DMQA Open Seminar

OOD Detection for Image Classification Part4
: Prototype-based Approach

2025. 12. 12

고려대학교 산업경영공학과

Data Mining & Quality Analytics Lab.

임 새 린

발표자 소개

❖ **임새린 (Saerin Lim)**

- 고려대학교 산업경영공학과 Data Mining & Quality Analytics Lab.
- Ph.D. Student (2021.03 ~ Present)
- 지도 교수 : 김성범 교수님

❖ **Research Interest**

- Self-supervised learning & Semi-supervised learning

❖ **Contact**

- E-mail : momo_om@korea.ac.kr

목차
Outline

Overview

OOD Detection Approach

❖ 2025년 ACM Computing Surveys 저널에 게재된 OOD detection survey paper

Out-of-Distribution Detection: A Task-Oriented Survey of Recent Advances

SHUO LU, NLPR & MAIS, Institute of Automation, Chinese Academy of Sciences; UCAS, China
YINGSHENG WANG*, Anhui University, China
LIJUN SHENG, University of Science and Technology of China, China
LINGXIAO HE, Meituan, China
AIHUA ZHENG, Anhui University, China
JIAN LIANG†, NLPR & MAIS, Institute of Automation, Chinese Academy of Sciences; UCAS, China

Out-of-distribution (OOD) detection aims to detect test samples outside the training category space, which is an essential component in building reliable machine learning systems. Existing reviews on OOD detection primarily focus on method taxonomy, surveying the field by categorizing various approaches. However, many recent works concentrate on non-traditional OOD detection scenarios, such as test-time adaptation, multi-modal data sources and other novel contexts. In this survey, we uniquely review recent advances in OOD detection from the task-oriented perspective for the first time. According to the user's access to the model, that is, whether the OOD detection method is allowed to modify or retrain the model, we classify the methods as training-driven or training-agnostic. Besides, considering the rapid development of pre-trained models, large pre-trained model-based OOD detection is also regarded as an important category and discussed separately. Furthermore, we provide a discussion of the evaluation scenarios, a variety of applications, and several future research directions. We believe this survey with new taxonomy will benefit the proposal of new methods and the expansion of more practical scenarios. A curated list of related papers is provided in the Github repository: https://github.com/shuolucs/Awesome-Out-Of-Distribution-Detection

Overview

OOD Detection Approach

- OOD Detection
  - Problem: Training-driven OOD detection(§ 3)
    - Approaches with only ID Data (§ 3.1)
      - Reconstruction-based: MoodCat [22], RAE [23], MOOD [24], MOODv2 [25], PRE [26], LMD [27] DiffGuard [28], DenoDiff [29]
      - Probability-based: HVCM [30], DDR [31], LID [32]
      - Logits-based: LogitNorm [33], UE-NL [34], DML [35]
      - OOD Synthesis: ConfiCali [36], CODEs [37], CMG [38], VOS [39], NPOS [40], SHIFT [41], ATOL [42], SSOD [43], SEM [44], Forte [45], HamOS [46], POP [47]
      - Prototype-based: Step [48], SIREN [49], CIDER [50], PALM [51], ReweightOOD [52], AROS [53], PFS [54]
      - A special case: Long-tail ID data: OLTR [55], II-Mixup [56], AREO [57], IDCP [58], Open-Sampling [59], COOD [60]
    - Approaches with both ID and OOD Data (§ 3.2)
      - Boundary Regularization: OE [61], ELOC [62], Why-ReLU [63], SSL-GOOD [64], EnergyOE [65], MixOE [66]
      - Outlier Mining: BD-Resamp [67], POEM [68], DAOL [69], DOE [70], MixOE [66], DivOE [71]
      - Imbalanced ID: PASCAL [72], COCL [73], BERL [74], EAT [75]
  - Problem: Training-agnostic OOD detection (§ 4)
    - Post-hoc Approaches (§ 4.1)
      - Output-based: MSP [1], MaxLogits [76], Energy [65], GEN [77], ZODE [78], LogicOOD [79]
      - Distance-based: Mahalanobis [80], NNGuide [81], KNN [82], SSD [83],Mahalanobis++ [84]
      - Gradient-based: Grad [85], GradNorm [86], GradOrth [87], GAIA [88], OPNP [89], PRO [90]
      - Feature-based: ODIN [91], ReAct [92], VRA [93], SHE [94], ViM [95], Neco [96], ASH [97], NAC [98] Optimal-FS [99], BLOOD [100], SCALE [101], DDCS [102], LINe [103], KANs [104], CADRef [105], ITP [106], NCI [107]
      - Density-based: GEM [108], ConjNorm [109]
    - Test-time Adaptive Approaches (§ 4.2)
      - Model-optimization-based: WOODS [110], AUTO [11], SODA [12], ATTA [13], SAL [14]
      - Model-optimization-free: ETLT [15], AdaOOD [111], GOODAT [16], RTL [112], OODD [113]

Overview

OOD Detection Approach

- OOD Detection
  - Problem: Training-driven OOD detection(§ 3)
    - Approaches with only ID Data (§ 3.1)
      - Reconstruction-based: MoodCat [22], RAE [23], MOOD [24], MOODv2 [25], PRE [26], LMD [27] DiffGuard [28], DenoDiff [29]
      - Probability-based: HVCM [30], DDR [31], LID [32]
      - Logits-based: LogitNorm [33], UE-NL [34], DML [35]
      - OOD Synthesis: ConfiCali [36], CODEs [37], CMG [38], VOS [39], NPOS [40], SHIFT [41], ATOL [42], SSOD [43], SEM [44], Forte [45], HamOS [46], POP [47]
      - Prototype-based: Step [48], SIREN [49], CIDER [50], PALM [51], ReweightOOD [52], AROS [53], PFS [54]
      - A special case: Long-tail ID data: OLTR [55], II-Mixup [56], AREO [57], IDCP [58], Open-Sampling [59], COOD [60]
    - Approaches with both ID and OOD Data (§ 3.2)
      - Boundary Regularization: OE [61], ELOC [62], Why-ReLU [63], SSL-GOOD [64], EnergyOE [65], MixOE [66]
      - Outlier Mining: BD-Resamp [67], POEM [68], DAOL [69], DOE [70], MixOE [66], DivOE [71]
      - Imbalanced ID: PASCAL [72], COCL [73], BERL [74], EAT [75]
  - Problem: Training-agnostic OOD detection (§ 4)
    - Post-hoc Approaches (§ 4.1)
      - Output-based: MSP [1], MaxLogits [76], Energy [65], GEN [77], ZODE [78], LogicOOD [79]
      - Distance-based: Mahalanobis [80], NNGuide [81], KNN [82], SSD [83],Mahalanobis++ [84]
      - Gradient-based: Grad [85], GradNorm [86], GradOrth [87], GAIA [88], OPNP [89], PRO [90]
      - Feature-based: ODIN [91], ReAct [92], VRA [93], SHE [94], ViM [95], Neco [96], ASH [97], NAC [98] Optimal-FS [99], BLOOD [100], SCALE [101], DDCS [102], LINe [103], KANs [104], CADRef [105], ITP [106], NCI [107]
      - Density-based: GEM [108], ConjNorm [109]
    - Test-time Adaptive Approaches (§ 4.2)
      - Model-optimization-based: WOODS [110], AUTO [11], SODA [12], ATTA [13], SAL [14]
      - Model-optimization-free: ETLT [15], AdaOOD [111], GOODAT [16], RTL [112], OODD [113]

종료

DMQA Open Seminar

Score-Based OOD Detection for Image Classification: Part1

2024. 01. 26

Score-Based OOD Detection for Image Cl

발표자: 임새린

2024년 1월 26일

오전 11시 ~

온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

종료

DMQA Open Seminar

Out-Of-Distribution Detection for Image Classification: Part2

2024. 09. 27

OOD Detection for Image Classification P

발표자: 임새린

2024년 9월 27일

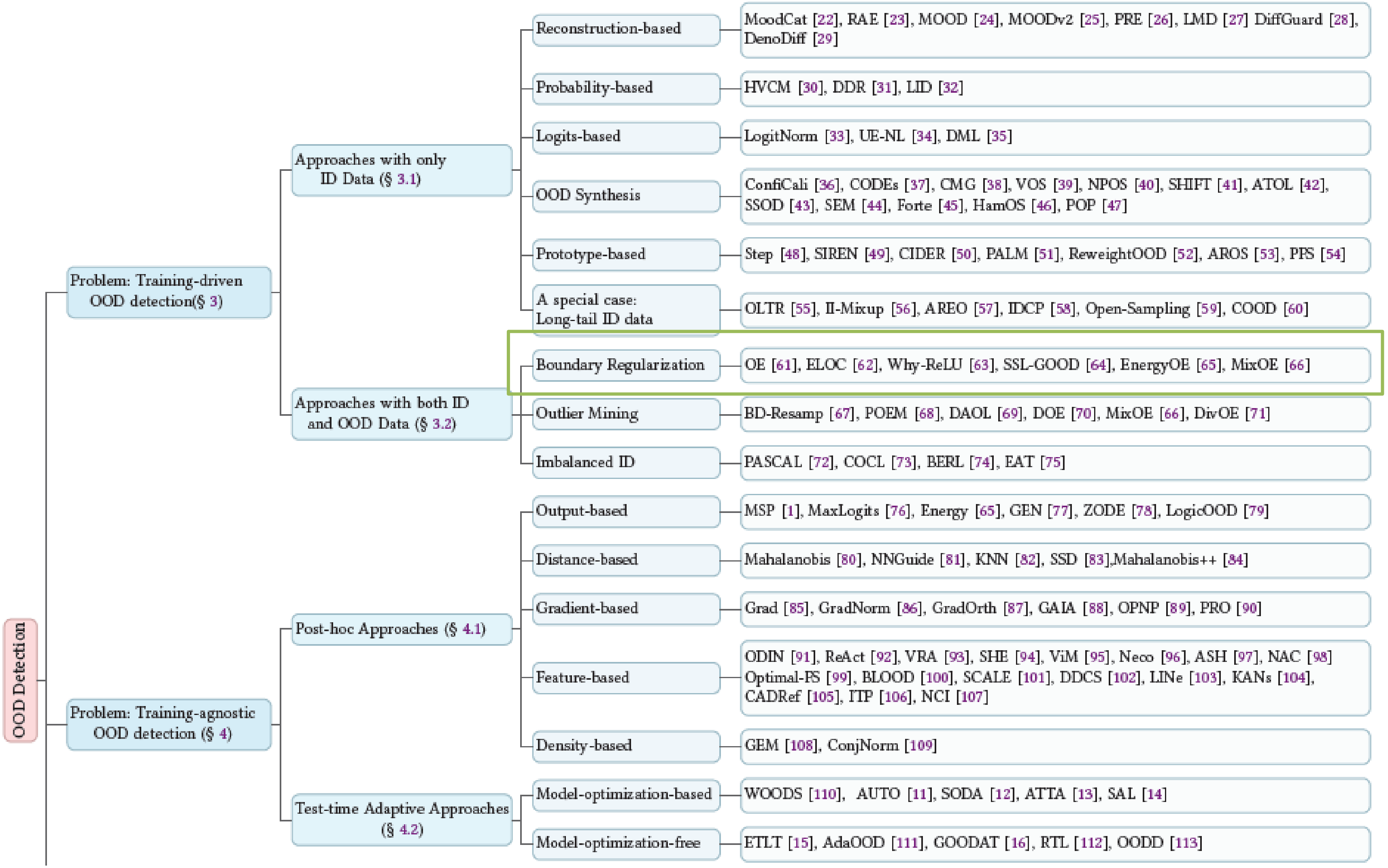
오전 10시 ~

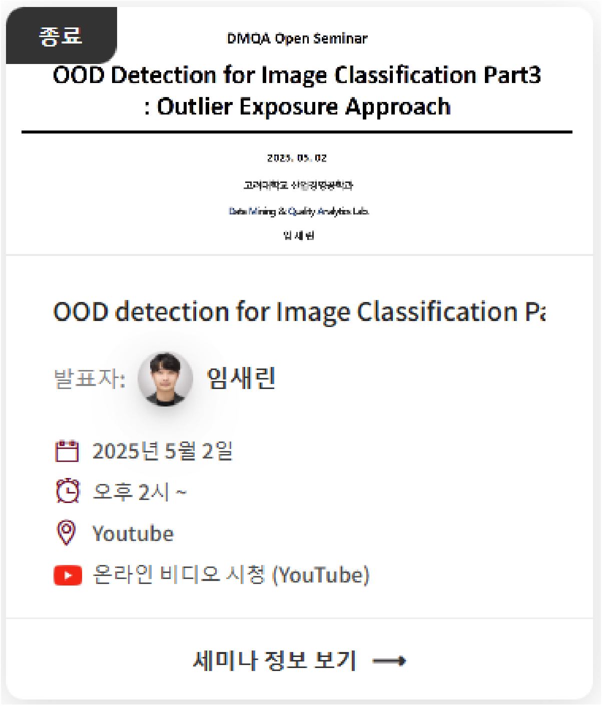
온라인 비디오 시청 (YouTube)

세미나 정보 보기 ⟶

Overview

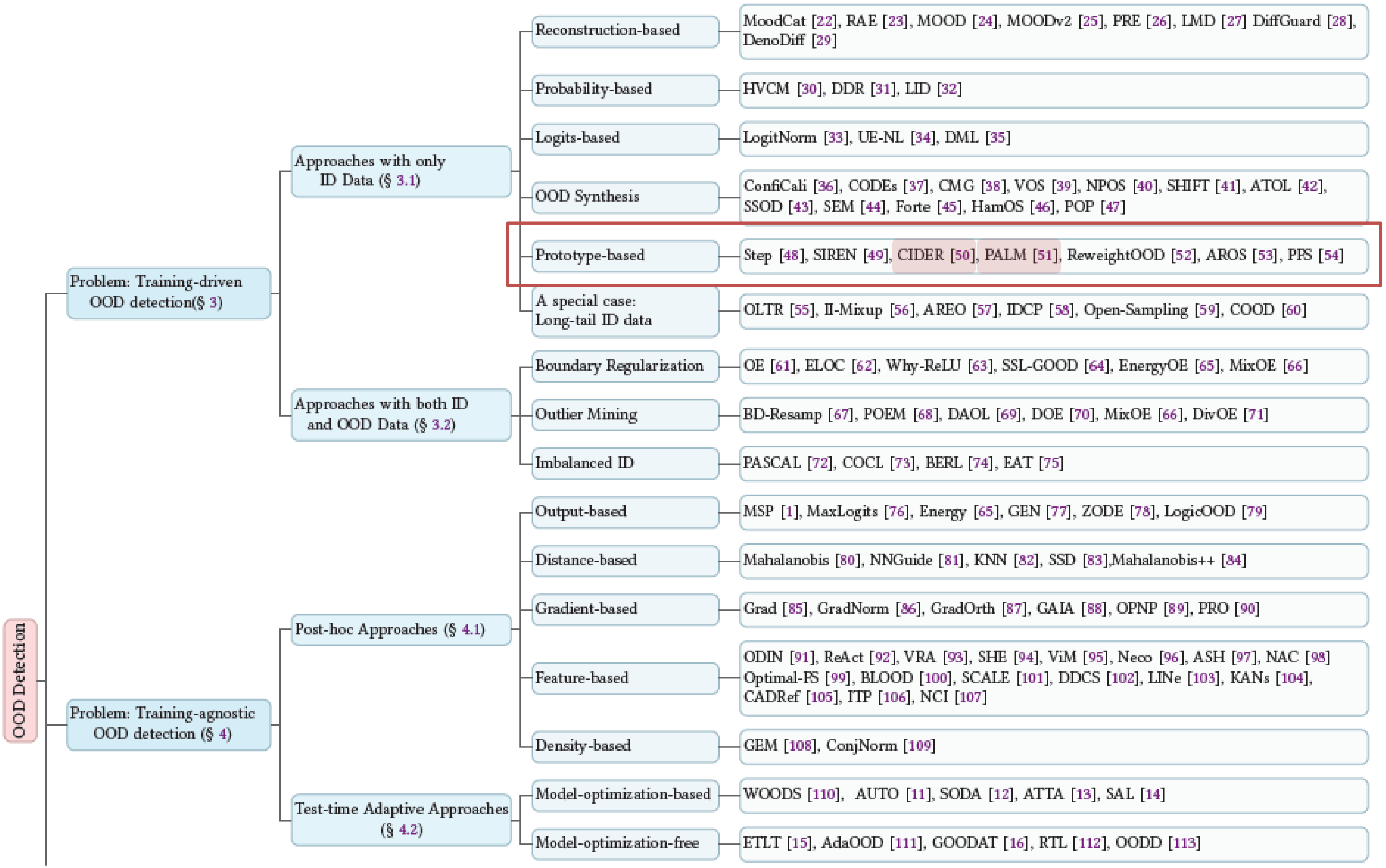
OOD Detection Approach





Overview

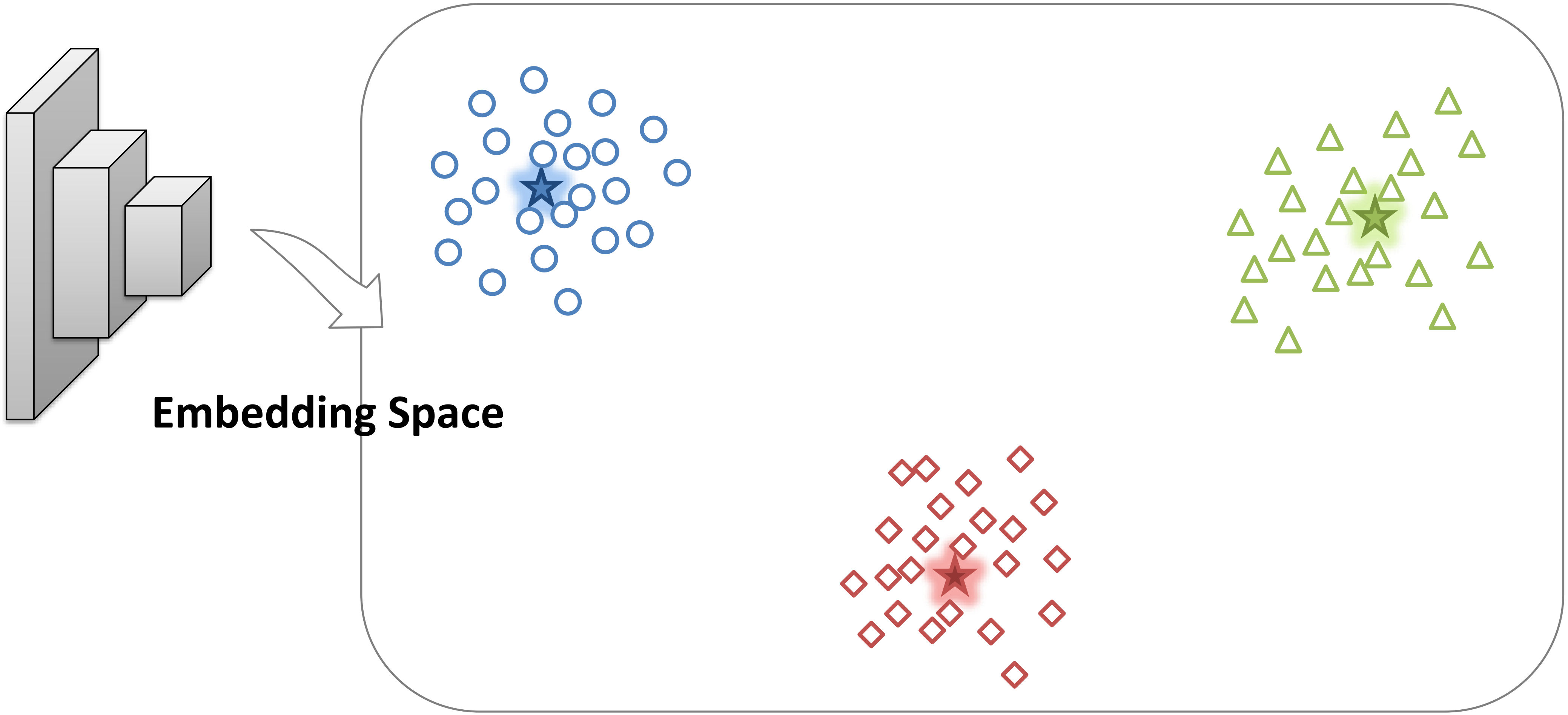
OOD Detection Approach



Overview

Prototype이란

❖ Prototype이란 어떤 클래스, 클러스터와 같은 집단을 대표하는 "전형적인 예시를 나타내는 벡터"를 의미

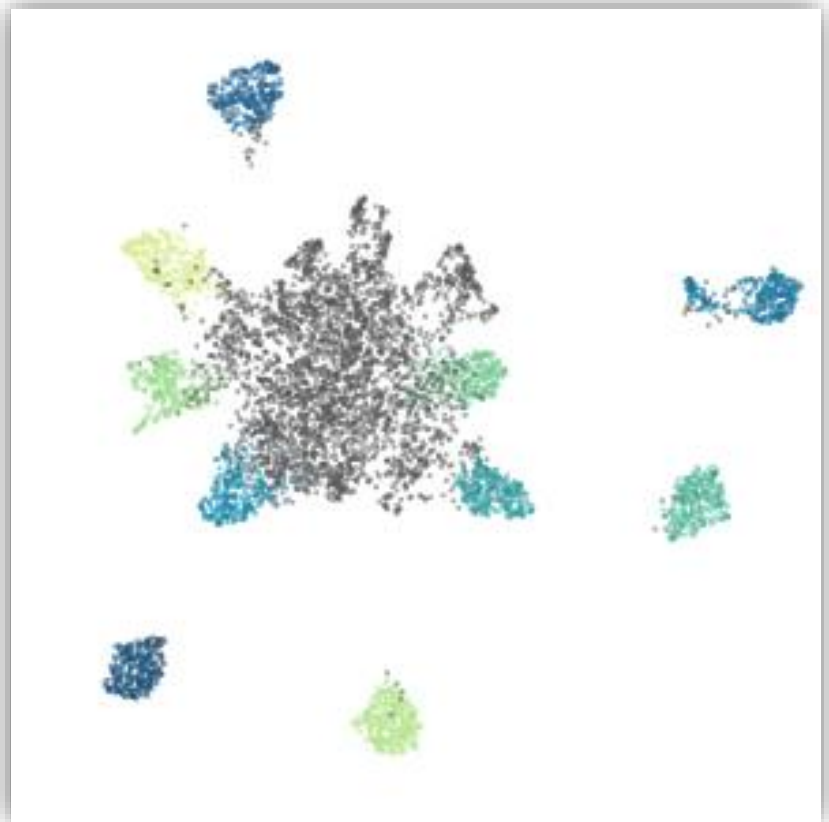


How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection

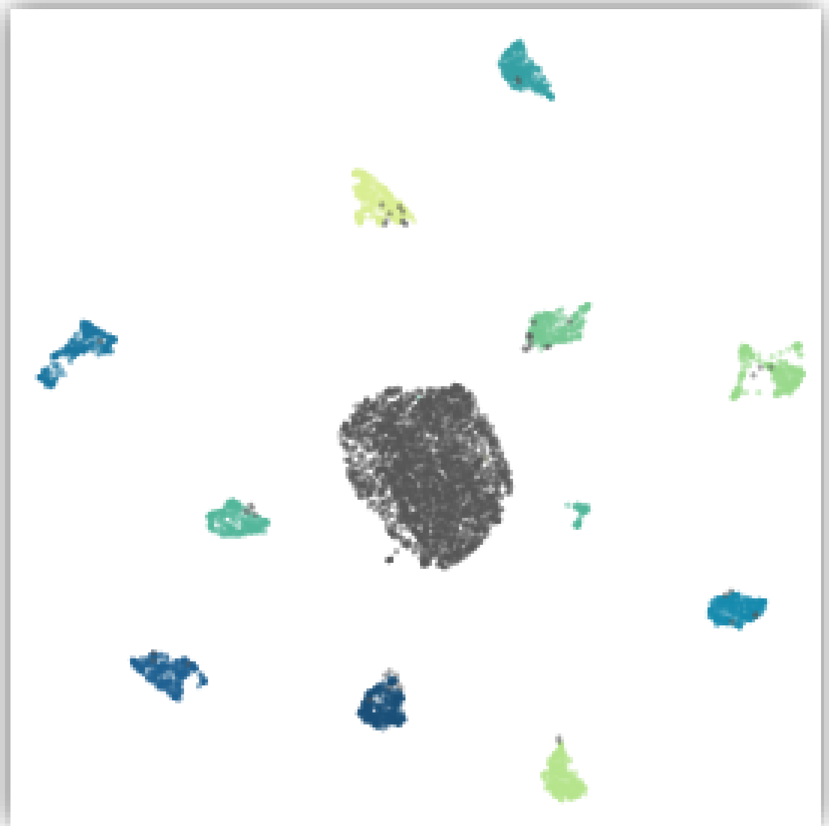
(2023, ICLR)

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)



Embedding Space
trained by Cross-Entropy



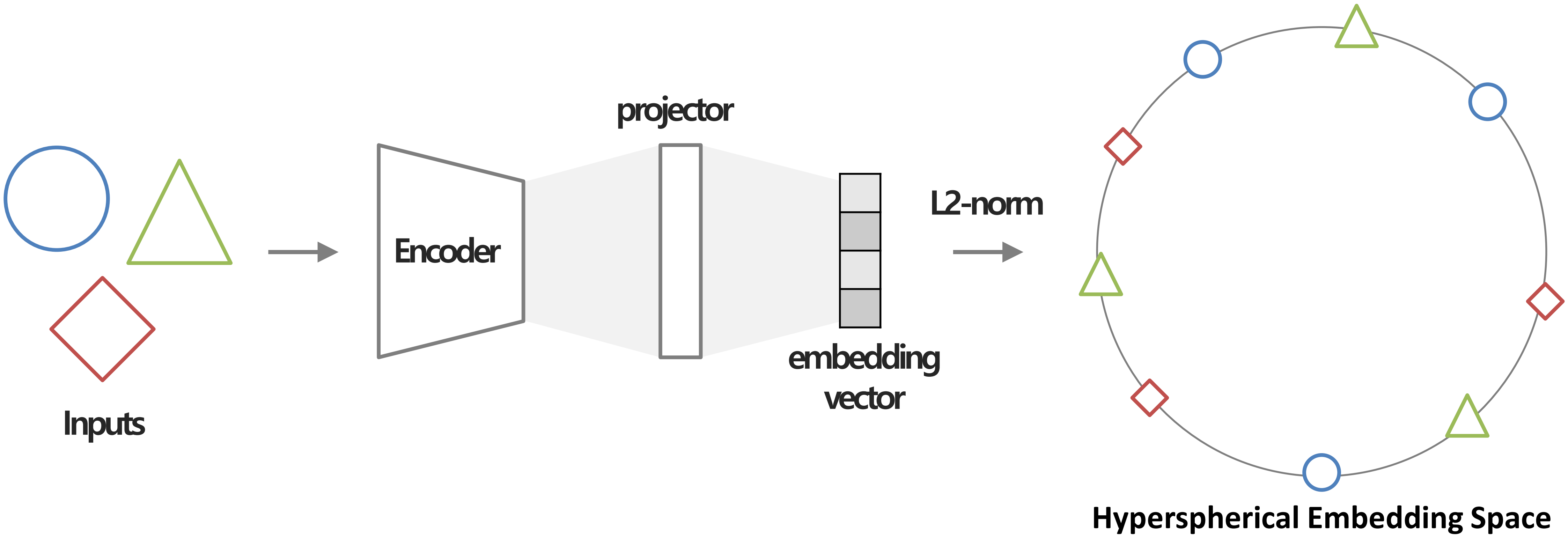
Embedding Space
trained by SupCon

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ Contrastive learning은 metric learning 중 하나로 데이터 간 상대적인 유사도를 학습해서 데이터들을 구분하기 쉽게 해주는 hyperspherical embedding space를 학습하는 방법

❖ **Anchor**를 기준으로 **positive samples**는 가깝도록, **negative samples**는 멀도록 학습



Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

- Contrastive learning은 metric learning 중 하나로 데이터 간 상대적인 유사도를 학습해서 데이터들을 구분하기 쉽게 해주는 hyperspherical embedding space를 학습하는 방법
- **Anchor**를 기준으로 **positive samples**는 가깝도록, **negative samples**는 멀도록 학습

$$L_{ctr}(x_i) = -\frac{1}{|P(i)|}\sum_{p\in P(i)} log \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a\in A(i)} \exp(z_i \cdot z_a/\tau)}$$

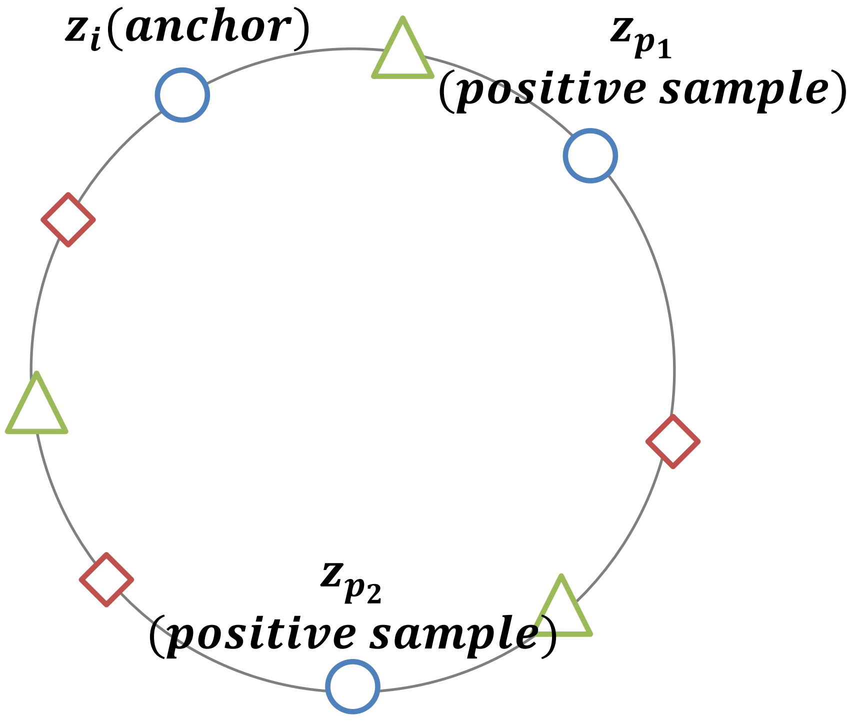
$x_i = anchor$(기준이 되는 샘플)

$z_i = embedidng\ vector\ of\ x_i$

$P(i) = x_i$의 $positive\ sample$ 집합

$A(i) = x_i$를 제외한 모든 $sample$ 집합

$\tau = Temperature$



Hyperspherical Embedding Space

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

- ❖ Contrastive learning은 metric learning 중 하나로 데이터 간 상대적인 유사도를 학습해서 데이터들을 구분하기 쉽게 해주는 hyperspherical embedding space를 학습하는 방법
- ❖ **Anchor**를 기준으로 **positive samples**는 가깝도록, **negative samples**는 멀도록 학습

$$L_{ctr}(x_i) = -\frac{1}{|P(i)|}\sum_{p\in P(i)} log \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a\in A(i)} \exp(z_i \cdot z_a/\tau)}$$

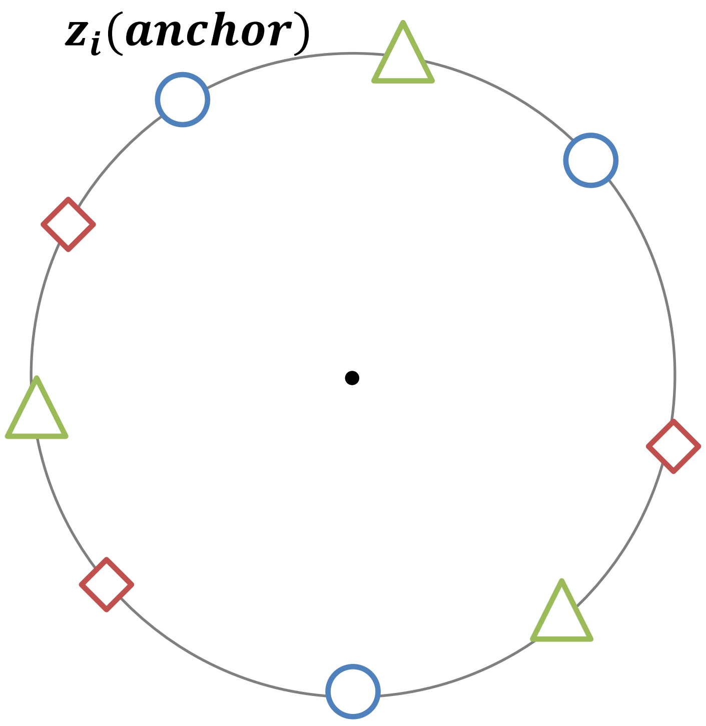
$x_i = anchor$(기준이 되는 샘플)

$z_i = embedidng\ vector\ of\ x_i$

$P(i) = x_i$의 $positive\ sample$ 집합

$A(i) = x_i$를 제외한 모든 $sample$ 집합

$\tau = Temperature$



Hyperspherical Embedding Space

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

- ❖ Contrastive learning은 metric learning 중 하나로 데이터 간 상대적인 유사도를 학습해서 데이터들을 구분하기 쉽게 해주는 hyperspherical embedding space를 학습하는 방법
- ❖ **Anchor**를 기준으로 **positive samples**는 가깝도록, **negative samples**는 멀도록 학습

$$L_{ctr}(x_i) = -\frac{1}{|P(i)|} \sum_{p \in P(i)} log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a / \tau)}$$

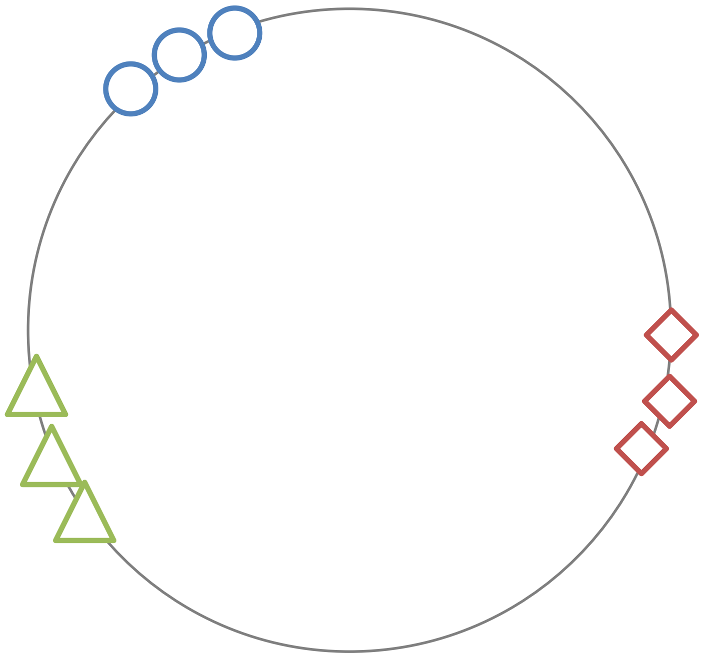
$x_i = anchor$(기준이 되는 샘플)

$z_i = embedidng\ vector\ of\ x_i$

$P(i)$ $= x_i$의 $positive\ sample$ 집합

$A(i) = x_i$를 제외한 모든 $sample$ 집합

$\tau = Temperature$

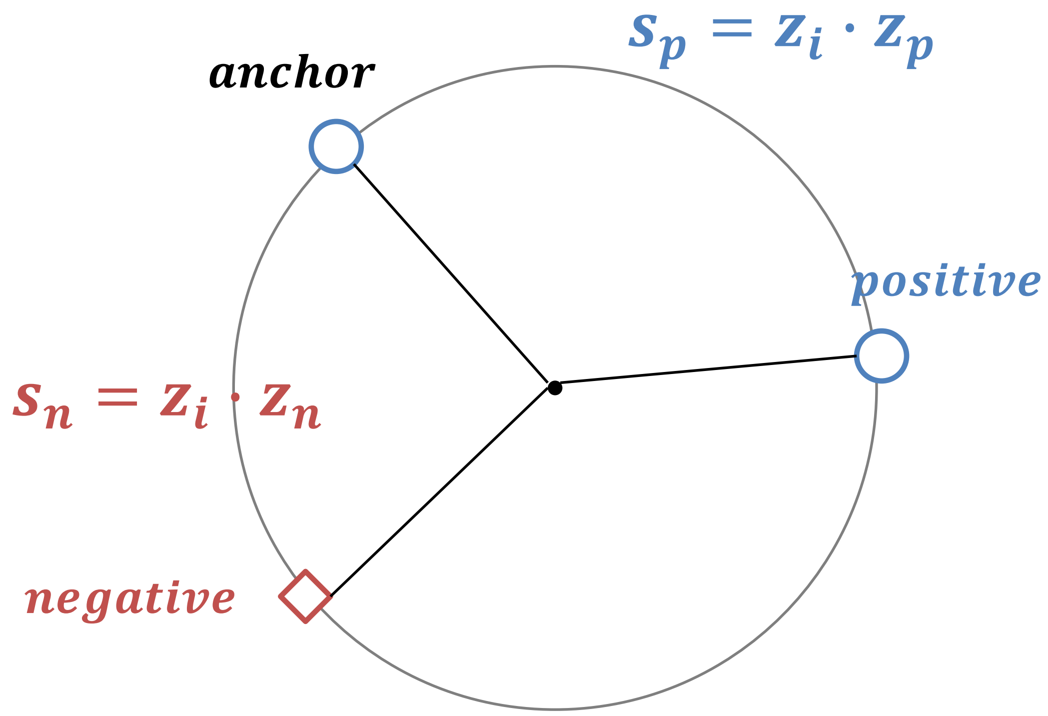


Hyperspherical Embedding Space

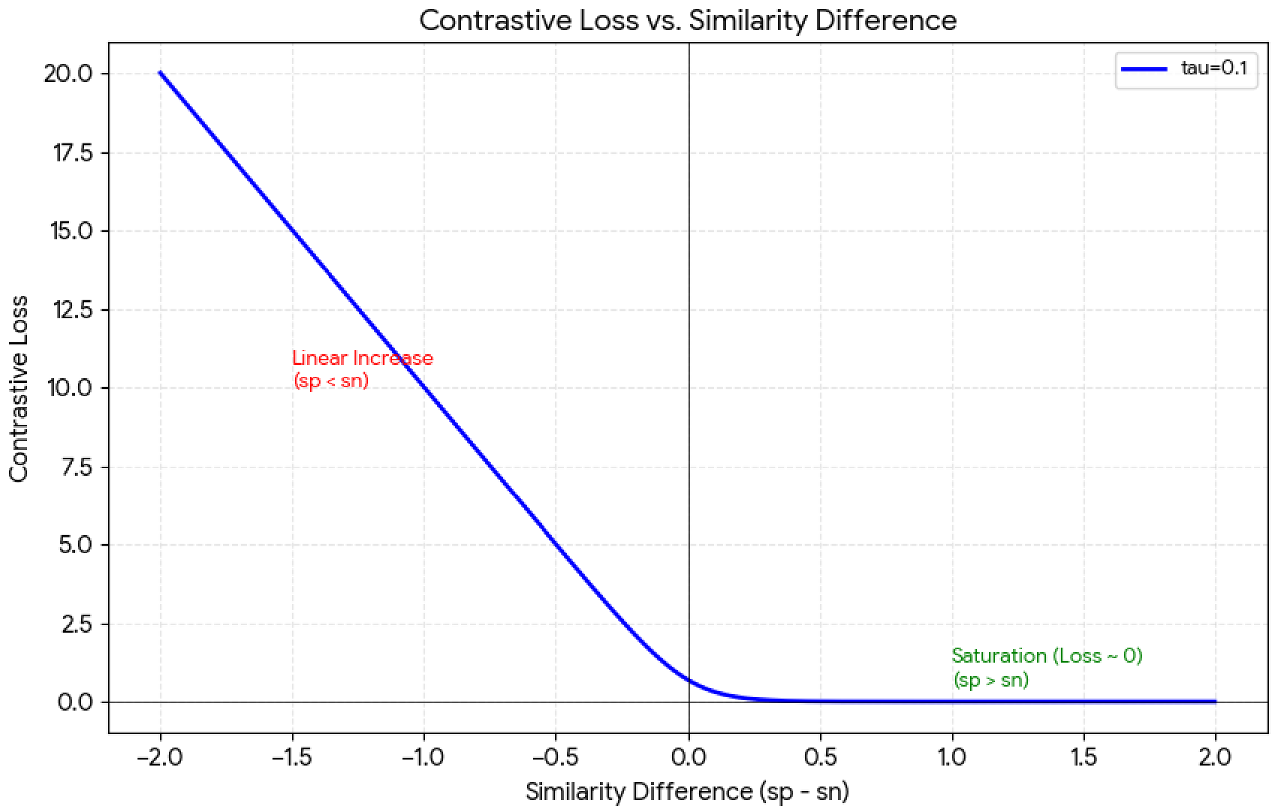
Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ Contrastive learning은 metric learning 중 하나로 **데이터 간 상대적인 유사도를 학습**해서 데이터들을 구분하기 쉽게 해주는 hyperspherical embedding space를 학습하는 방법



Hyperspherical Embedding Space

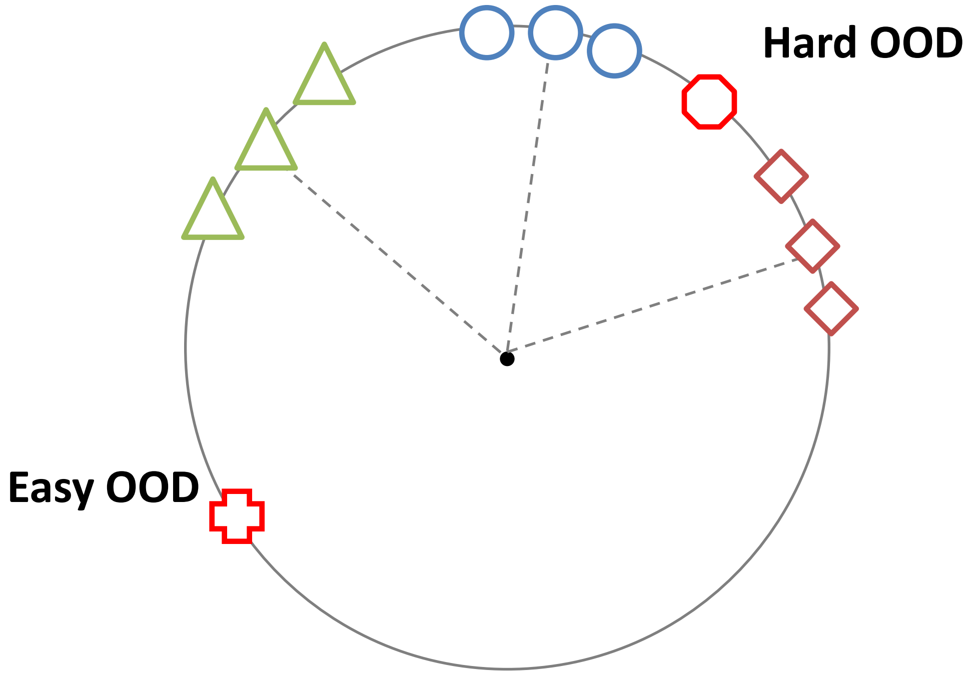


Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ Contrastive learning은 metric learning 중 하나로 **데이터 간 상대적인 유사도를 학습**해서 데이터들을 구분하기 쉽게 해주는 hyperspherical embedding space를 학습하는 방법

❖ SupCon loss는 같은 클래스 샘플끼리 잘 뭉치게 할 수는 있지만, 클래스 군집 간 거리에 대한 규제가 없음



High intra-class compact ness

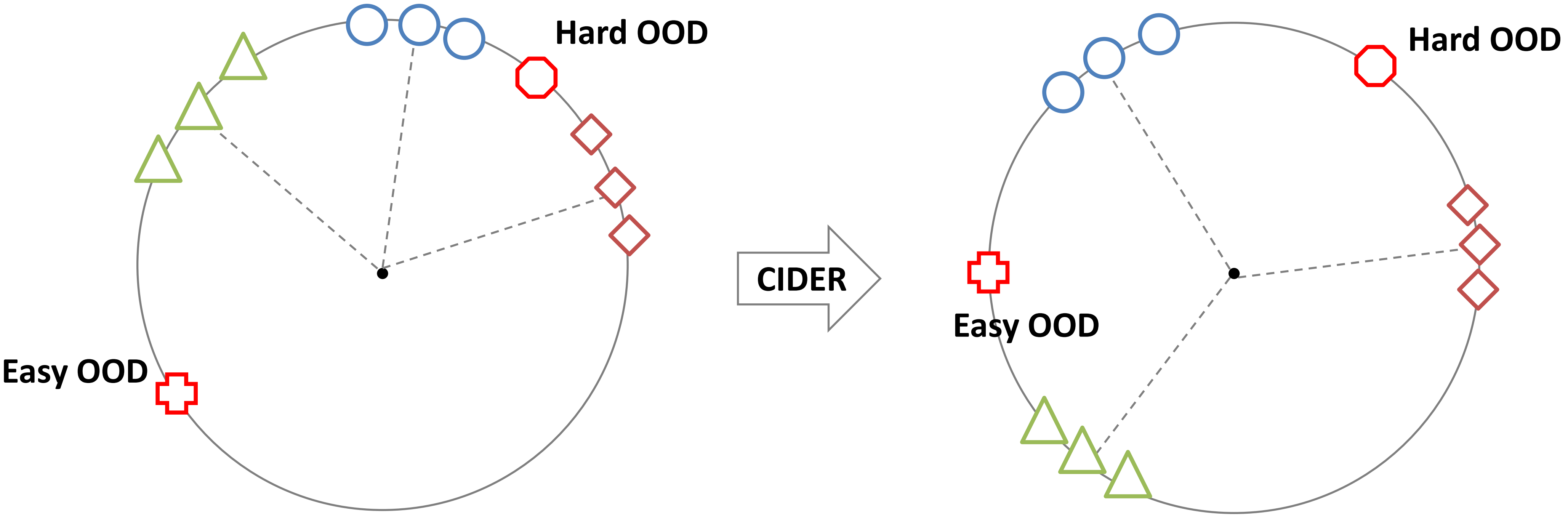
But, low inter-class distance

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ Prototype을 활용해서 클래스 내부 compactness와 함께 클래스 군집 간 거리를 높일 수 있는 **compactness and dispersion regularized learning (CIDER)** 프레임워크를 제안

❖ 이를 통해서, easy OOD 뿐만 아니라 hard OOD도 잘 구분할 수 있는 embedding space를 구축



High intra-class compactness
But, low inter-class distance

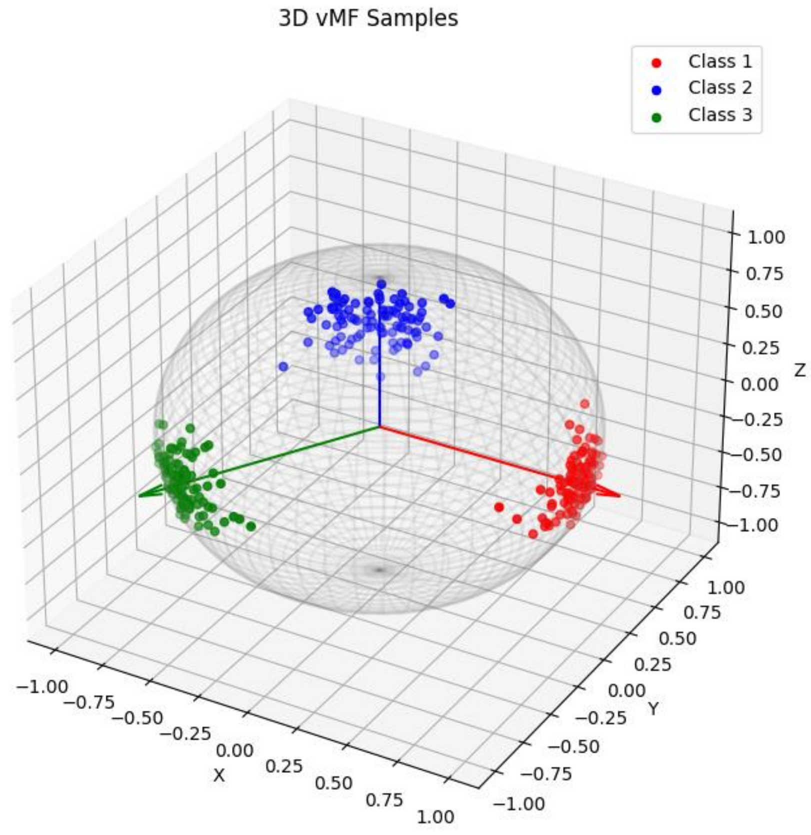
High intra-class compactness
And, high inter-class distance

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ Hyperspherical embedding space 상에서 클래스별 샘플들이 중심 벡터를 기준으로 형성되어 있다고 가정

❖ 이를 수학적으로 모델링하기 위해서 von-Mises-Fisher (vMF) 분포를 가정하고, 중심 벡터를 추정해 prototype으로 활용



$$pdf\ of\ vMF = p(z; \mu_c, \kappa) = Z(\kappa)\exp(\kappa\mu_c^T z)$$
$$= Z(\kappa)\exp(\mu_c^T z/\tau)$$

$z = embedding\ vector$

$\mu_c = mean\ vector\ of\ class\ c$

$\kappa = 1/\tau = concentration\ parameter$

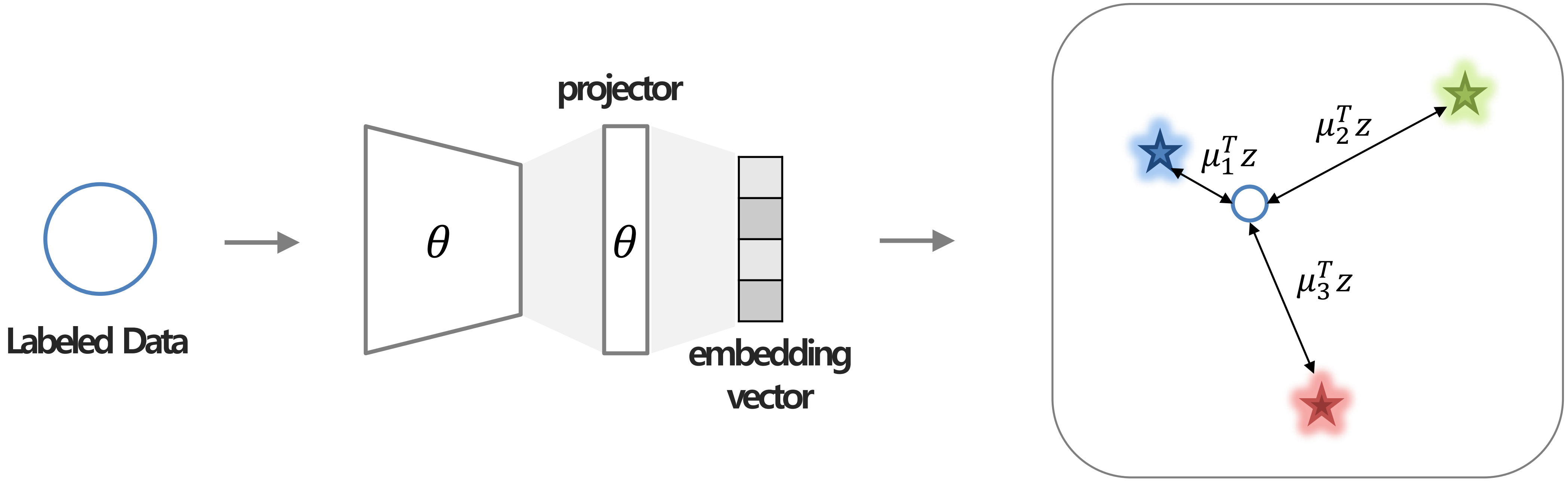
$Z(\kappa) = normalization\ constant$

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ 각 class prototype과 embedding vector 간 유사도를 기반으로 예측 확률 분포를 생성

$$\mathbb{P}\left(y = c | z; \{\kappa, \mu_j\}_{j=1}^{C}\right) = \frac{Z(\kappa)\exp(\mu_c^T z/\tau)}{\sum_{j=1}^{C} Z(\kappa)\exp(\mu_j^T z/\tau)} = \frac{\exp(\mu_c^T z/\tau)}{\sum_{j=1}^{C}\exp(\mu_j^T z/\tau)}$$

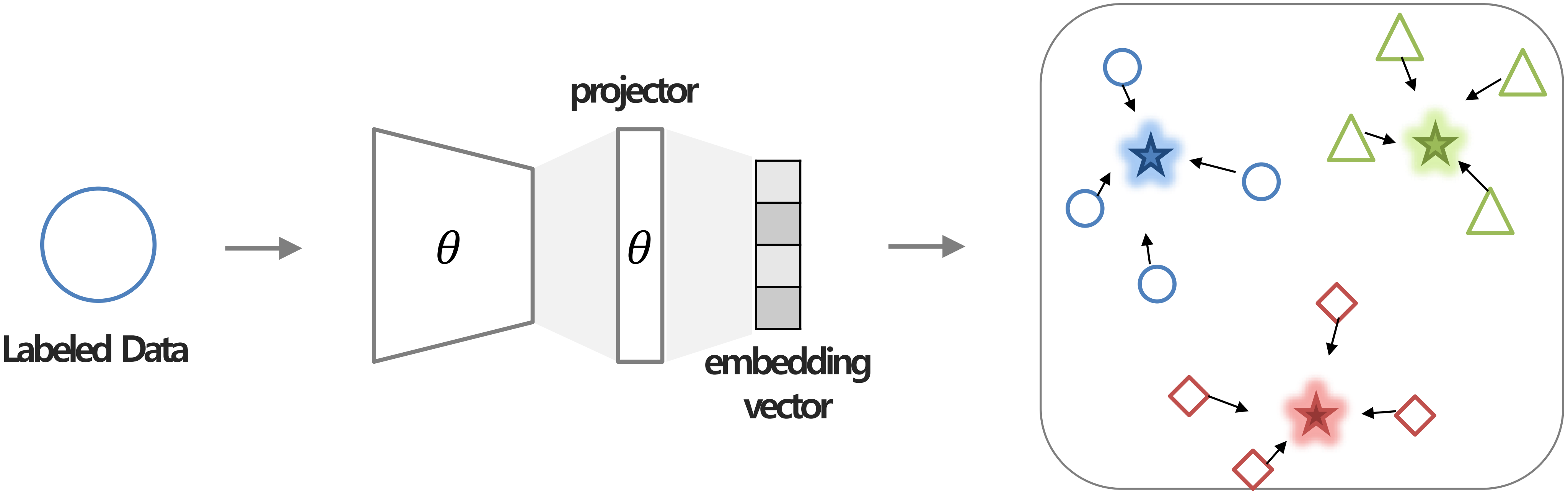


Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ Prototype이 주어진 상태에서 maximum likelihood estimation (MLE)을 활용하여 최적의 모델 파라미터를 학습할 수 있음

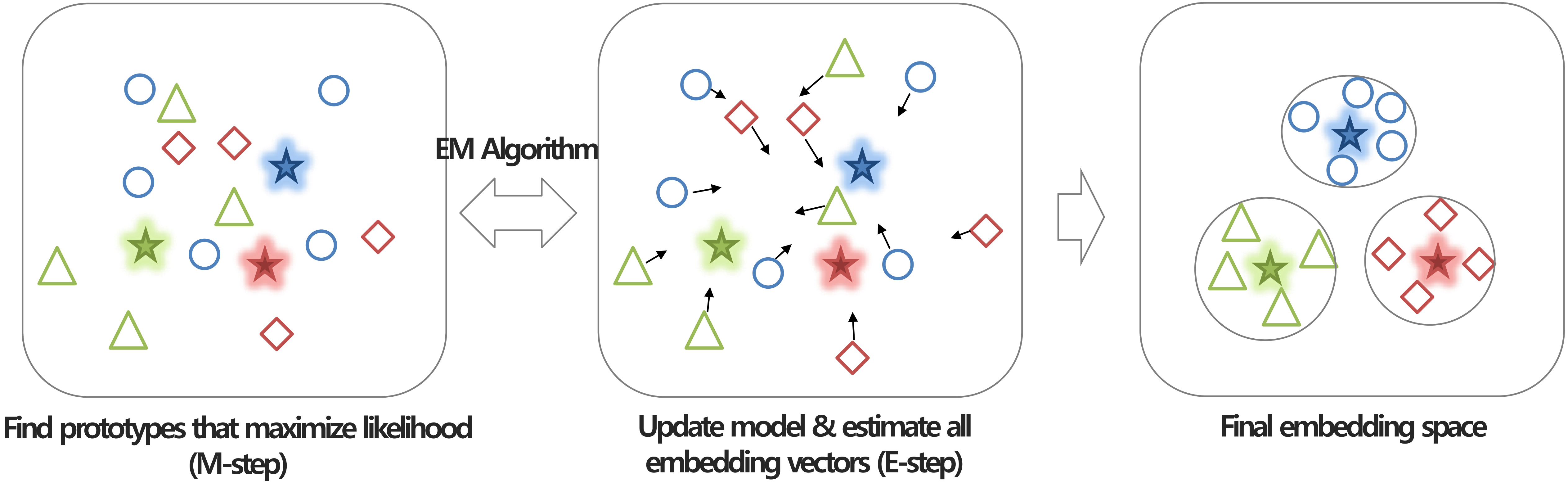
$$argmax_\theta \prod_{i=1}^{N} p(y_i|z_i;\{\kappa,\mu_j\}_{j=1}^{C}) \Leftrightarrow L_{comp} = minNLL = CE(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mu_{c(i)}^T z_i/\tau)}{\sum_{j=1}^{C}\exp(\mu_j^T z_i/\tau)}$$



Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

- ❖ Prototype 추정과 모델 업데이트를 점진적으로 하기 위해서는 Expectation-Maximization (EM) 알고리즘을 활용해야 함
- ❖ 하지만, 매 iteration마다 모든 데이터의 평균을 구하는 것은 연산량이 너무 많아 현실적으로 적용이 불가능함

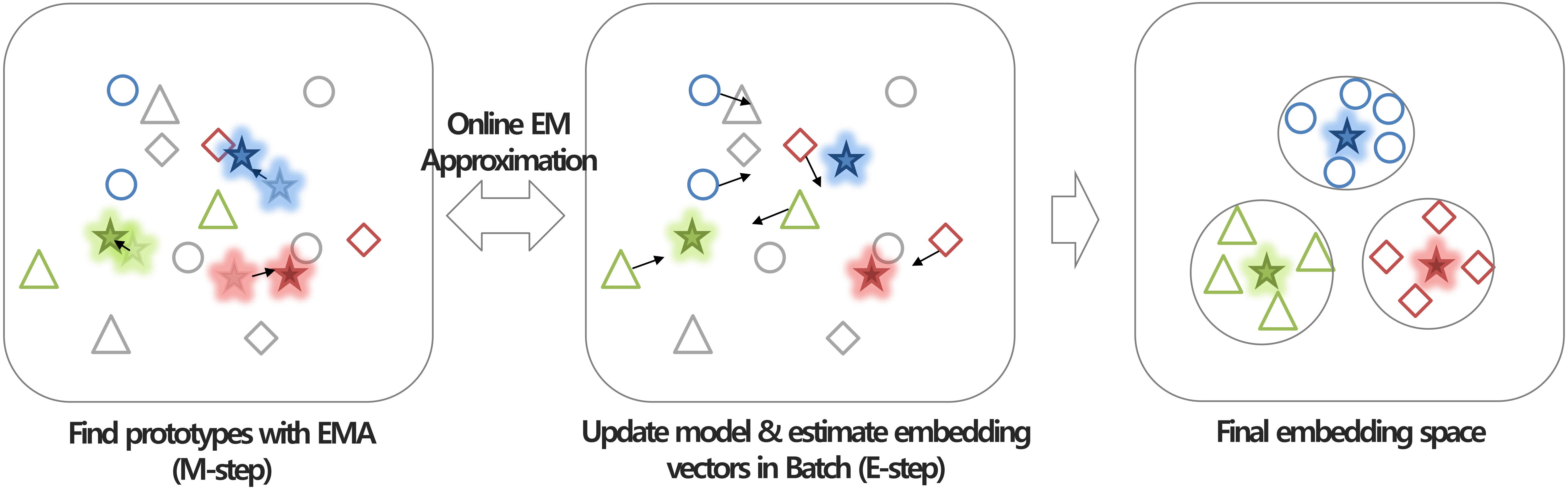


Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ EM algorithm을 온라인으로 근사하기 위해서 Exponential Moving Average (EMA)를 활용

$$\mu_{new} = \alpha\mu_{old} + (1-\alpha)\mu_{batch}$$

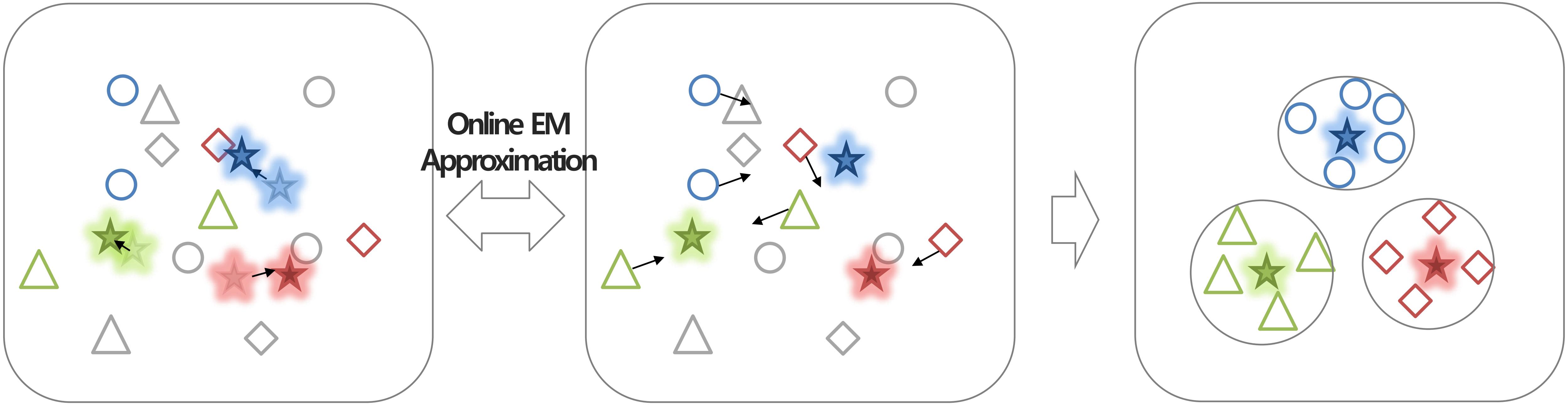


Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ Compactness loss만을 활용한다면 SupCon loss와 마찬가지로 compact하지만 클래스 군집 간 거리를 고려하지 못함

$$only\ L_{comp}$$



Find prototypes with EMA (M-step)

Update model & estimate embedding vectors in Batch (E-step)

High intra-class compactness
But, low inter-class distance

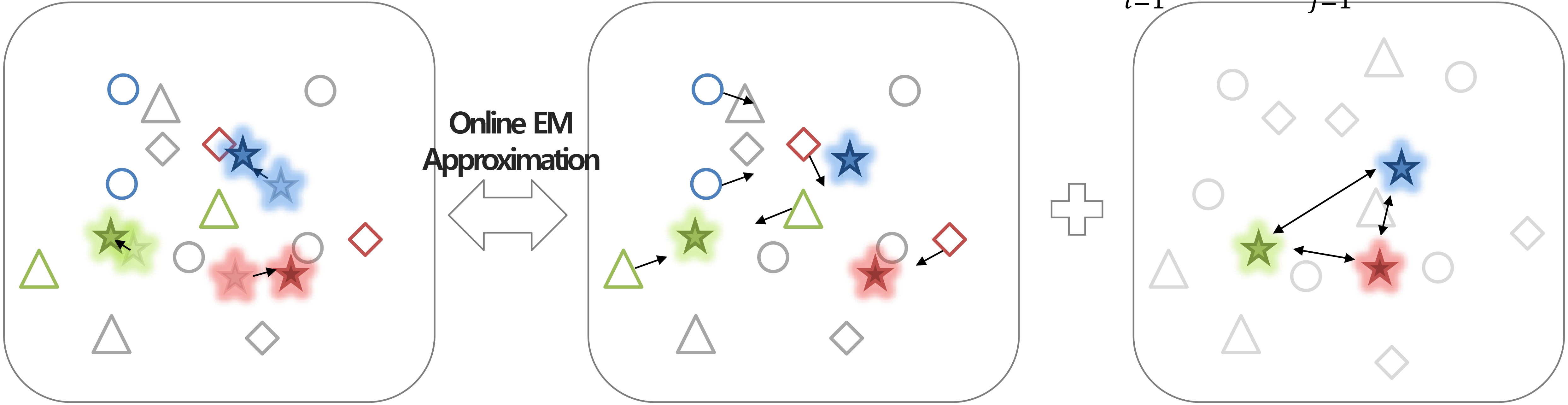
Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ 추가적으로 prototype 간의 각도 (거리)를 커지게 하는 dispersion loss를 도입하여 이 문제를 해결함

L_{comp}

$$L_{dis} = \frac{1}{C}\sum_{i=1}^{C}\log\frac{1}{C-1}\sum_{j=1}^{C}1\{j \neq i\}e^{\mu_i^T \mu_j/\tau}$$



Find prototypes with EMA (M-step)

Update model & estimate embedding vectors in Batch (E-step)

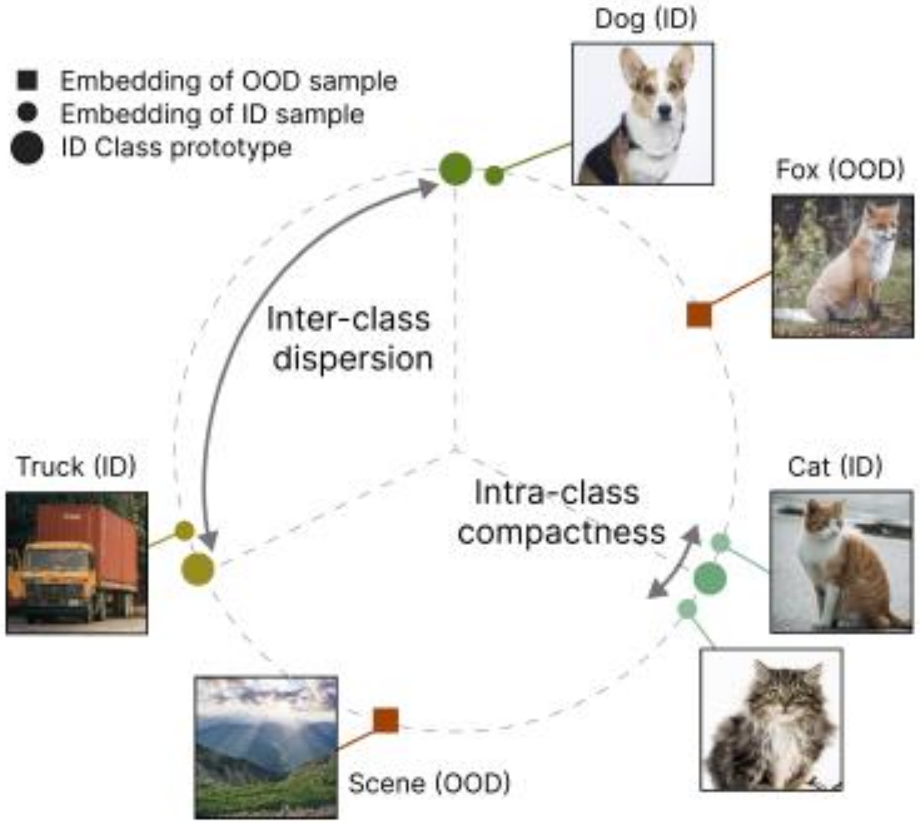
Repel different prototypes

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ 추가적으로 prototype 간의 각도 (거리)를 커지게 하는 dispersion loss를 도입하여 이 문제를 해결함

$$L_{CIDER} = L_{dis} + \lambda_c L_{comp} \quad \longrightarrow$$



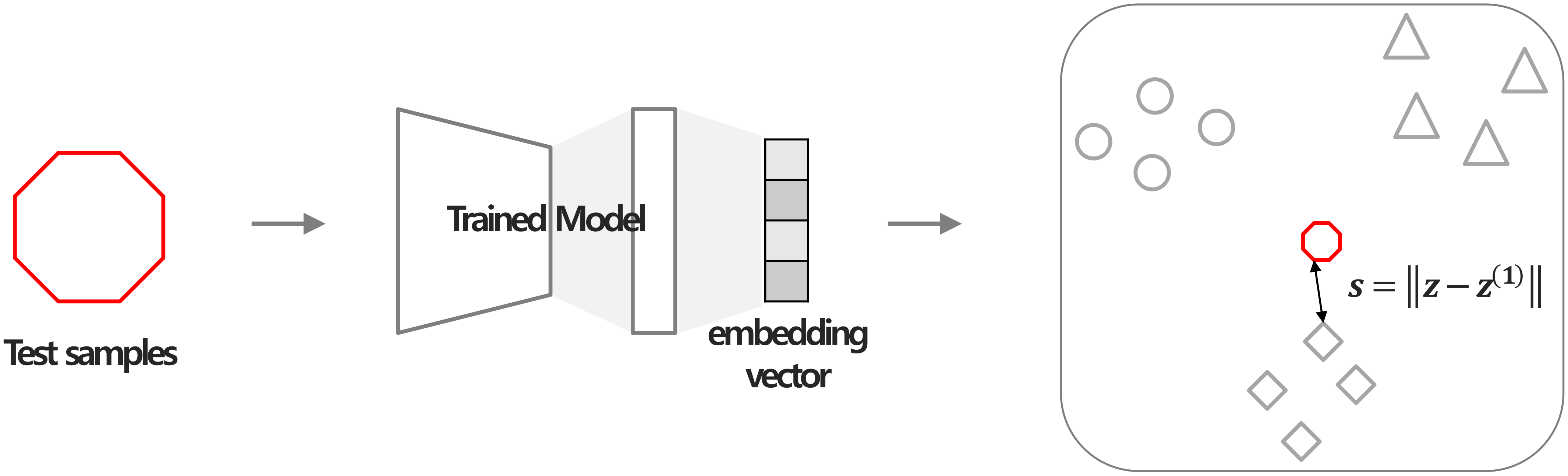
High intra-class compactness

And, high inter-class dispersion

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

❖ 인퍼런스 단계에서는 k번째 이웃과의 거리 (KNN+ 방법론)를 OOD score로 활용함



Example of K=1

Sun, Y., Ming, Y., Zhu, X., & Li, Y. (2022, June). Out-of-distribution detection with deep nearest neighbors. In *International conference on machine learning* (pp. 20827-20840). PMLR.

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

Table 1: OOD detection performance for CIFAR-100 (ID) with ResNet-34. Training with CIDER significantly improves OOD detection performance.

Method	SVHN		Places365		LSUN		iSUN		Texture		Average	
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑
Without Contrastive Learning												
MSP	78.89	79.80	84.38	74.21	83.47	75.28	84.61	74.51	86.51	72.53	83.12	75.27
ODIN	70.16	84.88	82.16	75.19	76.36	80.10	79.54	79.16	85.28	75.23	78.70	79.11
Mahalanobis	87.09	80.62	84.63	73.89	84.15	79.43	83.18	78.83	61.72	84.87	80.15	79.53
Energy	66.91	85.25	81.41	76.37	59.77	86.69	66.52	84.49	79.01	79.96	70.72	82.55
GODIN	74.64	84.03	89.13	68.96	93.33	67.22	94.25	65.26	86.52	69.39	87.57	70.97
LogitNorm	59.60	90.74	80.25	78.58	81.07	82.99	84.19	80.77	86.64	75.60	78.35	81.74
With Contrastive Learning												
ProxyAnchor	87.21	82.43	70.10	79.84	37.19	91.68	70.01	84.96	65.64	84.99	66.03	84.78
CE + SimCLR	24.82	94.45	86.63	71.48	56.40	89.00	66.52	83.82	63.74	82.01	59.62	84.15
CSI	44.53	92.65	79.08	76.27	75.58	83.78	76.62	84.98	61.61	86.47	67.48	84.83
SSD+	31.19	94.19	77.74	79.90	79.39	85.18	80.85	84.08	66.63	86.18	67.16	85.90
KNN+	39.23	92.78	80.74	77.58	48.99	89.30	74.99	82.69	57.15	88.35	60.22	86.14
CIDER	23.09	95.16	79.63	73.43	16.16	96.33	71.68	82.98	43.87	90.42	**46.89**	**87.67**

The five OOD dataset columns (SVHN, Places365, LSUN, iSUN, Texture) are grouped under the header "OOD Dataset".

Paper 1

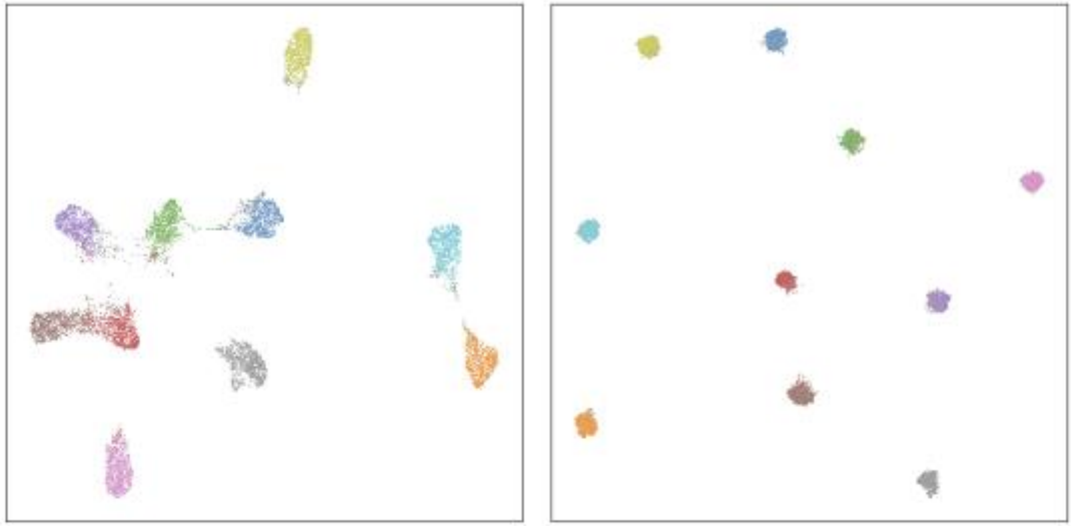
How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)

Table 3: Ablation study on loss component. Results (in AUROC) are based on CIFAR-100 trained with ResNet-34. Training with only $\mathcal{L}_{\text{comp}}$ suffices for ID classification. Inter-class dispersion induced by $\mathcal{L}_{\text{dis}}$ is key to OOD detection.

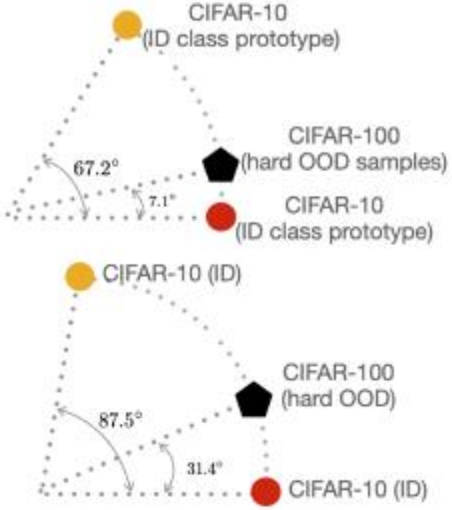
Loss Components		**AUROC↑**						**ID ACC↑**
$\mathcal{L}_{\text{comp}}$	$\mathcal{L}_{\text{dis}}$	Places365	LSUN	iSUN	Texture	SVHN	AVG	
✓		79.63	85.75	84.45	87.21	91.33	85.67	75.19
	✓	54.76	69.81	54.99	44.26	46.48	54.06	2.03
✓	✓	**73.43**	**96.33**	**82.98**	**90.42**	**95.16**	**87.67**	**75.35**

Paper 1

How To Exploit Hyperspherical Embeddings for Out-of-Distribution Detection (2023, ICLR)



(a) ID embeddings of CE (left) and CIDER (right)



(b) ID & OOD of CE (top) and CIDER (down)

Figure 3: (a): UMAP (McInnes et al., 2018) visualization of the features when the model is trained with CE vs. CIDER for CIFAR-10 (ID). (b): CIDER makes OOD samples more separable from ID compared to CE (*c.f.* Table 4).

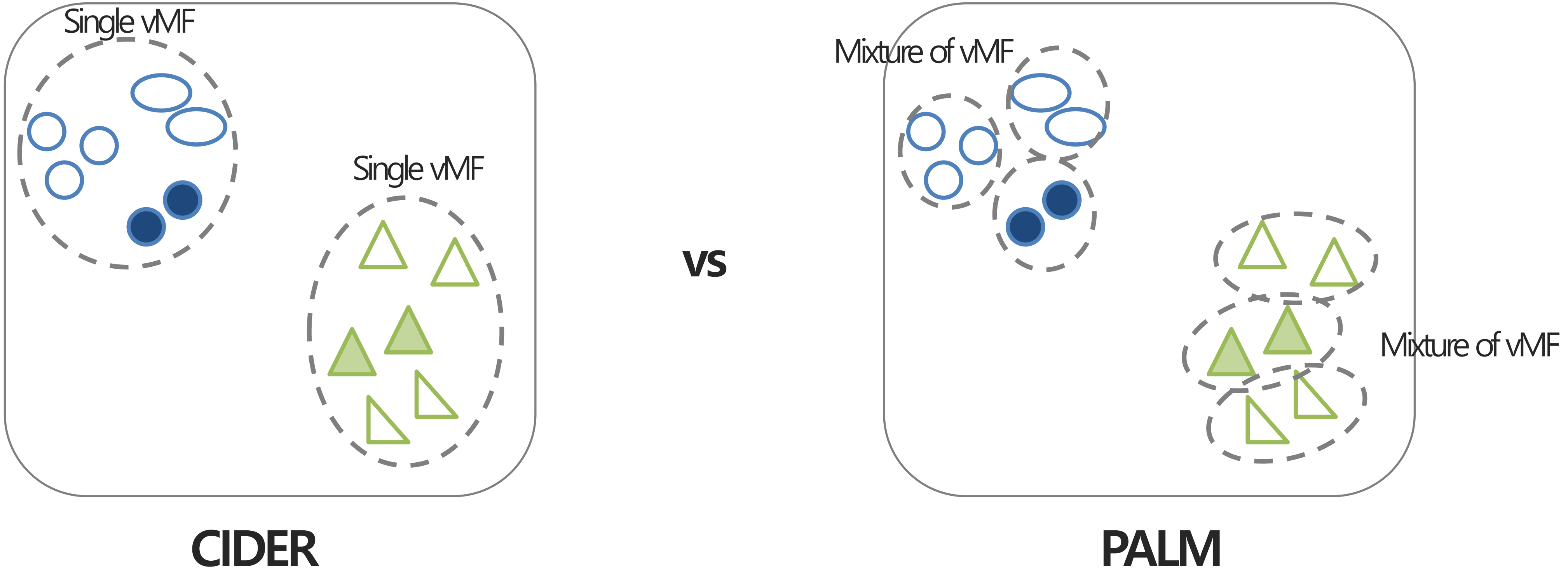
Learning with Mixture of Prototypes for Out-of-Distribution Detection

(2024, ICLR)

Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

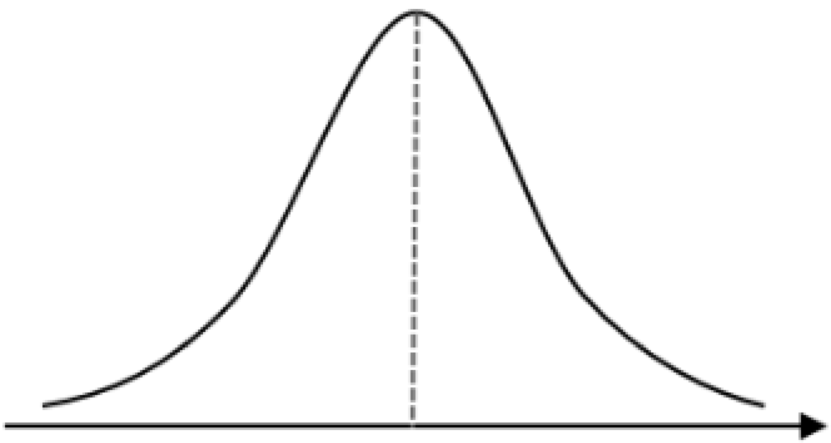
- ❖ CIDER는 한 클래스를 하나의 prototype으로 모델링하여 클래스 내의 다양성 (intra-class diversity)를 무시함
- ❖ 이를 해결하기 위해 저자들은 PrototypicAl Learning with a Mixture of prototypes (PALM) 방법론을 제안
- ❖ 각 클래스마다 single vMF 분포로 모델링하는 것이 아닌 Mixture of vMF로 모델링하여 더 정교한 embedding space 학습



Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

$$p(x) = N(x; \mu, \Sigma)$$

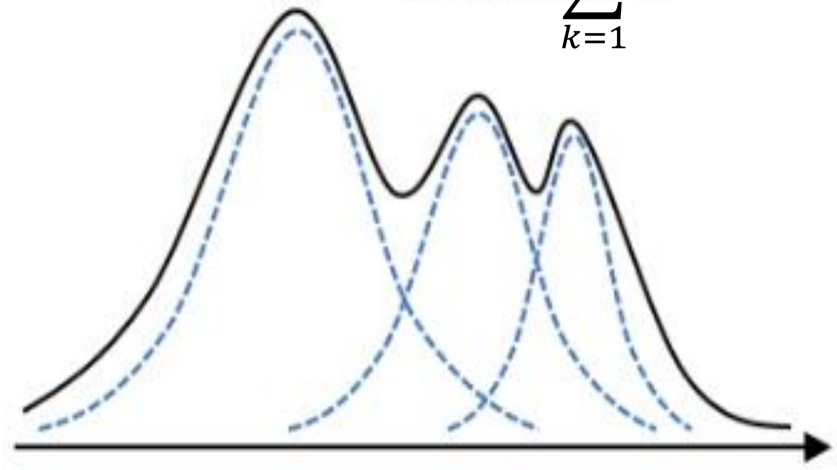


Single Gaussian distribution

vs

$$p(x) = \sum_{k=1}^{K} w_k \, N(x; \mu_k, \Sigma_k)$$

$$, where \le w_k \le 1 \; and \; \sum_{k=1}^{K} w_k = 1$$



Mixture of Gaussian distributions

Paper 2

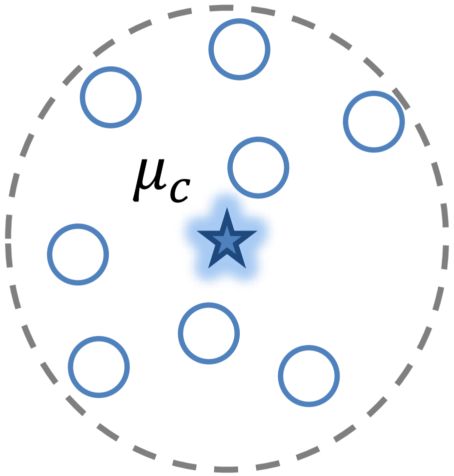
Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

$$p(z; \mu_c, \kappa) = Z(\kappa)\exp(\mu_c^T z/\tau)$$

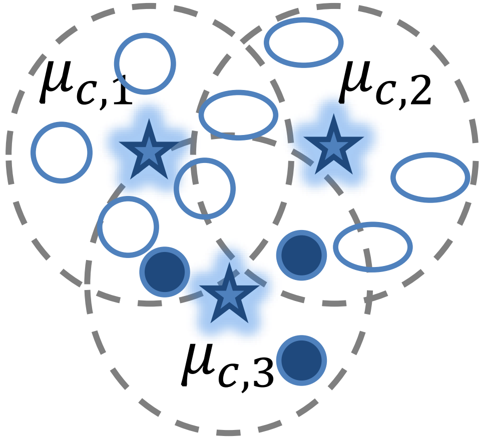
$$p(z; w_c, M_c, \kappa) = \sum_{k=1}^{K} w_{c,k}\, Z(\kappa)\exp(\mu_{c,k}^T z/\tau)$$

$$, where \le w_{c,k} \le 1 \ and \ \sum_{k=1}^{K} w_{c,k} = 1$$

$M_c = \{\mu_{c,k}\}_{k=1}^{K} = Class\ c$에 대한 K개의 $prototype$



vs



Single vMF distribution **Mixture of vMF distributions**

Paper 2

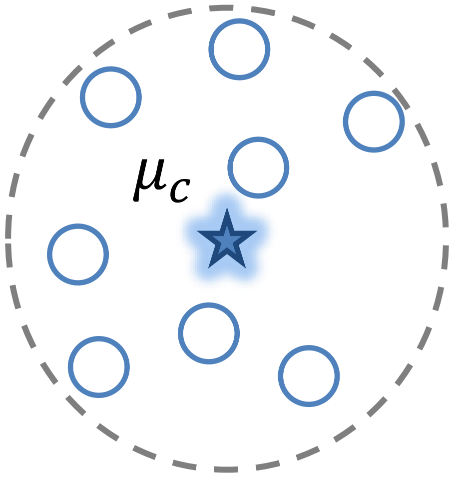
Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

$$p(z;\mu_c,\kappa) = Z(\kappa)\exp(\mu_c^T z/\tau)$$

$$p(z;w_c,M_c,\kappa) = \sum_{k=1}^{K} w_{c,k}\, Z(\kappa)\exp(\mu_{c,k}^T z/\tau)$$

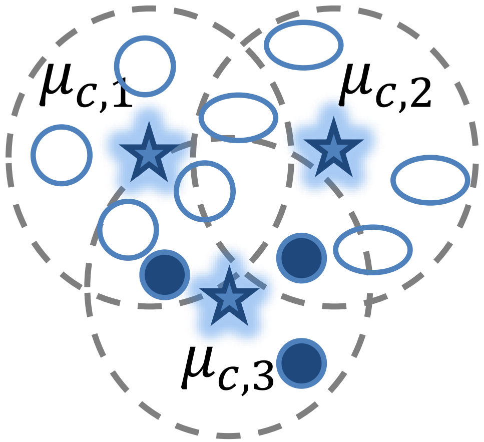
$$,where \le w_{c,k} \le 1\ and\ \sum_{k=1}^{K} w_{c,k} = 1$$

$$M_c = \{\mu_{c,k}\}_{k=1}^{K} = Class\ c\text{에 대한 } K\text{개의 } prototype$$



Single vMF distribution

vs



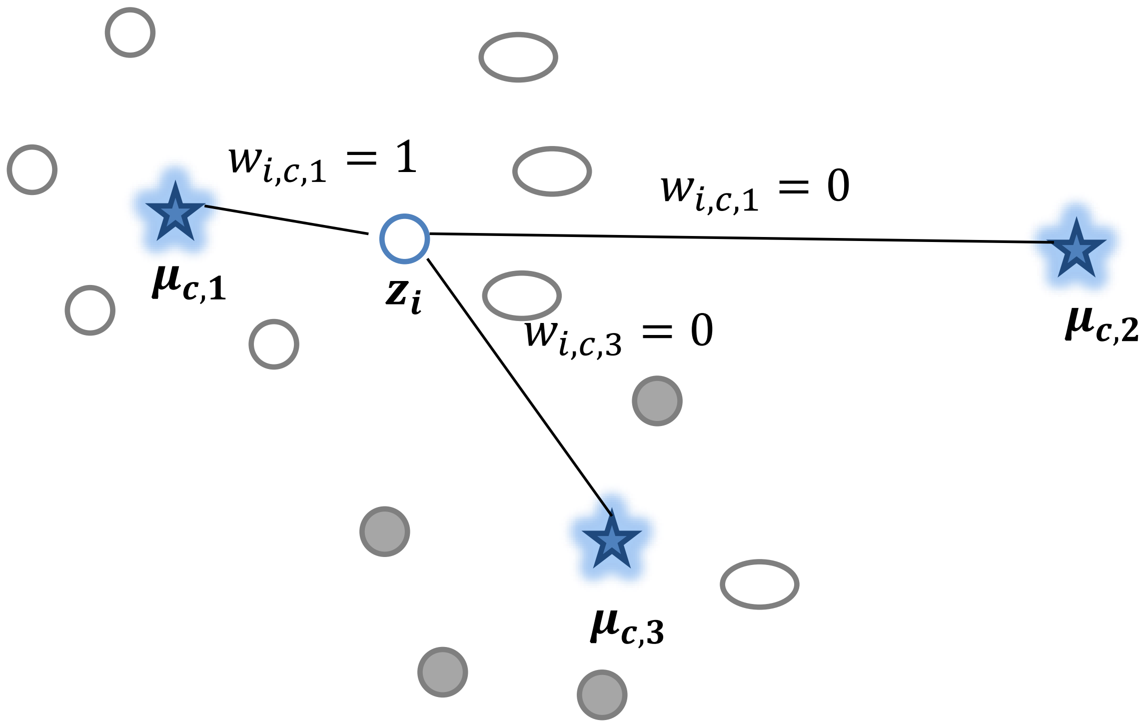
Mixture of vMF distributions

Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

❖ 클래스 c에 속하는 어떤 샘플의 embedding vector z와 각 prototype이 주어졌을 때, 할당 계수 w를 어떻게 부여할 것인가?

- Embedding space 상에서 가까운 prototype에 높은 계수를 부여해야 함

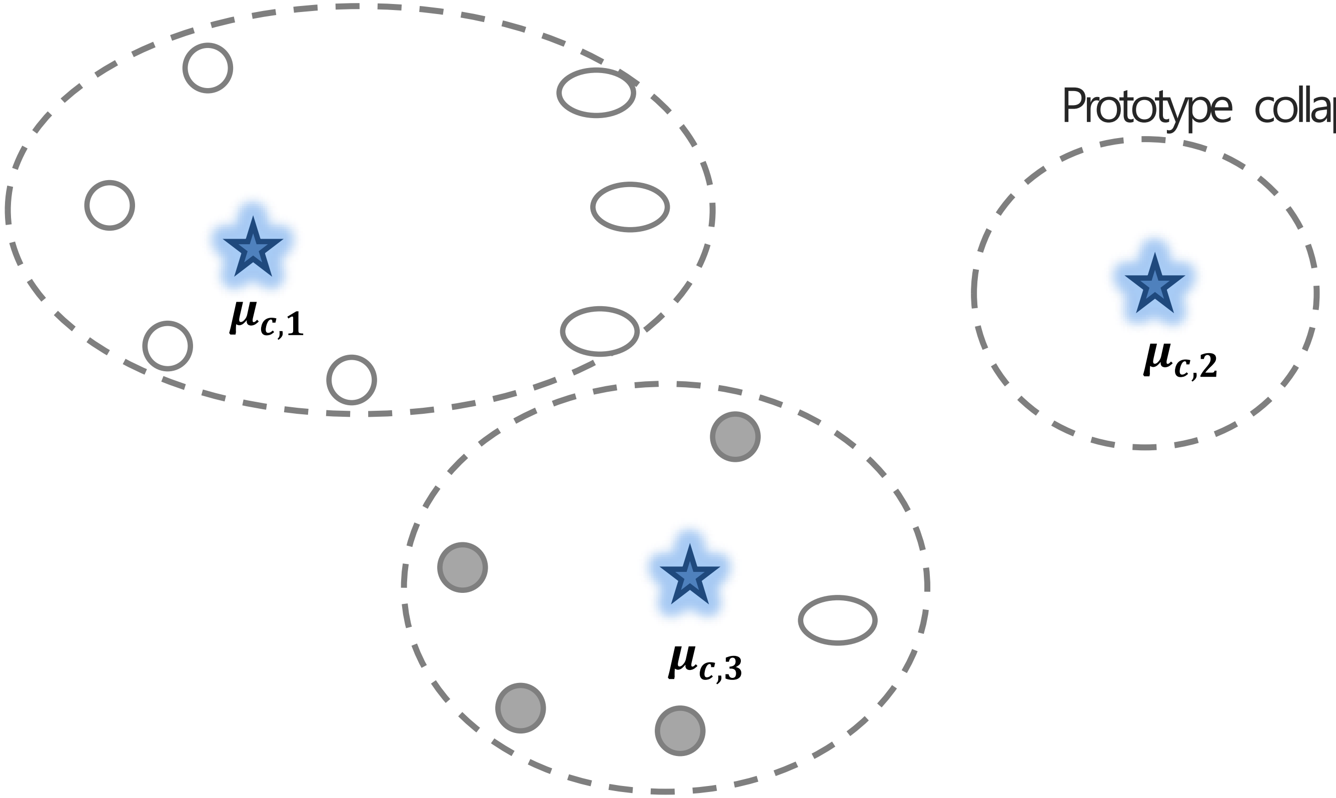


Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

❖ 클래스 c에 속하는 어떤 샘플의 embedding vector z와 각 prototype이 주어졌을 때, 할당 계수 w를 어떻게 부여할 것인가?

- Embedding space 상에서 가까운 prototype에 높은 계수를 부여해야 함

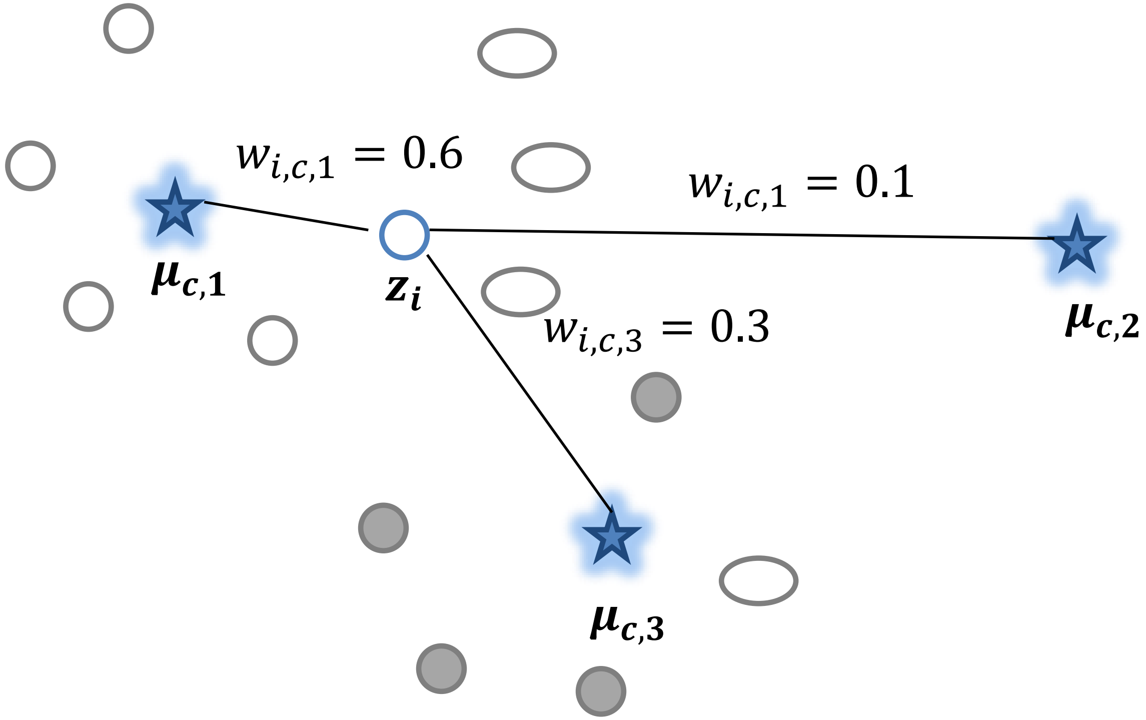


Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

❖ 클래스 c에 속하는 어떤 샘플의 embedding vector z와 각 prototype이 주어졌을 때, 할당 계수 w를 어떻게 부여할 것인가?

- Embedding space 상에서 가까운 prototype에 높은 계수를 부여
- SwAV (Caron et al., 2020)에서 제안한 Soft assignment를 통해 prototype collapse를 방지



$$\max_{W_c} Tr(W_c^T M_c^T Z_c) + \epsilon R(W_c)$$

$$M_c = [\mu_{c,1}, \mu_{c,2}, \mu_{c,3}] \in \mathbb{R}^{D\times K}$$

$$Z_c = [z_1, \dots, z_i, \dots, z_{B_c}] \in \mathbb{R}^{D\times B_c}$$

$$W_c = [w_1^c, \dots, w_i^c, \dots, w_{B_c}^c] \in \mathbb{R}^{K\times B_c}$$

$$R(\cdot) = entropy\ function$$

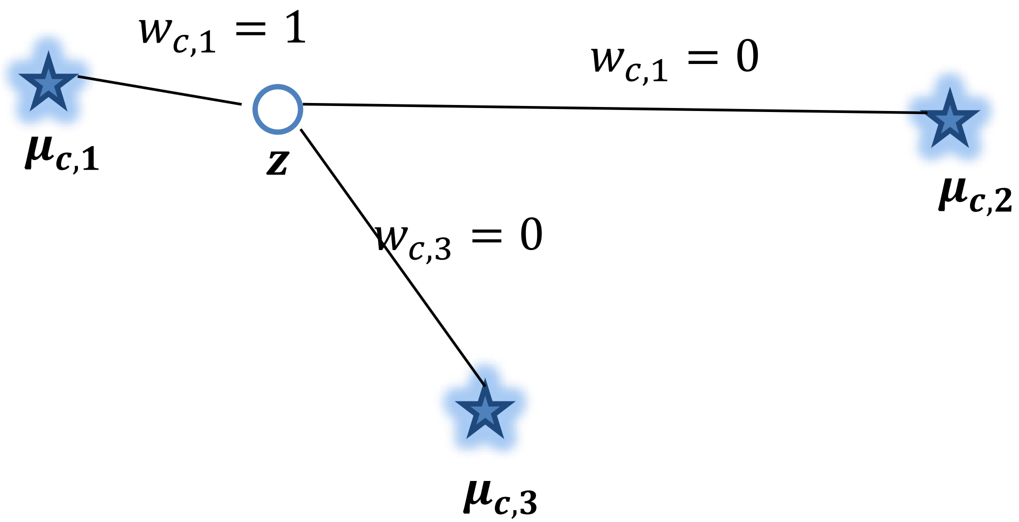
Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33, 9912-9924.

Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

❖ 클래스 c에 속하는 어떤 샘플의 embedding vector z와 각 prototype이 주어졌을 때, 할당 계수 w를 어떻게 부여할 것인가?

- Embedding space 상에서 가까운 prototype에 높은 계수를 부여
- SwAV (Caron et al., 2020)에서 제안한 Soft assignment를 통해 prototype collapse를 방지



$$\max_{W_c} Tr(W_c^T M_c^T Z_c) + \epsilon R(W_c)$$

$$Term\ 1 = Tr(w^T M_c^T z)$$

$$= Tr\left(\begin{bmatrix} w_{c,1} & w_{c,2} & w_{c,3} \end{bmatrix} \begin{bmatrix} \mu_{c,1}^T z \\ \mu_{c,2}^T z \\ \mu_{c,3}^T z \end{bmatrix} \right)$$

$$= w_{c,1} \cdot s_1 + w_{c,2} \cdot s_2 + w_{c,3} \cdot s_3$$

$$s.t.,\ w_{c,1} + w_{c,2} + w_{c,3} = 1$$

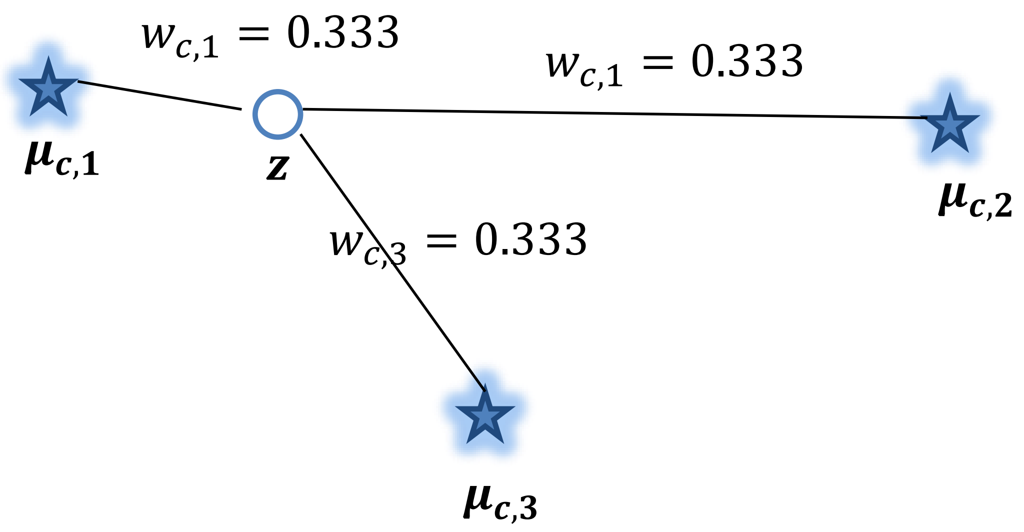
Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33, 9912-9924.

Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

❖ 클래스 c에 속하는 어떤 샘플의 embedding vector z와 각 prototype이 주어졌을 때, 할당 계수 w를 어떻게 부여할 것인가?

- Embedding space 상에서 가까운 prototype에 높은 계수를 부여
- SwAV (Caron et al., 2020)에서 제안한 Soft assignment를 통해 prototype collapse를 방지



$$\max_{W_c} Tr(W_c^T M_c^T Z_c) + \epsilon R(W_c)$$

$$Term\ 2 = entropy\ of \begin{bmatrix} w_{c,1} & w_{c,2} & w_{c,3} \end{bmatrix}$$

$$s.t., w_{c,1} + w_{c,2} + w_{c,3} = 1$$

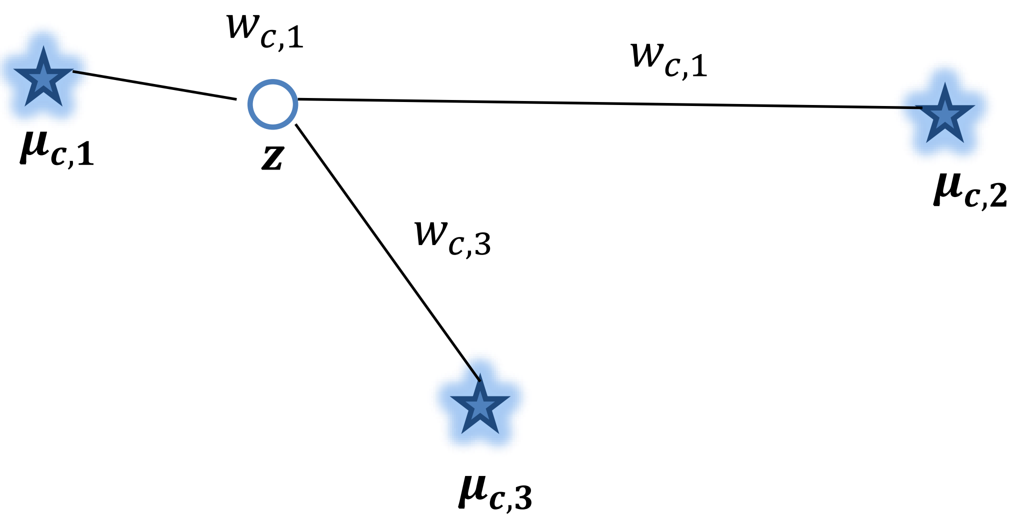
Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33, 9912-9924.

Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

❖ 클래스 c에 속하는 어떤 샘플의 embedding vector z와 각 prototype이 주어졌을 때, 할당 계수 w를 어떻게 부여할 것인가?

- Embedding space 상에서 가까운 prototype에 높은 계수를 부여
- SwAV (Caron et al., 2020)에서 제안한 Soft assignment를 통해 prototype collapse를 방지



$$\max_{W_c} Tr(W_c^T M_c^T Z_c) + \epsilon R(W_c)$$

$$s.t., w_{c,1} + w_{c,2} + w_{c,3} = 1$$

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33, 9912-9924.

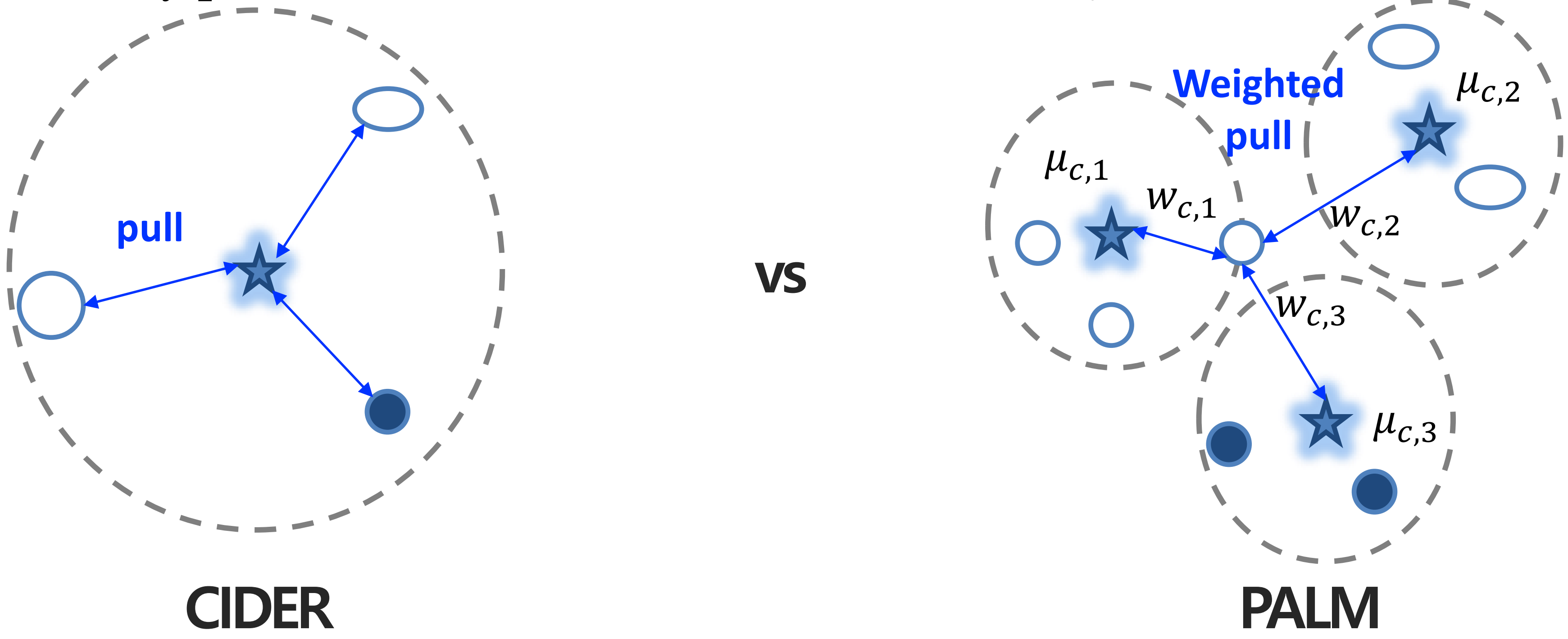
Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

- ❖ CIDER와 비슷하게 class prototype를 기준으로 그 클래스에 해당하는 embedding vector를 모아주는 compactness loss 존재
- ❖ 한 클래스 내에서 각 prototype의 **할당 계수에 따라 가중치를 부여하여** 좀 더 비슷한 prototype으로 이동하도록 설계

$$L_{comp} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mu_{c(i)}^{T} z_i/\tau)}{\sum_{j=1}^{C}\exp(\mu_j^{T} z_i/\tau)}$$

$$L_{comp} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\sum_{k=1}^{K} w_{i,c,k}\exp(\mu_{c(i),k'}^{T}, z_i/\tau)}{\sum_{j=1}^{C}\sum_{k'=1}^{K} w_{i,j,k'}\exp(\mu_{j,k'}^{T} z_i/\tau)}$$



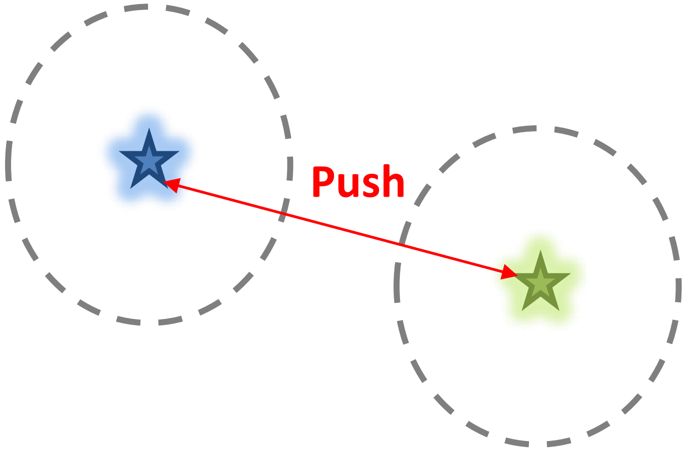
Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

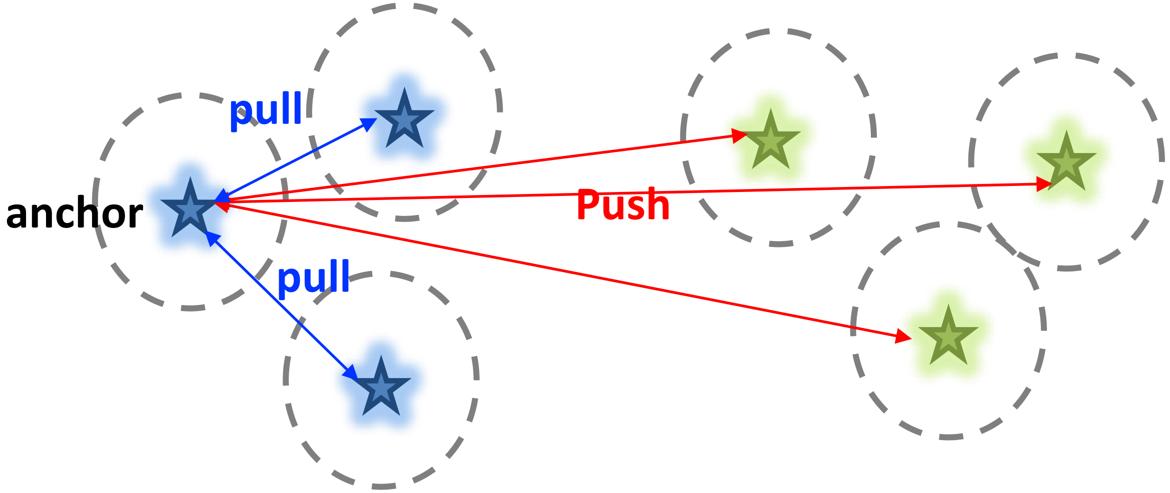
❖ 또한, 서로 다른 클래스 prototype 사이는 밀어내면서, 같은 클래스 내 prototype들은 당겨주는 contrastive loss 설계

$$L_{dis} = \frac{1}{C}\sum_{i=1}^{C}\log\frac{1}{C-1}\sum_{j=1}^{C}\mathbb{1}\{j \neq i\}e^{\mu_i^T\mu_j/\tau}$$

$$L_{proto-contra} = -\frac{1}{CK}\sum_{c=1}^{C}\sum_{k=1}^{K}\log\frac{\sum_{k'=1}^{K}\mathbb{1}\{k' \neq k\}\, e^{\mu_{c,k}^T\mu_{c,k'}/\tau}}{\sum_{c'=1}^{C}\sum_{k''=1}^{K}\mathbb{1}\{k'' \neq k\}e^{\mu_{c,k}^T\mu_{c',k''}/\tau}}$$



VS



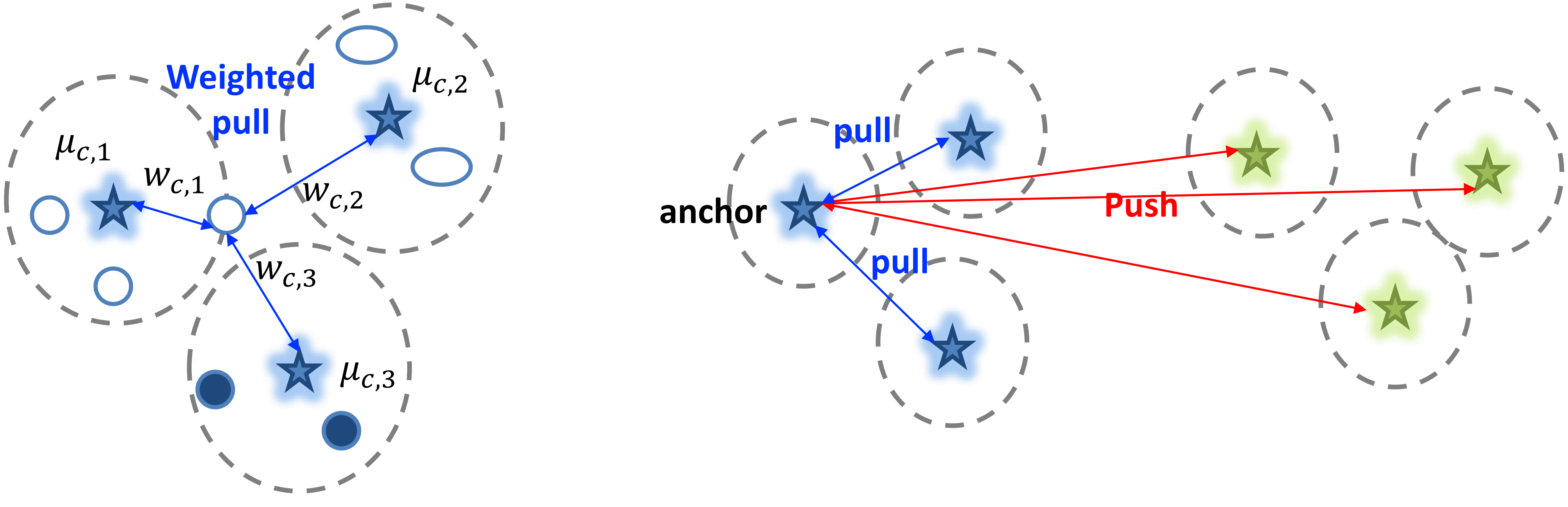
CIDER

PALM

Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

$$L_{PALM} = L_{comp} + \lambda L_{proto-contra}$$

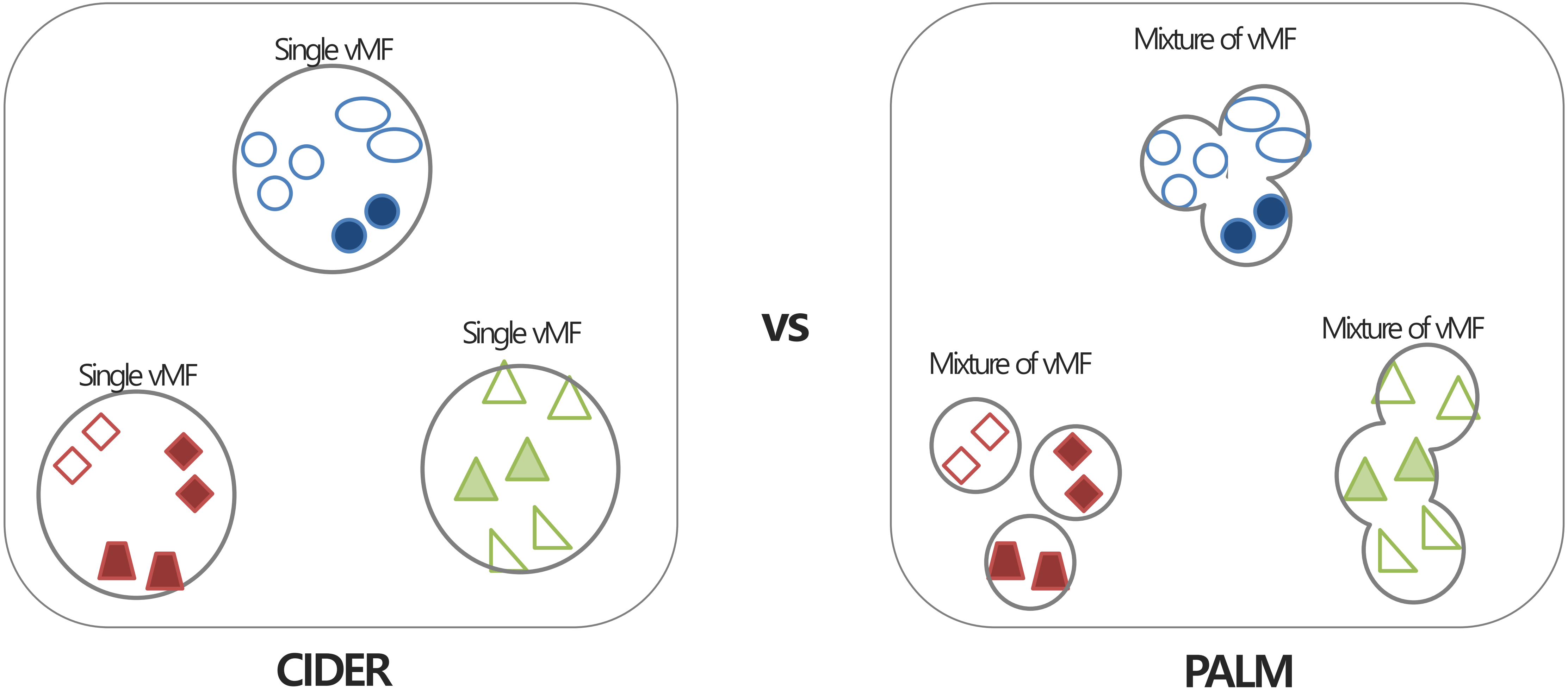


Prototype-Sample Compactness

Prototype-Prototype Compactness & Dispersion

Paper 2

Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)



Paper 2

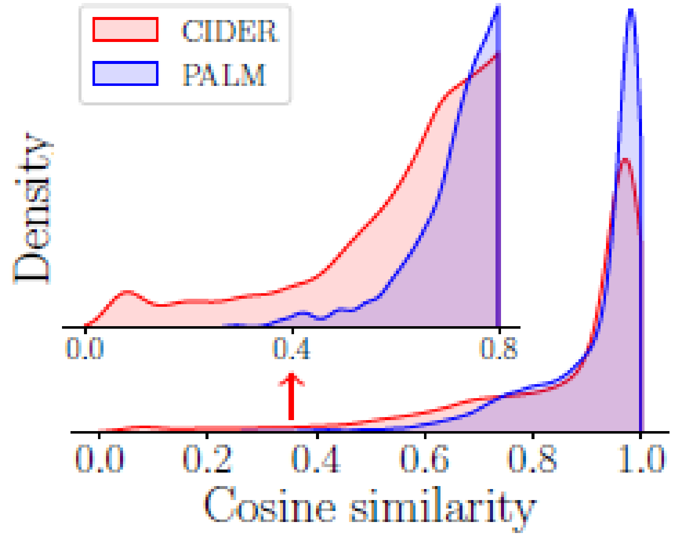
Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)

Table 1: OOD detection performance on methods trained on **labeled** CIFAR-100 as ID dataset using backbone network of ResNet-34. ↓ means smaller values are better and ↑ means larger values are better. **Bold** numbers indicate superior results.

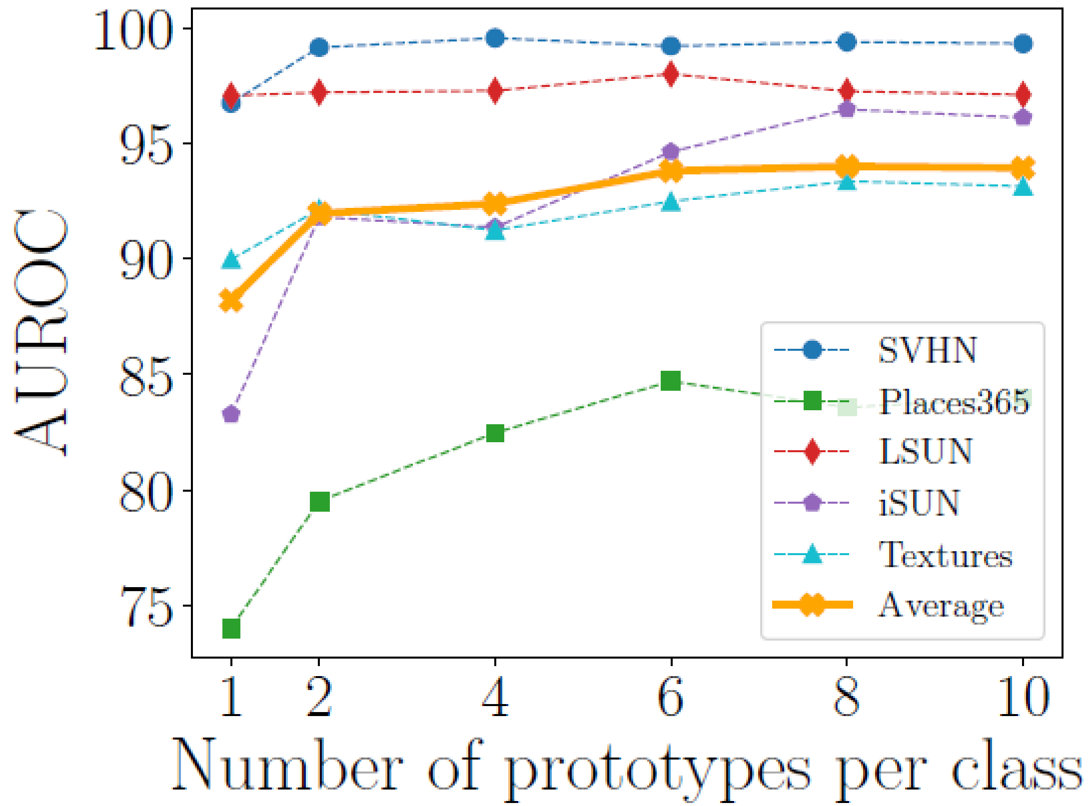
Methods	SVHN		Places365		OOD Datasets LSUN		iSUN		Textures		Average	
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑
MSP	78.89	79.80	84.38	74.21	83.47	75.28	84.61	74.51	86.51	72.53	83.57	75.27
Vim	73.42	84.62	85.34	69.34	86.96	69.74	85.35	73.16	74.56	76.23	81.13	74.62
ODIN	70.16	84.88	82.16	75.19	76.36	80.1	79.54	79.16	85.28	75.23	78.70	78.91
Energy	66.91	85.25	81.41	76.37	59.77	86.69	66.52	84.49	79.01	79.96	70.72	82.55
VOS	43.24	82.8	76.85	78.63	73.61	84.69	69.65	86.32	57.57	87.31	64.18	83.95
CSI	44.53	92.65	79.08	76.27	75.58	83.78	76.62	84.98	61.61	86.47	67.48	84.83
SSD+	31.19	94.19	77.74	79.90	79.39	85.18	80.85	84.08	66.63	86.18	67.16	85.91
kNN+	39.23	92.78	80.74	77.58	48.99	89.30	74.99	82.69	57.15	88.35	60.22	86.14
NPOS	10.62	97.49	67.96	78.81	20.61	92.61	35.94	88.94	**24.92**	91.35	32.01	89.84
CIDER	12.55	97.83	79.93	74.87	30.24	92.79	45.97	88.94	35.55	92.26	40.85	89.34
PALM (ours)	**3.29**	**99.23**	**64.66**	**84.72**	**9.86**	**98.01**	**28.71**	**94.64**	33.56	**92.49**	**28.02**	**93.82**

Paper 2

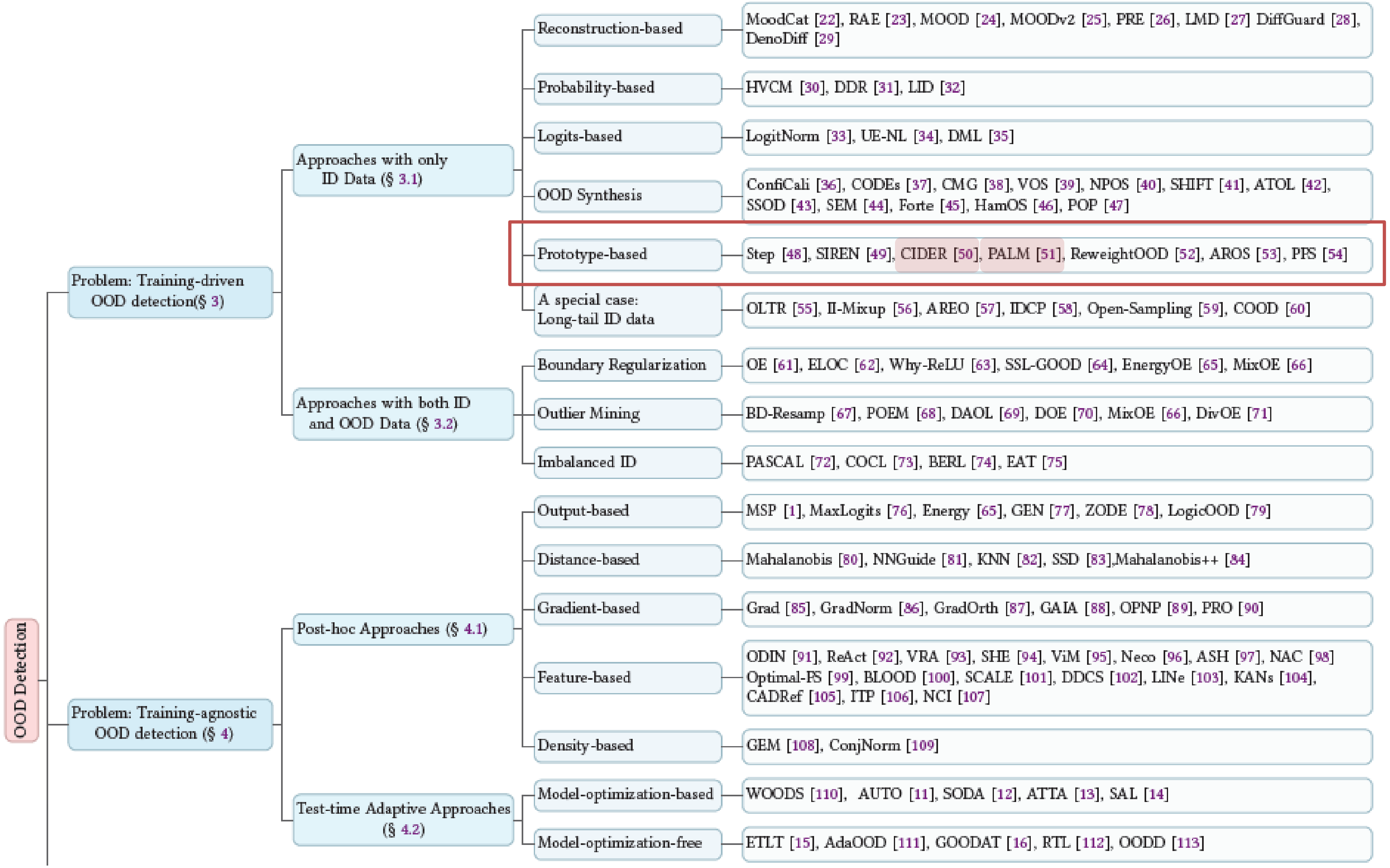
Learning with Mixture of Prototypes for Out-of-Distribution Detection (2024, ICLR)



Methods	Compactness ↓ (in degree)	Number of far ID samples (%) ↓
CIDER	31.08	25.79
PALM	**24.21**	**15.71**



Conclusions



감사합니다