Chrome 文件 编辑 视图 历史记录 书签 用户 标签页 窗口 帮助 100% 周二 下午6:17

Recording · SlidesLive Rec

recorder.slideslive.com/recording/#!token=ACL_posters_4066-6438_distance-metric-learning-with

Here are a few tips how to make your recording look good! Check out also our short video tutorial!

1. Select microphone you want to use to record your voice.
2. Select camera.
3. Check that you can see your screen below, on the right (Screen capture) – we'll be recording your slides aswell.
4. Press **Start recording** and present.

Tip: Try just 30s first and review to make sure all looks good (check out our recording tips!).

Video recording

No video will be recorded!

Microphone

默认 - Internal Microphone (Built-in)

Please select a microphone you want to use.

Camera

Screen capture

Change screen

Show previous recordings

If you have any questions please contact your production manager.
If you face any technical issues contact support@slideslive.com.

Chrome 文件 编辑 视图 历史记录 书签 用户 标签页 窗口 帮助

Recording · SlidesLive Rec

recorder.slideslive.com/recording/#!token=ACL_posters_4066-6438_distance-metric-learning-with

You video is not being recorded!

00:00:00

STOP RECORDING

Video recording

No video will be recorded!

Microphone

默认 - Internal Microphone (Built-in)

Please select a microphone you want to use.

Camera

Screen capture

Change screen

STOP RECORDING

If you have any questions please contact your production manager.
If you face any technical issues contact support@slideslive.com.

Distance Metric Learning with Joint Representation Diversification

Xu Chu[1,2] Yang Lin[1,2] Yasha Wang[2,3] Xiting Wang[4]
Hailong Yu[1,2] Xin Gao[1,2] Qi Tong[2,5]

[1]School of Electronics Engineering and Computer Science, Peking University

[2]Key Laboratory of High Confidence Software Technologies, Ministry of Education

[3]National Engineering Research Center of Software Engineering, Peking University

[4]Microsoft Research Asia

[5]School of Software and Microelectronics, Peking University

June, 2020

Distance Metric Learning with Joint Representation Diversification

Xu Chu[1,2] Yang Lin[1,2] Yasha Wang[2,3] Xiting Wang[4]
Hailong Yu[1,2] Xin Gao[1,2] Qi Tong[2,5]

[1]School of Electronics Engineering and Computer Science, Peking University

[2]Key Laboratory of High Confidence Software Technologies, Ministry of Education

[3]National Engineering Research Center of Software Engineering, Peking University

[4]Microsoft Research Asia

[5]School of Software and Microelectronics, Peking University

June, 2020

Distance Metric Learning with Joint Representation Diversification

Xu Chu[1,2] Yang Lin[1,2] Yasha Wang[2,3] Xiting Wang[4]
Hailong Yu[1,2] Xin Gao[1,2] Qi Tong[2,5]

[1]School of Electronics Engineering and Computer Science, Peking University

[2]Key Laboratory of High Confidence Software Technologies, Ministry of Education

[3]National Engineering Research Center of Software Engineering, Peking University

[4]Microsoft Research Asia

[5]School of Software and Microelectronics, Peking University

June, 2020

Distance Metric Learning with Joint Representation Diversification

Xu Chu[1,2] Yang Lin[1,2] Yasha Wang[2,3] Xiting Wang[4]
Hailong Yu[1,2] Xin Gao[1,2] Qi Tong[2,5]

[1]School of Electronics Engineering and Computer Science, Peking University

[2]Key Laboratory of High Confidence Software Technologies, Ministry of Education

[3]National Engineering Research Center of Software Engineering, Peking University

[4]Microsoft Research Asia

[5]School of Software and Microelectronics, Peking University

June, 2020

Distance Metric Learning with Joint Representation Diversification

Xu Chu[1,2] Yang Lin[1,2] Yasha Wang[2,3] Xiting Wang[4]
Hailong Yu[1,2] Xin Gao[1,2] Qi Tong[2,5]

[1]School of Electronics Engineering and Computer Science, Peking University

[2]Key Laboratory of High Confidence Software Technologies, Ministry of Education

[3]National Engineering Research Center of Software Engineering, Peking University

[4]Microsoft Research Asia

[5]School of Software and Microelectronics, Peking University

June, 2020

Distance Metric Learning with Joint Representation Diversification

Xu Chu[1,2] Yang Lin[1,2] Yasha Wang[2,3] Xiting Wang[4]
Hailong Yu[1,2] Xin Gao[1,2] Qi Tong[2,5]

[1]School of Electronics Engineering and Computer Science, Peking University

[2]Key Laboratory of High Confidence Software Technologies, Ministry of Education

[3]National Engineering Research Center of Software Engineering, Peking University

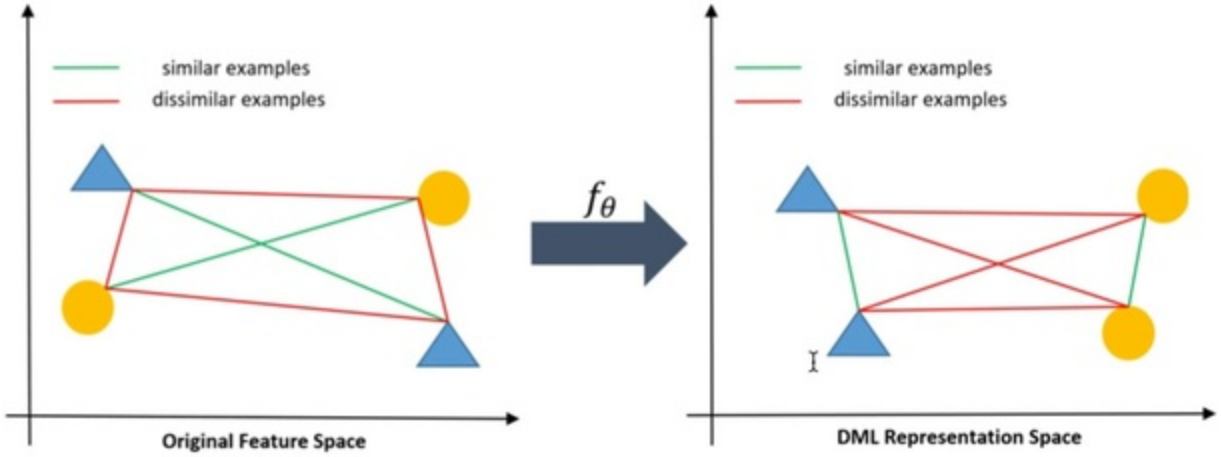
[4]Microsoft Research Asia

[5]School of Software and Microelectronics, Peking University

June, 2020

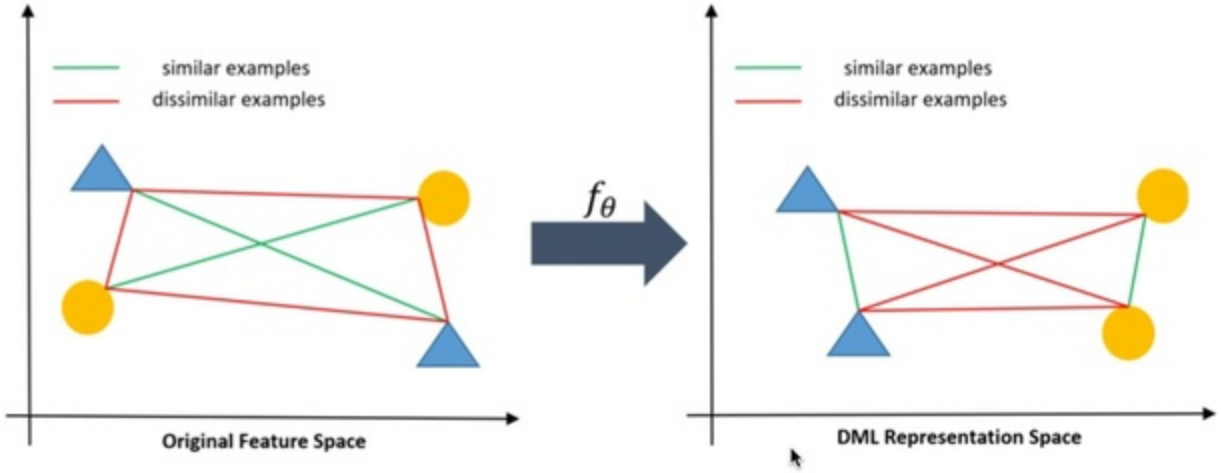
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



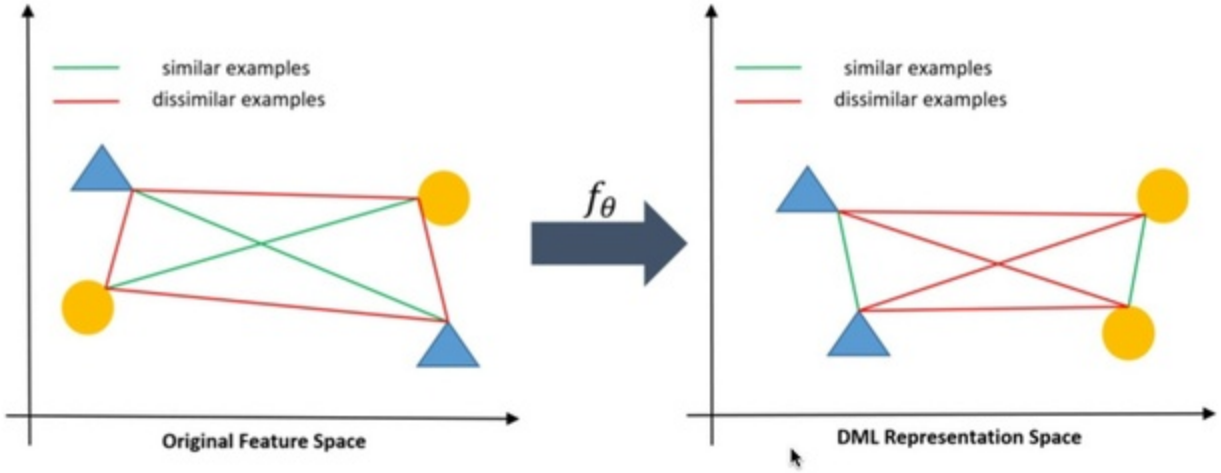
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



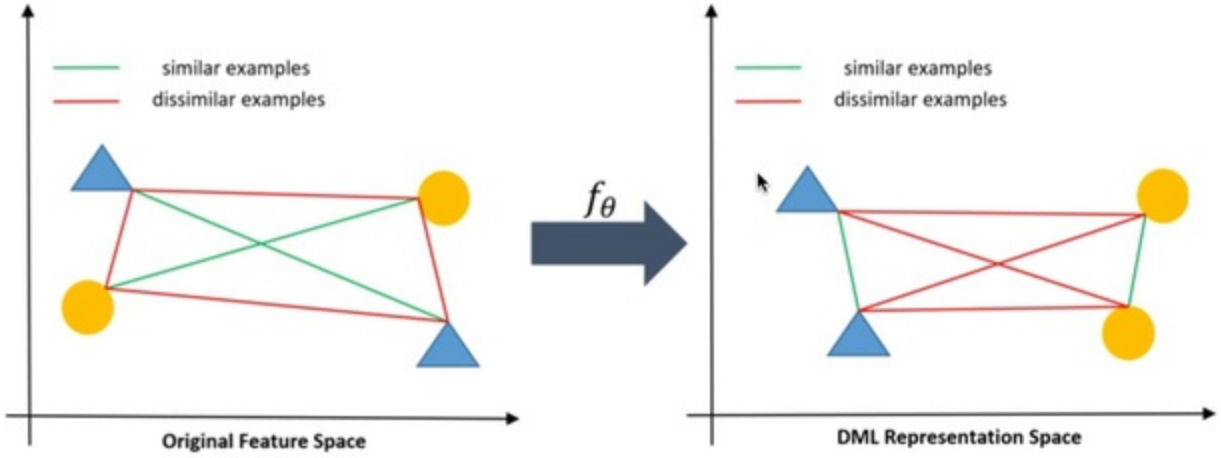
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



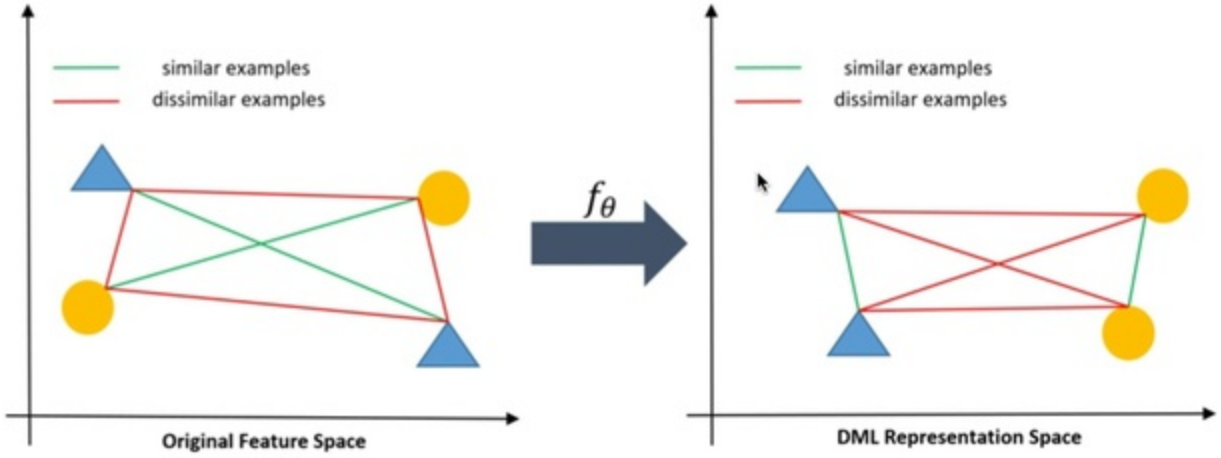
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



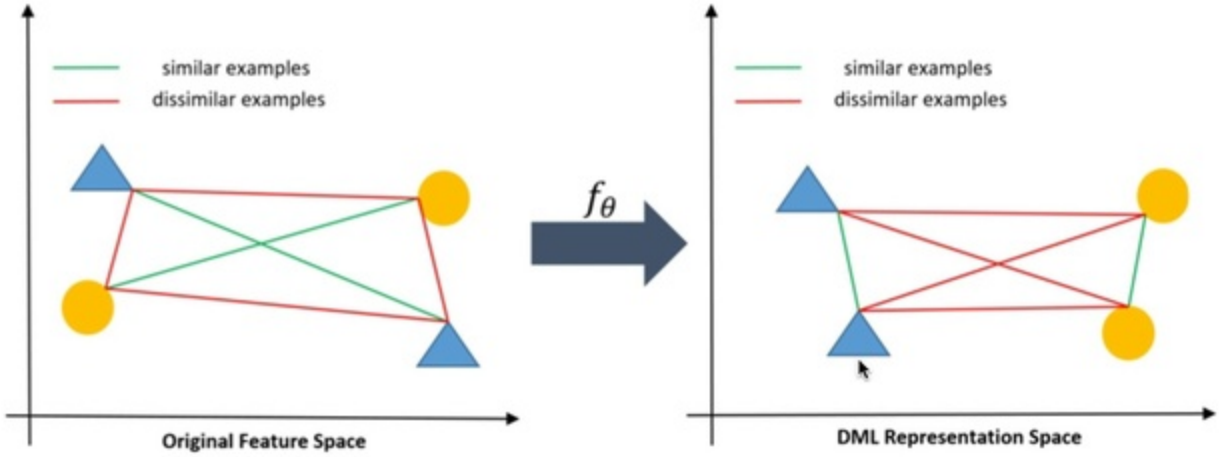
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



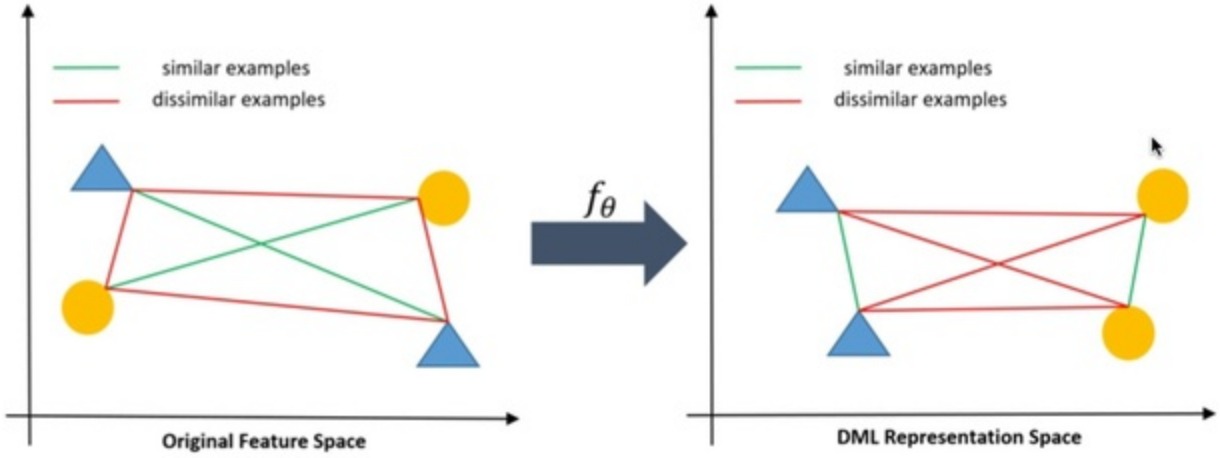
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



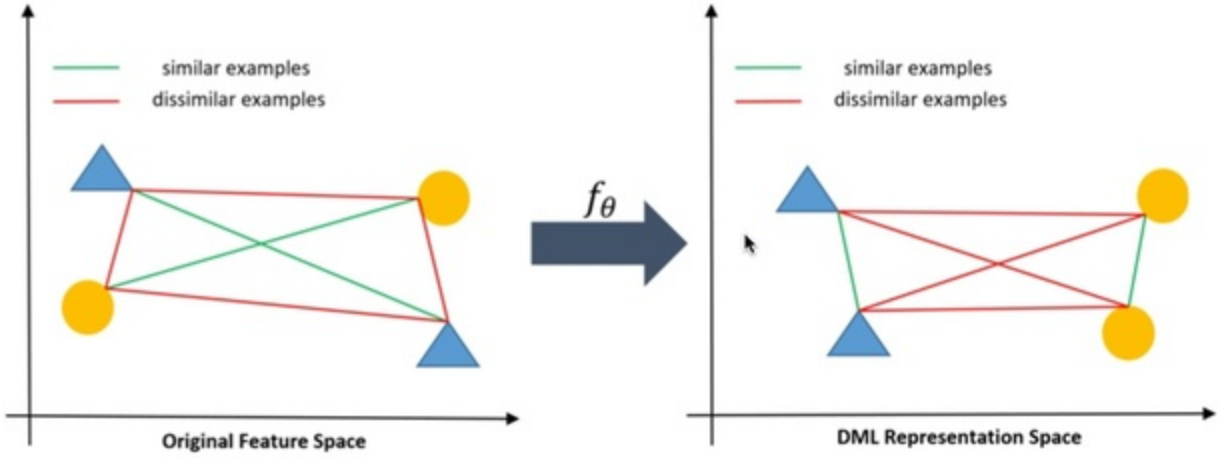
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



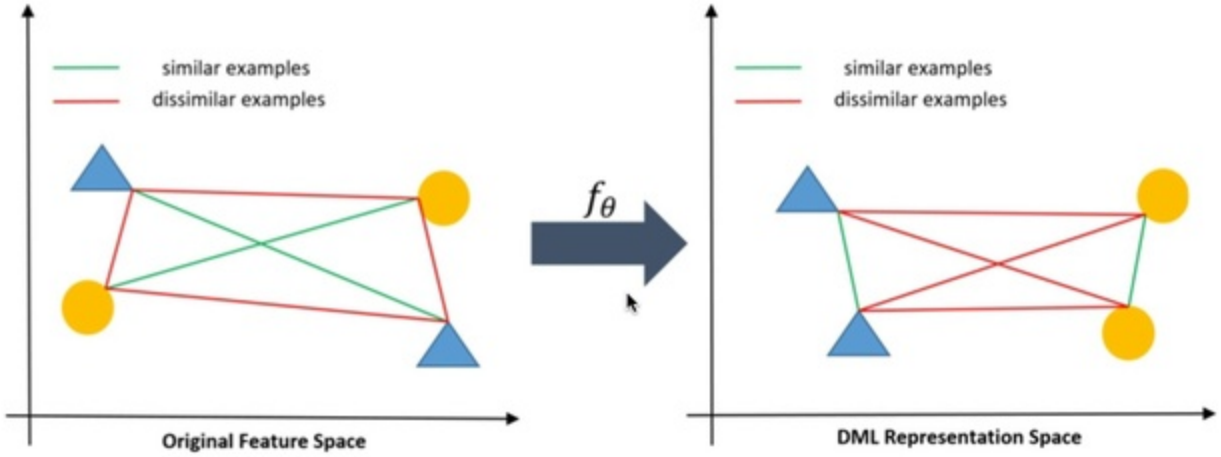
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



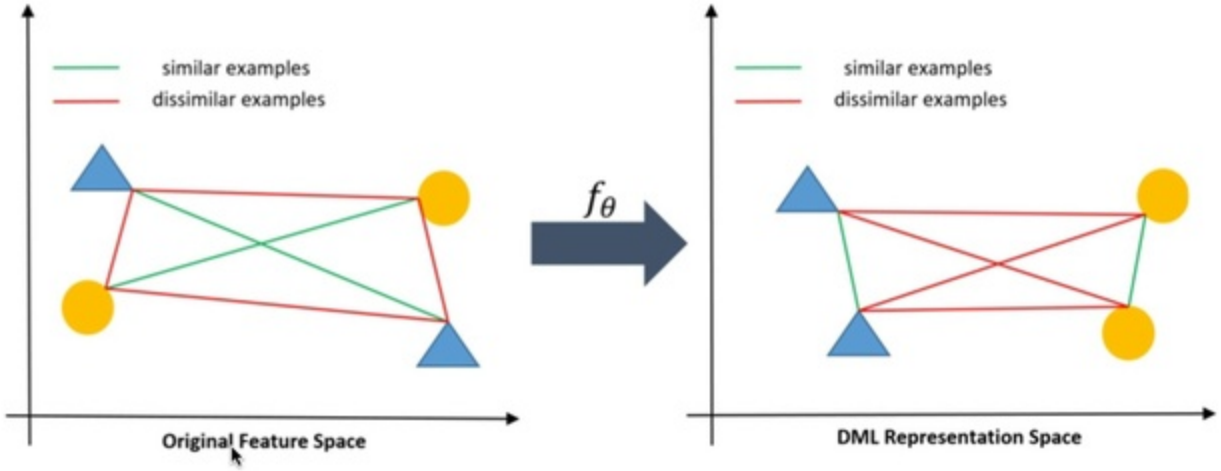
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



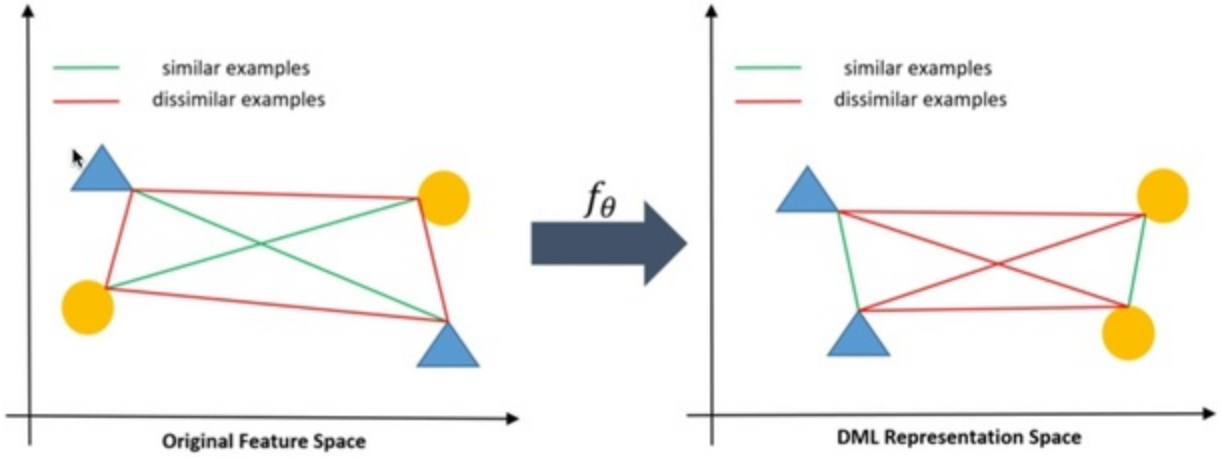
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



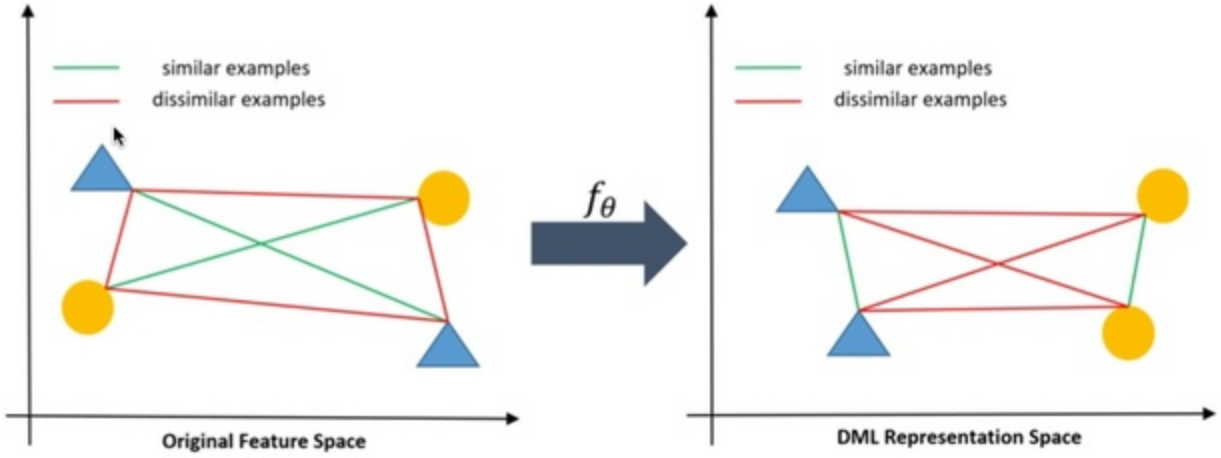
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



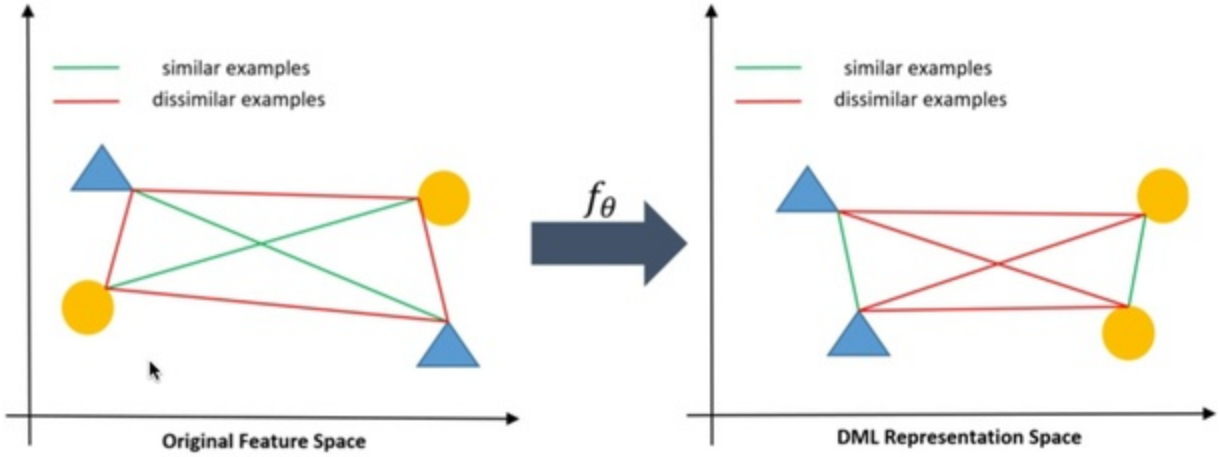
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



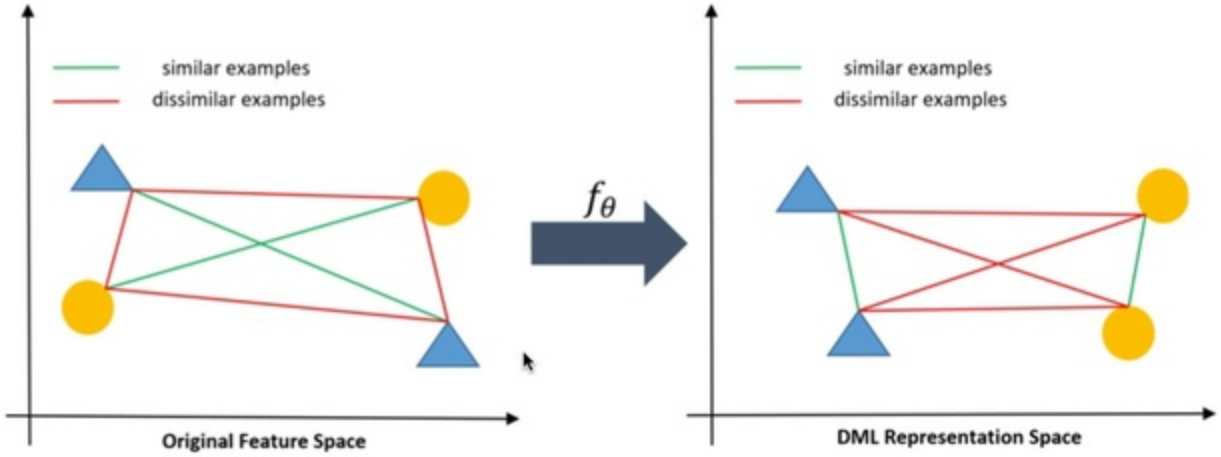
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



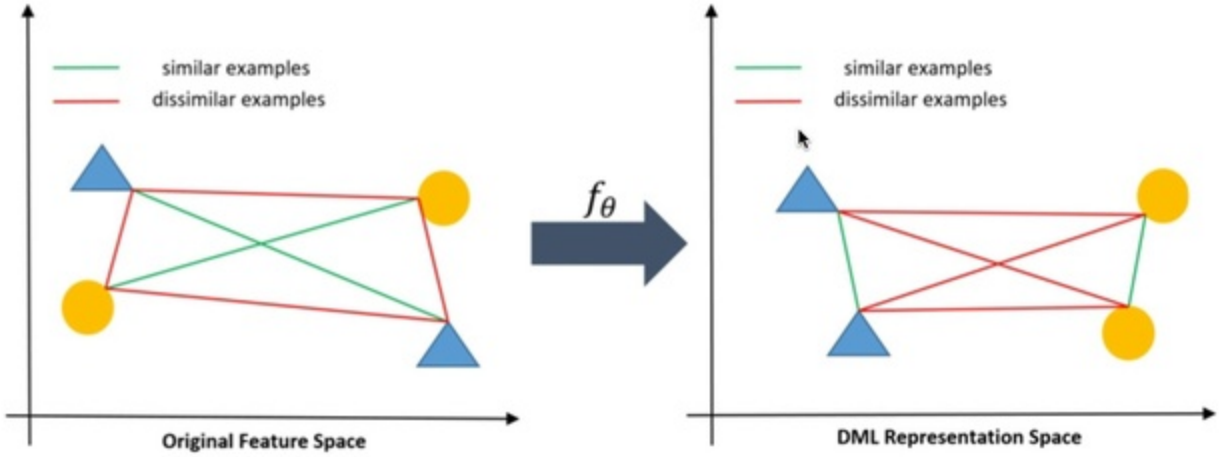
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



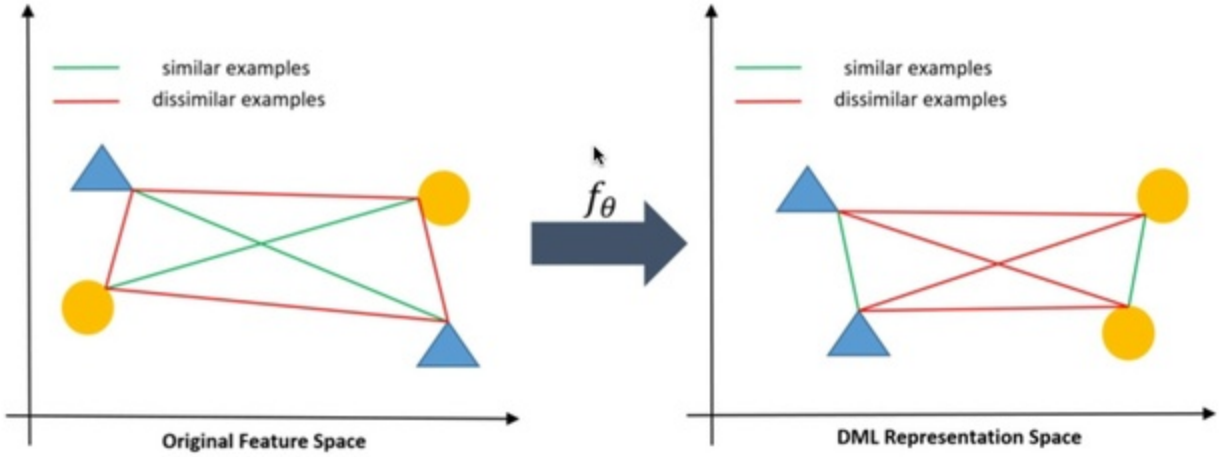
The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



The goal of distance metric learning (DML)

Learn a **mapping** f_θ from the original feature space to a representation space where similar examples are closer than dissimilar examples in the learned representation space.



The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...

The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...

The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...

The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...

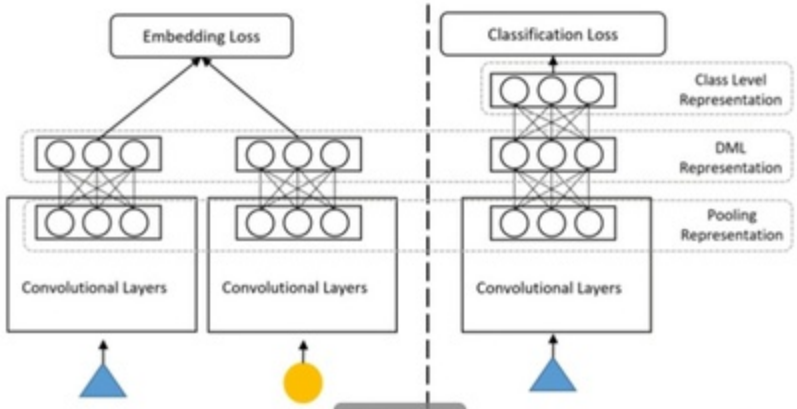
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]:
  $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



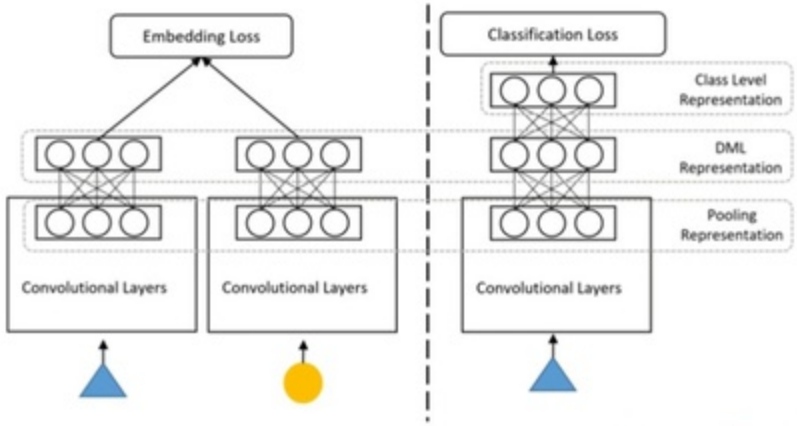
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]:
  $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



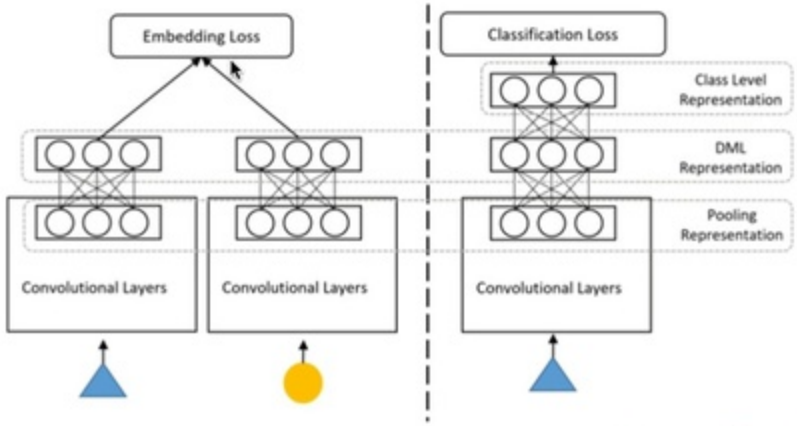
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]:
  $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



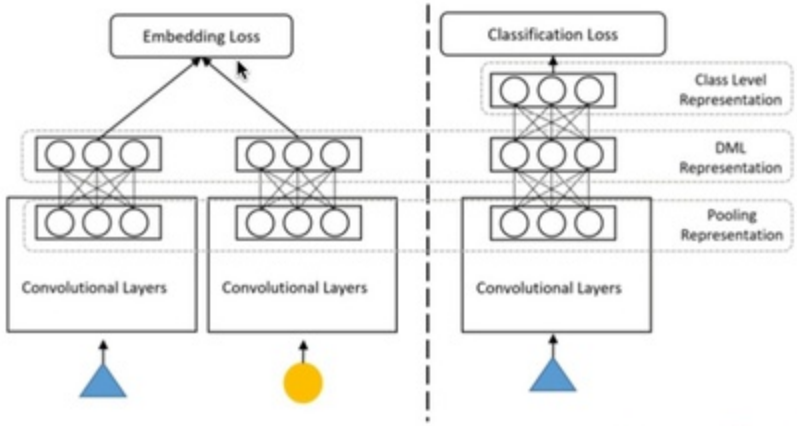
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



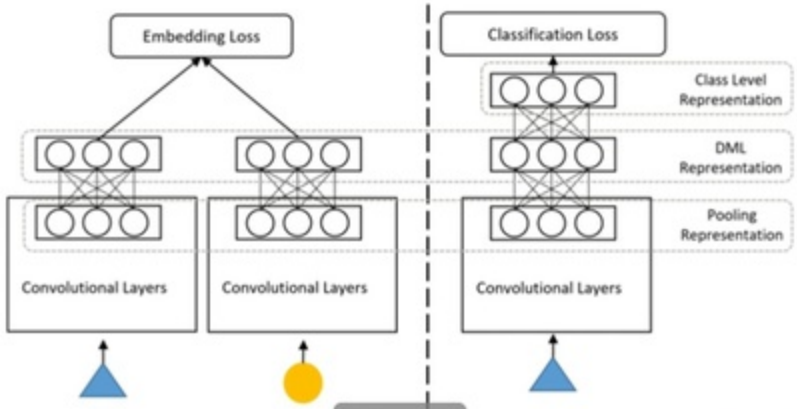
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

Embedding Loss

- Contrastive loss [Chopra et al., 2005]:
  $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

Classification Loss

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



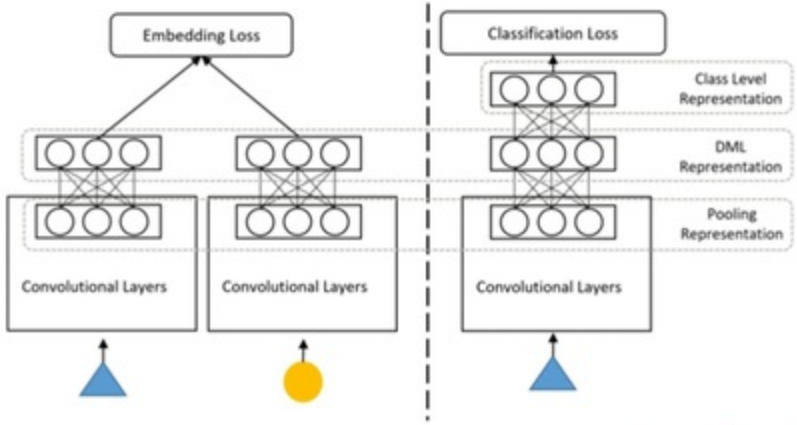
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \Sigma_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



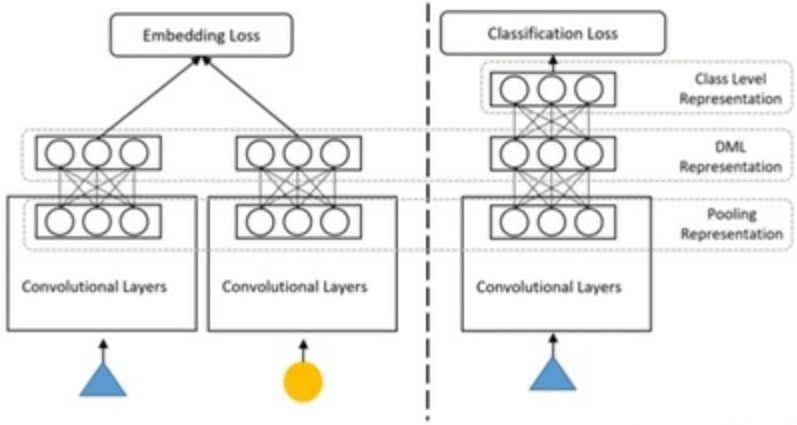
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



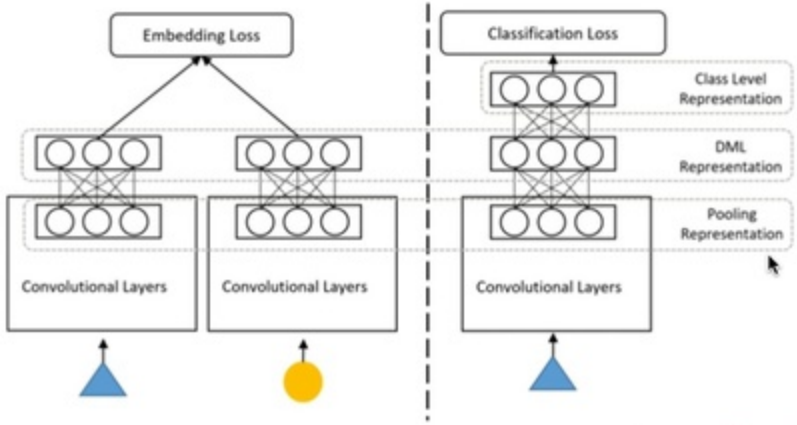
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

Embedding Loss

- Contrastive loss [Chopra et al., 2005]:
  $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

Classification Loss

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{s Sim(x_i, w_j)}}$
- ...



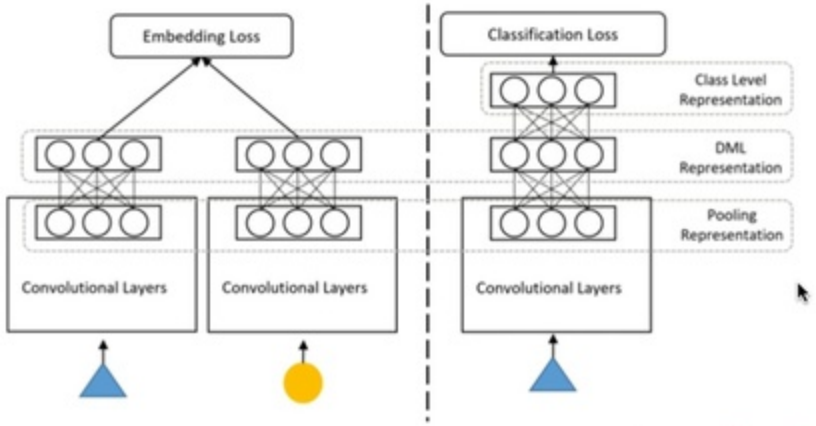
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



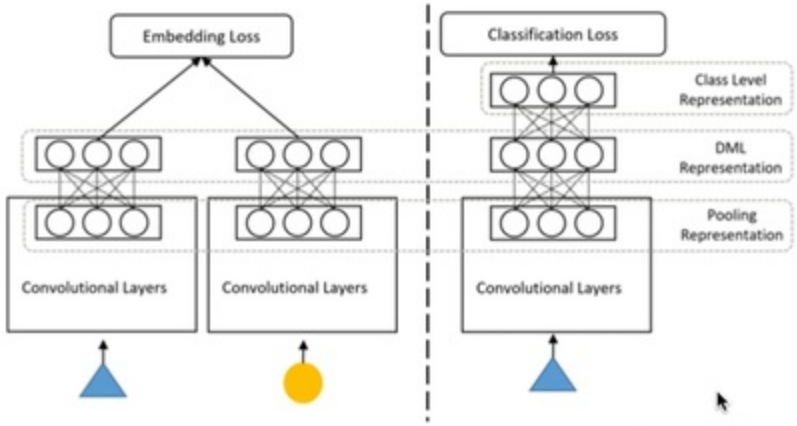
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



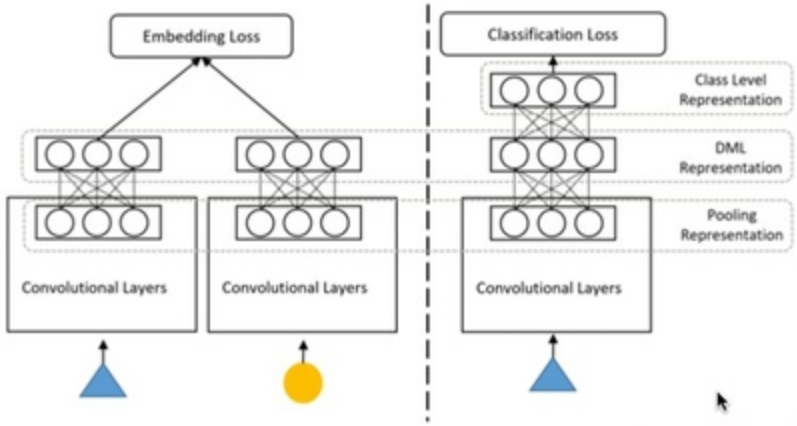
The training objectives of deep DML methods encourage intra-class compactness and inter-class separability.

EMBEDDING LOSS

- Contrastive loss [Chopra et al., 2005]: $\ell_{contrastive} = [d(x_a, x_p) - m_{pos}]_+ + [m_{neg} - d(x_a, x_n)]_+$
- Triplet loss [Schroff et al., 2015]: $\ell_{triplet} = [d(x_a, x_p) - d(x_a, x_n) + m]_+$
- ...

CLASSIFICATION LOSS

- AMSoftmax loss [Wang et al., 2018]: $\ell_{AM} = -log \frac{e^{s(Sim(x_i, w_{y_i}) - m)}}{e^{s(Sim(x_i, w_{y_i}) - m)} + \sum_{j \neq y_i}^{C} e^{sSim(x_i, w_j)}}$
- ...



Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors

Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors

Trade-off between intra-class compactness and inter-class separability.

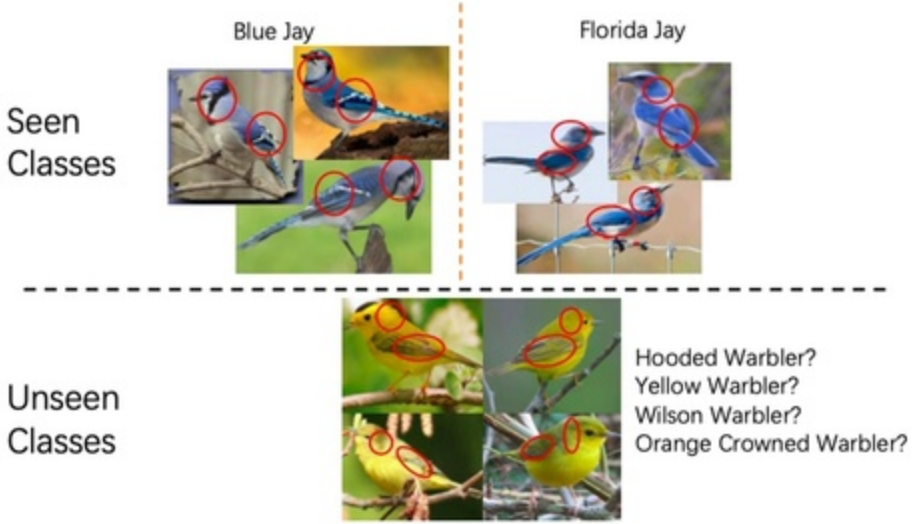
- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors

Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors

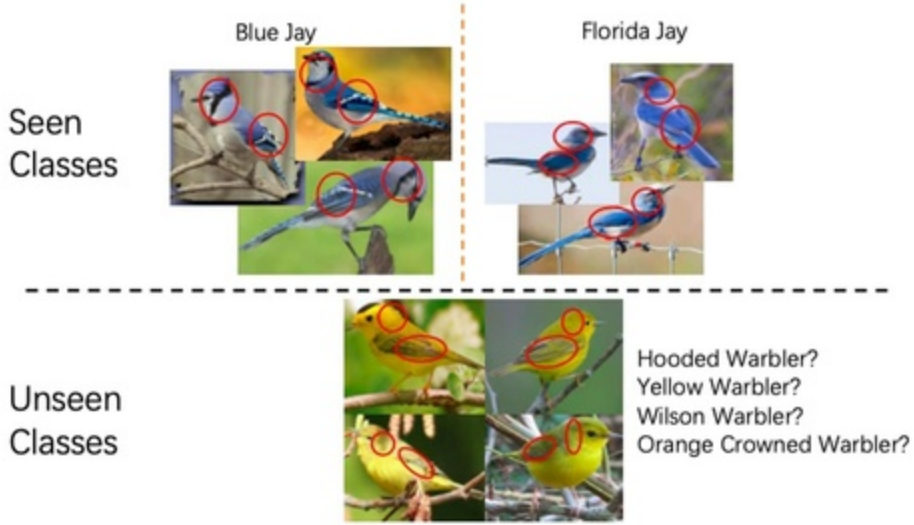
Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors



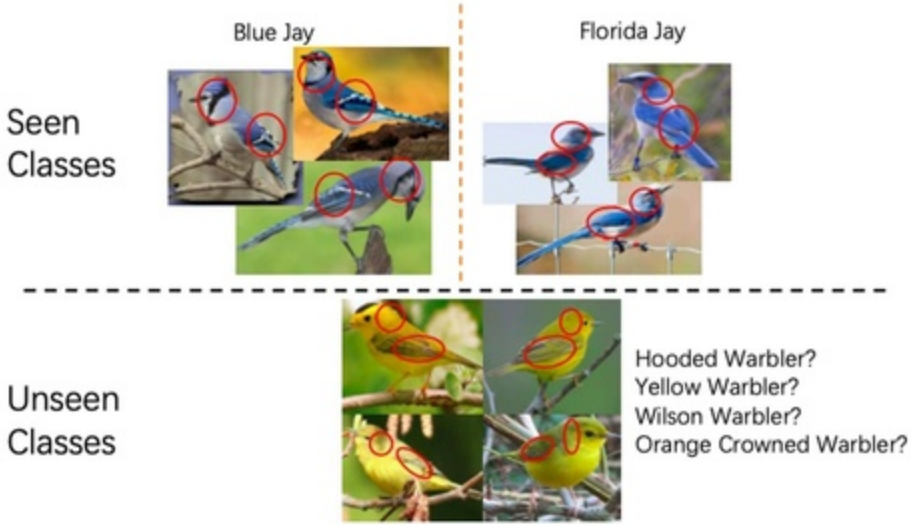
Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors



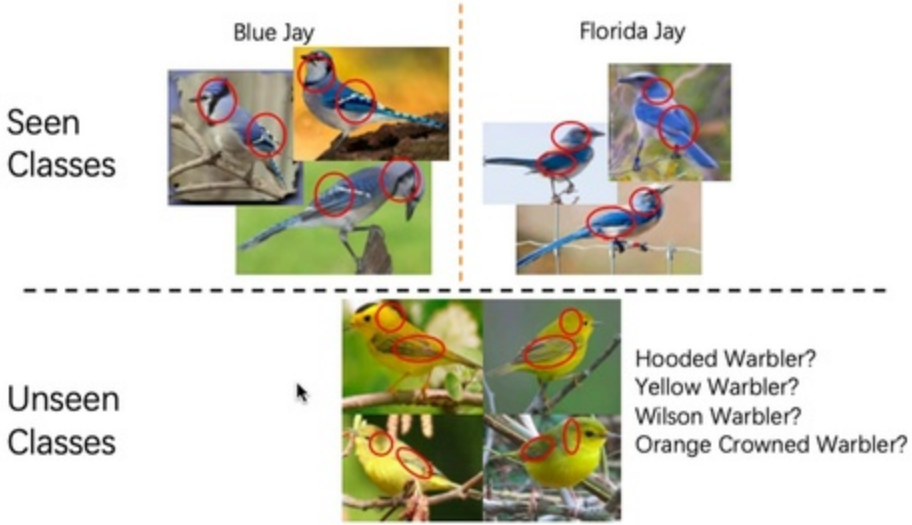
Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors



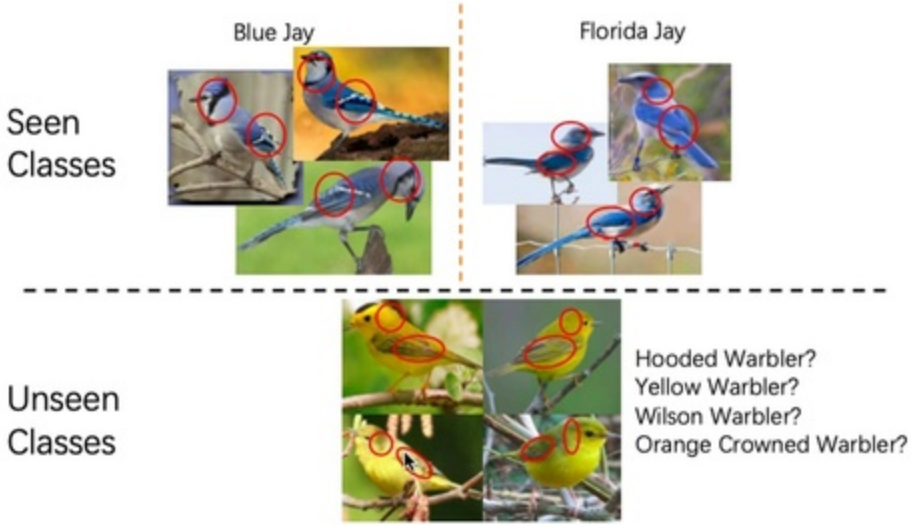
Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors



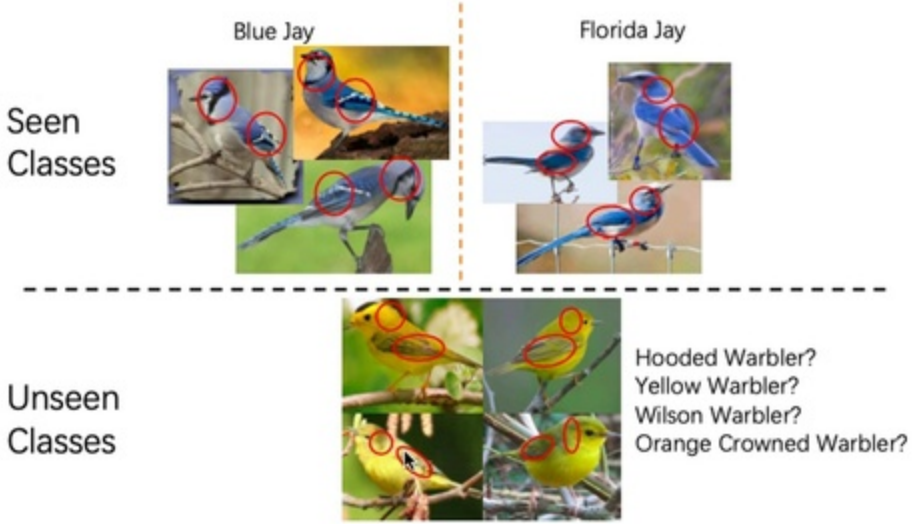
Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors



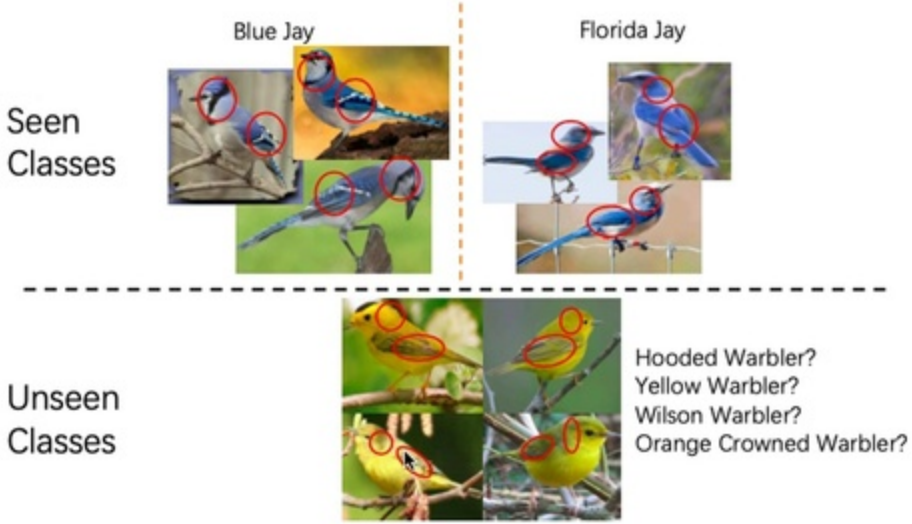
Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors



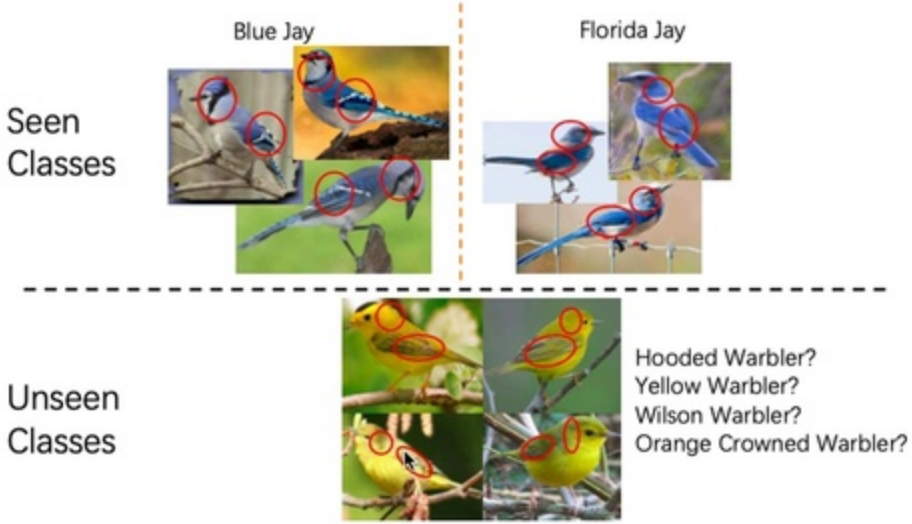
Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors



Trade-off between intra-class compactness and inter-class separability.

- Intra-class compactness: risk of filtering out useful factors (for open-set classification)
- Inter-class separability: risk of introducing nuisance factors



Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

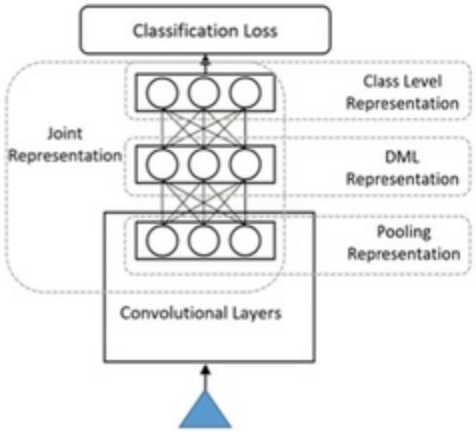
1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).

Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).

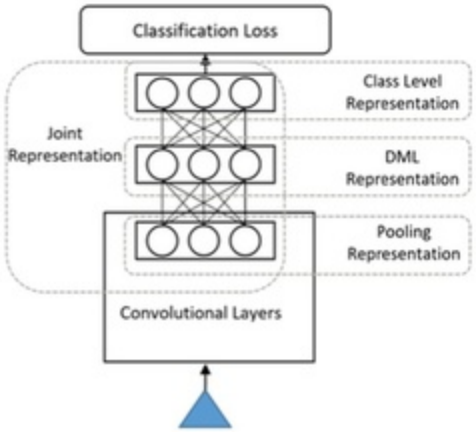


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).

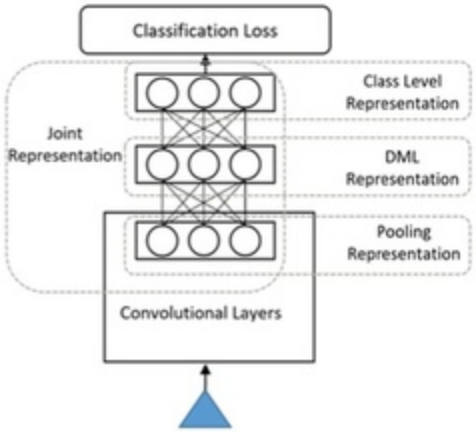


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).

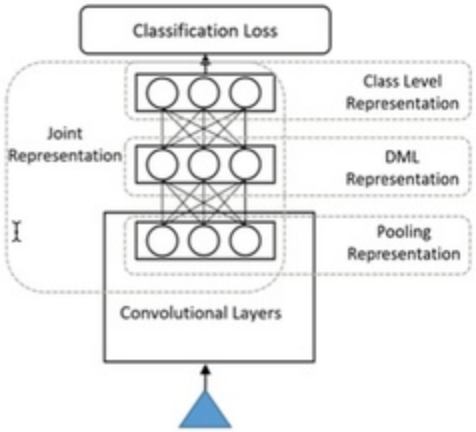


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).

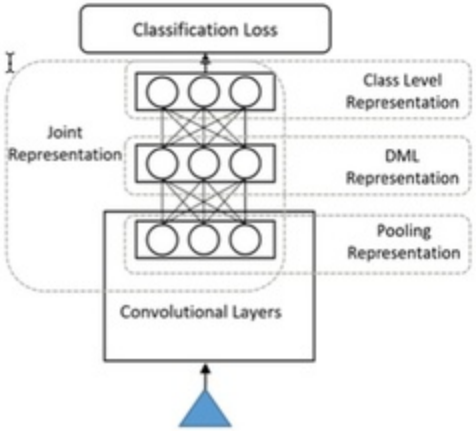


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).

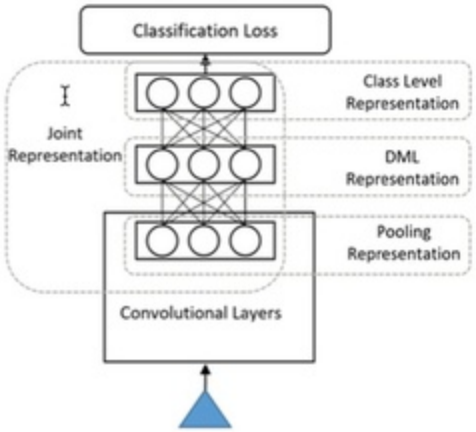


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).

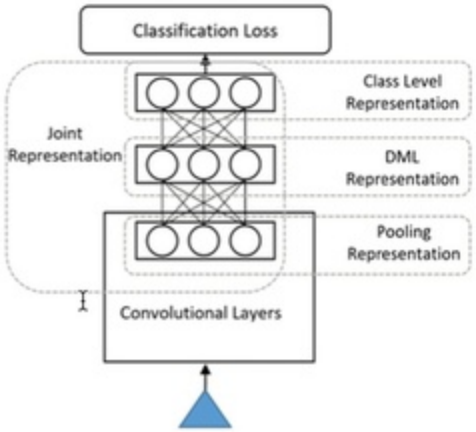


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).
3. We propose a framework distance metric learning with joint representation diversification (JRD).

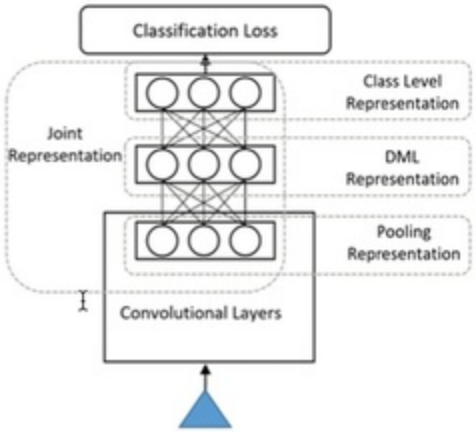


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).
3. We propose a framework distance metric learning with joint representation diversification (JRD).

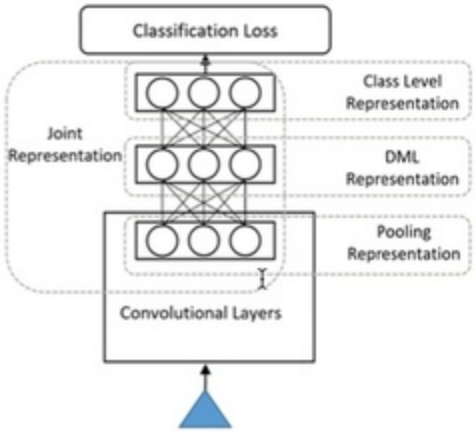


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).
3. We propose a framework distance metric learning with joint representation diversification (JRD).

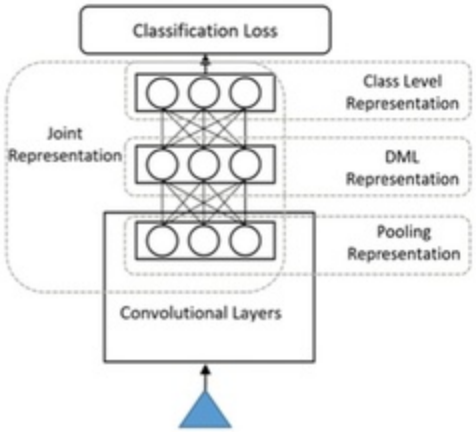


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).
3. We propose a framework distance metric learning with joint representation diversification (JRD).

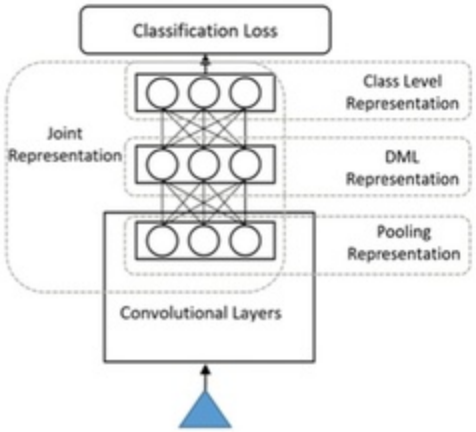


Motivation

- Is it possible to find a better balance point between intra-class compactness and inter-class separability?
- How to leverage the hierarchical representations of DNNs to improve the DML representation?

Results

1. Additional explicit penalizations on intra-class distances of representations is risky for the classification loss methods (AMSoftmax).
2. Encouraging inter-class separability by penalizing distributional similarities of joint representations is beneficial for the classification loss methods (AMSoftmax).
3. We propose a framework distance metric learning with joint representation diversification (JRD).



Challenge

- How to measure the similarities of joint distributions of representations across multiple layers?

Solution

- Representers of probability measures in the reproducing kernel Hilbert space (RKHS)

Definition 1 (kernel mean embedding).

Let $M_+^1(\mathcal{X})$ be the space of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \Sigma)$. $\mathcal{RKHS}$ is a reproducing kernel Hilbert space with reproducing kernel k. The kernel mean embedding is defined by the mapping,
$\mu : M_+^1(\mathcal{X}) \longrightarrow \mathcal{RKHS}, \quad \mathbb{P} \longmapsto \int k(\cdot, \mathbf{x}) \mathrm{d}\mathbb{P}(\mathbf{x}) \triangleq \mu_{\mathbb{P}}.$

Definition 2 (cross-covariance operator)

Let $M_+^1(\times_{l=1}^L \mathcal{X}^l)$ be the space of all probability measures $\mathbb{P}$ on $\times_{l=1}^L \mathcal{X}^l$. $\otimes_{l=1}^L \mathcal{RKHS}^l = \mathcal{RKHS}^1 \otimes \cdots \otimes \mathcal{RKHS}^L$ is a tensor product space with reproducing kernels $\{k^l\}_{l=1}^L$. The cross-covariance operator is defined by the mapping, $\mathcal{C}_{\mathbf{X}^{1:L}} : M_+^1(\times_{l=1}^L \mathcal{X}^l) \longrightarrow \otimes_{l=1}^L \mathcal{RKHS}^l$,
$\mathbb{P} \mapsto \int_{\times_{l=1}^L \mathbf{x}^l} (\otimes_{l=1}^L k^l(\cdot, \mathbf{x}^l)) \mathrm{d}\mathbb{P}(\mathbf{x}^1, \ldots, \mathbf{x}^L) \triangleq \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}).$

Challenge

- How to measure the similarities of joint distributions of representations across multiple layers?

Solution

- Representers of probability measures in the reproducing kernel Hilbert space (RKHS)

Definition 1 (kernel mean embedding).

Let $M_+^1(\mathcal{X})$ be the space of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \Sigma)$. $\mathcal{RKHS}$ is a reproducing kernel Hilbert space with reproducing kernel k. The kernel mean embedding is defined by the mapping,
$\mu : M_+^1(\mathcal{X}) \longrightarrow \mathcal{RKHS}, \quad \mathbb{P} \longmapsto \int k(\cdot, \mathbf{x}) \mathrm{d}\mathbb{P}(\mathbf{x}) \triangleq \mu_{\mathbb{P}}.$

Definition 2 (cross-covariance operator)

Let $M_+^1(\times_{l=1}^L \mathcal{X}^l)$ be the space of all probability measures $\mathbb{P}$ on $\times_{l=1}^L \mathcal{X}^l$. $\otimes_{l=1}^L \mathcal{RKHS}^l = \mathcal{RKHS}^1 \otimes \cdots \otimes \mathcal{RKHS}^L$ is a tensor product space with reproducing kernels $\{k^l\}_{l=1}^L$. The cross-covariance operator is defined by the mapping, $\mathcal{C}_{\mathbf{X}^{1:L}} : M_+^1(\times_{l=1}^L \mathcal{X}^l) \longrightarrow \otimes_{l=1}^L \mathcal{RKHS}^l$,
$\mathbb{P} \mapsto \int_{\times_{l=1}^L \mathbf{x}^l} (\otimes_{l=1}^L k^l(\cdot, \mathbf{x}^l)) \mathrm{d}\mathbb{P}(\mathbf{x}^1, \ldots, \mathbf{x}^L) \triangleq \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}).$

Challenge

- How to measure the similarities of joint distributions of representations across multiple layers?

Solution

- Representers of probability measures in the reproducing kernel Hilbert space (RKHS)

Definition 1 (kernel mean embedding).

Let $M_+^1(\mathcal{X})$ be the space of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \Sigma)$. $\mathcal{RKHS}$ is a reproducing kernel Hilbert space with reproducing kernel k. The kernel mean embedding is defined by the mapping,
$\mu : M_+^1(\mathcal{X}) \longrightarrow \mathcal{RKHS}, \quad \mathbb{P} \longmapsto \int k(\cdot, \mathbf{x}) \mathrm{d}\mathbb{P}(\mathbf{x}) \triangleq \mu_{\mathbb{P}}.$

Definition 2 (cross-covariance operator)

Let $M_+^1(\times_{l=1}^L \mathcal{X}^l)$ be the space of all probability measures $\mathbb{P}$ on $\times_{l=1}^L \mathcal{X}^l$. $\otimes_{l=1}^L \mathcal{RKHS}^l = \mathcal{RKHS}^1 \otimes \cdots \otimes \mathcal{RKHS}^L$ is a tensor product space with reproducing kernels $\{k^l\}_{l=1}^L$. The cross-covariance operator is defined by the mapping, $\mathcal{C}_{\mathbf{X}^{1:L}} : M_+^1(\times_{l=1}^L \mathcal{X}^l) \longrightarrow \otimes_{l=1}^L \mathcal{RKHS}^l$,
$\mathbb{P} \mapsto \int_{\times_{l=1}^L \mathbf{x}^l} (\otimes_{l=1}^L k^l(\cdot, \mathbf{x}^l)) \mathrm{d}\mathbb{P}(\mathbf{x}^1, \ldots, \mathbf{x}^L) \triangleq \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}).$

Challenge

- How to measure the similarities of joint distributions of representations across multiple layers?

Solution

- Representers of probability measures in the reproducing kernel Hilbert space (RKHS)

Definition 1 (kernel mean embedding).

Let $M_+^1(\mathcal{X})$ be the space of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \Sigma)$. $\mathcal{RKHS}$ is a reproducing kernel Hilbert space with reproducing kernel k. The kernel mean embedding is defined by the mapping,
$\mu : M_+^1(\mathcal{X}) \longrightarrow \mathcal{RKHS}, \quad \mathbb{P} \longmapsto \int k(\cdot, \mathbf{x}) \mathrm{d}\mathbb{P}(\mathbf{x}) \triangleq \mu_{\mathbb{P}}.$

Definition 2 (cross-covariance operator)

Let $M_+^1(\times_{l=1}^L \mathcal{X}^l)$ be the space of all probability measures $\mathbb{P}$ on $\times_{l=1}^L \mathcal{X}^l$. $\otimes_{l=1}^L \mathcal{RKHS}^l = \mathcal{RKHS}^1 \otimes \cdots \otimes \mathcal{RKHS}^L$ is a tensor product space with reproducing kernels $\{k^l\}_{l=1}^L$. The cross-covariance operator is defined by the mapping, $\mathcal{C}_{\mathbf{X}^{1:L}} : M_+^1(\times_{l=1}^L \mathcal{X}^l) \longrightarrow \otimes_{l=1}^L \mathcal{RKHS}^l$,
$\mathbb{P} \mapsto \int_{\times_{l=1}^L \mathbf{x}^l} (\otimes_{l=1}^L k^l(\cdot, \mathbf{x}^l)) \mathrm{d}\mathbb{P}(\mathbf{x}^1, \ldots, \mathbf{x}^L) \triangleq \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}).$

Challenge

- How to measure the similarities of joint distributions of representations across multiple layers?

Solution

- Representers of probability measures in the reproducing kernel Hilbert space (RKHS)

Definition 1 (kernel mean embedding).

Let $M_+^1(\mathcal{X})$ be the space of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \Sigma)$. $\mathcal{RKHS}$ is a reproducing kernel Hilbert space with reproducing kernel k. The kernel mean embedding is defined by the mapping,
$\mu : M_+^1(\mathcal{X}) \longrightarrow \mathcal{RKHS}, \quad \mathbb{P} \longmapsto \int k(\cdot, \mathbf{x}) \mathrm{d}\mathbb{P}(\mathbf{x}) \triangleq \mu_{\mathbb{P}}.$

Definition 2 (cross-covariance operator)

Let $M_+^1(\times_{l=1}^L \mathcal{X}^l)$ be the space of all probability measures $\mathbb{P}$ on $\times_{l=1}^L \mathcal{X}^l$. $\otimes_{l=1}^L \mathcal{RKHS}^l = \mathcal{RKHS}^1 \otimes \cdots \otimes \mathcal{RKHS}^L$ is a tensor product space with reproducing kernels $\{k^l\}_{l=1}^L$. The cross-covariance operator is defined by the mapping, $\mathcal{C}_{\mathbf{X}^{1:L}} : M_+^1(\times_{l=1}^L \mathcal{X}^l) \longrightarrow \otimes_{l=1}^L \mathcal{RKHS}^l$,
$\mathbb{P} \mapsto \int_{\times_{l=1}^L \mathbf{x}^l} (\otimes_{l=1}^L k^l(\cdot, \mathbf{x}^l)) \mathrm{d}\mathbb{P}(\mathbf{x}^1, \ldots, \mathbf{x}^L) \triangleq \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}).$

Challenge

- How to measure the similarities of joint distributions of representations across multiple layers?

Solution

- Representers of probability measures in the reproducing kernel Hilbert space (RKHS)

Definition 1 (kernel mean embedding).

Let $M_+^1(\mathcal{X})$ be the space of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \Sigma)$. $\mathcal{RKHS}$ is a reproducing kernel Hilbert space with reproducing kernel k. The kernel mean embedding is defined by the mapping,
$\mu : M_+^1(\mathcal{X}) \longrightarrow \mathcal{RKHS}, \quad \mathbb{P} \longmapsto \int k(\cdot, \mathbf{x}) \mathrm{d}\mathbb{P}(\mathbf{x}) \triangleq \mu_{\mathbb{P}}.$

Definition 2 (cross-covariance operator)

Let $M_+^1(\times_{l=1}^L \mathcal{X}^l)$ be the space of all probability measures $\mathbb{P}$ on $\times_{l=1}^L \mathcal{X}^l$. $\otimes_{l=1}^L \mathcal{RKHS}^l = \mathcal{RKHS}^1 \otimes \cdots \otimes \mathcal{RKHS}^L$ is a tensor product space with reproducing kernels $\{k^l\}_{l=1}^L$. The cross-covariance operator is defined by the mapping, $\mathcal{C}_{\mathbf{X}^{1:L}} : M_+^1(\times_{l=1}^L \mathcal{X}^l) \longrightarrow \otimes_{l=1}^L \mathcal{RKHS}^l$,
$\mathbb{P} \mapsto \int_{\times_{l=1}^L \mathbf{x}^l} (\otimes_{l=1}^L k^l(\cdot, \mathbf{x}^l)) \mathrm{d}\mathbb{P}(\mathbf{x}^1, \ldots, \mathbf{x}^L) \triangleq \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}).$

Challenge

- How to measure the similarities of joint distributions of representations across multiple layers?

Solution

- Representers of probability measures in the reproducing kernel Hilbert space (RKHS)

Definition 1 (kernel mean embedding).

Let $M_+^1(\mathcal{X})$ be the space of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \Sigma)$. $\mathcal{RKHS}$ is a reproducing kernel Hilbert space with reproducing kernel k. The kernel mean embedding is defined by the mapping,
$\mu : M_+^1(\mathcal{X}) \longrightarrow \mathcal{RKHS}, \quad \mathbb{P} \longmapsto \int k(\cdot, \mathbf{x}) \mathrm{d}\mathbb{P}(\mathbf{x}) \triangleq \mu_{\mathbb{P}}$.

Definition 2 (cross-covariance operator)

Let $M_+^1(\times_{l=1}^L \mathcal{X}^l)$ be the space of all probability measures $\mathbb{P}$ on $\times_{l=1}^L \mathcal{X}^l$. $\otimes_{l=1}^L \mathcal{RKHS}^l = \mathcal{RKHS}^1 \otimes \cdots \otimes \mathcal{RKHS}^L$ is a tensor product space with reproducing kernels $\{k^l\}_{l=1}^L$. The cross-covariance operator is defined by the mapping, $\mathcal{C}_{\mathbf{X}^{1:L}} : M_+^1(\times_{l=1}^L \mathcal{X}^l) \longrightarrow \otimes_{l=1}^L \mathcal{RKHS}^l$,
$\mathbb{P} \mapsto \int_{\times_{l=1}^L \mathbf{X}^l} (\otimes_{l=1}^L k^l(\cdot, \mathbf{x}^l)) \mathrm{d}\mathbb{P}(\mathbf{x}^1, \ldots, \mathbf{x}^L) \triangleq \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$.

Challenge

- How to measure the similarities of joint distributions of representations across multiple layers?

Solution

- Representers of probability measures in the reproducing kernel Hilbert space (RKHS)

Definition 1 (kernel mean embedding).

Let $M_+^1(\mathcal{X})$ be the space of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \Sigma)$. $\mathcal{RKHS}$ is a reproducing kernel Hilbert space with reproducing kernel k. The kernel mean embedding is defined by the mapping,
$\mu : M_+^1(\mathcal{X}) \longrightarrow \mathcal{RKHS}, \quad \mathbb{P} \longmapsto \int k(\cdot, \mathbf{x}) \mathrm{d}\mathbb{P}(\mathbf{x}) \triangleq \mu_{\mathbb{P}}$.

Definition 2 (cross-covariance operator)

Let $M_+^1(\times_{l=1}^L \mathcal{X}^l)$ be the space of all probability measures $\mathbb{P}$ on $\times_{l=1}^L \mathcal{X}^l$. $\otimes_{l=1}^L \mathcal{RKHS}^l = \mathcal{RKHS}^1 \otimes \cdots \otimes \mathcal{RKHS}^L$ is a tensor product space with reproducing kernels $\{k^l\}_{l=1}^L$. The cross-covariance operator is defined by the mapping, $\mathcal{C}_{\mathbf{X}^{1:L}} : M_+^1(\times_{l=1}^L \mathcal{X}^l) \longrightarrow \otimes_{l=1}^L \mathcal{RKHS}^l$,
$\mathbb{P} \mapsto \int_{\times_{l=1}^L \mathbf{X}^l} (\otimes_{l=1}^L k^l(\cdot, \mathbf{x}^l)) \mathrm{d}\mathbb{P}(\mathbf{x}^1, \ldots, \mathbf{x}^L) \triangleq \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$.

Challenge

- How to measure the similarities of joint distributions of representations across multiple layers?

Solution

- Representers of probability measures in the reproducing kernel Hilbert space (RKHS)

Definition 1 (kernel mean embedding).

Let $M_+^1(\mathcal{X})$ be the space of all probability measures $\mathbb{P}$ on a measurable space $(\mathcal{X}, \Sigma)$. $\mathcal{RKHS}$ is a reproducing kernel Hilbert space with reproducing kernel k. The kernel mean embedding is defined by the mapping,
$\mu : M_+^1(\mathcal{X}) \longrightarrow \mathcal{RKHS}, \quad \mathbb{P} \longmapsto \int k(\cdot, \mathbf{x}) \mathrm{d}\mathbb{P}(\mathbf{x}) \triangleq \mu_{\mathbb{P}}$.

Definition 2 (cross-covariance operator)

Let $M_+^1(\times_{l=1}^L \mathcal{X}^l)$ be the space of all probability measures $\mathbb{P}$ on $\times_{l=1}^L \mathcal{X}^l$. $\otimes_{l=1}^L \mathcal{RKHS}^l = \mathcal{RKHS}^1 \otimes \cdots \otimes \mathcal{RKHS}^L$ is a tensor product space with reproducing kernels $\{k^l\}_{l=1}^L$. The cross-covariance operator is defined by the mapping, $\mathcal{C}_{\mathbf{X}^{1:L}} : M_+^1(\times_{l=1}^L \mathcal{X}^l) \longrightarrow \otimes_{l=1}^L \mathcal{RKHS}^l$,
$\mathbb{P} \mapsto \int_{\times_{l=1}^L \mathbf{X}^l} (\otimes_{l=1}^L k^l(\cdot, \mathbf{x}^l)) \mathrm{d}\mathbb{P}(\mathbf{x}^1, \ldots, \mathbf{x}^L) \triangleq \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 3 (joint representation similarity)

Suppose that $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L)$ and $\mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L)$ are probability measures on $\times_{l=1}^{L} \mathcal{X}^l$. Given L reproducing kernels $\{k^l\}_{l=1}^{L}$, the joint representation similarity between $\mathbb{P}$ and $\mathbb{Q}$ is defined as the inner product of $\mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P})$ and $\mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q})$ in $\otimes_{l=1}^{L} \mathcal{RKHS}^l$, i.e.,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) \triangleq \langle \mathcal{C}_{\mathbf{X}^{1:L}}(\mathbb{P}), \mathcal{C}_{\mathbf{X}'^{1:L}}(\mathbb{Q}) \rangle_{\otimes_{l=1}^{L} \mathcal{RKHS}^l} \tag{1}$$

Proposition 1 (interpretation for translation invariant kernels)

Suppose that $\{k^l(\mathbf{x}, \mathbf{x}') = \psi^l(\mathbf{x} - \mathbf{x}')\}_{l=1}^{L}$ on $\mathbb{R}^d$ are bounded, continuous reproducing kernels. Let $P^l \triangleq \mathbb{P}(\mathbf{X}^l | \mathbf{X}^{1:l-1})$ for $l = 1, \ldots, L$ with $P^1 = \mathbb{P}(\mathbf{X}^1)$. Then for any $\mathbb{P}(\mathbf{X}^1, \ldots, \mathbf{X}^L), \mathbb{Q}(\mathbf{X}'^1, \ldots, \mathbf{X}'^L) \in M_+^1(\times_{l=1}^{L} \mathcal{X}^l)$,

$$\mathcal{S}_{JRS}(\mathbb{P}, \mathbb{Q}) = \prod_{l=1}^{L} \langle \phi_{P^l}(\omega), \phi_{Q^l}(\omega) \rangle_{L^2(\mathbb{R}^d, \Lambda^l)}, \tag{2}$$

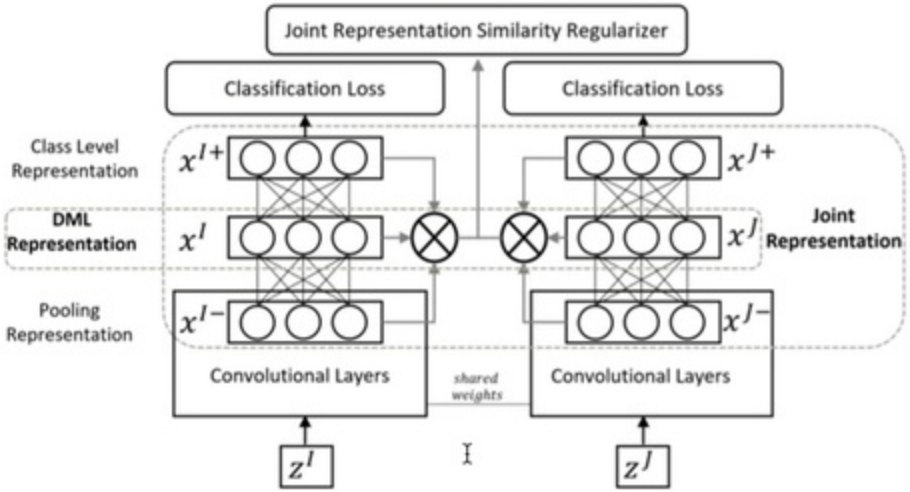
where $\phi_{P^l}(\omega)$ and $\phi_{Q^l}(\omega)$ are the characteristic functions of the distributions P^l and Q^l, and Λ^l is a (normalized) non-negative Borel measure characterized by $\psi^l(\mathbf{x} - \mathbf{x}')$.

Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

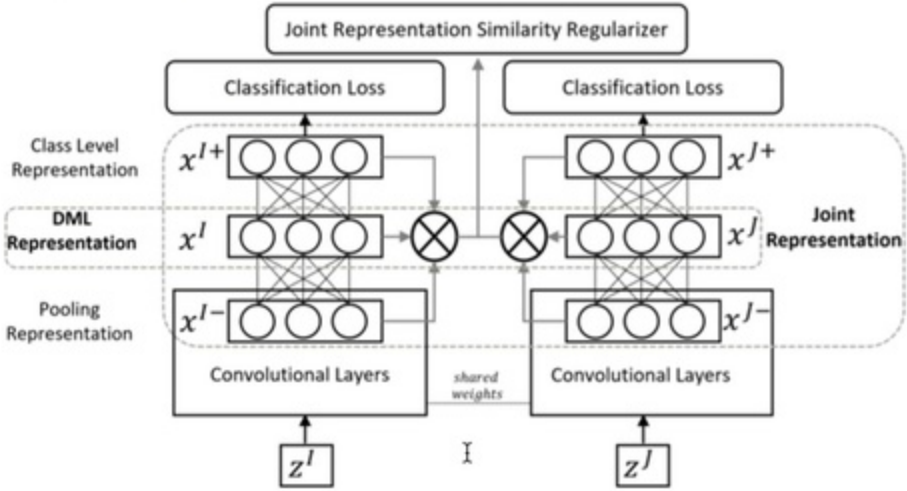


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

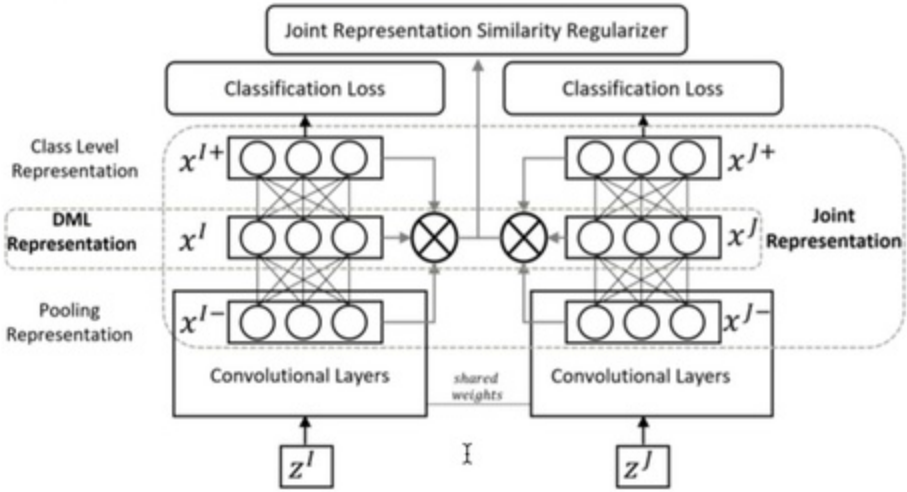


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

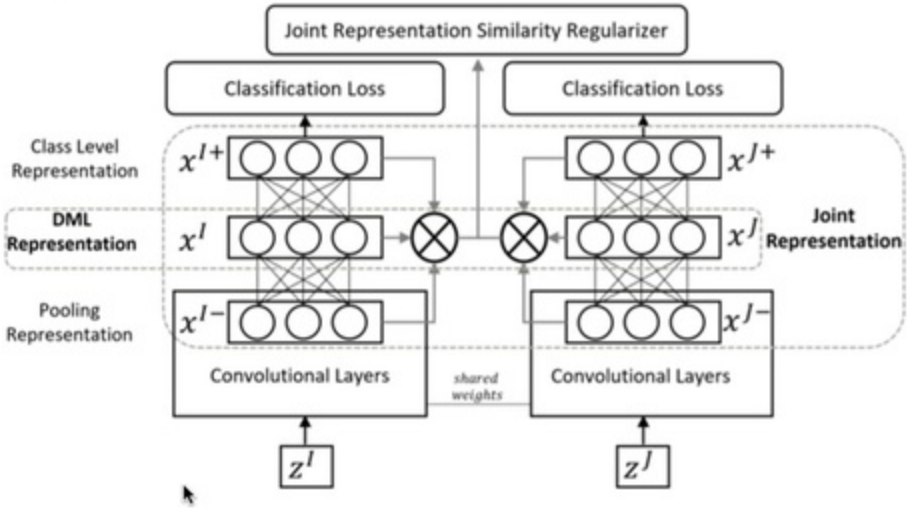


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

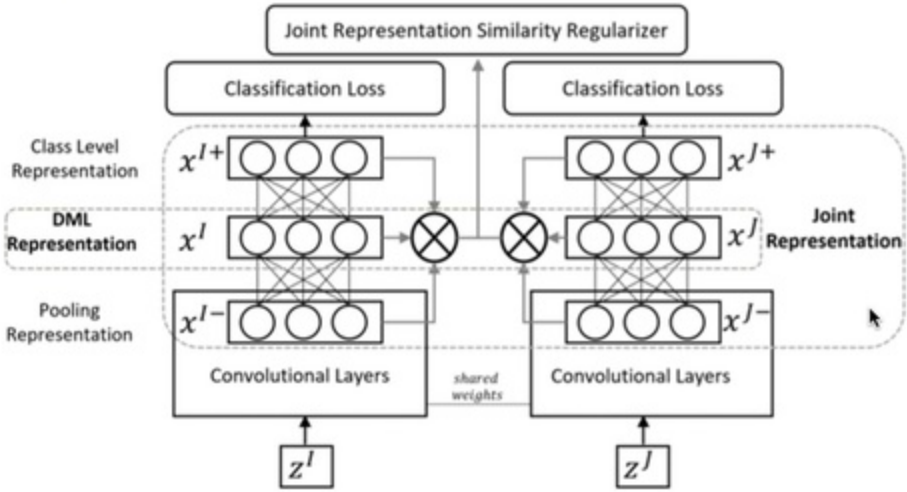


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

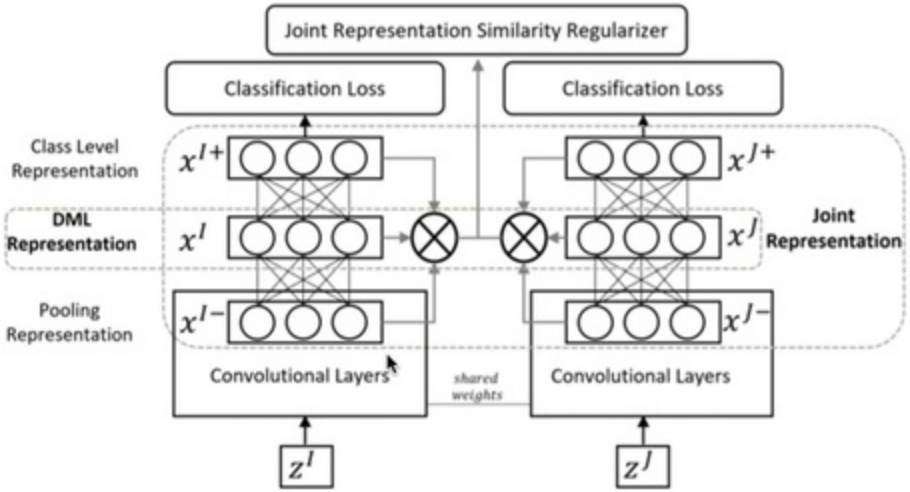


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

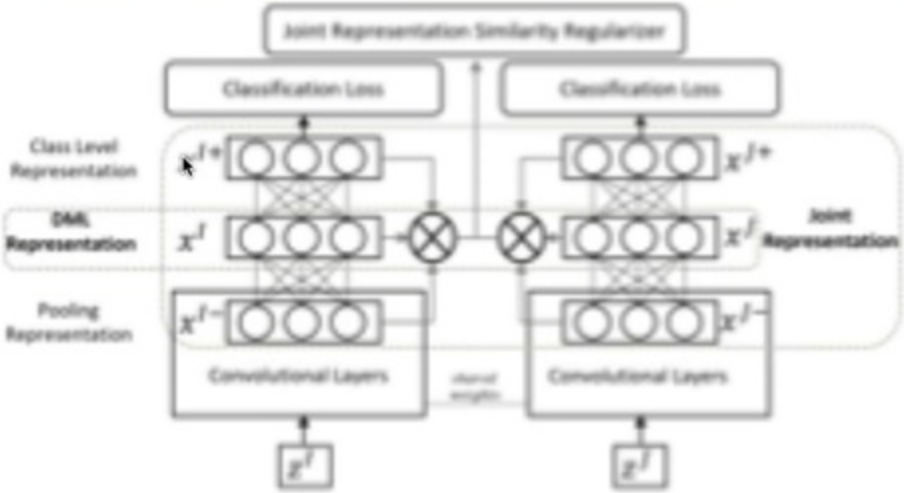


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

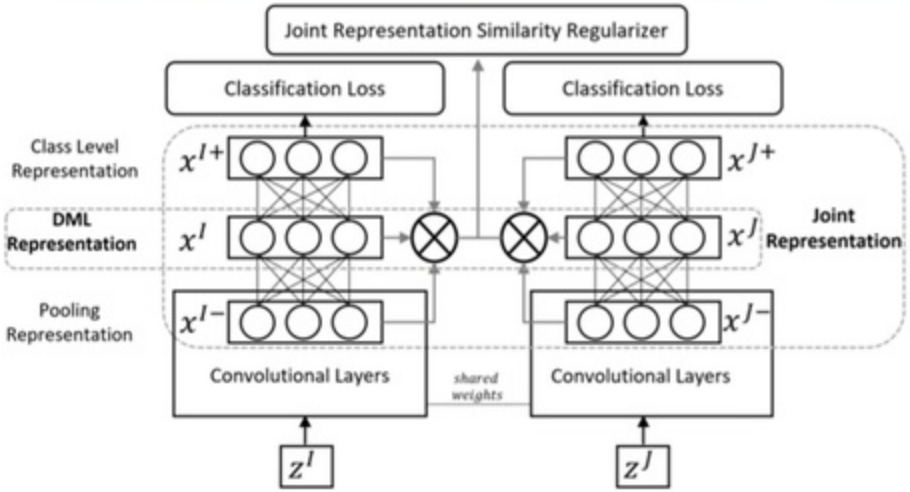


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

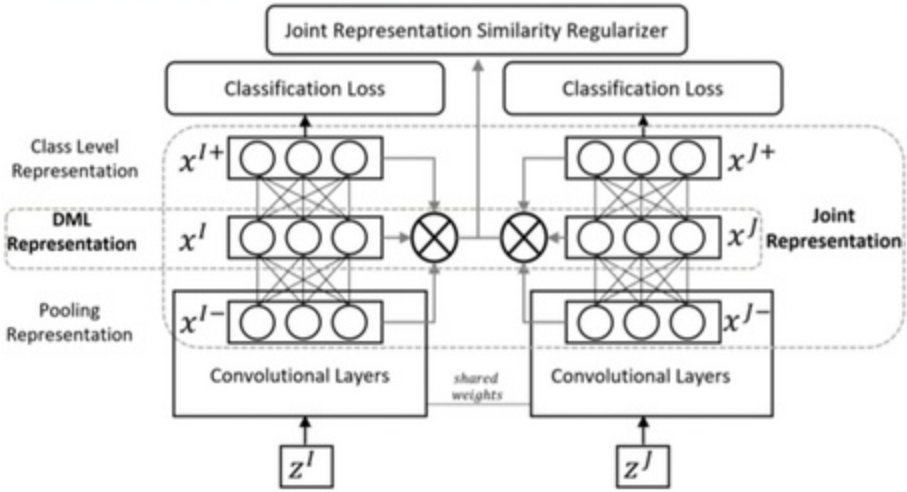


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

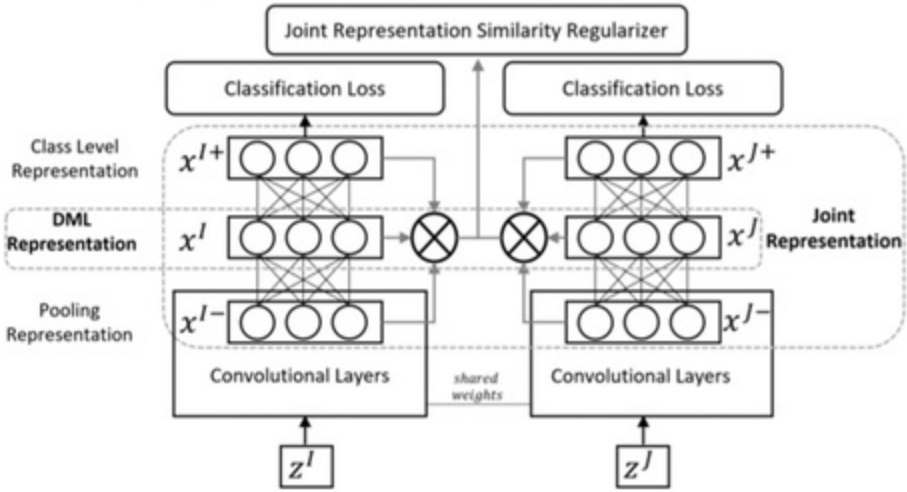


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

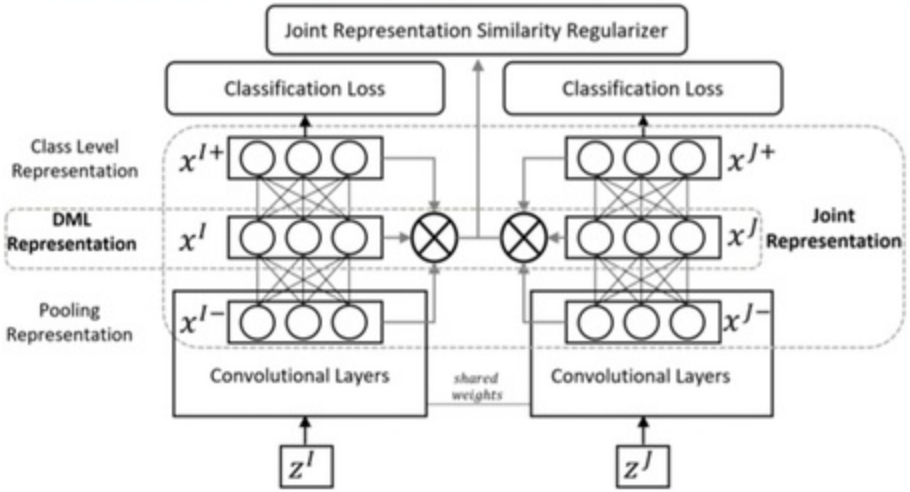


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.

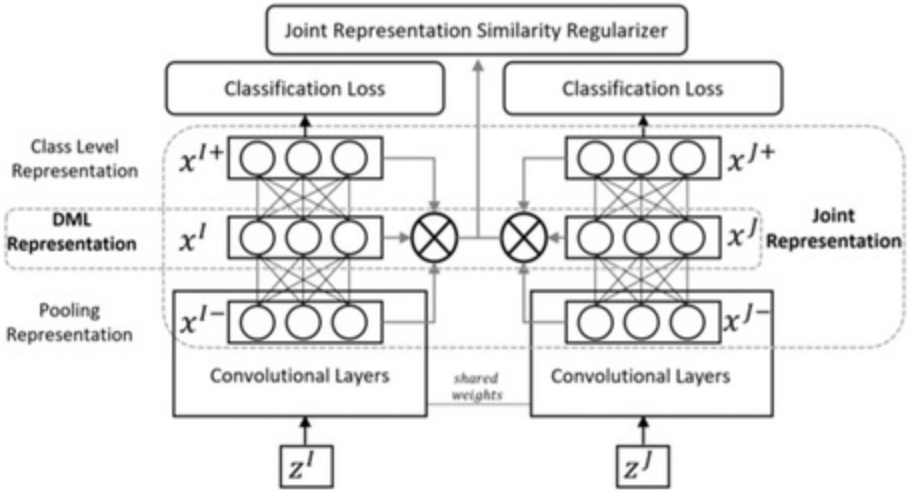


Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.



Training Objective:

$$\mathcal{L}_{JRD} = \mathcal{L}_{AMSoft} + \alpha \frac{1}{N_{pairs}} \mathcal{L}_{JRS}, \quad (4)$$

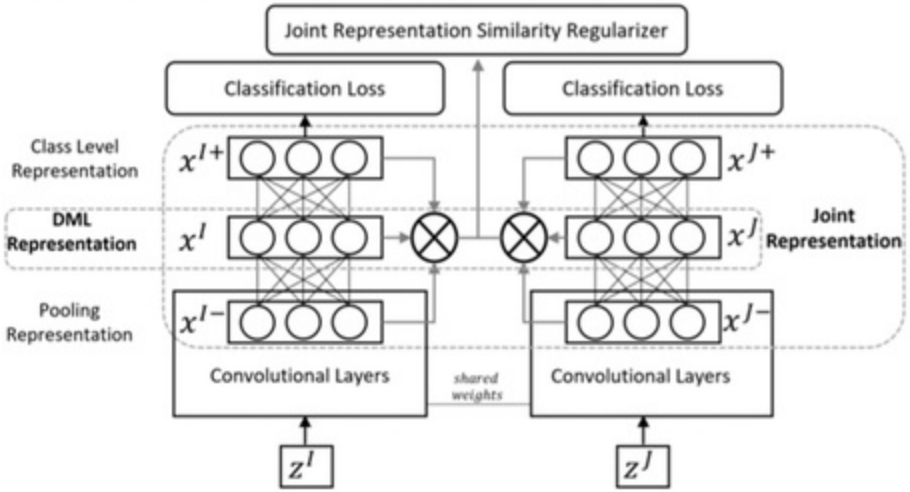
where N_{pairs} denotes the number of pairs of instances from different classes in a mini-batch.

Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.



Training Objective:

$$\mathcal{L}_{JRD} = \mathcal{L}_{AMSoft} + \alpha \frac{1}{N_{pairs}} \mathcal{L}_{JRS}, \quad (4)$$

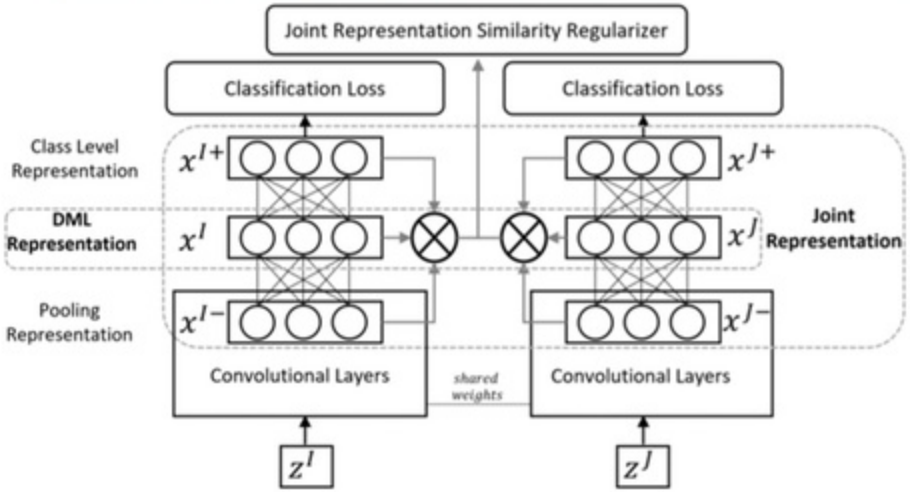
where N_{pairs} denotes the number of pairs of instances from different classes in a mini-batch.

Definition 4 (joint representation similarity regularizer)

Considering $\mathbb{P}(\mathbf{X}^-, \mathbf{X}, \mathbf{X}^+)$, the joint representation similarity regularizer $\mathcal{L}_{JRS}$ penalizes the empirical joint representation similarities for all class pairs, specifically,

$$\mathcal{L}_{JRS} \triangleq \sum_{I \neq J} n^I n^J \widehat{\mathcal{S}_{JRS}}(\mathbb{P}^I, \mathbb{P}^J) = \sum_{I \neq J} \sum_{i=1}^{n^I} \sum_{j=1}^{n^J} k^-(\mathbf{x}_i^{I-}, \mathbf{x}_j^{J-}) k(\mathbf{x}_i^I, \mathbf{x}_j^J) k^+(\mathbf{x}_i^{I+}, \mathbf{x}_j^{J+}), \quad (3)$$

where k^-, k and k^+ are reproducing kernels, I, J are indexes of class, $n^I n^J$ re-weights class pair (I, J) according to its credibility.



Training Objective:

$$\mathcal{L}_{JRD} = \mathcal{L}_{AMSoft} + \alpha \frac{1}{N_{pairs}} \mathcal{L}_{JRS}, \quad (4)$$

where N_{pairs} denotes the number of pairs of instances from different classes in a mini-batch.

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels
  $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K[1]

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Experimental Settings

Datasets

1. CUB-200-2011 (CUB)
2. Cars196 (CARS)
3. Standard Online Products (SOP)

Kernel design

- Mixture of K Gaussian kernels $k(\mathbf{x}, \mathbf{x}') = \frac{1}{K}\sum_{k=1}^{K} exp(\frac{-(\mathbf{x}-\mathbf{x}')^2}{\sigma_k^2})$
- $K = 3$ for $\mathbf{X}^-$ and $\mathbf{X}$, $K' = 1$ for $\mathbf{X}^+$

Evaluation Metric

- Recall@K

Implementation details

- Backbone: Inception-BN
- Embedding size: 512
- Data augmentation: Random crop, random horizontal mirroring
- Optimizer: Adam
- Epochs: 50 for CUB and CARS,80 for SOP
- Learning rate decay: Divided by 10 every 20(40) epochs for CUB and CARS (SOP)
- Mini-batch sampling: Random sampling
- ...

Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

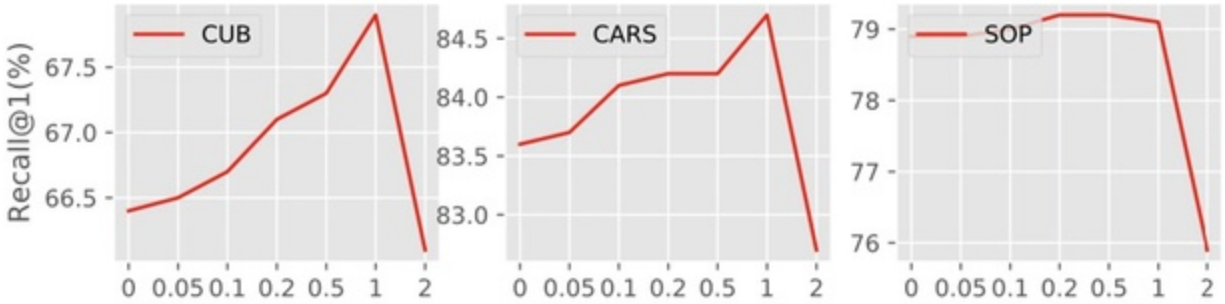
Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

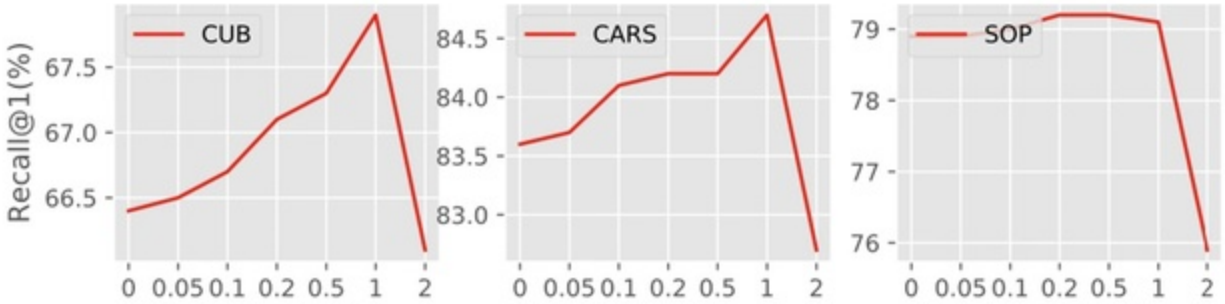
Sensitivity of α



Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

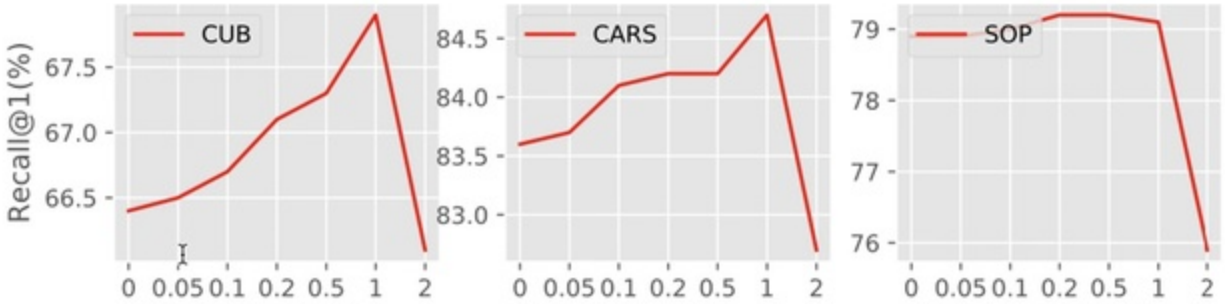
Sensitivity of α



Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

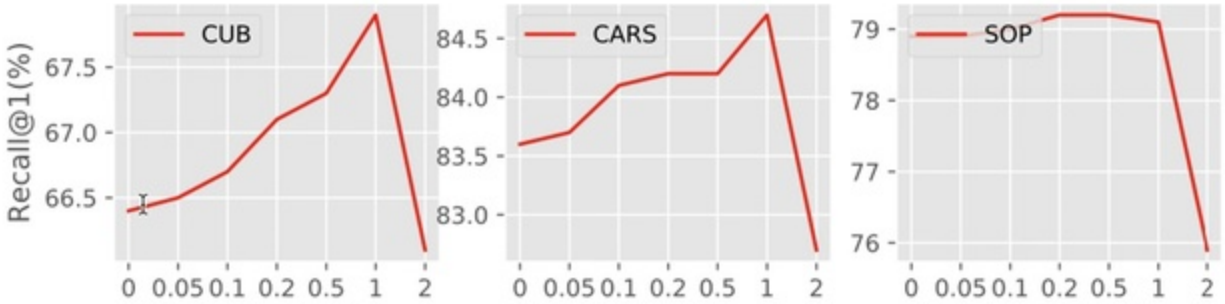
Sensitivity of α



Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

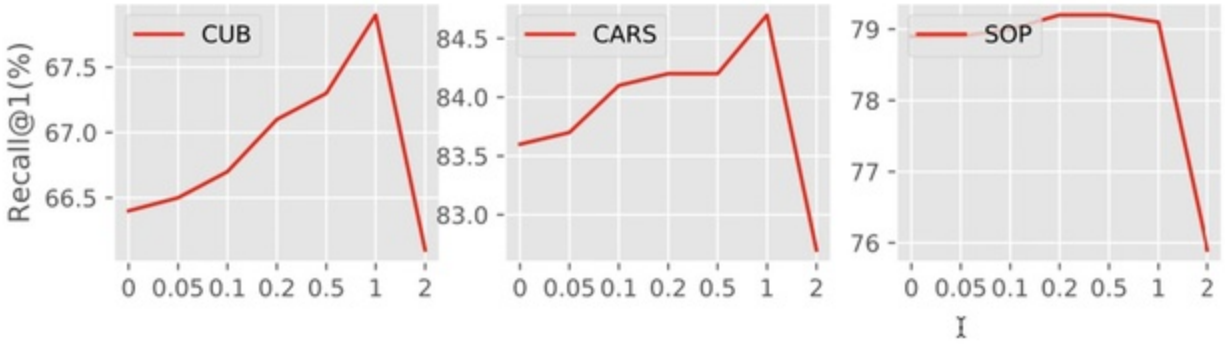
Sensitivity of α



Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

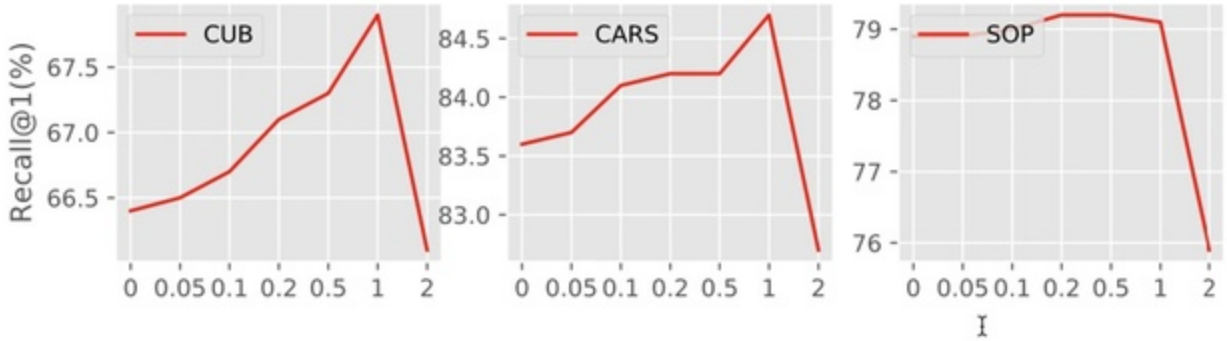
Sensitivity of α



Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

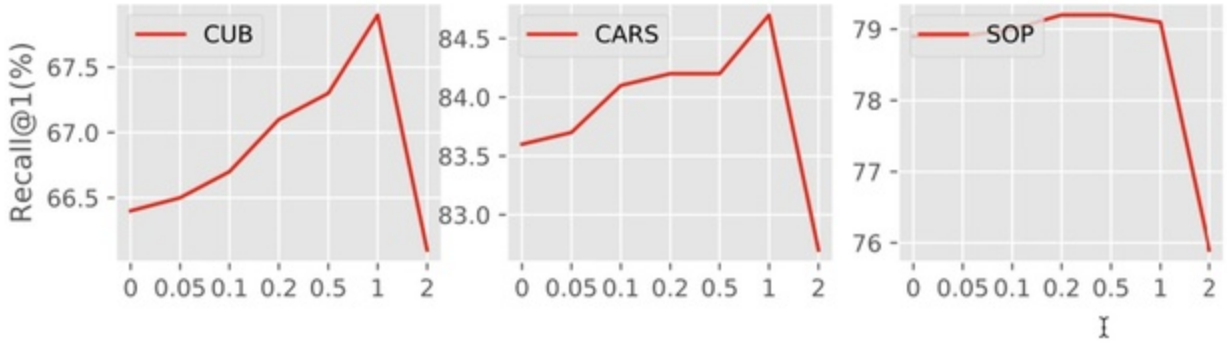
Sensitivity of α



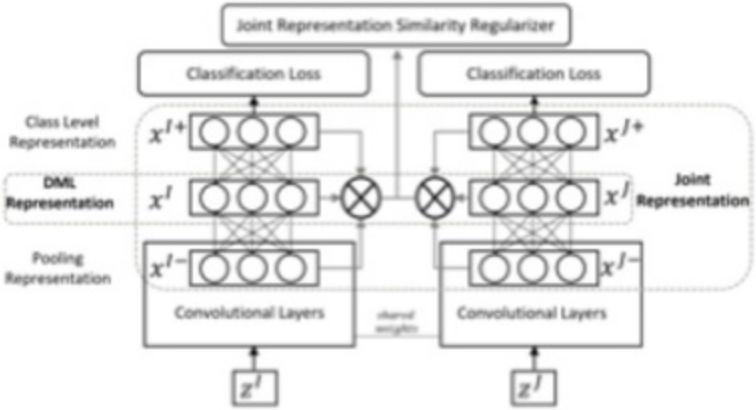
Comparing JRD with 2019 DML baselines

	CUB				CARS				SOP		
Recall@K(%)	1	2	4	8	1	2	4	8	1	10	100
DE_DSP [Duan et al., 2019]	53.6	65.5	76.9	-	72.9	81.6	88.8	-	68.9	84.0	92.6
HDML [Zheng et al., 2019]	53.7	65.7	76.7	85.7	79.1	87.1	92.1	95.5	68.7	83.2	92.4
DAMLRRM [Xu et al., 2019]	55.1	66.5	76.8	85.3	73.5	82.6	89.1	93.5	69.7	85.2	93.2
ECAML [Chen and Deng, 2019a]	55.7	66.5	76.7	85.1	84.5	90.4	93.8	96.6	71.3	85.6	93.6
DeML [Chen and Deng, 2019b]	65.4	75.3	83.7	89.5	**86.3**	**91.2**	94.3	97.0	76.1	88.4	94.9
SoftTriple Loss [Qian et al., 2019]	65.4	76.4	84.5	90.4	84.5	90.7	**94.5**	96.9	78.3	90.3	95.9
MS [Wang et al., 2019]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	**90.5**	**96.0**
JRD	**67.9**	**78.7**	86.2	**91.3**	84.7	90.7	94.4	**97.2**	**79.2**	**90.5**	**96.0**

Sensitivity of α



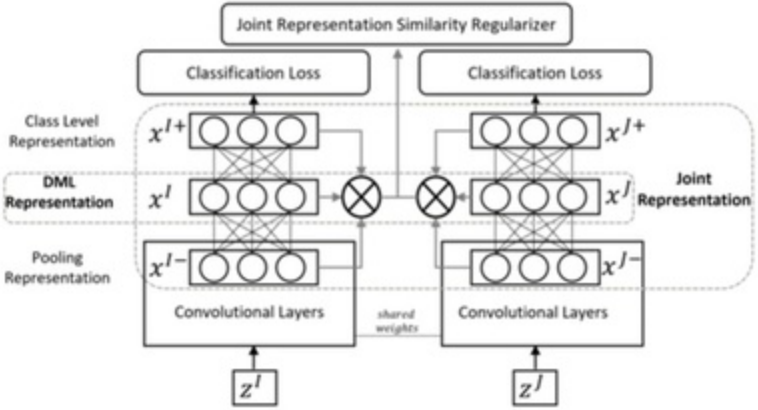
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	9[illegible]4	95.9

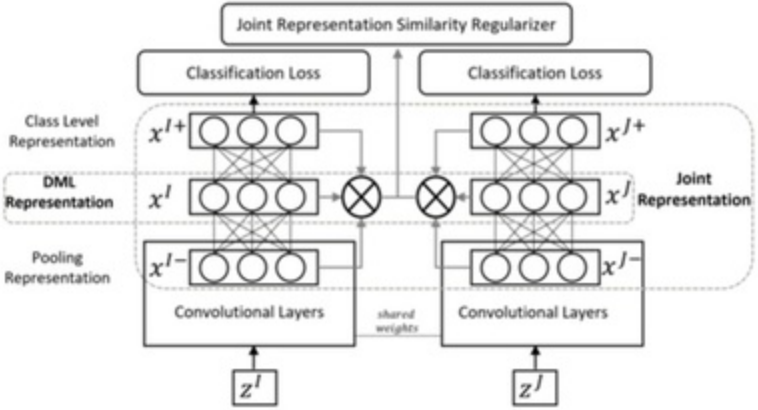
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

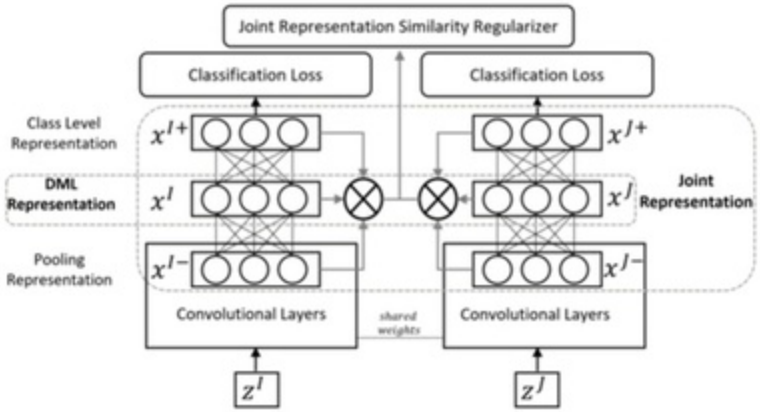
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

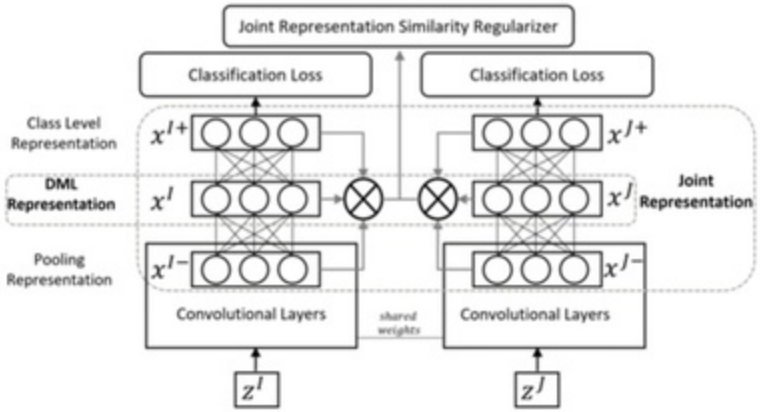
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

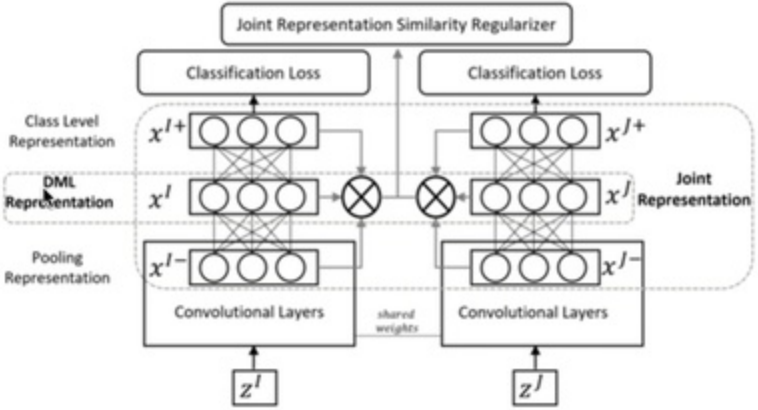
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

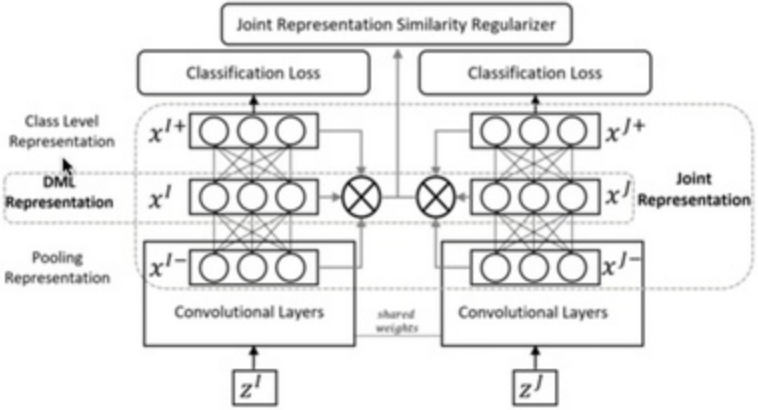
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

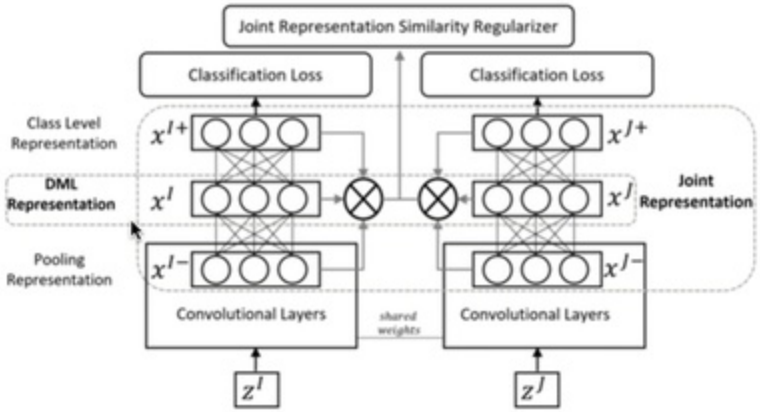
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

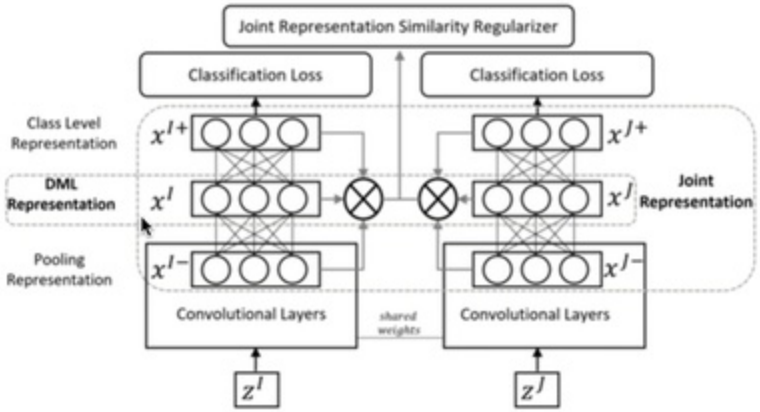
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

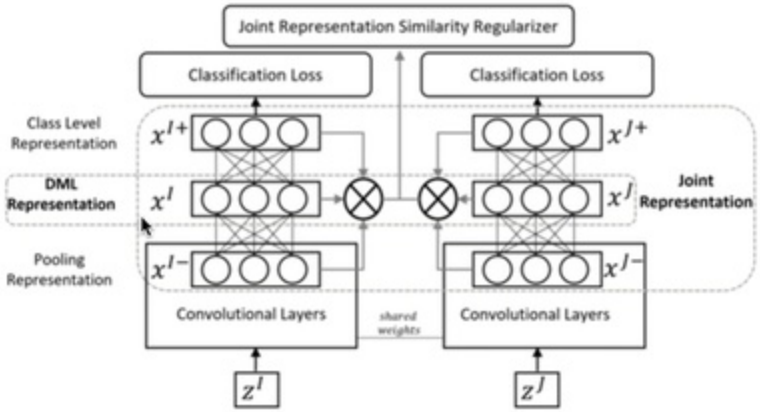
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

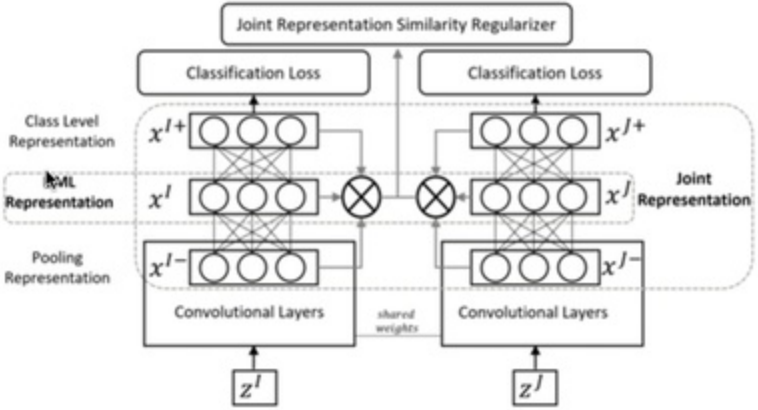
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

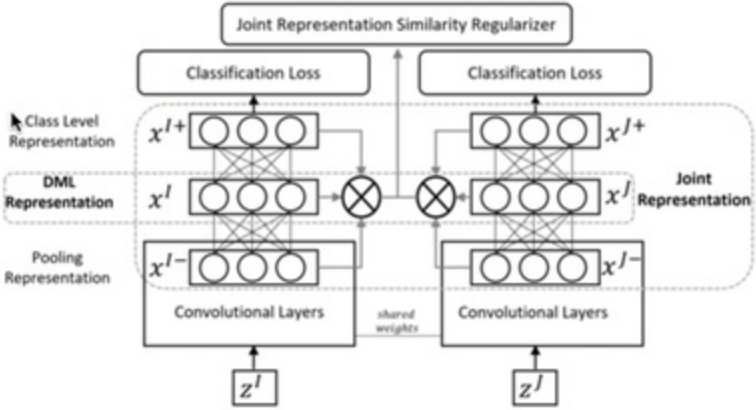
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

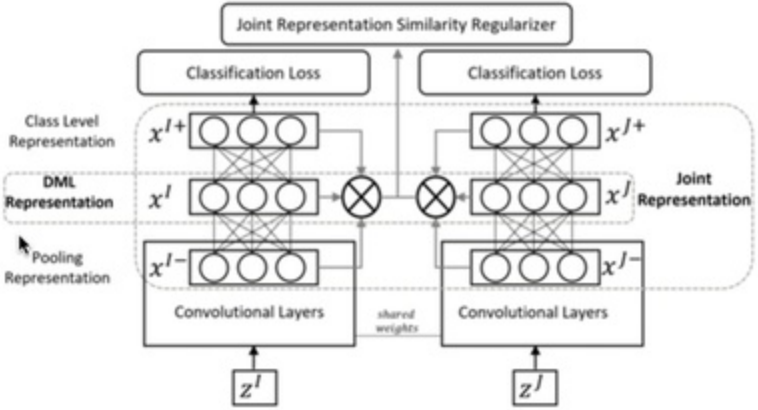
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

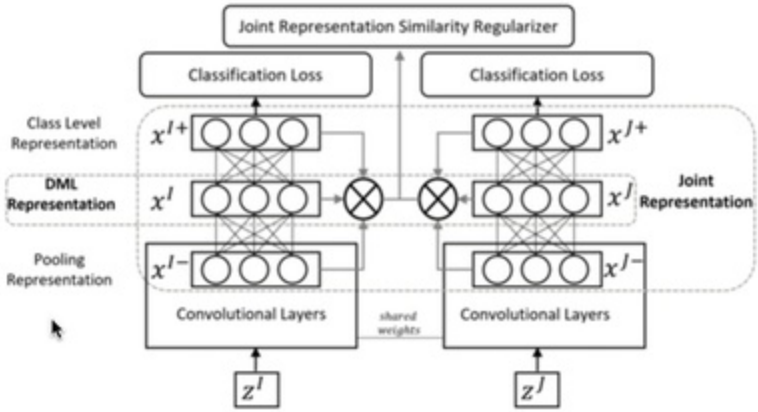
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

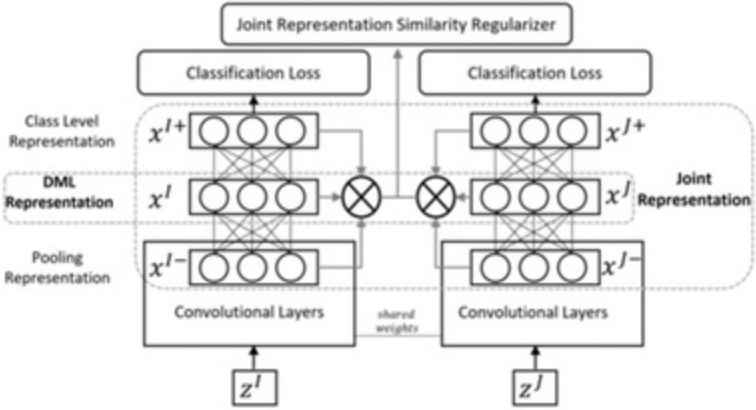
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

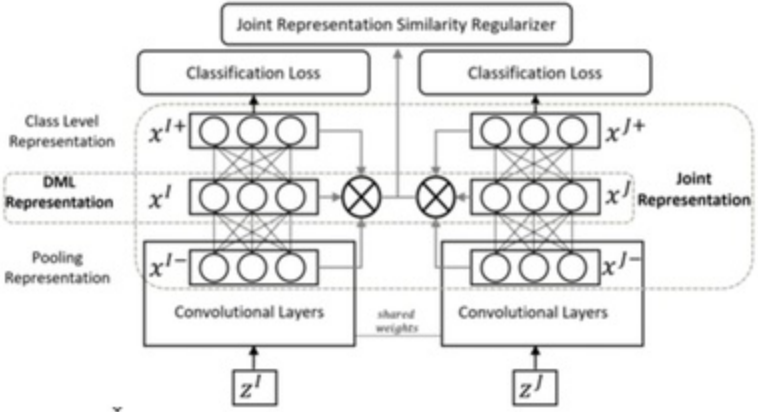
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

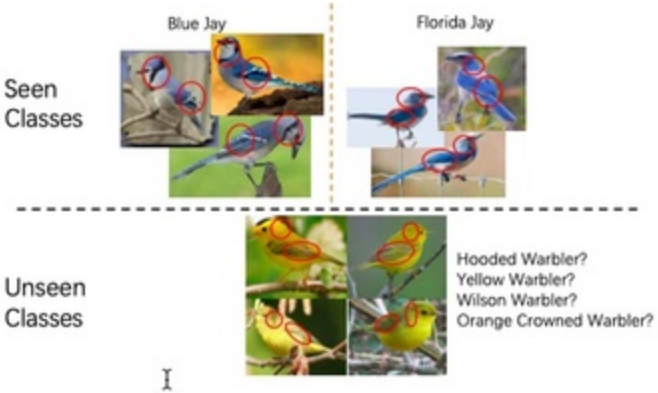
Effects of modeling the joint representation



	CUB			
Recall@K(%)	1	2	4	8
JRD	50.7(1.1)	63.7(1.1)	74.8(1.2)	84.1(1.2)
MRD	49.4(1.1)	62.3(1.1)	74.5(1.2)	83.6(1.2)
JRD-C	48.6(1.5)	61.4(1.4)	73.4(1.5)	83.0(1.4)
JRD-Pooling	49.4(1.2)	62.2(1.0)	74.1(1.2)	83.3(1.0)

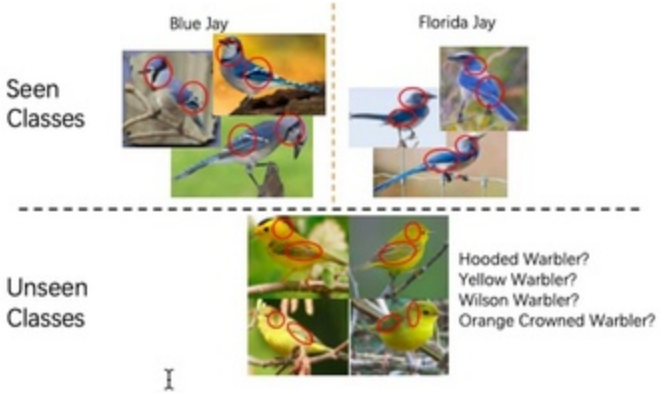
	CARS				SOP		
Recall@K(%)	1	2	4	8	1	10	100
JRD	61.2(1.3)	72.6(0.9)	82.2(0.6)	89.2(0.7)	79.2	90.5	96.0
MRD	59.8(1.3)	71.5(1.2)	80.6(0.9)	88.0(0.9)	78.8	90.4	95.9
JRD-C	58.5(1.5)	69.6(1.3)	79.1(0.7)	86.6(0.9)	77.7	89.8	95.6
JRD-Pooling	59.1(1.5)	70.7(1.2)	80.3(0.5)	87.7(0.6)	79.0	90.4	95.9

Explicit penalization on intra-class distances



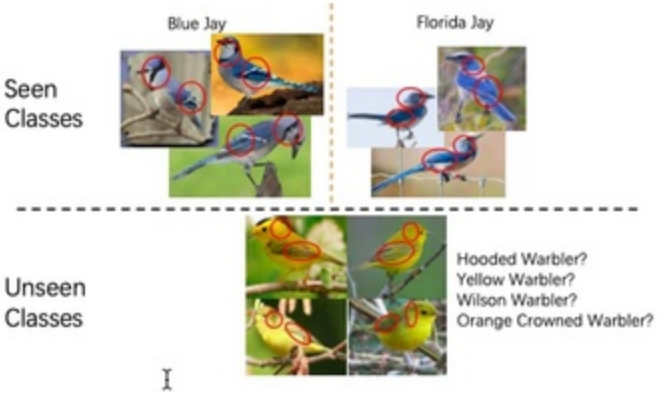
$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i},\mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i}-\mathbf{x}^{l}_{j})^{2}} \quad (5)$$

Explicit penalization on intra-class distances



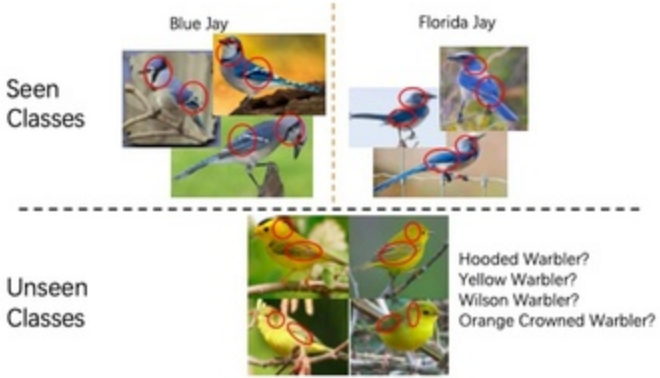
$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$

Explicit penalization on intra-class distances



$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$

Explicit penalization on intra-class distances



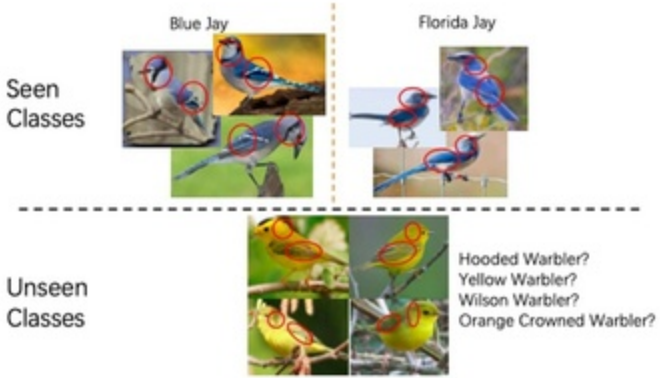
$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{x^{l}_{i}, x^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(x^{l}_{i} - x^{l}_{j})^{2}} \quad (5)$$

Theorem 1 [Ben-David et al., 2010]

Let $\mathcal{H}$ be a hypothesis space. Denote by ϵ_s and ϵ_u the generalization errors on $\mathcal{D}_s$ and $\mathcal{D}_u$, then for every $h \in \mathcal{H}$:

$$\epsilon_u(h) \leq \epsilon_s(h) + \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_u) + \lambda. \quad (6)$$

Explicit penalization on intra-class distances



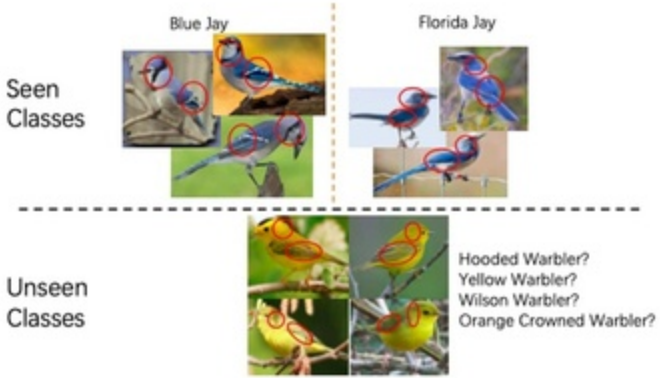
$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$

Theorem 1 [Ben-David et al., 2010]

Let $\mathcal{H}$ be a hypothesis space. Denote by ϵ_s and ϵ_u the generalization errors on $\mathcal{D}_s$ and $\mathcal{D}_u$, then for every $h \in \mathcal{H}$:

$$\epsilon_u(h) \leq \epsilon_s(h) + \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_u) + \lambda. \quad (6)$$

Explicit penalization on intra-class distances



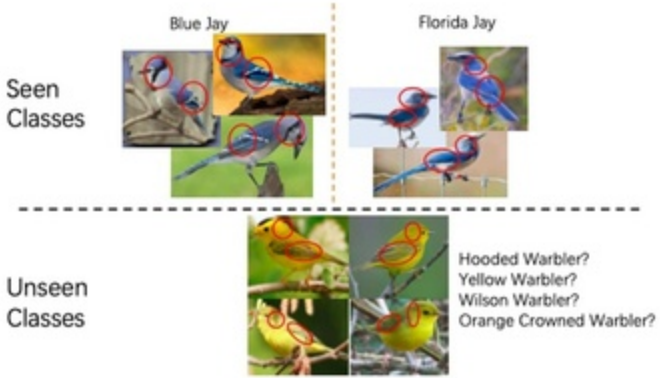
$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$

Theorem 1 [Ben-David et al., 2010]

Let $\mathcal{H}$ be a hypothesis space. Denote by ϵ_s and ϵ_u the generalization errors on $\mathcal{D}_s$ and $\mathcal{D}_u$, then for every $h \in \mathcal{H}$:

$$\epsilon_u(h) \leq \epsilon_s(h) + \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_u) + \lambda. \quad (6)$$

Explicit penalization on intra-class distances



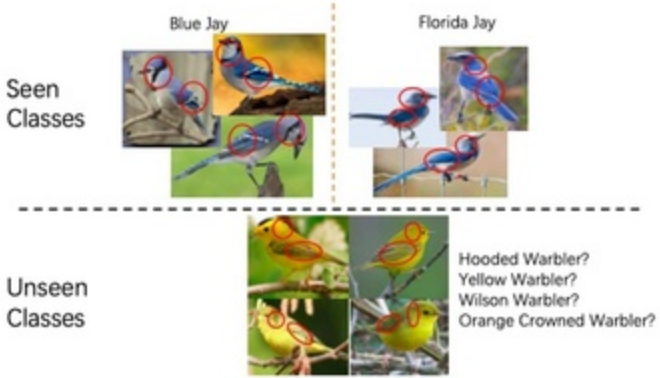
$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$

Theorem 1 [Ben-David et al., 2010]

Let $\mathcal{H}$ be a hypothesis space. Denote by ϵ_s and ϵ_u the generalization errors on $\mathcal{D}_s$ and $\mathcal{D}_u$, then for every $h \in \mathcal{H}$:

$$\epsilon_u(h) \leq \xi_s(h) + \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_u) + \lambda. \quad (6)$$

Explicit penalization on intra-class distances



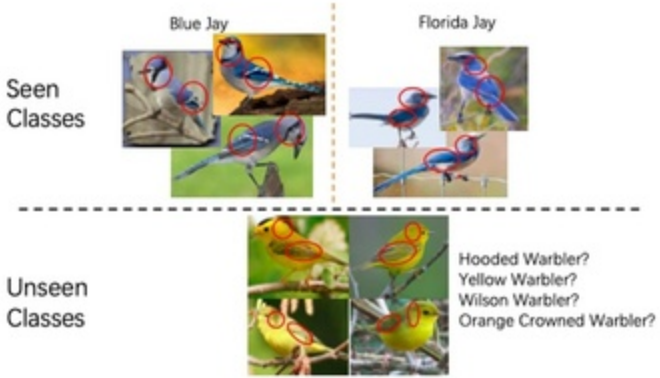
$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$

Theorem 1 [Ben-David et al., 2010]

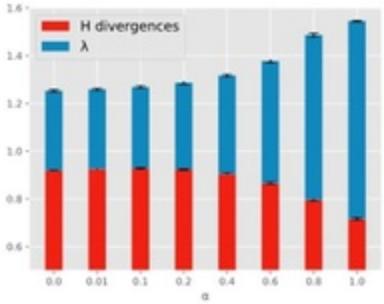
Let $\mathcal{H}$ be a hypothesis space. Denote by ϵ_s and ϵ_u the generalization errors on $\mathcal{D}_s$ and $\mathcal{D}_u$, then for every $h \in \mathcal{H}$:

$$\epsilon_u(h) \leq \epsilon_s(h) + \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_u) + \lambda. \quad (6)$$

Explicit penalization on intra-class distances



$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$

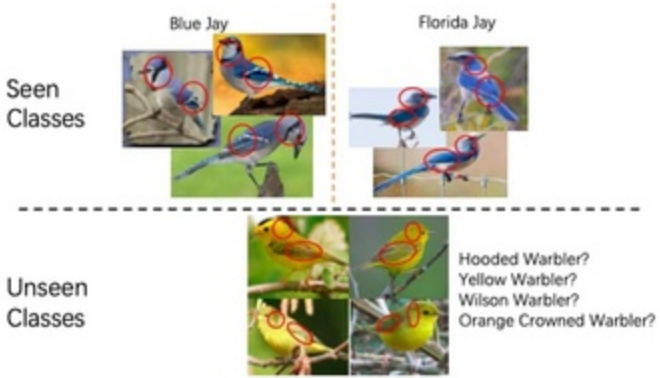


Theorem 1 [Ben-David et al., 2010]

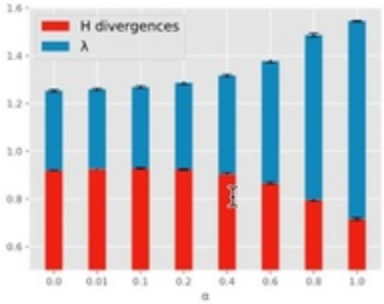
Let $\mathcal{H}$ be a hypothesis space. Denote by ϵ_s and ϵ_u the generalization errors on $\mathcal{D}_s$ and $\mathcal{D}_u$, then for every $h \in \mathcal{H}$:

$$\epsilon_u(h) \leq \epsilon_s(h) + \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_u) + \lambda. \quad (6)$$

Explicit penalization on intra-class distances



$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$

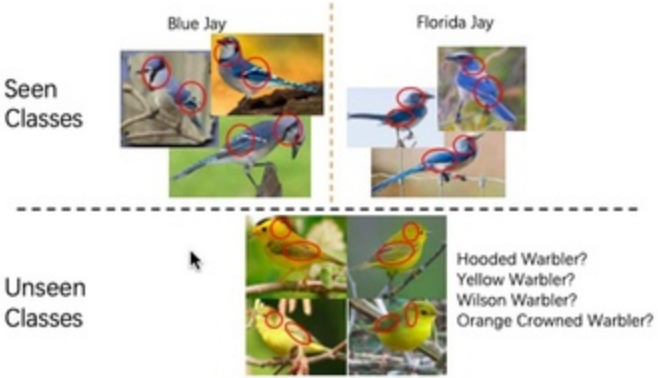


Theorem 1 [Ben-David et al., 2010]

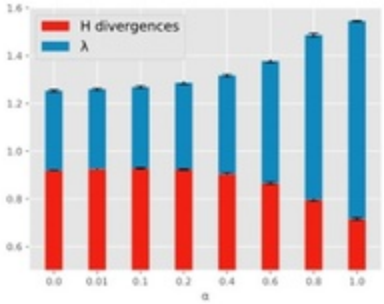
Let $\mathcal{H}$ be a hypothesis space. Denote by ϵ_s and ϵ_u the generalization errors on $\mathcal{D}_s$ and $\mathcal{D}_u$, then for every $h \in \mathcal{H}$:

$$\epsilon_u(h) \leq \epsilon_s(h) + \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_u) + \lambda. \quad (6)$$

Explicit penalization on intra-class distances



$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$

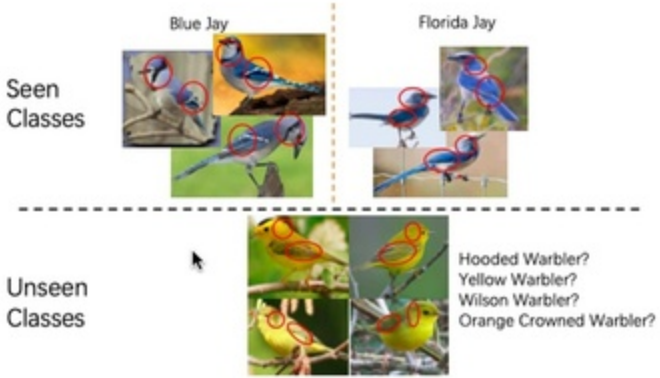


Theorem 1 [Ben-David et al., 2010]

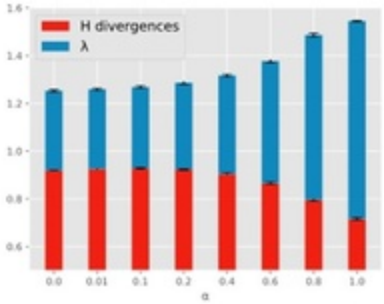
Let $\mathcal{H}$ be a hypothesis space. Denote by ϵ_s and ϵ_u the generalization errors on $\mathcal{D}_s$ and $\mathcal{D}_u$, then for every $h \in \mathcal{H}$:

$$\epsilon_u(h) \leq \epsilon_s(h) + \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_u) + \lambda. \quad (6)$$

Explicit penalization on intra-class distances



$$\mathcal{L}_{AMSoft} - \alpha \sum_{l} \frac{1}{N^{l}_{pairs}} \sum_{\mathbf{x}^{l}_{i}, \mathbf{x}^{l}_{j} \in \mathcal{T}_{l}} e^{-\frac{1}{2}(\mathbf{x}^{l}_{i} - \mathbf{x}^{l}_{j})^{2}} \quad (5)$$



Theorem 1 [Ben-David et al., 2010]

Let $\mathcal{H}$ be a hypothesis space. Denote by ϵ_s and ϵ_u the generalization errors on $\mathcal{D}_s$ and $\mathcal{D}_u$, then for every $h \in \mathcal{H}$:

$$\epsilon_u(h) \leq \epsilon_s(h) + \hat{d}_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_u) + \lambda. \quad (6)$$

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

JRS versus MMD

$$\widehat{\mathrm{MMD}}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}^{\mathrm{I}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

JRS versus MMD

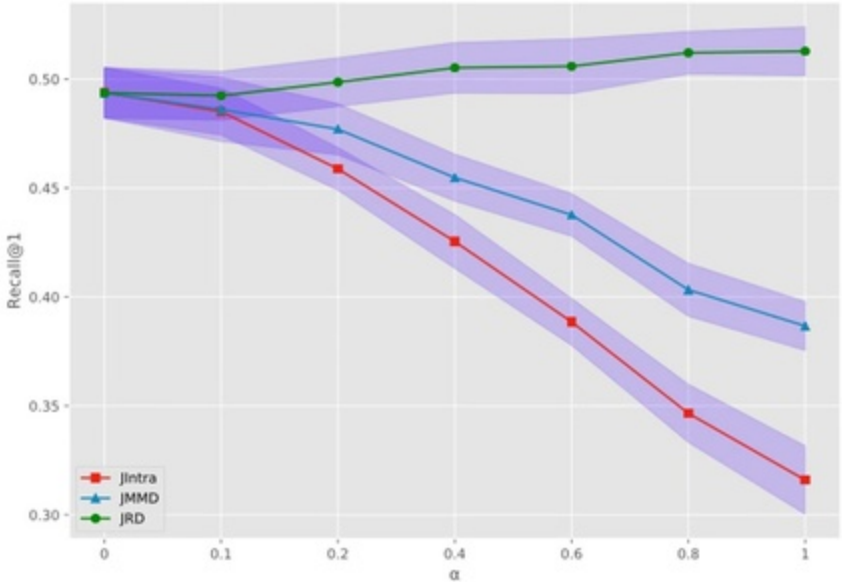
$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

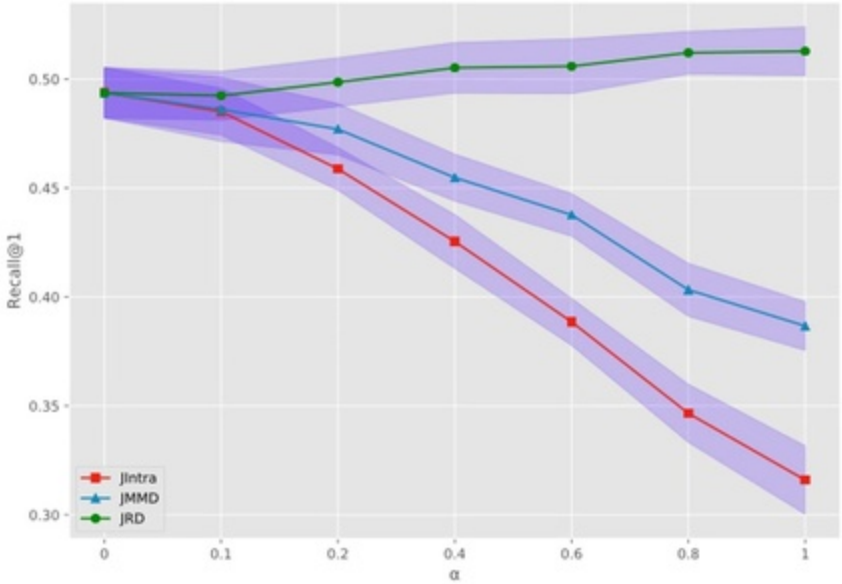


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}}{}^{NN}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506([illegible])	[illegible]10(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

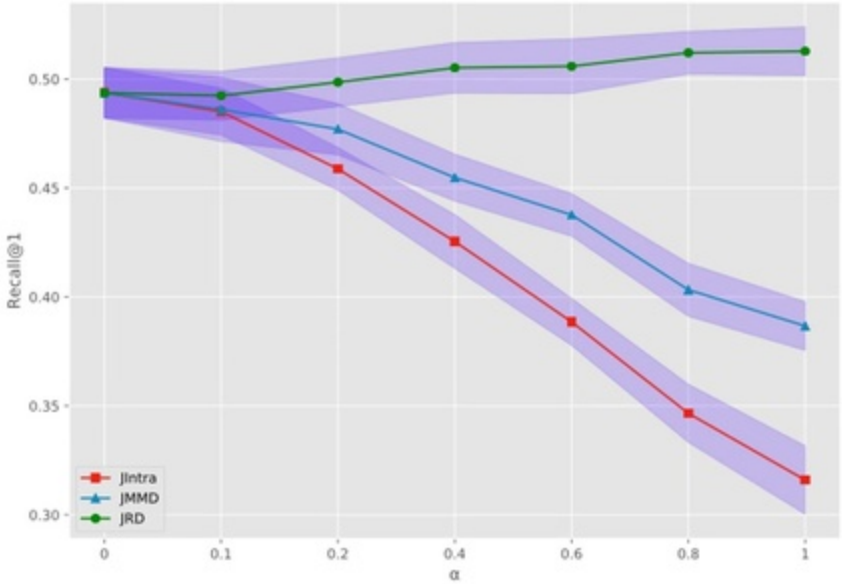


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

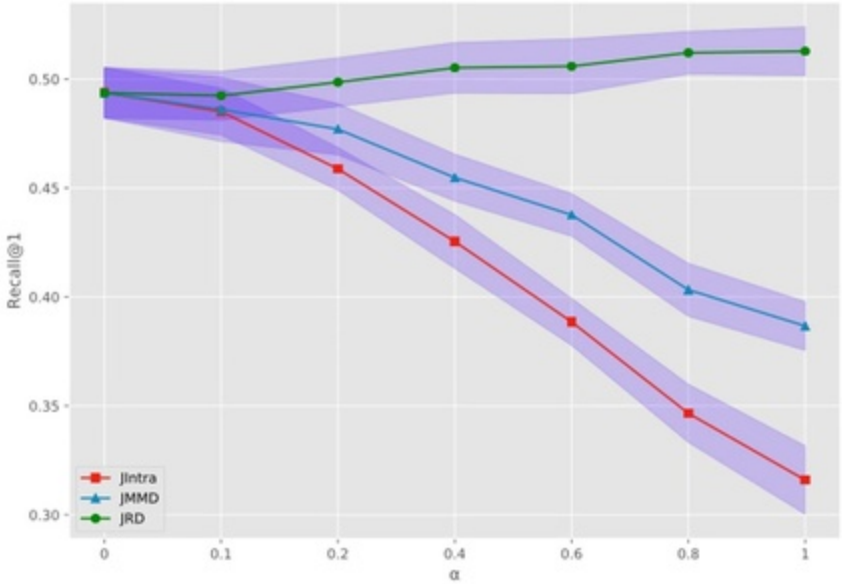


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

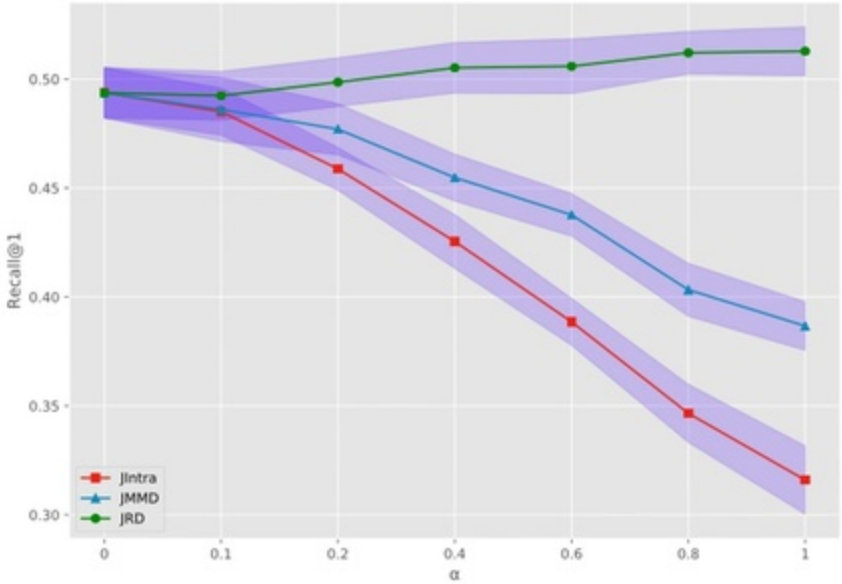


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

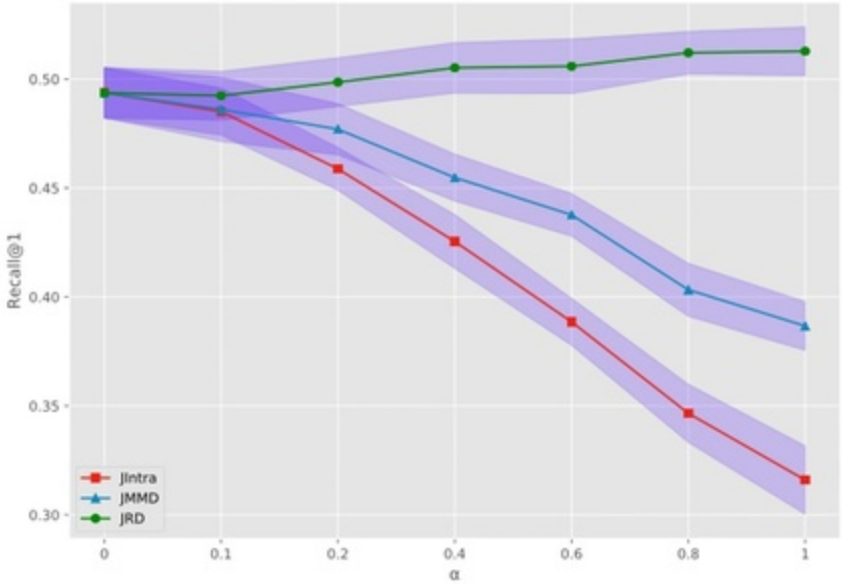


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

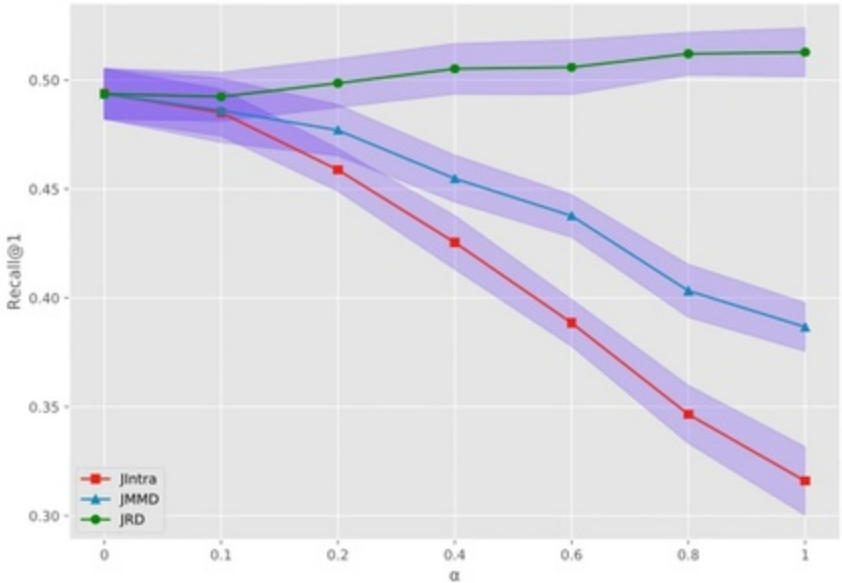


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

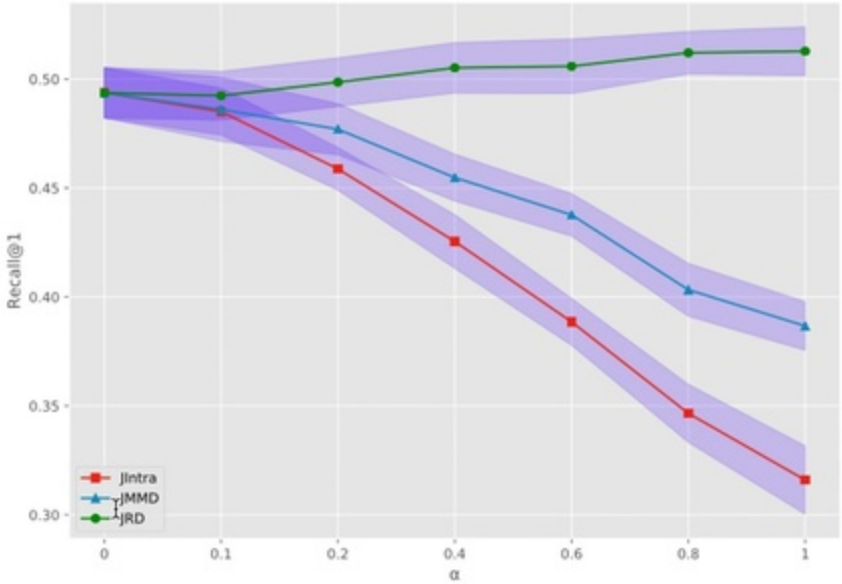


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

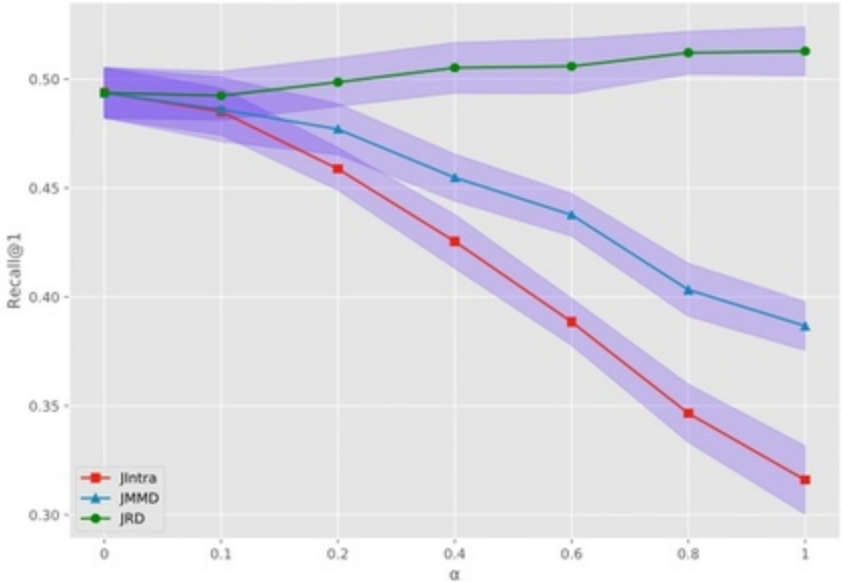


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

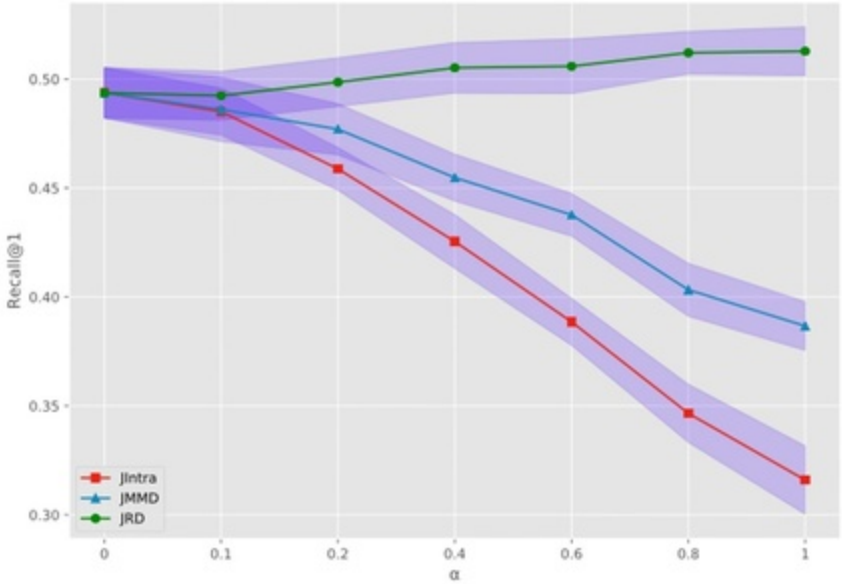


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$

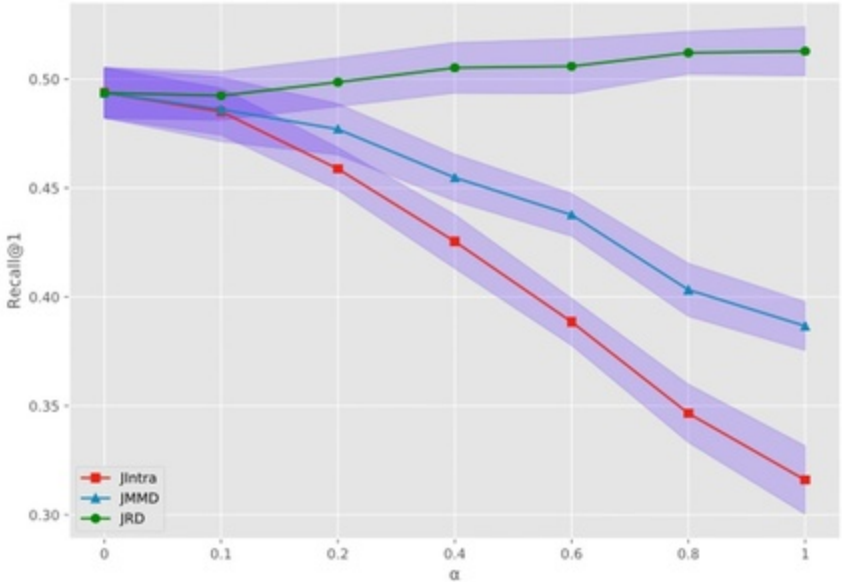


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \quad (7)$$

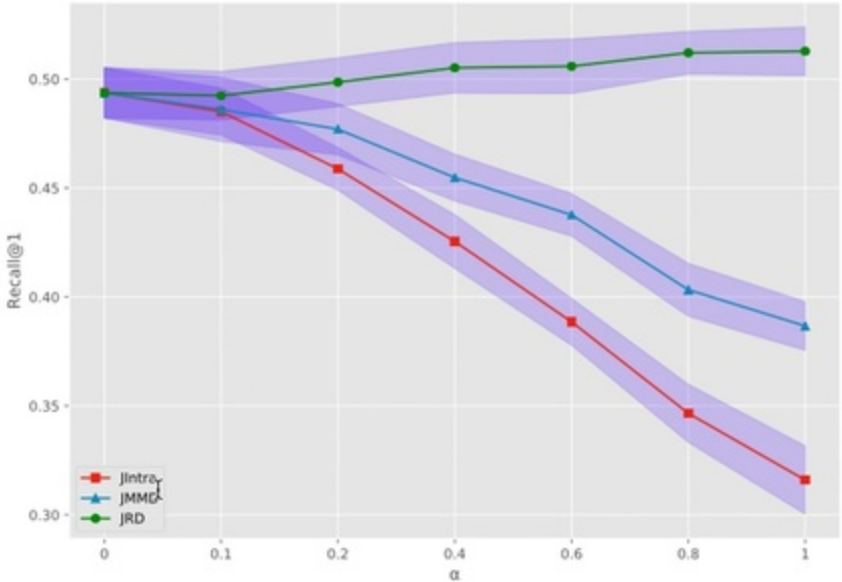


$$\mathcal{L}_{AMSoft} + \alpha Regularizer \quad (8)$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}^{NN}}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

JRS versus MMD

$$\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} = \|\mu_{\mathbb{P}}\|^2_{\mathcal{RKHS}} + \|\mu_{\mathbb{Q}}\|^2_{\mathcal{RKHS}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{RKHS}} \tag{7}$$



$$\mathcal{L}_{AMSoft} + \alpha Regularizer \tag{8}$$

Regularizers	Recall@1	λ^{NN}	$\hat{d}_{\mathcal{H}}{}^{NN}$
JMMD(α@0.01)	0.486(0.015)	0.321(0.006)	0.9275(0.003)
JRD(α@1)	0.506(0.013)	0.310(0.006)	0.934(0.004)

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x}\cdot\mathbf{x}'+1)^p$
Kernel inducing MGF	$exp(\mathbf{x}\cdot\mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x}\cdot\mathbf{x}'+1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x}\cdot\mathbf{x}'+1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x}\cdot\mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Source Code: https://github.com/YangLin122/JRD

Contact Email: chu_xu@pku.edu.cn

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x}\cdot\mathbf{x}'+1)^p$
Kernel inducing MGF	$exp(\mathbf{x}\cdot\mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x}\cdot\mathbf{x}'+1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x}\cdot\mathbf{x}'+1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x}\cdot\mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Source Code: https://github.com/YangLin122/JRD
Contact Email: chu_xu@pku.edu.cn

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Source Code: https://github.com/YangLin122/JRD

Contact Email: chu_xu@pku.edu.cn

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x}\cdot\mathbf{x}'+1)^p$
Kernel inducing MGF	$exp(\mathbf{x}\cdot\mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x}\cdot\mathbf{x}'+1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x}\cdot\mathbf{x}'+1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x}\cdot\mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Source Code: https://github.com/YangLin122/JRD

Contact Email: chu_xu@pku.edu.cn

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x}\cdot\mathbf{x}'+1)^p$
Kernel inducing MGF	$exp(\mathbf{x}\cdot\mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x}\cdot\mathbf{x}'+1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x}\cdot\mathbf{x}'+1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x}\cdot\mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Source Code: https://github.com/YangLin122/JRD

Contact Email: chu_xu@pku.edu.cn

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x}\cdot\mathbf{x}'+1)^p$
Kernel inducing MGF	$exp(\mathbf{x}\cdot\mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x}\cdot\mathbf{x}'+1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x}\cdot\mathbf{x}'+1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x}\cdot\mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Source Code: https://github.com/YangLin122/JRD

Contact Email: chu_xu@pku.edu.cn

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x}\cdot\mathbf{x}'+1)^p$
Kernel inducing MGF	$exp(\mathbf{x}\cdot\mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x}\cdot\mathbf{x}'+1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x}\cdot\mathbf{x}'+1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x}\cdot\mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Source Code: https://github.com/YangLin122/JRD

Contact Email: chu_xu@pku.edu.cn

预览 文件 编辑 显示 前往 工具 窗口 帮助
Recording · SlidesLive Rec
recorder.slideslive.com/recording/#!token=ACL_posters_4066-6438_distance-metric-learning-with
You video is not being recorded!
Distance_Metric_Learning_with_Joint_Representation_Diversification.pdf（页码：36/39）

Kernel Choice

Kernel	$k(\mathbf{x}, \mathbf{x}')$
Gaussian	$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$
Laplace	$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$
degree-p Inhomogeneous polynomial kernel	$(\mathbf{x} \cdot \mathbf{x}' + 1)^p$
Kernel inducing MGF	$exp(\mathbf{x} \cdot \mathbf{x}')$

$k(\mathbf{x}, \mathbf{x}')$	Recall@1(%)	Recall@2(%)	Recall@4(%)	Recall@8(%)
$exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{\sigma^2})$ (α@1)	67.9	78.5	86.1	91.2
$exp(-\frac{\lVert\mathbf{x}-\mathbf{x}'\rVert_1}{\sigma})$ (α@1)	68.1	78.2	86.4	91.8
$(\mathbf{x} \cdot \mathbf{x}' + 1)^2$ (α@1e-3)	66.1	77	85.3	90.9
$(\mathbf{x} \cdot \mathbf{x}' + 1)^5$ (α@1e-3)	65.2	76.2	86.4	90.7
$exp(\mathbf{x} \cdot \mathbf{x}')$ (α@1e-3)	66.1	76.7	85.4	91.1

Chrome 文件 编辑 视图 历史记录 书签 用户 标签页 窗口 帮助

Recording · SlidesLive Rec

recorder.slideslive.com/recording/#!token=ACL_posters_4066-6438_distance-metric-learning-with

You video is not being recorded!

00:14:24

STOP RECORDING

Video recording

No video will be recorded!

Microphone

默认 - Internal Microphone (Built-in)

Please select a microphone you want to use.

Camera

Screen capture

Change screen

STOP RECORDING

If you have any questions please contact your production manager.
If you face any technical issues contact support@slideslive.com.