CORLEY, C. D.; AND HODAS, N. O., 2018. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376*, (2018). (cited on page 23)
- HONG, J.; FANG, P.; LI, W.; HAN, J.; PETERSSON, L.; AND HARANDI, M., 2023a. Curved geometric networks for visual anomaly recognition. *IEEE Transactions on Neural Networks and Learning Systems*, (2023). (cited on page 47)
- HONG, J.; FANG, P.; LI, W.; ZHANG, T.; SIMON, C.; HARANDI, M.; AND PETERSSON, L., 2021. Reinforced attention for few-shot learning and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 913–923. (cited on page 15)
- HONG, J.; HAYDER, Z.; HAN, J.; FANG, P.; HARANDI, M.; AND PETERSSON, L., 2023b. Hyperbolic audio-visual zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7873–7883. (cited on page 31)
- HONG, J.; LI, W.; HAN, J.; ZHENG, J.; FANG, P.; HARANDI, M.; AND PETERSSON, L., 2023c. Goss: Towards generalized open-set semantic segmentation. *The Visual Computer*, (2023), 1–14. (cited on page 69)
- HONG, J.; QIU, S.; LI, W.; ANWAR, S.; HARANDI, M.; BARNES, N.; AND PETERSSON, L., 2023d. Pointcam: Cut-and-mix for open-set point cloud learning. *arXiv preprint arXiv:2212.02011*, (2023). (cited on page 85)
- HOU, R.; CHANG, H.; BINGPENG, M.; SHAN, S.; AND CHEN, X., 2019. Cross attention network for few-shot classification. In *Advances in Neural Information Processing Systems*, 4005–4016. (cited on pages 16, 18, and 25)
- Hsu, Y.-C.; SHEN, Y.; JIN, H.; AND KIRA, Z., 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10951–10960. (cited on page 50)
- HU, D.; LI, X.; MOU, L.; JIN, P.; CHEN, D.; JING, L.; ZHU, X.; AND DOU, D., 2020a. Cross-task transfer for geotagged audiovisual aerial scene recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, 68–84. Springer. (cited on page 32)
- HU, J.; SHEN, L.; AND SUN, G., 2018a. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141. (cited on pages 17, 28, and 29)
- HU, Q.; YANG, B.; XIE, L.; ROSA, S.; GUO, Y.; WANG, Z.; TRIGONI, N.; AND MARKHAM, A., 2020b. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11108–11117. (cited on pages 86, 92, and 94)

-
- HU, R.; DOLLÁR, P.; HE, K.; DARRELL, T.; AND GIRSHICK, R., 2018b. Learning to segment every thing. In *CVPR*. (cited on page 79)
- HU, Y.; HUGONOT, J.; FUA, P.; AND SALZMANN, M., 2019. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3385–3394. (cited on page 73)
- HUANG, G.; LIU, Z.; VAN DER MAATEN, L.; AND WEINBERGER, K. Q., 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708. (cited on pages xxv, 25, 57, and 59)
- HUANG, R.; GENG, A.; AND LI, Y., 2021. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34 (2021), 677–689. (cited on pages 98 and 99)
- HWANG, J.; OH, S. W.; LEE, J.-Y.; AND HAN, B., 2021. Exemplar-based open-set panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1175–1184. (cited on page 73)
- HWANG, J.-J.; YU, S. X.; SHI, J.; COLLINS, M. D.; YANG, T.-J.; ZHANG, X.; AND CHEN, L.-C., 2019. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. (cited on page 78)
- ISAACS, O.; SHAYER, O.; AND LINDENBAUM, M., 2020. Enhancing generic segmentation with learned region representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12946–12955. (cited on pages 71 and 72)
- JABOYEDOFF, M.; OPPIKOFE, T.; ABELLÁN, A.; DERRON, M.-H.; LOYE, A.; METZGER, R.; AND PEDRAZZINI, A., 2012. Use of lidar in landslide investigations: a review. *Natural hazards*, 61, 1 (2012), 5–28. (cited on page 85)
- JAFARI, O. H.; GROTH, O.; KIRILLOV, A.; YANG, M. Y.; AND ROTHER, C., 2017. Analyzing modular cnn architectures for joint depth prediction and semantic segmentation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 4620–4627. IEEE. (cited on page 73)
- JANG, J. AND KIM, C. O., 2022. Collective decision of one-vs-rest networks for open-set recognition. *IEEE Transactions on Neural Networks and Learning Systems*, (2022), 1–12. doi:10.1109/TNNLS.2022.3189996. (cited on page 50)
- JAYASUMANA, S.; RAMALINGAM, S.; AND KUMAR, S., 2022. Model-efficient deep learning with kernelized classification. <https://openreview.net/forum?id=30SXt3-vvnM>. (cited on page 52)
- JOSEPH, K.; KHAN, S.; KHAN, F. S.; AND BALASUBRAMANIAN, V. N., 2021. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5830–5840. (cited on pages 11 and 79)

- JUNG, S.; LEE, J.; GWAK, D.; CHOI, S.; AND CHOO, J., 2021. Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15425–15434. (cited on pages 72 and 81)
- KAZAKOS, E.; NAGRANI, A.; ZISSERMAN, A.; AND DAMEN, D., 2019. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5492–5501. (cited on page 32)
- KHRULKOV, V.; MIRVAKHABOVA, L.; USTINOVA, E.; OSELEDETS, I.; AND LEMPITSKY, V., 2020. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6418–6428. (cited on pages xvii, 3, 8, 9, 24, 33, 34, 35, 42, 48, 52, and 54)
- KIM, W.; KANEZAKI, A.; AND TANAKA, M., 2020. Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Transactions on Image Processing*, 29 (2020), 8055–8068. (cited on pages xxvii and 82)
- KINGMA, D. P. AND BA, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014). (cited on pages 23, 80, and 96)
- KIRILLOV, A.; GIRSHICK, R.; HE, K.; AND DOLLÁR, P., 2019a. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6399–6408. (cited on page 73)
- KIRILLOV, A.; HE, K.; GIRSHICK, R.; ROTHER, C.; AND DOLLÁR, P., 2019b. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9404–9413. (cited on pages 73 and 74)
- KIRILLOV, A.; MINTUN, E.; RAVI, N.; MAO, H.; ROLLAND, C.; GUSTAFSON, L.; XIAO, T.; WHITEHEAD, S.; BERG, A. C.; LO, W.-Y.; DOLLÁR, P.; AND GIRSHICK, R., 2023. Segment anything. *arXiv:2304.02643*, (2023). (cited on page 106)
- KOCH, G.; ZEMEL, R.; AND SALAKHUTDINOV, R., 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, vol. 2. Lille. (cited on page 15)
- KOKKINOS, I., 2017. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6129–6138. (cited on page 73)
- KONG, S. AND RAMANAN, D., 2021. Opengan: Open-set recognition via open data generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 813–822. (cited on pages 50, 72, and 86)

-
- KORBAR, B.; TRAN, D.; AND TORRESANI, L., 2018. Cooperative learning of audio and video models from self-supervised synchronization. *Advances in Neural Information Processing Systems*, 31 (2018). (cited on page 32)
- KRIZHEVSKY, A.; HINTON, G.; ET AL., 2009. Learning multiple layers of features from tiny images. *Toronto, ON, Canada*, (2009). (cited on pages xx, xxv, xxvi, 23, 57, 58, 59, 60, 64, and 66)
- KRIZHEVSKY, A.; SUTSKEVER, I.; AND HINTON, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25 (2012). (cited on page 1)
- KRULL, A.; BRACHMANN, E.; NOWOZIN, S.; MICHEL, F.; SHOTTON, J.; AND ROTHER, C., 2017. Poseagent: Budget-constrained 6d object pose estimation via reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6702–6710. (cited on page 18)
- LAN, X.; WANG, H.; GONG, S.; AND ZHU, X., 2017. Deep reinforcement learning attention selection for person re-identification. *arXiv preprint arXiv:1707.02785*, (2017). (cited on page 18)
- LECUN, Y.; BOTTOU, L.; BENGIO, Y.; AND HAFFNER, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 11 (1998), 2278–2324. (cited on pages xxiii and 27)
- LEE, C.-Y.; XIE, S.; GALLAGHER, P.; ZHANG, Z.; AND TU, Z., 2015. Deeply-supervised nets. In *Artificial intelligence and statistics*, 562–570. PMLR. (cited on page 95)
- LEE, D.; LEE, J.; LEE, J.; LEE, H.; LEE, M.; WOO, S.; AND LEE, S., 2021. Regularization strategy for point cloud via rigidly mixed sample. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15900–15909. (cited on page 89)
- LEE, J. M., 2006. *Riemannian manifolds: an introduction to curvature*, vol. 176. Springer Science & Business Media. (cited on page 7)
- LEE, K.; LEE, H.; LEE, K.; AND SHIN, J., 2018. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*. (cited on page 50)
- LEE, K.; MAJI, S.; RAVICHANDRAN, A.; AND SOATTO, S., 2019. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10657–10665. (cited on page 24)
- LI, C.-L.; SOHN, K.; YOON, J.; AND PFISTER, T., 2021. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9664–9674. (cited on pages 51, 63, and 89)

- LI, W.; FARAZI, M.; HONG, J.; AND PETERSSON, L., 2023. Rein: Reusing imagenet to improve open-set object detection. In *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 523–530. IEEE. (cited on page 11)
- LI, W.; GUO, Q.; JAKUBOWSKI, M. K.; AND KELLY, M., 2012. A new method for segmenting individual trees from the lidar point cloud. *Photogrammetric Engineering & Remote Sensing*, 78, 1 (2012), 75–84. (cited on pages 85 and 106)
- LI, X.; GUIVANT, J.; AND KHAN, S., 2020. Real-time 3d object proposal generation and classification using limited processing resources. *Robotics and Autonomous Systems*, 130 (2020), 103557. (cited on page 85)
- LI, X. AND GUIVANT, J. E., 2023. Efficient and accurate object detection with simultaneous classification and tracking under limited computing power. *IEEE Transactions on Intelligent Transportation Systems*, (2023). (cited on page 85)
- LI, X.; SUN, Q.; LIU, Y.; ZHOU, Q.; ZHENG, S.; CHUA, T.-S.; AND SCHIELE, B., 2019. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, 10276–10286. (cited on page 24)
- LI, Y.; WANG, L.; YANG, T.; AND GONG, B., 2018. How local is the local diversity? reinforcing sequential determinantal point processes with dynamic ground sets for supervised video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 151–167. (cited on page 18)
- LIANG, S.; LI, Y.; AND SRIKANT, R., 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*. (cited on pages 11, 12, 50, 57, 58, and 59)
- LILICRAP, T. P.; HUNT, J. J.; PRITZEL, A.; HEES, N.; EREZ, T.; TASSA, Y.; SILVER, D.; AND WIERSTRA, D., 2016. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*. (cited on page 18)
- LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; AND ZITNICK, C. L., 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer. (cited on pages 1 and 79)
- LIN, Y.-B.; TSENG, H.-Y.; LEE, H.-Y.; LIN, Y.-Y.; AND YANG, M.-H., 2021. Exploring cross-video and cross-modality signals for weakly-supervised audio-visual video parsing. *Advances in Neural Information Processing Systems*, 34 (2021), 11449–11461. (cited on page 31)
- LIS, K.; NAKKA, K.; FUA, P.; AND SALZMANN, M., 2019. Detecting the unexpected via image resynthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2152–2161. (cited on pages 4, 50, 70, 72, and 79)
- LIU, B.; CAO, Y.; LIN, Y.; LI, Q.; ZHANG, Z.; LONG, M.; AND HU, H., 2020a. Negative margin matters: Understanding margin in few-shot classification. *arXiv preprint arXiv:2003.12060*, (2020). (cited on pages xvii, 11, 24, 25, and 26)

-
- LIU, S.; CHEN, J.; PAN, L.; NGO, C.-W.; CHUA, T.-S.; AND JIANG, Y.-G., 2020b. Hyperbolic visual embedding learning for zero-shot recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9273–9281. (cited on pages 1, 10, 33, 35, 48, and 52)
- LIU, W.; LI, R.; ZHENG, M.; KARANAM, S.; WU, Z.; BHANU, B.; RADKE, R. J.; AND CAMPS, O., 2020c. Towards visually explaining variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8642–8651. (cited on pages 51, 60, and 62)
- LIU, W.; RABINOVICH, A.; AND BERG, A. C., 2015. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, (2015). (cited on page 80)
- LIU, W.; WEN, Y.; YU, Z.; LI, M.; RAJ, B.; AND SONG, L., 2017a. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212–220. (cited on pages 48, 52, and 54)
- LIU, W.; ZHANG, Y.-M.; LI, X.; YU, Z.; DAI, B.; ZHAO, T.; AND SONG, L., 2017b. Deep hyperspherical learning. *arXiv preprint arXiv:1711.03189*, (2017). (cited on page 52)
- LIU, Y.; FAN, B.; XIANG, S.; AND PAN, C., 2019. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8895–8904. (cited on page 88)
- LIU, Y.; JIANG, P.-T.; PETROSYAN, V.; LI, S.-J.; BIAN, J.; 0001, L. Z.; AND CHENG, M.-M., 2018a. Del: Deep embedding learning for efficient image segmentation. In *IJCAI*, vol. 864, 870. (cited on page 72)
- LIU, Y.; LEE, J.; PARK, M.; KIM, S.; YANG, E.; HWANG, S. J.; AND YANG, Y., 2018b. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv preprint arXiv:1805.10002*, (2018). (cited on page 24)
- LIU, Y.; LI, Z.; PAN, S.; GONG, C.; ZHOU, C.; AND KARYPIS, G., 2021. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*, 33, 6 (2021), 2378–2392. (cited on page 51)
- LONG, J.; SHELHAMER, E.; AND DARRELL, T., 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440. (cited on pages 69 and 95)
- LONG, T.; METTES, P.; SHEN, H. T.; AND SNOEK, C. G. M., 2020. Searching for actions on the hyperbole. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (cited on page 35)
- MA, H.; XIONG, R.; WANG, Y.; KODAGODA, S.; AND SHI, L., 2018. Towards open-set semantic labeling in 3d point clouds: Analysis on the unknown class. *Neurocomputing*, 275 (2018), 1282–1294. (cited on pages 4, 86, 88, and 101)

- MAJUMDER, S.; CHEN, C.; AL-HALAH, Z.; AND GRAUMAN, K., 2022. Few-shot audio-visual learning of environment acoustics. *arXiv preprint arXiv:2206.04006*, (2022). (cited on page 32)
- MANGLA, P.; KUMARI, N.; SINHA, A.; SINGH, M.; KRISHNAMURTHY, B.; AND BALASUBRAMANIAN, V. N., 2020. Charting the right manifold: Manifold mixup for few-shot learning. In *The IEEE Winter Conference on Applications of Computer Vision*, 2218–2227. (cited on page 26)
- MANNING, C. AND SCHUTZE, H., 1999. *Foundations of statistical natural language processing*. MIT press. (cited on pages 57, 73, and 91)
- MAO, H.; VENKATAKRISHNAN, S. B.; SCHWARZKOPF, M.; AND ALIZADEH, M., 2018. Variance reduction for reinforcement learning in input-driven environments. *arXiv preprint arXiv:1807.02264*, (2018). (cited on page 19)
- MARTIN, D.; FOWLKES, C.; TAL, D.; AND MALIK, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, 416–423. IEEE. (cited on page 73)
- MARTIN, D. R.; FOWLKES, C. C.; AND MALIK, J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26, 5 (2004), 530–549. (cited on page 74)
- MASSOLI, F. V.; FALCHI, F.; KANTARCI, A.; AKTI, Ş.; EKENEL, H. K.; AND AMATO, G., 2021. Mocca: Multilayer one-class classification for anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, (2021). (cited on pages 51 and 54)
- MATHE, S.; PIRINEN, A.; AND SMINCHISESCU, C., 2016. Reinforcement learning for visual object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2894–2902. (cited on page 18)
- MATURANA, D. AND SCHERER, S., 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 922–928. IEEE. (cited on page 88)
- MAZUMDER, P.; SINGH, P.; PARIDA, K. K.; AND NAMBOODIRI, V. P., 2021. Avgzslnet: Audio-visual generalized zero-shot learning by reconstructing label features from multi-modal embeddings. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3090–3099. (cited on pages 32, 33, 34, 35, 39, and 40)
- MEILA, M., 2005. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning*, 577–584. (cited on page 74)

-
- MERCEA, O.-B.; HUMMEL, T.; KOEPKE, A. S.; AND AKATA, Z., 2022a. Temporal and cross-modal attention for audio-visual zero-shot learning. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX*, 488–505. Springer. (cited on pages 3, 32, 33, 34, and 35)
- MERCEA, O.-B.; RIESCH, L.; KOEPKE, A.; AND AKATA, Z., 2022b. Audio-visual generalised zero-shot learning with cross-modal attention and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10553–10563. (cited on pages xxiv, 3, 32, 33, 34, 35, 38, 39, 40, 41, and 42)
- MITTAL, H.; MORGADO, P.; JAIN, U.; AND GUPTA, A., 2022. Learning state-aware visual representations from audible interactions. *arXiv preprint arXiv:2209.13583*, (2022). (cited on page 32)
- MNIH, V.; HEES, N.; GRAVES, A.; ET AL., 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, 2204–2212. (cited on pages 17, 18, and 19)
- MNIH, V.; KAVUKCUOGLU, K.; SILVER, D.; RUSU, A. A.; VENESS, J.; BELLEMARE, M. G.; GRAVES, A.; RIEDMILLER, M.; FIDJELAND, A. K.; OSTROVSKI, G.; ET AL., 2015. Human-level control through deep reinforcement learning. *Nature*, 518, 7540 (2015), 529–533. (cited on page 18)
- MORGADO, P.; MISRA, I.; AND VASCONCELOS, N., 2021. Robust audio-visual instance discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12934–12945. (cited on page 32)
- NEAL, L.; OLSON, M.; FERN, X.; WONG, W.-K.; AND LI, F., 2018. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 613–628. (cited on page 89)
- NICKEL, M. AND KIELA, D., 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30 (2017), 6338–6347. (cited on pages xvii, 8, 9, and 48)
- OWENS, A.; WU, J.; McDERMOTT, J. H.; FREEMAN, W. T.; AND TORRALBA, A., 2016. Ambient sound provides supervision for visual learning. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I* 14, 801–816. Springer. (cited on page 32)
- PARIDA, K.; MATIYALI, N.; GUHA, T.; AND SHARMA, G., 2020. Coordinated joint multimodal embeddings for generalized audio-visual zero-shot classification and retrieval of videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3251–3260. (cited on pages 32, 33, 34, 35, 39, and 40)
- PARK, J.; CHO, J.; CHANG, H. J.; AND CHOI, J. Y., 2021. Unsupervised hyperbolic representation learning via message passing auto-encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5516–5526. (cited on page 48)

-
- PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; ET AL., 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32 (2019), 8026–8037. (cited on pages 23 and 81)
- PENG, W.; VARANKA, T.; MOSTAFA, A.; SHI, H.; AND ZHAO, G., 2021. Hyperbolic deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2021). (cited on page 34)
- PERERA, P.; NALLAPATI, R.; AND XIANG, B., 2019. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2898–2906. (cited on page 50)
- PHAM, H.; DAI, Z.; GHIASI, G.; KAWAGUCHI, K.; LIU, H.; YU, A. W.; YU, J.; CHEN, Y.-T.; LUONG, M.-T.; WU, Y.; ET AL., 2021. Combined scaling for open-vocabulary image classification. *arXiv e-prints*, (2021), arXiv-2111. (cited on page 12)
- PIDHORSKYI, S.; ALMOHSEN, R.; AND DORETTO, G., 2018. Generative probabilistic novelty detection with adversarial autoencoders. *Advances in neural information processing systems*, 31 (2018). (cited on pages 11 and 12)
- PINHEIRO, P. O.; COLLOBERT, R.; AND DOLLÁR, P., 2015. Learning to segment object candidates. In *NeurIPS*. (cited on page 79)
- POMERLEAU, F.; COLAS, F.; SIEGWART, R.; ET AL., 2015. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4, 1 (2015), 1–104. (cited on pages 4, 85, and 106)
- PU, J.; PANAGAKIS, Y.; PETRIDIS, S.; AND PANTIC, M., 2017. Audio-visual object localization and separation using low-rank and sparsity. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2901–2905. IEEE. (cited on page 31)
- QI, C. R.; LITANY, O.; HE, K.; AND GUIBAS, L. J., 2019. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, 9277–9286. (cited on page 86)
- QI, C. R.; SU, H.; MO, K.; AND GUIBAS, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 652–660. (cited on pages 4, 85, 86, 88, 90, 96, 97, 98, 99, and 103)
- QI, C. R.; YI, L.; SU, H.; AND GUIBAS, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 5099–5108. (cited on pages 4, 85, 86, 88, 90, 94, 96, 97, 98, 99, 102, and 104)
- QIAN, R.; HU, D.; DINKEL, H.; WU, M.; XU, N.; AND LIN, W., 2020. Multiple sound sources localization from coarse to fine. In *European Conference on Computer Vision*, 292–308. Springer. (cited on page 31)

-
- QIU, S.; ANWAR, S.; AND BARNES, N., 2021a. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 24 (2021), 1943–1955. (cited on pages 86 and 92)
- QIU, S.; ANWAR, S.; AND BARNES, N., 2021b. Pnp-3d: A plug-and-play for 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2021). (cited on page 92)
- RADFORD, A.; KIM, J. W.; HALLACY, C.; RAMESH, A.; GOH, G.; AGARWAL, S.; SASTRY, G.; ASKELL, A.; MISHKIN, P.; CLARK, J.; ET AL., 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR. (cited on pages 12 and 106)
- RAND, W. M., 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66, 336 (1971), 846–850. (cited on page 74)
- RAVI, S. AND LAROCHELLE, H., 2017. Optimization as a model for few-shot learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*. (cited on pages xix, xxiii, 15, 23, 24, 25, and 49)
- REISS, T.; COHEN, N.; BERGMAN, L.; AND HOSHEN, Y., 2021. Panda: Adapting pre-trained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2806–2814. (cited on page 63)
- REN, J.; LIU, P. J.; FERTIG, E.; SNOEK, J.; POPLIN, R.; DEPRISTO, M. A.; DILLON, J. V.; AND LAKSHMINARAYANAN, B., 2019a. Likelihood ratios for out-of-distribution detection. In *NeurIPS*. (cited on page 50)
- REN, L.; YUAN, X.; LU, J.; YANG, M.; AND ZHOU, J., 2018. Deep reinforcement learning with iterative shift for visual tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 684–700. (cited on page 18)
- REN, M.; LIAO, R.; FETAYA, E.; AND ZEMEL, R., 2019b. Incremental few-shot learning with attention attractor networks. In *Advances in Neural Information Processing Systems*, 5276–5286. (cited on pages 16, 18, and 25)
- RODRÍGUEZ, P.; LARADJI, I.; DROUIN, A.; AND LACOSTE, A., 2020. Embedding propagation: Smoother manifold for few-shot classification. *arXiv preprint arXiv:2003.04151*, (2020). (cited on pages 24 and 26)
- RONNEBERGER, O.; FISCHER, P.; AND BROX, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer. (cited on pages 94 and 95)
- RUFF, L.; VANDERMEULEN, R.; GOERNITZ, N.; DEECKE, L.; SIDDIQUI, S. A.; BINDER, A.; MÜLLER, E.; AND KLOFT, M., 2018. Deep one-class classification. In *International conference on machine learning*, 4393–4402. PMLR. (cited on pages 51 and 59)

-
- SABOKROU, M.; POURREZA, M.; FAYYAZ, M.; ENTEZARI, R.; FATHY, M.; GALL, J.; AND ADELI, E., 2018. Avid: Adversarial visual irregularity detection. In *Asian Conference on Computer Vision*, 488–505. Springer. (cited on pages 50 and 62)
- SAITO, T. AND REHMSMEIER, M., 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10, 3 (2015), e0118432. (cited on pages 57, 73, and 91)
- SALEHI, M.; MIRZAEI, H.; HENDRYCKS, D.; LI, Y.; ROHBAN, M. H.; AND SABOKROU, M., 2022. A unified survey on anomaly, novelty, open-set, and out of-distribution detection: Solutions and future challenges. *Transactions of Machine Learning Research*, (2022). (cited on pages 2 and 12)
- SALEHI, M.; SADJADI, N.; BASELIZADEH, S.; ROHBAN, M. H.; AND RABIEE, H. R., 2021. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14902–14912. (cited on pages xxvi, 51, 55, 56, 62, and 63)
- SCHEIRER, W. J.; DE REZENDE ROCHA, A.; SAPKOTA, A.; AND BOULT, T. E., 2012. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35, 7 (2012), 1757–1772. (cited on pages 1, 2, 3, and 11)
- SCHLEGL, T.; SEEBÖCK, P.; WALDSTEIN, S. M.; SCHMIDT-ERFURTH, U.; AND LANGS, G., 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, 146–157. Springer. (cited on page 62)
- SENOCAK, A.; OH, T.-H.; KIM, J.; YANG, M.-H.; AND KWEON, I. S., 2018. Learning to localize sound source in visual scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4358–4366. (cited on page 31)
- SEVILLA-LARA, L.; SUN, D.; JAMPANI, V.; AND BLACK, M. J., 2016. Optical flow with semantic segmentation and localized layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3889–3898. (cited on page 73)
- SHAO, J.; JIANG, Y.; WANG, G.; LI, Z.; AND JI, X., 2020. Pfrl: Pose-free reinforcement learning for 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11454–11463. (cited on page 18)
- SHEN, J.; XIAO, Z.; ZHEN, X.; AND ZHANG, L., 2021. Spherical zero-shot learning. *IEEE Transactions on Circuits and Systems for Video Technology*, (2021). (cited on page 48)
- SHI, J. AND MALIK, J., 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 8 (2000), 888–905. (cited on pages 71 and 72)
- SHOTTON, J.; WINN, J.; ROTHER, C.; AND CRIMINISI, A., 2006. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision*, 1–15. (cited on page 69)

-
- SIMON, C.; KONIUSZ, P.; NOCK, R.; AND HARANDI, M., 2020. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4136–4145. (cited on page 24)
- SIMONYAN, K. AND ZISSERMAN, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, (2014). (cited on pages xxiii and 27)
- SKOPEK, O.; GANEA, O.-E.; AND BÉCIGNEUL, G., 2020. Mixed-curvature variational autoencoders. In *ICLR*. (cited on pages 7, 48, 52, and 63)
- SNELL, J.; SWERSKY, K.; AND ZEMEL, R., 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, 4077–4087. (cited on pages xxiii, 3, 15, 24, 25, 26, and 28)
- SOHN, K., 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, 1857–1865. (cited on page 78)
- SOOMRO, K.; ZAMIR, A. R.; AND SHAH, M., 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, (2012). (cited on page 38)
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; AND SALAKHUTDINOV, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15, 1 (2014), 1929–1958. (cited on page 2)
- SU, H.; MAJI, S.; KALOGERAKIS, E.; AND LEARNED-MILLER, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, 945–953. (cited on page 88)
- SU, J.-C.; MAJI, S.; AND HARIHARAN, B., 2019. When does self-supervision improve few-shot learning? *arXiv preprint arXiv:1910.03560*, (2019). (cited on page 24)
- SUN, D.; SUDDERTH, E. B.; AND BLACK, M. J., 2012. Layered segmentation and optical flow estimation over time. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 1768–1775. (cited on page 73)
- SUNG, F.; YANG, Y.; ZHANG, L.; XIANG, T.; TORR, P. H.; AND HOSPEDALES, T. M., 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1199–1208. (cited on pages 15 and 26)
- SZELISKI, R., 2022. *Computer vision: algorithms and applications*. Springer Nature. (cited on page 72)
- TACK, J.; MO, S.; JEONG, J.; AND SHIN, J., 2020. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *NeurIPS*. (cited on pages 51, 57, 59, and 60)

- TANG, Y.; TIAN, Y.; LU, J.; LI, P.; AND ZHOU, J., 2018. Deep progressive reinforcement learning for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5323–5332. (cited on page 18)
- TIFREA, A.; BÉCIGNEUL, G.; AND GANEA, O.-E., 2018. Poincaré glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, (2018). (cited on page 48)
- TORRALBA, A. AND EFROS, A. A., 2011. Unbiased look at dataset bias. In *CVPR 2011*, 1521–1528. IEEE. (cited on page 2)
- TUNG, F. AND MORI, G., 2019. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1365–1374. (cited on page 37)
- UY, M. A.; PHAM, Q.-H.; HUA, B.-S.; NGUYEN, T.; AND YEUNG, S.-K., 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1588–1597. (cited on pages 90 and 91)
- VAN DER MAATEN, L. AND HINTON, G., 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9, 11 (2008). (cited on pages xix, 34, and 49)
- VAZE, S.; HAN, K.; VEDALDI, A.; AND ZISSERMAN, A., 2021. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*. (cited on pages 98 and 99)
- VEERAMACHENENI, L. AND VALDENEGRO-TORO, M., 2022. A benchmark for out of distribution detection in point cloud 3d semantic segmentation. In *NeurIPS 2022 Workshop on Robot Learning: Trustworthy Robotics*. (cited on page 12)
- VENKATARAMANAN, S.; PENG, K.-C.; SINGH, R. V.; AND MAHALANOBIS, A., 2020. Attention guided anomaly localization in images. In *European Conference on Computer Vision*, 485–503. Springer. (cited on pages xxvi, 51, 62, and 63)
- VINYALS, O.; BLUNDELL, C.; LILLICRAP, T.; WIERSTRA, D.; ET AL., 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29 (2016), 3630–3638. (cited on pages 1, 2, 10, 16, 18, 25, and 26)
- VOJIR, T.; ŠÍPKA, T.; ALJUNDI, R.; CHUMERIN, N.; REINO, D. O.; AND MATAS, J., 2021. Road anomaly detection by partial image reconstruction with segmentation coupling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15651–15660. (cited on page 72)
- VYAS, A.; JAMMALAMADAKA, N.; ZHU, X.; DAS, D.; KAUL, B.; AND WILLKE, T. L., 2018. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 550–564. (cited on pages 51, 57, 58, and 59)

-
- WAH, C.; BRANSON, S.; WELINDER, P.; PERONA, P.; AND BELONGIE, S., 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology. (cited on pages xxiii, 23, 25, 26, 28, and 29)
- WAN, J.; LIU, Y.; WEI, D.; BAI, X.; AND XU, Y., 2020. Super-bpd: Super boundary-to-pixel direction for fast image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9253–9262. (cited on pages 71, 72, 77, and 79)
- WANG, P.; LIU, L.; SHEN, C.; HUANG, Z.; VAN DEN HENGEL, A.; AND TAO SHEN, H., 2017a. Multi-attention network for one shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2721–2729. (cited on pages 16 and 18)
- WANG, S.; WU, L.; CUI, L.; AND SHEN, Y., 2021a. Glancing at the patch: Anomaly localization with global and local feature comparison. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 254–263. (cited on pages 51, 55, and 63)
- WANG, T.-C.; MALLYA, A.; AND LIU, M.-Y., 2021b. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10039–10049. (cited on page 32)
- WANG, W.; YU, R.; HUANG, Q.; AND NEUMANN, U., 2018. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2569–2578. (cited on page 86)
- WANG, W.; ZHOU, T.; YU, F.; DAI, J.; KONUKOGLU, E.; AND VAN GOOL, L., 2021c. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 7303–7313. (cited on pages 4 and 78)
- WANG, X.; GIRSHICK; GUPTA, A.; AND HE, K., 2017b. Non-local Neural Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 17)
- WANG, X.; HUANG, Q.; CELIKYILMAZ, A.; GAO, J.; SHEN, D.; WANG, Y.-F.; WANG, W. Y.; AND ZHANG, L., 2019a. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6629–6638. (cited on page 18)
- WANG, Y.; CHAO, W.-L.; WEINBERGER, K. Q.; AND VAN DER MAATEN, L., 2019b. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, (2019). (cited on page 26)
- WANG, Y.; SUN, Y.; LIU, Z.; SARMA, S. E.; BRONSTEIN, M. M.; AND SOLOMON, J. M., 2019c. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38, 5 (2019), 1–12. (cited on pages 88, 92, 98, and 99)

- WILLIAMS, R. J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8, 3-4 (1992), 229–256. (cited on pages 3 and 22)
- WONG, K.; WANG, S.; REN, M.; LIANG, M.; AND URTASUN, R., 2020. Identifying unknown instances for autonomous driving. In *Conference on Robot Learning*, 384–393. PMLR. (cited on pages 86, 87, 88, and 89)
- WOO, S.; PARK, J.; LEE, J.-Y.; AND SO KWEON, I., 2018. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19. (cited on pages 17, 28, and 29)
- WU, P.; LIU, J.; SHI, Y.; SUN, Y.; SHAO, F.; WU, Z.; AND YANG, Z., 2020. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *European conference on computer vision*, 322–339. Springer. (cited on page 32)
- WU, Z.; SONG, S.; KHOSLA, A.; YU, F.; ZHANG, L.; TANG, X.; AND XIAO, J., 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 1912–1920. (cited on pages 86, 90, and 91)
- XIA, Y.; ZHANG, Y.; LIU, F.; SHEN, W.; AND YUILLE, A. L., 2020. Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In *European Conference on Computer Vision*, 145–161. Springer. (cited on pages 50, 60, 61, and 72)
- XIAN, Y.; LAMPERT, C. H.; SCHIELE, B.; AND AKATA, Z., 2018. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41, 9 (2018), 2251–2265. (cited on pages 1 and 10)
- XU, B.; LU, C.; GUO, Y.; AND WANG, J., 2020. Discriminative multi-modality speech recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14433–14442. (cited on page 32)
- XU, P.; EHINGER, K. A.; ZHANG, Y.; FINKELSTEIN, A.; KULKARNI, S. R.; AND XIAO, J., 2015. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, (2015). (cited on page 57)
- XU, Z.; DAI, A. M.; KEMP, J.; AND METZ, L., 2019. Learning an adaptive learning rate schedule. *arXiv preprint arXiv:1909.09712*, (2019). (cited on page 18)
- YAMAGUCHI, K.; MCALLESTER, D.; AND URTASUN, R., 2014. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision*, 756–771. Springer. (cited on page 73)
- YAN, S.; ZHANG, S.; HE, X.; ET AL., 2019. A dual attention network with semantic embedding for few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 9079–9086. (cited on pages 3, 15, 16, 18, and 25)

-
- YAN, X.; ZHENG, C.; LI, Z.; WANG, S.; AND CUI, S., 2020. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5589–5598. (cited on page 92)
- YE, Q.; SHEN, X.; GAO, Y.; WANG, Z.; BI, Q.; LI, P.; AND YANG, G., 2021. Temporal cue guided video highlight detection with low-rank audio-visual fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7950–7959. (cited on page 32)
- YEUNG, S.; RUSSAKOVSKY, O.; MORI, G.; AND FEI-FEI, L., 2016. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2678–2687. (cited on page 18)
- YI, J. AND YOON, S., 2020. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian Conference on Computer Vision*. (cited on pages 51, 60, 62, and 63)
- YOU, Z.; CUI, L.; SHEN, Y.; YANG, K.; LU, X.; ZHENG, Y.; AND LE, X., 2022. A unified model for multi-class anomaly detection. In *Advances in Neural Information Processing Systems*. (cited on page 61)
- YU, F.; SEFF, A.; ZHANG, Y.; SONG, S.; FUNKHOUSER, T.; AND XIAO, J., 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, (2015). (cited on page 57)
- YU, Q. AND AIZAWA, K., 2019. Unsupervised out-of-distribution detection by maximum classifier discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9518–9526. (cited on pages 51 and 55)
- YU, Z.; NGUYEN, T.; GAL, Y.; JU, L.; CHANDRA, S. S.; ZHANG, L.; BONNINGTON, P.; MAR, V.; WANG, Z.; AND GE, Z., 2022. Skin lesion recognition with class-hierarchy regularized hyperbolic embeddings. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 594–603. (cited on page 35)
- YUN, S.; CHOI, J.; YOO, Y.; YUN, K.; AND YOUNG CHOI, J., 2017. Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2711–2720. (cited on page 18)
- YUN, S.; HAN, D.; OH, S. J.; CHUN, S.; CHOE, J.; AND YOO, Y., 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. (cited on page 89)
- ZAGORUYKO, S. AND KOMODAKIS, N., 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*, (2016). (cited on pages xxv, 58, 59, and 60)

- ZE, Y.; HANSEN, N.; CHEN, Y.; JAIN, M.; AND WANG, X., 2023. Visual reinforcement learning with self-supervised 3d representations. *IEEE Robotics and Automation Letters*, 8, 5 (2023), 2890–2897. (cited on page 85)
- ZHANG, C.; CAI, Y.; LIN, G.; AND SHEN, C., 2020a. Deepemd: Few-shot image classification with differentiable earth mover’s distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12203–12213. (cited on page 26)
- ZHANG, H.; LI, A.; GUO, J.; AND GUO, Y., 2020b. Hybrid models for open set recognition. In *European Conference on Computer Vision*, 102–117. Springer. (cited on pages 11 and 86)
- ZHANG, J.; CHEN, L.; OUYANG, B.; LIU, B.; ZHU, J.; CHEN, Y.; MENG, Y.; AND WU, D., 2022. Pointcutmix: Regularization strategy for point cloud classification. *Neurocomputing*, (2022). (cited on page 89)
- ZHANG, Y.; LUO, L.; XIAN, W.; AND HUANG, H., 2021a. Learning better visual data similarities via new grouplet non-euclidean embedding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9918–9927. (cited on page 48)
- ZHANG, Y.; WANG, X.; SHI, C.; JIANG, X.; AND YE, Y. F., 2021b. Hyperbolic graph attention network. *IEEE Transactions on Big Data*, (2021), 1690–1701. (cited on page 35)
- ZHAO, H.; JIANG, L.; JIA, J.; TORR, P. H.; AND KOLTUN, V., 2021a. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16259–16268. (cited on pages 85, 88, 94, 96, 97, and 102)
- ZHAO, N.; CHUA, T.-S.; AND LEE, G. H., 2021b. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8873–8882. (cited on pages 86, 87, 88, and 89)
- ZHAO, S.; WANG, Y.; YANG, Z.; AND CAI, D., 2019. Region mutual information loss for semantic segmentation. *arXiv preprint arXiv:1910.12037*, (2019). (cited on page 78)
- ZHENG, J.; LI, W.; HONG, J.; PETERSSON, L.; AND BARNES, N., 2022. Towards open-set object detection and discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 3961–3970. (cited on pages xvii, 11, 12, and 51)
- ZHENG, Q.; HONG, J.; AND FARAZI, M., 2023. A generative approach to audio-visual generalized zero-shot learning: Combining contrastive and discriminative techniques. In *2023 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE. (cited on pages 33 and 35)

-
- ZHENG, S.; LU, J.; ZHAO, H.; ZHU, X.; LUO, Z.; WANG, Y.; FU, Y.; FENG, J.; XIANG, T.; TORR, P. H.; ET AL., 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6881–6890. (cited on page 69)
- ZHOU, B.; ZHAO, H.; PUIG, X.; FIDLER, S.; BARRIUSO, A.; AND TORRALBA, A., 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 633–641. (cited on page 69)
- ZHOU, H.; FENG, Y.; FANG, M.; WEI, M.; QIN, J.; AND LU, T., 2021a. Adaptive graph convolution for point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4965–4974. (cited on pages 98 and 99)
- ZHOU, J.; ZHENG, L.; ZHONG, Y.; HAO, S.; AND WANG, M., 2021b. Positive sample propagation along the audio-visual event line. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8436–8444. (cited on page 31)
- ZHOU, K.; LI, J.; XIAO, Y.; YANG, J.; CHENG, J.; LIU, W.; LUO, W.; LIU, J.; AND GAO, S., 2021c. Memorizing structure-texture correspondence for image anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, (2021). (cited on page 51)
- ZHOU, Y.; SUN, X.; ZHA, Z.-J.; AND ZENG, W., 2019. Context-reinforced semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4046–4055. (cited on page 18)
- ZHU, F.; ZHU, L.; AND YANG, Y., 2019. Sim-real joint reinforcement transfer for 3d indoor navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11388–11397. (cited on page 18)
- ZIKO, I. M.; DOLZ, J.; GRANGER, E.; AND AYED, I. B., 2020. Laplacian regularized few-shot learning. *arXiv preprint arXiv:2006.15486*, (2020). (cited on pages 24, 25, and 26)
- ZORAN, D.; CHRZANOWSKI, M.; HUANG, P.-S.; GOWAL, S.; MOTT, A.; AND KOHLI, P., 2020. Towards robust image classification using sequential attention models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9483–9492. (cited on pages 17 and 18)