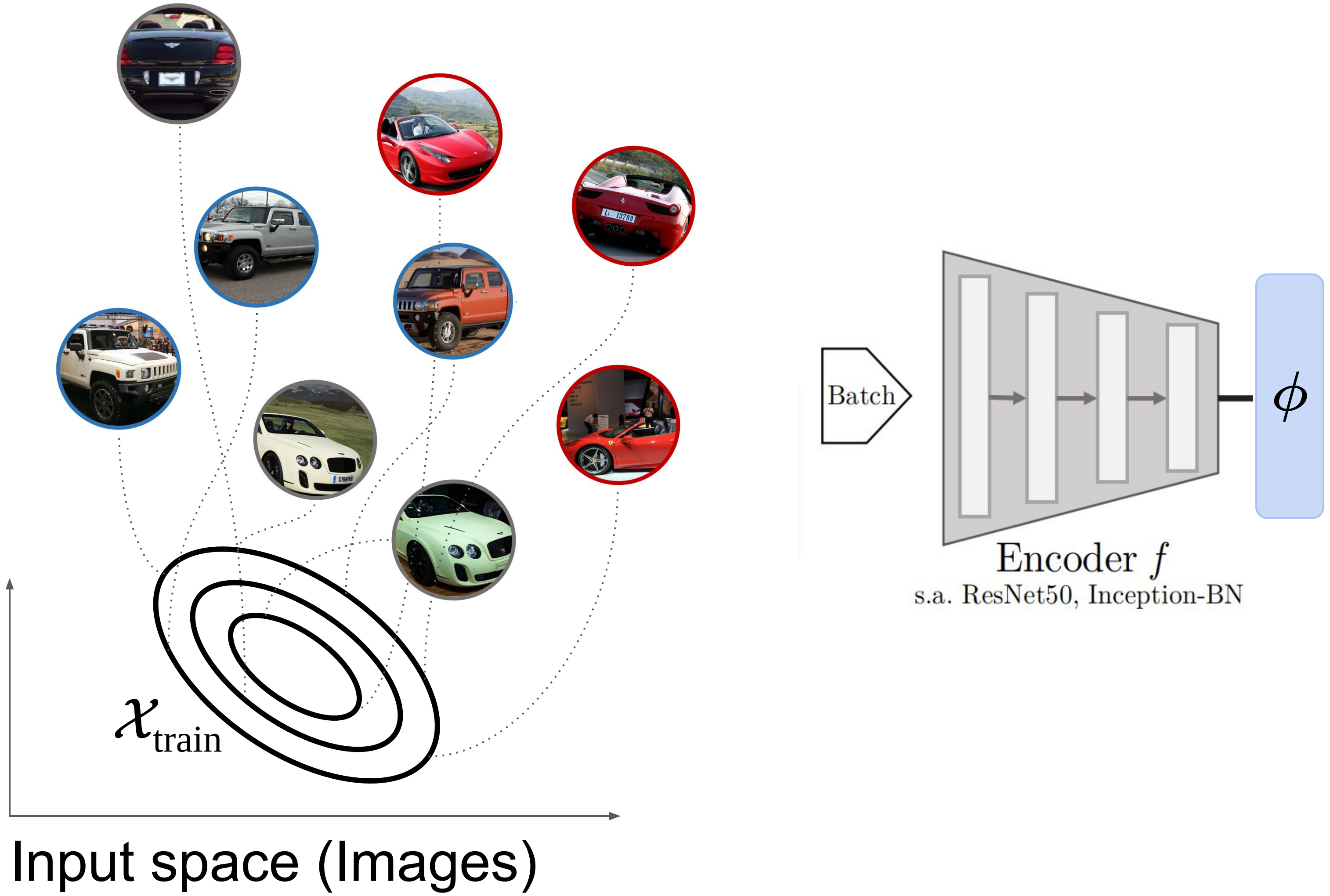
The Family of Objective Functions in DML

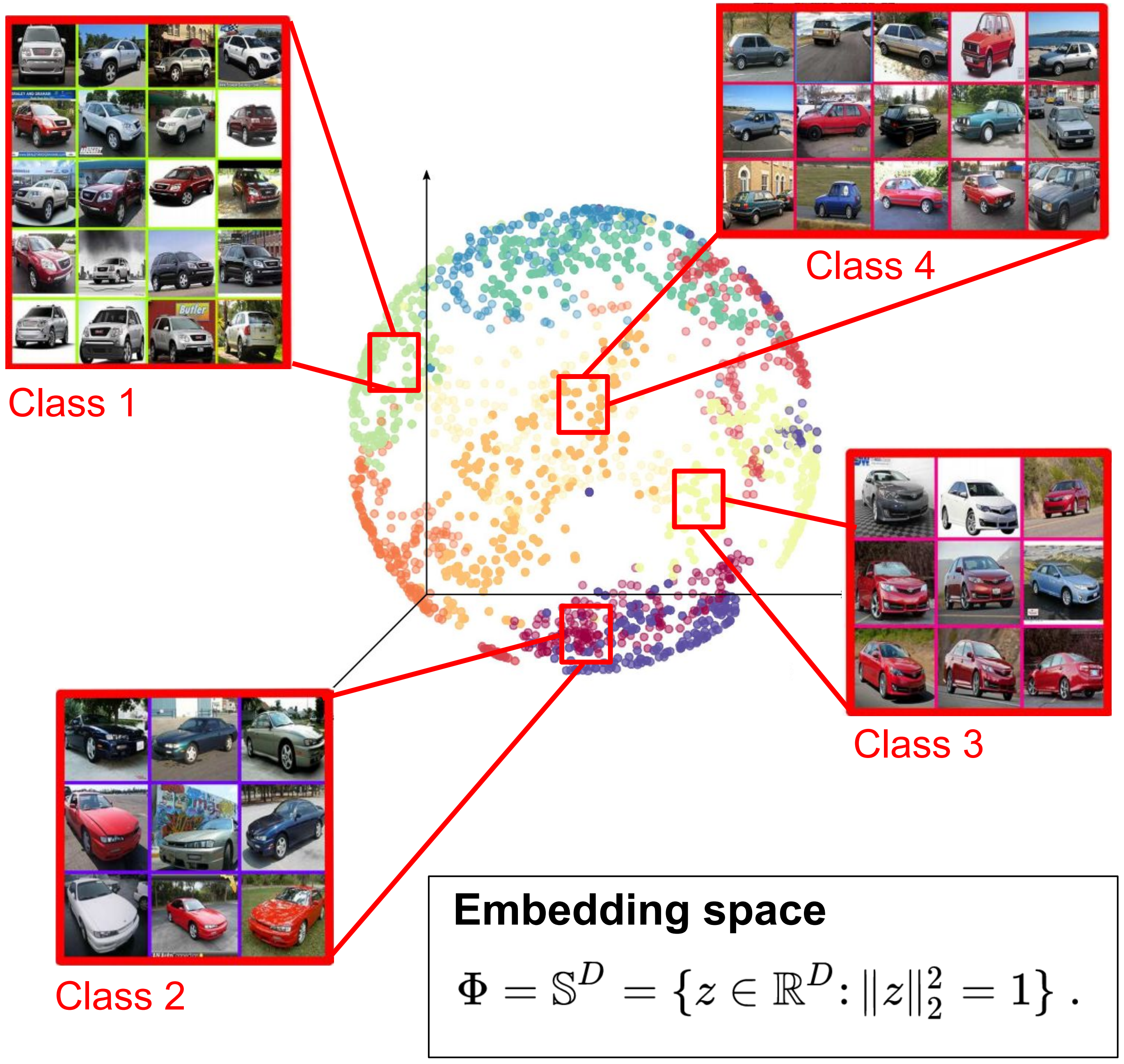
CVPR Tutorial: **Deep Visual Similarity and Metric Learning**

Timo Milbich

Visual Similarity Learning

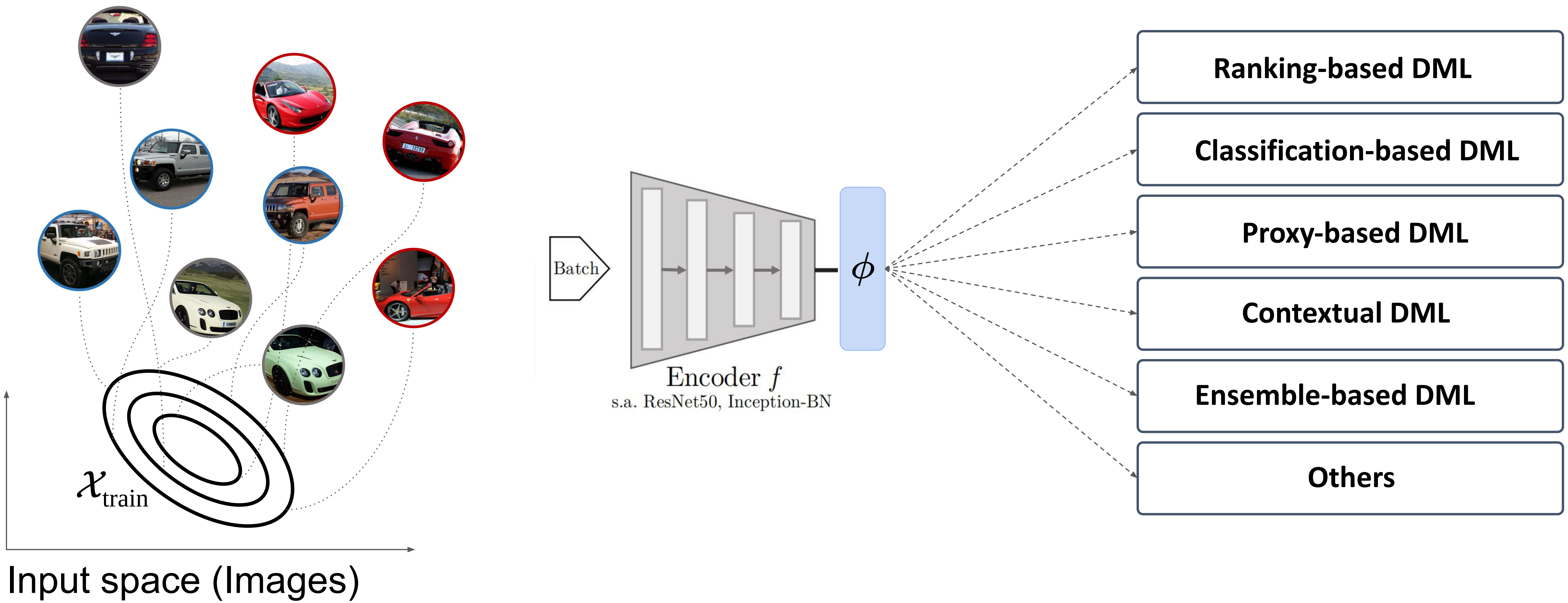
Learn **representation** $\phi(x)$ which reflects **semantic similarity** $d(\phi(x_i), \phi(x_j))$ within training distribution $\mathcal{X}_{\text{train}}$.





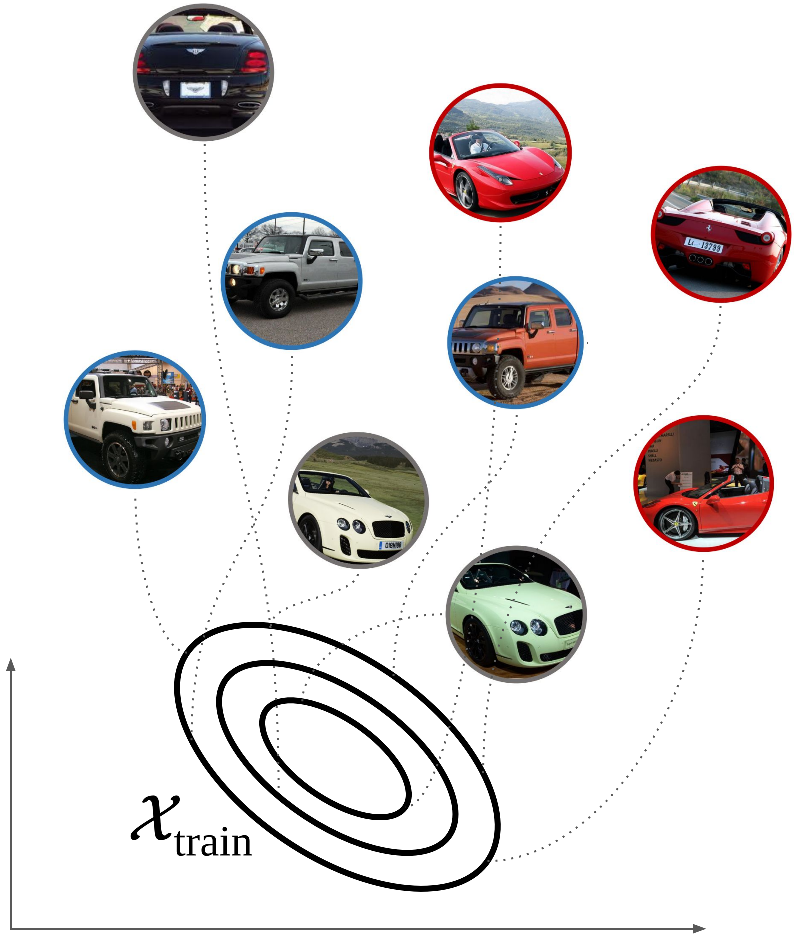
Deep Metric Learning (DML)

Learn **representation** $\phi(x)$ which reflects **semantic similarity** $d(\phi(x_i), \phi(x_j))$ within training distribution $\mathcal{X}_{\text{train}}$.

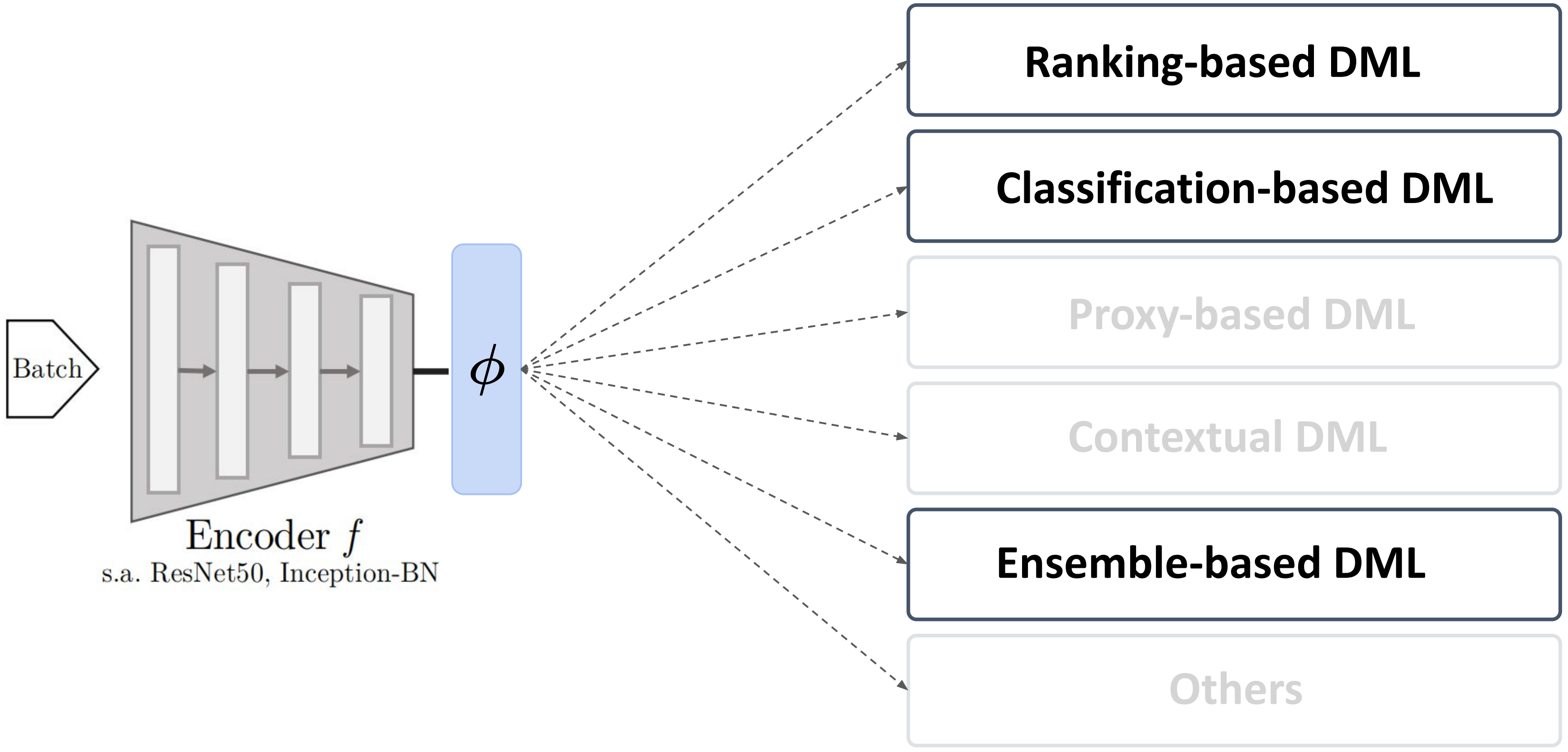


Deep Metric Learning (DML)

Learn **representation** $\phi(x)$ which reflects **semantic similarity** $d(\phi(x_i), \phi(x_j))$ within training distribution $\mathcal{X}_{\text{train}}$.



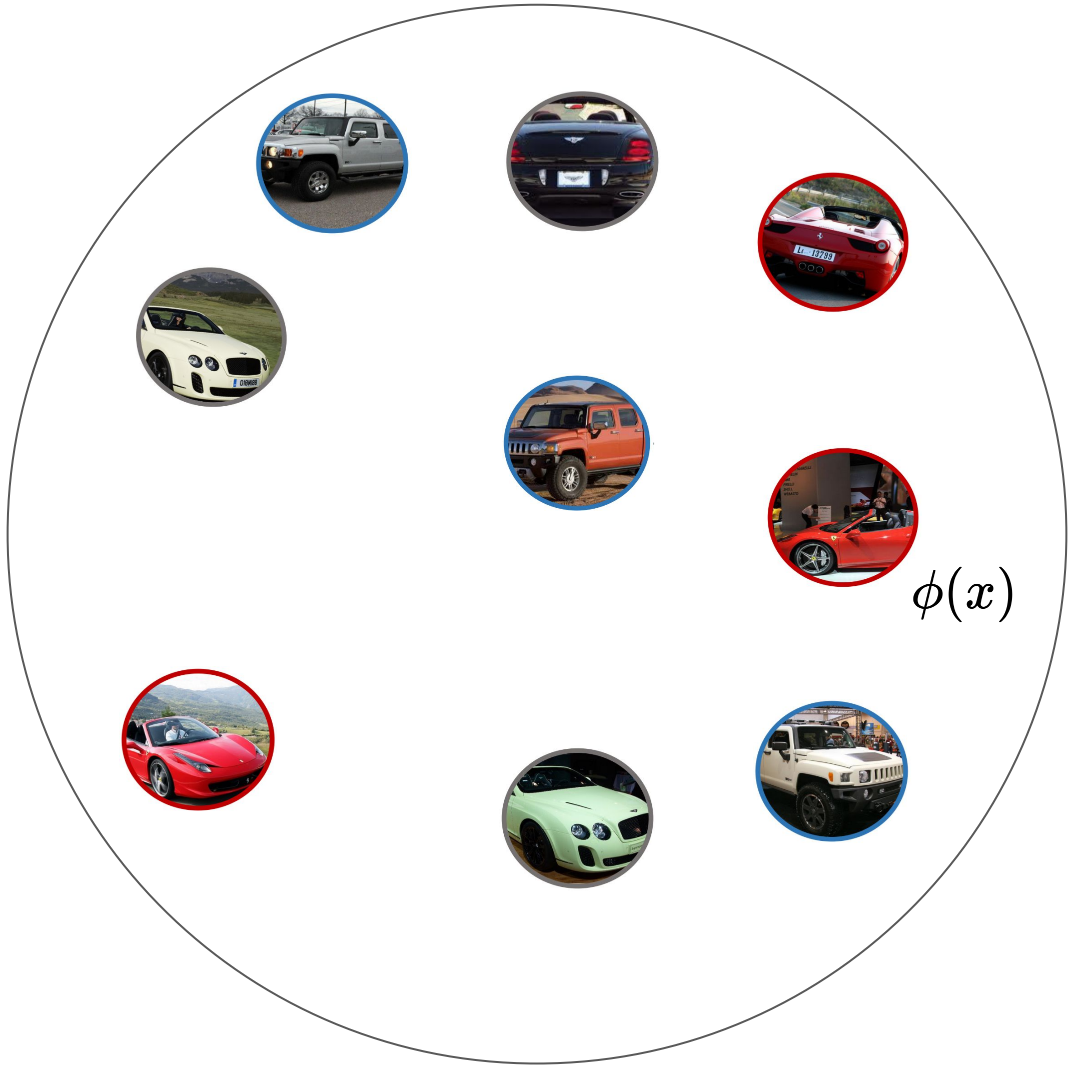
Input space (Images)



Ranking-based DML

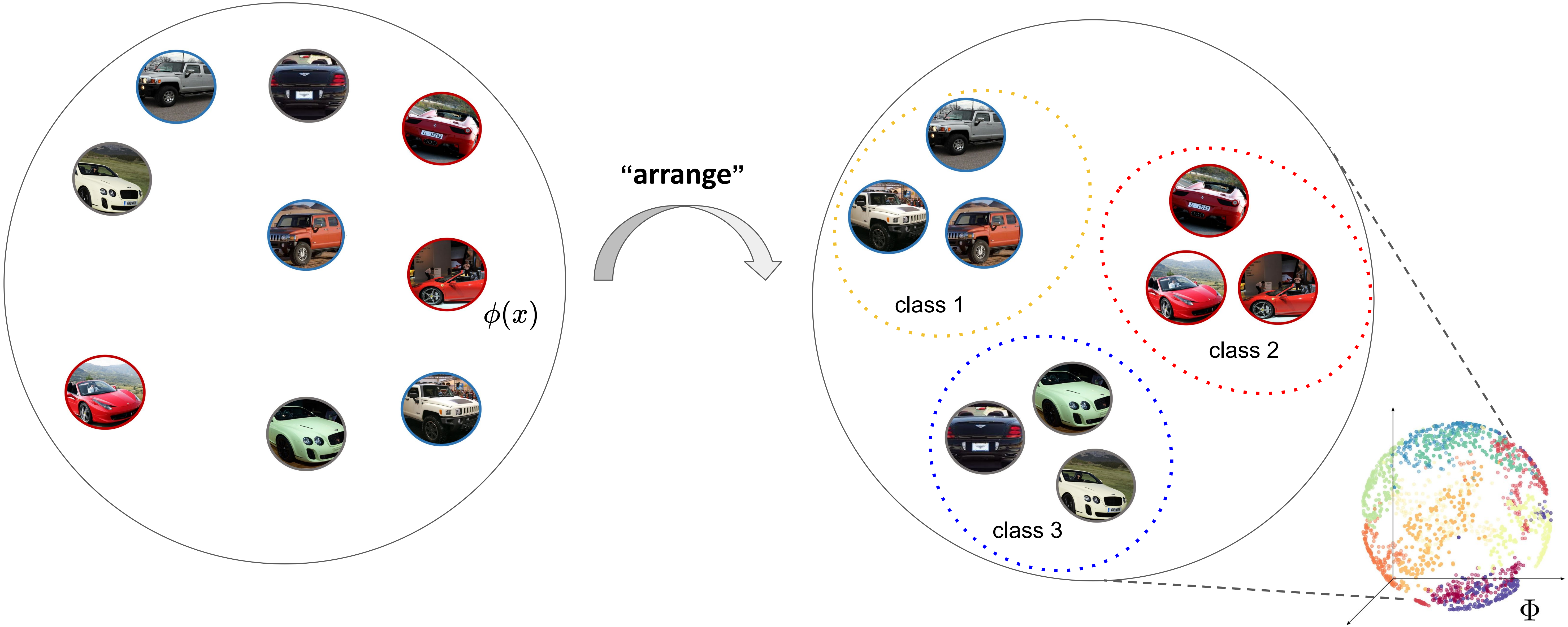
Ranking-based DML

- Learning informative embedding space requires to **“arrange”** data under $\phi(x)$



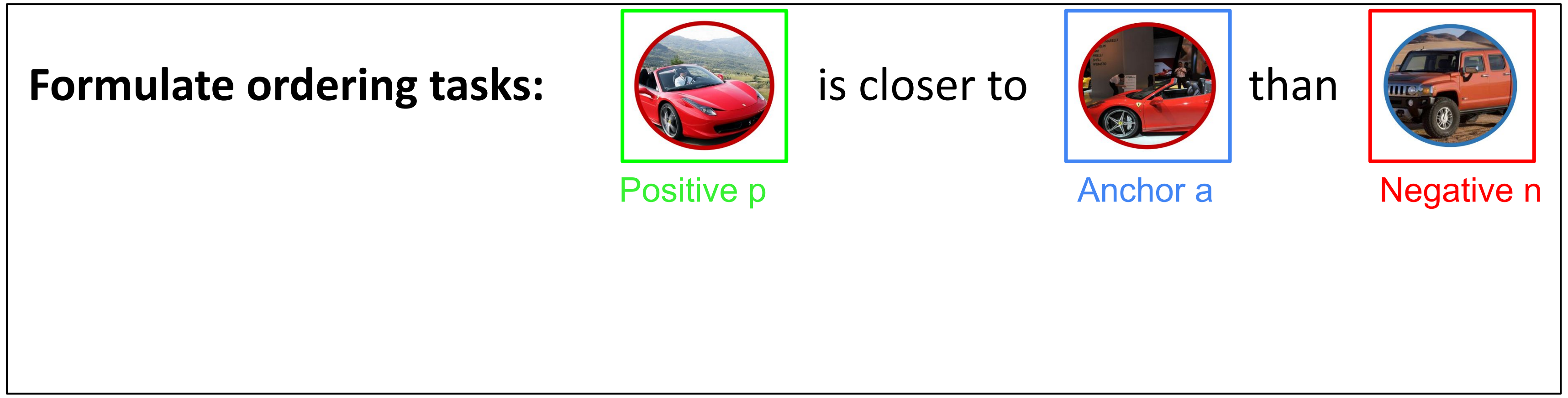
Ranking-based DML

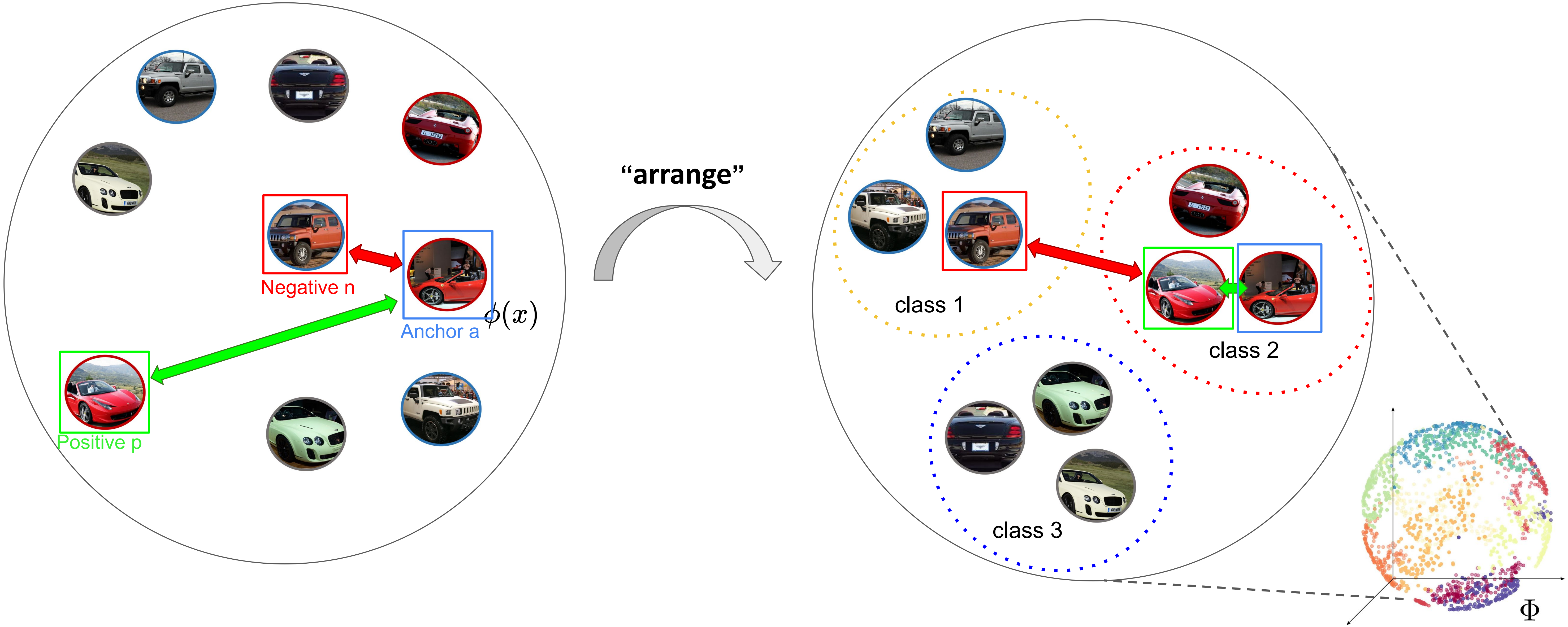
- Learning informative embedding space requires to **"arrange"** data under $\phi(x)$



Ranking-based DML

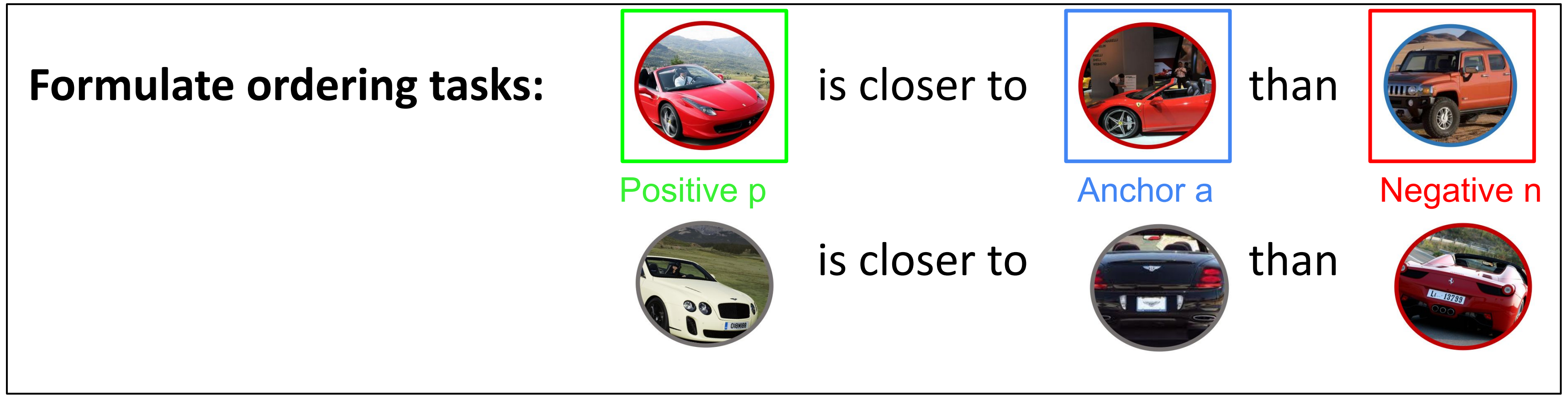
- Learning informative embedding space requires to **“arrange”** data under $\phi(x)$
- Naturally formulate the learning problem as orderings **“A is closer to B then C”**

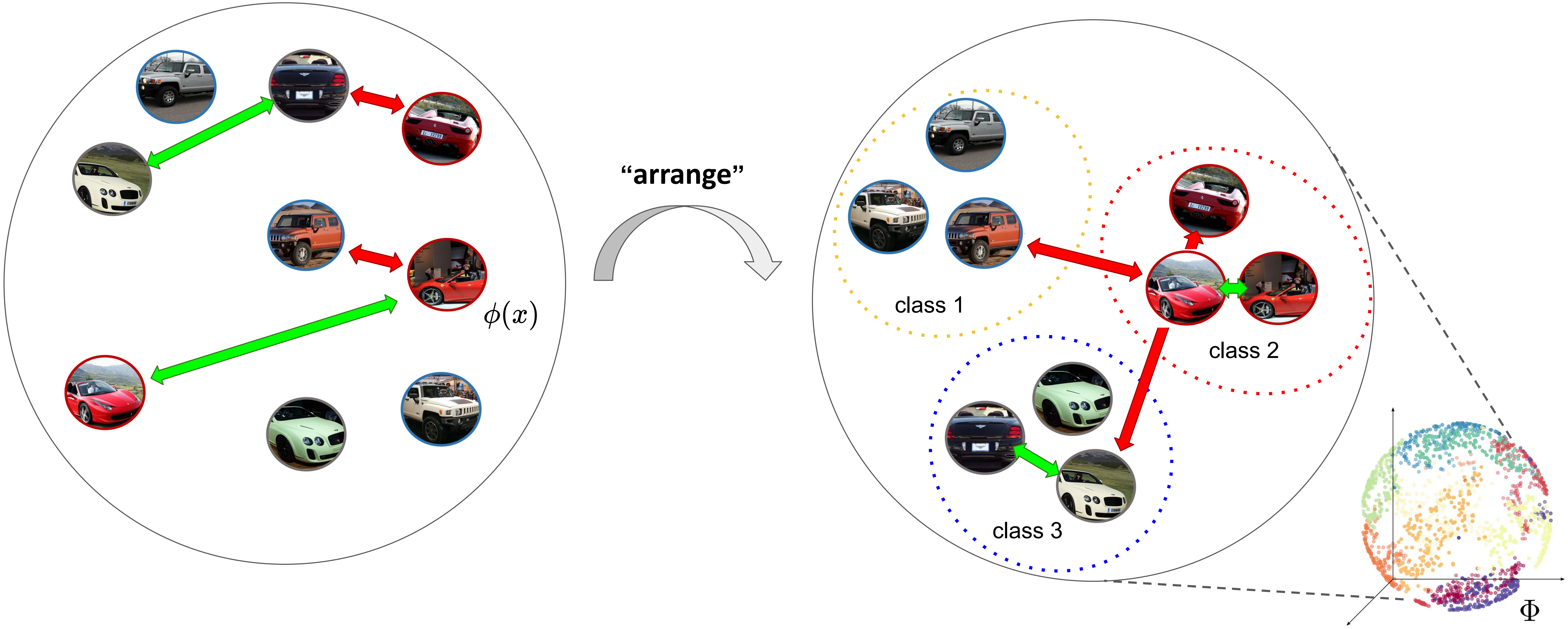




Ranking-based DML

- Learning informative embedding space requires to **“arrange”** data under $\phi(x)$
- Naturally formulate the learning problem as orderings **“A is closer to B then C”**



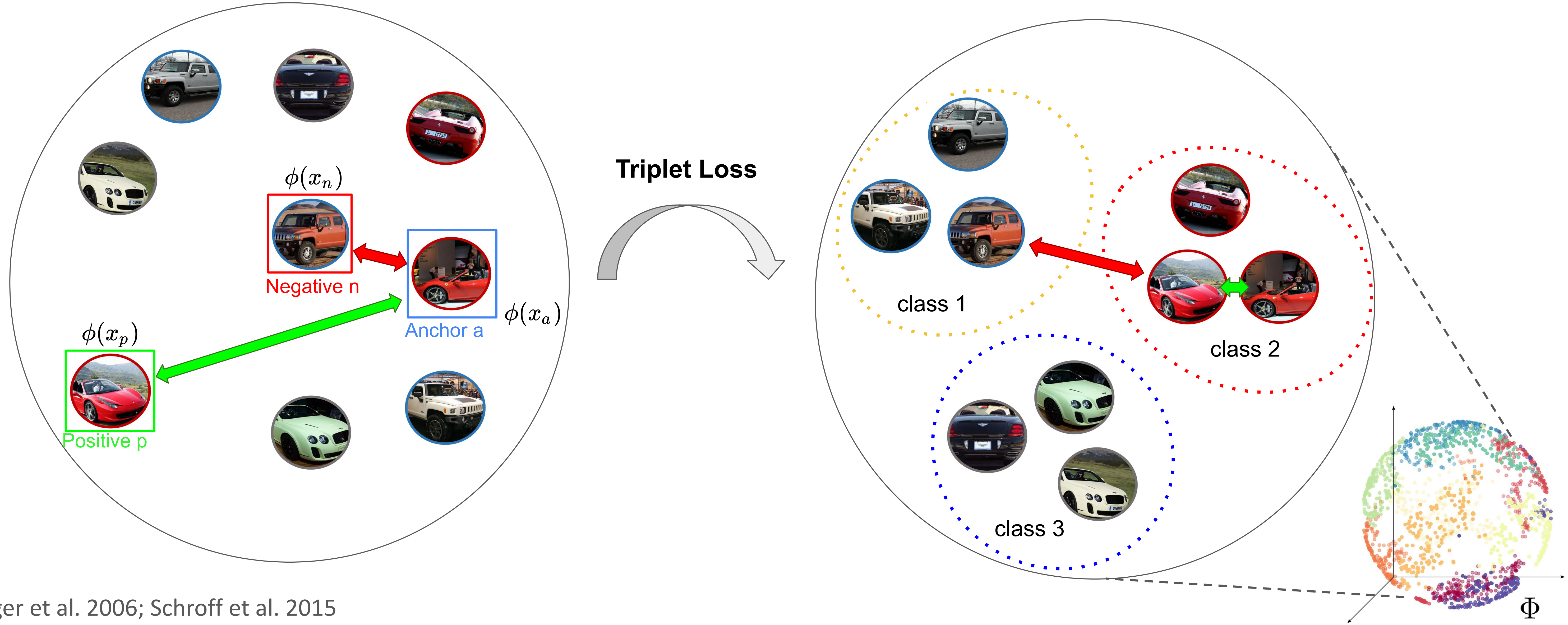


Ranking-based DML

- Learning informative embedding space requires to **“arrange”** data under $\phi(x)$
- Naturally formulate the learning problem as orderings **“A is closer to B then C”**

Triplet loss[1]: $t = \{x_a, x_p, x_n\}$

$$\ell^{\text{triplet}}(a,p,n) = ||\phi(x_a) - \phi(x_p)||_2^2 - ||\phi(x_a) - \phi(x_n)||_2^2$$



[1] Weinberger et al. 2006; Schroff et al. 2015

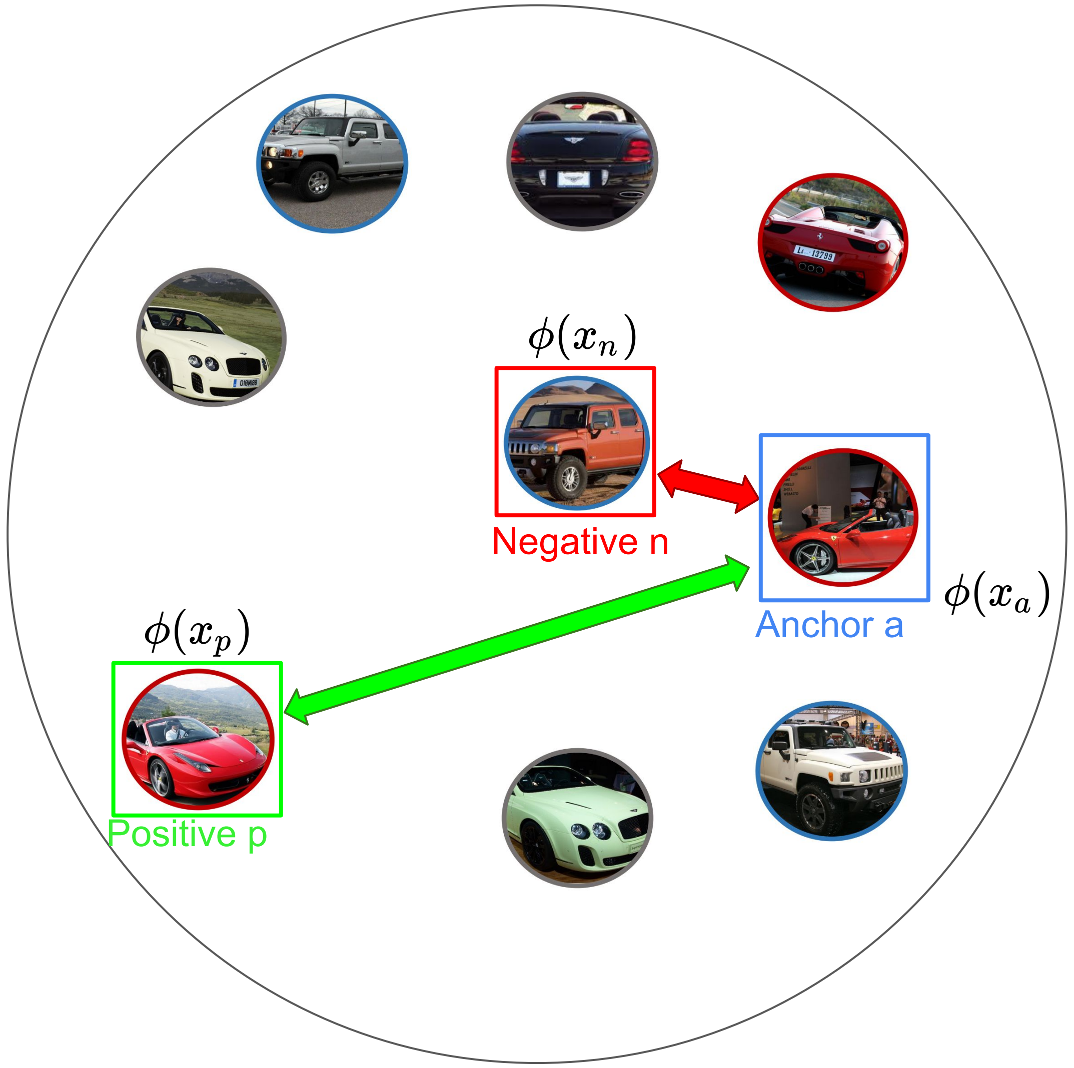
Ranking-based DML

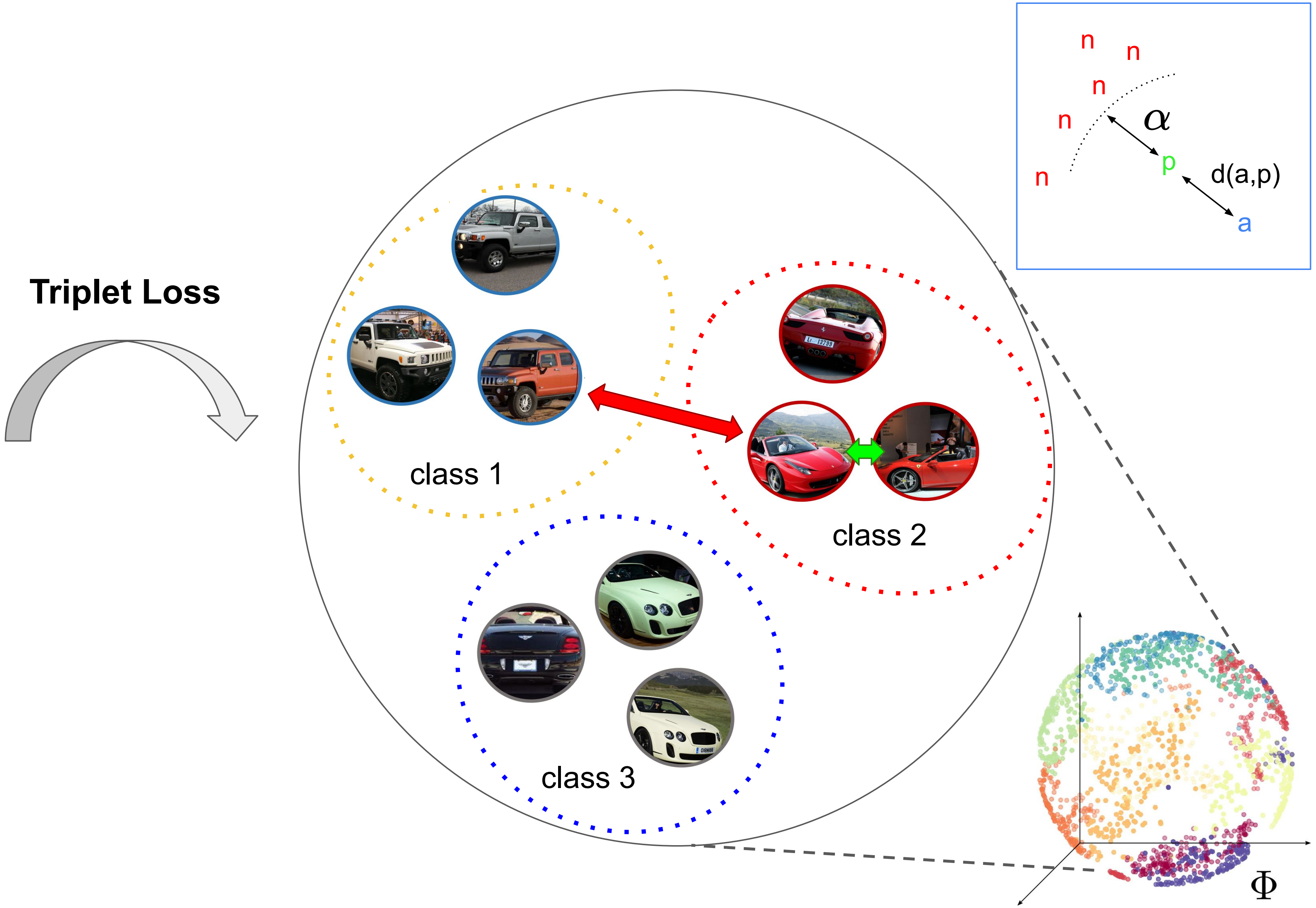
- Learning informative embedding space requires to **“arrange”** data under $\phi(x)$
- Naturally formulate the learning problem as orderings **“A is closer to B then C”**

Triplet loss[1]**:** $t = \{x_a, x_p, x_n\}$

$$\ell^{\text{triplet}}(a,p,n) = \max(||\phi(x_a) - \phi(x_p)||_2^2 - ||\phi(x_a) - \phi(x_n)||_2^2 + \alpha, 0)$$

fixed margin





[1] Weinberger et al. 2006; Schroff et al. 2015

Ranking-based DML

- Learning informative embedding space requires to **“arrange”** data under $\phi(x)$
- Naturally formulate the learning problem as orderings **“A is closer to B then C”**

Margin loss[2]:

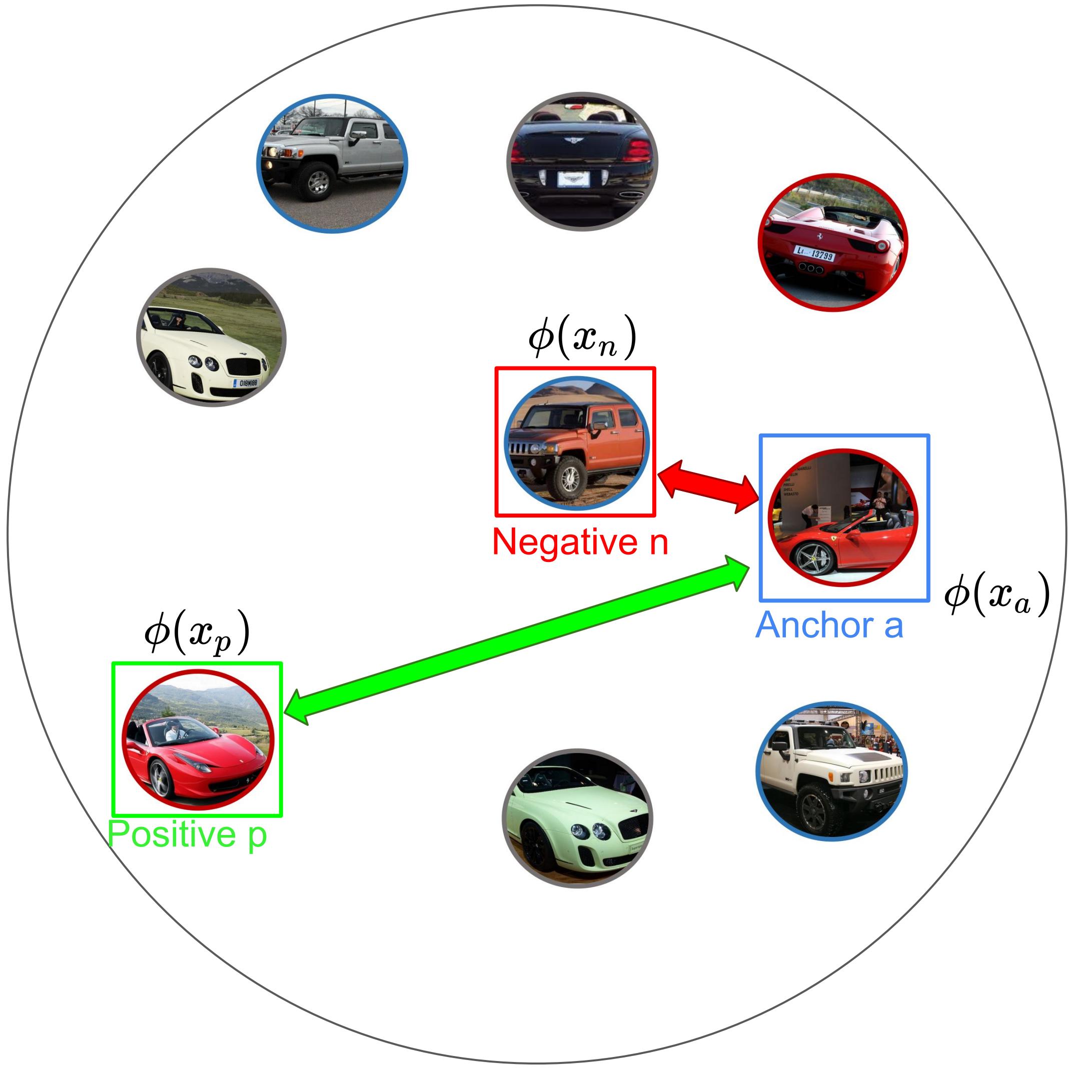
$$\ell^{\text{triplet}}(a,p,n) := \left[D_{ap}^2 - D_{an}^2 + \alpha\right]_+$$

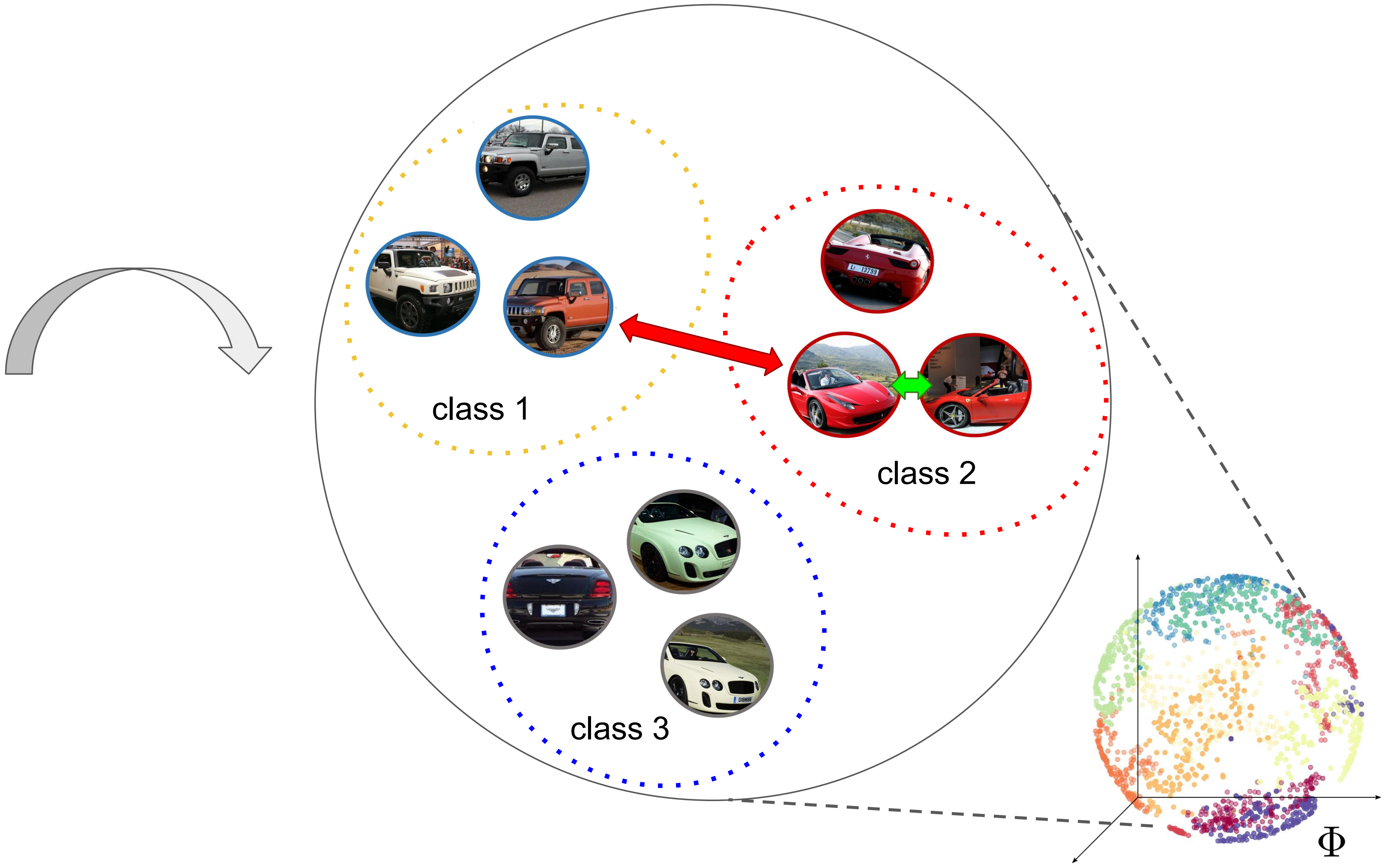
$$\ell^{\text{margin}}(i,j) := (\alpha + y_{ij}(D_{ij} - \beta))_+ \, .$$

fixed margin

learned margin parameter

$$\boxed{\beta(i) := \beta^{(0)} + \beta^{(\text{class})}_{c(i)} + \beta^{(\text{img})}_i}$$





[2] Wu et al. 2017

Ranking-based DML

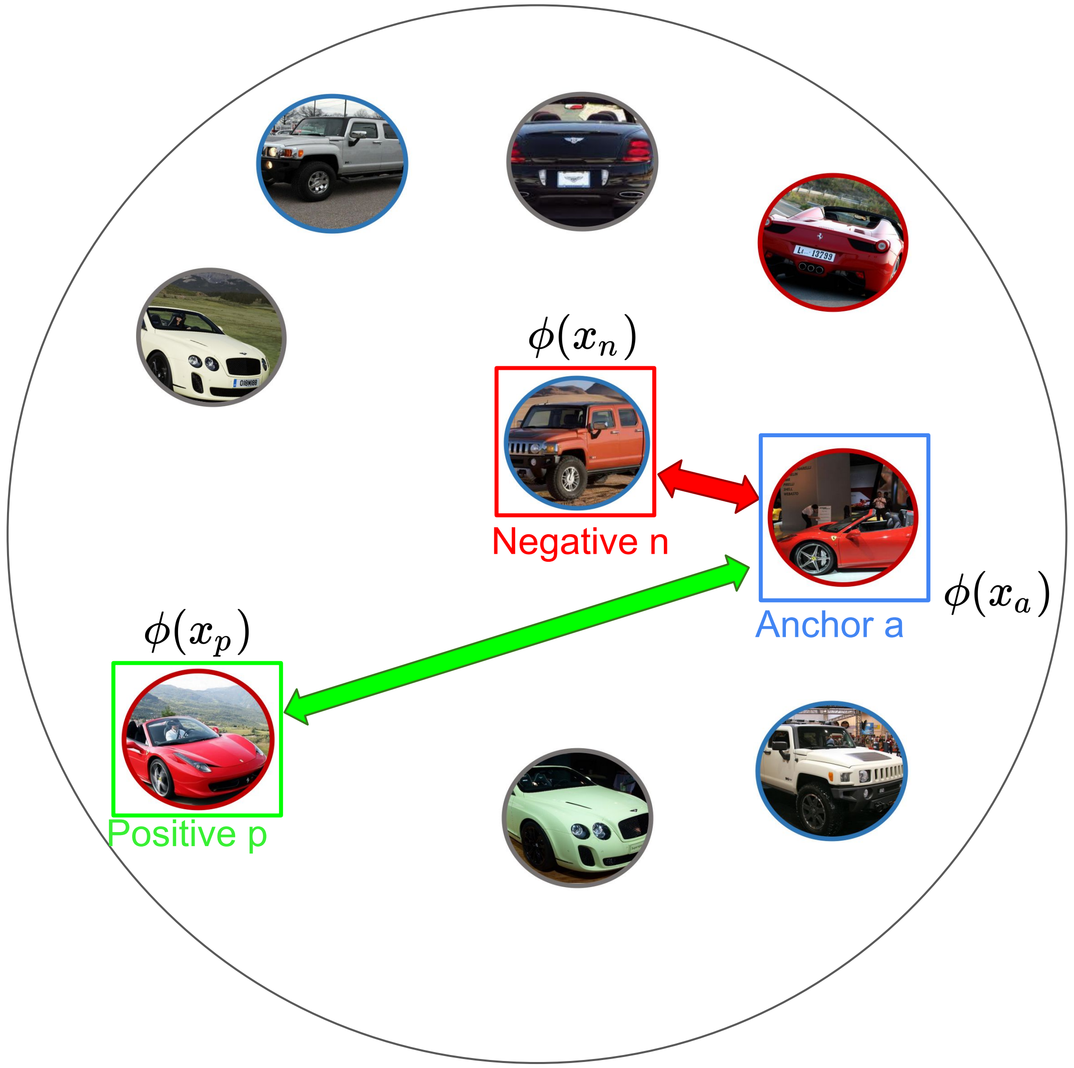
- Learning informative embedding space requires to **"arrange"** data under $\phi(x)$
- Naturally formulate the learning problem as orderings **"A is closer to B then C"**

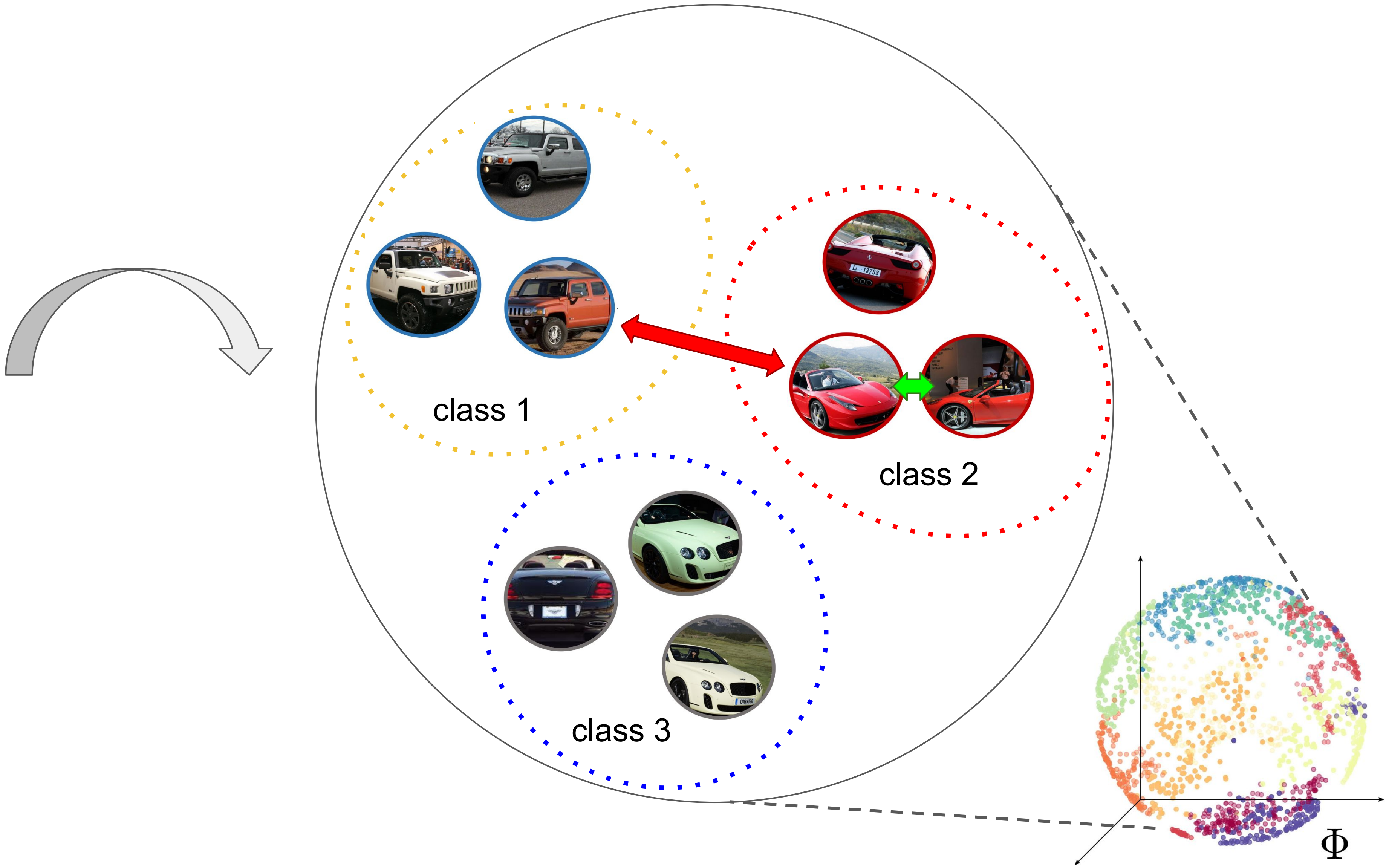
Margin loss[2] (Joint optimization):

$$\text{minimize} \sum_{(i,j)} \ell^{\text{margin}}(i,j) + \nu \left(\beta^{(0)} + \beta^{(\text{class})}_{c(i)} + \beta^{(\text{img})}_i \right)$$

$$\ell^{\text{margin}}(i,j) := (\alpha + y_{ij}(D_{ij} - \beta))_+$$

$$\beta(i) := \beta^{(0)} + \beta^{(\text{class})}_{c(i)} + \beta^{(\text{img})}_i$$





[2] Wu et al. 2017

Ranking-based DML

- Learning informative embedding space requires to **"arrange"** data under $\phi(x)$
- Naturally formulate the learning problem as orderings **"A is closer to B then C"**

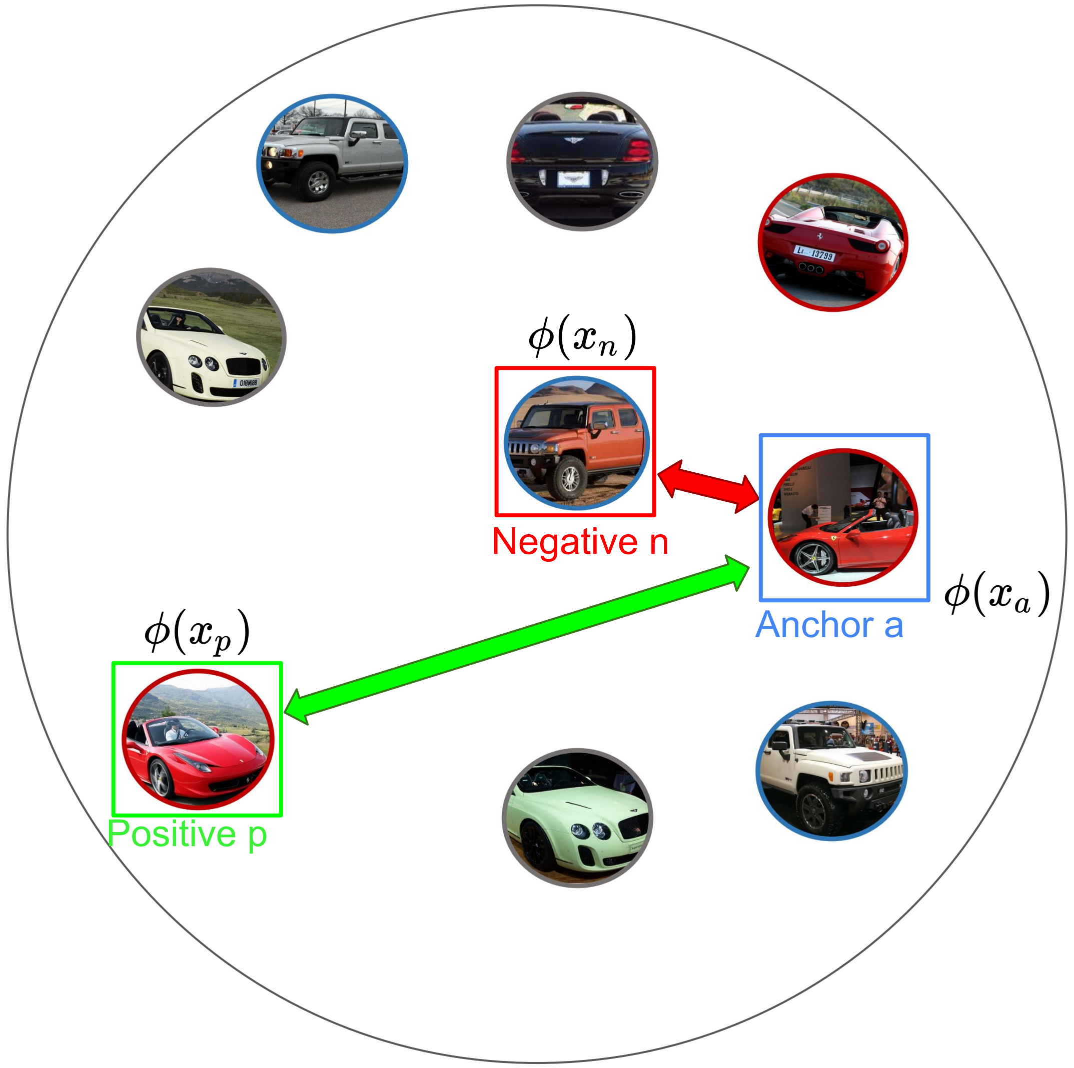
Margin loss[2] (Joint optimization):

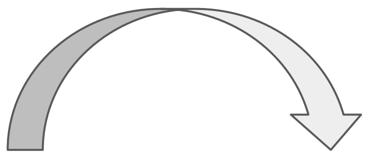
$$\text{minimize} \sum_{(i,j)} \ell^{\text{margin}}(i,j) + \nu \left(\beta^{(0)} + \beta^{(\text{class})}_{c(i)} + \beta^{(\text{img})}_{i} \right)$$

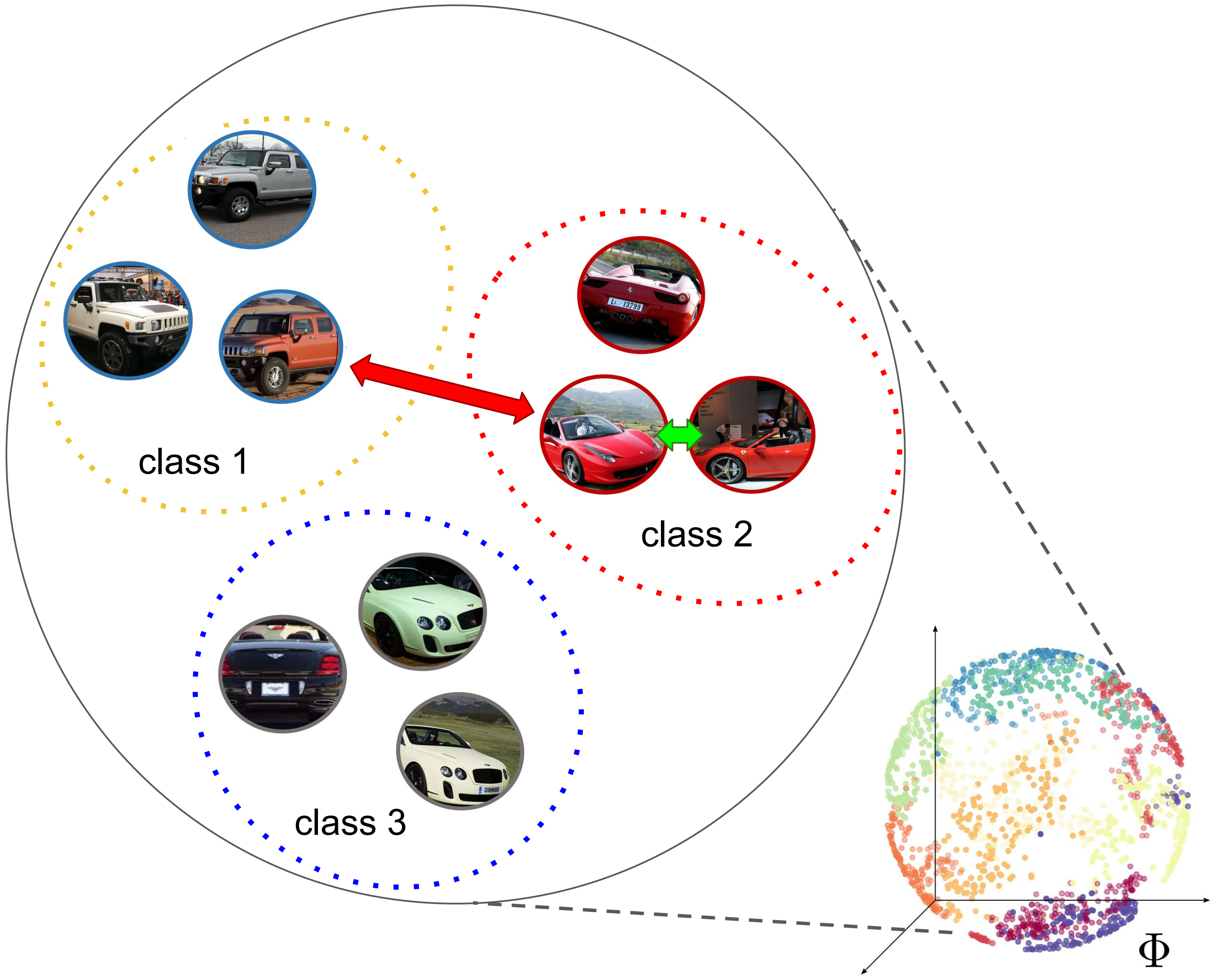
$$\ell^{\text{margin}}(i,j) := (\alpha + y_{ij}(D_{ij} - \beta))_+$$

$$\beta(i) := \beta^{(0)} + \beta^{(\text{class})}_{c(i)} + \cancel{\beta^{(\text{img})}_{i}}$$

in practice



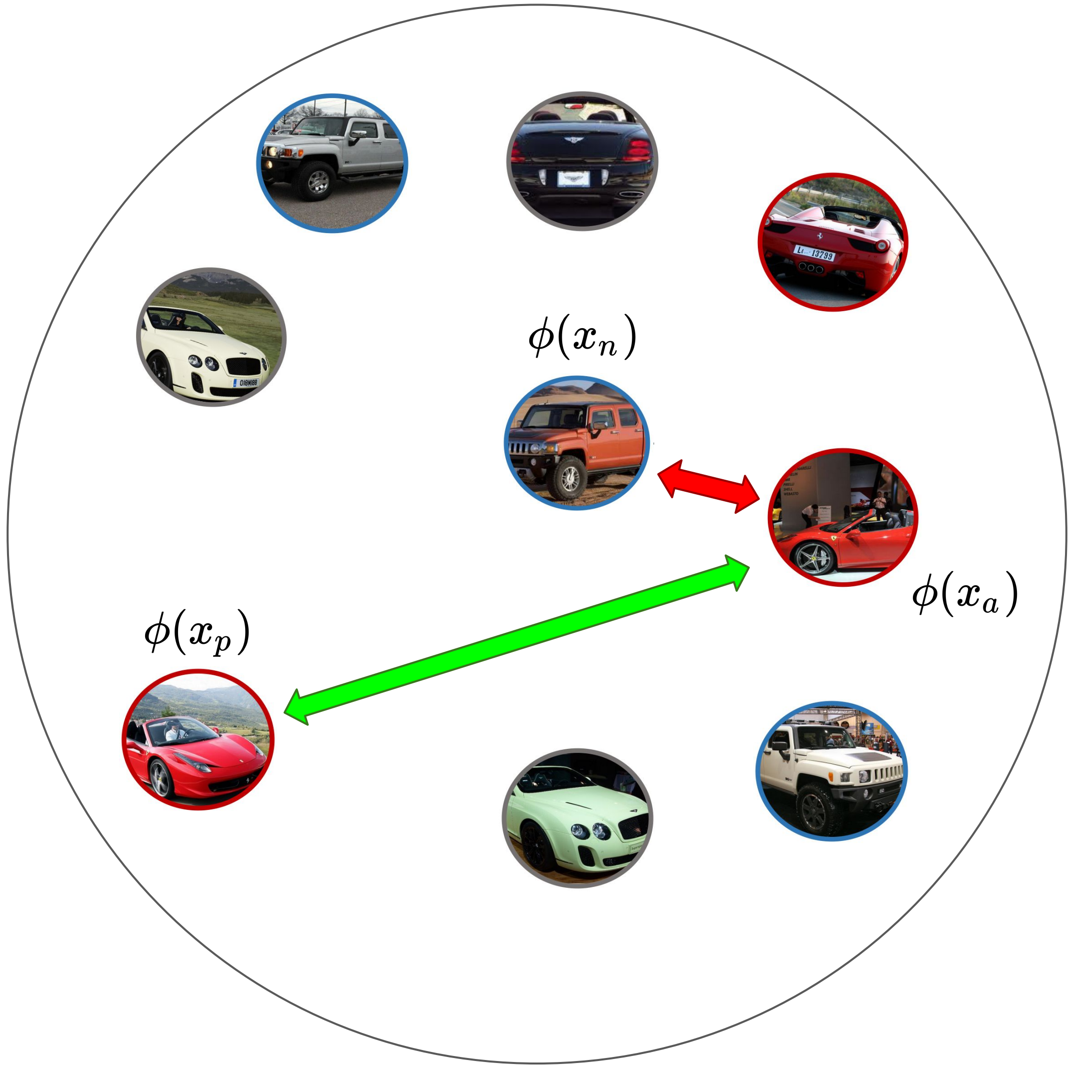


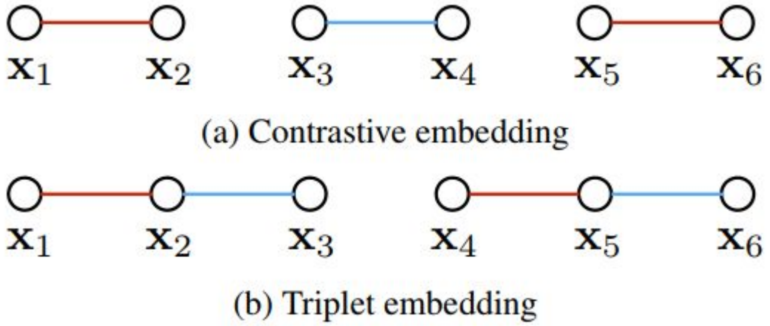


[2] Wu et al. 2017

Ranking-based DML

- So far: Per sample in batch: **form a single triplet**



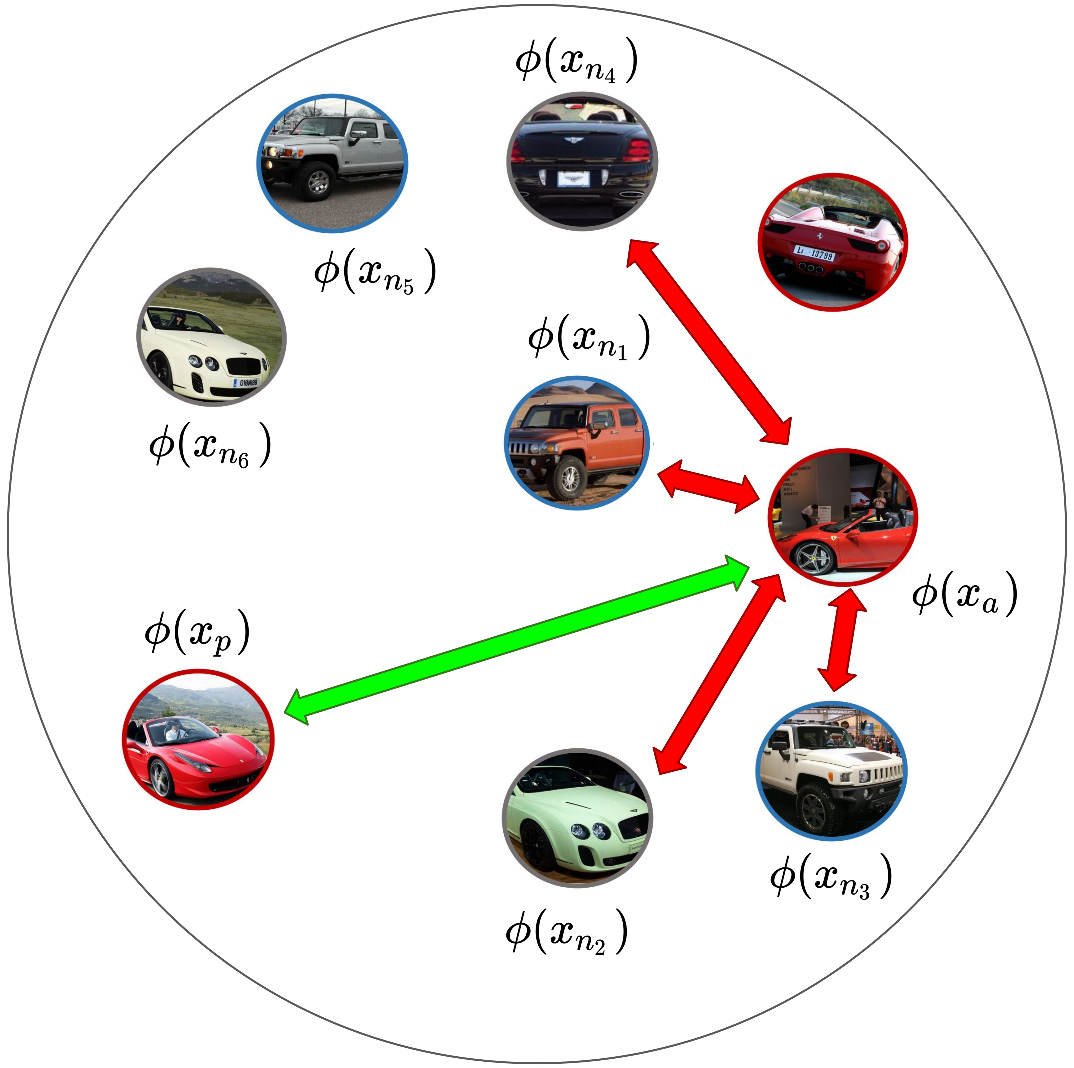


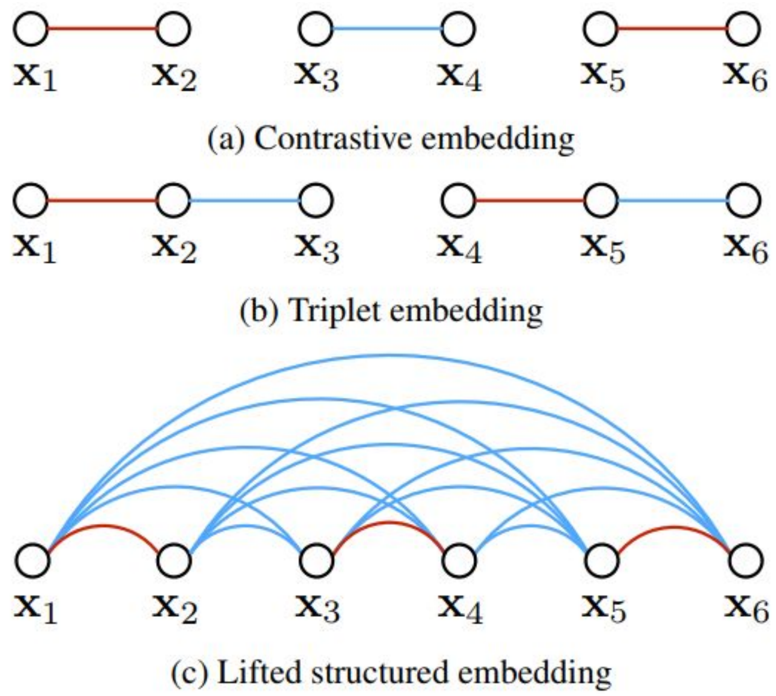
$\mathcal{O}(B)$ relations

Images partly taken from: Song et al. 2015

Ranking-based DML

- So far: Per sample in batch: **form a single triplet**
- Use **all possible relations** within batch for learning!





$\mathcal{O}(B)$ relations

$\mathcal{O}(B^2)$ relations

Images partly taken from: Song et al. 2015

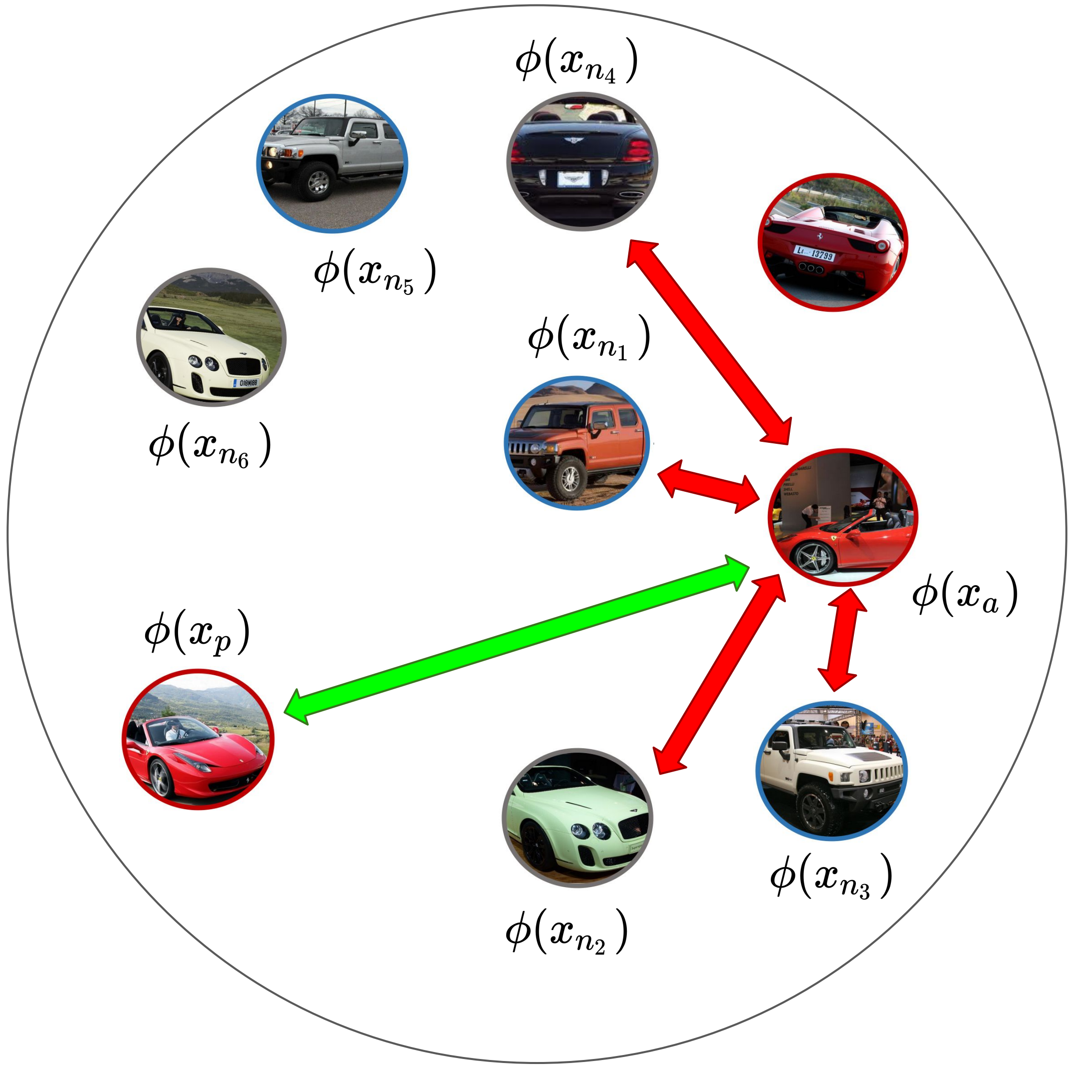
Ranking-based DML

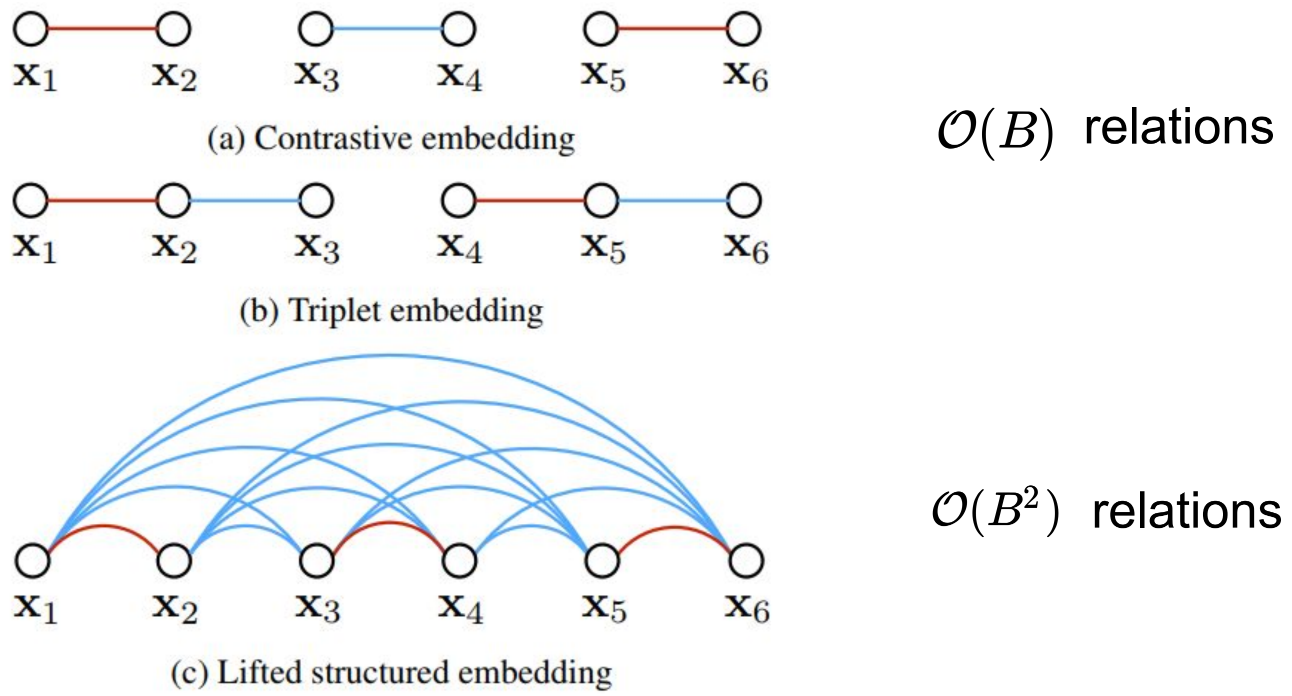
- So far: Per sample in batch: **form a single triplet**
- Use **all possible relations** within batch for learning!

Lifted Structures Feature Embedding Loss[3] :

$$\tilde{J}_{i,j} = \log\left(\sum_{(i,k)\in\mathcal{N}} \exp\{\alpha - D_{i,k}\} + \sum_{(j,l)\in\mathcal{N}} \exp\{\alpha - D_{j,l}\}\right) + D_{i,j}$$

$$\tilde{J} = \frac{1}{2|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \max\left(0,\ \tilde{J}_{i,j}\right)^2,$$





[3] Song et al. 2015

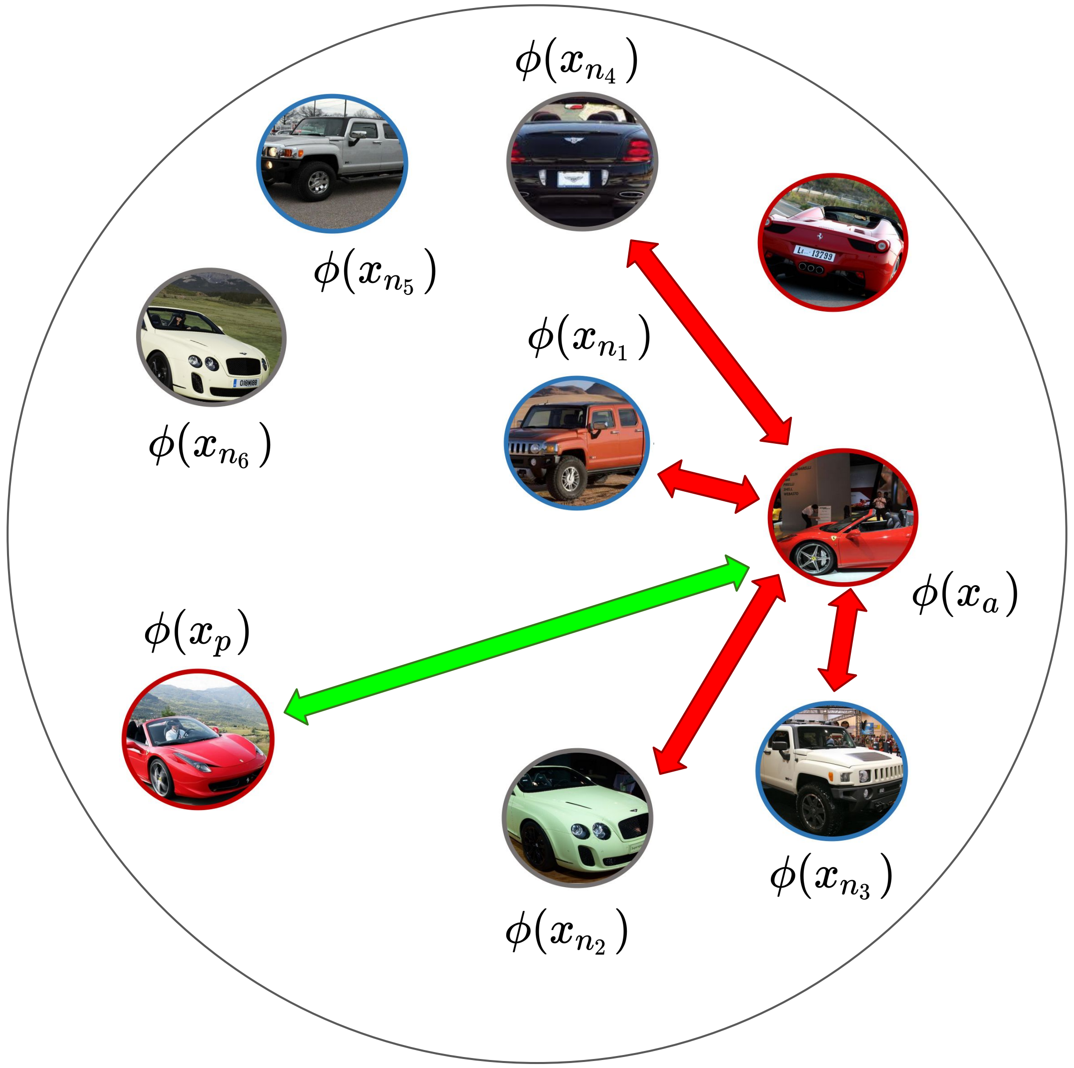
Ranking-based DML

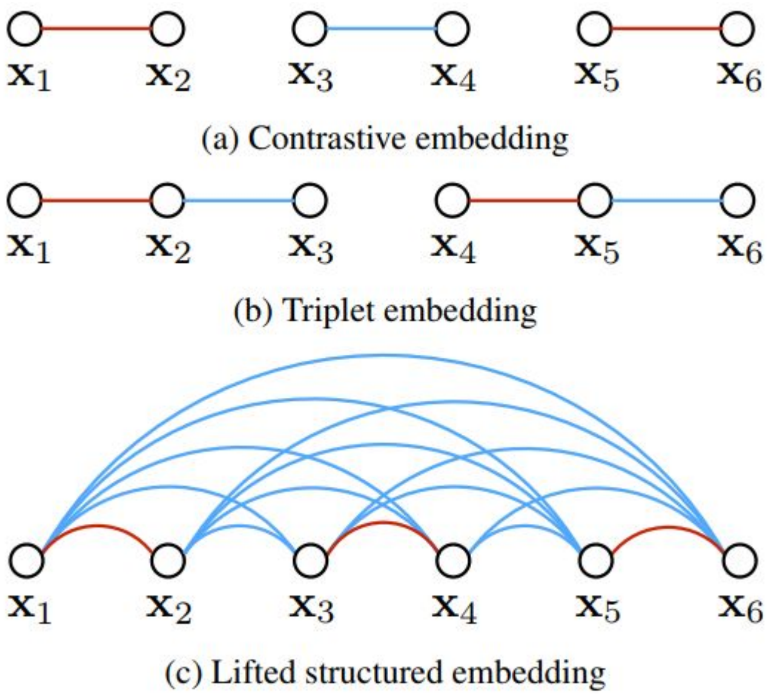
- So far: Per sample in batch: **form a single triplet**
- Use **all possible relations** within batch for learning!

Lifted Structures Feature Embedding Loss[3] :

$$\tilde{J}_{i,j} = \log\left(\overbrace{\boxed{\sum_{(i,k)\in\mathcal{N}} \exp\{\alpha - D_{i,k}\}}}^{\text{anchor to all negatives in batch}} + \overbrace{\boxed{\sum_{(j,l)\in\mathcal{N}} \exp\{\alpha - D_{j,l}\}}}^{\text{positive to all negatives in batch}}\right) + \underbrace{\boxed{D_{i,j}}}_{\text{anchor to positive}}$$

$$\tilde{J} = \frac{1}{2|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \max\left(0,\ \tilde{J}_{i,j}\right)^2,$$





(a) Contrastive embedding — $\mathcal{O}(B)$ relations

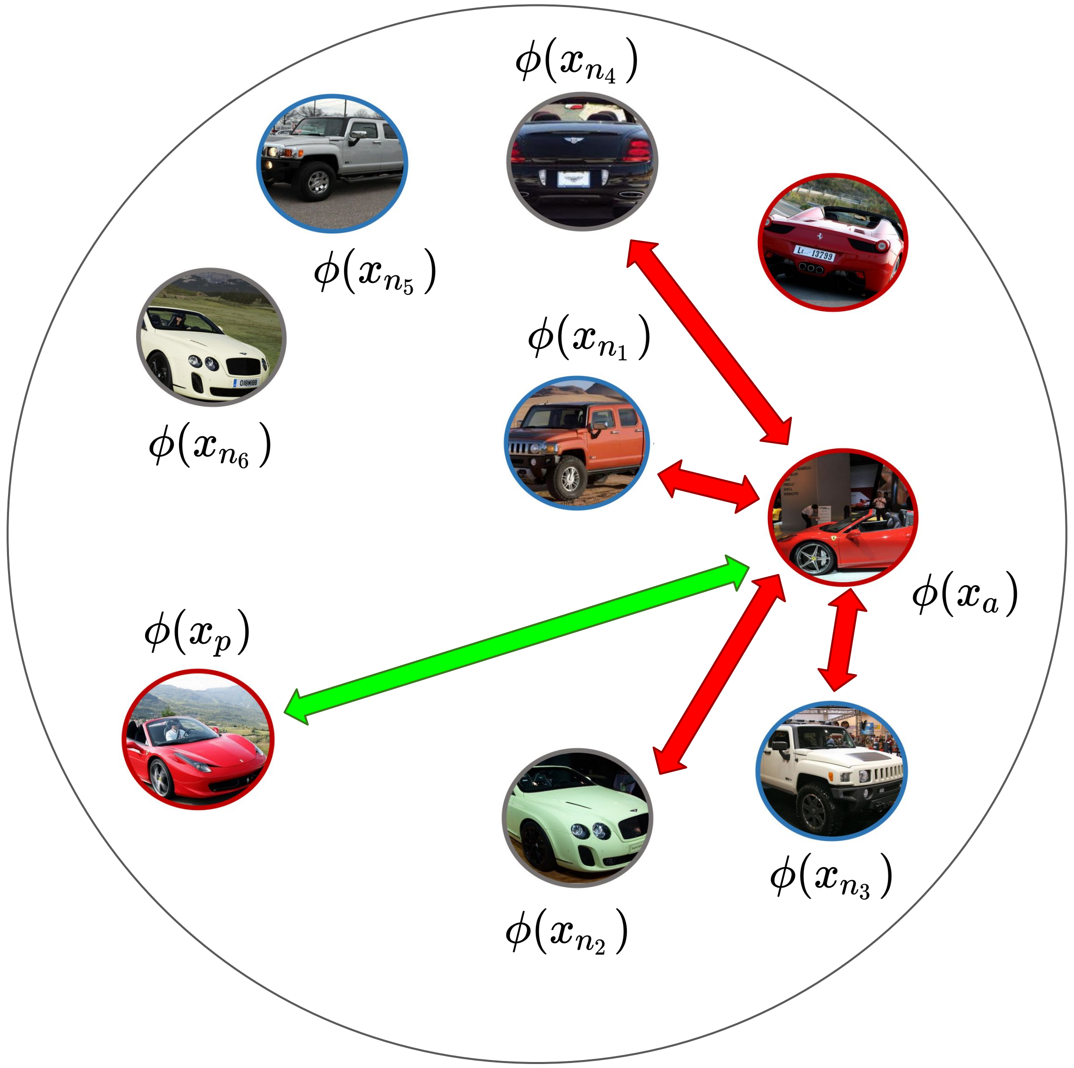
(b) Triplet embedding

(c) Lifted structured embedding — $\mathcal{O}(B^2)$ relations

[3] Song et al. 2015

Ranking-based DML

- So far: Per sample in batch: **form a single triplet**
- Use **all possible relations** within batch for learning!

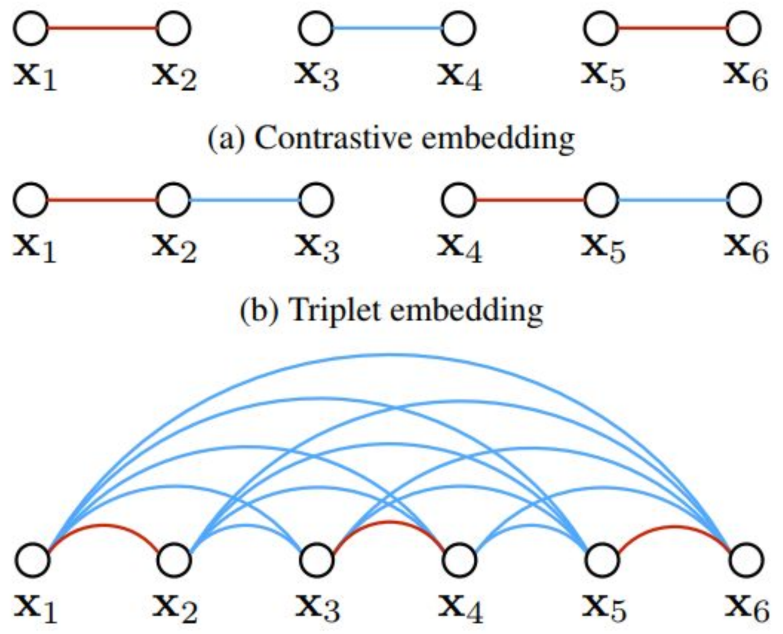


Multi-Similarity Loss[4]:

anchor to positives in batch

$$\mathcal{L}_{MS} = \frac{1}{m}\sum_{i=1}^{m}\left\{\frac{1}{\alpha}\log\left[1+\boxed{\sum_{k\in\mathcal{P}_i} e^{-\alpha(S_{ik}-\lambda)}}\right] + \frac{1}{\beta}\log\left[1+\boxed{\sum_{k\in\mathcal{N}_i} e^{\beta(S_{ik}-\lambda)}}\right]\right\}.$$

anchor to negatives in batch



(a) Contrastive embedding

$\mathcal{O}(B)$ relations

(b) Triplet embedding

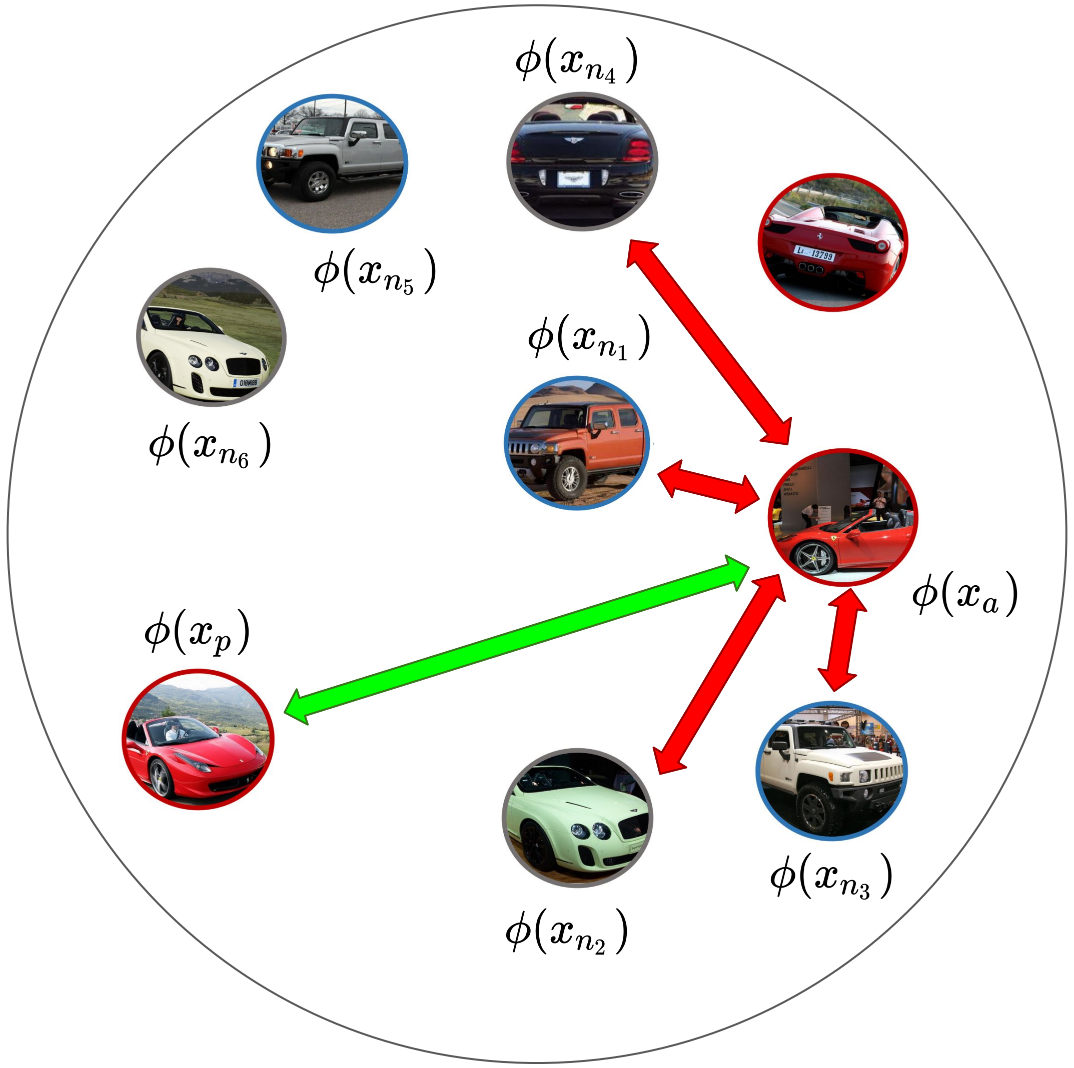
(c) Lifted structured embedding

$\mathcal{O}(B^2)$ relations

[4] Wang et al. 2019

Ranking-based DML

- So far: Per sample in batch: **form a single triplet**
- Use **all possible relations** within batch for learning!



Multi-Similarity Loss[4]:

anchor to **positives in batch**

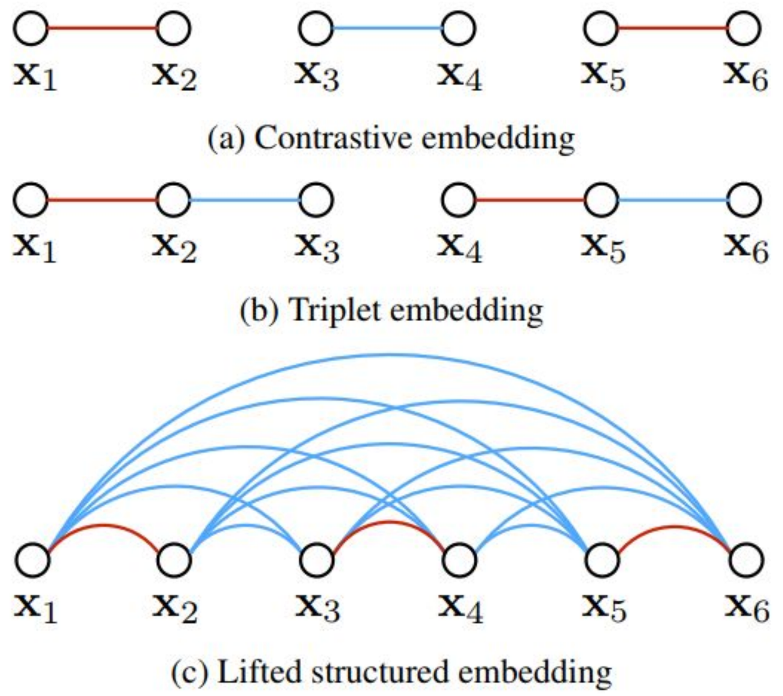
$$\mathcal{L}_{MS} = \frac{1}{m}\sum_{i=1}^{m}\left\{\frac{1}{\alpha}\log\left[1+\boxed{\sum_{k\in\mathcal{P}_i} e^{-\alpha(S_{ik}-\lambda)}}\right] + \frac{1}{\beta}\log\left[1+\boxed{\sum_{k\in\mathcal{N}_i} e^{\beta(S_{ik}-\lambda)}}\right]\right\}.$$

anchor to **negatives** in batch

Informative pair mining:

$$S_{ij}^{+} < \max_{y_k \neq y_i} S_{ik} + \epsilon.$$ "positive less similar to anchor than best negative"

$$S_{ij}^{-} > \min_{y_k = y_i} S_{ik} - \epsilon,$$ "negative more similar to anchor than worst positive"



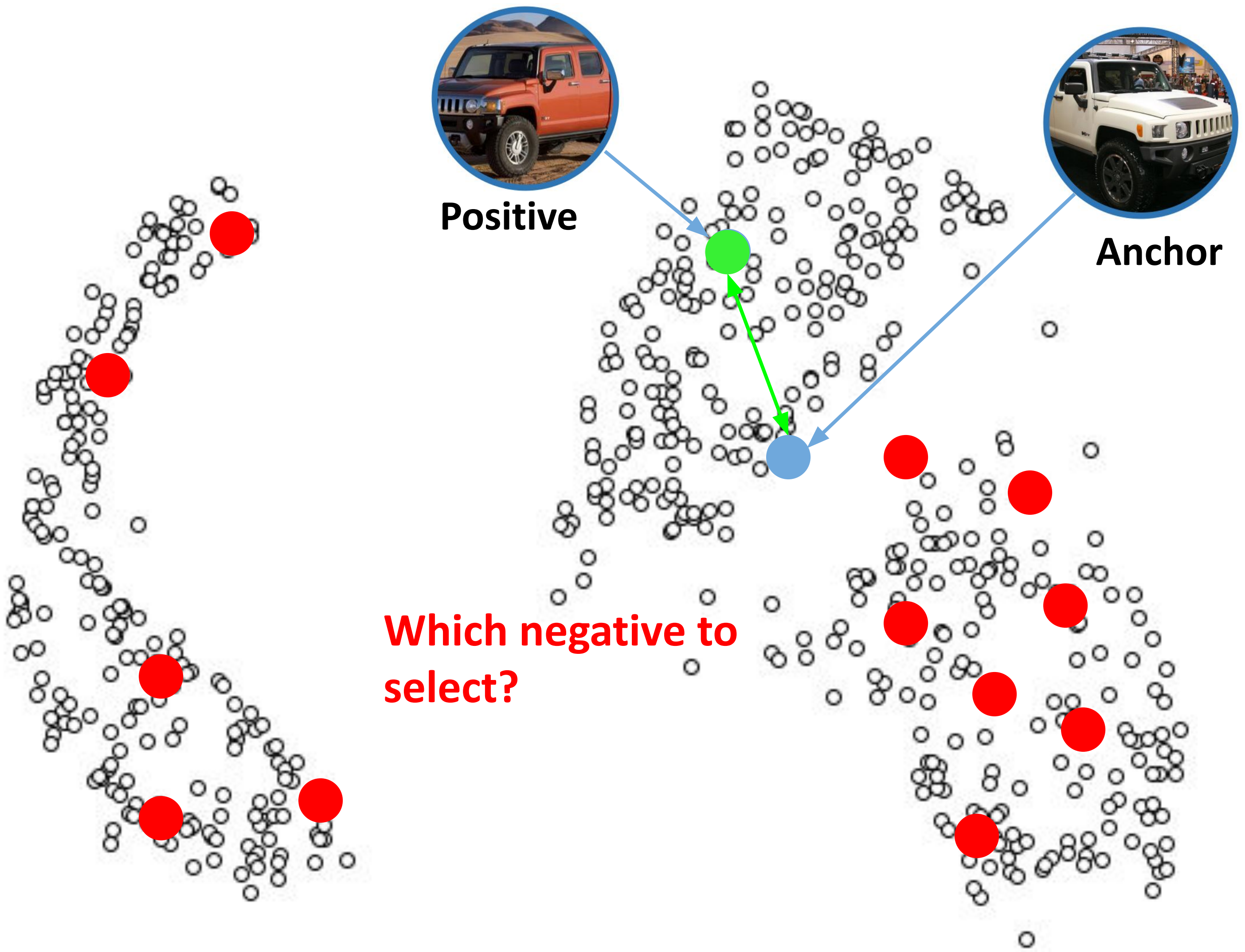
$\mathcal{O}(B)$ relations

$\mathcal{O}(B^2)$ relations

[4] Wang et al. 2019

Negative Sampling in Ranking-based DML

- **Infeasible** amount of triplets to learn from (Scale $\mathcal{O}(n^3)$)

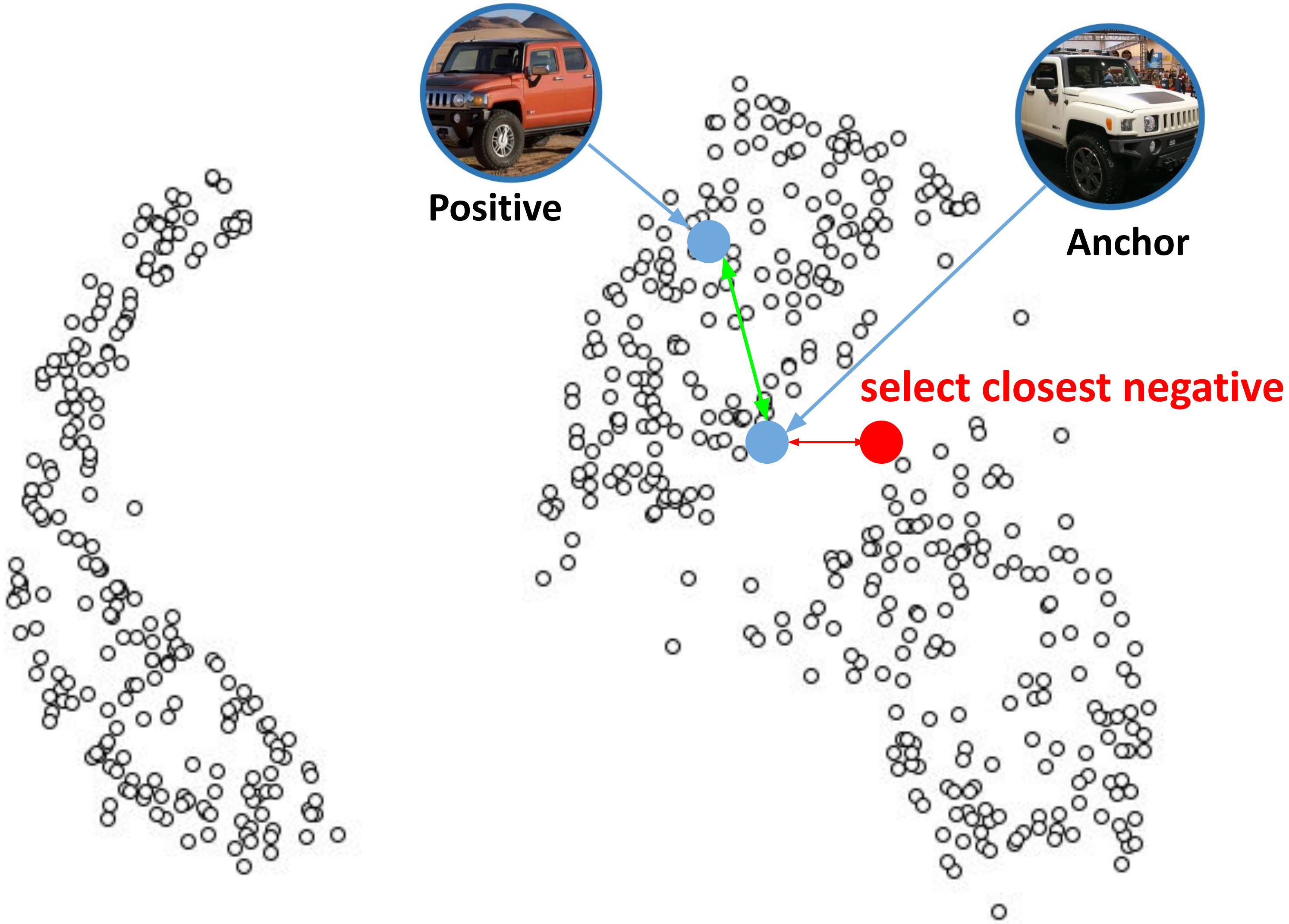


Negative Sampling in Ranking-based DML

- **Infeasible** amount of triplets to learn from (Scale $\mathcal{O}(n^3)$)
- Fixed sampling heuristics:

- **(Semi-)Hard negative mining** [Schroff et al. 2015]

$$n^* = \arg \min_{n: d_{\text{an}} > d_{\text{ap}}} d_{\text{an}}$$



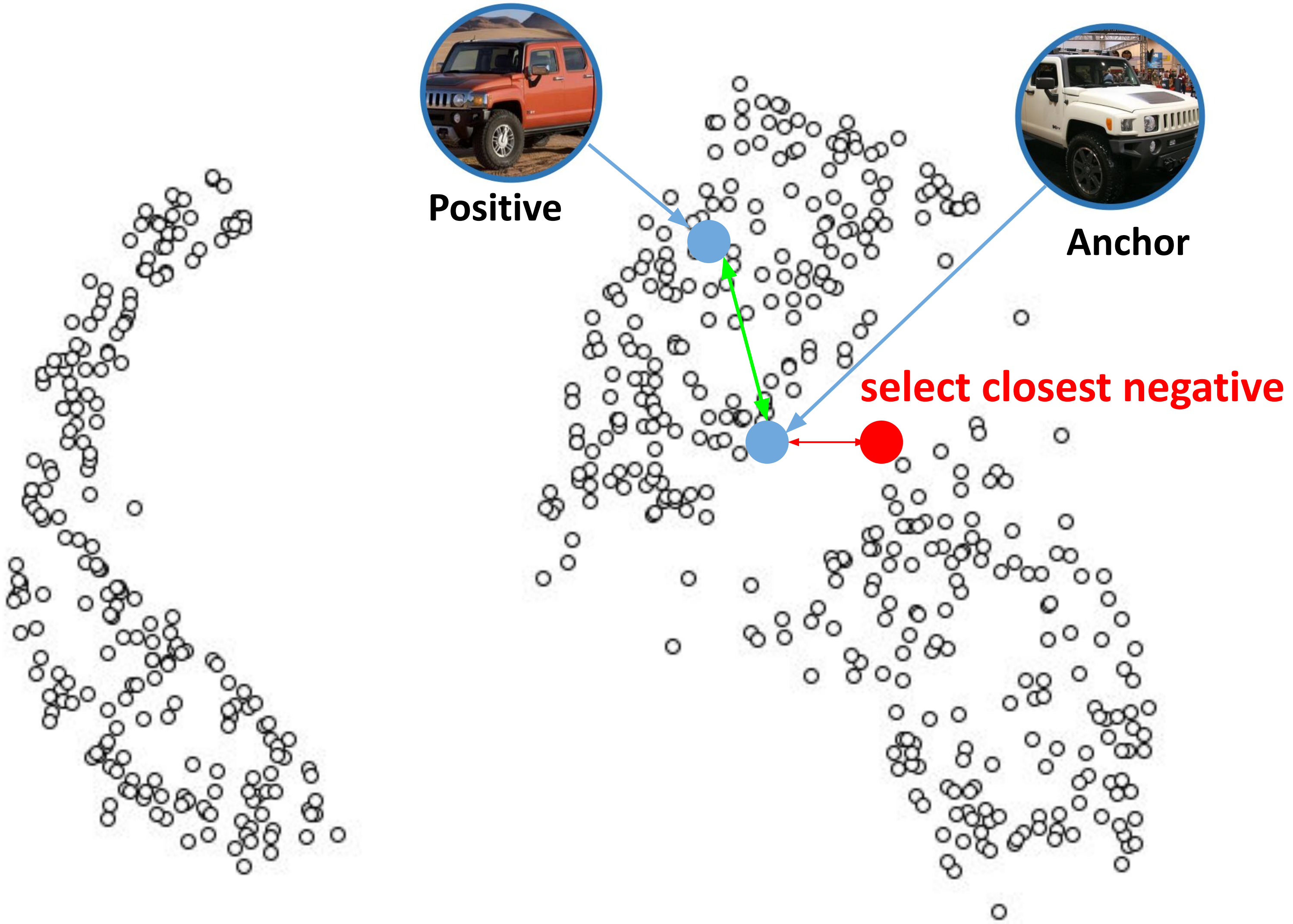
Negative Sampling in Ranking-based DML

- **Infeasible** amount of triplets to learn from (Scale $\mathcal{O}(n^3)$)
- Fixed sampling heuristics:

- **(Semi-)Hard negative mining** [Schroff et al. 2015]

$$n^* = \arg \min_{n: d_{\text{an}} > d_{\text{ap}}} d_{\text{an}}$$

- May suffer from **unstable gradients** due to focus on hardest negatives only

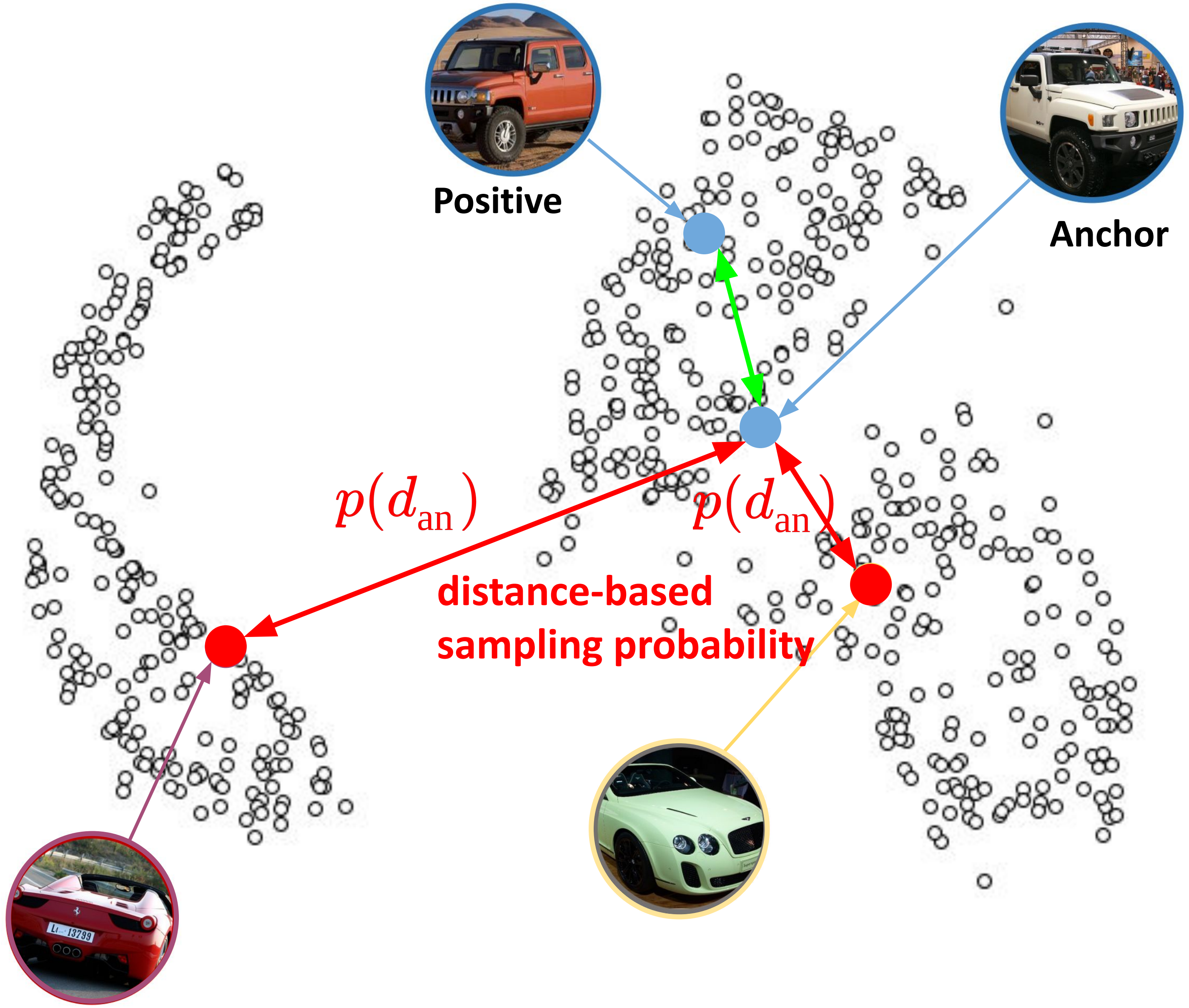


Negative Sampling in Ranking-based DML

- **Infeasible** amount of triplets to learn from (Scale $\mathcal{O}(n^3)$)
- Fixed sampling heuristics:

- **Distance-weighted sampling** [Wu et al. 2017]

$$n^* \sim p(I_n|I_a) = p(d_{\text{an}})$$



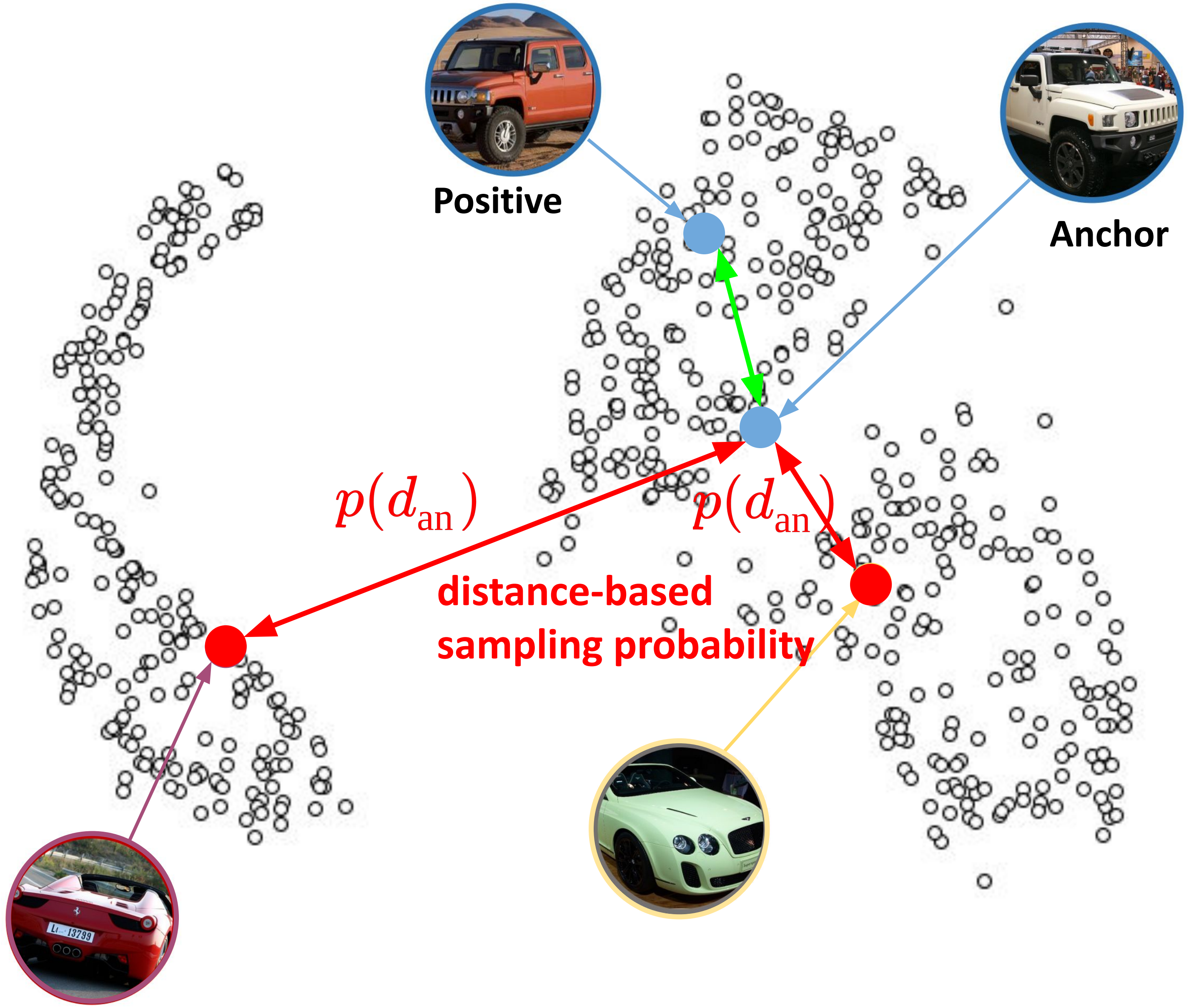
Negative Sampling in Ranking-based DML

- **Infeasible** amount of triplets to learn from (Scale $\mathcal{O}(n^3)$)
- Fixed sampling heuristics:

- **Distance-weighted sampling** [Wu et al. 2017]

$$n^* \sim p(I_n|I_a) = p(d_{\text{an}})$$

- Soften hard negative constraint by uniform sampling from **entire range of distances** d_{an}



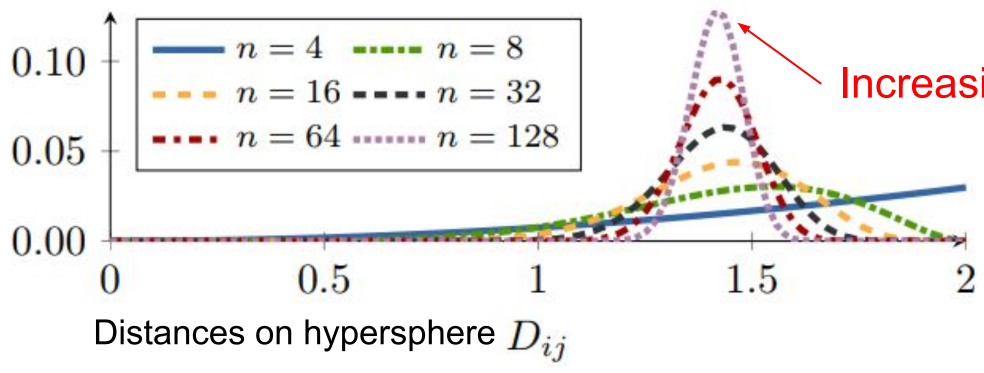
Negative Sampling in Ranking-based DML

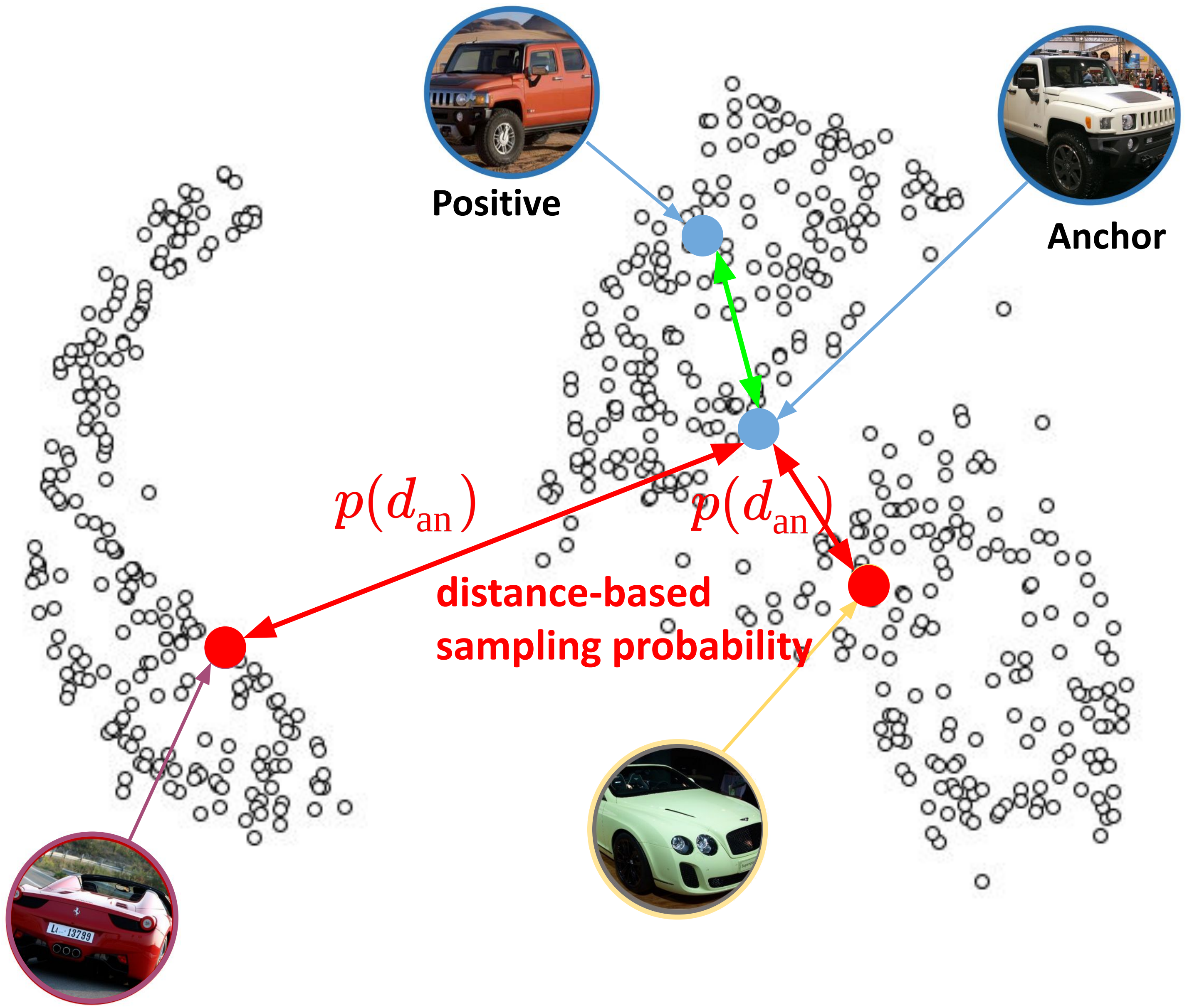
- **Infeasible** amount of triplets to learn from (Scale $\mathcal{O}(n^3)$)
- Fixed sampling heuristics:

- **Distance-weighted sampling** [Wu et al. 2017]

$$n^* \sim p(I_n|I_a) = p(d_{\text{an}})$$

- Soften hard negative constraint by uniform sampling from **entire range of distances** d_{an}





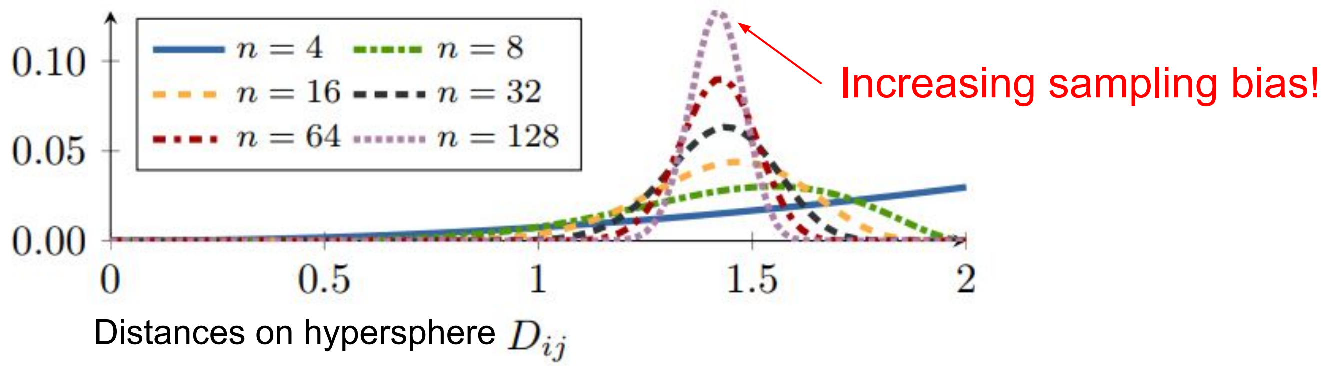
Negative Sampling in Ranking-based DML

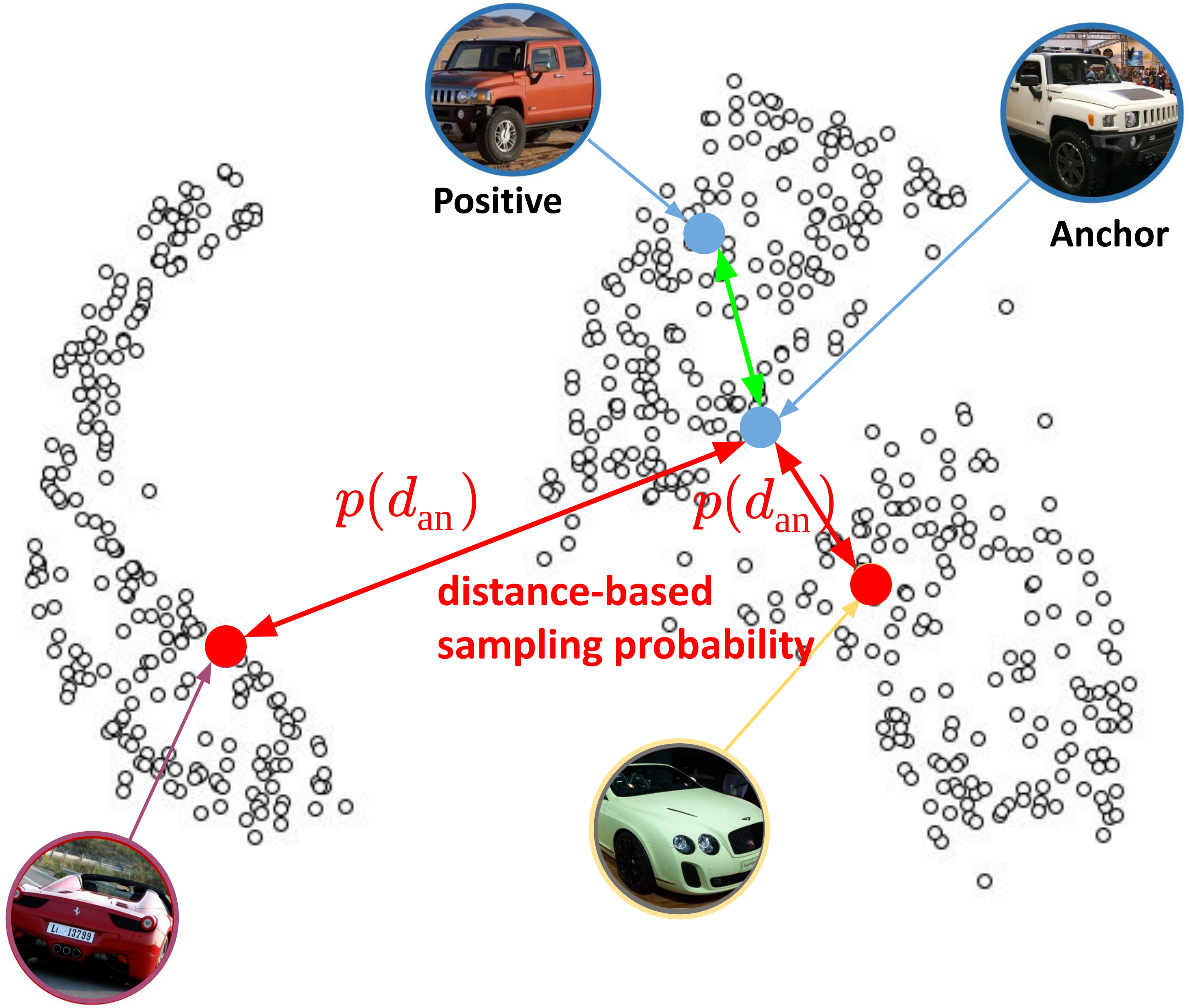
- **Infeasible** amount of triplets to learn from (Scale $\mathcal{O}(n^3)$)
- Fixed sampling heuristics:

- **Distance-weighted sampling** [Wu et al. 2017]

$$n^* \sim p(I_n|I_a) \propto \min(\lambda, q(d_{an})^{-1})$$

- Soften hard negative constraint by uniform sampling from **entire range of distances** d_{an}
- Uniform distribution on $\mathbb{S}^D$: $q(d) \propto d^{k-2}[1 - \frac{1}{4}d^2]^{\frac{k-3}{2}}$





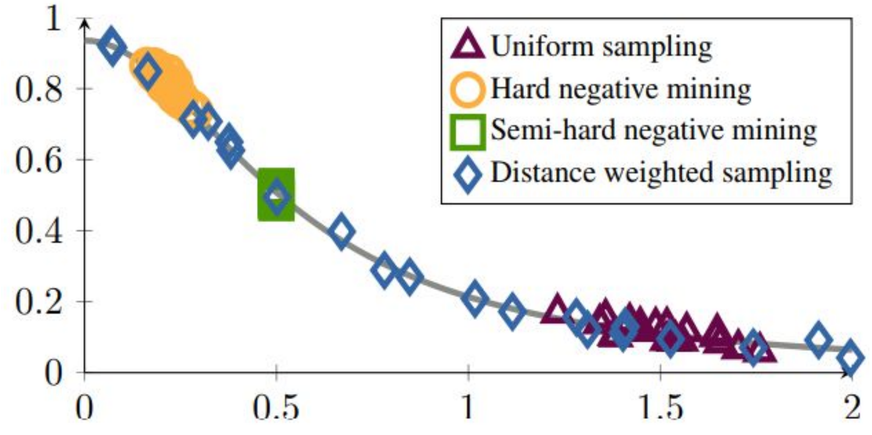
Negative Sampling in Ranking-based DML

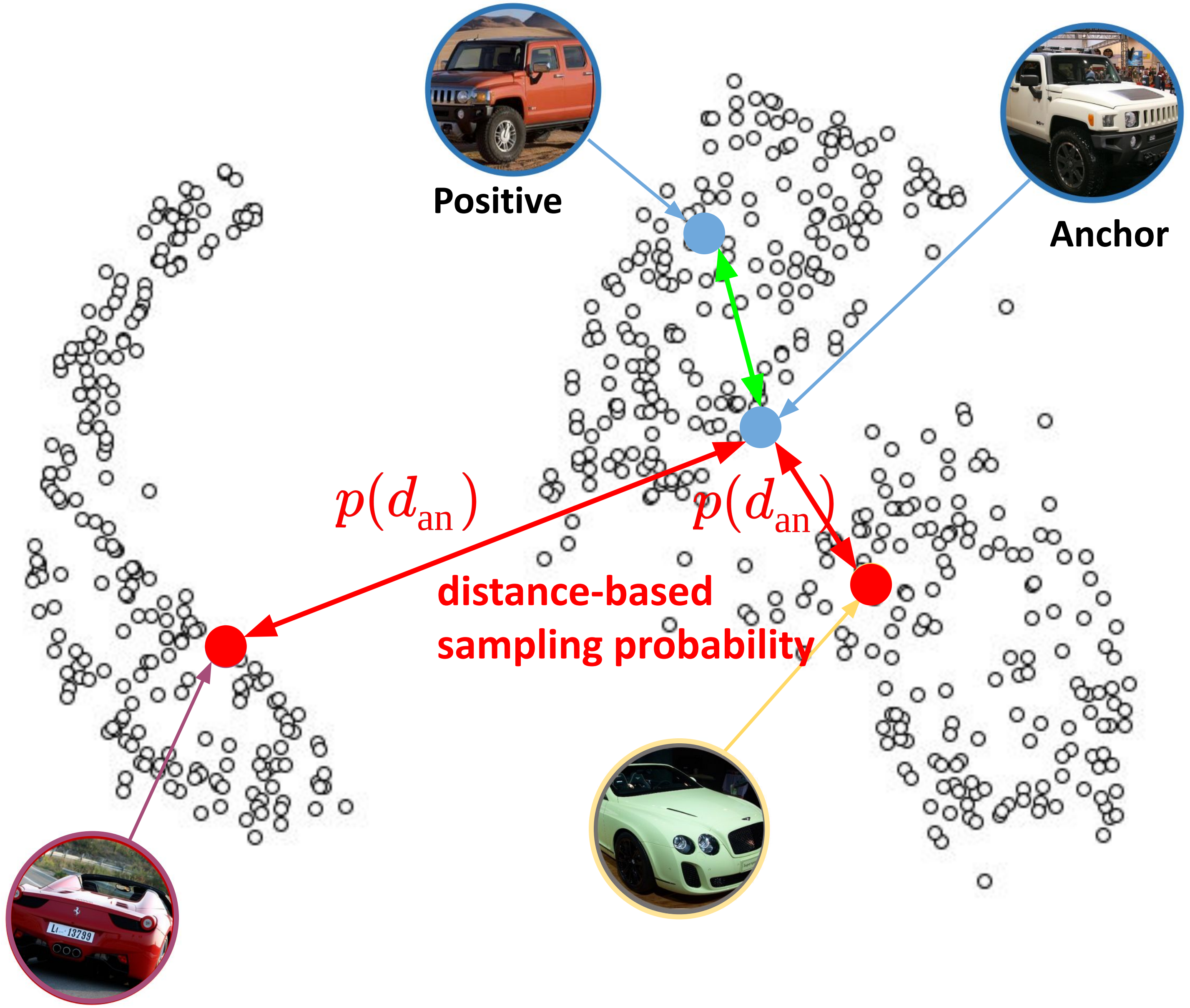
- **Infeasible** amount of triplets to learn from (Scale $\mathcal{O}(n^3)$)
- Fixed sampling heuristics:

- **Distance-weighted sampling** [Wu et al. 2017]

$$n^* \sim p(I_n|I_a) \propto \min(\lambda, q(d_{\text{an}})^{-1})$$

- Soften hard negative constraint by uniform sampling from **entire range of distances** d_{an}
- Uniform distribution on $\mathbb{S}^D$: $q(d) \propto d^{k-2}[1-\frac{1}{4}d^2]^{\frac{k-3}{2}}$





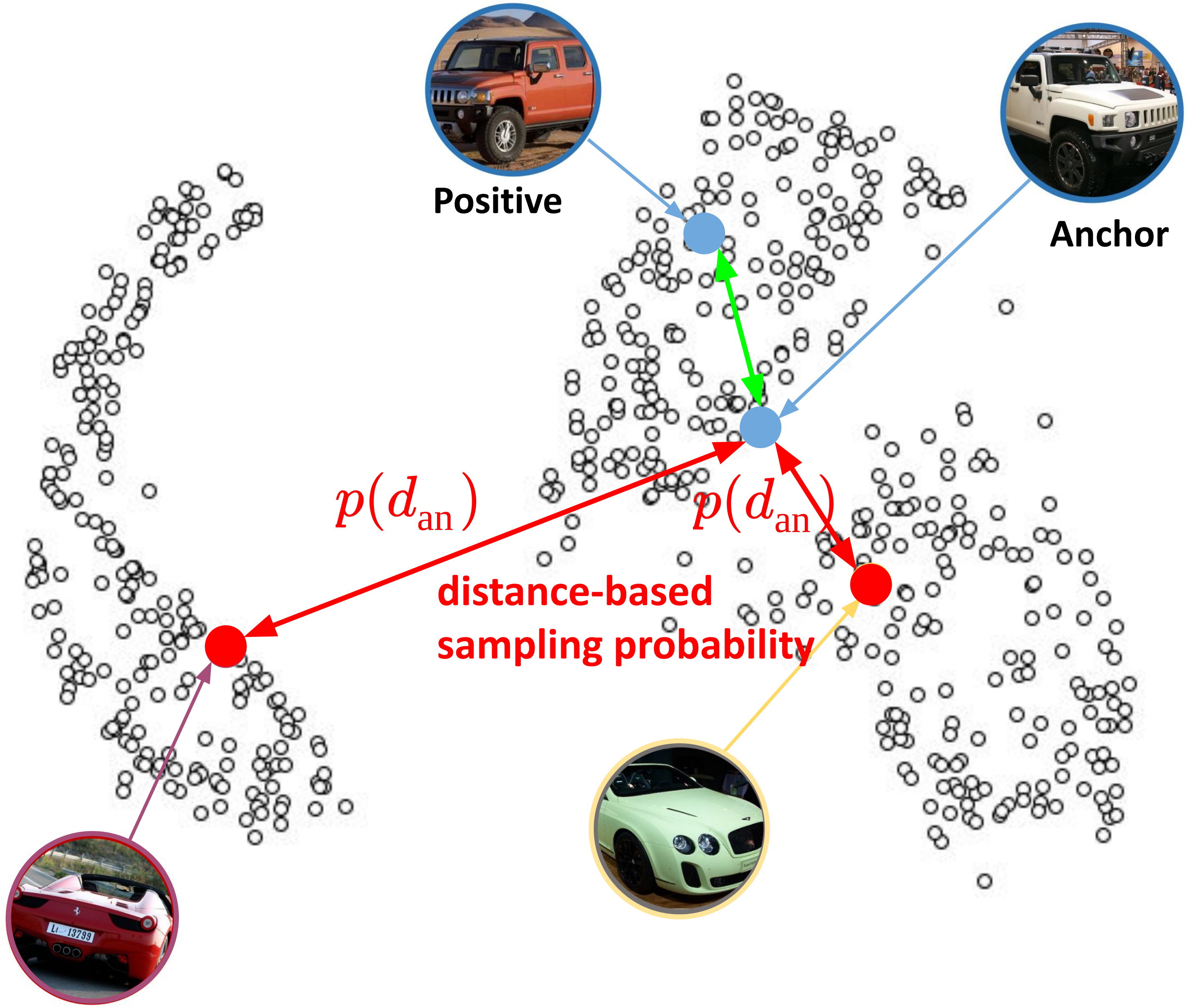
Negative Sampling in Ranking-based DML

- **Infeasible** amount of triplets to learn from (Scale $\mathcal{O}(n^3)$)
- Fixed sampling heuristics:

  - **(Semi-)Hard negative mining** [Schroff et al. 2015]

  - **Distance-weighted sampling** [Wu et al. 2017]

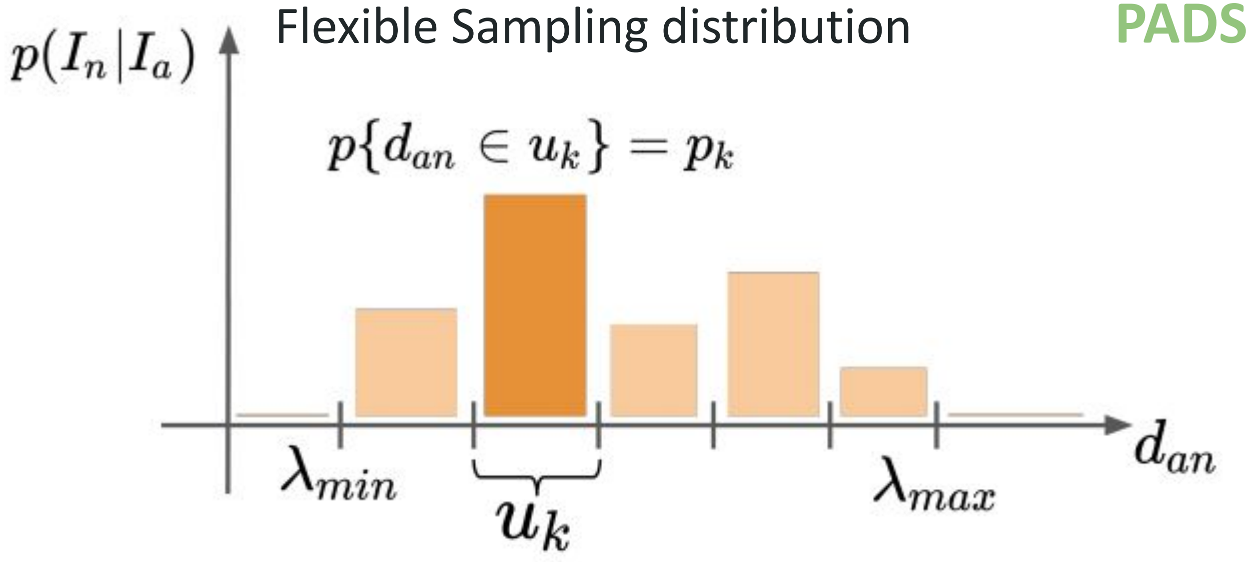
- **Drawbacks**:
  - Predefined and independent of DML optimization
  - Fixed and disconnected from learning process
    (see **Curriculum Learning**)
  - Optimal distribution $p(d_{\text{an}})$?

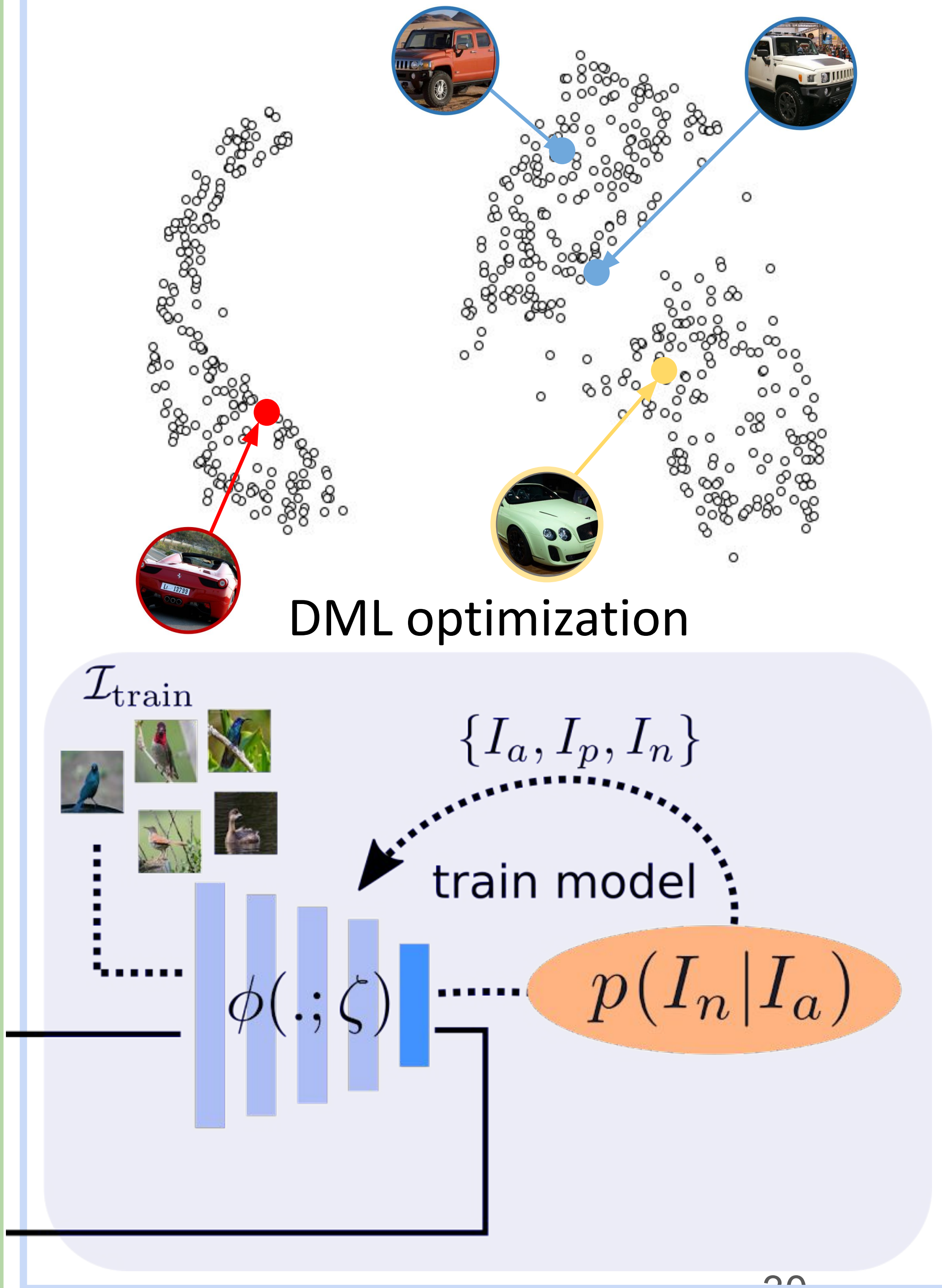


Negative Sampling in Ranking-based DML

PADS:Policy-adapted Sampling[5]

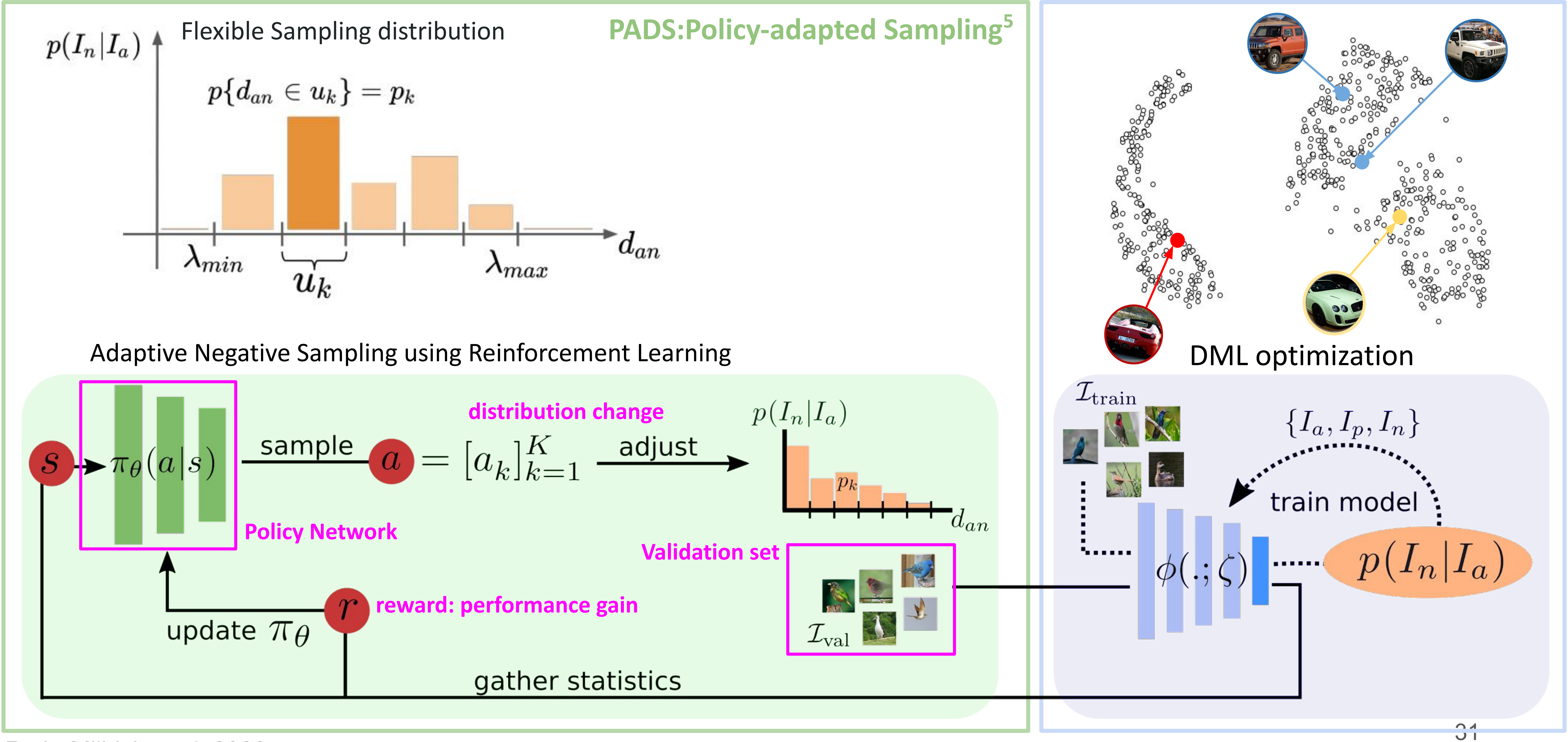
Flexible Sampling distribution





[5] Roth, Milbich et al. 2020

Negative Sampling in Ranking-based DML

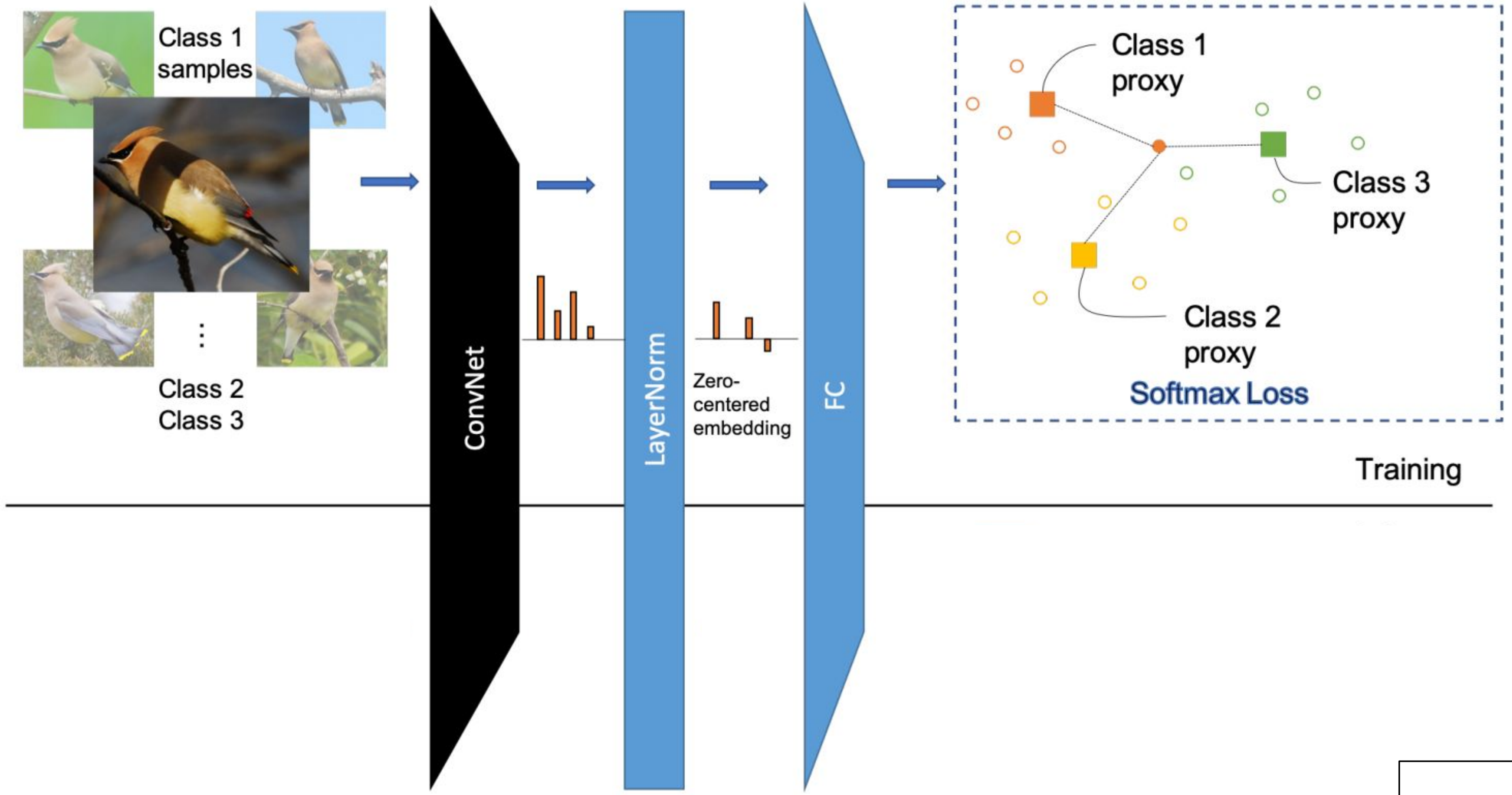


[5] Roth, Milbich et al. 2020

Classification-based DML

Classification-based DML

- **Classification** is "classical" approach to representation learning

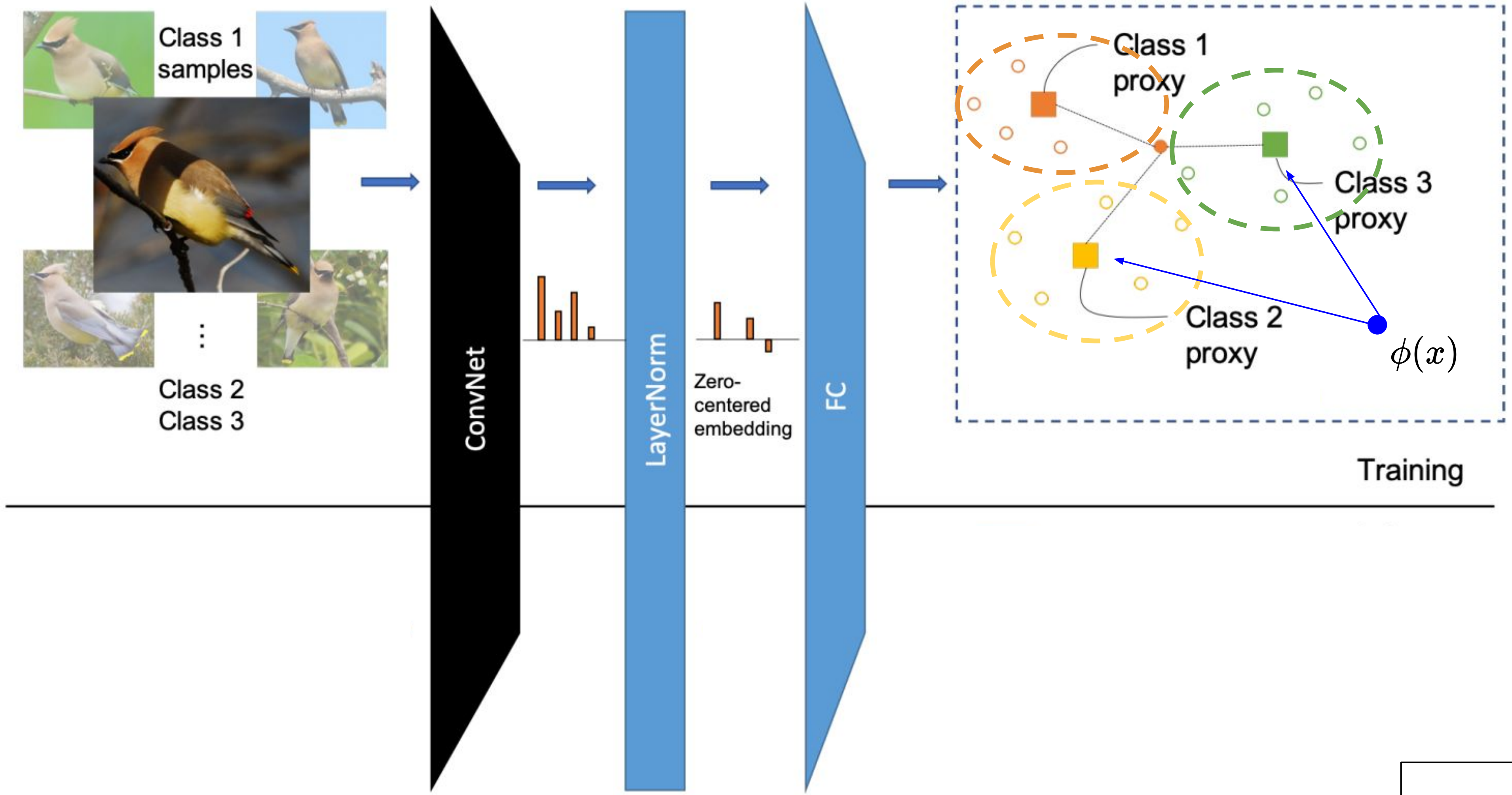


Softmax Loss :

$$\sigma(\phi(x_i)) = -\log\left[\frac{\exp(W_{y_i}^\top \phi(x_i))}{\sum_{k=1}^{K}\exp(W_k^\top \phi(x_i))}\right]$$

Images partly taken from: Zhai, Wu et al. 2019

Classification-based DML



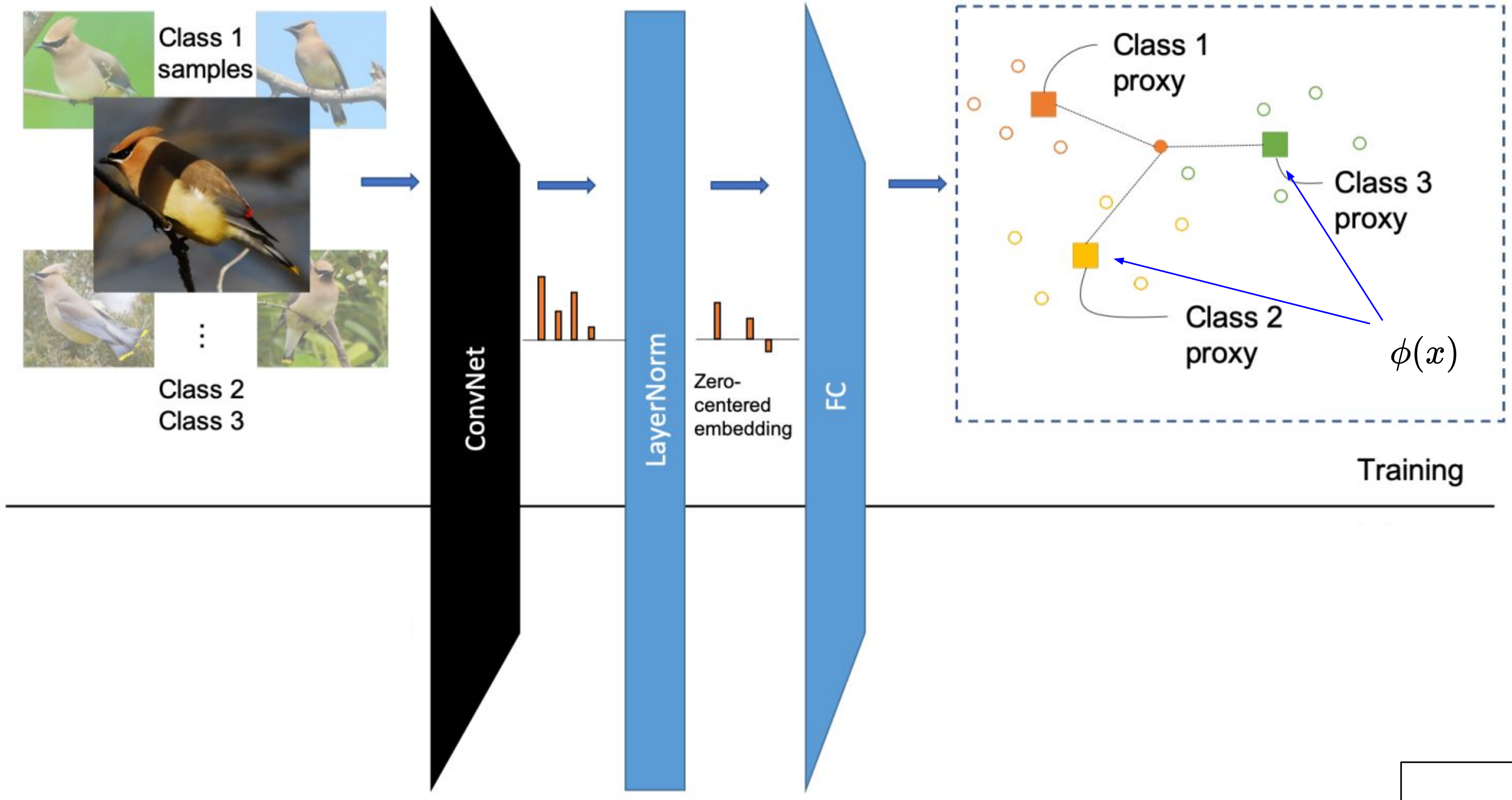
- **Classification** is "classical" approach to representation learning
- Rows of weights matrix W_k act as **class distribution proxies**
  ⇨ alleviate the sampling problem of ranking-based methods
- (Connections to **Proxy-based DML**)

Softmax Loss :

$$\sigma(\phi(x_i)) = -\log\left[\frac{\exp(\boxed{W_{y_i}^\top}\phi(x_i))}{\sum_{k=1}^{K}\exp(\boxed{W_k^\top}\phi(x_i))}\right]$$

Images partly taken from: Zhai, Wu et al. 2019

Classification-based DML



- **Classification** is "classical" approach to representation learning
- Rows of weights matrix W_k act as **class distribution proxies**
  ⇨ alleviate the sampling problem of ranking-based methods
- (Connections to **Proxy-based DML**)

Classification as **competitive baseline**[6]:

- Regularize embedding space to hyphersphere

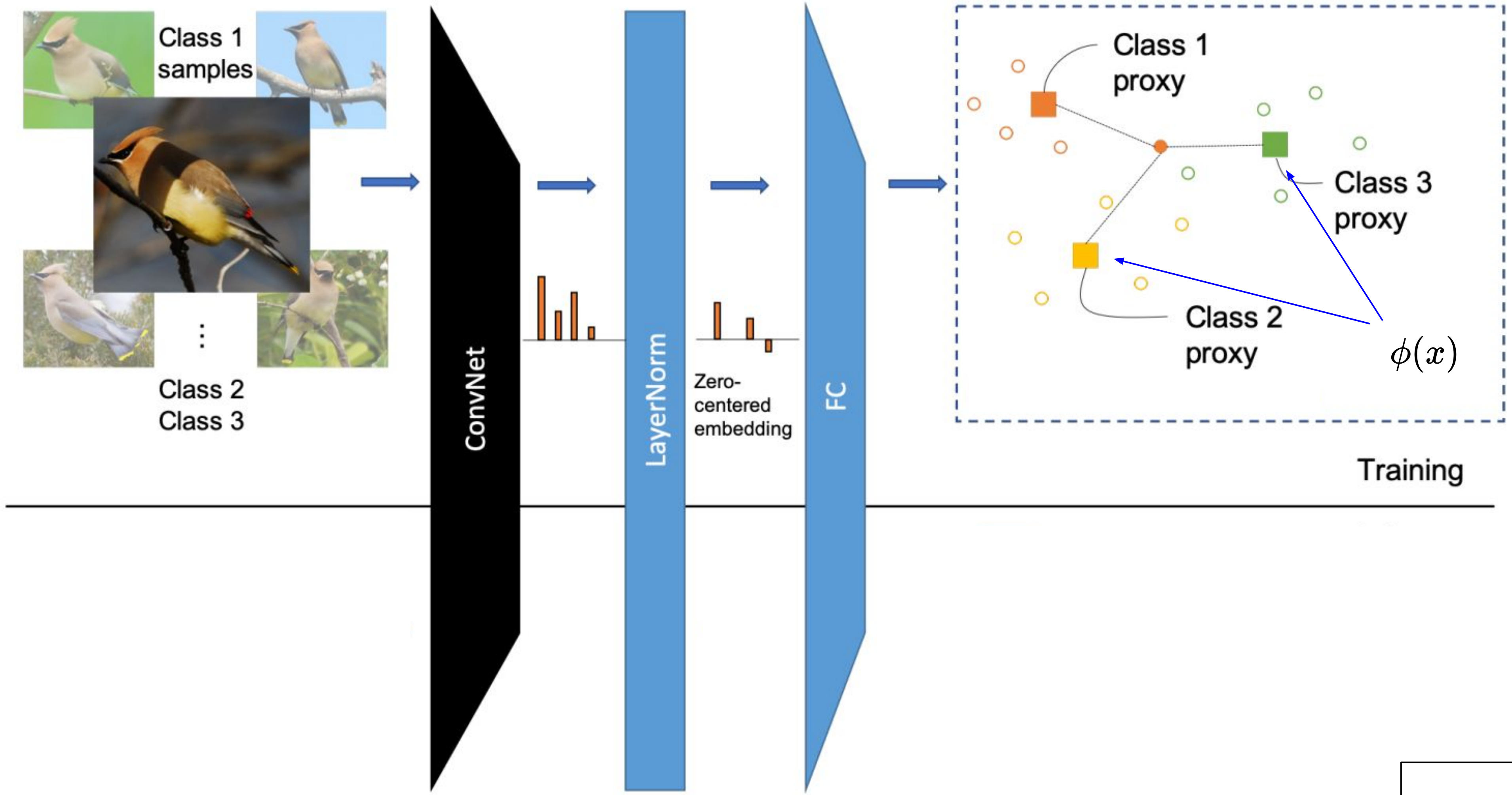
Normalized Softmax Loss :

L2 normalize

$$\sigma(\phi(x_i)) = -\log\left[\frac{\exp(\boxed{W_{y_i}^\top}\boxed{\phi(x_i)})}{\sum_{k=1}^{K}\exp(W_k^\top\phi(x_i))}\right]$$

[6]Zhai, Wu et al. 2019

Classification-based DML



- **Classification** is "classical" approach to representation learning
- Rows of weights matrix W_k act as **class distribution proxies**
  ⇨ alleviate the sampling problem of ranking-based methods
- (Connections to **Proxy-based DML**)

Classification as **competitive baseline**[6]:

- Regularize embedding space to hyphersphere
- Temperature scaling to enforce compact intra-class clusters

Normalized Softmax Loss :

L2 normalize

Temperature scaling

$$\sigma(\phi(x_i)) = -\log\left[\frac{\exp(\boxed{W_{y_i}^\top}\boxed{\phi(x_i)}/\boxed{\tau})}{\sum_{k=1}^{K}\exp(W_k^\top\phi(x_i)/\tau)}\right]$$

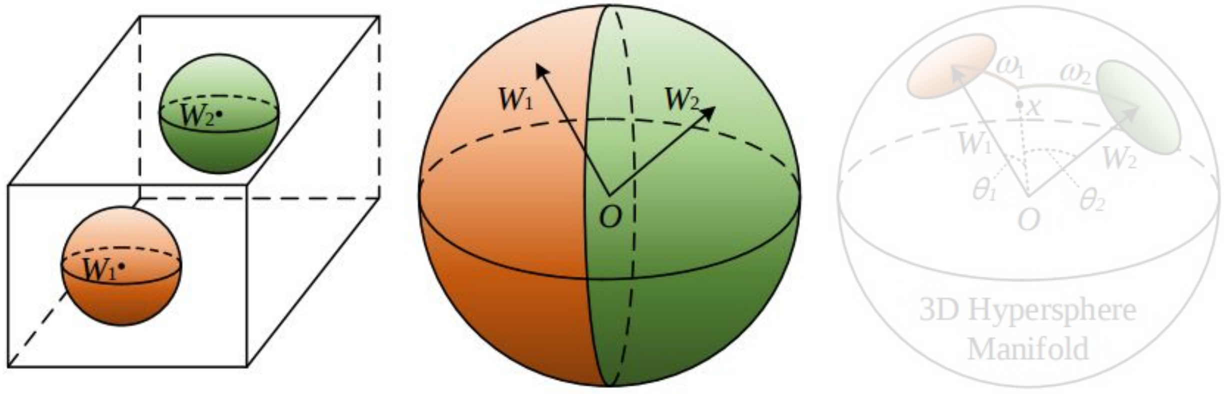
[6]Zhai, Wu et al. 2019

Classification-based DML

$$\sigma(\phi(x_i)) = -\log\left[\frac{\exp(W_{y_i}^\top \phi(x_i))}{\sum_{k=1}^{K}\exp(W_k^\top \phi(x_i))}\right]$$

$x^T y$

- So far, we optimize actual **distances** (e.g. Euclidean, Cosine) between data samples/proxies
- Now: Express learning constraints explicitly as actual **angles on (Hyphersphere-)Manifold**
- very popular for **Face recognition applications**



Euclidean Margin Loss Modified Softmax Loss A-Softmax Loss (m≥2)

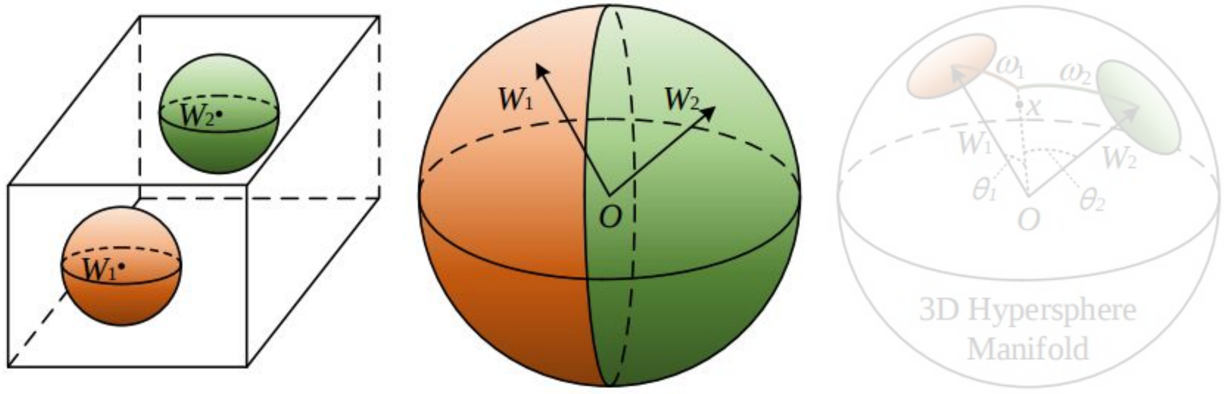
Images partly taken from: Liu et al. 2018

Classification-based DML

- So far, we optimize actual **distances** (e.g. Euclidean, Cosine) between data samples/proxies
- Now: Express learning constraints explicitly as actual **angles on (Hyphersphere-)Manifold**
- very popular for **Face recognition applications**

$$
\begin{aligned}
\sigma(\phi(x_i)) &= -\log\left[\frac{\exp(W_{y_i}^{\top}\phi(x_i))}{\sum_{k=1}^{K}\exp(W_k^{\top}\phi(x_i))}\right] \\
&= -\log\left[\frac{\exp(\|W_{y_i}\|\,\|\phi(x_i)\|\cos(\theta_{i,y_i}))}{\sum_{k=1}^{K}\exp(\|W_k\|\,\|\phi(x_i)\|\cos(\theta_{i,k}))}\right].
\end{aligned}
$$

$\boxed{x^T y = \|x\|_2 \|y\|_2 \cos(\theta)}$



Euclidean Margin Loss | Modified Softmax Loss | A-Softmax Loss (m≥2)

Images partly taken from: Liu et al. 2018

Classification-based DML

- So far, we optimize actual **distances** (e.g. Euclidean, Cosine) between data samples/proxies
- Now: Express learning constraints explicitly as actual **angles on (Hyphersphere-)Manifold**
- very popular for **Face recognition applications**

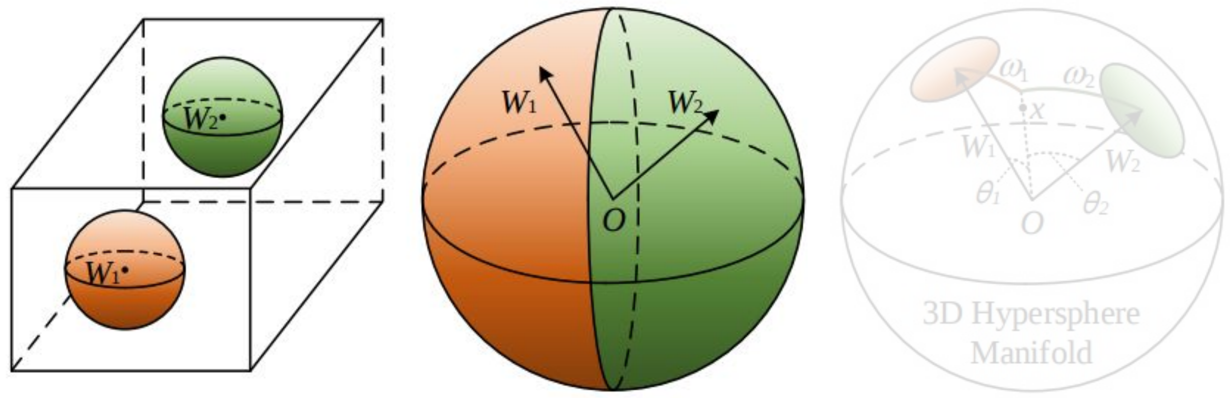
$$\sigma(\phi(x_i)) = -\log\left[\frac{\exp(W_{y_i}^\top \phi(x_i))}{\sum_{k=1}^{K}\exp(W_k^\top \phi(x_i))}\right]$$
$$= -\log\left[\frac{\exp(\|W_{y_i}\|\,\|\phi(x_i)\|\cos(\theta_{i,y_i}))}{\sum_{k=1}^{K}\exp(\|W_k\|\,\|\phi(x_i)\|\cos(\theta_{i,k}))}\right].$$

$x^T y = \|x\|_2 \|y\|_2 \cos(\theta)$

- Constrain $\|W_{y_i}\| = 1$ and $\|\phi(x_i)\|_2 = s$

$$\sigma(\phi(x_i)) = -\log\left[\frac{\exp(s\cos(\boxed{\theta_{i,y_i}}))}{\sum_{k=1}^{K}\exp(s\cos(\boxed{\theta_{i,k}}))}\right].$$

- We now optimize the angles $\theta_{i,k}$ between $\phi(x_i)$ and W_k



Euclidean Margin Loss Modified Softmax Loss A-Softmax Loss (m≥2)

Images partly taken from: Liu et al. 2018

Classification-based DML

- So far, we optimize actual **distances** (e.g. Euclidean, Cosine) between data samples/proxies
- Now: Express learning constraints explicitly as actual **angles on (Hyphersphere-)Manifold**
- very popular for **Face recognition applications**

$$\sigma(\phi(x_i)) = -\log\left[\frac{\exp(W_{y_i}^\top \phi(x_i))}{\sum_{k=1}^{K}\exp(W_k^\top \phi(x_i))}\right]$$

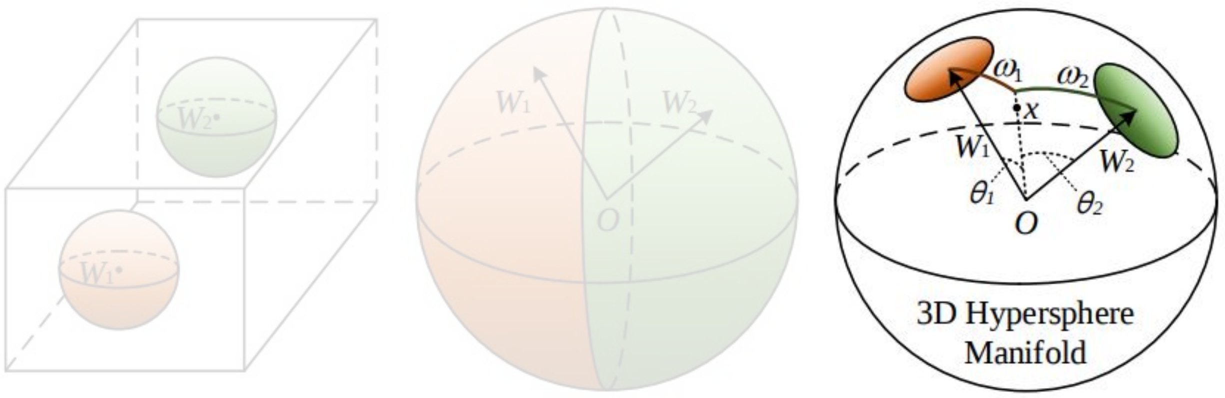
$$x^T y = \|x\|_2 \|y\|_2 \cos(\theta)$$

$$= -\log\left[\frac{\exp(\|W_{y_i}\|\,\|\phi(x_i)\|\cos(\theta_{i,y_i}))}{\sum_{k=1}^{K}\exp(\|W_k\|\,\|\phi(x_i)\|\cos(\theta_{i,k}))}\right].$$

- Constrain $\|W_{y_i}\| = 1$ and $\|\phi(x_i)\|_2 = s$

$$\sigma(\phi(x_i)) = -\log\left[\frac{\exp(s\cos(\boxed{\theta_{i,y_i}}))}{\sum_{k=1}^{K}\exp(s\cos(\boxed{\theta_{i,k}}))}\right].$$

- We now optimize the angles $\theta_{i,k}$ between $\phi(x_i)$ and W_k
- Introduce a margin β similar to Ranking-based DML



Euclidean Margin Loss | Modified Softmax Loss | A-Softmax Loss (m≥2)

Sphere[7]-/ArcFace[8] Loss:

fixed margin

$$\sigma(\phi(x_i);\beta) = -\log\left[\frac{\exp(s\cos(\boxed{\beta} + \theta_{i,y_i}))}{\exp(s\cos(\beta + \theta_{i,y_i})) + \sum_{k=1,k\neq y_i}^{M}\exp(s\cos(\theta_{i,k}))}\right].$$

[7]Liu et al. 2018; [8]Deng, Guo et al. 2019

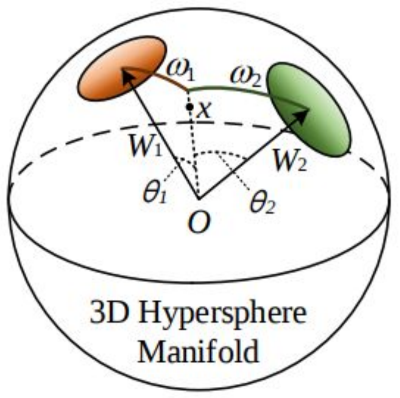
Classification-based DML

- So far, we optimize actual **distances** (e.g. Euclidean, Cosine) between data samples/proxies
- Now: Express learning constraints explicitly as actual **angles on (Hyphersphere-)Manifold**
- very popular for **Face recognition applications**

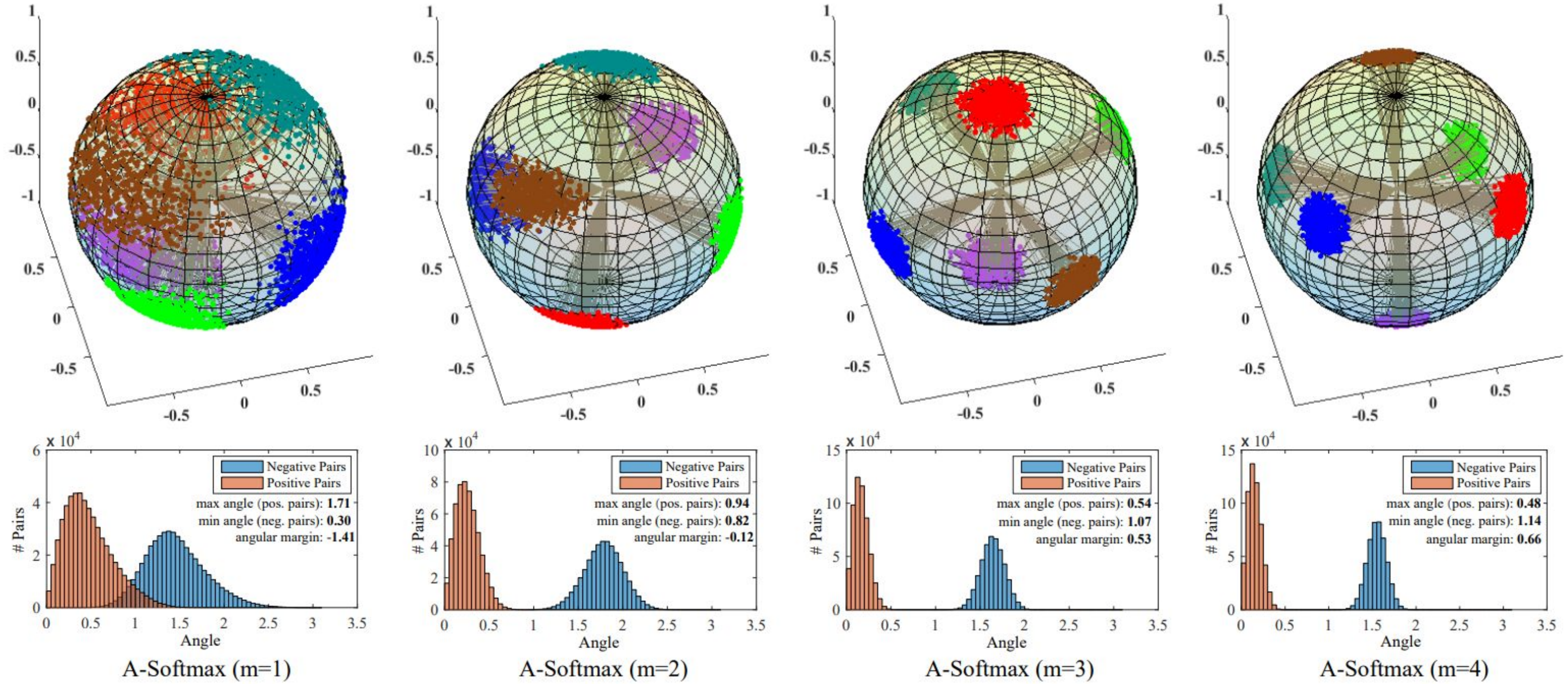
Many different extensions based on this formulations!

Sphere[7]-/ArcFace[8] Loss:

$$\sigma(\phi(x_i);\beta) = -\log\left[\frac{\exp(s\cos(\beta+\theta_{i,y_i}))}{\exp(s\cos(\beta+\theta_{i,y_i})) + \sum_{k=1,k\neq y_i}^{M}\exp(s\cos(\theta_{i,k}))}\right].$$



A-Softmax Loss (m≥2)

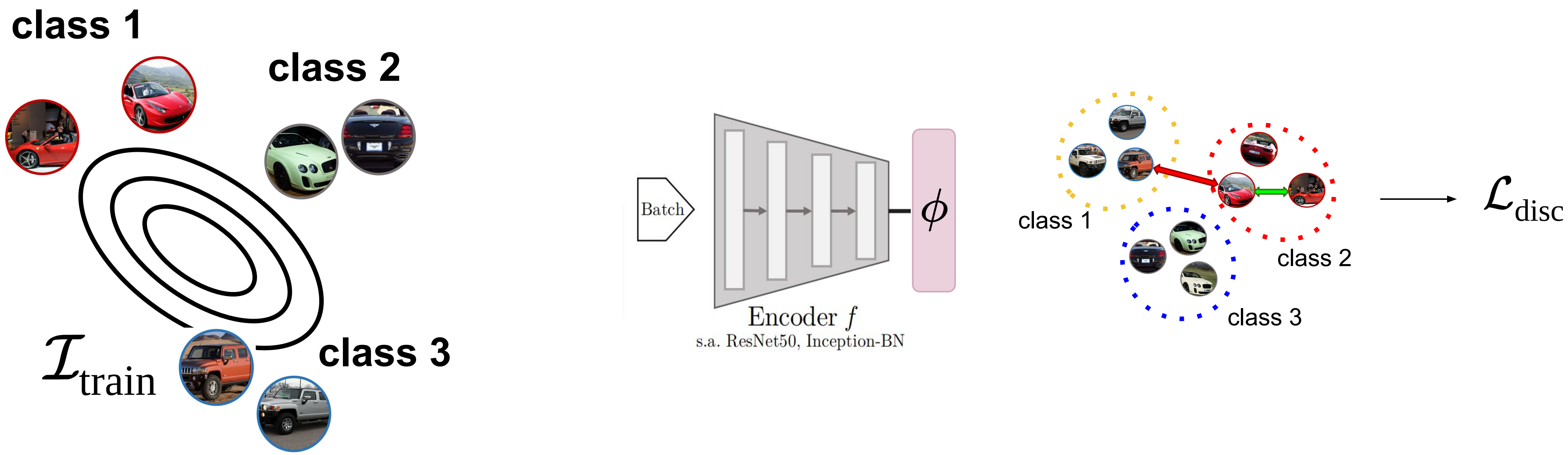


[7]Liu et al. 2018; [8]Deng, Guo et al. 2019

Ensemble-based DML

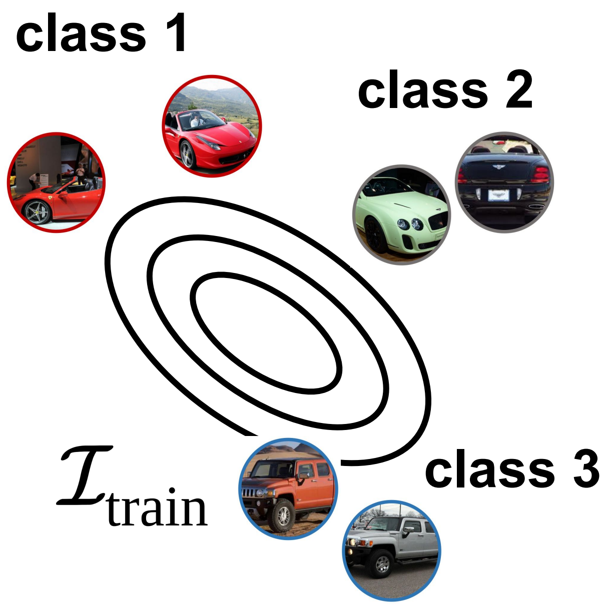
Ensemble-based DML

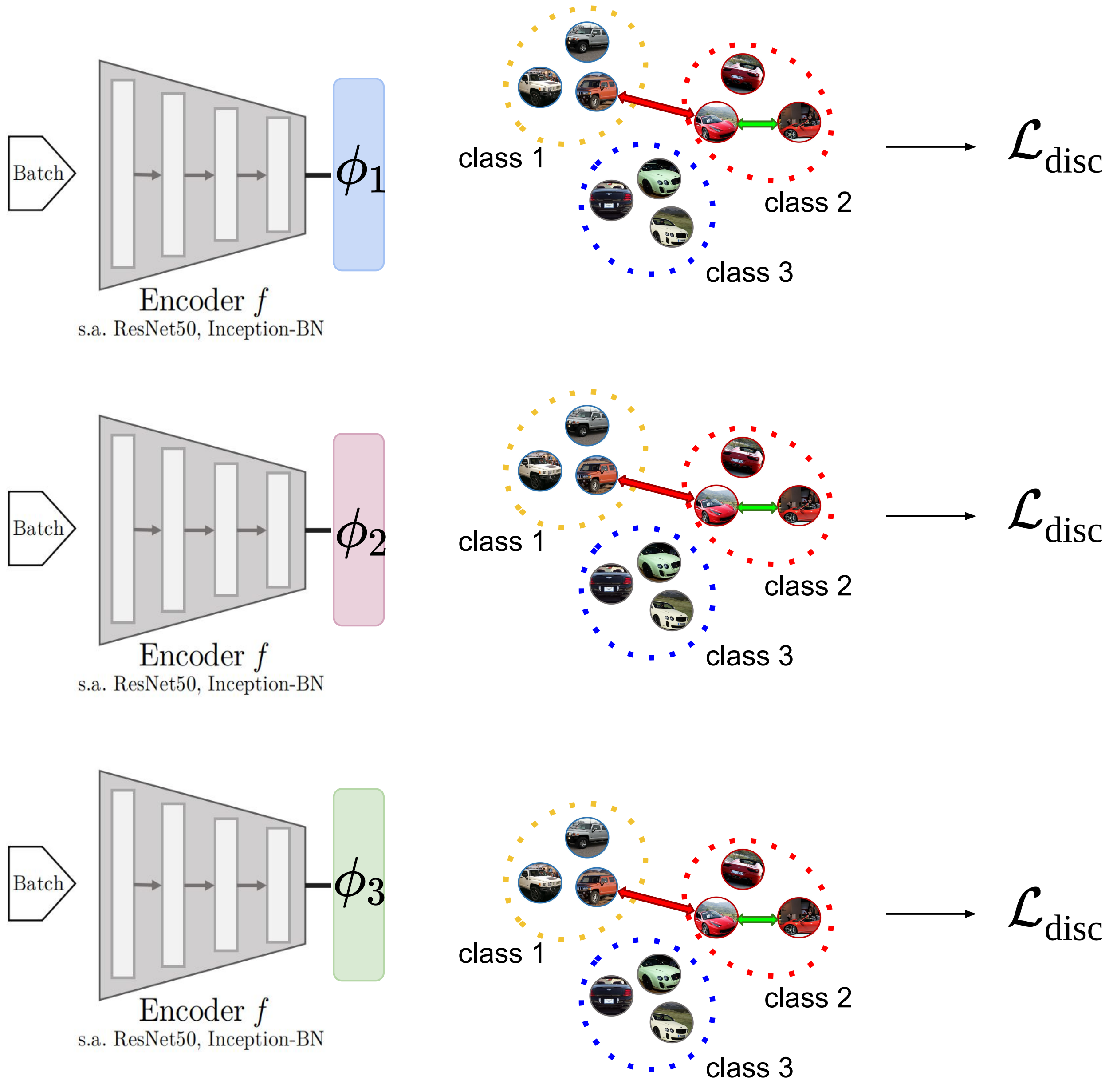
- **Learner ensembles** are common approach to improve overall performance.



Ensemble-based DML

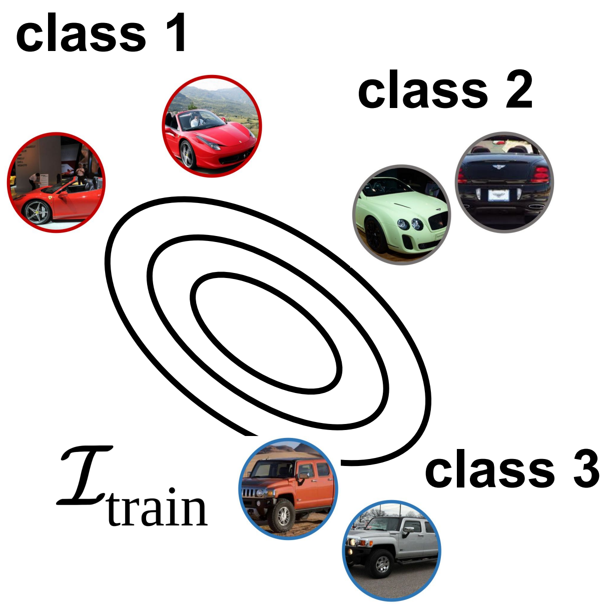
- **Learner ensembles** are common approach to improve overall performance.
- **Naive approach**: Independently train K independent models using the same, standard discriminative loss $\mathcal{L}_{\text{disc}}$

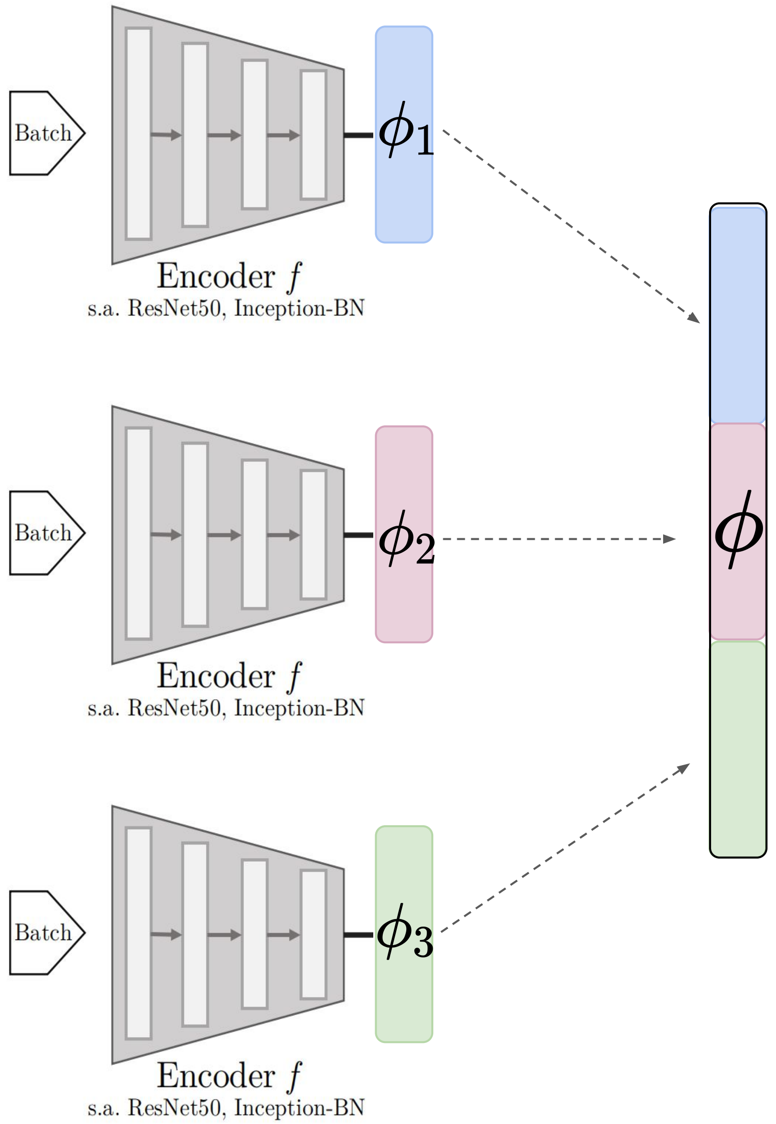




Ensemble-based DML

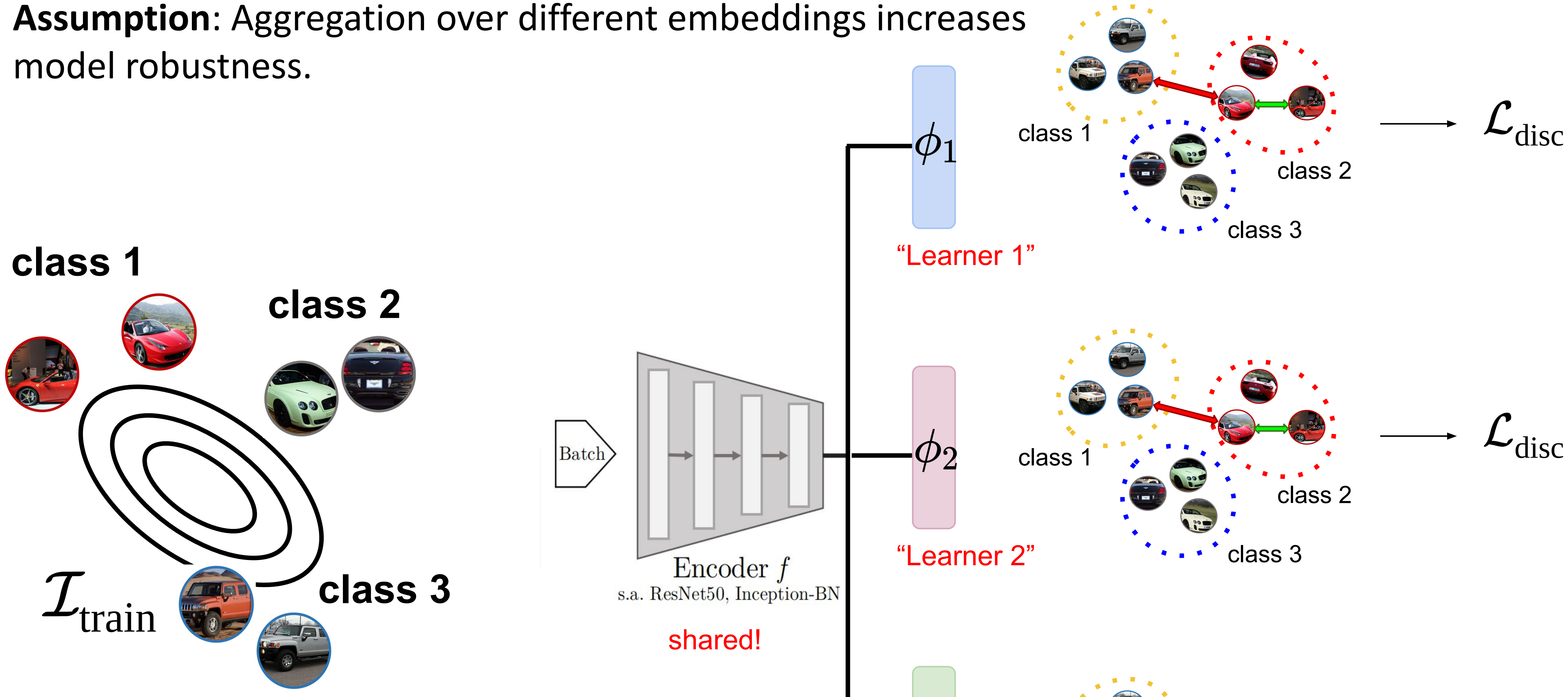
- **Learner ensembles** are common approach to improve overall performance.
- **Naive approach**: Independently train K different models using the same, standard discriminative loss $\mathcal{L}_{\text{disc}}$
- **Assumption**: Aggregation over different embeddings increases model robustness.





Ensemble-based DML

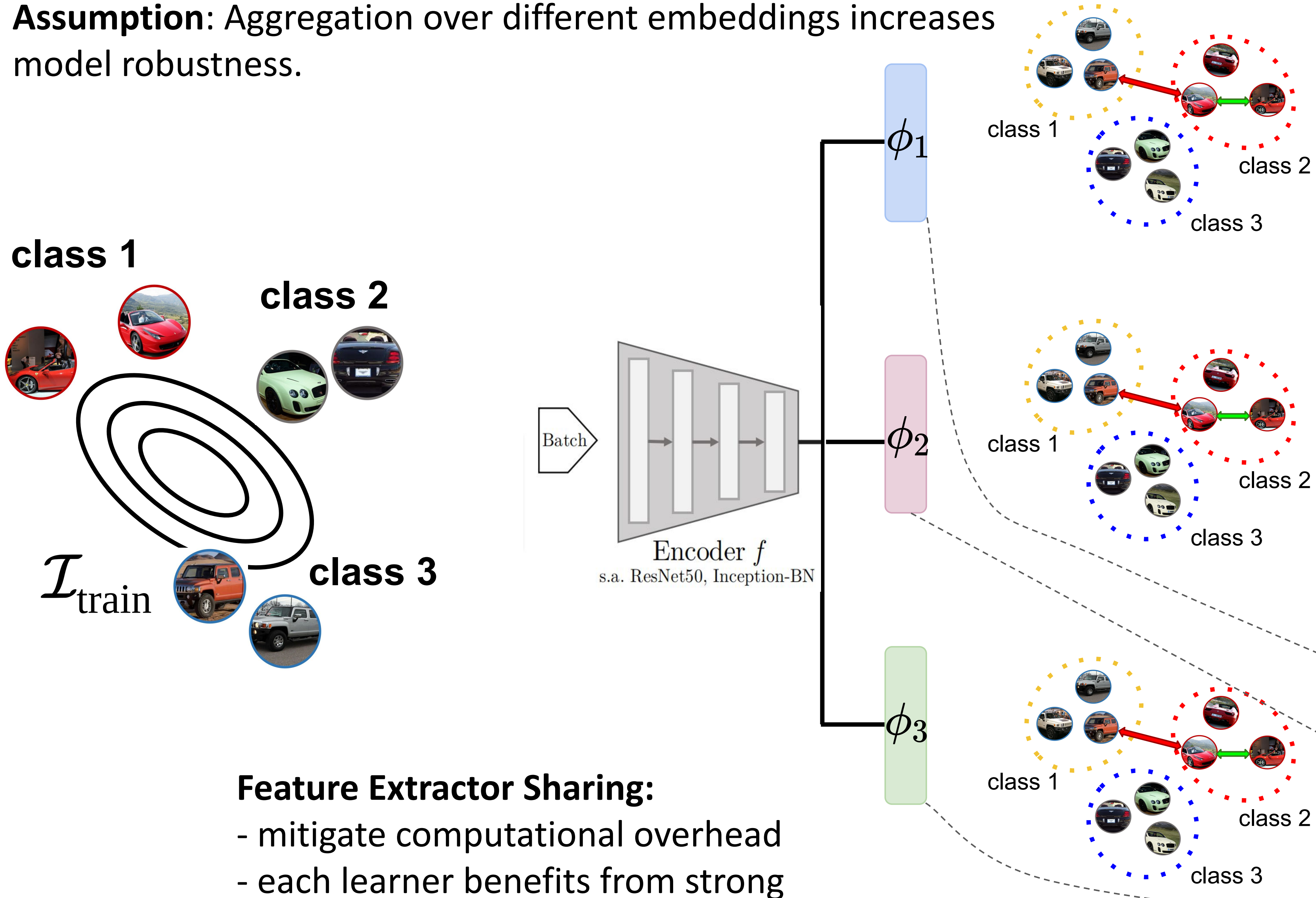
- **Learner ensembles** are common approach to improve overall performance.
- **Naive approach**: Independently train K different models using the same, standard discriminative loss $\mathcal{L}_{\text{disc}}$
- **Assumption**: Aggregation over different embeddings increases model robustness.



Feature Extractor Sharing:
- mitigate computational overhead
- each learner benefits from strong feature representation

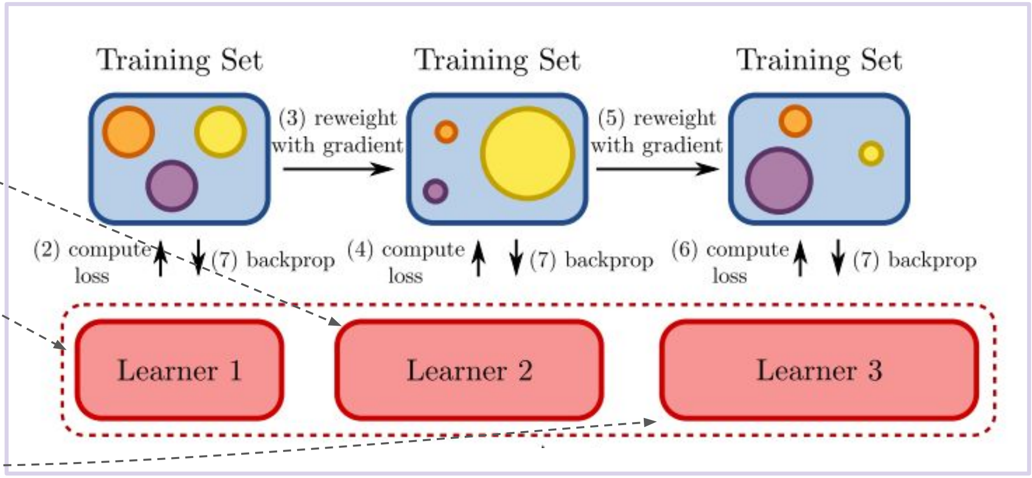
Ensemble-based DML

- **Learner ensembles** are common approach to improve overall performance.
- **Naive approach**: Independently train K different models using the same, standard discriminative loss $\mathcal{L}_{\text{disc}}$
- **Assumption**: Aggregation over different embeddings increases model robustness.



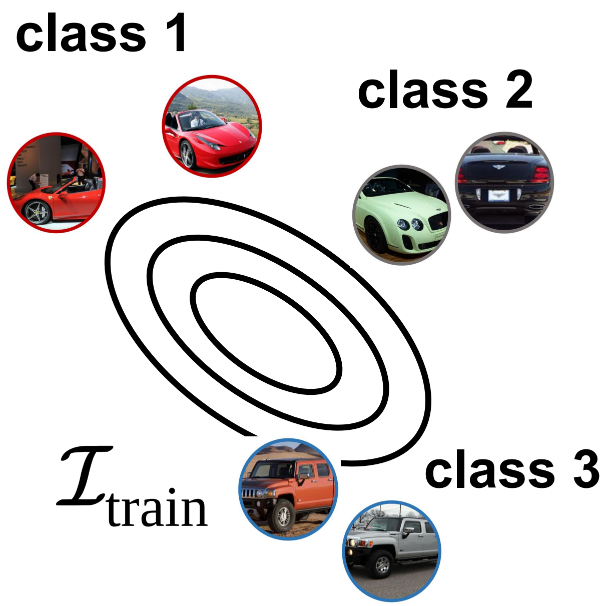
Feature Extractor Sharing:
- mitigate computational overhead
- each learner benefits from strong feature representation

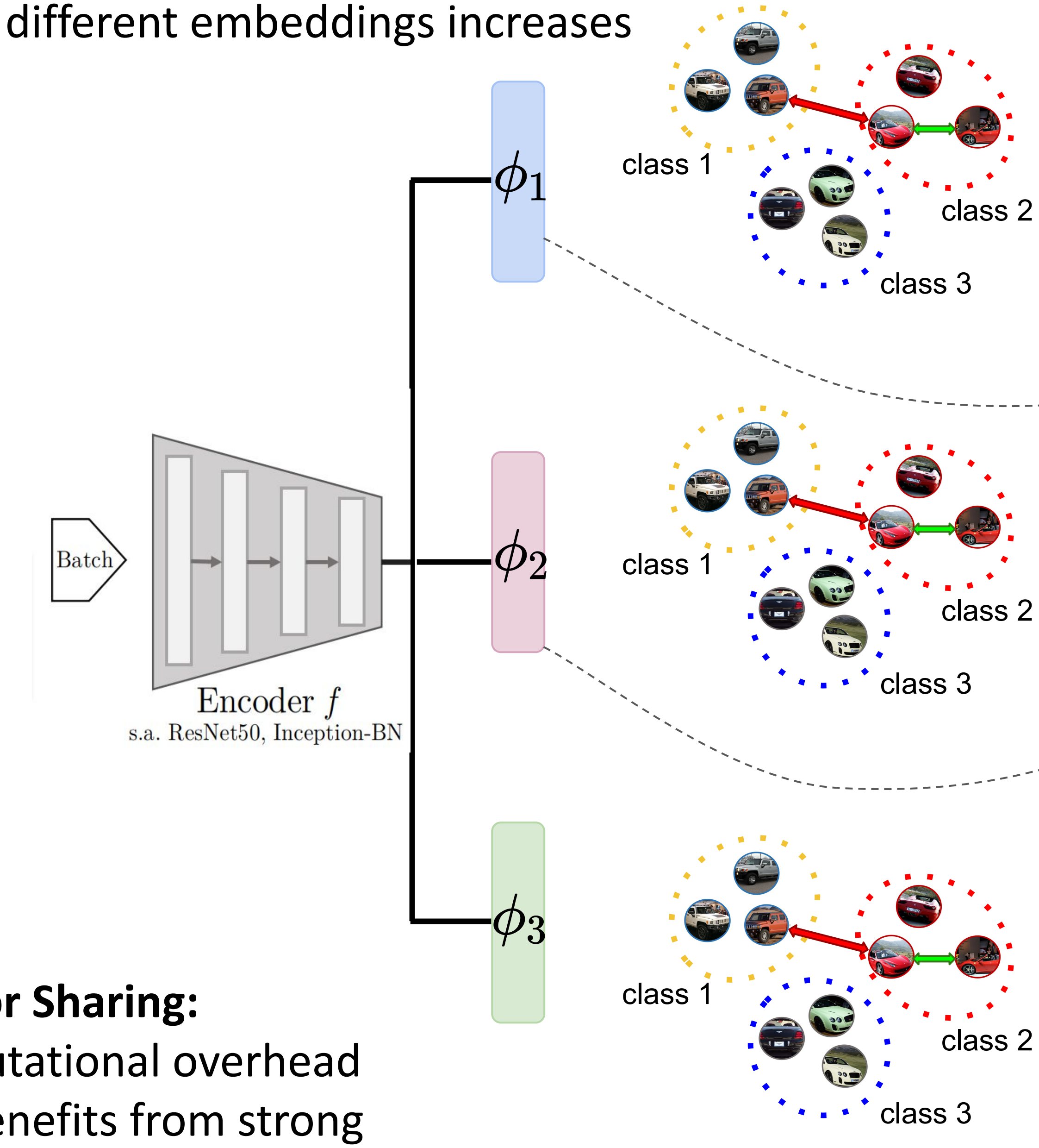
BIER: Online Gradient Boosting [Opitz et al. 2018]: implicit specialization of learners

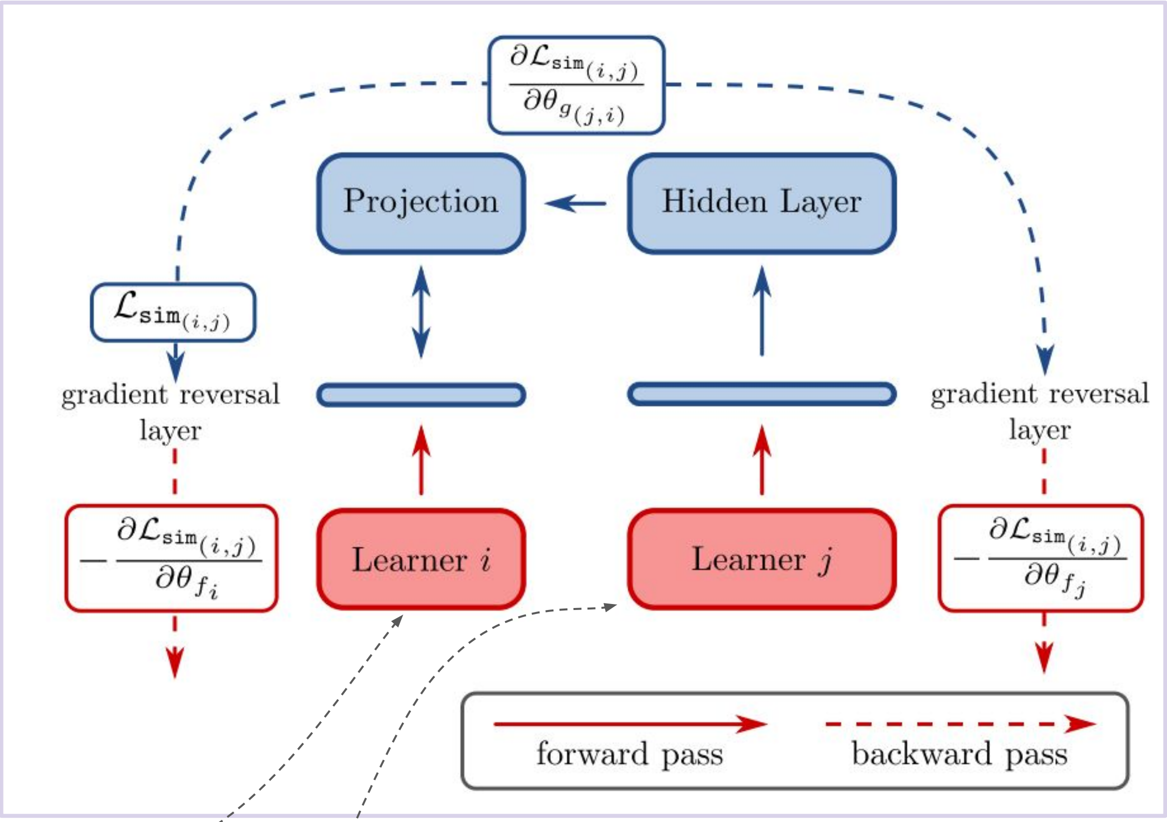


Ensemble-based DML

- **Learner ensembles** are common approach to improve overall performance.
- **Naive approach**: Independently train K different models using the same, standard discriminative loss
- **Assumption**: Aggregation over different embeddings increases model robustness.

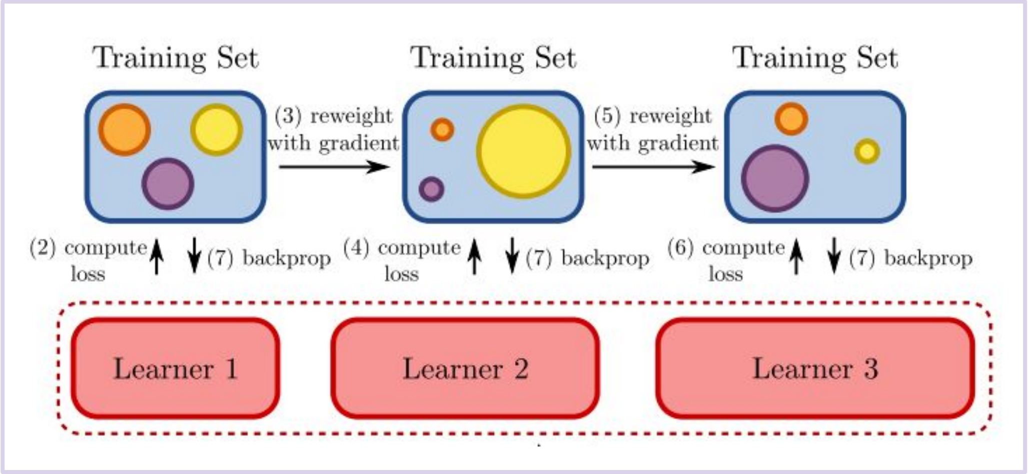






BIER: Adversarial Decorrelation [Opitz et al. 2018]: encourage learners to focus on different features

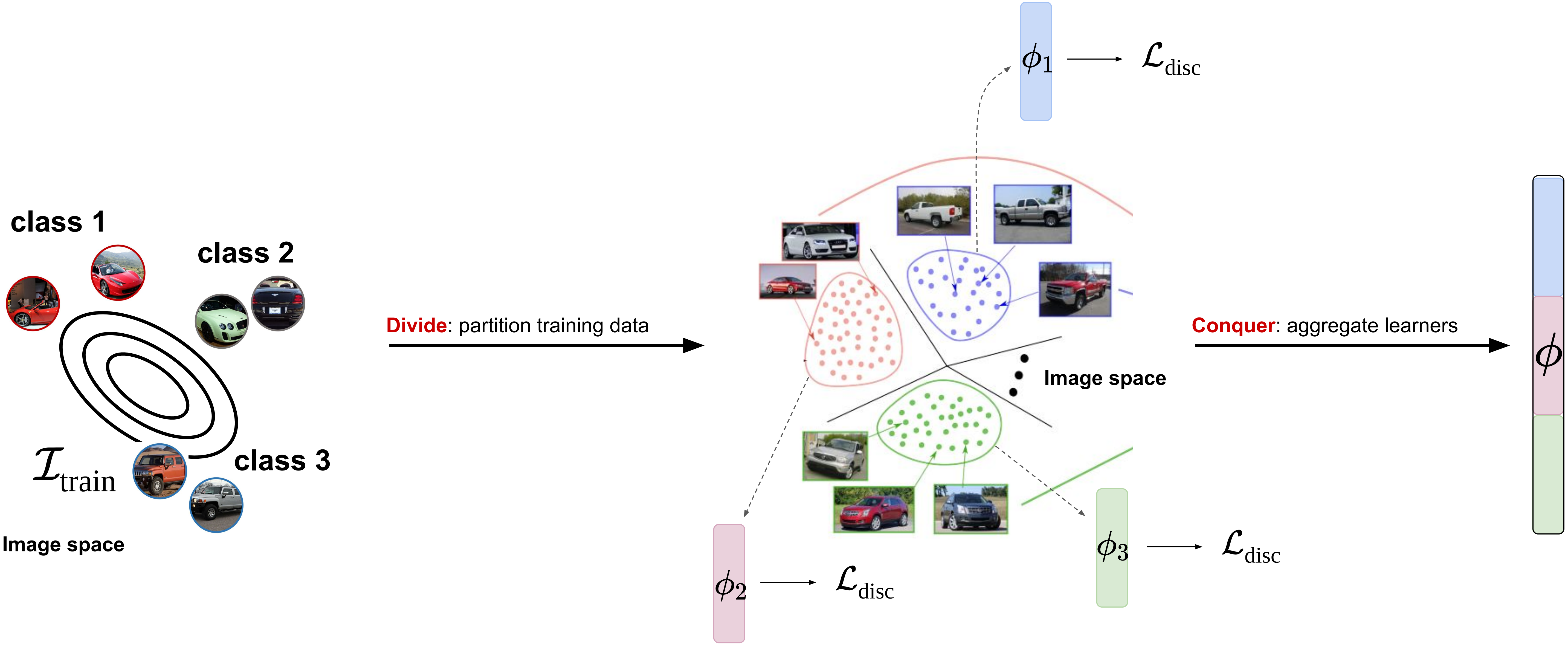
BIER: Online Gradient Boosting [Opitz et al. 2018]: implicit specialization of learners



Feature Extractor Sharing:
- mitigate computational overhead
- each learner benefits from strong feature representation

Ensemble-based DML

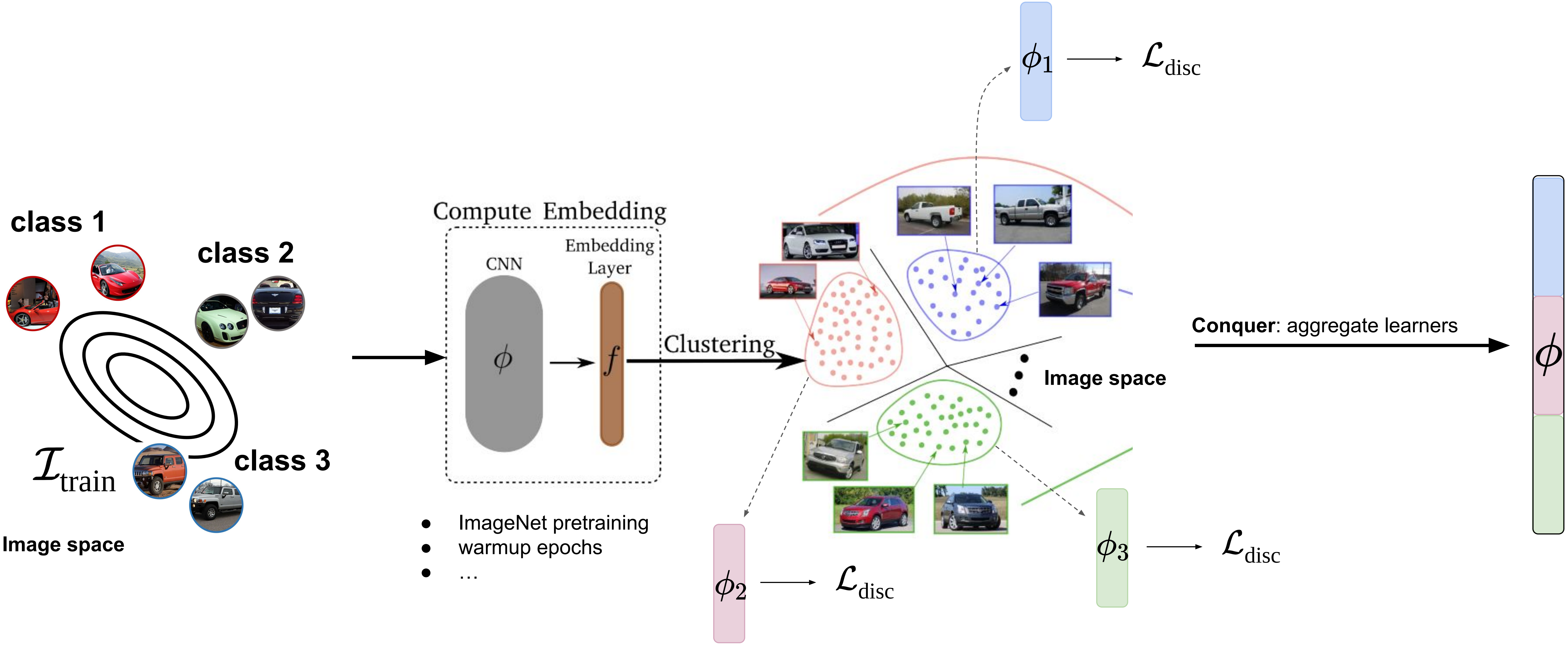
- Typically training distribution is **multimodal**
- **Explicit** data specialization of learners following **Divide & Conquer strategy**[9]



[9]Sanakoyeu, Tschernetzski, Büchler, Ommer 2019

Ensemble-based DML

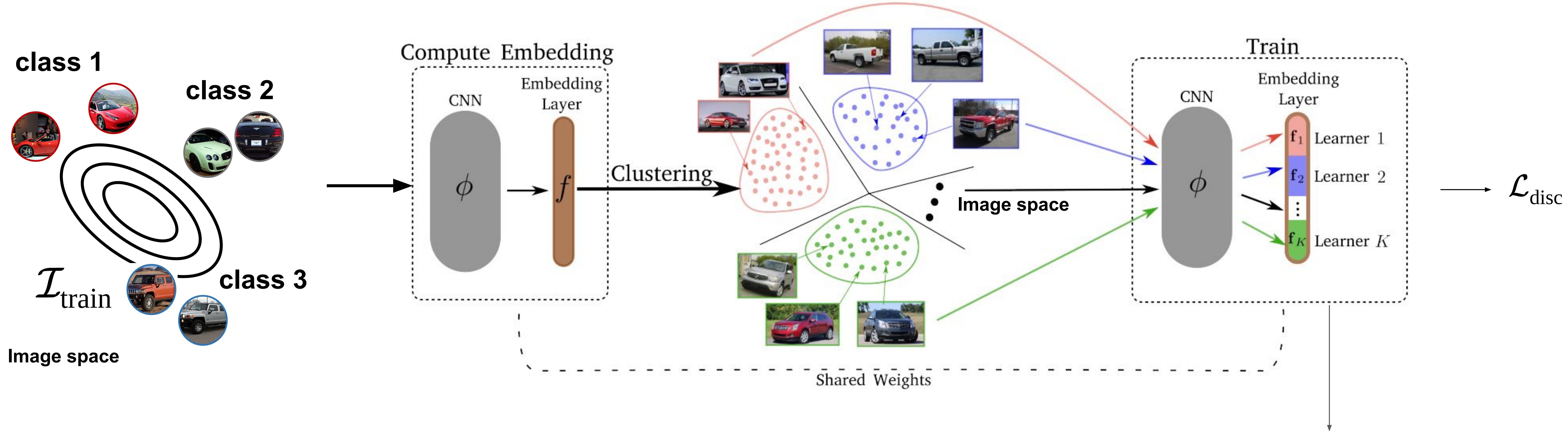
- Typically training distribution is **multimodal**
- **Explicit** data specialization of learners following **Divide & Conquer strategy**[9]



[9]Sanakoyeu, Tschernetzski, Büchler, Ommer 2019

Ensemble-based DML

- Typically training distribution is **multimodal**
- **Explicit** data specialization of learners following **Divide & Conquer strategy**[9]

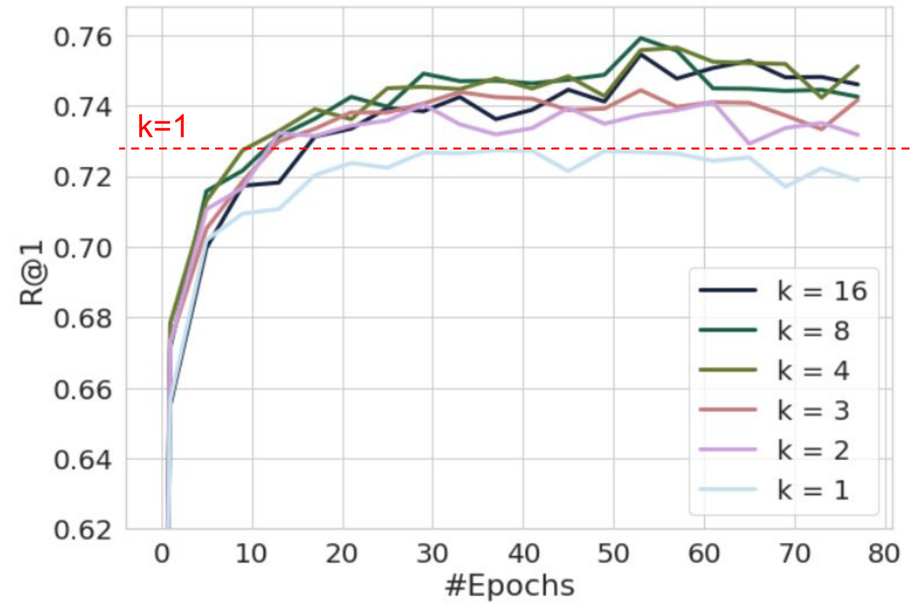


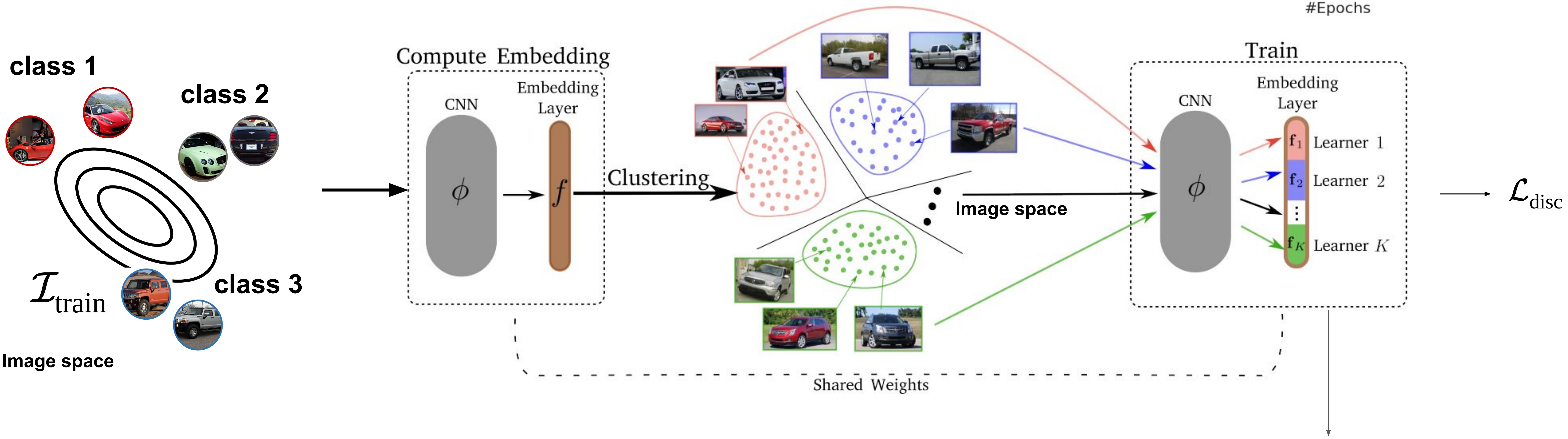
Postprocessing: Joint global finetuning of concatenated learners to improve consistency of embeddings.

[9]Sanakoyeu, Tschernetzski, Büchler, Ommer 2019

Ensemble-based DML

- Typically training distribution is **multimodal**
- **Explicit** data specialization of learners following **Divide & Conquer strategy**[9]



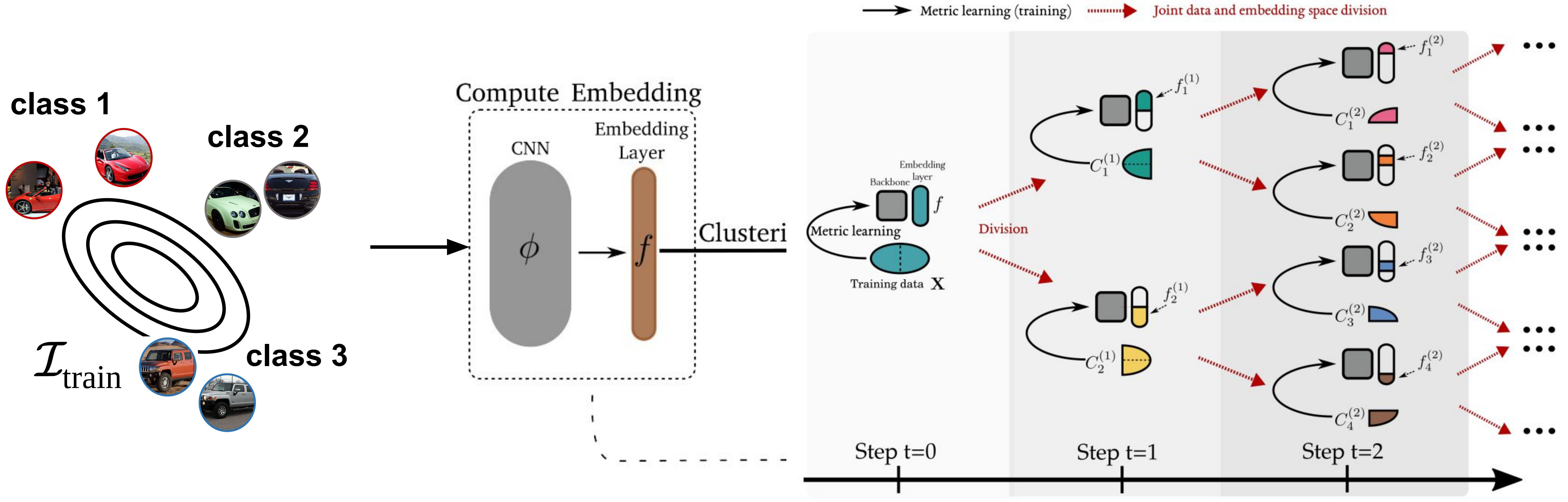


Postprocessing: Joint global finetuning of concatenated learners to improve consistency of embeddings.

[9] Sanakoyeu, Tschernetzski, Büchler, Ommer 2019

Ensemble-based DML

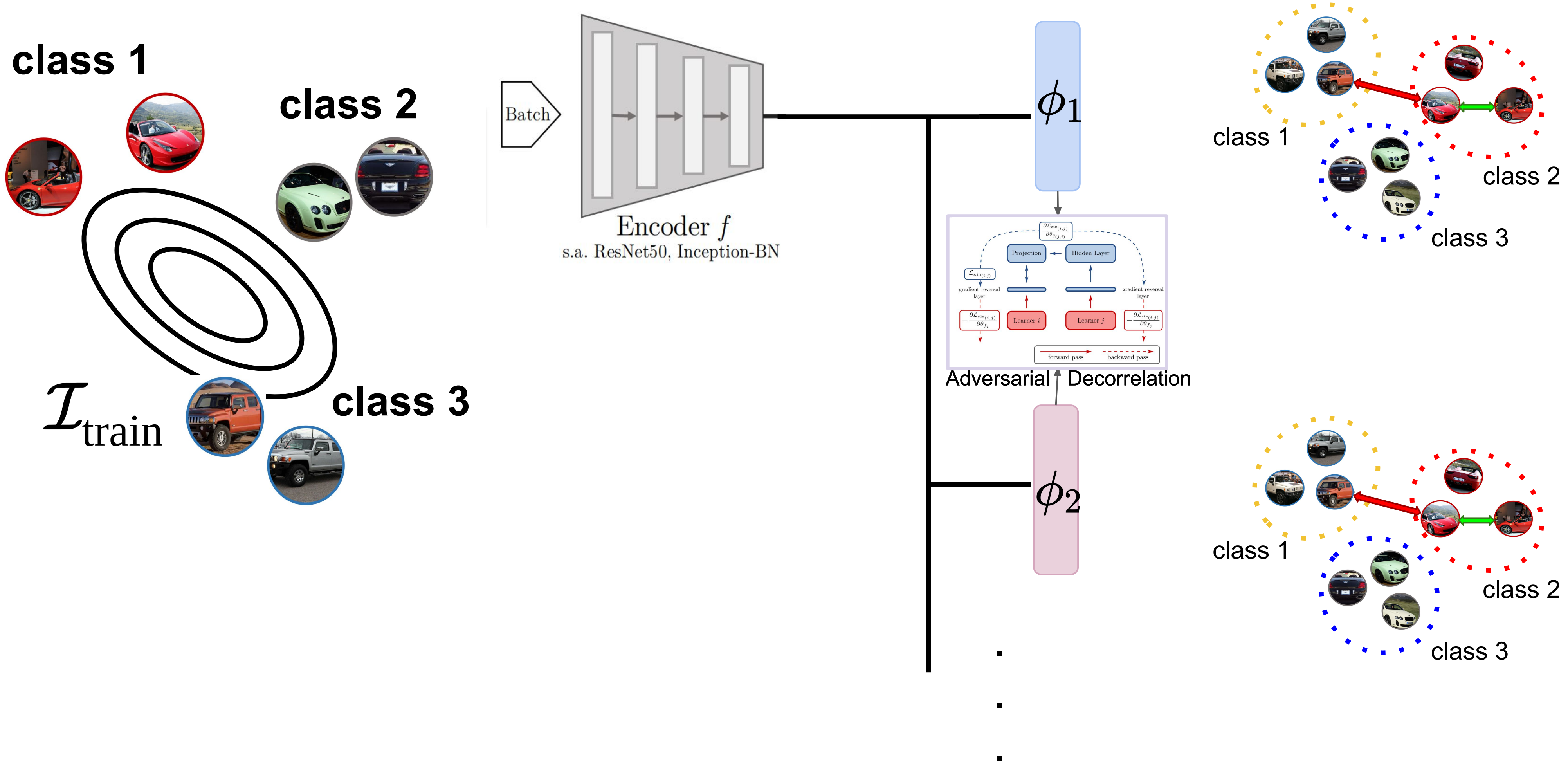
- Typically training distribution is **multimodal**
- **Explicit** data specialization of learners following **Hierarchical Divide & Conquer strategy**[10]



Sequential Splitting: Continuous specialization by splitting data and adding learners during optimization

[10]Sanakoyeu, Ma, Tschernetzski, Ommer 2021

Ensemble-based DML



Target similar object features!

$$\xrightarrow[y_a = y_p \neq y_n]{t = \{I_a, I_p, I_n\}} \boxed{\mathcal{L}_{\text{disc}}}$$

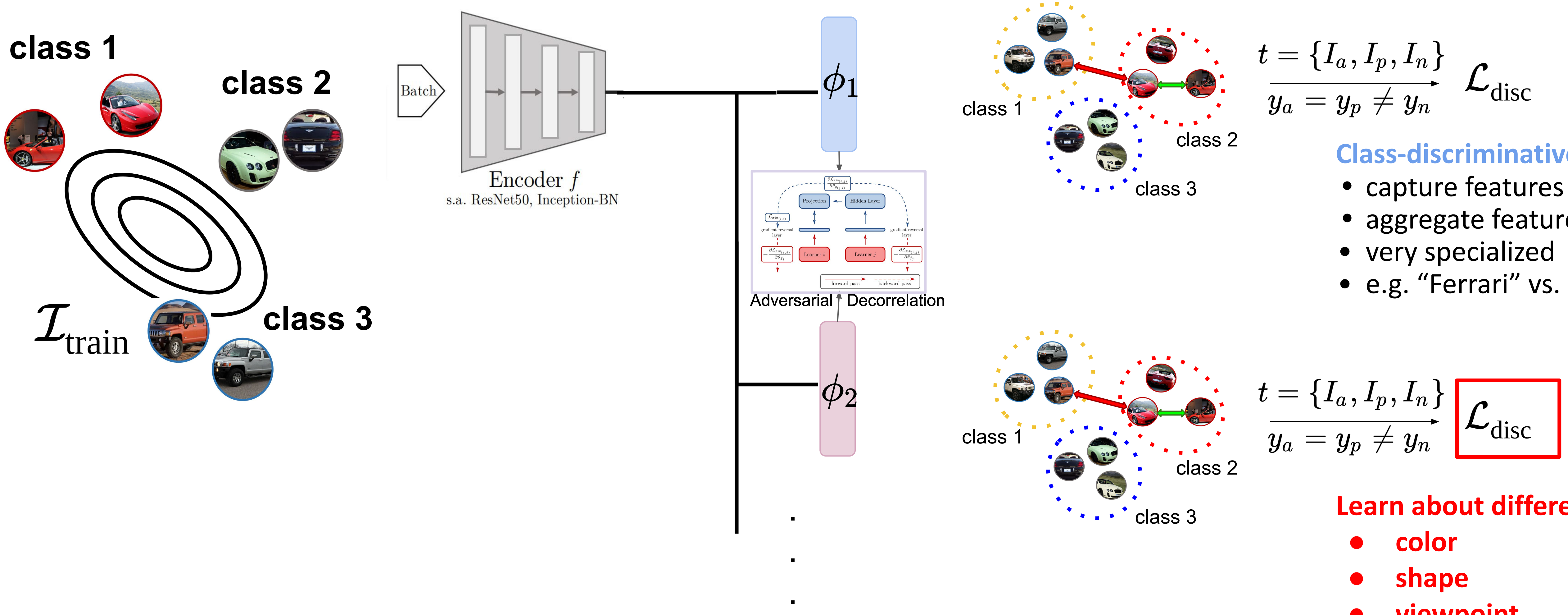
Class-discriminative features

- capture features **separating between classes**
- aggregate features into ‘classes’
- very specialized
- e.g. “Ferrari” vs. “Hummer”

$$\xrightarrow[y_a = y_p \neq y_n]{t = \{I_a, I_p, I_n\}} \boxed{\mathcal{L}_{\text{disc}}}$$

Ensemble-based DML

- **Assumption**: Different features improve robustness to OOD data (new classes, etc.)



Class-discriminative features

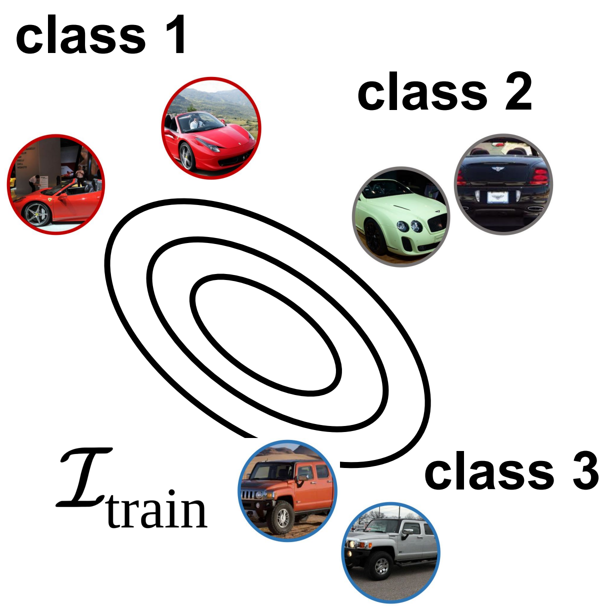
- capture features **separating between classes**
- aggregate features into 'classes'
- very specialized
- e.g. "Ferrari" vs. "Hummer"

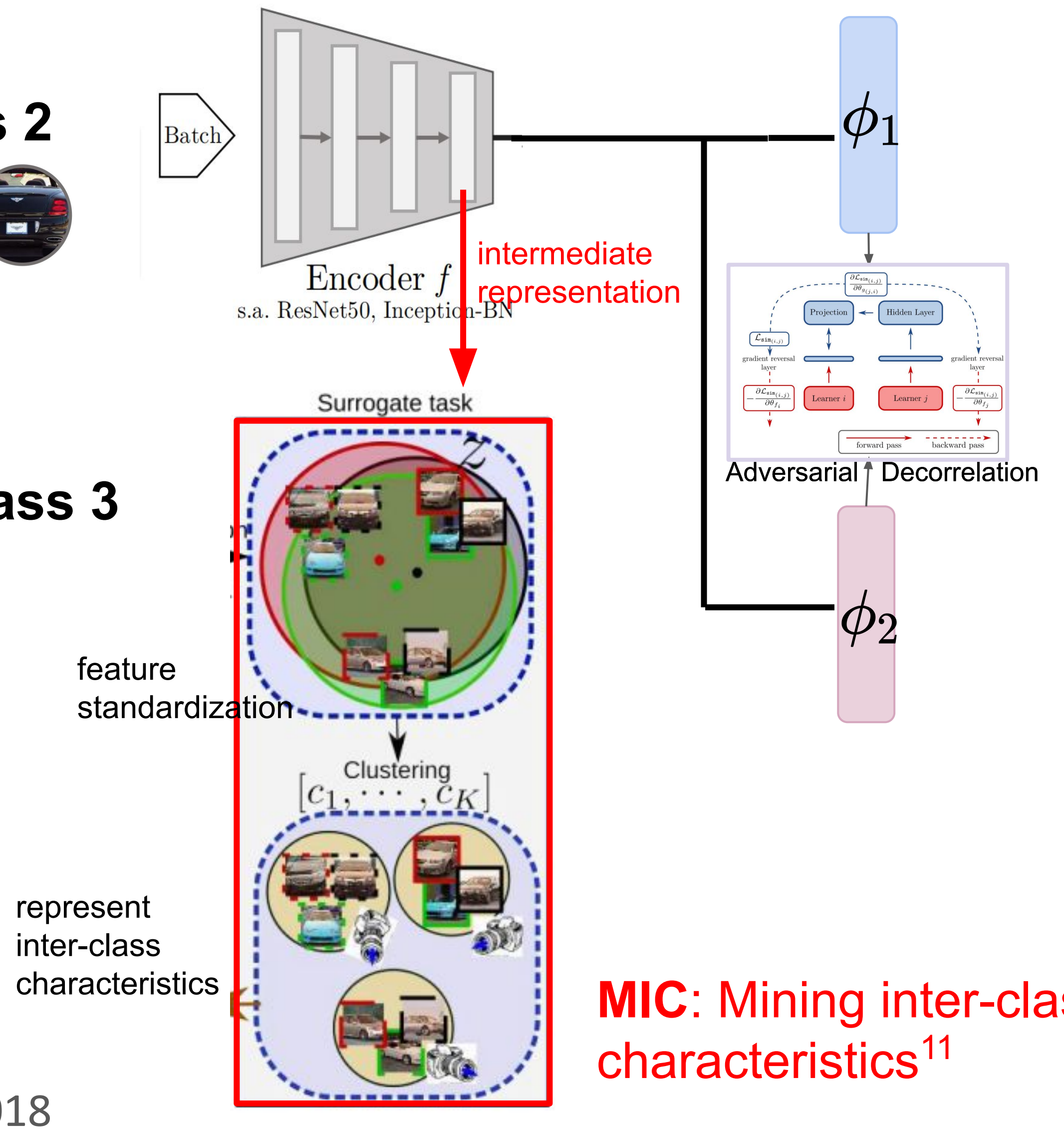
Learn about different/more general features?

- **color**
- **shape**
- **viewpoint**
- **....**

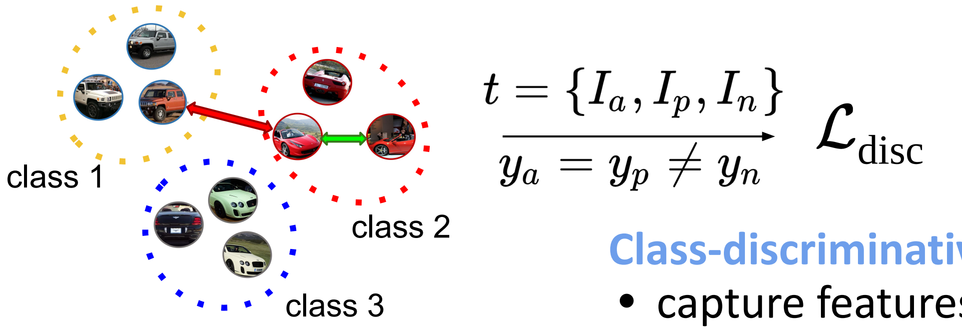
Ensemble-based DML

- **Assumption**: Different features improve robustness to OOD data (new classes, etc.)
- Only class-labels available: use **unsupervised learning**





MIC: Mining inter-class characteristics[11]



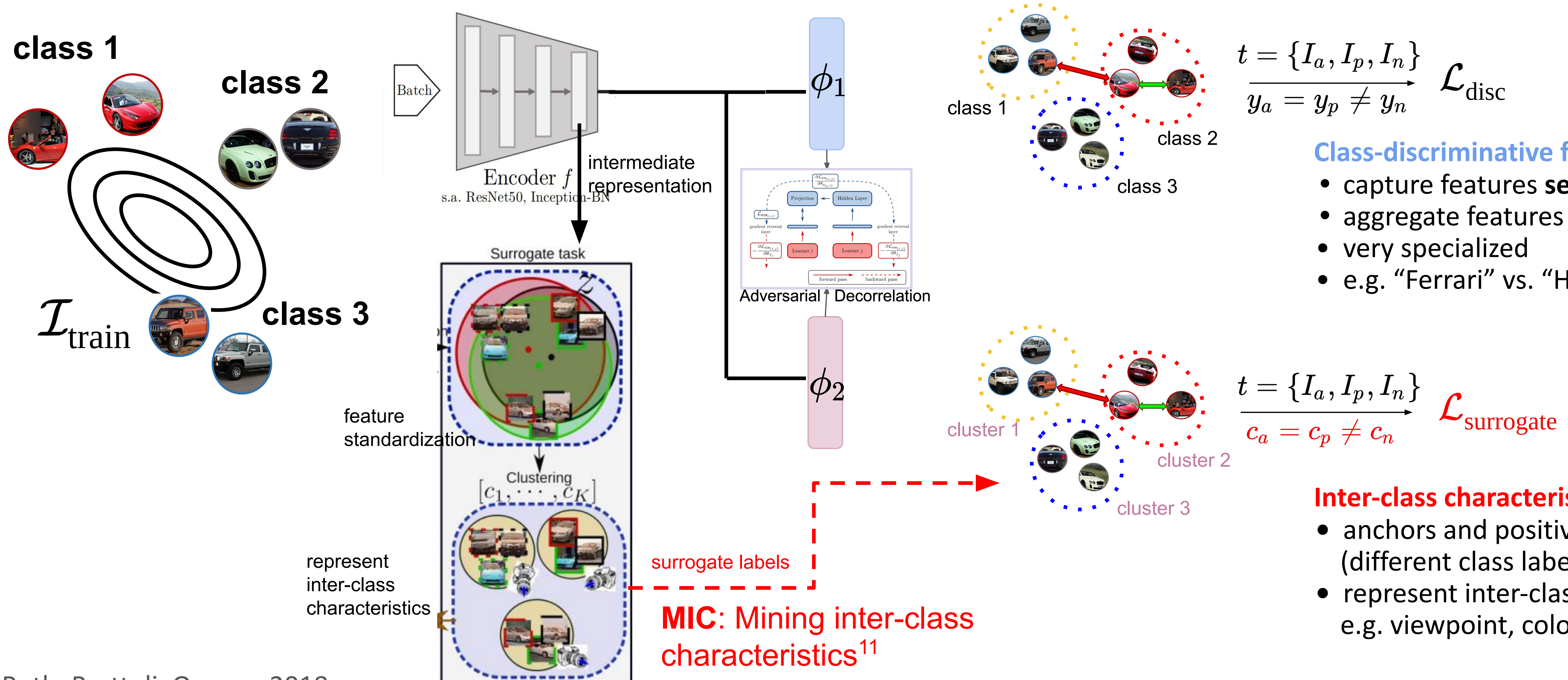
Class-discriminative features

- capture features **separating between classes**
- aggregate features into ‘classes’
- very specialized
- e.g. “Ferrari” vs. “Hummer”

[11]Roth, Brattoli, Ommer 2018

Ensemble-based DML

- **Assumption**: Different features improve robustness to OOD data (new classes, etc.)
- Only class-labels available: use **unsupervised learning**



Class-discriminative features

- capture features **separating between classes**
- aggregate features into 'classes'
- very specialized
- e.g. "Ferrari" vs. "Hummer"

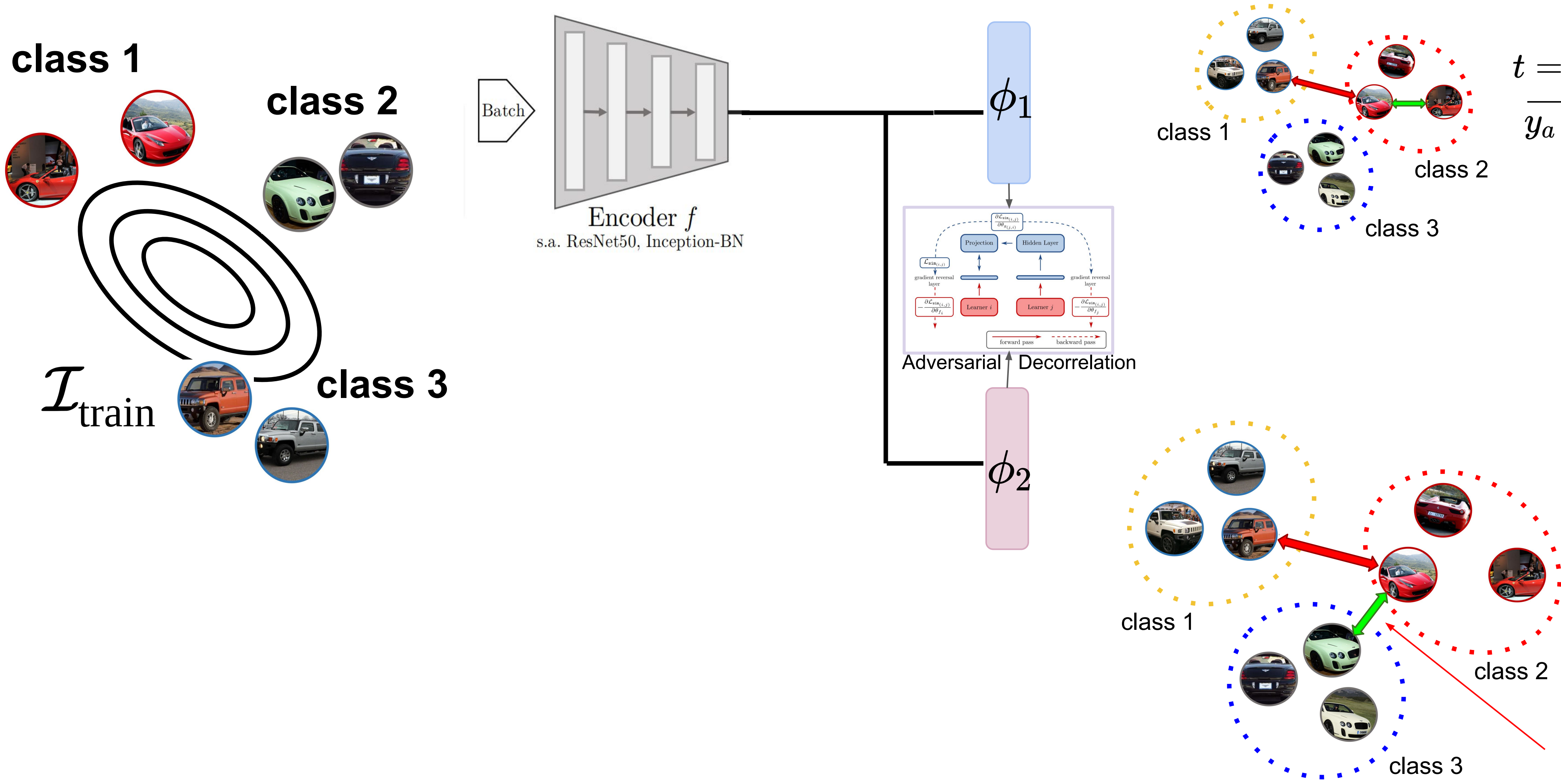
Inter-class characteristics[11]

- anchors and positives **from same cluster** (different class labels!)
- represent inter-class characteristics, e.g. viewpoint, color, etc.

[11]Roth, Brattoli, Ommer 2018

Ensemble-based DML

- **Assumption**: Different features improve robustness to OOD data (new classes, etc.)
- Only class-labels available: use **unsupervised learning**



$$t = \{I_a, I_p, I_n\} \xrightarrow[y_a = y_p \neq y_n]{} \mathcal{L}_{\text{disc}}$$

Class-discriminative features
- capture features **separating between classes**
- aggregate features into 'classes'
- very specialized
- e.g. "Ferrari" vs. "Hummer"

Class-shared features[12]
- anchors and positives **from different classes**
- represent **shared, general object** characteristics

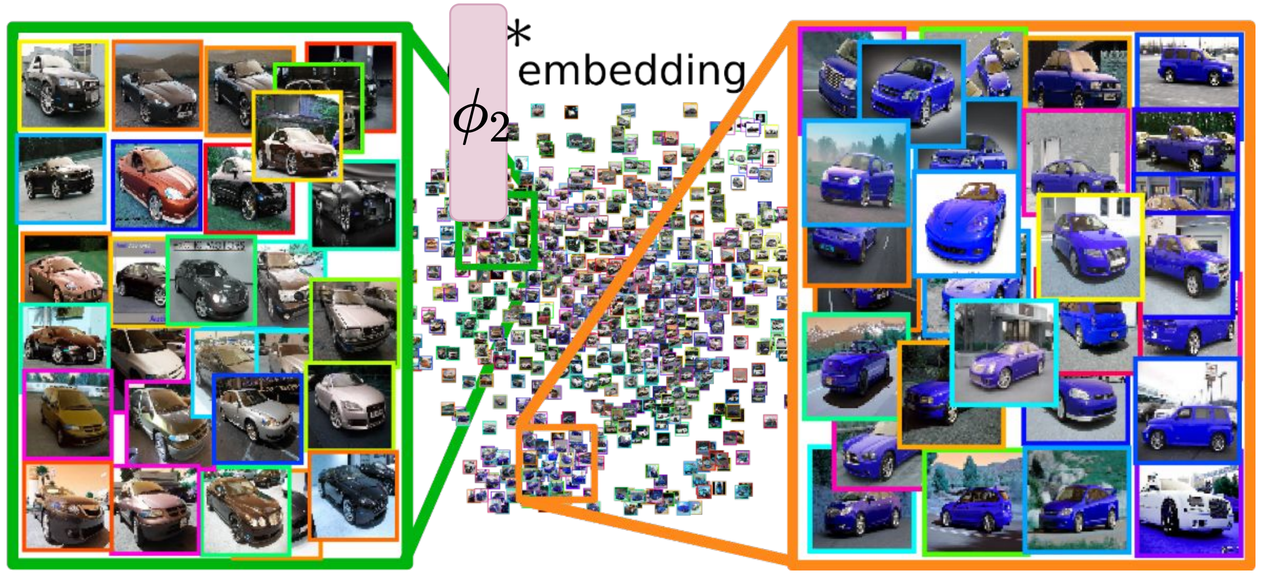
$$t = \{I_a, I_p, I_n\} \xrightarrow[y_a \neq y_p \neq y_n]{} \mathcal{L}_{\text{shared}}$$

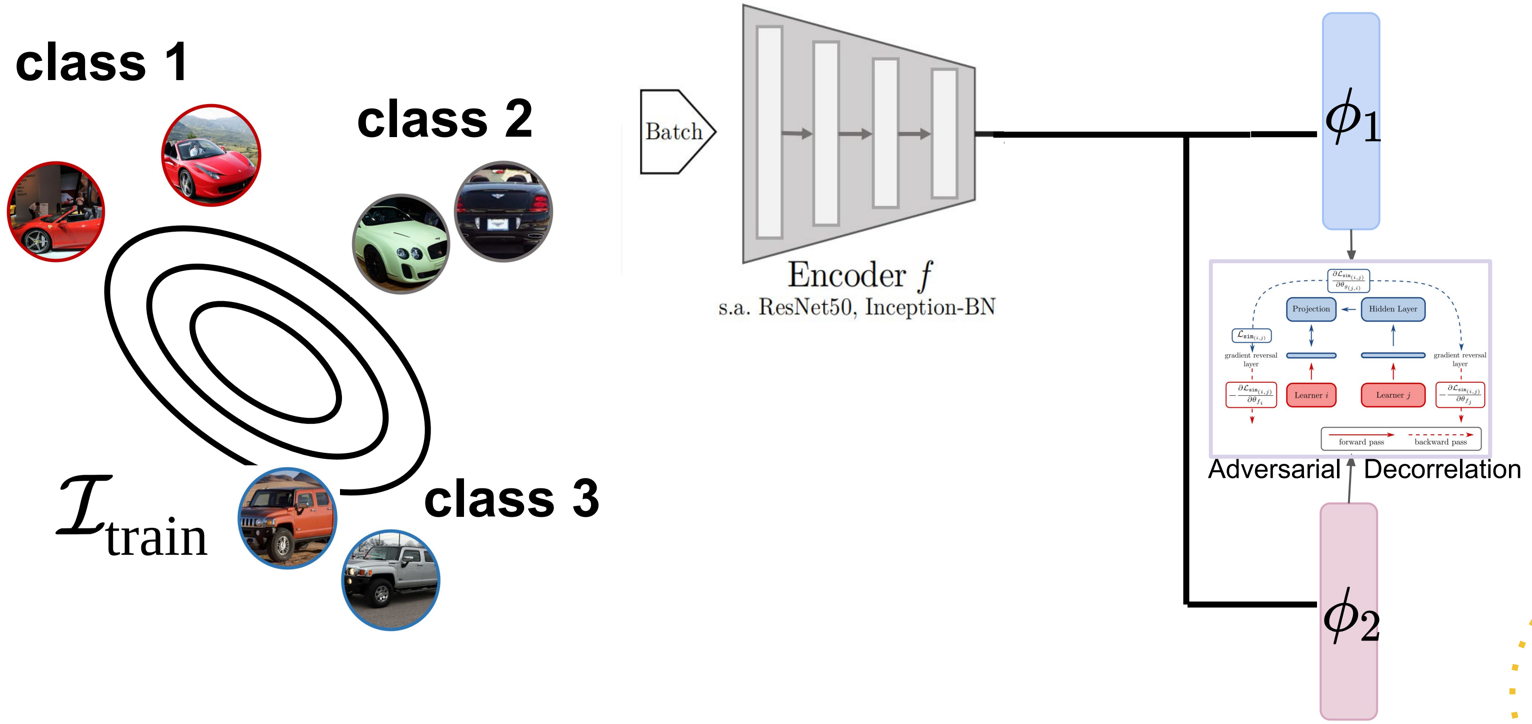
- Explicitlt sample anchor and positive from **DIFFERENT** classes!
- Can be achieved by simply changing triplet sampling

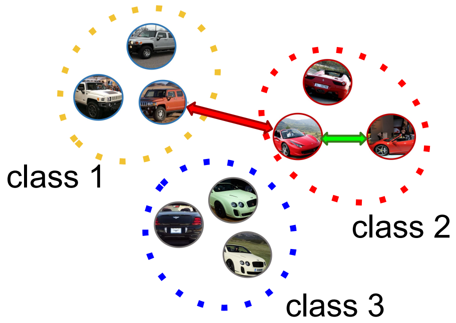
[12]Milbich, Roth, Brattoli, Ommer 2020

Ensemble-based DML

- **Assumption**: Different features improve robustness to OOD data (new classes, etc.)
- Only class-labels available: use **unsupervised learning**



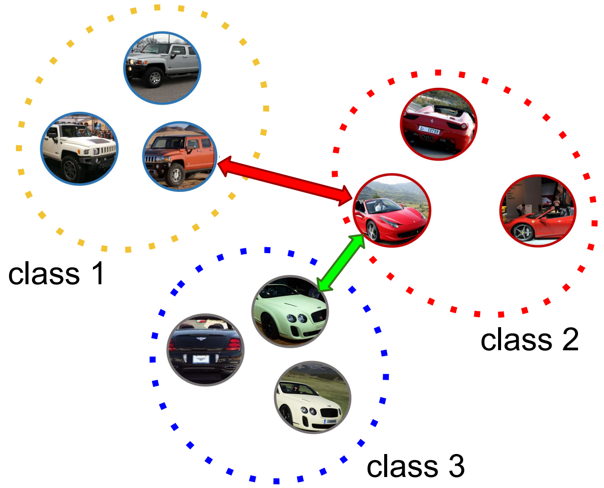




$$t = \{I_a, I_p, I_n\} \xrightarrow[y_a = y_p \neq y_n]{} \mathcal{L}_{\text{disc}}$$

Class-discriminative features

- capture features **separating between classes**
- aggregate features into ‘classes’
- very specialized
- e.g. “Ferrari” vs. “Hummer”



Class-shared features[12]

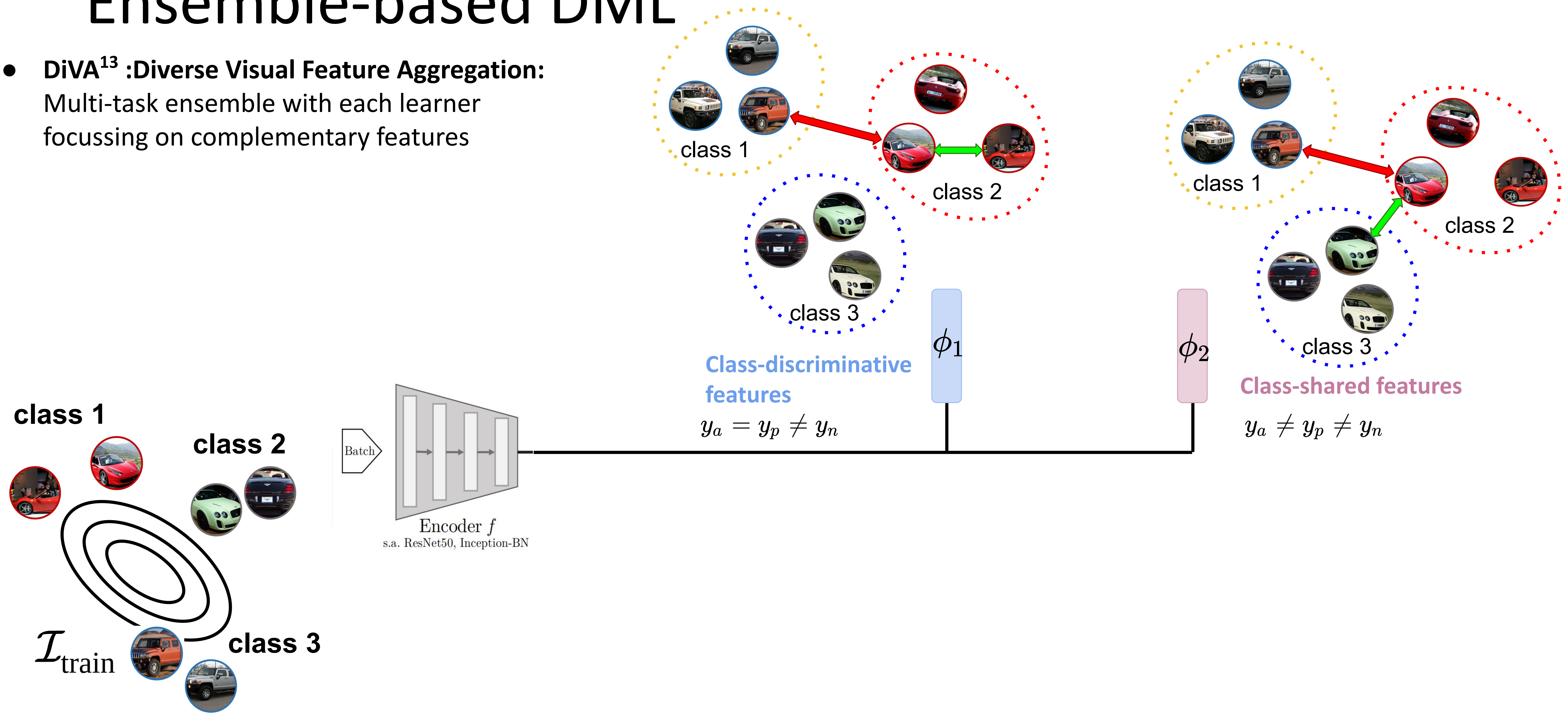
- anchors and positives **from different classes**
- represent **shared, general object** characteristics

$$t = \{I_a, I_p, I_n\} \xrightarrow[y_a \neq y_p \neq y_n]{} \mathcal{L}_{\text{shared}}$$

[12] Milbich, Roth, Brattoli, Ommer 2020

Ensemble-based DML

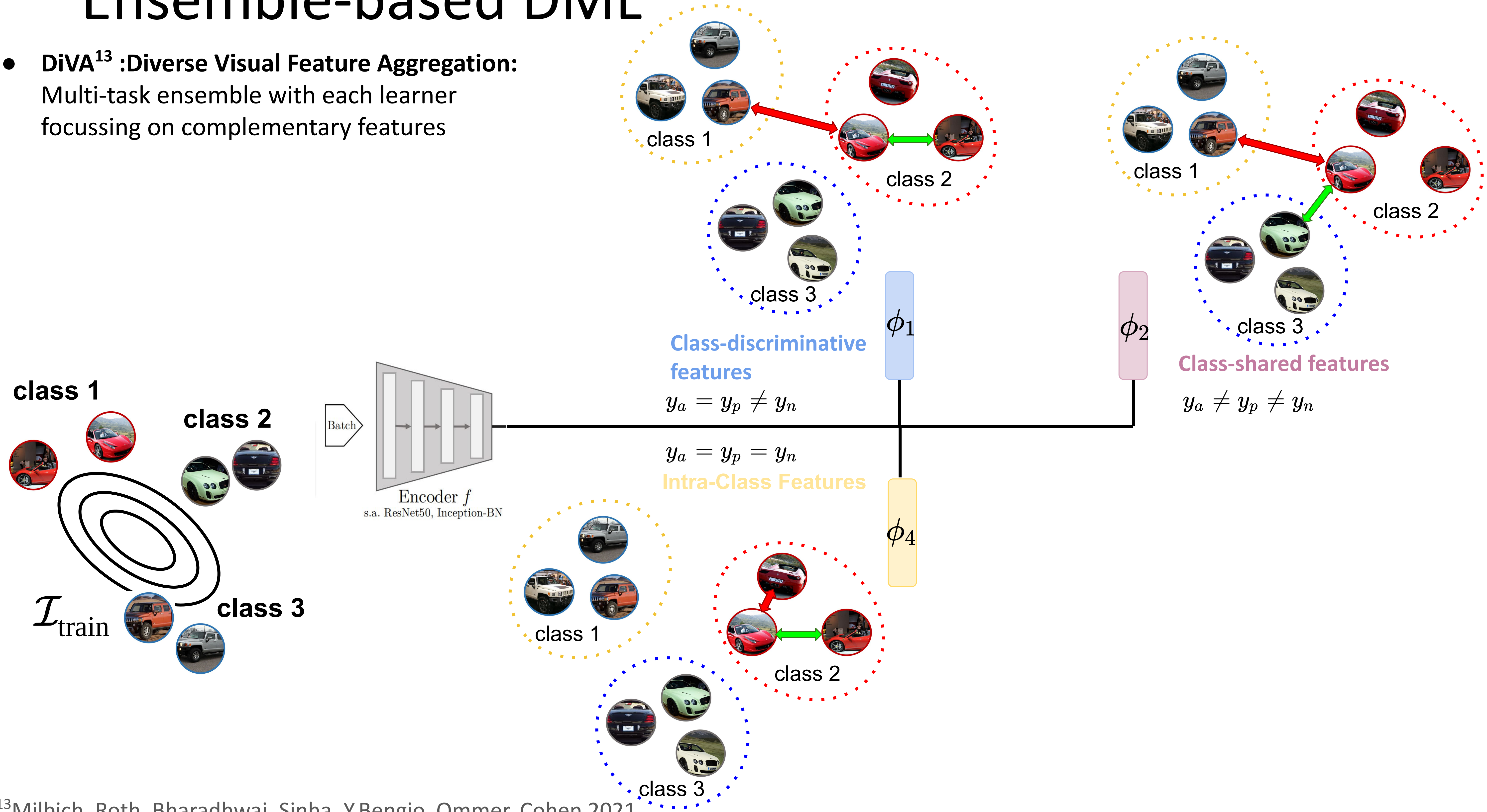
- **DiVA[13] :Diverse Visual Feature Aggregation:** Multi-task ensemble with each learner focussing on complementary features



[13]Milbich, Roth, Bharadhwaj, Sinha, Y.Bengio, Ommer, Cohen 2021

Ensemble-based DML

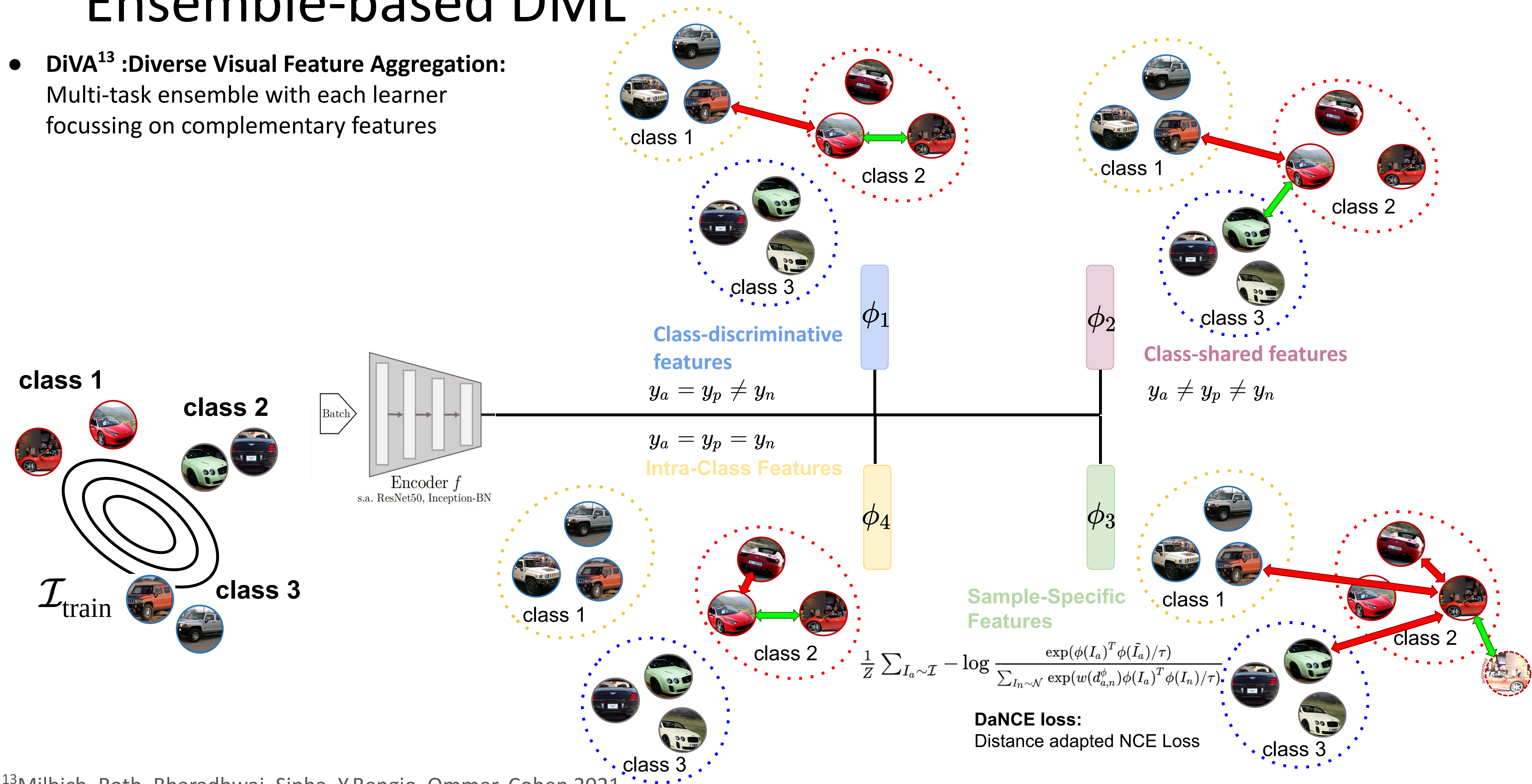
- **DiVA[13] :Diverse Visual Feature Aggregation:** Multi-task ensemble with each learner focussing on complementary features



[13]Milbich, Roth, Bharadhwaj, Sinha, Y.Bengio, Ommer, Cohen 2021

Ensemble-based DML

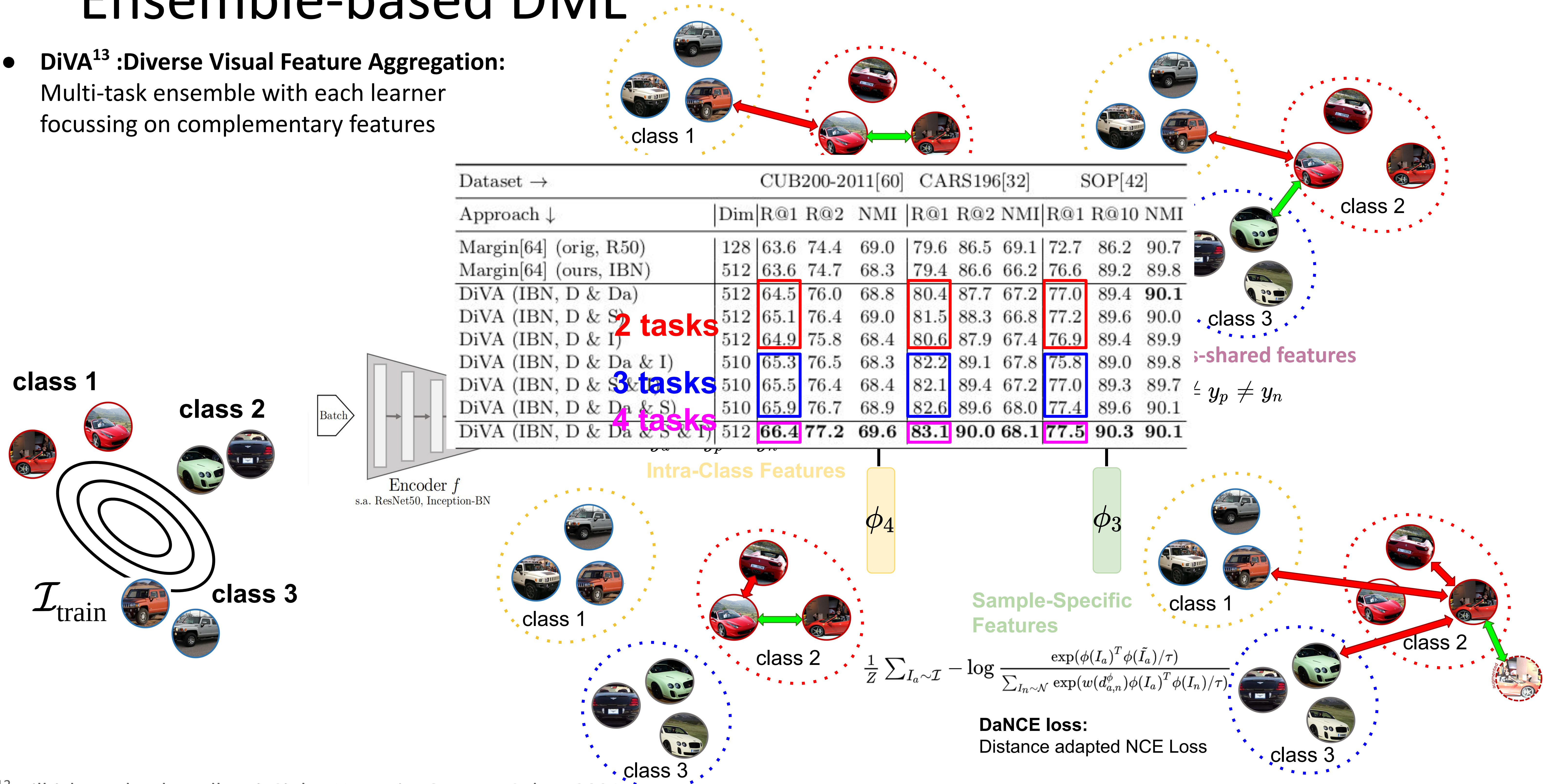
- **DiVA[13] :Diverse Visual Feature Aggregation:** Multi-task ensemble with each learner focussing on complementary features



[13]Milbich, Roth, Bharadhwaj, Sinha, Y.Bengio, Ommer, Cohen 2021

Ensemble-based DML

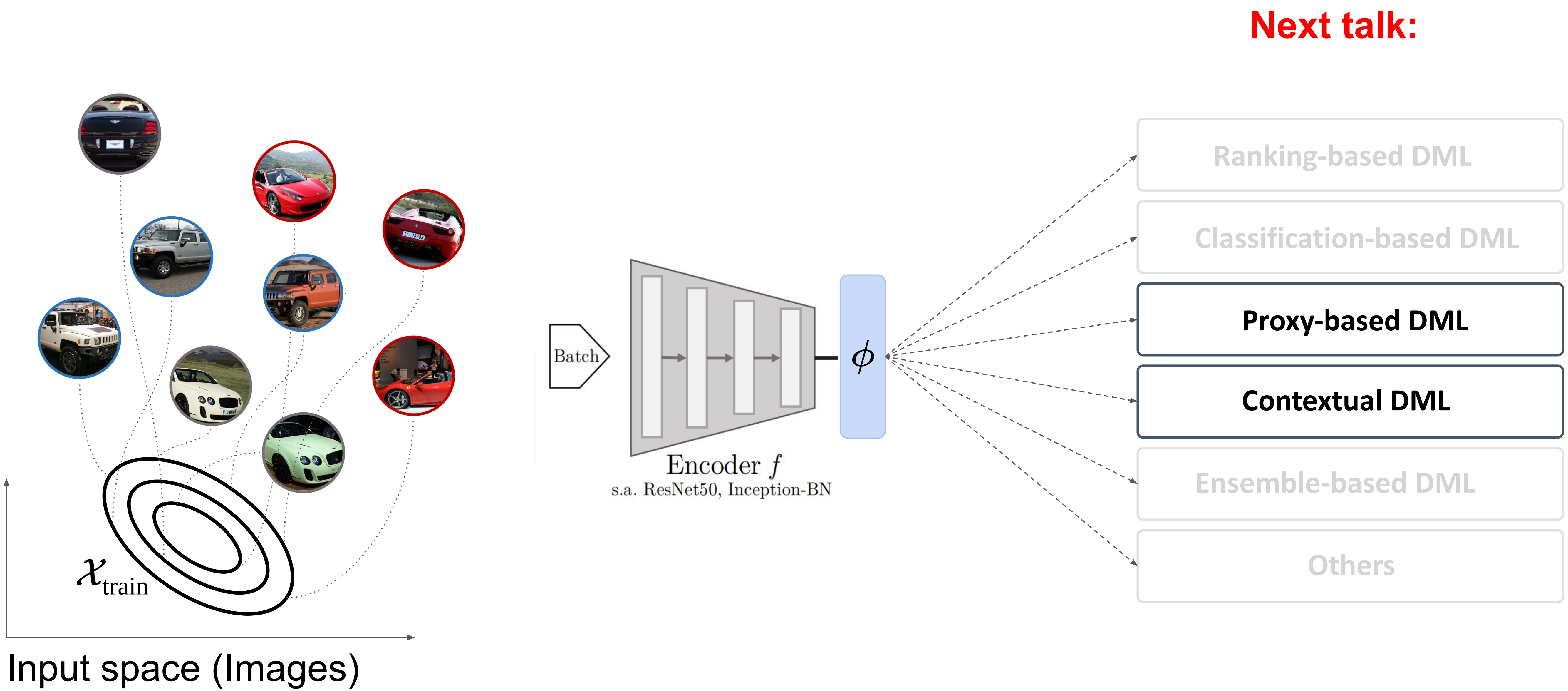
- **DiVA[13] :Diverse Visual Feature Aggregation:** Multi-task ensemble with each learner focussing on complementary features



Dataset →		CUB200-2011[60]			CARS196[32]			SOP[42]		
Approach ↓	Dim	R@1	R@2	NMI	R@1	R@2	NMI	R@1	R@10	NMI
Margin[64] (orig, R50)	128	63.6	74.4	69.0	79.6	86.5	69.1	72.7	86.2	90.7
Margin[64] (ours, IBN)	512	63.6	74.7	68.3	79.4	86.6	66.2	76.6	89.2	89.8
DiVA (IBN, D & Da)	512	64.5	76.0	68.8	80.4	87.7	67.2	77.0	89.4	**90.1**
DiVA (IBN, D & S)	512	65.1	76.4	69.0	81.5	88.3	66.8	77.2	89.6	90.0
DiVA (IBN, D & I)	512	64.9	75.8	68.4	80.6	87.9	67.4	76.9	89.4	89.9
DiVA (IBN, D & Da & I)	510	65.3	76.5	68.3	82.2	89.1	67.8	75.8	89.0	89.8
DiVA (IBN, D & S & I)	510	65.5	76.4	68.4	82.1	89.4	67.2	77.0	89.3	89.7
DiVA (IBN, D & Da & S)	510	65.9	76.7	68.9	82.6	89.6	68.0	77.4	89.6	90.1
DiVA (IBN, D & Da & S & I)	512	**66.4**	**77.2**	**69.6**	**83.1**	**90.0**	**68.1**	**77.5**	**90.3**	**90.1**

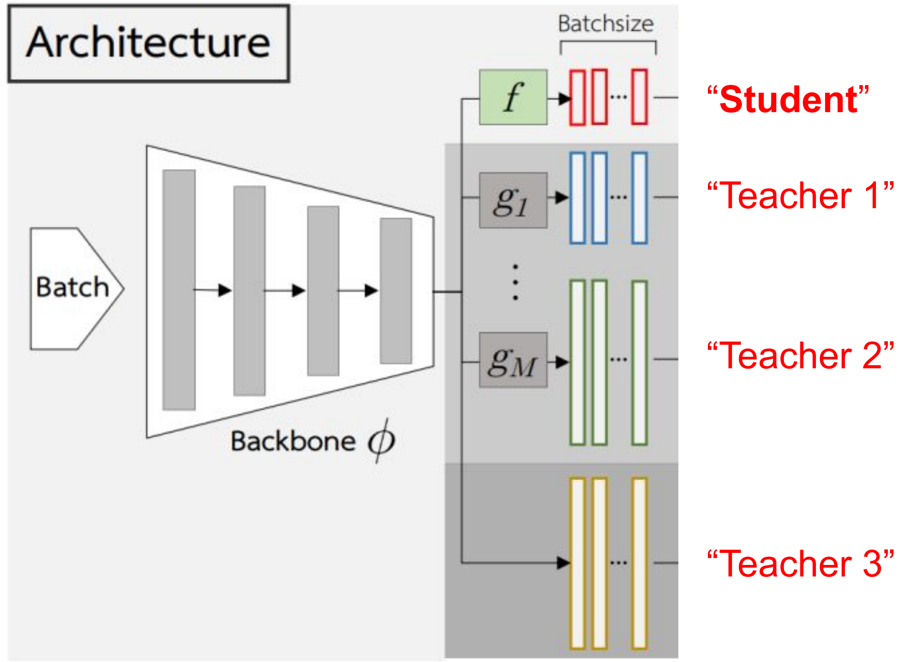
[13]Milbich, Roth, Bharadhwaj, Sinha, Y.Bengio, Ommer, Cohen 2021

Deep Metric Learning (DML)



Ensemble-based DML

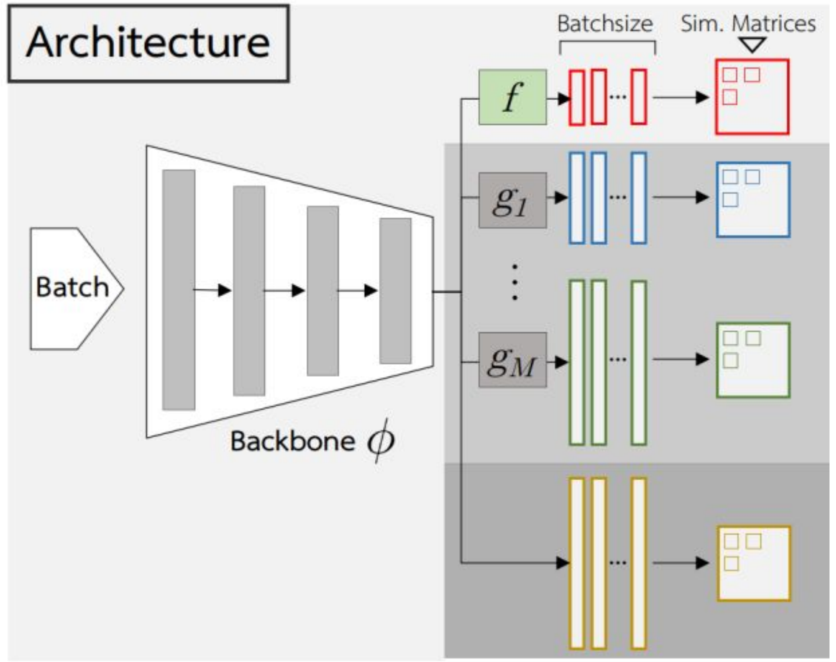
- **S2SD**[14]: Learn ensemble of teachers and distill their ‘knowledge’ into student



[14]Roth, Milbich, Ommer, Cohen, Ghassemi 2021

Ensemble-based DML

- **S2SD**[14]: Learn ensemble of teachers and distill their 'knowledge' into student



"**Student**"

"Teacher 1"

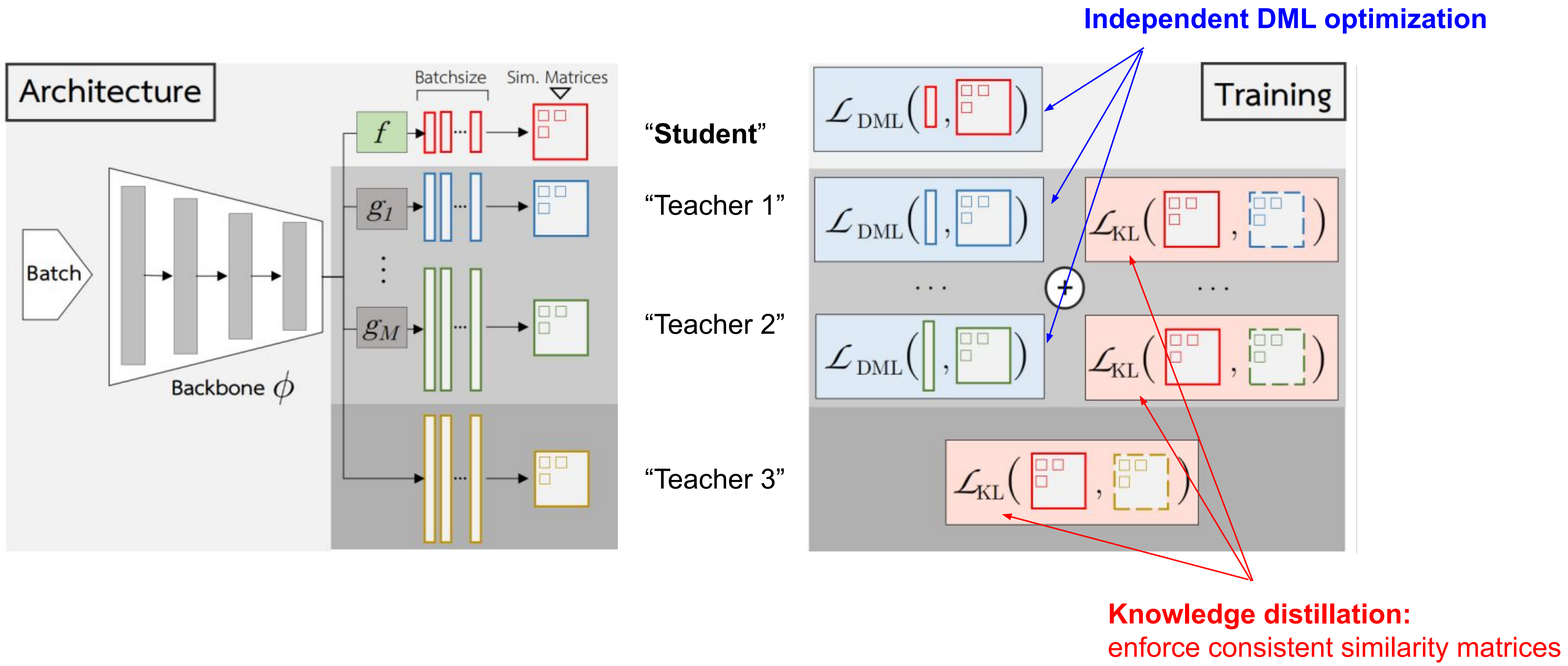
"Teacher 2"

"Teacher 3"

[14]Roth, Milbich, Ommer, Cohen, Ghassemi 2021

Ensemble-based DML

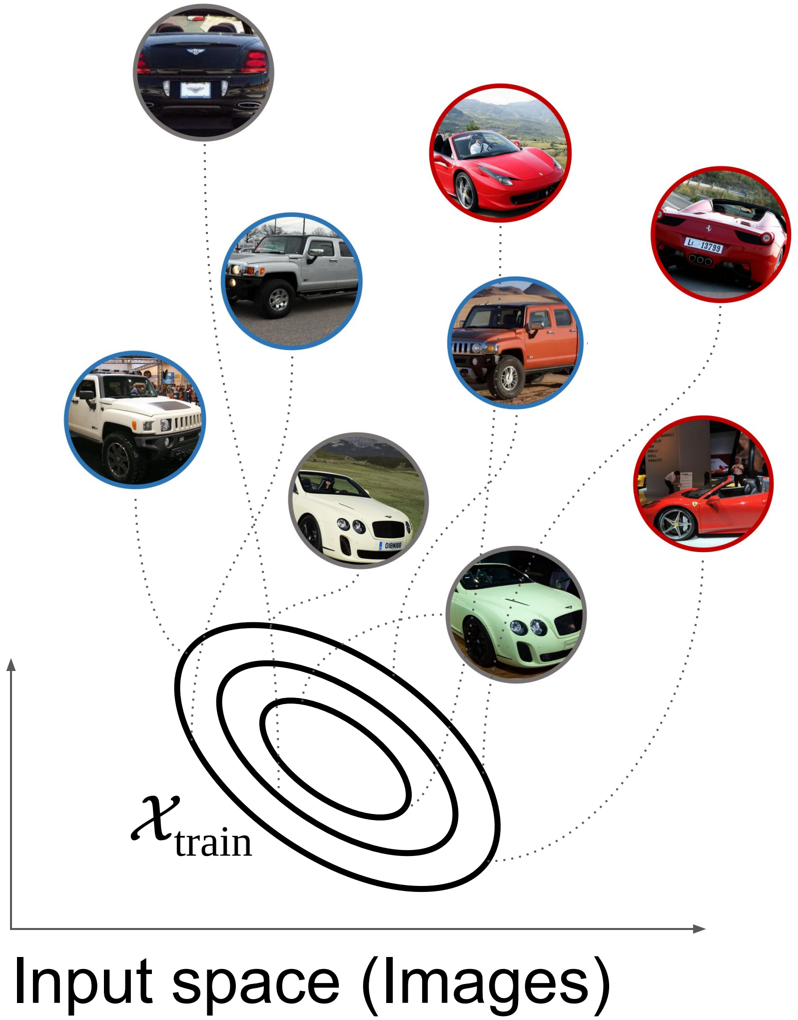
- **S2SD**[14]: Learn **ensemble of teachers** and distill their ‘knowledge’ into student
- use different embedding dimensionalities for teachers to increase robustness of student



[14]Roth, Milbich, Ommer, Cohen, Ghassemi 2021

Visual Similarity Learning

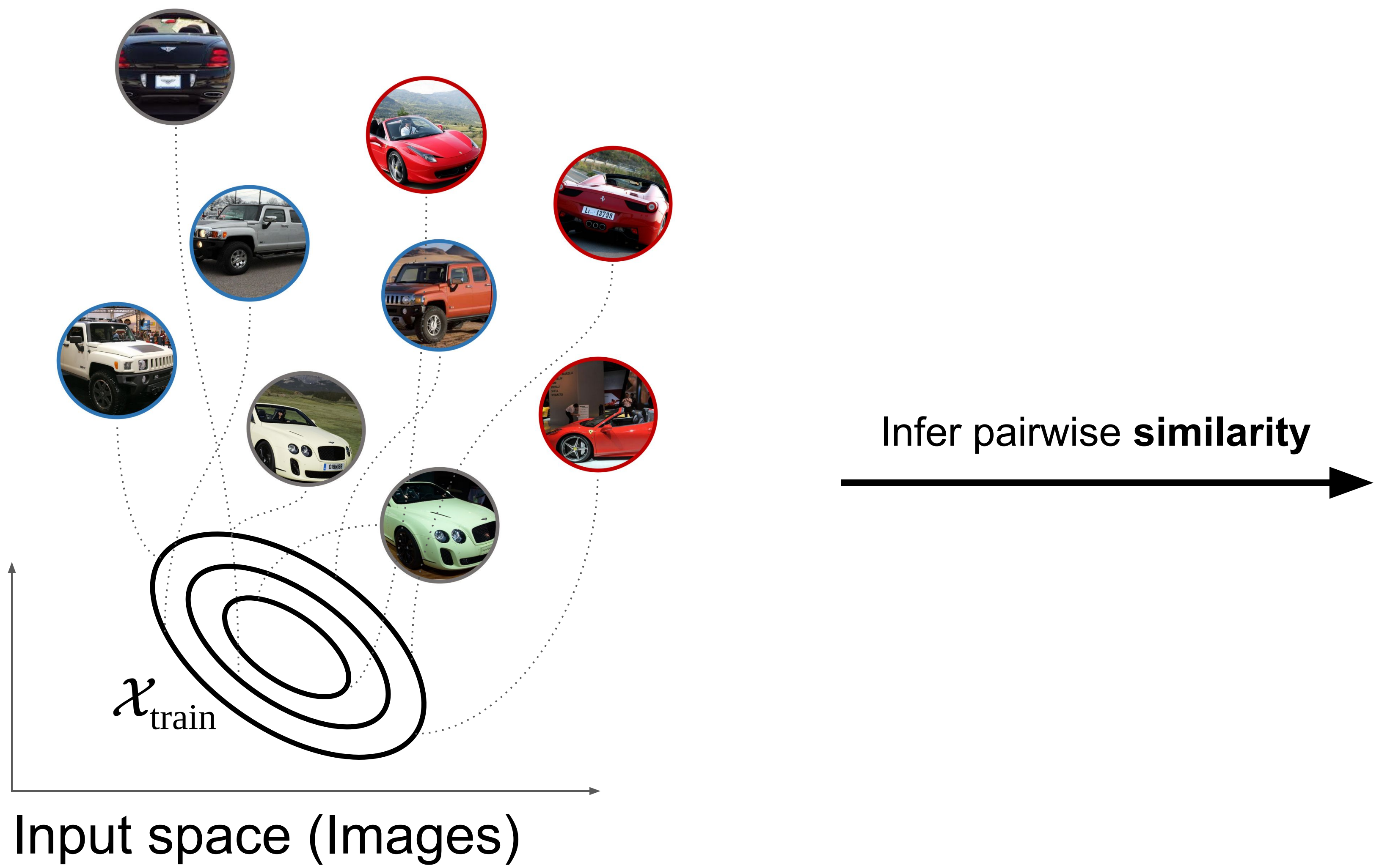
Given data distribution $\mathcal{X}_{\text{train}}$, infer pairwise **semantic similarity**.



Infer pairwise **similarity**

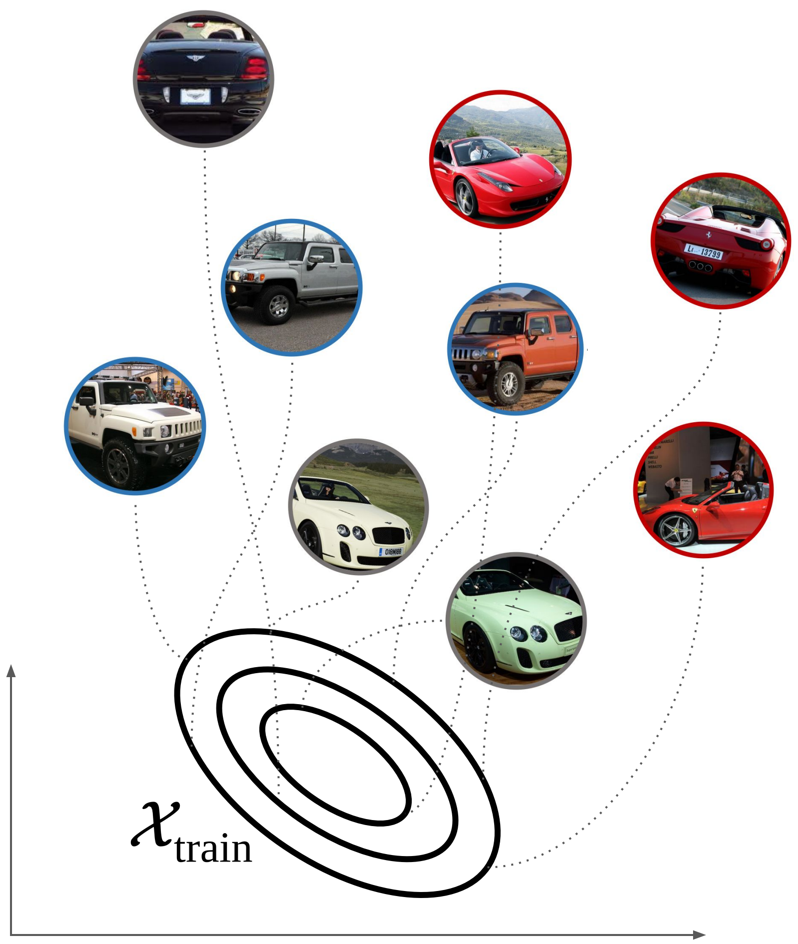
Visual Similarity Learning

Given data distribution $\mathcal{X}_{\text{train}}$, infer pairwise **semantic similarity**.



Visual Similarity Learning

Given data distribution $\mathcal{X}_{\text{train}}$, infer pairwise **semantic similarity**.



Input space (Images)

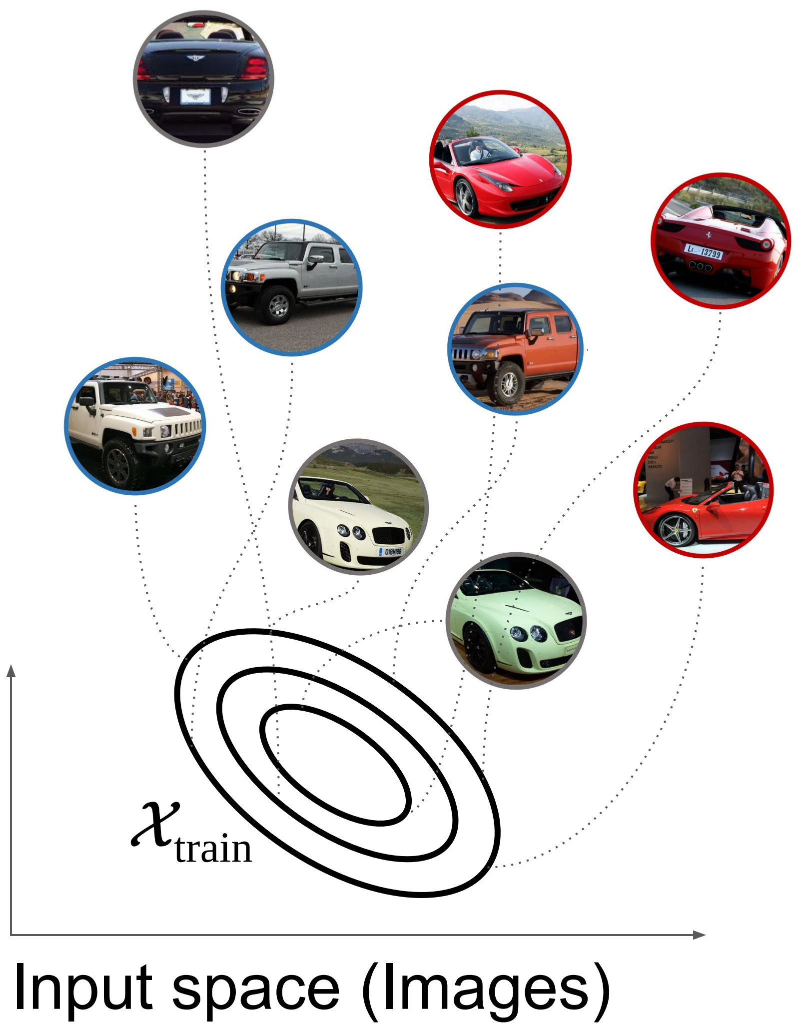
Infer pairwise **similarity**

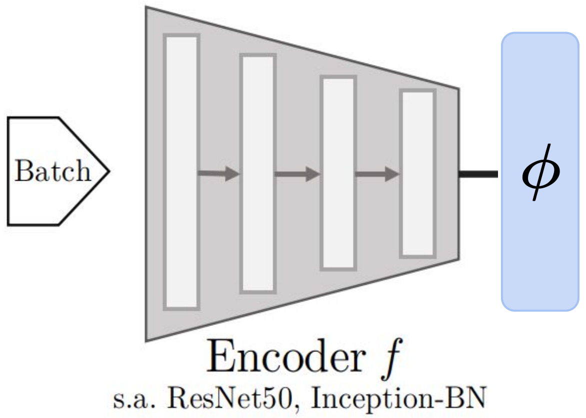
indicated by **class labels**

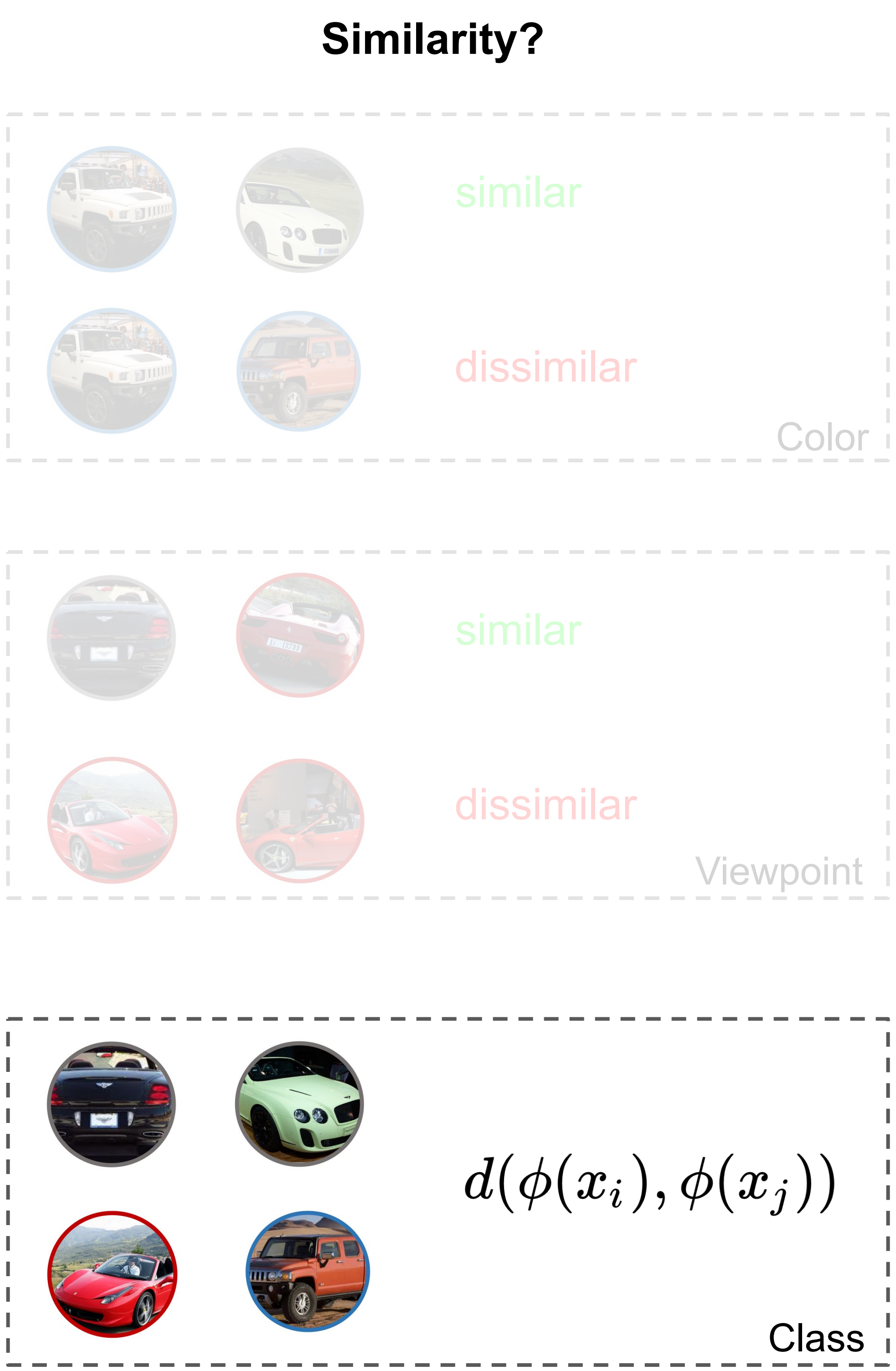
Similarity?

Visual Similarity Learning

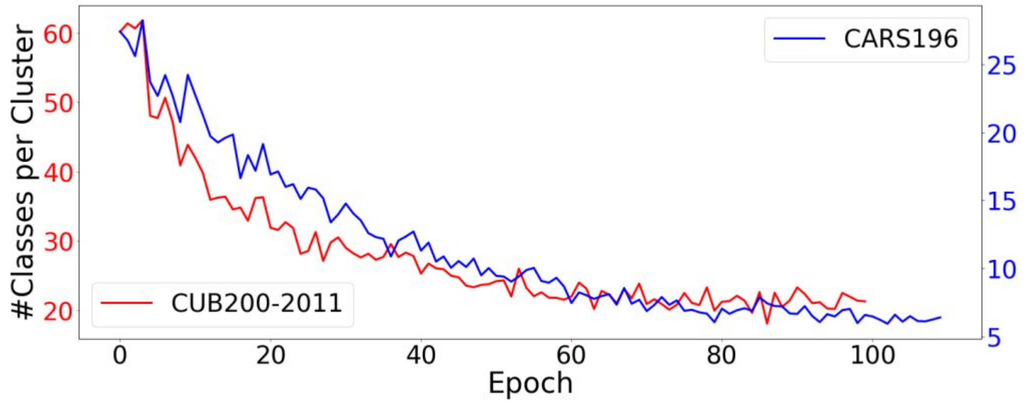
Learn **representation** $\phi(x)$ which reflects **semantic similarity** $d(\phi(x_i), \phi(x_j))$ within training distribution $\mathcal{X}_{\text{train}}$.



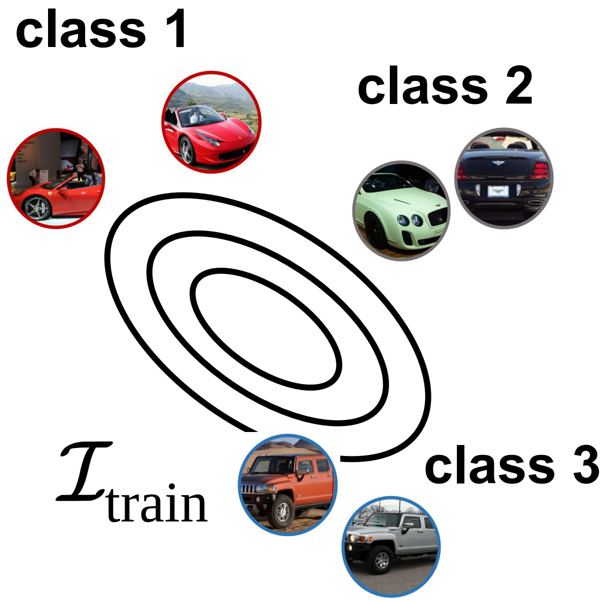


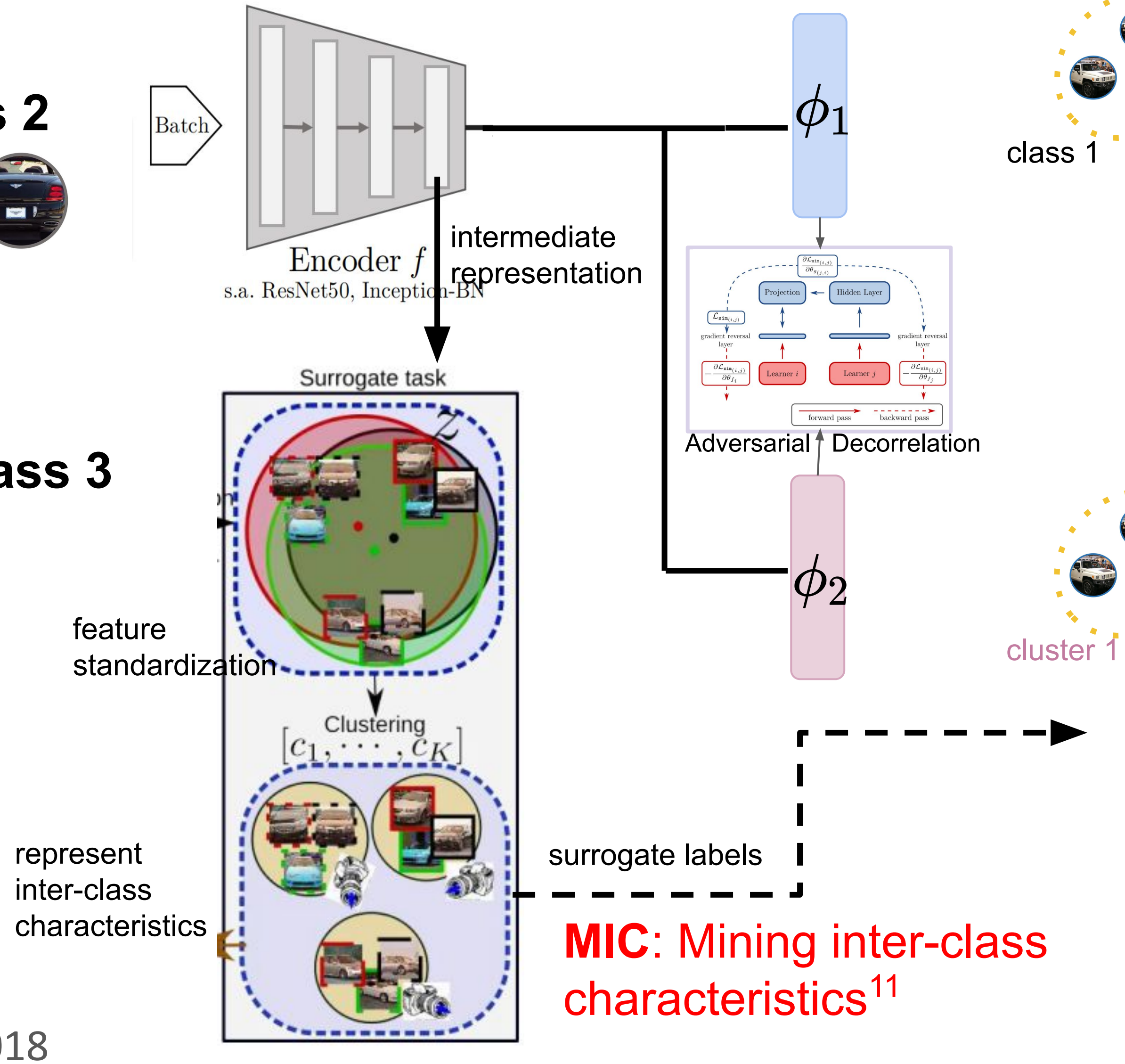


Ensemble-based DML

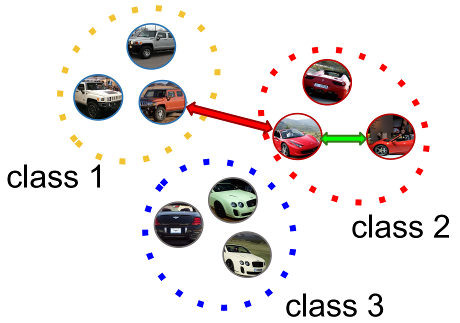


- Typically there are **different notions of similarity** (classes, color, viewpoint, …)
- **Assumption**: Different features improve robustness to OOD data (new classes, etc.)
- **Explicit** similarity specialization of learners
- Only class-labels available: use **unsupervised learning**





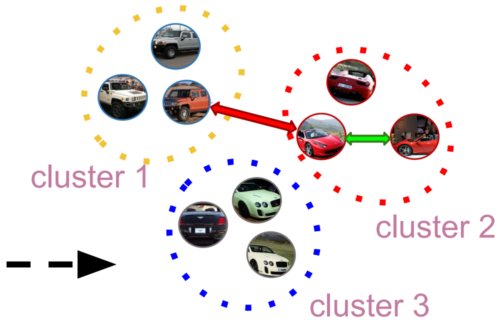
MIC: Mining inter-class characteristics[11]



$$t = \{I_a, I_p, I_n\} \xrightarrow[y_a = y_p \neq y_n]{} \mathcal{L}_{\text{disc}}$$

Class-discriminative features

- capture features **separating between classes**
- aggregate features into 'classes'
- very specialized
- e.g. "Ferrari" vs. "Hummer"



$$t = \{I_a, I_p, I_n\} \xrightarrow[c_a = c_p \neq c_n]{} \mathcal{L}_{\text{surrogate}}$$

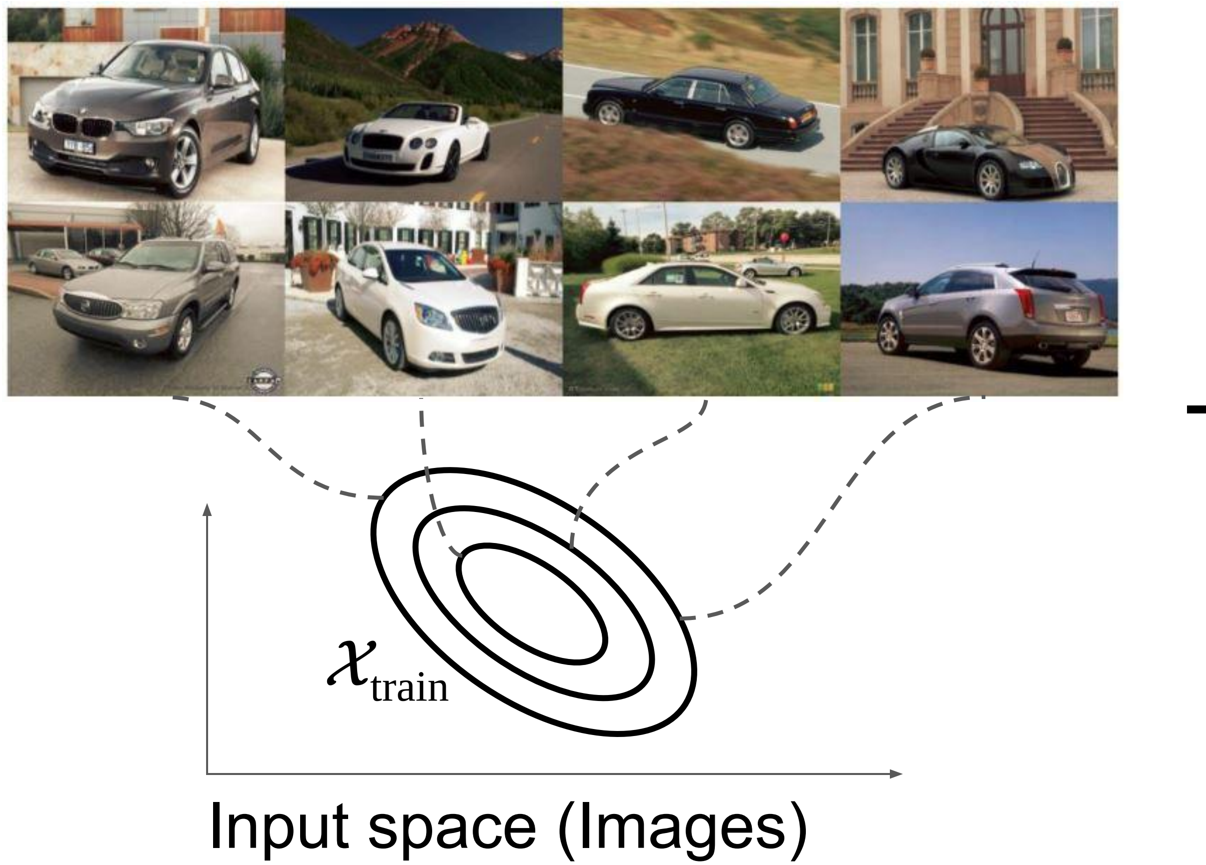
Inter-class characteristics[11]

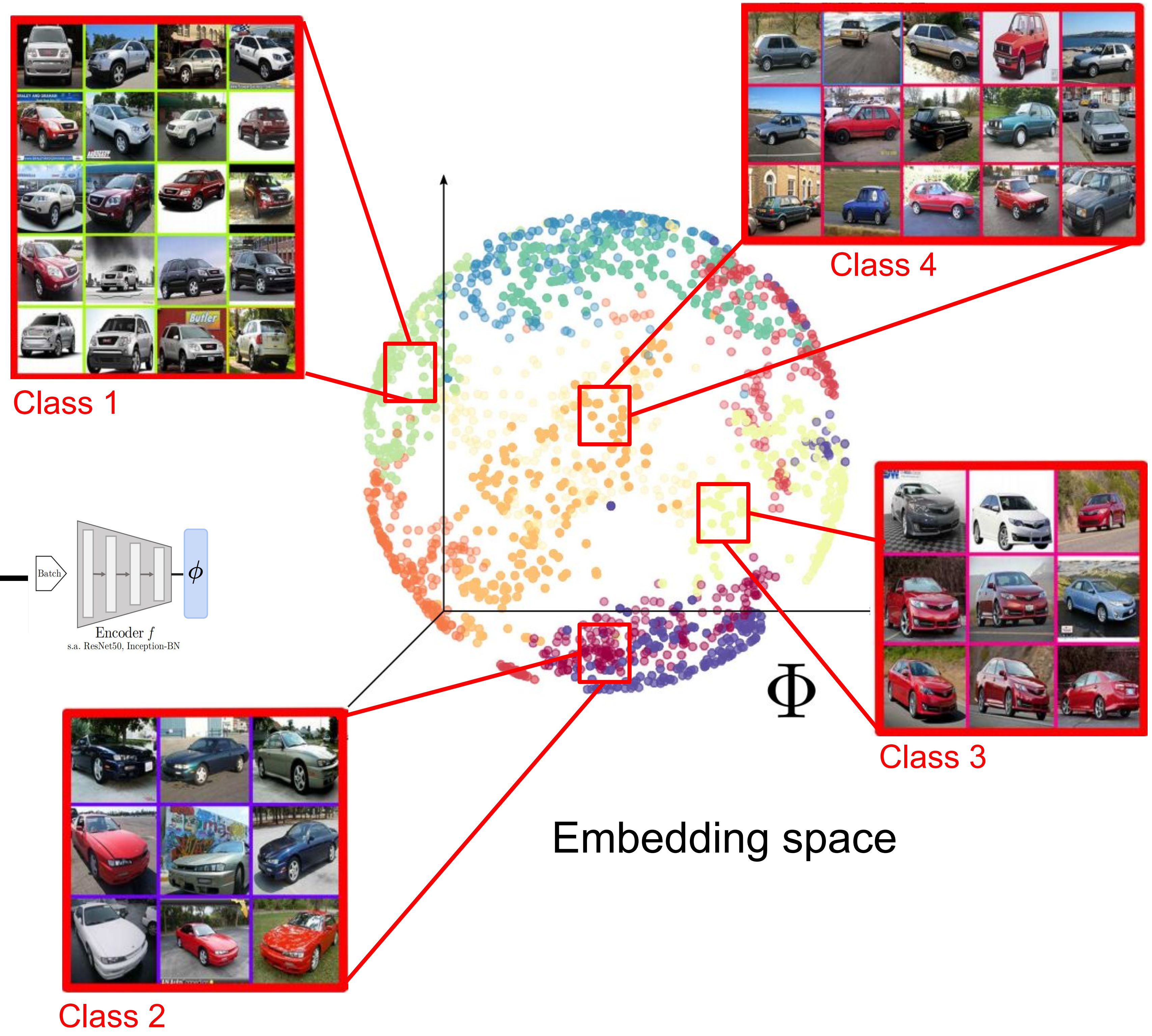
- anchors and positives **from same cluster** (different class labels!)
- represent inter-class characteristics, e.g. viewpoint, color, etc.

[11] Roth, Brattoli, Ommer 2018

Learning Visual Representations

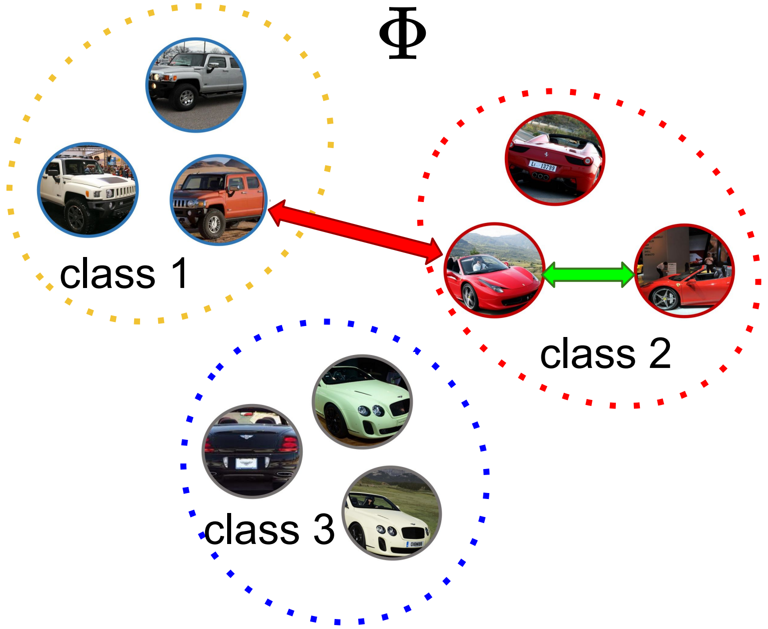
Learn **representation** Φ which reflects **semantic similarity** within training distribution $\mathcal{X}_{\text{train}}$.

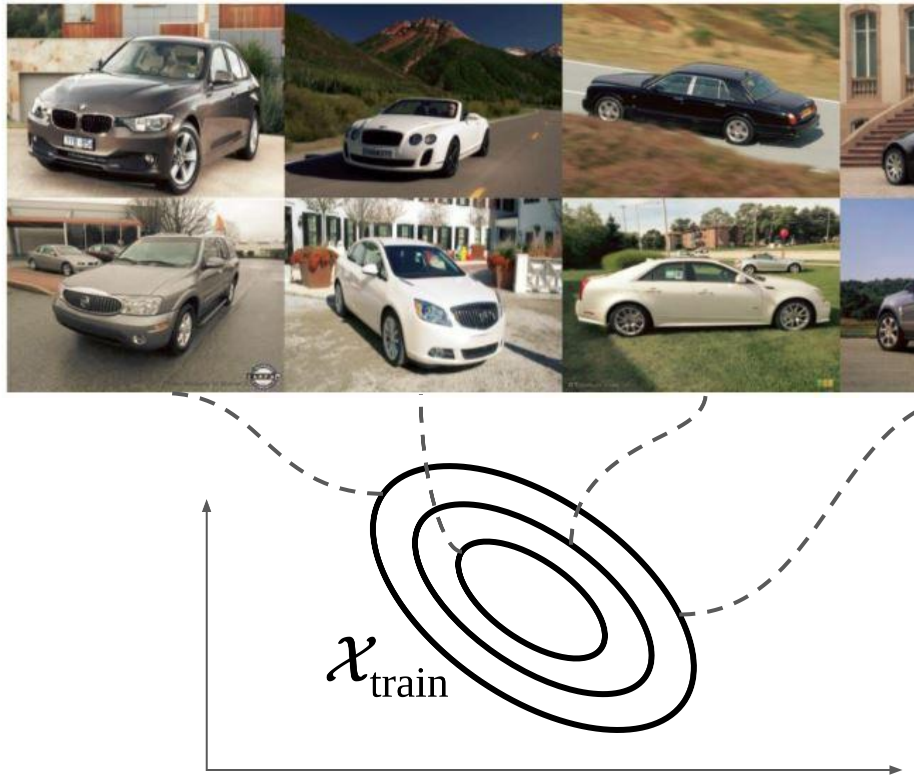




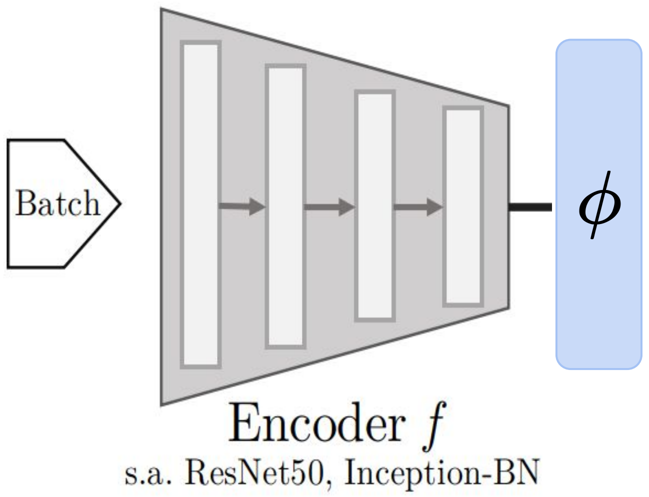
Learning Visual Representations

Learn **representation** $\phi(x)$ which reflects **semantic similarity** within training distribution $\mathcal{X}_{\text{train}}$.





Input space (Images)



Deep Metric Learning (DML)

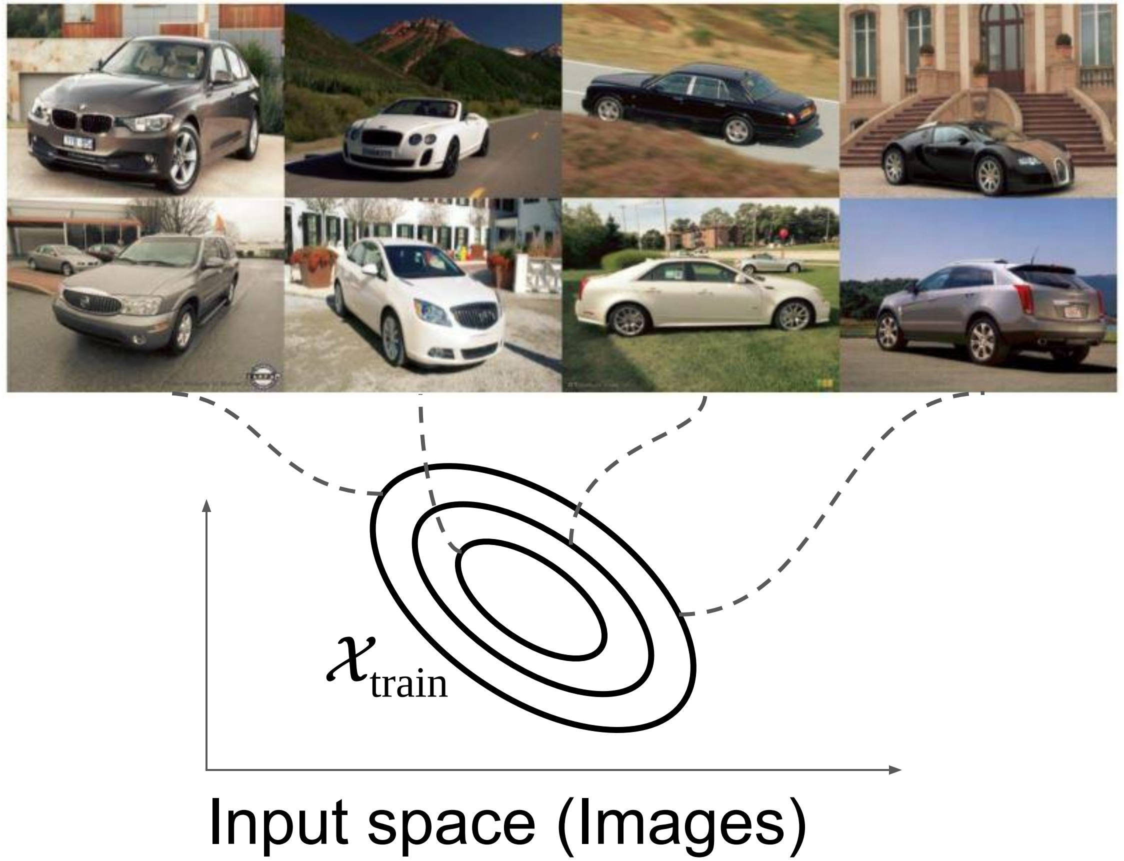
Classification

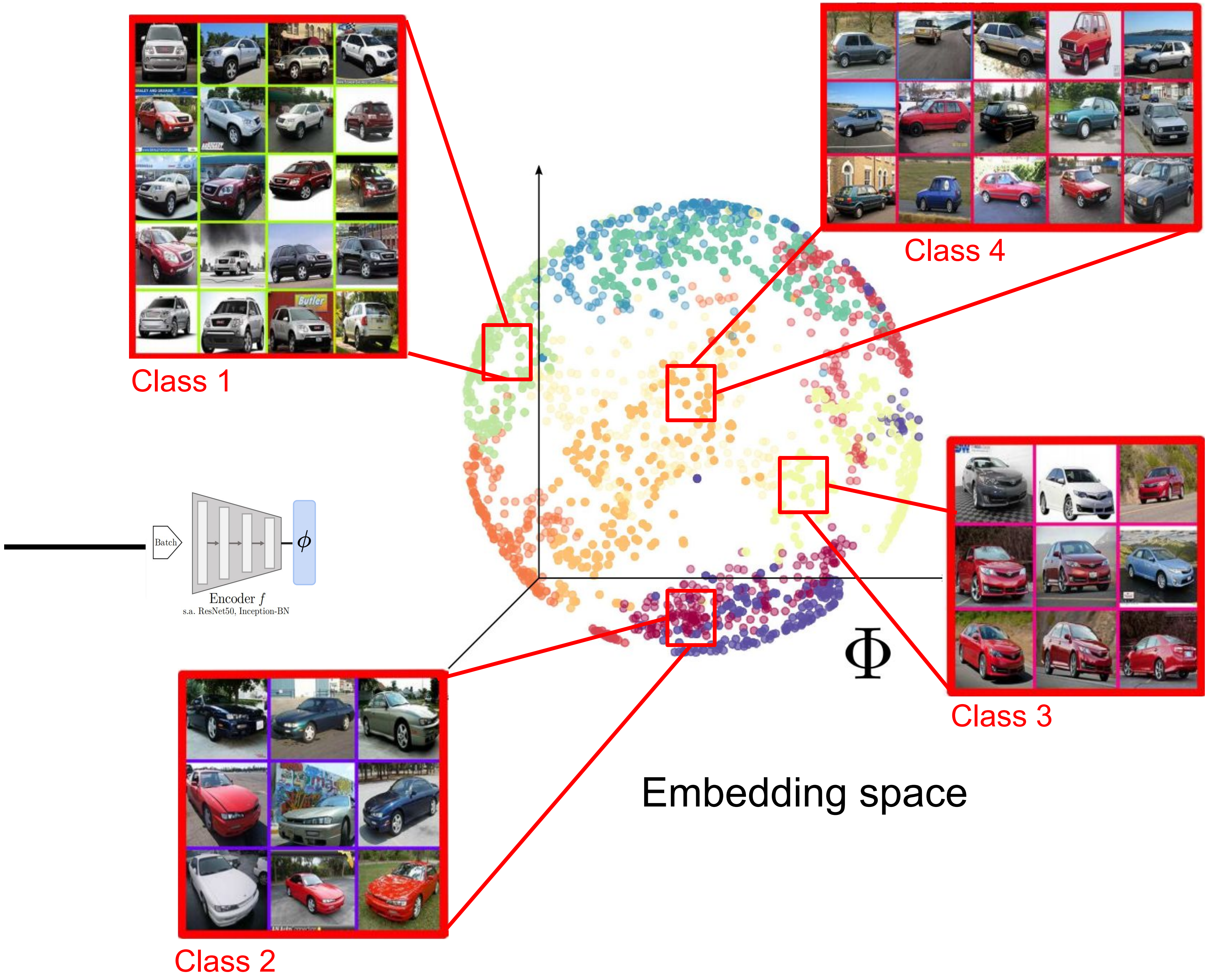
Surrogate Tasks

Representation Learning

Learning Visual Representations

Learn **representation** Φ which reflects **semantic similarity** within training distribution $\mathcal{X}_{\text{train}}$.





Introduction DML

Model Architecture

- Backbone architecture
- Embedding dimensionality
- Basic Regularization
- Batchsize
- etc.

Objective Function

- Formulation:
  - Ranking-based
  - Proxy-based
  - Classification-based
- Task-specific regularization
- etc.

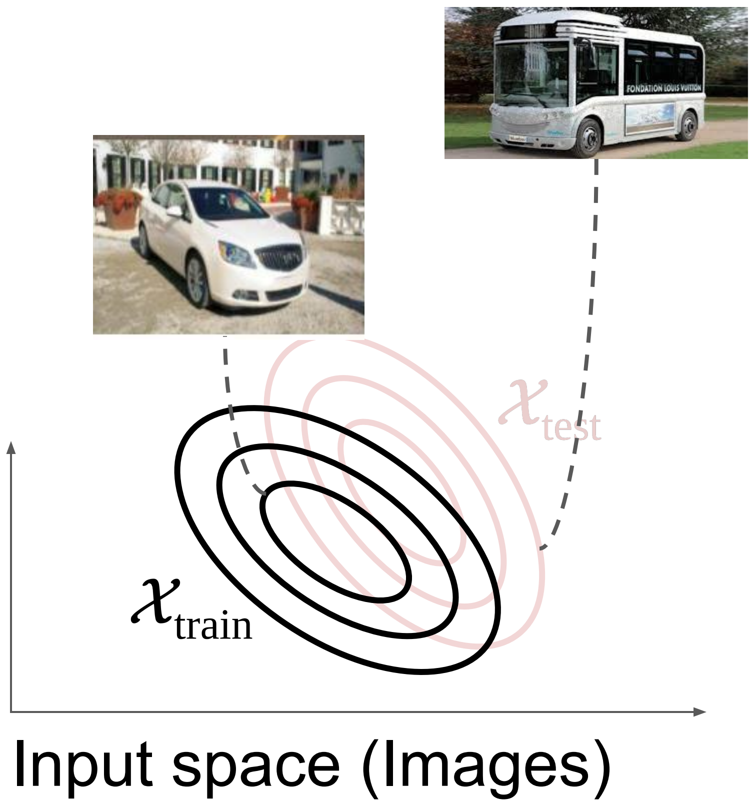
Data Sampling

- Batch sampling
- Triplet, Pair, Tupel sampling
- data synthesis
- etc.

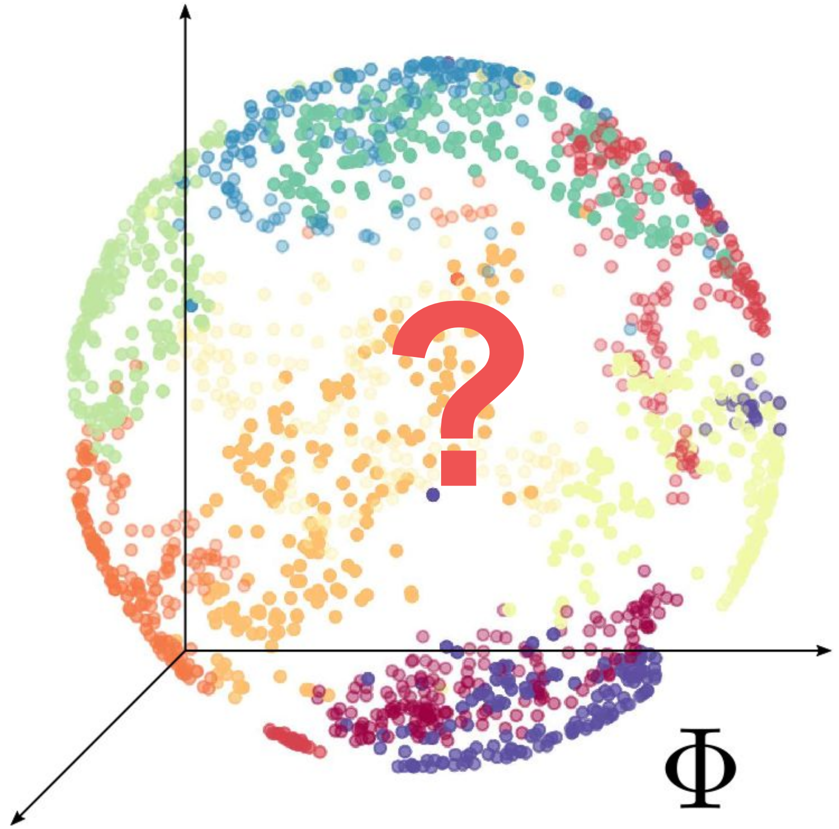
OOD Generalization?

Learning Visual Representations

(**Out-Of-Distribution-Generalization**)
How well does Φ capture **unseen** classes, **unknown** surroundings, viewpoints, continual class **changes**?

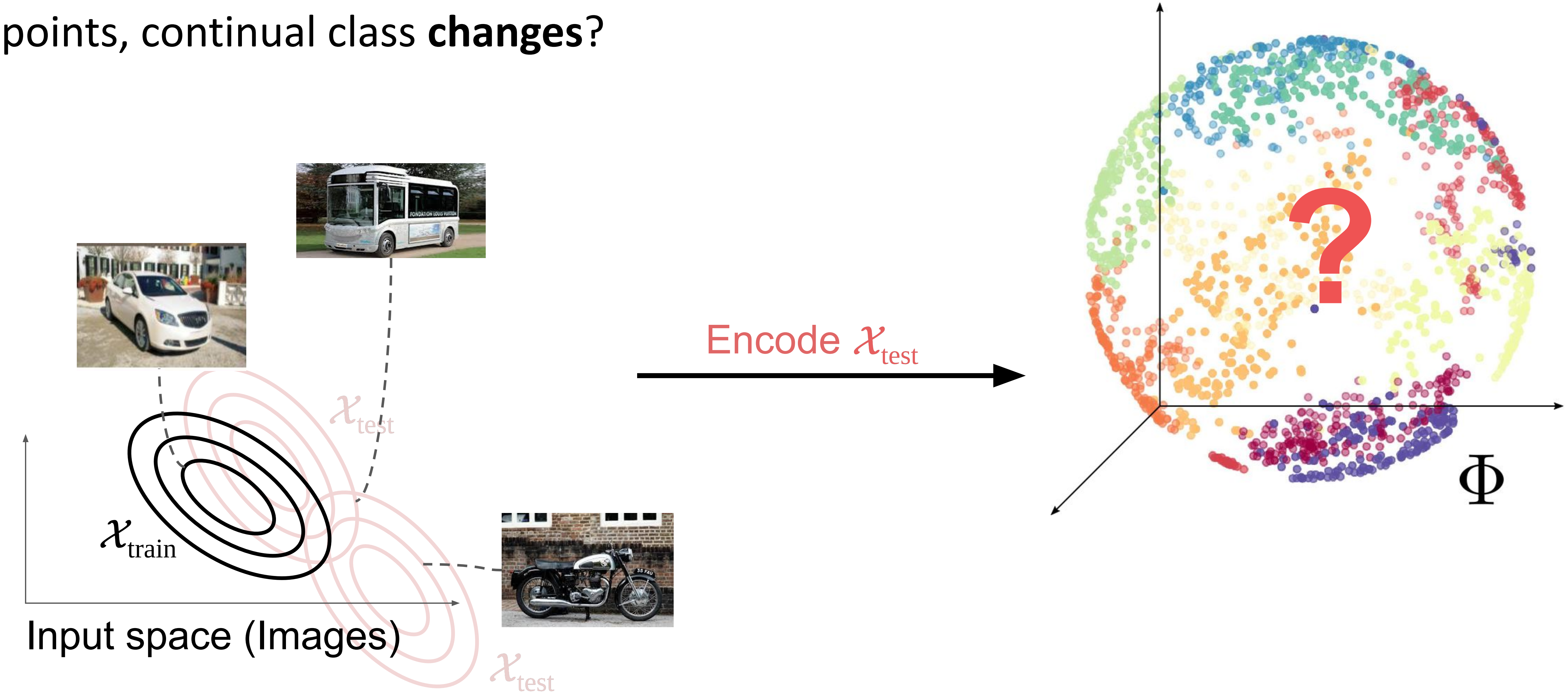


Encode $\mathcal{X}_{\text{test}}$



Learning Visual Representations

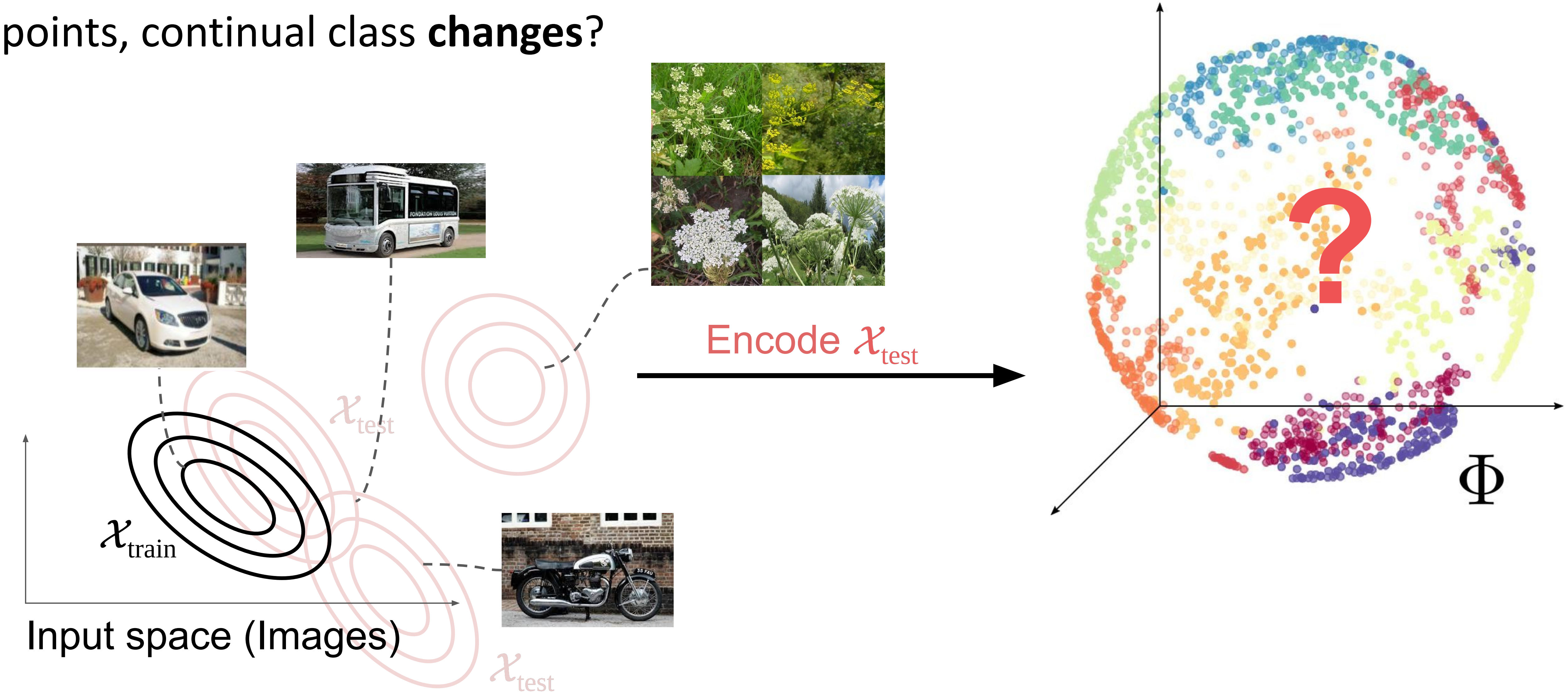
(**Out-Of-Distribution-Generalization**)
How well does Φ capture **unseen** classes, **unknown** surroundings, viewpoints, continual class **changes**?



Learning Visual Representations

(**Out-Of-Distribution-Generalization**)

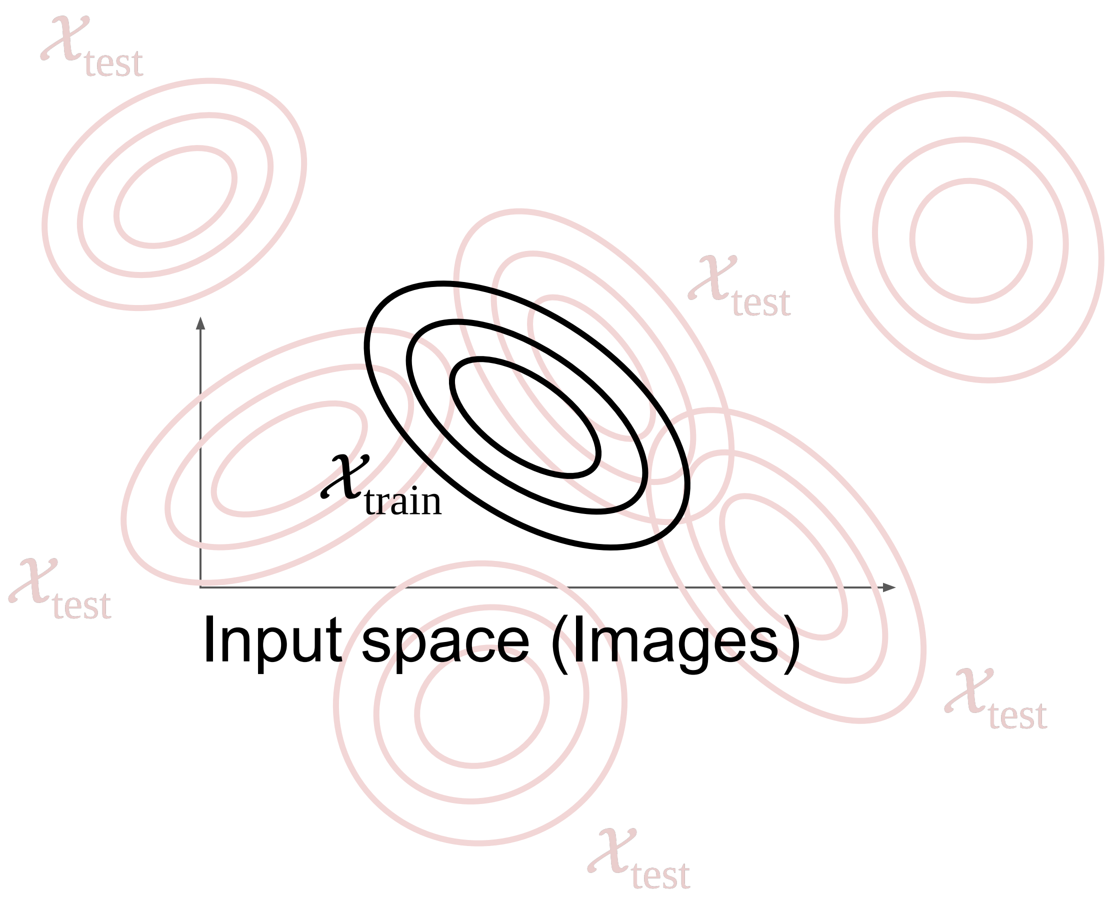
How well does Φ capture **unseen** classes, **unknown** surroundings, viewpoints, continual class **changes**?

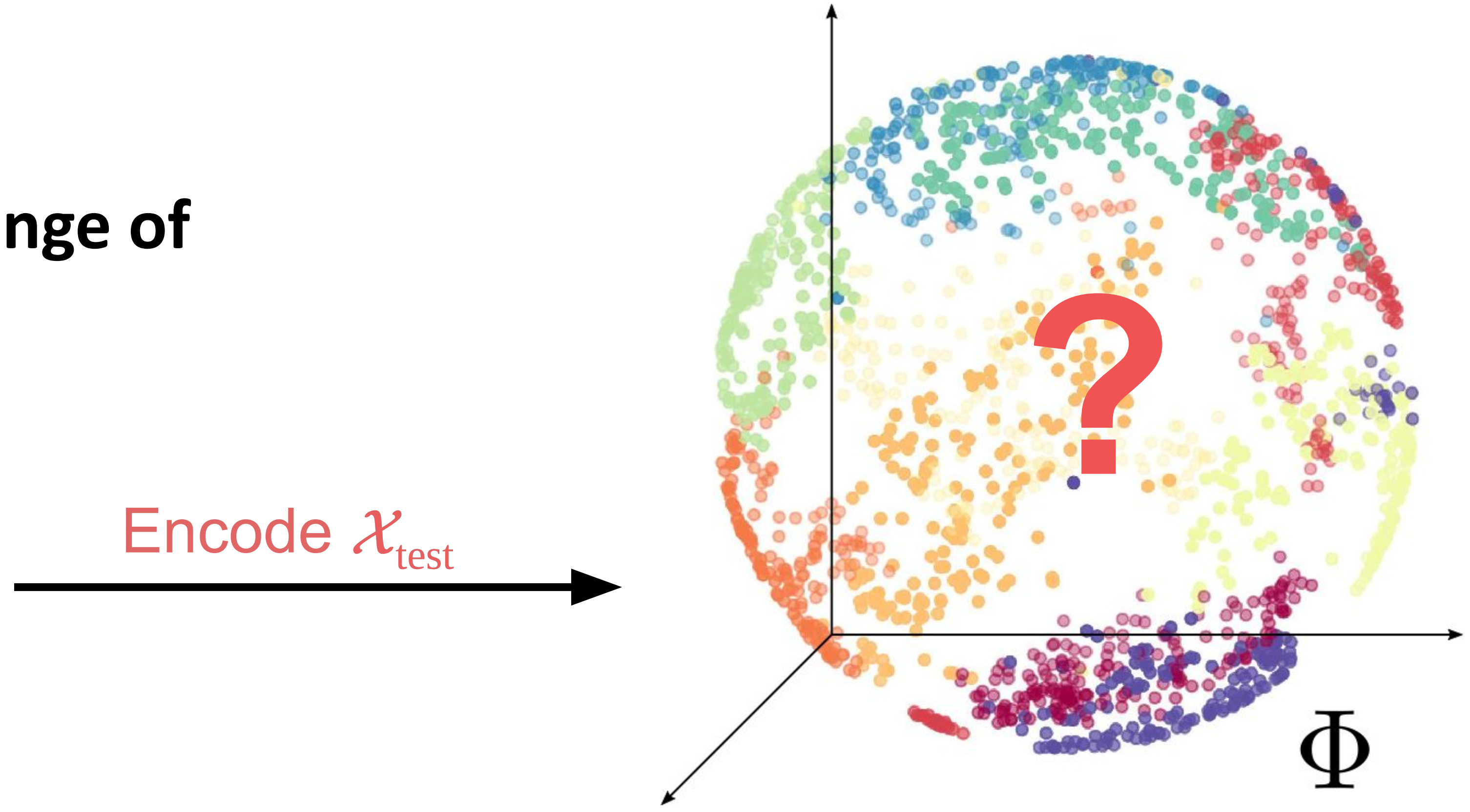


Learning Visual Representations

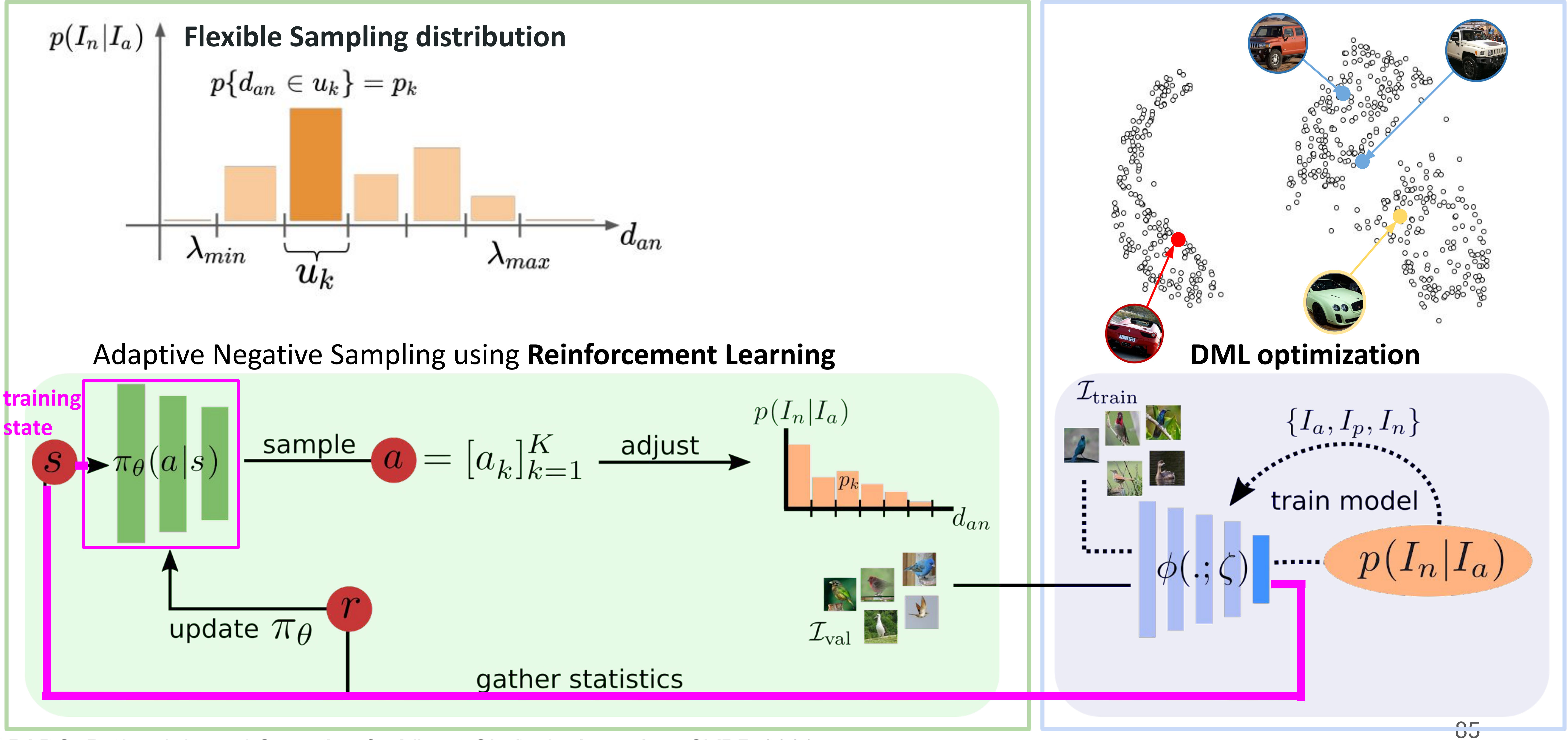
(**Out-Of-Distribution-Generalization**)
How well does Φ capture **unseen** classes, **unknown** surroundings, viewpoints, continual class **changes**?

Evaluation needs to consider **broad range of test distributions** and **difficulty.**

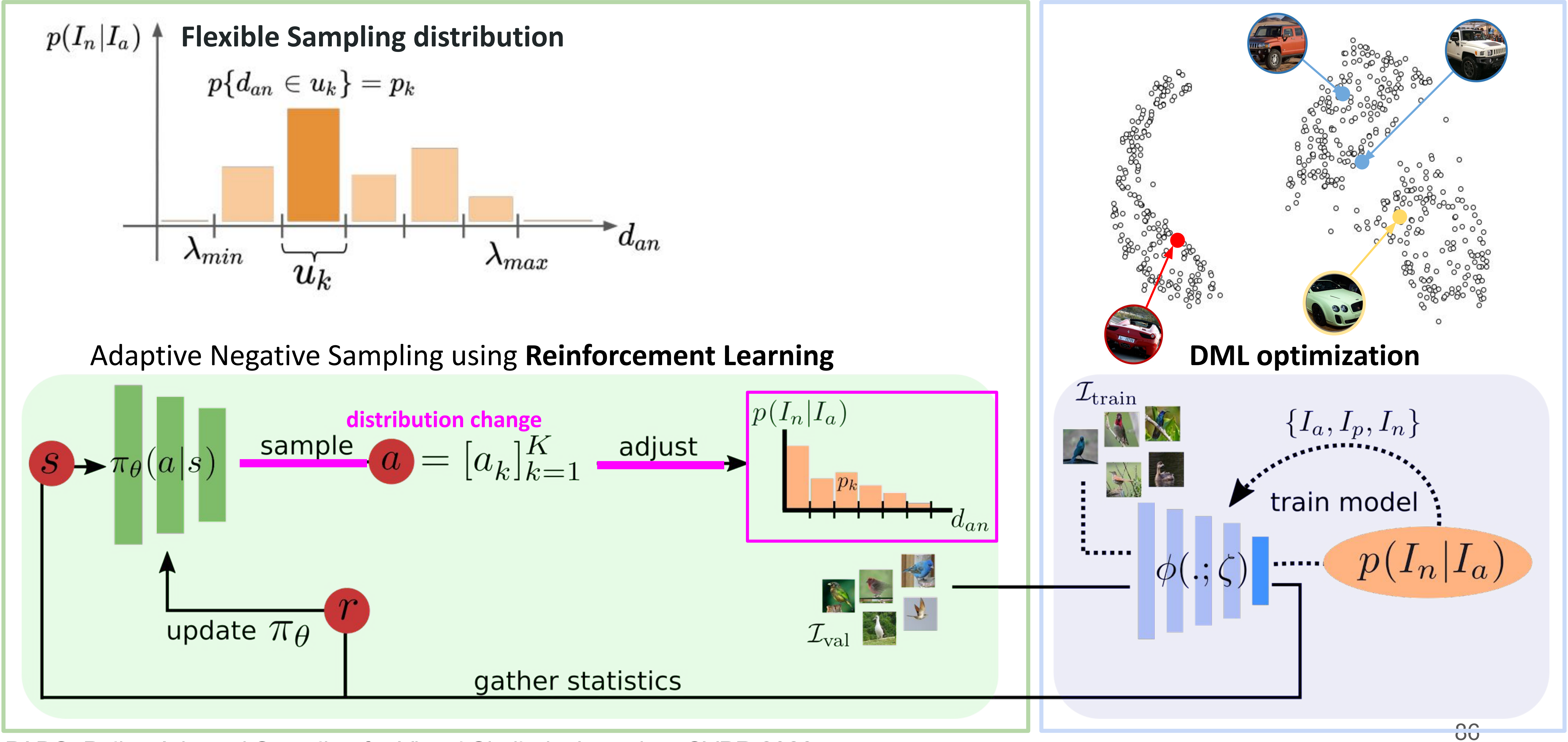




Negative Sampling in Ranking-based DML

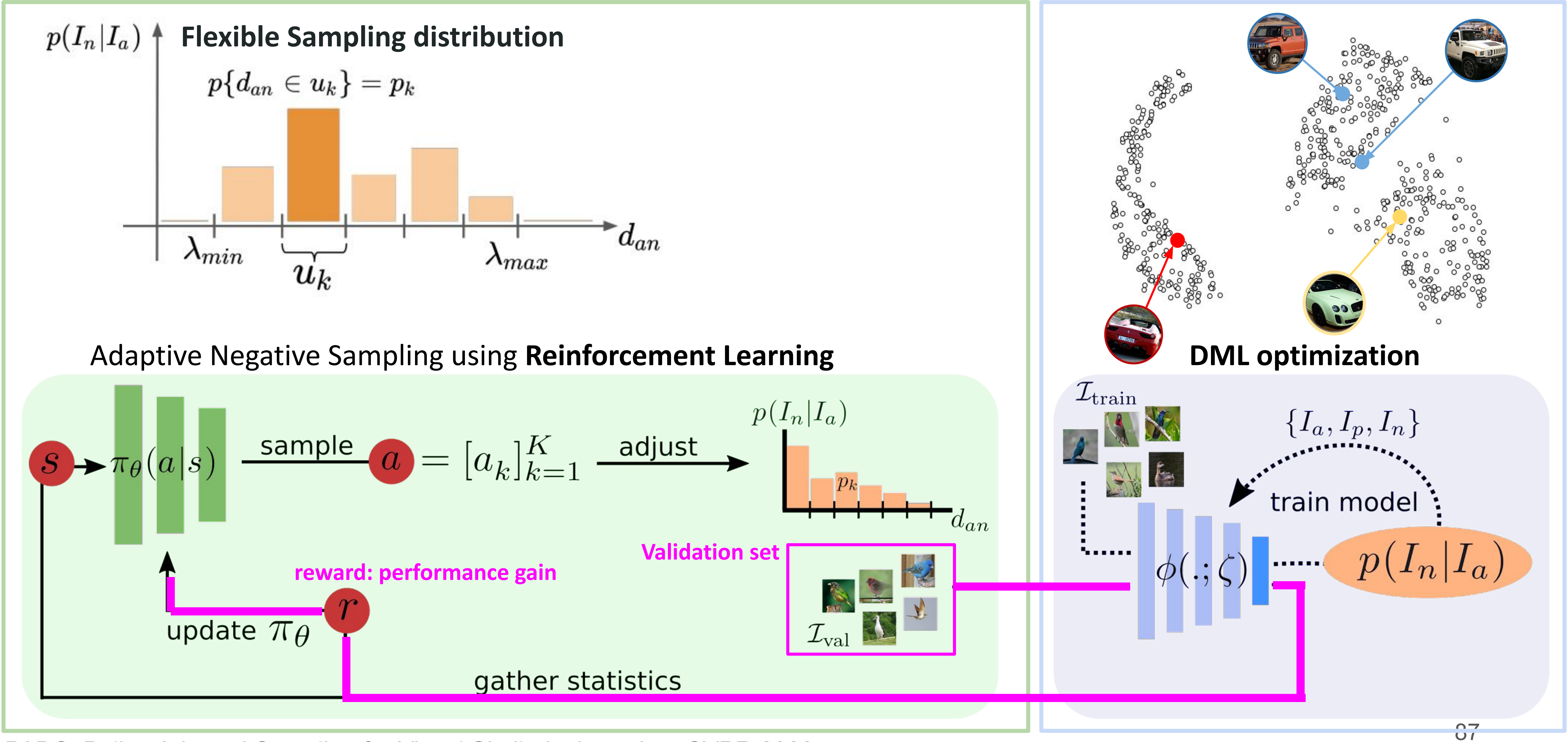


Negative Sampling in Ranking-based DML



PADS: Policy-Adapted Sampling for Visual Similarity Learning, CVPR 2020

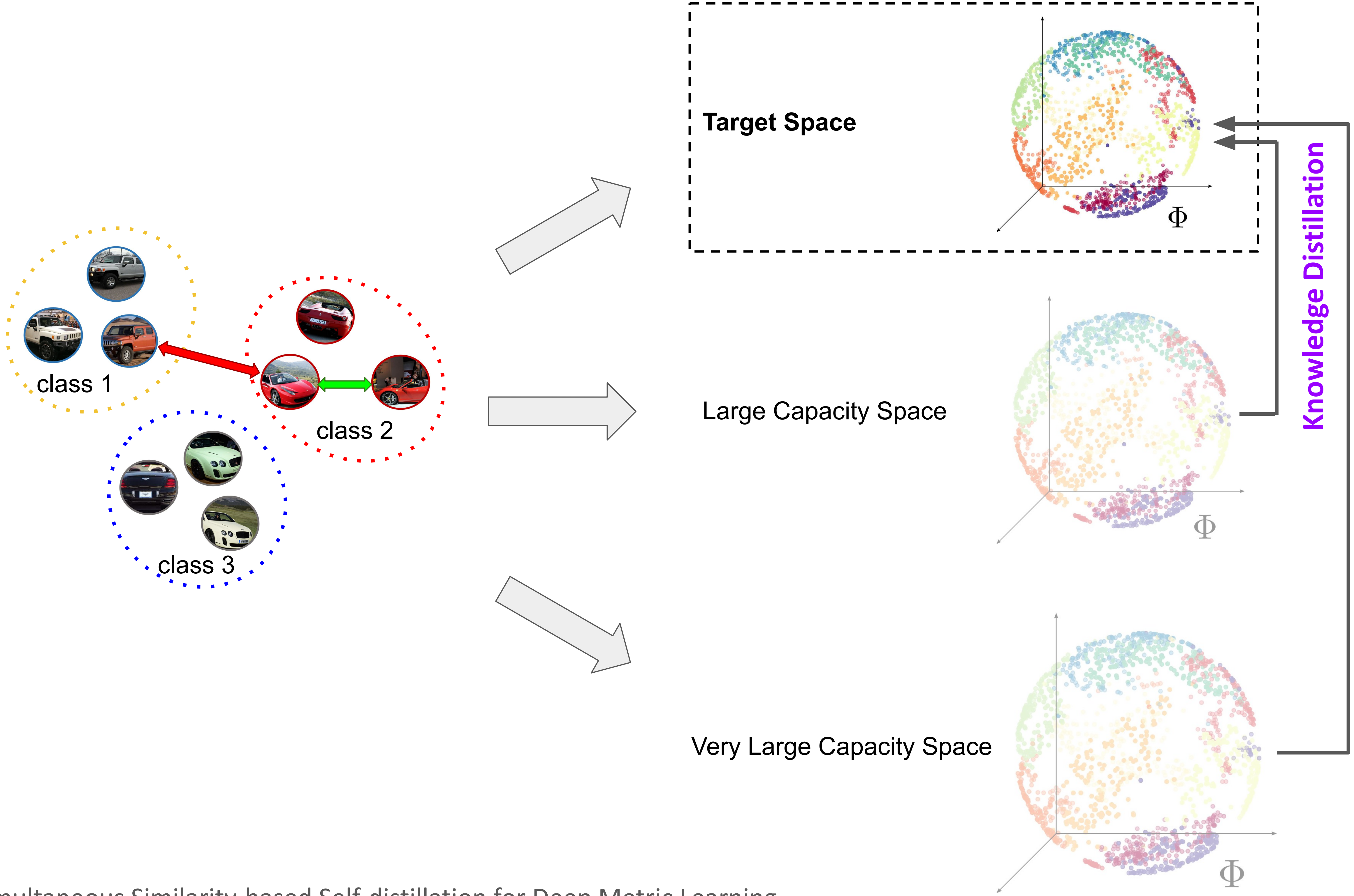
Negative Sampling in Ranking-based DML



PADS: Policy-Adapted Sampling for Visual Similarity Learning, CVPR 2020

Ensemble-based DML

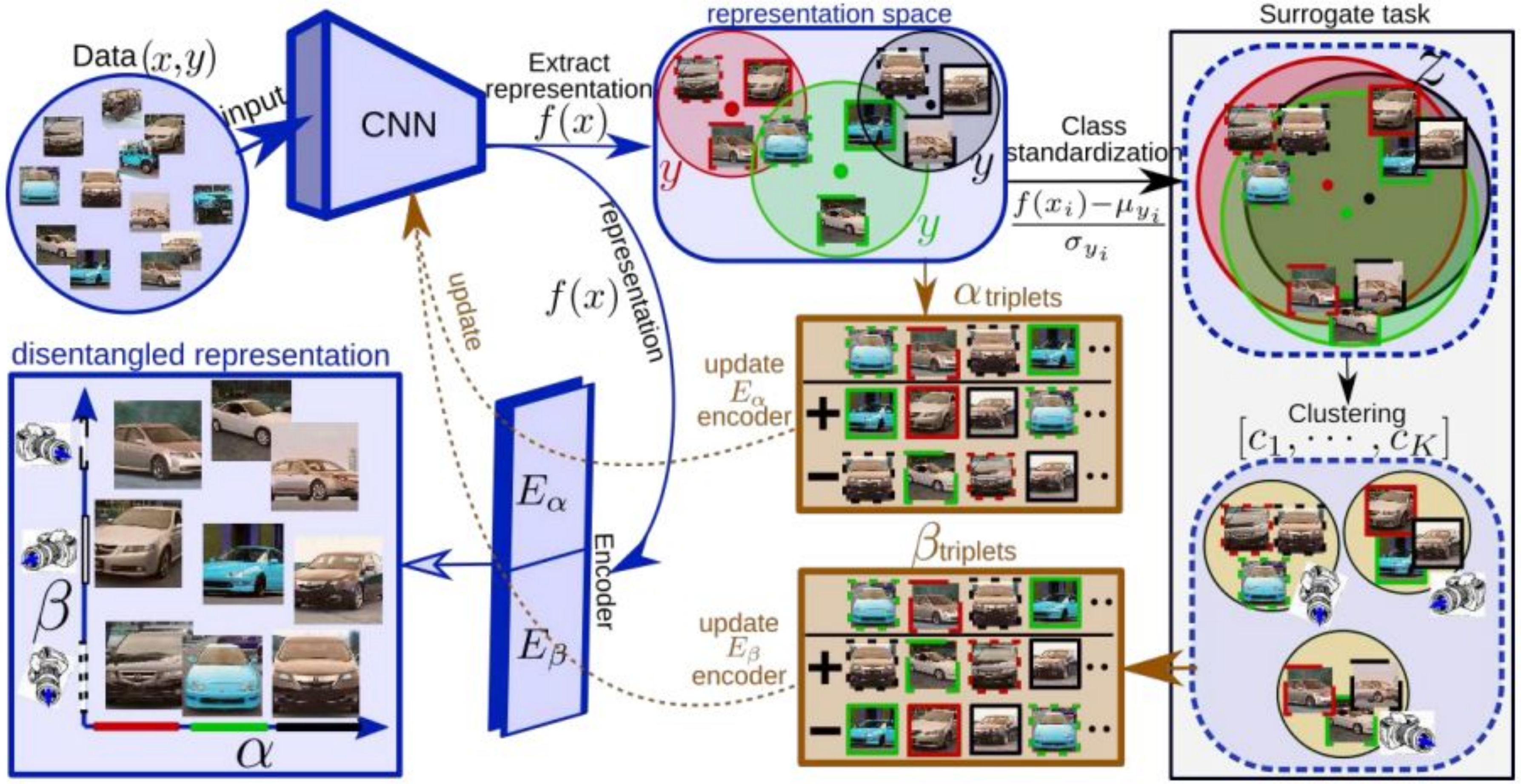
- **Multi Self-Distillation[3]**



[3] Roth, Milbich et al.; ICML 2021; S2SD: Simultaneous Similarity-based Self-distillation for Deep Metric Learning

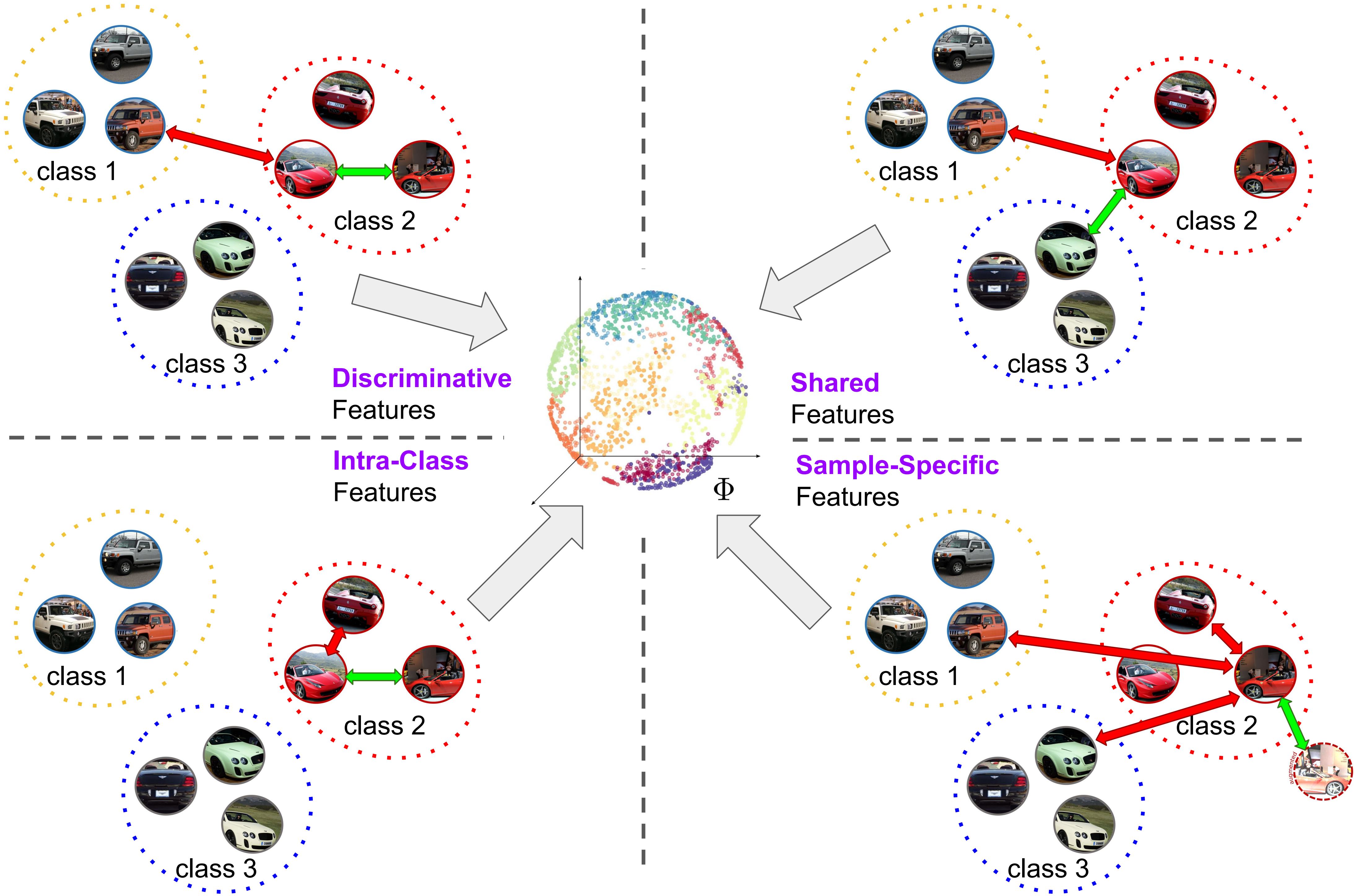
Ensemble-based DML

- Shared Features, DiVA, **MIC** (complementary semantics, “multi-task learning”)
- **explicit** specialization of learners (semantics)



Overview: Typical Learning Concepts for DML Generalization

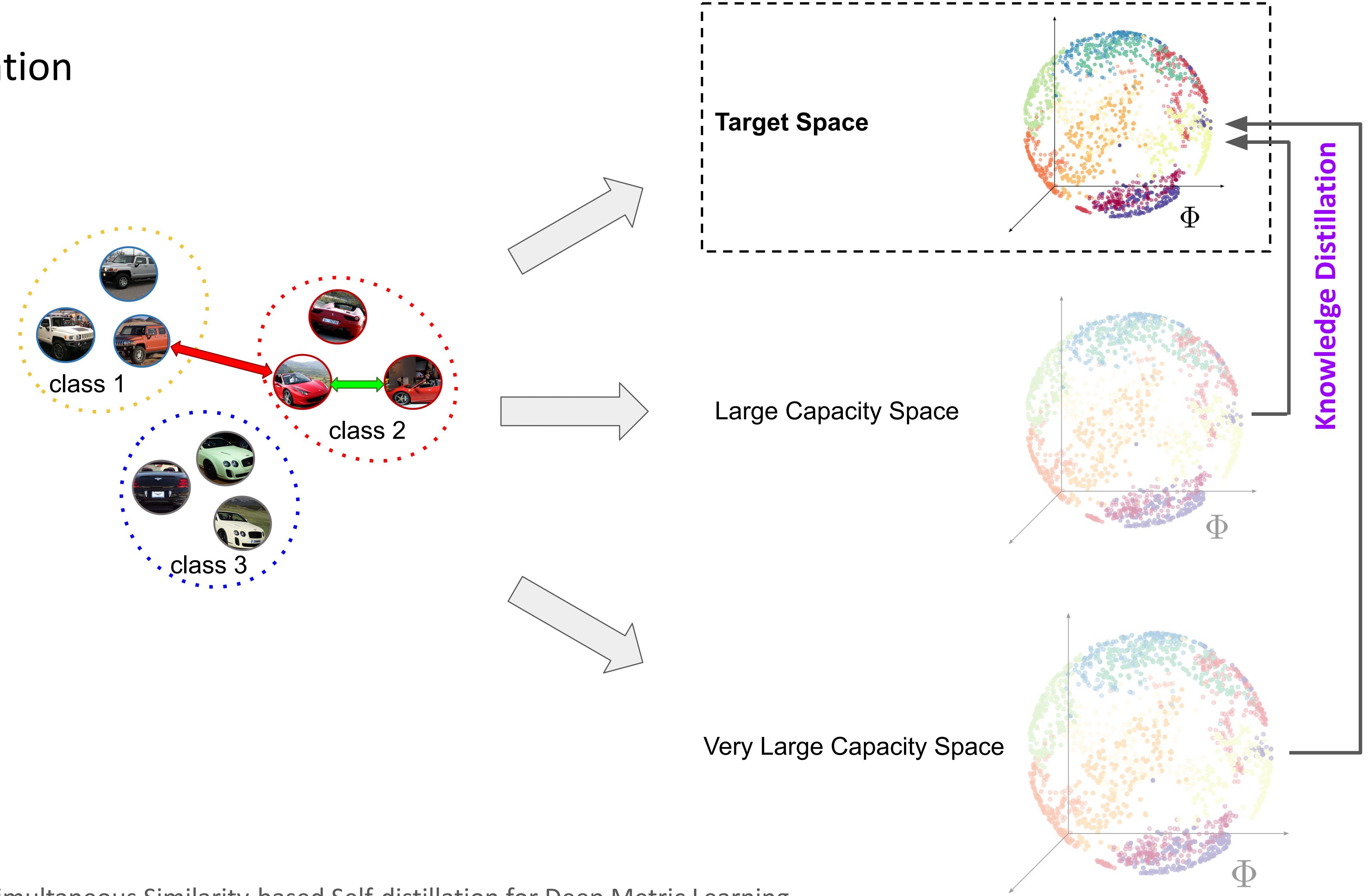
- **Diverse Feature Aggregation**[2]
- Multi Self-Distillation
- Proxy-Learning
- ...



[2] Milbich, Roth et al.; ECCV 2020; DiVA: Diverse Visual Feature Aggregation for Deep Metric Learning

Overview: Typical Learning Concepts for DML Generalization

- Diverse Feature Aggregation
- **Multi Self-Distillation[3]**
- Proxy Learning
- ...



[3] Roth, Milbich et al.; ICML 2021; S2SD: Simultaneous Similarity-based Self-distillation for Deep Metric Learning