Unsupervised Domain Adaptation in Regression: Why We Need Methods for Regression?

2024-04-12

Korea University

Data Mining & Quality Analytics Lab.

심 세 진

발표자 소개

❖ 심세진 (Sejin Sim)

- 고려대학교 산업경영공학과 대학원 재학
- Data Mining & Quality Analytics Lab. (김성범 교수님)
- 석박통합 과정 (2022. 03 ~ Present, 5학기)

❖ 관심 연구 분야

- Deep Learning for Multichannel Signal Analysis
- Semi-supervised Regression

❖ 연락처

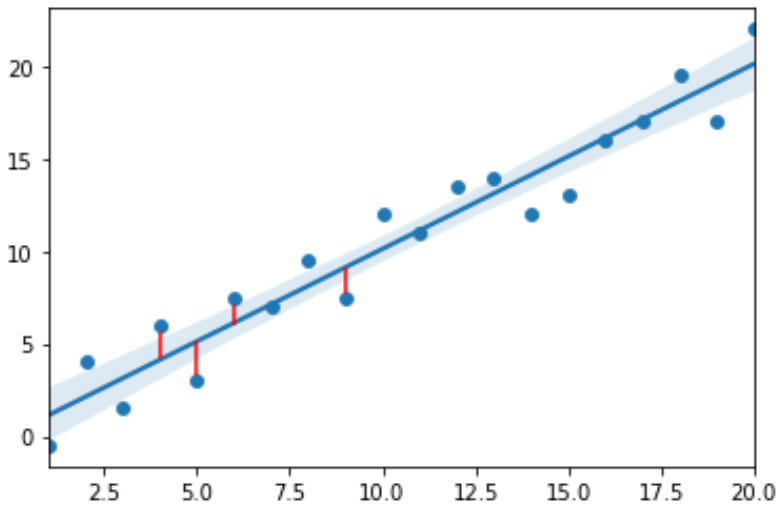
- ssj259@korea.ac.kr

Introduction

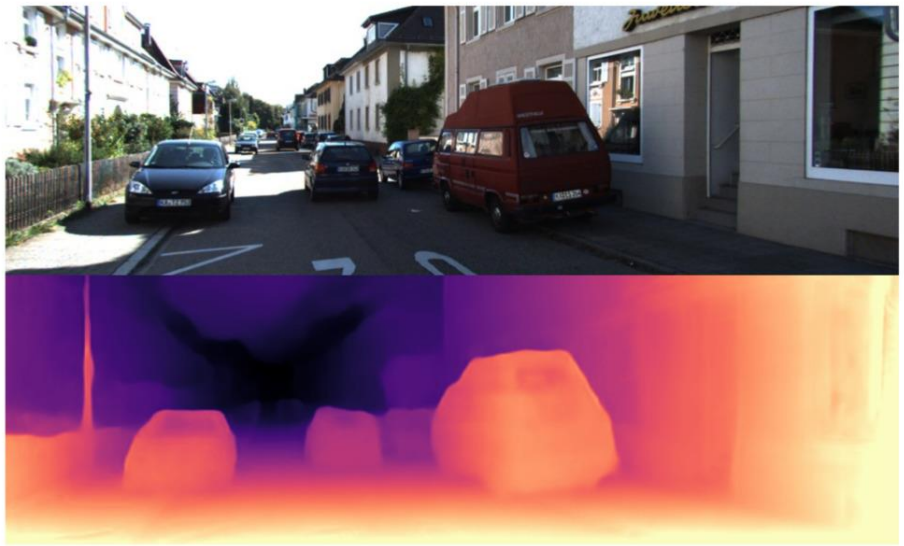
Introduction

❖ Unsupervised Domain Adaptation Regression의 배경

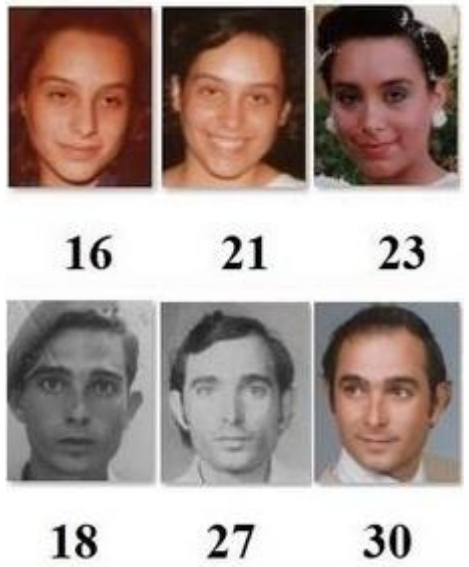
- 회귀는 연속적인 값을 예측하는 작업으로 깊이 추정(depth estimation), 사람 자세 추정(human pose estimation), 연령 추정(age estimation) 등 실제 애플리케이션에서 광범위하게 활용 됨



[회귀(Regression)]



[깊이 추정(depth estimation)]

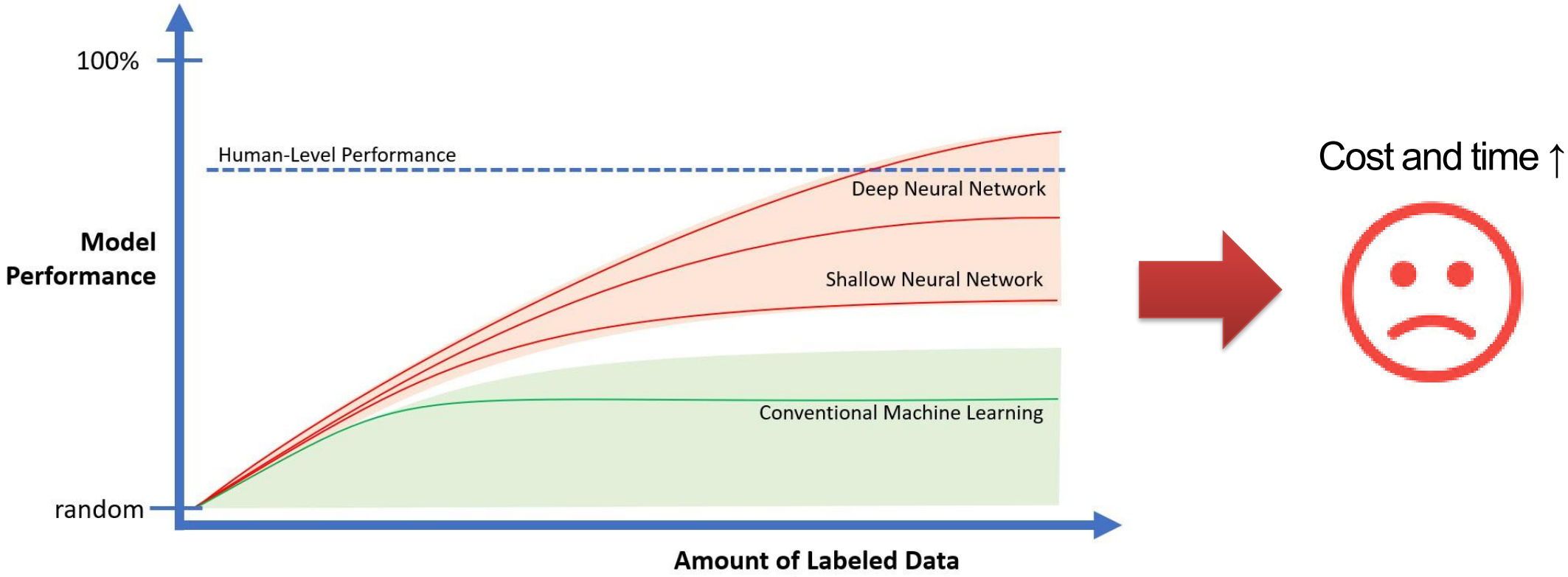


[연령 추정(age estimation)]

Introduction

❖ Unsupervised Domain Adaptation Regression의 배경

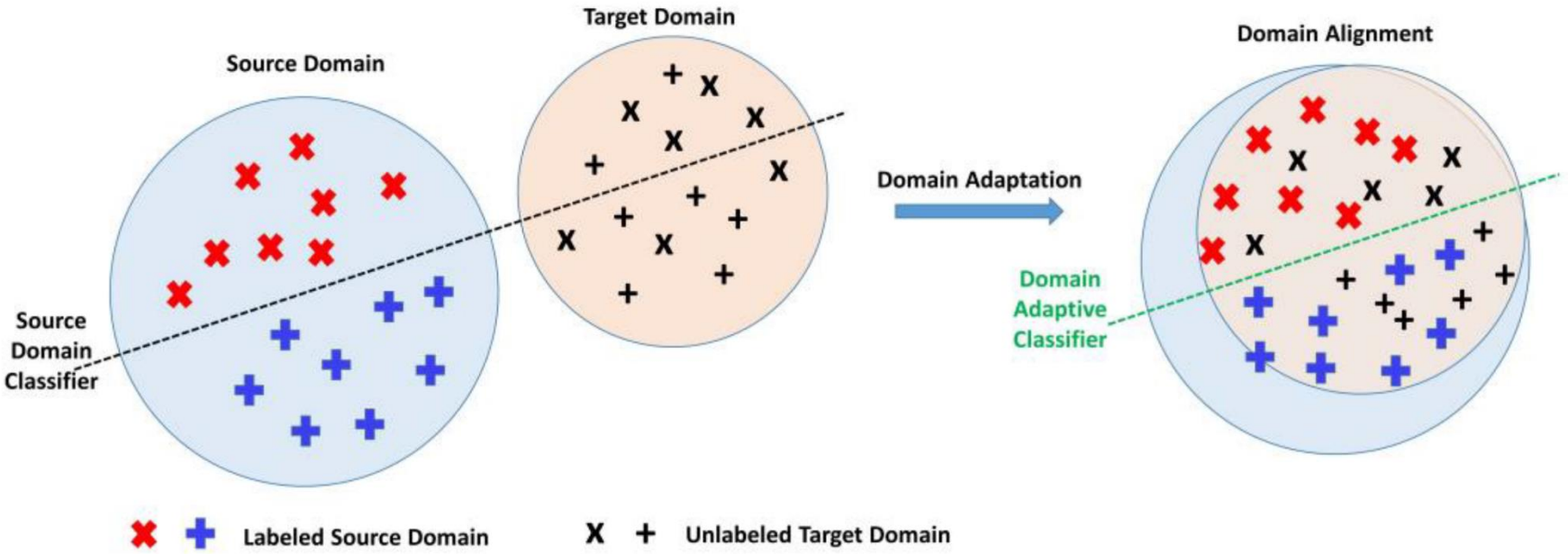
- 좋은 성능을 내는 딥러닝 모델을 학습하려면 다량의 레이블 데이터가 필요 → 시간과 비용이 많이 소요 됨



Introduction

❖ Unsupervised Domain Adaptation Regression 의 배경

- Domain adaptation 방법론을 통해 기존에 라벨링 된 데이터(=source domain)을 활용하여 레이블 되지 않은 새로운 도메인 데이터(=target domain)를 예측하고자 함 → Unsupervised domain adaptation
- 학습 목표 : Source와 Target 도메인 간 차이를 줄이는 것



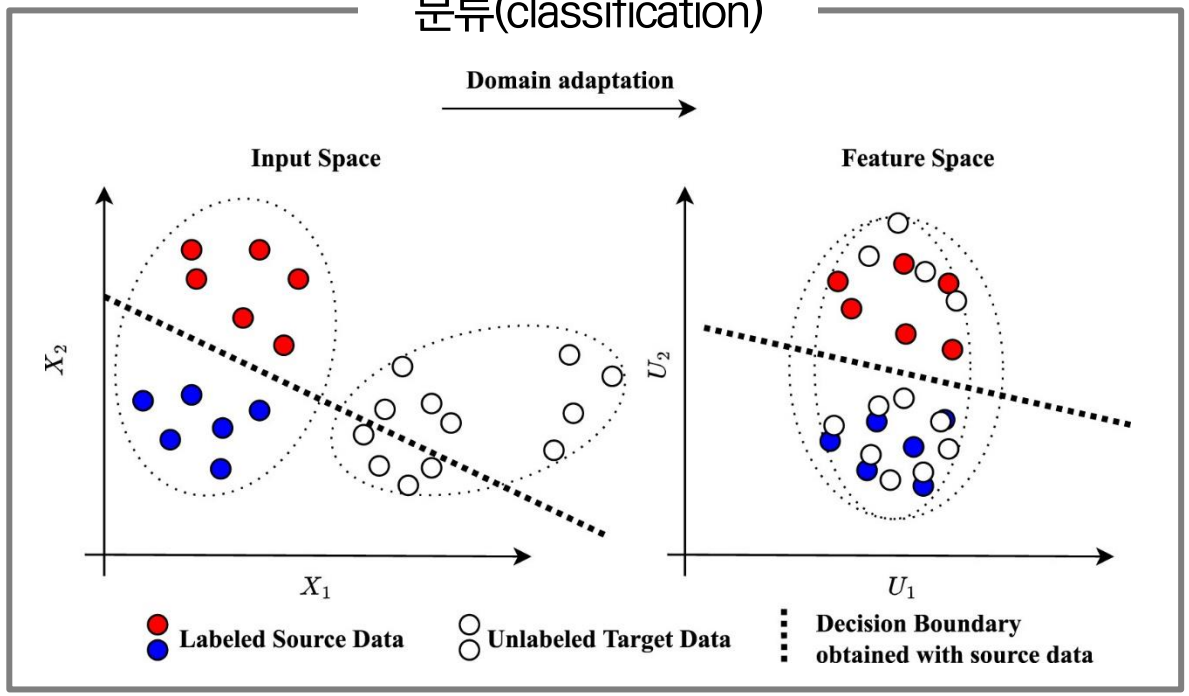
Goel P, Ganatra A. Unsupervised Domain Adaptation for Image Classification and Object Detection Using Guided Transfer Learning Approach and JS Divergence. *Sensors*. 2023; 23(9):4436

Introduction

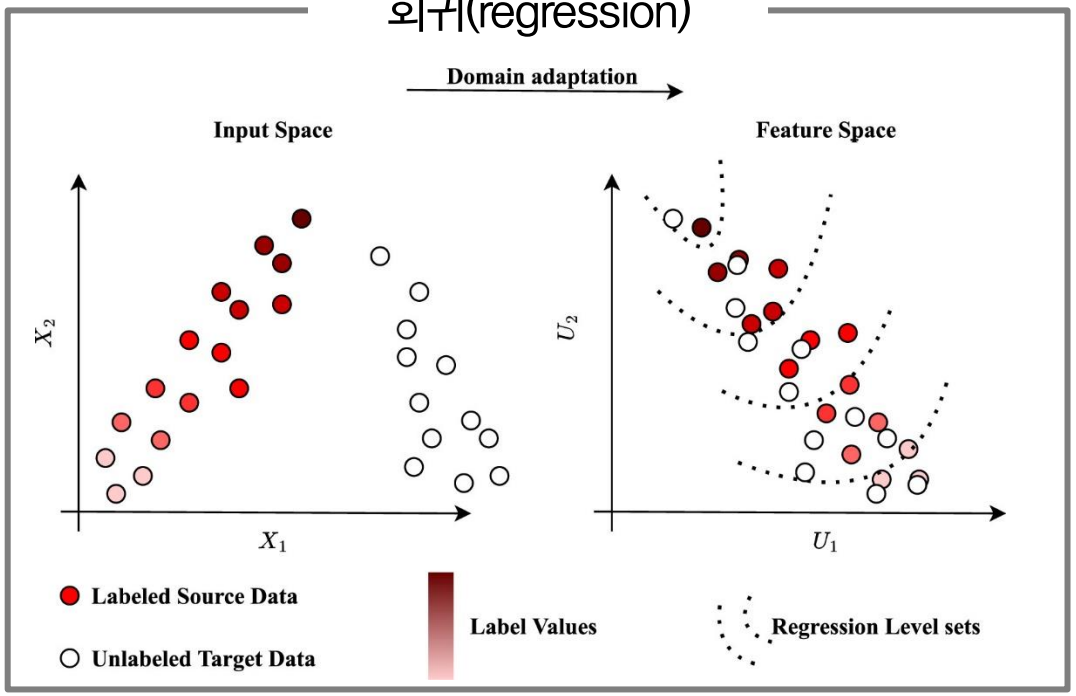
❖ Unsupervised Domain Adaptation Regression 의 배경

- Unsupervised Domain Adaptation 방법론들은 기존에 주로 분류(classification)에 초점을 맞추어 왔으며, 분류 방법론을 회귀(Regression)에 직접 적용하는 데에는 종종 어려움이 발생

분류(classification)



회귀(regression)



Dhaini, M., Berar, M., Honeine, P., & Van Exem, A. (2023). Unsupervised domain adaptation for regression using dictionary learning. Knowledge-Based Systems, 267, 110439.

Introduction

❖ Unsupervised Domain Adaptation Regression의 배경

- Unsupervised domain adaptation regression 관련 연구들을 아래와 같이 소개

1. Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)
   - ➢ 회귀가 분류에 비해 feature scaling에 민감한 것을 실험으로 증명
   - ➢ 도메인 사이의 새로운 기하학 거리를 정의함에 따라 회귀 데이터셋에서 좋은 성능을 보임

2. DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)
   - ➢ 회귀 문제의 경우 Inverse gram matrix(역행렬)를 통해 모델을 학습해야 한다는 근거를 제시
   - ➢ Pseudo-inverse gram matrix(유사 역행렬)을 모델 학습에 사용하여 도메인간 거리를 줄여 성능을 높임

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

Methods

❖ Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

- 회귀 작업이 분류 작업에 비해 feature scaling에 민감한 것을 실험으로 증명
- 도메인 사이의 새로운 기하학 거리를 정의함에 따라 회귀 데이터셋에서 좋은 성능을 보임

2024.04.12 기준 69회 인용

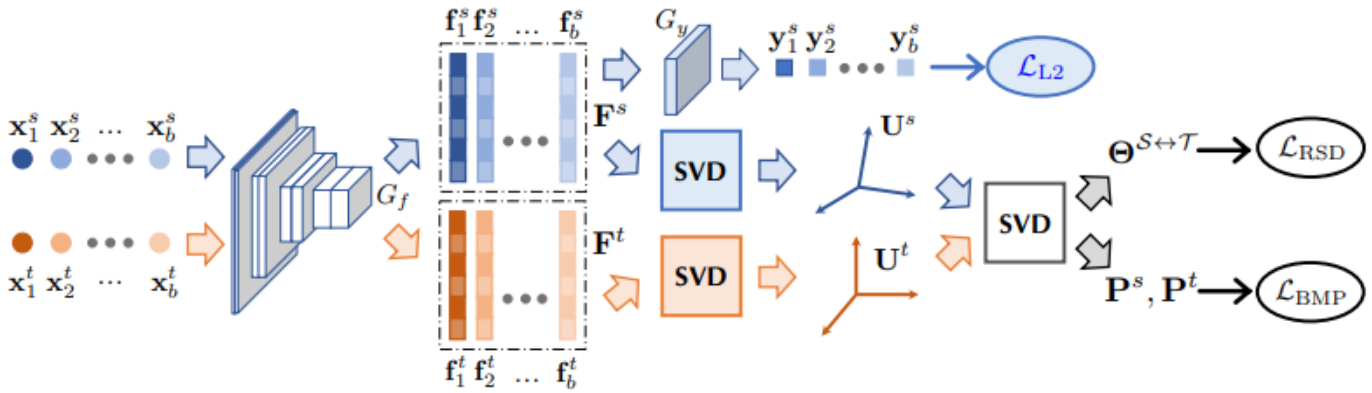
Representation Subspace Distance for Domain Adaptation Regression

Xinyang Chen [1] Sinan Wang [1] Jianmin Wang [1] Mingsheng Long [1]

Abstract

Regression, as a counterpart to classification, is a major paradigm with a wide range of applications. Domain adaptation regression extends it by generalizing a regressor from a labeled source domain to an unlabeled target domain. Existing domain adaptation regression methods have achieved positive results limited only to the shallow regime. A question arises: Why learning invariant representations in the deep regime less pronounced? A key finding of this paper is that classification is robust to feature scaling but regression is not, and aligning the distributions of deep representations will alter feature scale and impede domain adaptation regression. Based on this finding, we propose to close the domain gap through *orthogonal bases* of the representation spaces, which are free from feature scaling. Inspired by Riemannian geometry of Grassmann manifold, we define a geometrical distance over representation subspaces and learn deep transferable representations by minimizing it. To avoid breaking the geometrical properties of deep representations, we further introduce the such as object localization, image registration, human pose estimation, to name a few (Lathuilière et al., 2019).

Deep learning has made remarkable changes in diverse regression applications across many fields. Nonetheless, training high-quality deep models relies on large-scale labeled datasets. And in many real-world regression applications, precisely annotating abundant training instances is time-consuming and laborious. A solution to this problem is leveraging off-the-shelf labeled data from a relevant domain and applying domain adaptation approaches to overcome the domain shift or dataset bias (Shimodaira et al., 2009). There are many pioneers hammering at tackling domain adaptation regression (**DAR**) problem at the theoretical or algorithmic level. Mansour et al. (2009) and Cortes & Mohri (2011) conducted extensive theoretical analyses of the DAR problem. Meanwhile, a series of algorithms are proposed for the DAR problem. These methods focus on importance weighting (Geng et al., 2007; Guo et al., 2008; Yamada et al., 2014) or learning invariant representations (Cao et al., 2010; Pan et al., 2011) in the shallow regime, achieving impressive improvements and bringing inspiration for future works. However, most methods rely on labeled target data. And there are rare DAR approaches in the deep regime.



Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 회귀가 분류에 비해 feature scaling에 민감한 것을 실험으로 증명

- Regression의 대표 손실 함수 = Mean Squared Error(MSE) loss

  ➢ MSE = $\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

  ex) y_i = 10, $\hat{y}_i$ = 8 (연속적인 실수)

- Classification의 대표 손실 함수 = Cross Entropy loss(CE) loss

  ➢ CE = $-\sum_{i=1}^{N} y_i \log(\hat{y}_i)$

  ex) y_i = [1, 0, 0, 0], $\hat{y}_i$ = [0.6, 0.4, 0, 0] (범주형)

  ➢ 분류 예측 확률 $\hat{y}_i$ 을 도출하기 위해 사용 되는 **Softmax fuction은 서로 다른 범주 값이 경쟁하도록 함**

  → 경쟁 메커니즘을 훈련에 사용함으로써 **분류기(Classifier)가 feature scale의 변화에 빠르게 적응**할 수 있음

  → 회귀에는 이러한 경쟁 메커니즘이 없기 때문에 변화에 잘 적응이 잘 안될 수 있음 → 실험을 진행

Methods

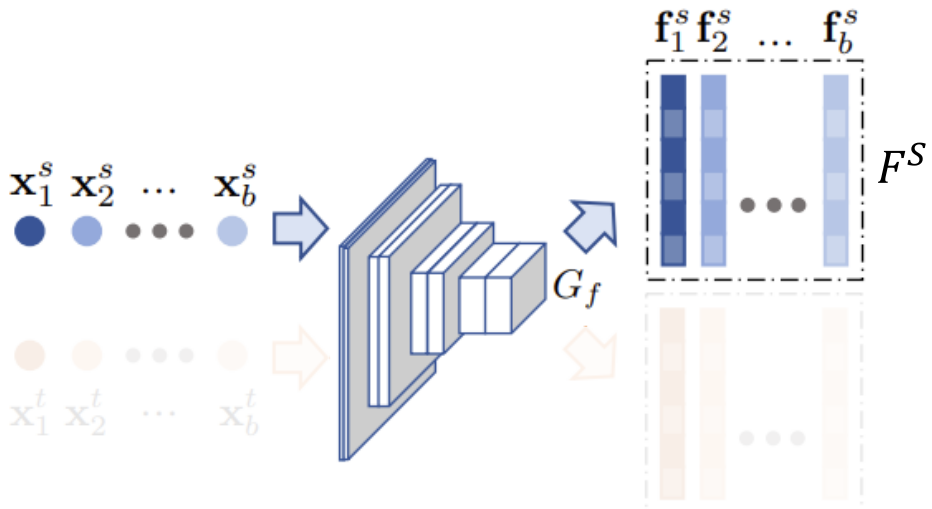
Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 회귀가 분류에 비해 feature scaling에 민감한 것을 실험으로 증명

- 분류, 회귀 각각 모델 훈련 시 규제 항 reg_{norm} 을 추가

  $= Loss_{supervised} + reg_{norm}(\|F^S\|_F^2 - R)^2$

  - $Loss_{supervised}$ = 분류는 CE, 회귀는 MSE
  - R = Expected feature 크기 = feature scaling 실험을 위해 최소~최대로 조절하는 값
  - $\|F^S\|_F^2$ = 특징 추출기를 거친 feature F^S에 프로베니우스 놈 $\|\cdot\|_F$을 연산해 행렬의 크기를 구함



프로베니우스 놈 $\|\cdot\|_F$ 은 L2(유클리디안) norm을 행렬에 확장

ex 1) 벡터 $v = [4, -3]$ 의 L2 norm

$$\|v\|_2 = \sqrt{4^2 + (-3)^2} = \sqrt{16 + 9} = 5$$

ex 2) 행렬 $A = \begin{bmatrix} 2 & -1 & 3 \\ -1 & 5 & 4 \end{bmatrix}$ 의 프로베니우스 놈 $\|\cdot\|_F$

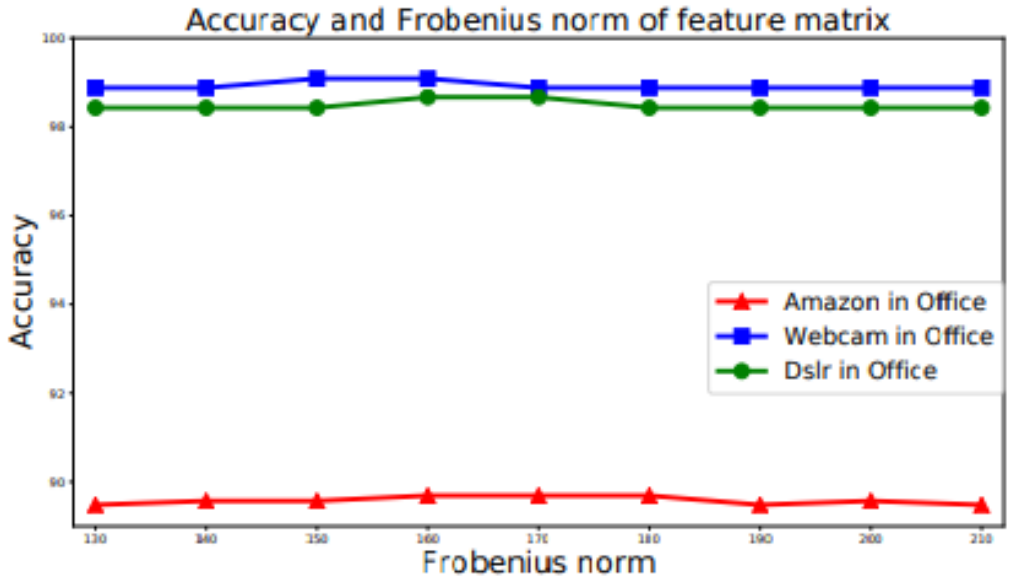
$$\|A\|_F = \sqrt{2^2 + (-1)^2 + \cdots + 4^2} = \sqrt{56} \approx 7.48$$

Methods

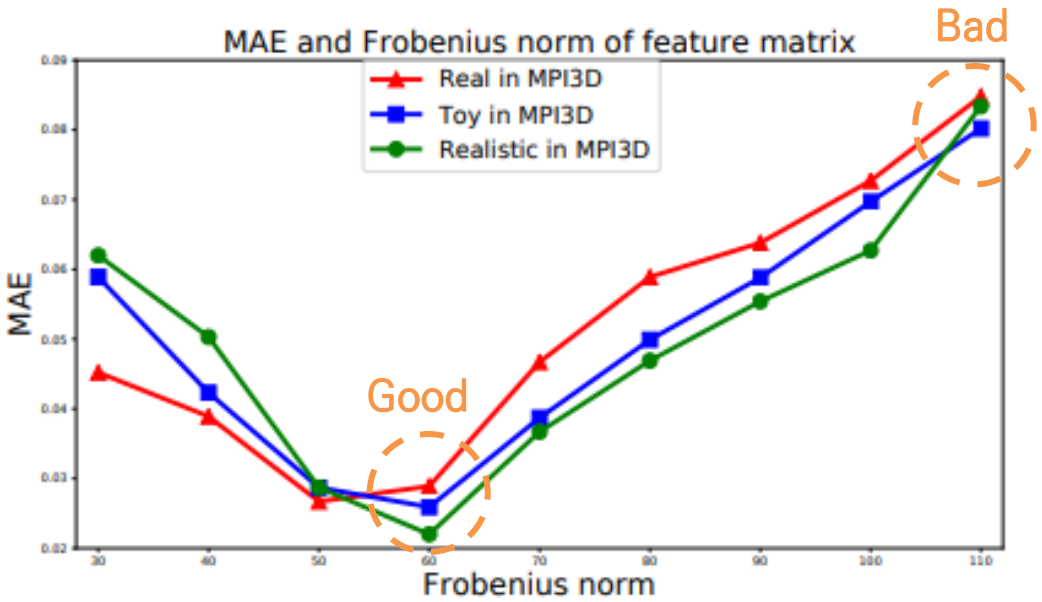
Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 회귀가 분류에 비해 feature scaling에 민감한 것을 실험으로 증명

- 프로베니우스 놈의 크기 즉, feature scale을 변경해 가면서 실험한 결과
- 분류의 경우 feature scale에 변동이 없지만, 회귀의 경우 성능에 큰 영향을 미치는 것을 볼 수 있음



(a) Classification



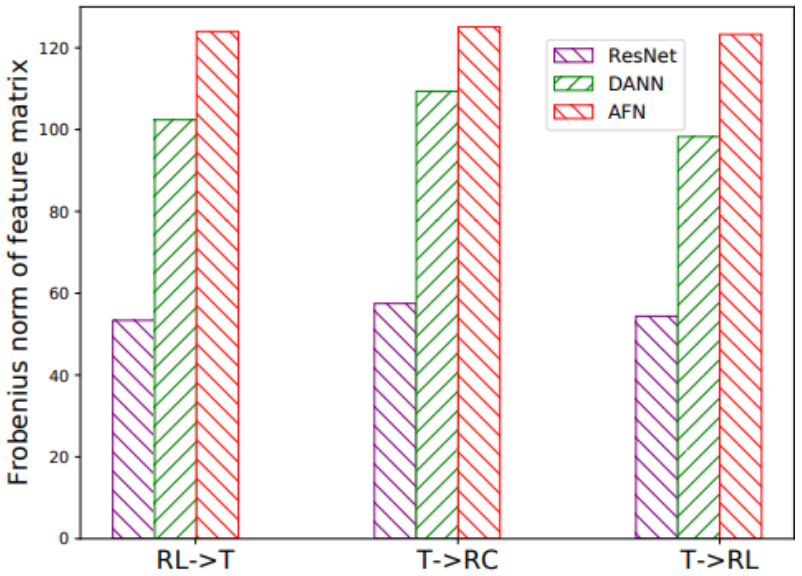
(b) Regression

Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 회귀가 분류에 비해 feature scaling에 민감한 것을 실험으로 증명

- Unsupervised Domain Adaptation Classification 방법론인 DANN, AFN 방법론을 회귀 데이터에 실험한 결과
- 지도 학습(ResNet)에서의 feature scale보다 DANN, AFN을 적용하는 경우 feature scale이 훨씬 커짐을 볼 수 있음



(c) Frobenius norm

분류 방법론을 적용하는 경우 feature scale이 크게 변동 됨

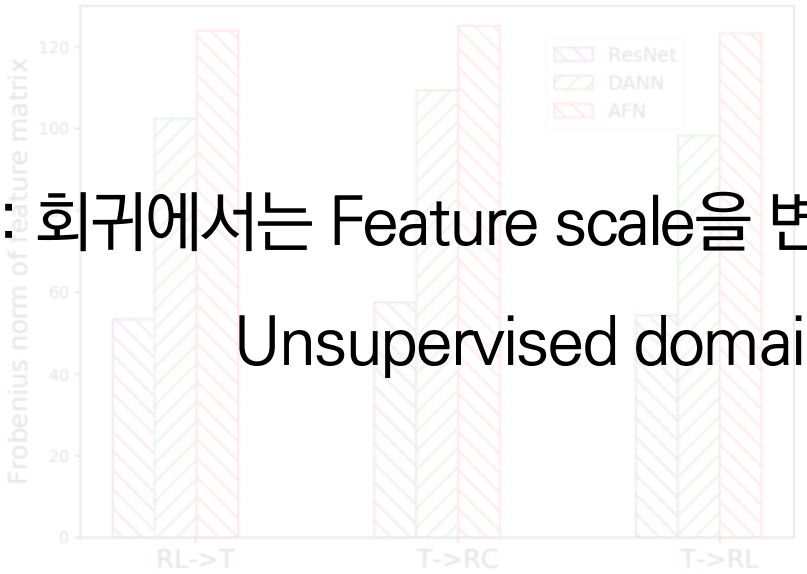
→ 성능이 변동될(=낮아질) 위험 ↑

Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 회귀가 분류에 비해 feature scaling에 민감한 것을 실험으로 증명

- Unsupervised Domain Adaptation Classification 방법론인 DANN, AFN 방법론을 회귀 데이터에 실험한 결과
- 지도 학습(ResNet)에서의 feature scale보다 DANN, AFN을 적용하는 경우 feature scale이 훨씬 커짐을 볼 수 있음



(c) Frobenius norm

주장 : 회귀에서는 Feature scale을 변동 시키지 않으면서 도메인간의 차이를 줄이는

Unsupervised domain adaptation 학습을 해야 한다!

분류 방법론을 적용하는 경우 feature scale이 크게 변동 됨

Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

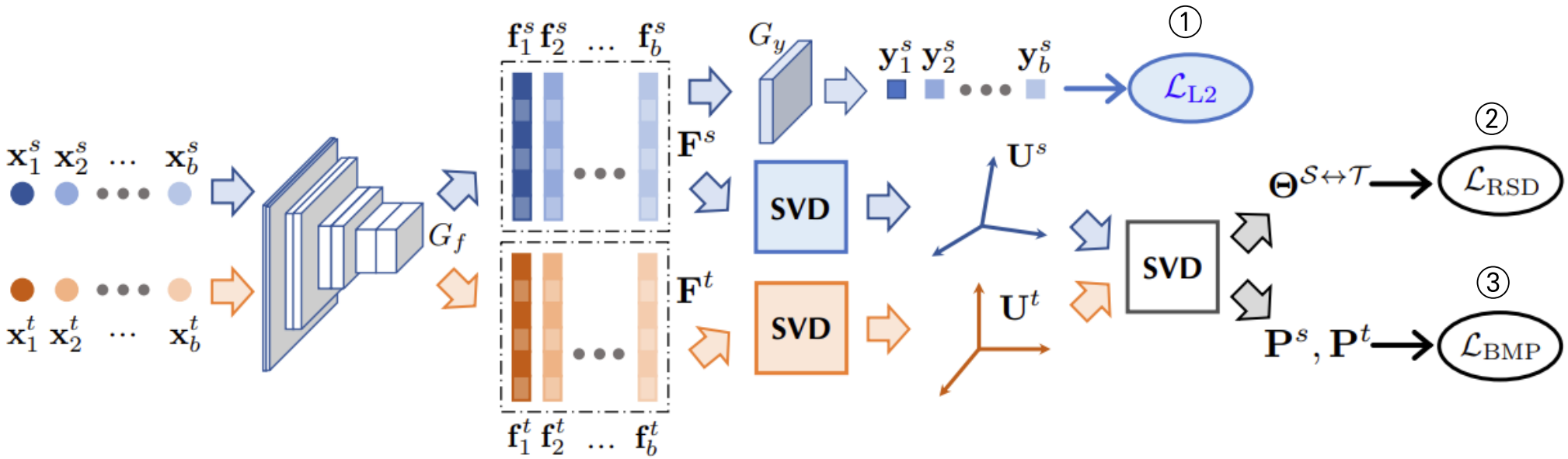
❖ 모델 훈련 단계

β and γ = hyperparameters

- Total loss = $L_{L2}(G_f, G_y) + \beta L_{RSD}(G_f) + \gamma L_{BMP}(G_f)$

① ② ③

② 도메인간 차이 ↓ ③ 특이 값 중요 순서 고려



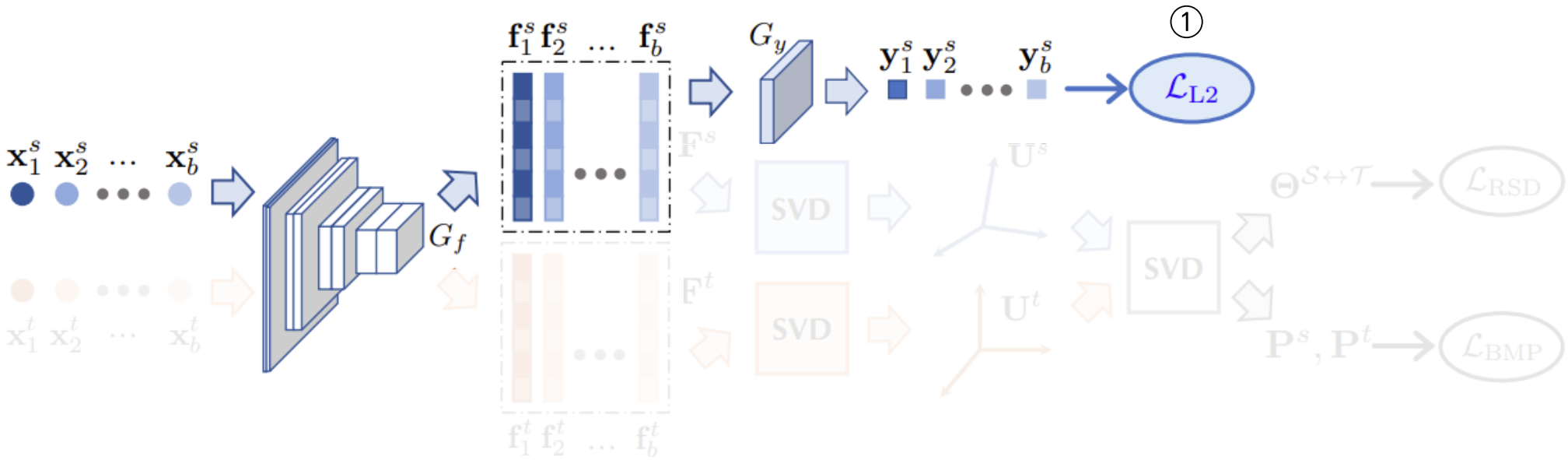
Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 단계 : ① $L_{L2}(G_f, G_y)$

- Total loss = $L_{L2}(G_f, G_y) + \beta L_{RSD}(G_f) + \gamma L_{BMP}(G_f)$

① $L_{L2}(G_f, G_y)$ = 일반적인 supervised loss = $MSE(G_y(G_f(x_b^s),)y_b^s) = MSE(\hat{y}_b^s,\ y_b^s)$



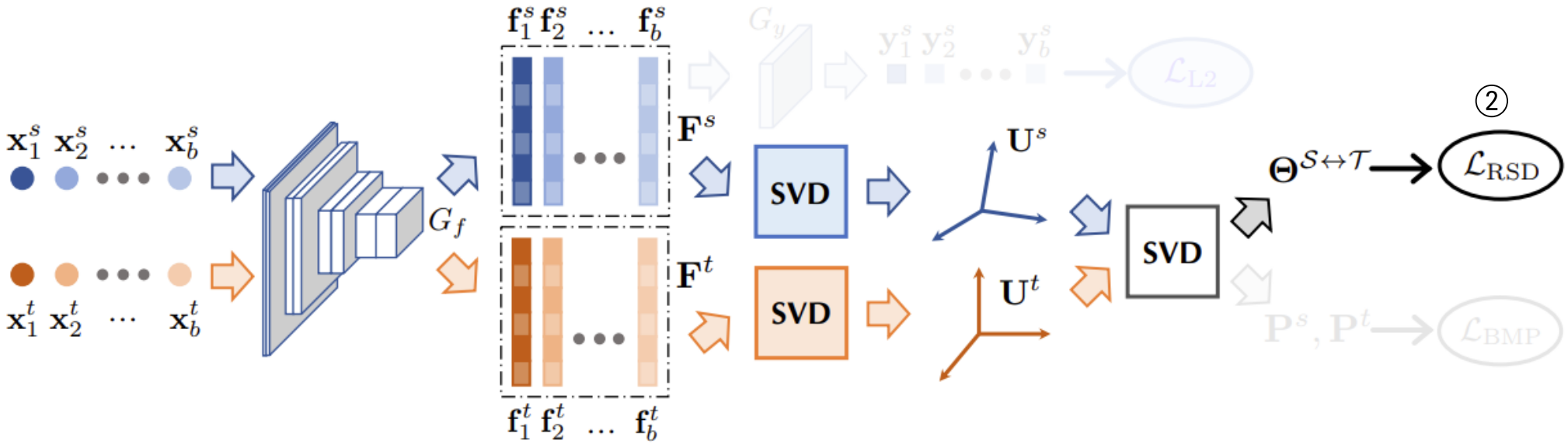
Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 과정 : ② $L_{RSD}(G_f)$

- Total loss = $L_{L2}(G_f, G_y) + \beta L_{RSD}(G_f) + \gamma L_{BMP}(G_f)$

② $L_{RSD}(G_f)$ = Representation Subspace Distance(하위 공간 거리 표현) loss

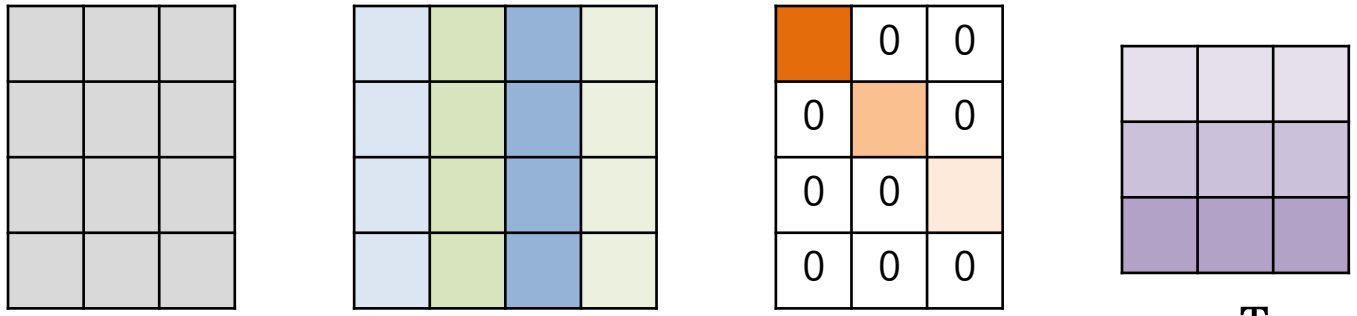


Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 과정 : ② $L_{RSD}(G_f)$ – SVD

- 특이값 분해(SVD)란? = 임의의 $m \times n$ 차원의 행렬 A는 $A = \mathrm{U}\Sigma\mathrm{V}^{\mathrm{T}}$ 와 같이 분해하는 방식 → 차원 축소 & 특성 유지 가능
  - ➢ U = $m \times m$ 크기의 열 기반 직교 행렬(orthogonal matrix)
  - ➢ Σ = $m \times n$ 크기의 대각 행렬(diagonal matrix) → 벡터의 길이를 조정하는 특이 값
  - ➢ V =$n \times n$ 크기의 행 기반 직교 행렬(orthogonal matrix)



A = U Σ V^{T}

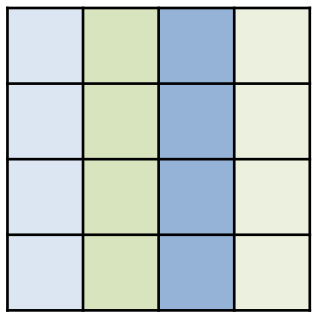
$m \times n$ $m \times m$ $m \times n$ $n \times n$

Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

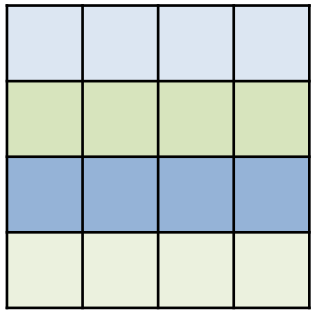
❖ 모델 훈련 과정 : ② $L_{RSD}(G_f)$ – SVD

- 특이값 분해(SVD)란? = 임의의 $m \times n$ 차원의 행렬 A는 $A = \mathrm{U}\Sigma\mathrm{V}^{\mathrm{T}}$ 와 같이 분해하는 방식 → 차원 축소 & 특성 유지 가능
  - ➢ U = $m \times m$ 크기의 열 기반 직교 행렬(orthogonal matrix)
  - ➢ Σ = $m \times n$ 크기의 대각 행렬(diagonal matrix) → 벡터의 길이를 조정하는 특이 값
  - ➢ V =$n \times n$ 크기의 행 기반 직교 행렬(orthogonal matrix)



U

$m \times m$



U^{T}

$m \times m$

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

I

$m \times m$

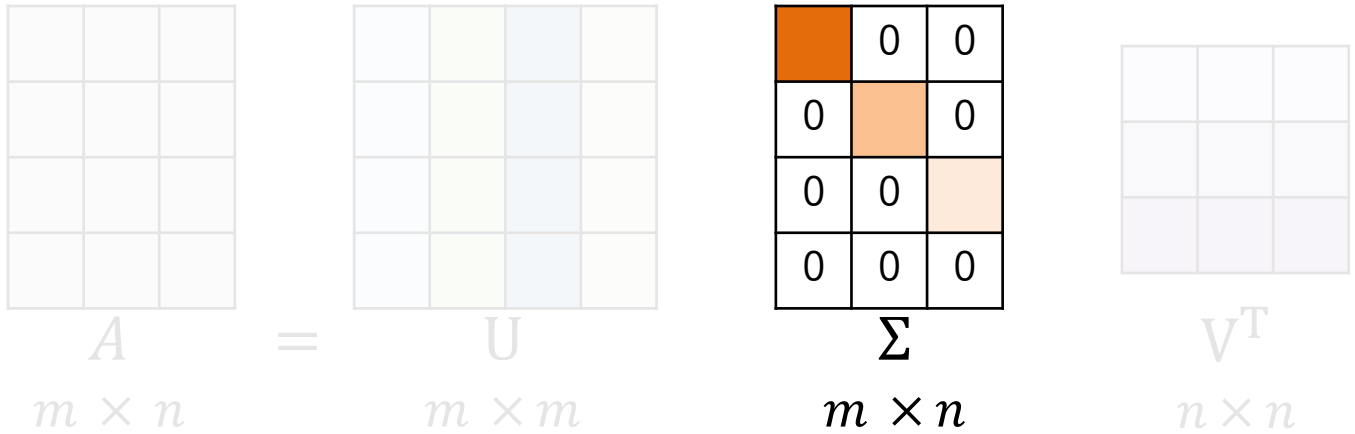
  - ➢ 직교 행렬(orthogonal matrix) = 자기 자신을 곱했을 때 Identity 행렬이 됨 ($\mathrm{U}^{\mathrm{T}}\mathrm{U} = \mathrm{I}, \mathrm{V}^{\mathrm{T}}\mathrm{V} = \mathrm{I}$)
  - ➢ 대각 행렬(diagonal matrix) = 주 대각선 상 위치 원소가 아닌 나머지 원소가 0인 행렬

Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 과정 : ② $L_{RSD}(G_f)$ – SVD

- 특이값 분해(SVD)란? = 임의의 $m \times n$ 차원의 행렬 A는 $A = \mathrm{U}\Sigma\mathrm{V}^{\mathrm{T}}$ 와 같이 분해하는 방식 → 차원 축소 & 특성 유지 가능
  - ➢ U = $m \times m$ 크기의 열 기반 직교 행렬(orthogonal matrix)
  - ➢ Σ = $m \times n$ 크기의 대각 행렬(diagonal matrix) → 벡터의 길이를 조정하는 특이 값
  - ➢ V =$n \times n$ 크기의 행 기반 직교 행렬(orthogonal matrix)



  - ➢ 직교 행렬(orthogonal matrix) = 자기 자신을 곱했을 때 Identity 행렬이 됨 ($\mathrm{U}^{\mathrm{T}}\mathrm{U} = \mathrm{I}, \mathrm{V}^{\mathrm{T}}\mathrm{V} = \mathrm{I}$)
  - ➢ 대각 행렬(diagonal matrix) = 주 대각선 상 위치 원소가 아닌 나머지 원소가 0인 행렬

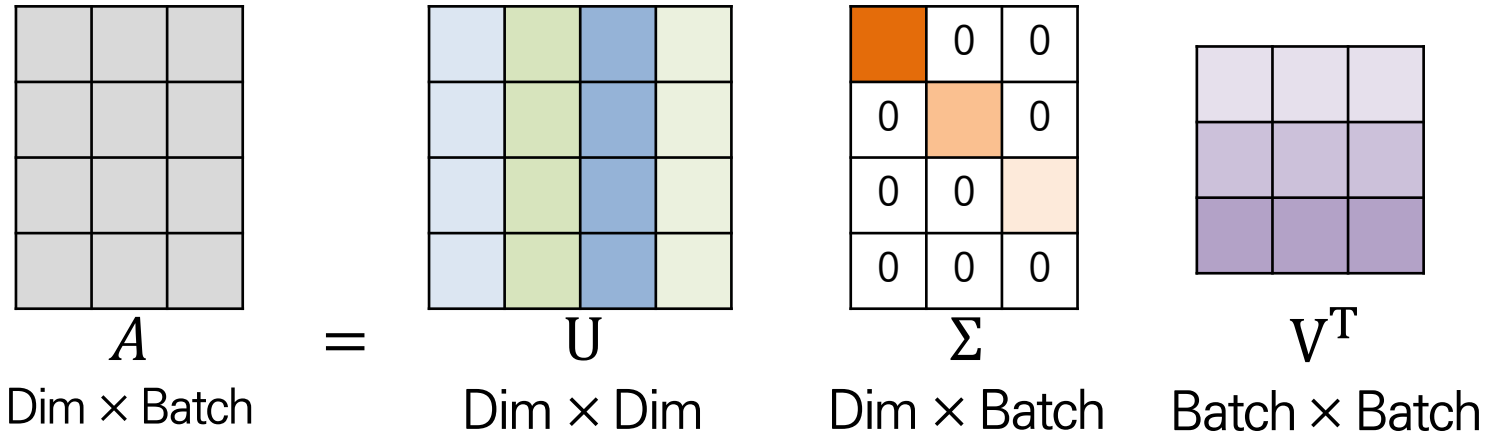
Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 과정 : ② $L_{RSD}(G_f)$ – SVD

- 특이값 분해(SVD)란? = 임의의 $m \times n$ 차원의 행렬 A는 $A = \mathrm{U}\Sigma\mathrm{V}^{\mathrm{T}}$ 와 같이 분해 가능함
  - ➢ U = 원본 행렬 A의 **열들이(=Dim) 공간 내에서 어떻게 분포하는지**(=서로 관련되어 있는지)를 설명
  - ➢ Σ = 원본 행렬 A의 데이터 구조에서 **스케일이 어떻게 되는지**
  - ➢ V = 원본 행렬 A의 **행들이(=Batch) 공간 내에서 어떻게 분포하는지**(=서로 관련되어 있는지)를 설명

"A = Backbone에서 얻은 Feature"



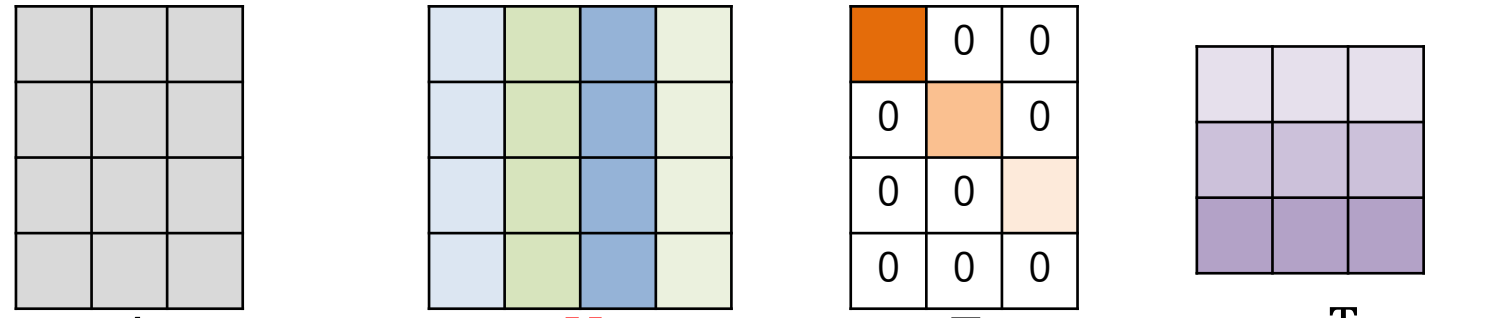
Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 과정 : ② $L_{RSD}(G_f)$ – SVD

- 특이값 분해(SVD)란? = 임의의 $m \times n$ 차원의 행렬 A는 $A = U\Sigma V^T$ 와 같이 분해 가능함
  - ➢ U = 원본 행렬 A의 **열들이(=Dim) 공간 내에서 어떻게 분포하는지**(=서로 관련되어 있는지)를 설명
  - ➢ Σ = 원본 행렬 A의 데이터 구조에서 **스케일이 어떻게 되는지**
  - ➢ V = 원본 행렬 A의 **행들이(=Batch) 공간 내에서 어떻게 분포하는지**(=서로 관련되어 있는지)를 설명

"A = Backbone에서 얻은 Feature"



Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

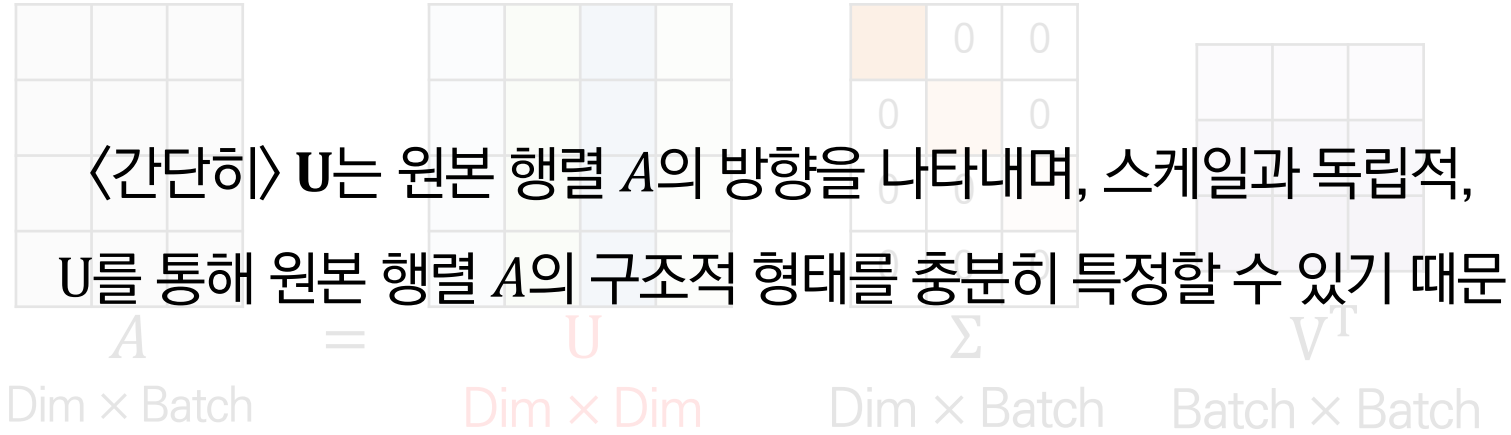
❖ 모델 훈련 과정 : ② $L_{RSD}(G_f)$ – SVD

- 특이값 분해(SVD)란? = 임의의 $m \times n$ 차원의 행렬 A는 $A = U\Sigma V^T$ 와 같이 분해 가능함
  - ➢ U = 원본 행렬 A의 열들이(=Dim) 공간 내에서 어떻게 분포하는지(=서로 관련되어 있는지)를 설명
  - ➢ Σ = 원본 행렬 A의 데이터 구조에서 스케일이 어떻게 되는지
  - ➢ V = 원본 행렬 A의 행들이(=Batch) 공간 내에서 어떻게 분포하는지(=서로 관련되어 있는지)를 설명

그라스만 다양체(Grassmann manifold)의 리만 기하학(Riemannian geometry) 이론을 기반으로

U인 열 기반 직교 행렬만을 사용

"A = Backbone에서 얻은 Feature"



〈간단히〉 **U**는 원본 행렬 A의 방향을 나타내며, 스케일과 독립적,

U를 통해 원본 행렬 A의 구조적 형태를 충분히 특정할 수 있기 때문

Methods

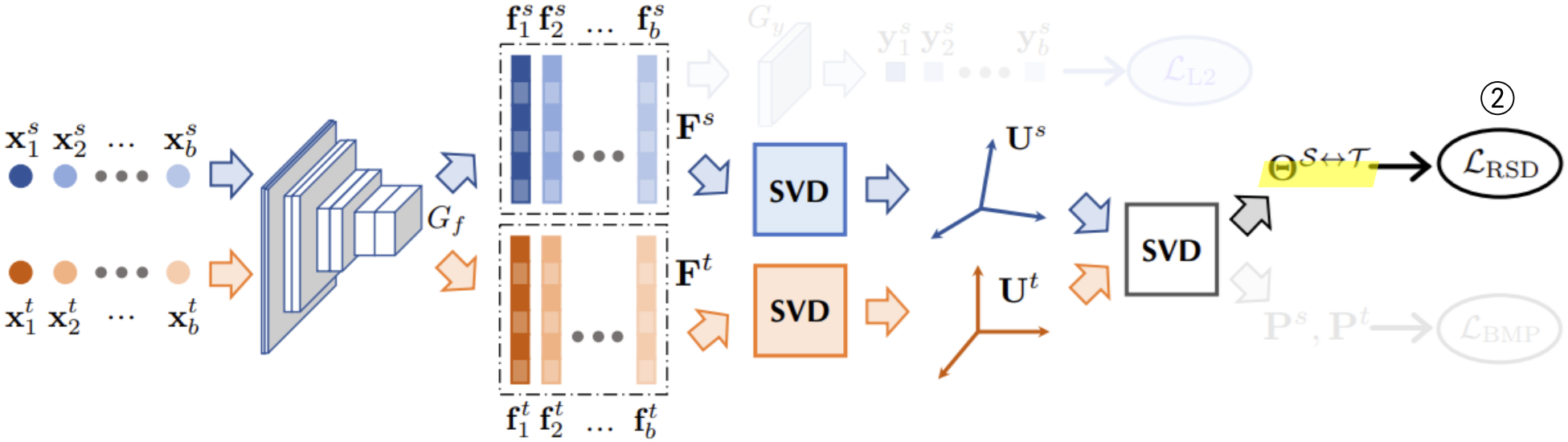
Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 과정 : ② $L_{RSD}(G_f)$

- $L_{RSD}(G_f) = \mathrm{dis}_{RSD}(\mathrm{U^s}, \mathrm{U^t}) = \left\|\sin \Theta^{S \leftrightarrow T}\right\|_1$
  - ➢ SVD(특이값 분해)로 $\mathrm{F^s}$와 $\mathrm{F^t}$로부터 각각 $\mathrm{U^s}$와 $\mathrm{U^t}$를 도출
  - ➢ Golub & Van Loan(1996)에 제시한 주 각도(principal angles) 계산식 $\mathrm{U^s U^t} = \mathrm{P^s}(\mathrm{diag}(\cos \Theta^{S \leftrightarrow T}))\,\mathrm{P^t}$ 을 사용

주 각도(principal angles)는 ③ $L_{BMP}(G_f)$에서 사용

두 도메인 사이의 관계를 나타냄



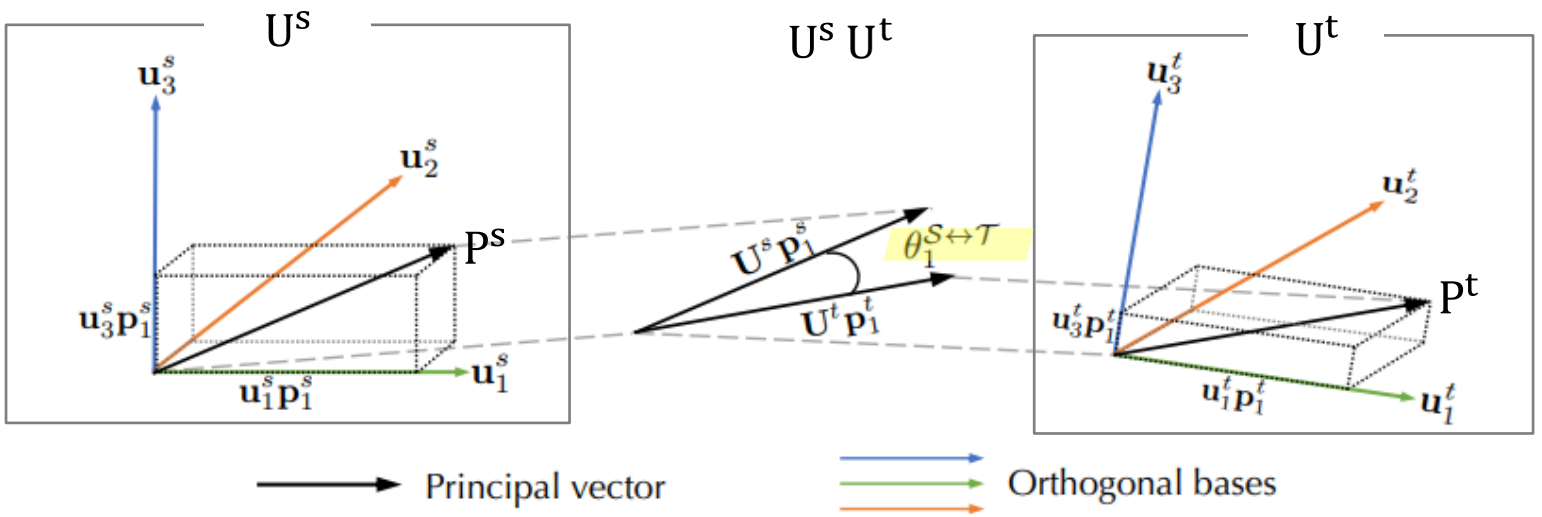
Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 과정 : ② $L_{RSD}(G_f)$

- $L_{RSD}(G_f) = \mathrm{dis}_{RSD}(\mathrm{U^s}, \mathrm{U^t}) = \left\| \sin \Theta^{S \leftrightarrow T} \right\|_1$
  - SVD(특이값 분해)로 $\mathrm{F^s}$와 $\mathrm{F^t}$로부터 각각 $\mathrm{U^s}$와 $\mathrm{U^t}$를 도출
  - Golub & Van Loan(1996)에 제시한 주 각도(principal angles) 계산식 $\mathrm{U^s U^t} = \mathrm{P^s}(\mathrm{diag}(\cos \Theta^{S \leftrightarrow T}))\ \mathrm{P^t}$ 을 사용

    → 도출 된 $\cos \Theta^{S \leftrightarrow T}$ 다음 식에 대입하여 $sin\theta = \sqrt{1 - \cos^2(\theta)}$ 구하고자 했던 $\sin \Theta^{S \leftrightarrow T}$을 도출한 후 L1 norm $\|\cdot\|_1$을 취함
  - L_{RSD} 결론 : 두 도메인간 차이를 Loss로 두어 유사해지게!

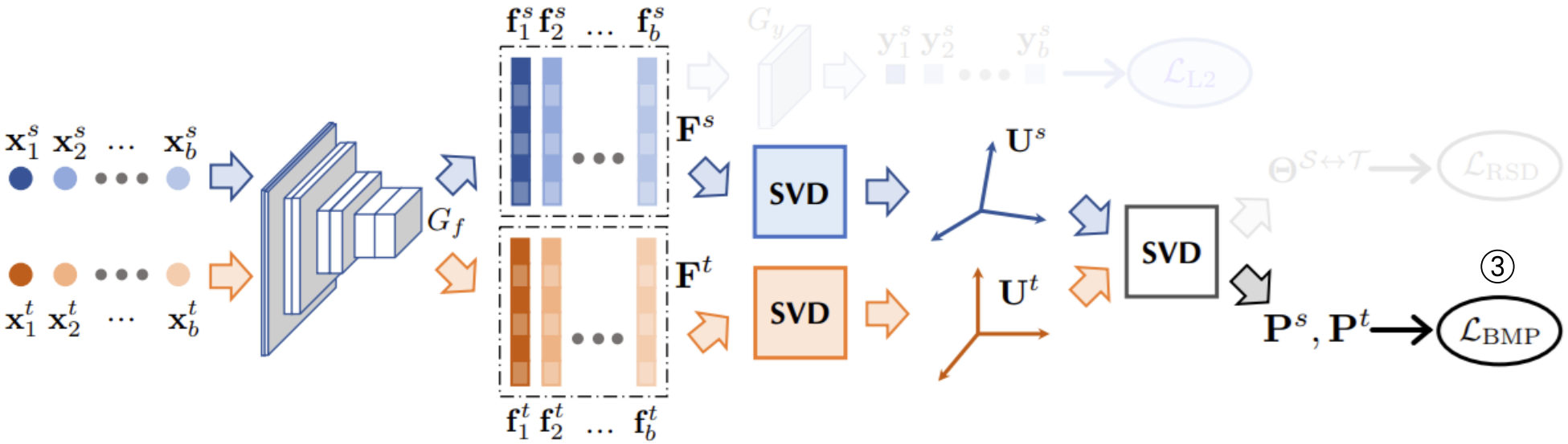


Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 과정 : ③ $L_{BMP}(G_f)$

- $L_{BMP}(G_f) = \mathrm{reg}_{BMP}^{S\leftrightarrow T}(\mathrm{U}^s, \mathrm{U}^t) = \| |\mathrm{P}^s| - |\mathrm{P}^t| \|_F^2$
  - ➢ 다시 ! Golub & Van Loan(1996)에 제시한 주 각도(principal angles) 계산식 $\mathrm{U}^s\mathrm{U}^t = \mathrm{P}^s(\mathrm{diag}(\cos\Theta^{S\leftrightarrow T}))\,\mathrm{P}^t$ 을 사용
  - ➢ ②에서 계산한 주 각도 P^s, P^t의 절대 값 차이를 계산 후, 프로베니우스 놈 $\|\cdot\|_F$을 취한 뒤 제곱을 구함

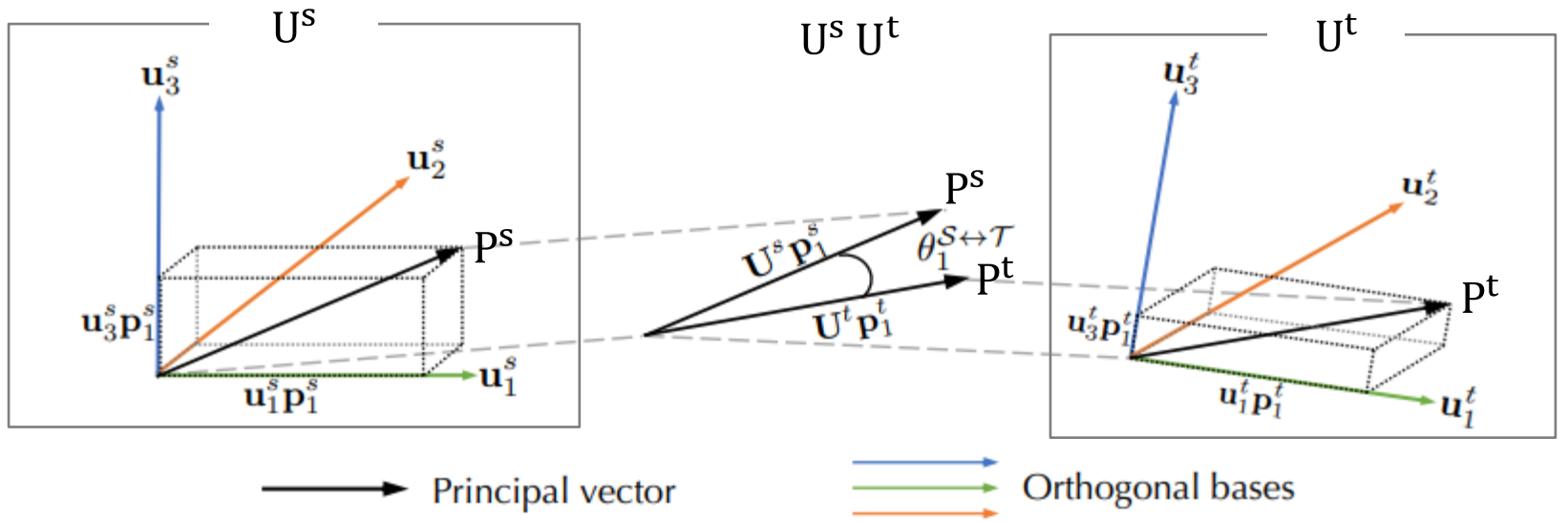


Methods

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 모델 훈련 과정 : ③ $L_{BMP}(G_f)$

- $L_{BMP}(G_f) = \mathrm{reg}_{BMP}^{S\leftrightarrow T}(\mathrm{U^s}, \mathrm{U^t}) = \| |\mathrm{P}^s| - |\mathrm{P}^t| \|_F^2$
  - $\mathrm{P^s}$는 소스 도메인에 대한, $\mathrm{P^t}$는 타겟 도메인에 대한 주 각도
  - $\mathrm{U^s}$와 $\mathrm{U^t}$는 중요도의 순서를 고려하고 있지 않음
  - $\mathrm{P^s}$와 $\mathrm{P^t}$는 직교 기저(ex. u_1^s, u_1^t 등)의 중요도(=가중치 ex. p_1^s, p_1^t)를 고려한 가중합임
  - L_{BMP} 결론 : 주 각 $\mathrm{P^s}$, $\mathrm{P^t}$가 유사해야 한다! ex) 소스 도메인의 기저가 중요하다고 판단 → 타겟 도메인의 기저도 중요하게



Experiments

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 데이터 셋 (1/2)

- dSprites – 2D 합성 데이터셋
  - ➢ Color (C), Noisy (N), Scream (S) 총 3개의 도메인으로 실험
  - ➢ Target은 regression task 중 Orientation을 제외한 3개로 실험(Scale, Position X and Y)
    - ✓ 도형 이미지기 때문에 방향 값을 결정이 까다로움
- MPI3D – 3D 시뮬레이션 데이터셋
  - ➢ Toy (T), RealistiC (RC), ReaL (RL) 총 3개의 도메인으로 실험
  - ➢ Target은 regression task 2개로 실험(Horizontal and Vertical Axis)

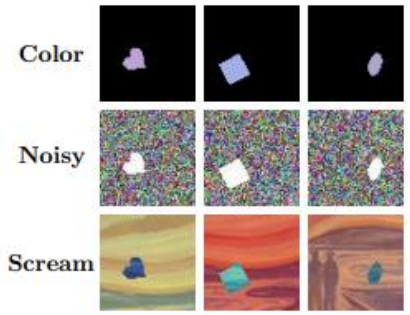


Table 1. Factors of variations in dSprites

Factor	Possible Values	Task
Shape	square, ellipse, heart	recognition
Scale	6 values in $[0.5, 1]$	regression
Orientation	40 values in $[0, 2\pi]$	regression
Position X	32 values in $[0, 1]$	regression
Position Y	32 values in $[0, 1]$	regression

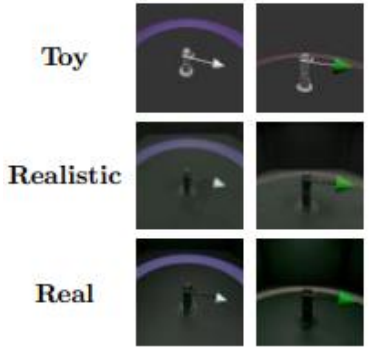


Table 2. Factors of variations in MPI3D

Factor	Possible Values	Task
Object Color	5 values	recognition
Object Shape	6 values	recognition
Object Size	2 values	recognition
Camera Height	3 values	recognition
Background Color	3 values	recognition
Horizontal Axis	40 values in $[0, 1]$	regression
Vertical Axis	40 values in $[0, 1]$	regression

Experiments

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 데이터 셋 (2/2)

- Biwi Kinect – 머리 자세 추정 이미지 데이터 셋
  - ➢ Female (F), Male (M) 총 2개의 도메인으로 실험
  - ➢ Target은 regression task 3개로 실험(Pitch, Yaw, Roll)

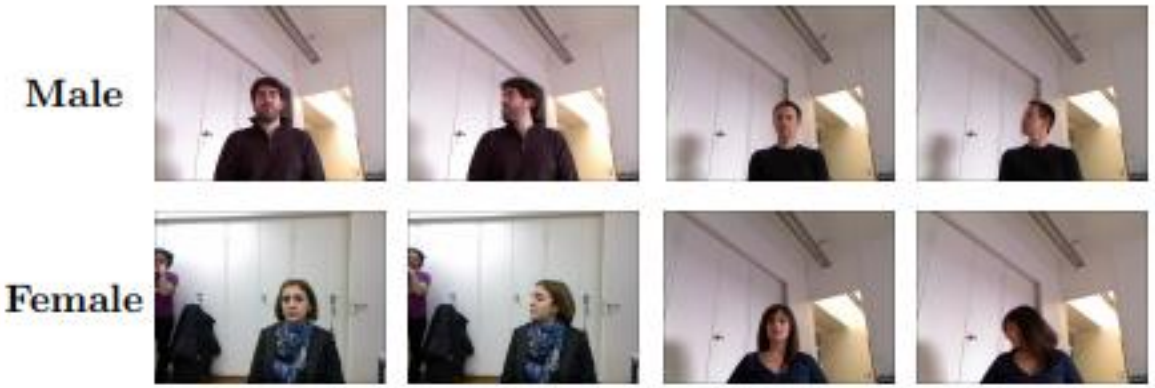


Table 3. Factors of variations in Biwi Kinect

Factor	Possible Values	Task
Pitch	values in $[-92.044, 231.352]$	regression
Yaw	values in $[-87.7066, 246.684]$	regression
Roll	values in $[754.182, 1297.45]$	regression

Experiments

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 실험 결과

- 3개 데이터 셋에서 대부분 $L_{RSD}(G_f) + L_{BMP}(G_f)$ 인 제안 방법론이 가장 좋은 성능을 보임

Table 4. Sum of MAE across three regression tasks on dSprites: unsupervised domain adaptation (ResNet-18).

Method	C → N	C → S	N → C	N → S	S → C	S → N	Avg
ResNet-18 (He et al., 2016)	0.94 ± 0.06	0.90 ± 0.08	0.16 ± 0.02	0.65 ± 0.02	0.08 ± 0.01	0.26 ± 0.03	0.498
TCA (Pan et al., 2011)	0.94 ± 0.03	0.87 ± 0.02	0.19 ± 0.02	0.66 ± 0.05	0.10 ± 0.02	0.23 ± 0.04	0.498
DAN (Long et al., 2015)	0.70 ± 0.05	0.77 ± 0.09	**0.12** ± 0.03	0.50 ± 0.05	0.06 ± 0.02	0.11 ± 0.04	0.377
DANN (Ganin et al., 2016)	0.47 ± 0.07	0.46 ± 0.07	0.16 ± 0.02	0.65 ± 0.05	**0.05** ± 0.00	0.10 ± 0.01	0.315
JDOT (Courty et al., 2017)	0.86 ± 0.03	0.79 ± 0.02	0.19 ± 0.02	0.64 ± 0.05	0.10 ± 0.02	0.23 ± 0.04	0.468
MCD (Saito et al., 2018)	0.81 ± 0.09	0.81 ± 0.12	0.17 ± 0.12	0.65 ± 0.03	0.07 ± 0.02	0.19 ± 0.04	0.450
AFN (Xu et al., 2019)	1.00 ± 0.04	0.96 ± 0.05	0.16 ± 0.03	0.62 ± 0.04	0.08 ± 0.01	0.32 ± 0.06	0.523
RSD (ours)	0.32 ± 0.02	0.35 ± 0.02	0.16 ± 0.02	0.57 ± 0.01	0.08 ± 0.01	0.09 ± 0.02	0.258
RSD+BMP (ours)	**0.31** ± 0.03	**0.31** ± 0.03	**0.12** ± 0.02	**0.53** ± 0.01	0.07 ± 0.00	**0.08** ± 0.01	**0.237**

Table 5. Sum of MAE across two regression tasks on MPI3D: unsupervised domain adaptation (ResNet-18).

Method	RL → RC	RL → T	RC → RL	RC → T	T → RL	T → RC	Avg
ResNet-18 (He et al., 2016)	0.17 ± 0.02	0.44 ± 0.04	0.19 ± 0.02	0.45 ± 0.03	0.51 ± 0.01	0.50 ± 0.03	0.377
TCA (Pan et al., 2011)	0.17 ± 0.02	0.42 ± 0.01	0.19 ± 0.02	0.42 ± 0.02	0.50 ± 0.02	0.50 ± 0.02	0.373
DAN (Long et al., 2015)	0.12 ± 0.03	0.35 ± 0.02	0.12 ± 0.02	0.27 ± 0.02	0.40 ± 0.02	0.41 ± 0.04	0.278
DANN (Ganin et al., 2016)	**0.09** ± 0.01	0.24 ± 0.04	0.11 ± 0.03	0.41 ± 0.03	0.48 ± 0.02	0.37 ± 0.04	0.283
JDOT (Courty et al., 2017)	0.16 ± 0.02	0.41 ± 0.01	0.16 ± 0.02	0.41 ± 0.02	0.47 ± 0.02	0.47 ± 0.02	0.353
MCD (Saito et al., 2018)	0.13 ± 0.02	0.40 ± 0.04	0.15 ± 0.02	0.45 ± 0.01	0.52 ± 0.02	0.50 ± 0.03	0.358
AFN (Xu et al., 2019)	0.18 ± 0.03	0.45 ± 0.02	0.20 ± 0.03	0.46 ± 0.03	0.53 ± 0.02	0.52 ± 0.04	0.390
RSD (ours)	0.10 ± 0.01	0.23 ± 0.03	0.11 ± 0.01	0.17 ± 0.02	0.41 ± 0.01	0.42 ± 0.01	0.242
RSD+BMP (ours)	**0.09** ± 0.01	**0.19** ± 0.02	**0.08** ± 0.00	**0.15** ± 0.03	**0.36** ± 0.01	**0.36** ± 0.02	**0.205**

Table 6. Sum of MAE across three regression tasks on Biwi Kinect

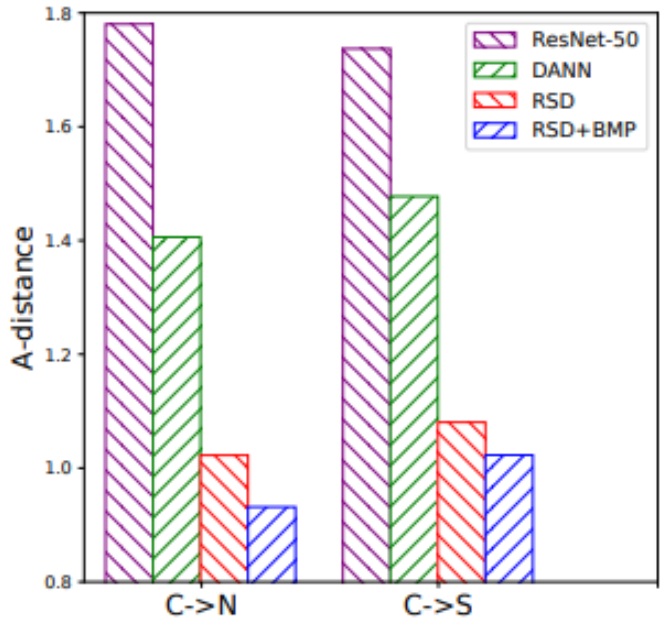
Method	F → M	M → F
ResNet-18 (He et al., 2016)	0.38 ± 0.02	0.29 ± 0.01
TCA (Pan et al., 2011)	0.39 ± 0.01	0.31 ± 0.01
DAN (Long et al., 2015)	0.37 ± 0.01	0.28 ± 0.01
DANN (Ganin et al., 2016)	0.37 ± 0.02	0.30 ± 0.01
JDOT (Courty et al., 2017)	0.39 ± 0.01	0.29 ± 0.02
MCD (Saito et al., 2018)	0.37 ± 0.02	0.31 ± 0.02
AFN (Xu et al., 2019)	0.41 ± 0.02	0.32 ± 0.02
RSD (ours)	0.33 ± 0.02	0.27 ± 0.01
RSD+BMP (ours)	**0.30** ± 0.02	**0.26** ± 0.01

Experiments

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 실험 결과

- 도메인 분포 차이 거리 측정 결과
  - ➢ Source와 Target을 구별하도록 훈련된 Classifier의 test error ϵ를 사용하여 A-distance = $1 - 2\epsilon$ 계산
  - ➢ 제안 방법론이 A-distance가 제일 적은 것으로 보여 짐



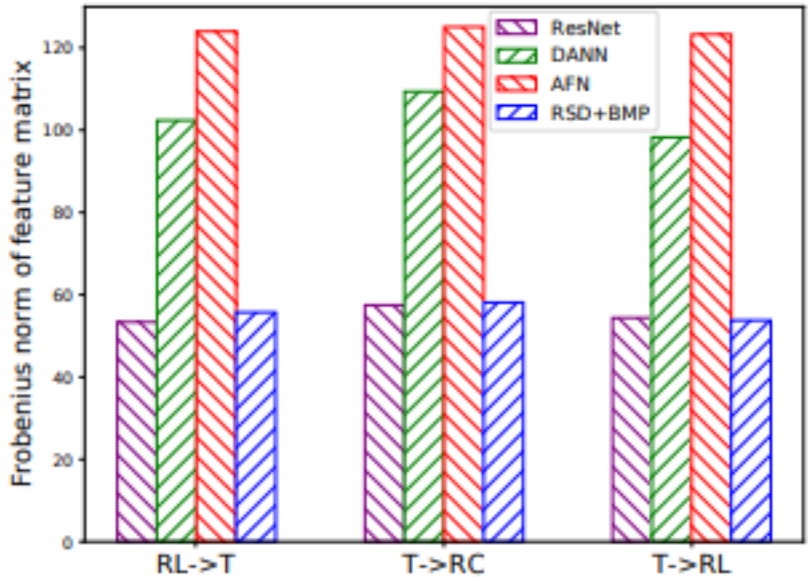
(a) A-distance

Experiments

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

❖ 실험 결과

- Feature scale 정도
  - ➢ Source feature의 프로베니우스 놈 평균을 구한 결과
  - ➢ 제안 방법론의 경우 classification 방법론과 다르게 feature scale이 거의 변하지 않은 것을 볼 수 있음



(a) Feature Scale

Experiments

Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)

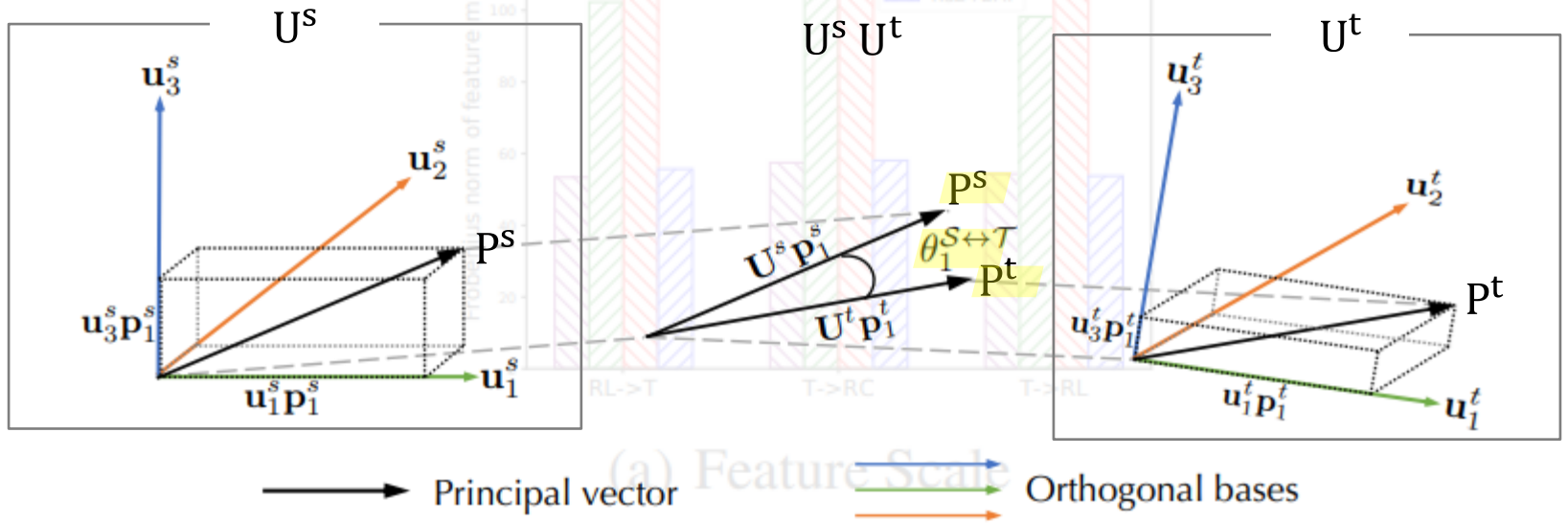
❖ 실험 결과

- Feature scale 정도
  - ➢ Source feature의 프로베니우스 놈 평균을 구한 결과
  - ➢ 제안 방법론의 경우 feature scale이 Unsupervised domain adaptation classification과 다르게 변하지 않은 것을 볼 수 있음

방법론 정리

① 회귀의 경우 feature scale에 민감하며, 분류 방법론 적용 시 feature scale의 변동 Risk를 실험으로 입증

② SVD를 통해 새로운 거리를 정의하며 차이를 줄이 돼, feature scale은 유지하여 좋은 성능을 달성



DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

Methods

❖ DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

- 회귀 문제의 경우 Inverse gram matrix(역행렬)를 통해 모델을 학습해야 한다는 근거를 제시
- Pseudo-inverse gram matrix(유사 역행렬)을 모델 학습에 사용하여 도메인간 거리를 줄여 성능을 높임

2024.04.12 기준 6회 인용

InterpretML: A Unified Framework for Machine Learning Interpretability

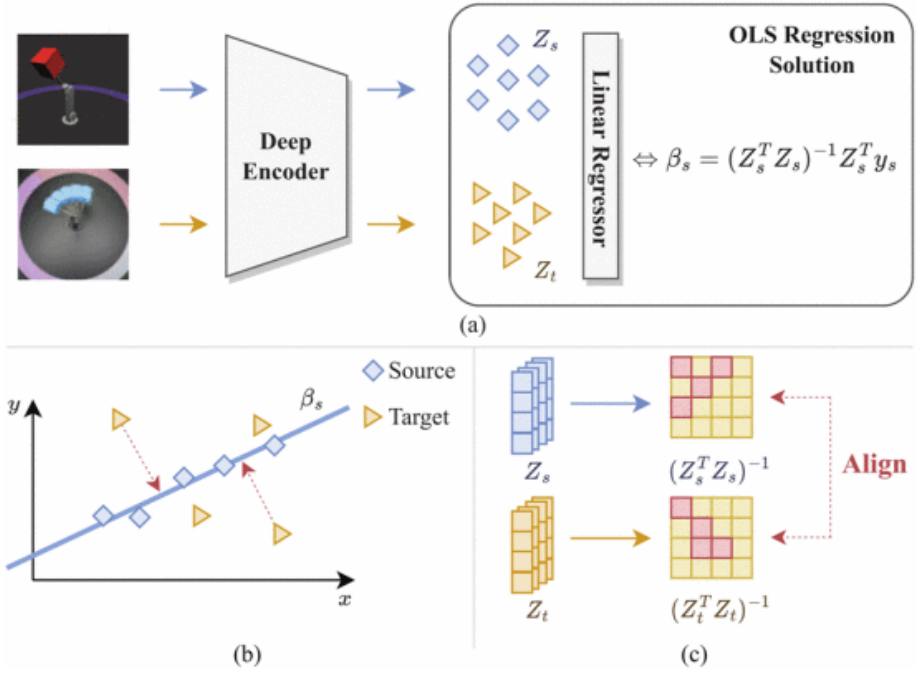
Harsha Nori HANORI@MICROSOFT.COM
Samuel Jenkins SAJENKIN@MICROSOFT.COM
Paul Koch PAULKOCH@MICROSOFT.COM
Rich Caruana RCARUANA@MICROSOFT.COM
Microsoft Corporation
1 Microsoft Way
Redmond, WA 98052, USA

Editor:

Abstract

InterpretML is an open-source Python package which exposes machine learning interpretability algorithms to practitioners and researchers. InterpretML exposes two types of interpretability – `glassbox`, which are machine learning models designed for interpretability (ex: linear models, rule lists, generalized additive models), and `blackbox` explainability techniques for explaining existing systems (ex: Partial Dependence, LIME). The package enables practitioners to easily compare interpretability algorithms by exposing multiple methods under a unified API, and by having a built-in, extensible visualization platform. InterpretML also includes the first implementation of the Explainable Boosting Machine, a powerful, interpretable, glassbox model that can be as accurate as many blackbox models. The MIT licensed source code can be downloaded from github.com/microsoft/interpret.

Keywords: Interpretability, Explainable Boosting Machine, Glassbox, Blackbox

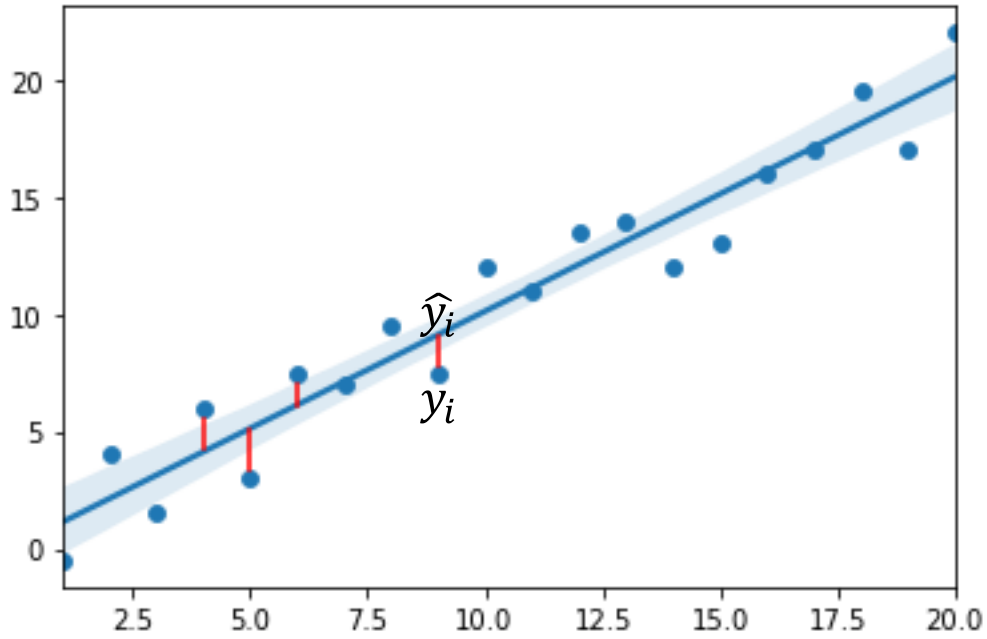


Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ Inverse gram matrix(역행렬)를 통해 모델을 학습해야 한다는 근거

- 최소제곱법(OLS) 이란
  - ➢ 선형 모델 예측 값과 실제 값 간 제곱의 합(Sum of Squared Errors, SSE)를 최소화하는 모델의 파라미터(β)를 찾아내는 기법
- $SSE = \sum \epsilon^2 = \sum (y - \hat{y})^2$

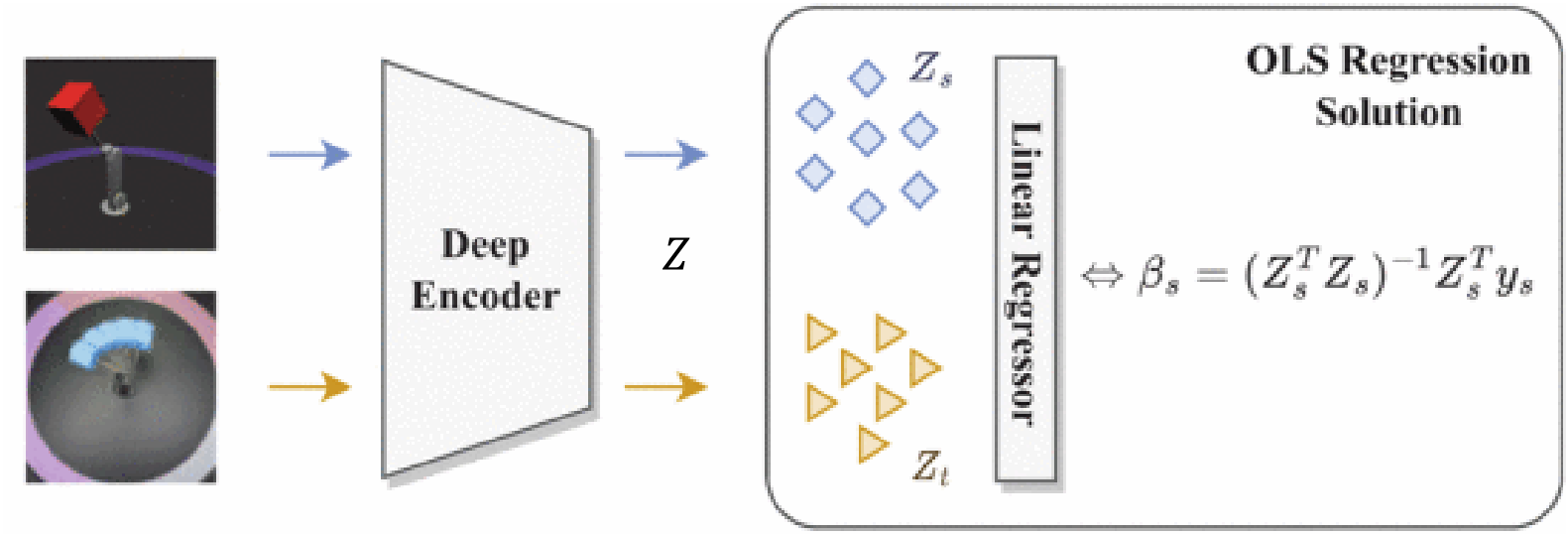


Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ 최소제곱법(OLS)의 inverse Gram matrix를 구하는 식

- $SSE = \sum \epsilon^2 = \sum (y - \hat{y})^2$를 행렬 기준으로 변환, $\hat{Y} = Z\beta$ (Z는 feature, β는 파라미터)



Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ 최소제곱법(OLS)의 inverse Gram matrix를 구하는 식

- $SSE = \sum \epsilon^2 = \sum (y - \hat{y})^2$를 행렬 기준으로 변환 → $Y = Z\beta$ (Z는 feature, β는 파라미터)
- $$SSE = (Y - Z\beta)^{\mathrm{T}}(Y - Z\beta)$$
$$= Y^TY - Y^TZ\beta - Z^T\beta^TY + Z^T\beta^TZ\beta$$
$$= Y^TY - 2\beta^TZ^TY + Z^TZ\,\beta^T\beta$$
- $\frac{\partial SSE}{\partial \beta} = 2(Z^TY) - 2(Z^TZ)\beta = 0$ ← 미분

$$2(Z^TZ)\beta = 2(Z^TY)$$

$$\hat{\beta} = (Z^TZ)^{-1}Z^TY$$

Z^TZ가 가역행렬(역행렬이 존재)이라고 가정

Inverse gram matrix(역행렬) → 회귀 task에서 역행렬을 사용하는 것이 올바를 것이다는 근거 1)

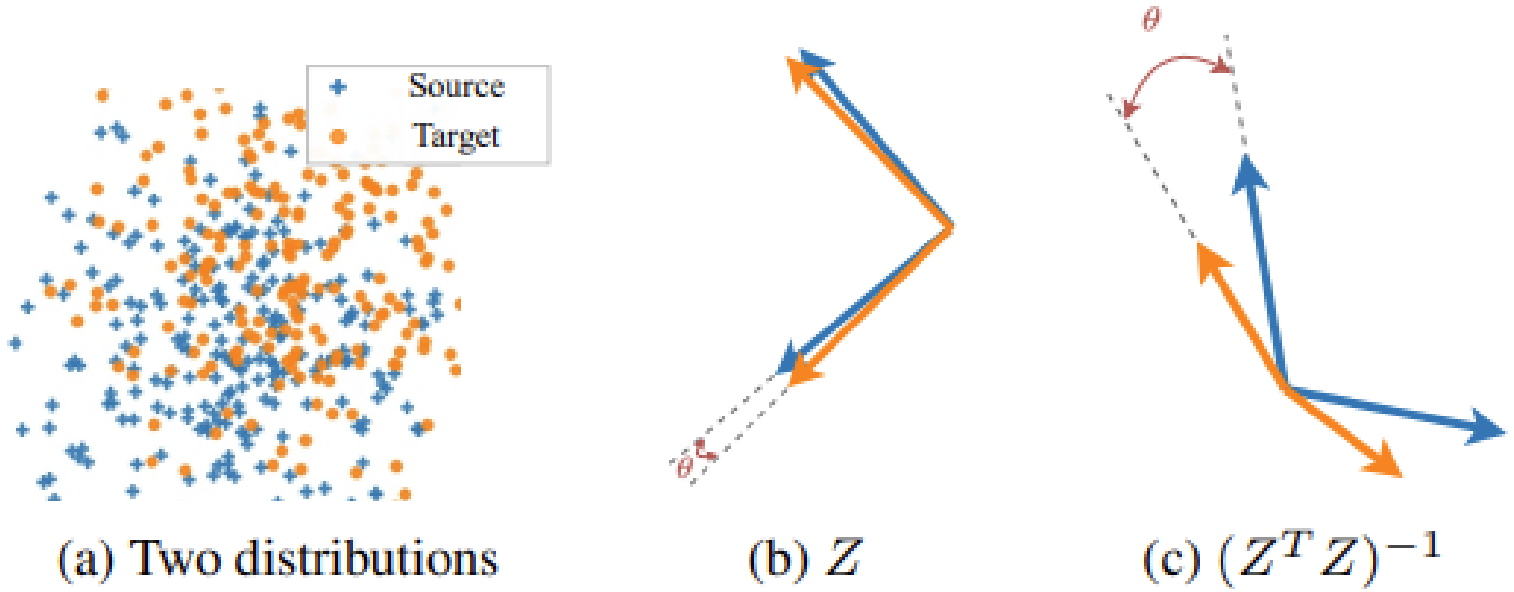
Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ 인스턴스 기반 feature와 inverse gram matrix(역행렬) 차이

- 소스와 타겟 도메인의 feature인 Z_s와 Z_t의 cosine 거리를 구하면 그림 (b)의 경우 작지만
- Inverse Gram matrix로 변환했을 때 $(Z_s^T Z_s)^{-1}$ 와 $(Z_t^T Z_t)^{-1}$의 거리는 그림 (c)와 같이 커짐

  → 회귀 task에서 역행렬을 사용하는 것이 올바를 것이다는 근거 2)



(a) Two distributions (b) Z (c) $(Z^T Z)^{-1}$

Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ 모델 훈련 단계

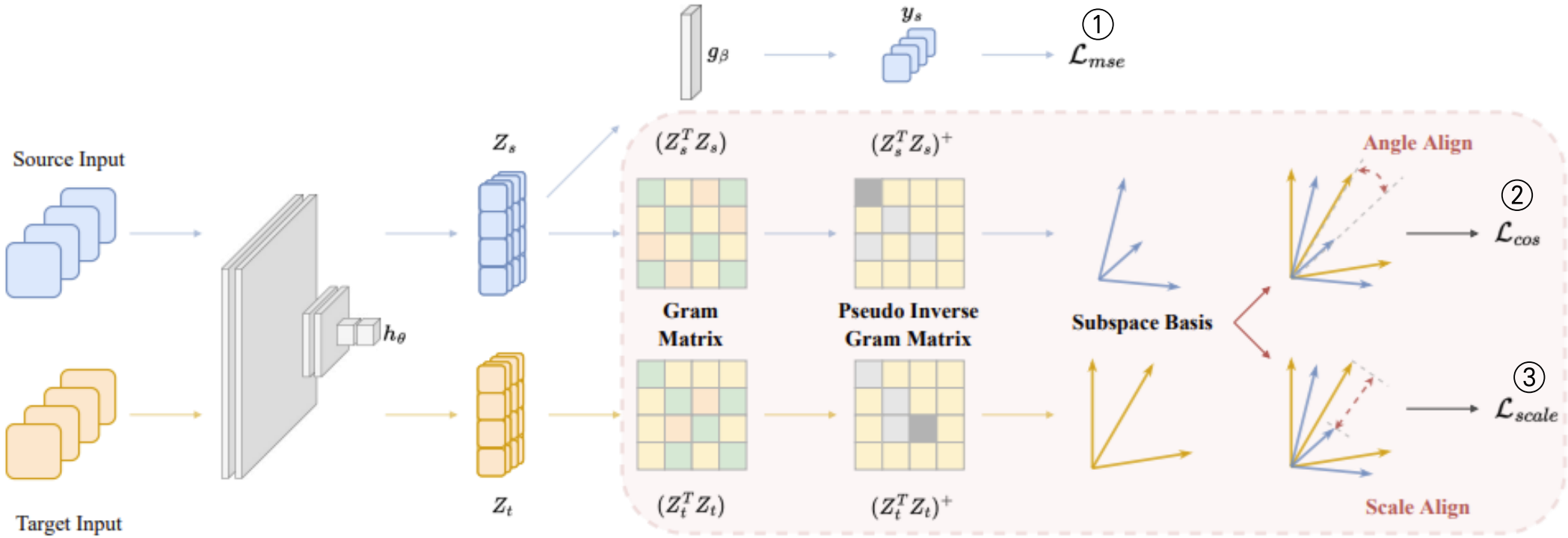
α_{cos} and γ_{scale} = hyperparameters

- Total loss = $L_{src} + \alpha_{cos} L_{cos}(Z^S, Z^T) + \gamma_{scale} L_{scale}(Z^S, Z^T)$

① ② ③

② 코사인 유사성 정렬

③ 스케일 정렬



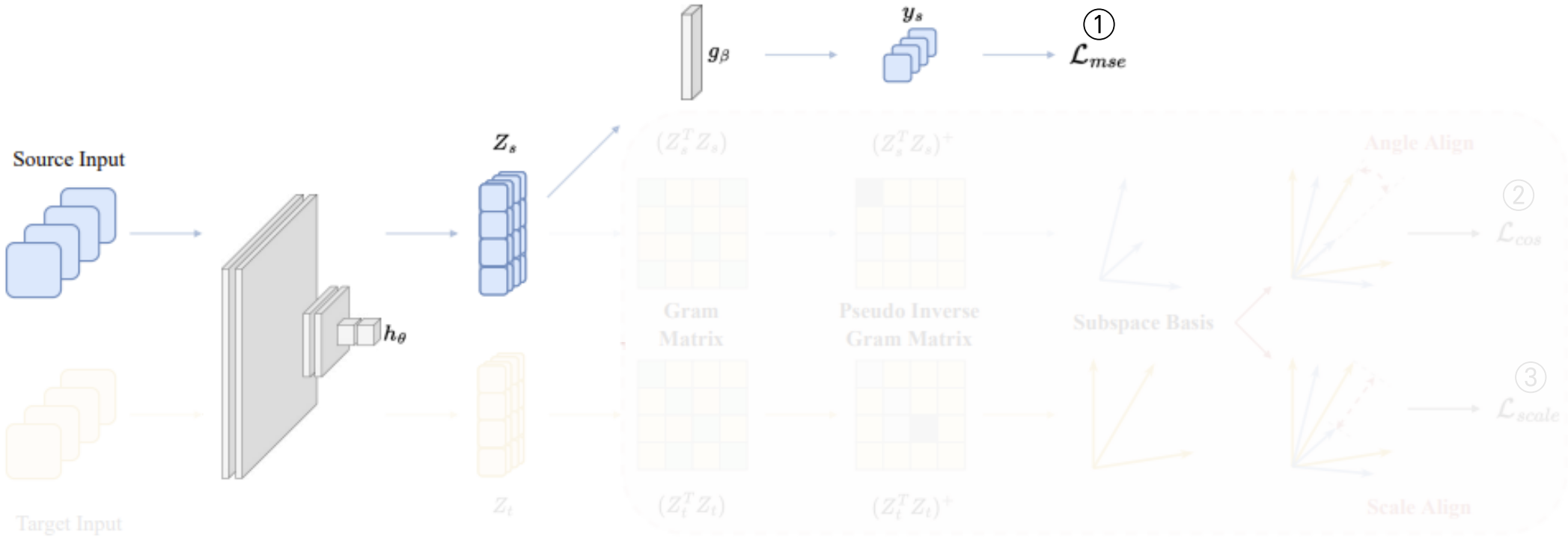
Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ 모델 훈련 단계 : ① L_{src}

- Total loss = $L_{src} + \alpha_{cos} L_{cos}(Z^S, Z^T) + \gamma_{scale} L_{scale}(Z^S, Z^T)$

① L_{src} = 일반적인 supervised loss = $MSE\left(g_\beta(h_\theta(x^S),)y^S\right) = MSE(\hat{y}^S,\ y^S)$



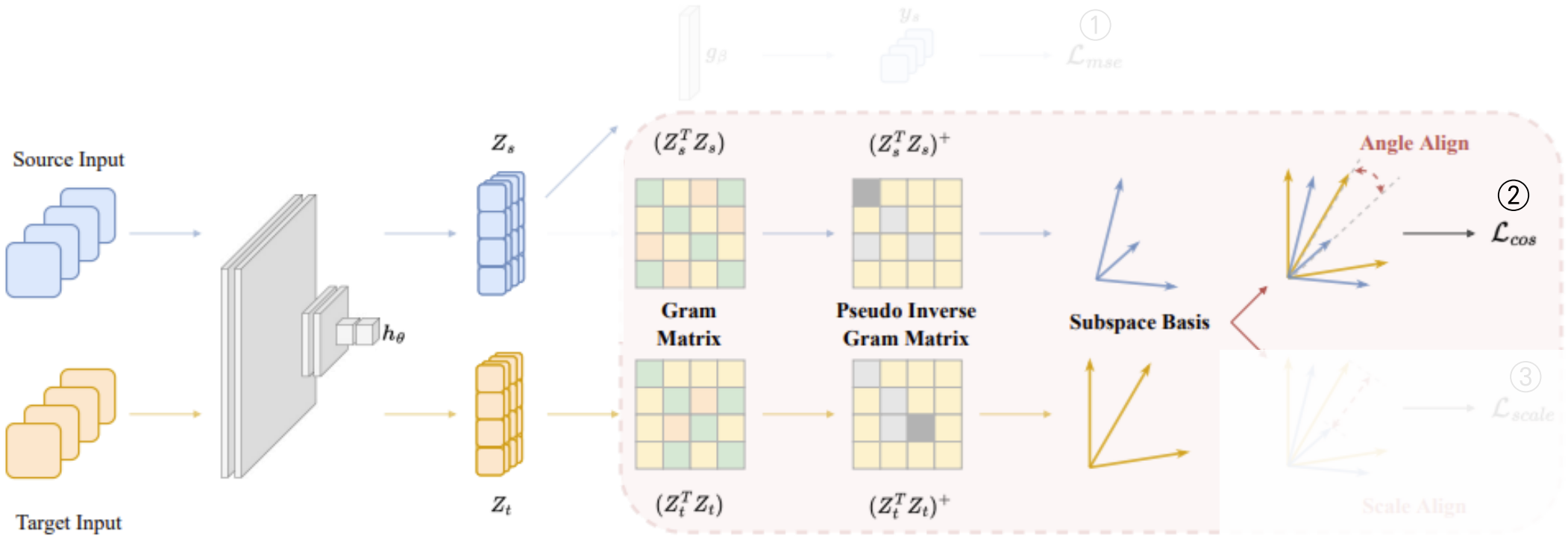
Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ 모델 훈련 단계 : ② $L_{cos}(Z^S, Z^T)$

- Total loss = $L_{src} + \alpha_{cos} L_{cos}(Z^S, Z^T) + \gamma_{scale} L_{scale}(Z^S, Z^T)$

② $L_{cos}(Z^S, Z^T)$= 도메인간 코사인 유사성 정렬

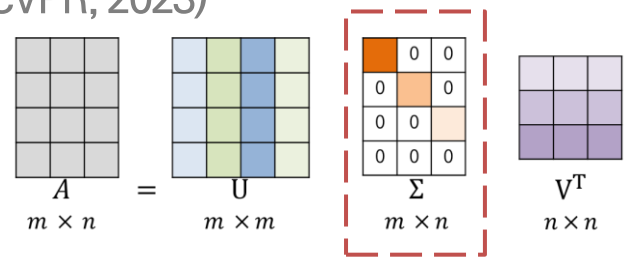


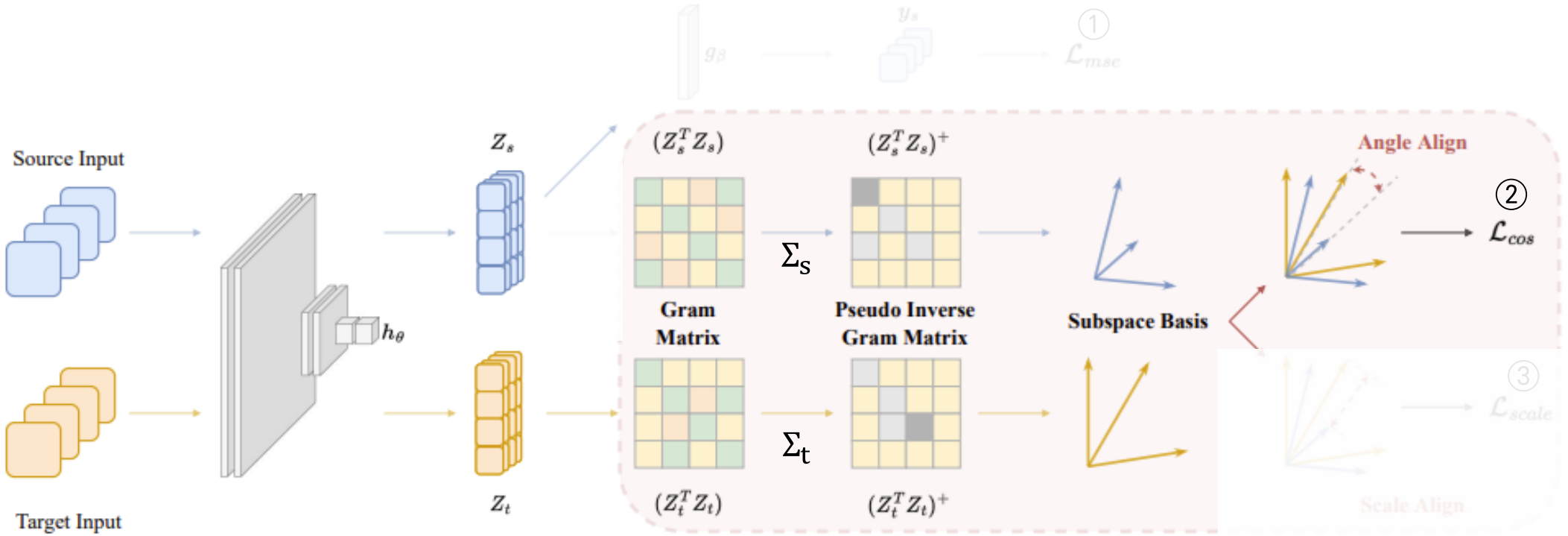
Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ **모델 훈련 단계 :** ② $L_{cos}(Z^s, Z^T)$

- Gram matrix $(Z^T Z)$ 를 구한 뒤, SVD로 특이 값 Σ (diagonal matrix)를 계산
- Σ 중 threshold(하이퍼파라미터) 이상의 분산을 가진 특이 값 $\lambda_1, \dots, \lambda_k$만을 선택
- 선택한 받지 못한 특이 값의 역수를 0으로 설정하는 유사 역행렬 Pseudo-inverse gram matrix $(Z, Z^T)^+$ 도출





Methods

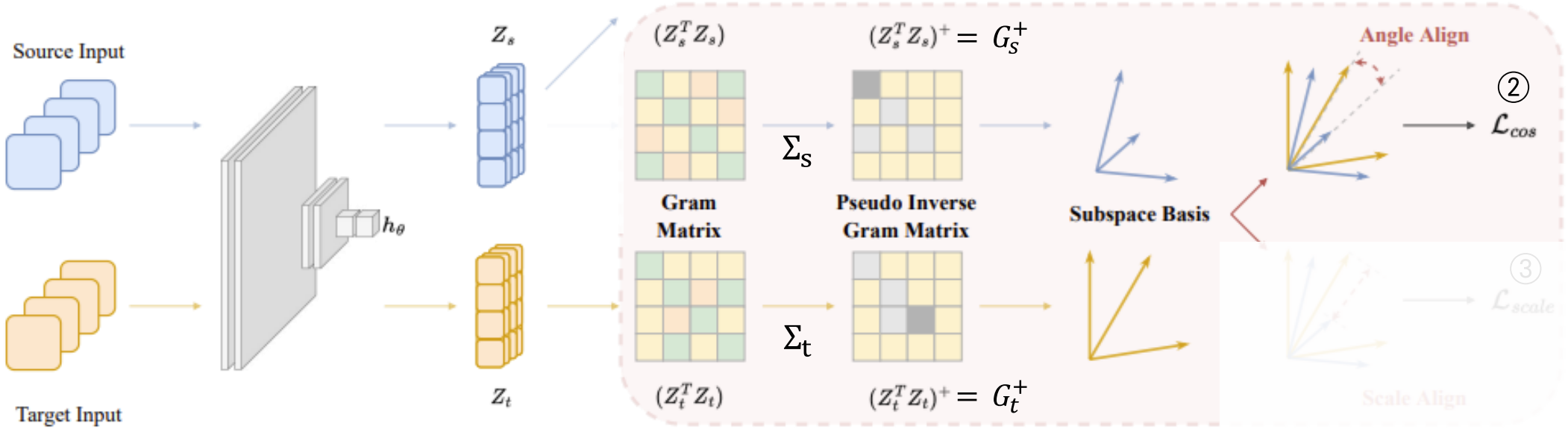
DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ **모델 훈련 단계 :** ② $L_{cos}(Z^S, Z^T)$

- Pseudo-inverse gram matrix $(Z, Z^T)^+ = V\,\Sigma^+ U^T = V \begin{pmatrix} \frac{1}{\lambda_1} & & & 0 \\ & \ddots & & \\ & & \frac{1}{\lambda_k} & \\ 0 & & & 0 \end{pmatrix} U^T = G^+$

$\|\cdot\|_1$ 은 L1 norm

- 각각의 G_s^+, G_t^+로 부터 코사인 유사도를 계산 $\cos(\theta_i^{S\leftrightarrow T})$하여 ② $L_{cos}(Z^S, Z^T) = \left\|1_i - \cos(\theta_i^{S\leftrightarrow T})\right\|_1$ 를 계산
- $L_{cos}(Z^S, Z^T)$ 의 결론 : 1_i은 이상적인 유사도 의미. Source와 Target의 angle이 같고, 1이 되게 정렬 하자!



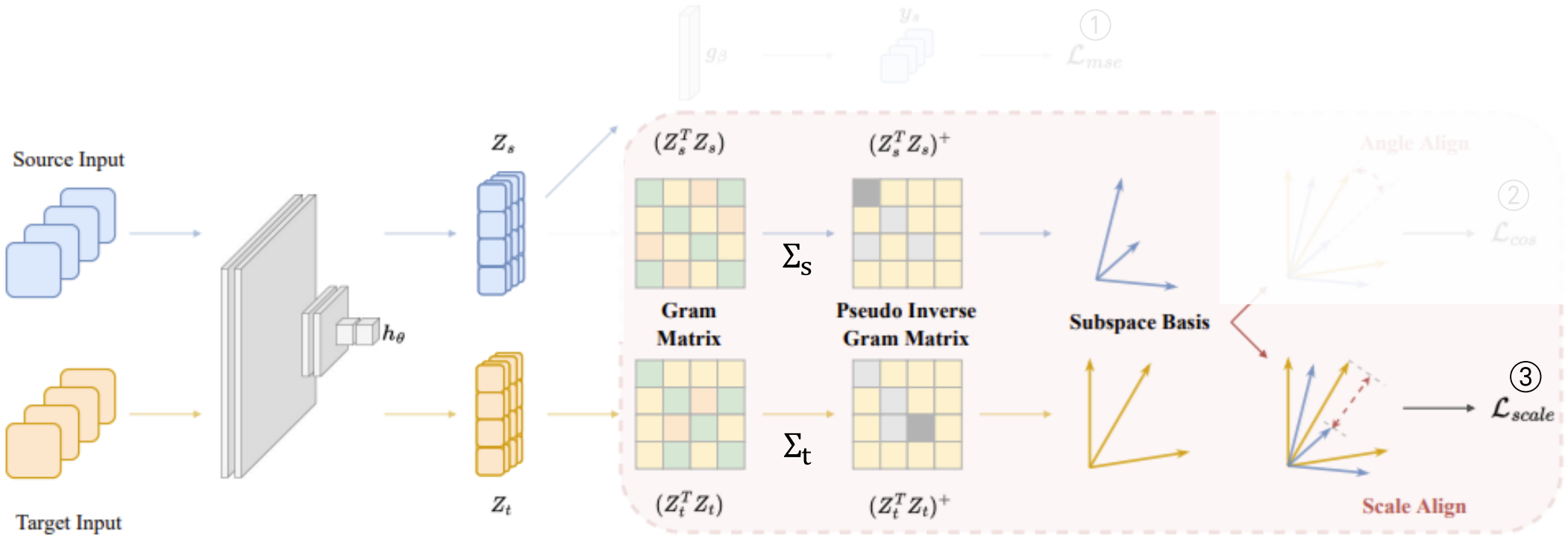
Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ 모델 훈련 단계 : ③ $L_{scale}(Z^S, Z^T)$

- Total loss = $L_{src} + \alpha_{cos} L_{cos}(Z^S, Z^T) + \gamma_{scale} L_{scale}(Z^S, Z^T)$

③ $L_{cos}(Z^S, Z^T)$= 도메인간 스케일 정렬

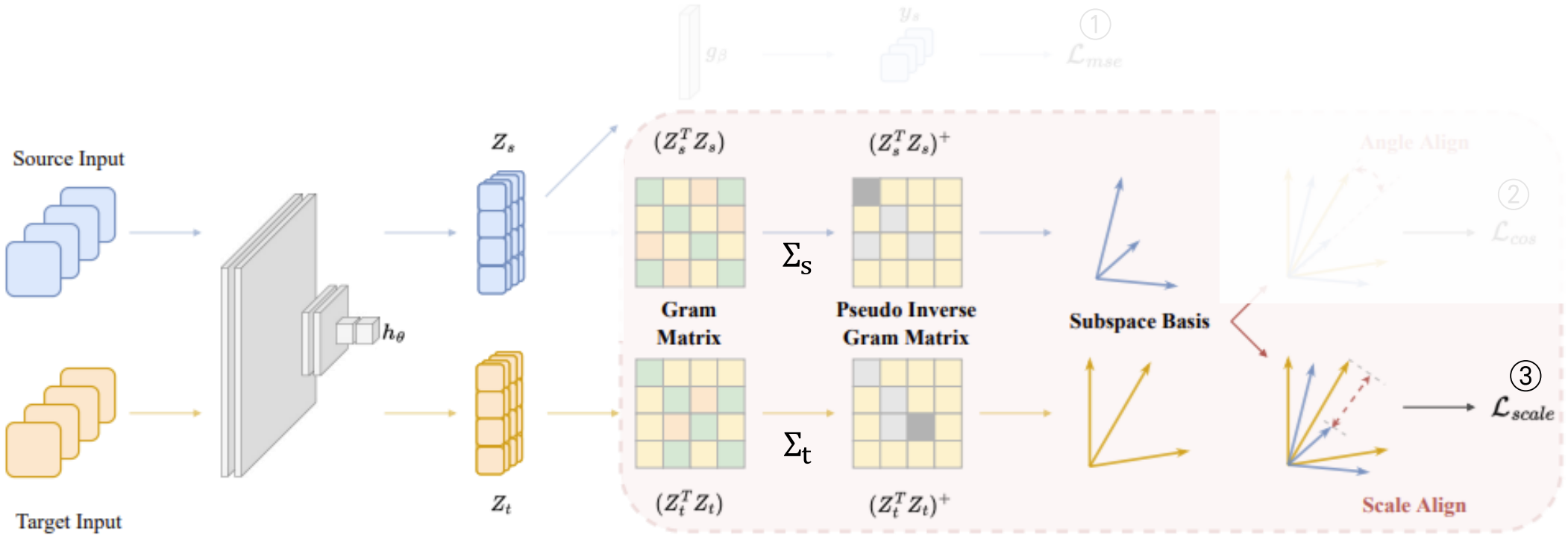


Methods

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ 모델 훈련 단계 : ③ $L_{scale}(Z^S, Z^T) = \|\lambda_{s,i=1,\dots,k}, -\lambda_{t,i=1,\dots,k}\|_2$

- ② 특이값 분해를 통해 구한 Σ_s, Σ_t 에서 각각 $\lambda_1, \dots, \lambda_k$만을 구한 뒤 $\|\cdot\|_2$는 L2 norm으로 계산
- $L_{scale}(Z^S, Z^T)$의 결론 : Source와 Target의 스케일을 일치시키는 것도 필수적이다!



Experiments

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

❖ 실험 결과

- 데이터 셋과 도메인 셋팅은 이전 방법론(RSD)과 동일
- 3개 데이터셋(dSprites, MPI3D, Biwi Kinect)에서 **첫 번째 방법론(RSD)보다 더 좋은 성능을 보임**

Method	C → N	C → S	N → C	N → S	S → C	S → N	Avg
Resnet-18 [27]	0.94	0.90	0.16	0.65	0.08	0.26	0.498
TCA [51]	0.94	0.87	0.19	0.66	0.10	0.23	0.498
MCD [57]	0.81	0.81	0.17	0.65	0.07	0.19	0.450
JDOT [13]	0.86	0.79	0.19	0.64	0.10	0.23	0.468
AFN [73]	1.00	0.96	0.16	0.62	0.08	0.32	0.523
DAN [44]	0.70	0.77	0.12	0.50	0.06	0.11	0.377
DANN [18]	0.47	0.46	0.16	0.65	**0.05**	0.10	0.315
RSD [10].	0.31	0.31	0.12	0.53	0.07	0.08	0.237
DARE-GRAM (ours)	**0.30**	**0.20**	**0.11**	**0.25**	**0.05**	**0.07**	**0.164**

Table 1. Comparisons with previous works on the dSprites regression tasks. All results are shown in sum of MAE with the ResNet-18.

Methods	RL → RC	RL → T	RC → RL	RC → T	T → RL	T → RC	Avg
Resnet-18 [27]	0.17	0.44	0.19	0.45	0.51	0.50	0.377
TCA [51]	0.17	0.42	0.19	0.42	0.50	0.50	0.373
MCD [57]	0.13	0.40	0.15	0.45	0.52	0.50	0.358
JDOT [13]	0.16	0.41	0.16	0.41	0.47	0.47	0.353
AFN [73]	0.18	0.45	0.20	0.46	0.53	0.53	0.390
DAN [44]	0.12	0.35	0.12	0.27	0.40	0.41	0.278
DANN [18]	0.09	0.24	0.11	0.41	0.48	0.37	0.283
RSD [10].	**0.09**	0.19	**0.08**	0.15	0.36	0.36	0.205
DARE-GRAM (ours)	**0.09**	**0.15**	0.10	**0.14**	**0.24**	**0.24**	**0.160**

Table 2. Comparisons with related works on the MPI3D regression tasks. All results are shown in sum of MAE with the ResNet-18.

Method	M → F	F → M	Avg
Resnet-18 [27]	0.29	0.38	0.335
TCA [51]	0.31	0.39	0.350
MCD [57]	0.31	0.37	0.340
JDOT [13]	0.29	0.39	0.340
AFN [73]	0.32	0.41	0.365
DAN [44]	0.28	0.37	0.325
DANN [18]	0.30	0.37	0.335
RSD [10]	0.26	0.30	0.280
DARE-GRAM (ours)	**0.23**	**0.29**	**0.260**

Table 3. Comparisons with previous works on Biwi Kinect.

Experiments

DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)

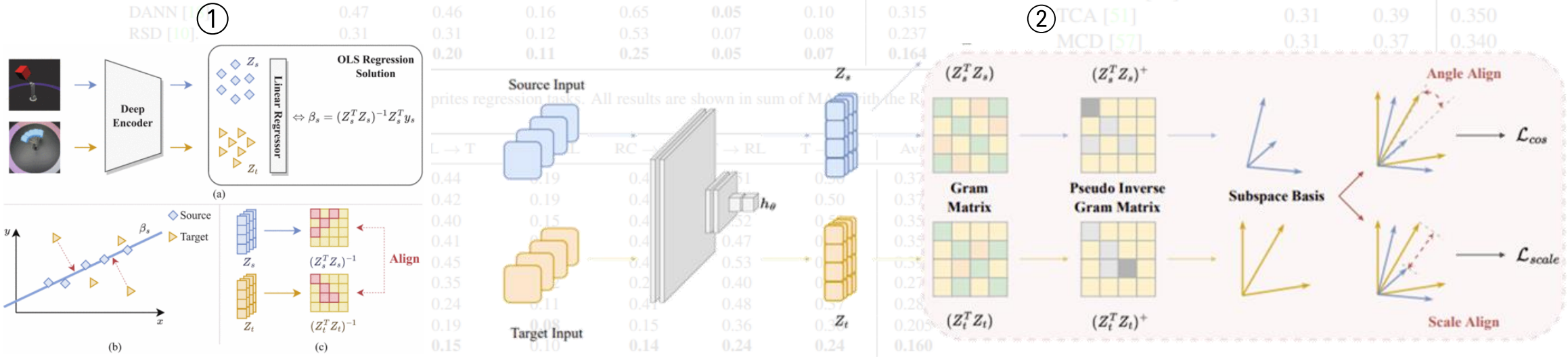
❖ 실험 결과

- 데이터 셋과 도메인 셋팅은 이전 방법론(RSD)과 동일
- 3개 데이터셋(dSprites, MPI3D, Biwi Kinect)에 대해 이전 방법론(RSD)보다 더 좋은 성능을 보임

방법론 정리

① 회귀 문제의 경우 Inverse gram matrix(역행렬)를 통해 모델을 학습해야 한다는 근거를 제시

② Pseudo-inverse gram matrix(유사 역행렬)을 모델 학습에 사용하여 도메인간 거리를 줄여 성능을 높임



Conclusion

Conclusion

❖ Conclusion

- 회귀에 적합한 도메인간 거리를 구하는 Unsupervised Domain Adaptation Regression 방법론에 대해 알아 봄
  - ① Representation Subspace Distance for Domain Adaptation Regression (ICML, 2021)
    - ➢ feature scaling에 민감한 것을 실험적으로 입증하고 SVD를 통해 새로운 도메인간 거리를 제시함
  - ② DARE-GRAM: Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (CVPR, 2023)
    - ➢ Inverse gram matrix(역행렬)를 통해 모델을 학습해야 한다는 근거를 제시했고, Pseudo-inverse gram matrix(유사 역행렬)을 모델 학습에 사용하여 좋은 성능을 보임

- 정리 : ①은 **feature scale 민감성**을 고려하여 **Σ(대각 행렬)을 사용하지 않고 U(직교 행렬)를 활용**, ②는 feature간 **상호작용을 고려해 Σ(대각 행렬)도 활용**하여 도메인 차이를 줄이려고 시도한 연구

Conclusion

❖ Conclusion

- 느낀 점

① 인스턴스 기반 feature 공간이 아닌, SVD(특이값 분해)를 활용하여 새로운 특징 공간(feature space)을 구성하고 이를 통한 도메인 정렬 방식을 시도한 논문들을 소개

② 그라스만 다양체(Grassmann manifold), 리만 기하학(Riemannian geometry) 등 모르는 부분에 대해 더 공부가 필요하며, 이론과 성능이 뒷받침 하는 알고리즘 제시 하기 위해 수학, 통계 등 여러 분야를 접하고 관심을 가지고 있어야 한다고 다시 한번 느낌

Thank you

References

[1] Goel, P., & Ganatra, A. (2023). Unsupervised domain adaptation for image classification and object detection using guided transfer learning approach and JS divergence. *Sensors*, 23(9), 4436.

[2] Dhaini, M., Berar, M., Honeine, P., & Van Exem, A. (2023). Unsupervised domain adaptation for regression using dictionary learning. *Knowledge-Based Systems*, 267, 110439.

[3] Chen, X., Wang, S., Wang, J., & Long, M. (2021, July). Representation Subspace Distance for Domain Adaptation Regression. In *ICML* (pp. 1749-1759).

[4] Nejjar, I., Wang, Q., & Fink, O. (2023). DARE-GRAM: Unsupervised domain adaptation regression by aligning inverse gram matrices. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11744-11754).