Test-Time Domain Adaptation

2025. 2. 7

Data Mining & Quality Analytics Lab.

황순혁

발표자 소개

❖ 황 순 혁 (Sunhyeok Hwang)

- 고려대학교 일반대학원 산업경영공학과 재학
- Data Mining & Quality Analytics Lab. (김성범 교수님)
- 석 · 박사 통합과정 7학기차 (2022. 03 ~)

❖ Research Interest

- Domain Adaptation/Generalization
- Test-Time Adaptation

❖ Contact

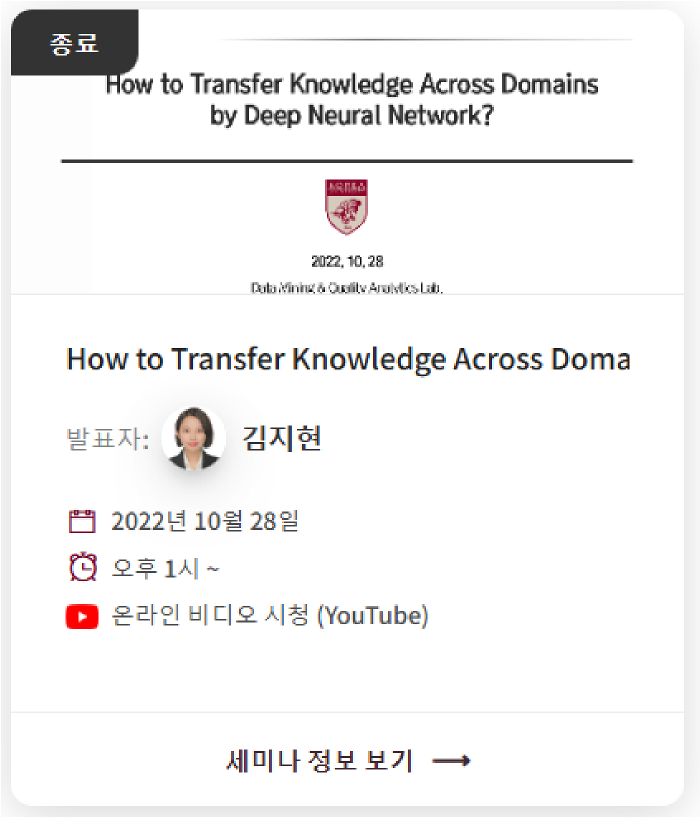
- shhwang1@korea.ac.kr

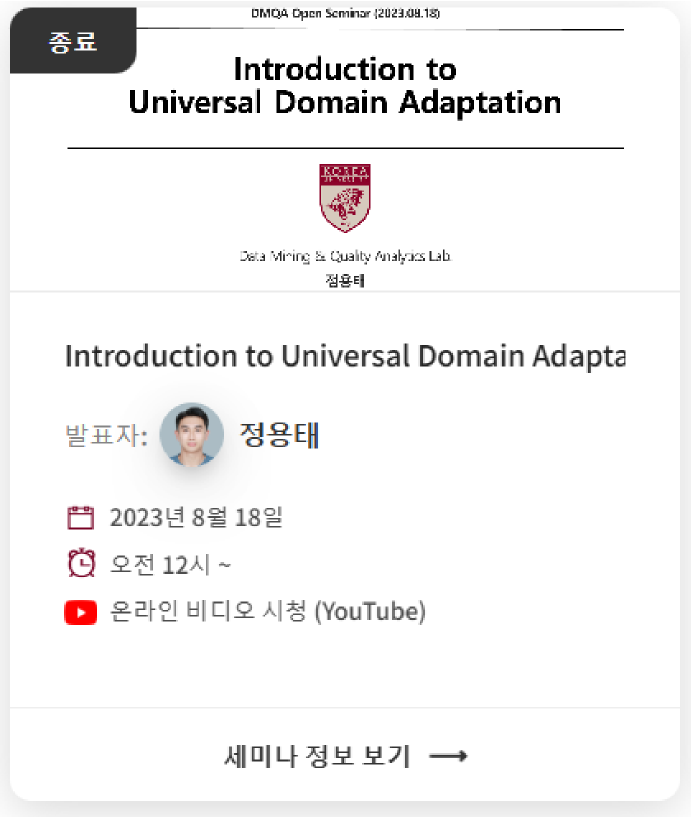
Contents

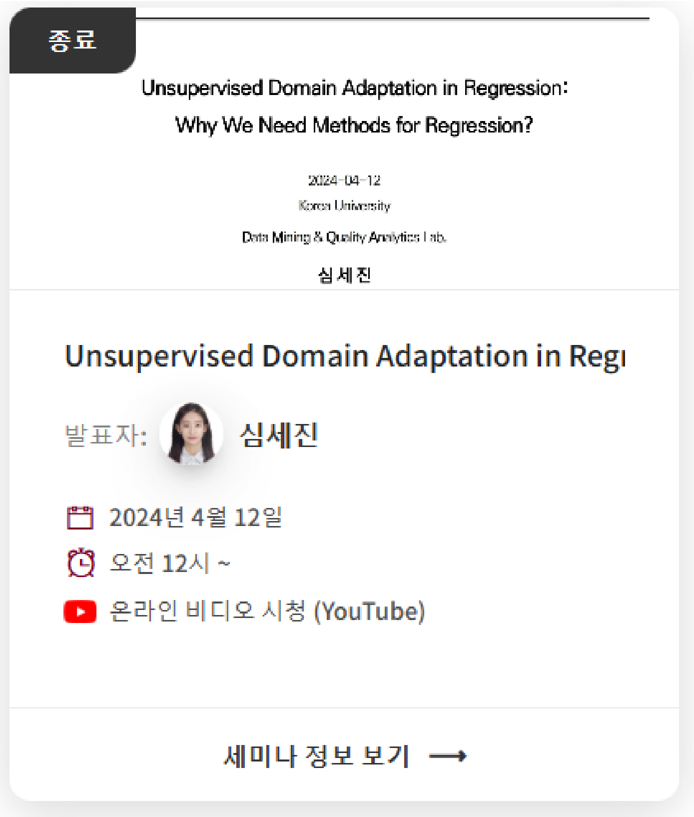
1. Introduction

Introduction

Background







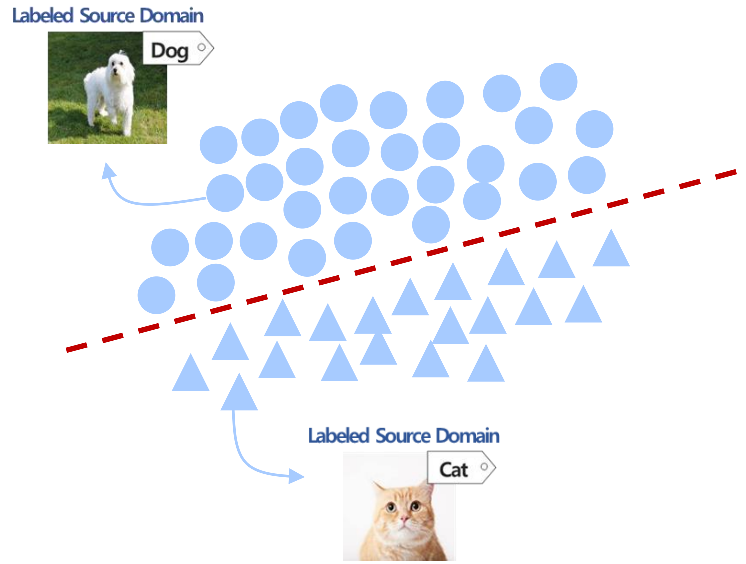
Introduction

Background

도메인 적응 방법론 (Domain Adaptation)

→ **서로 다른 도메인** 데이터셋간 **일반화 성능**을 높이는 방법론

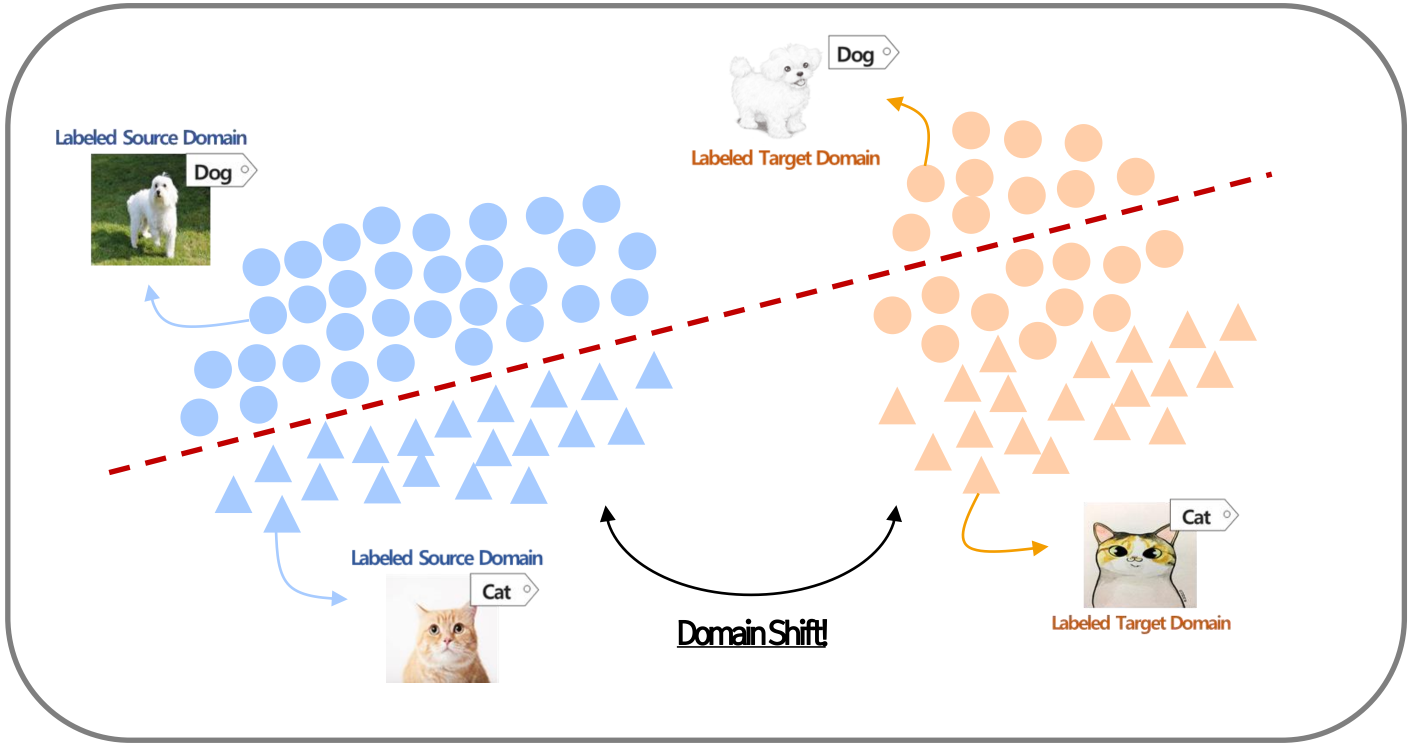
특징 : 유사하지만 분포가 서로 약간 다름



도메인 적응 방법론 (Domain Adaptation)

→ **서로 다른 도메인** 데이터셋간 **일반화 성능**을 높이는 방법론

특징 : 유사하지만 분포가 서로 약간 다름



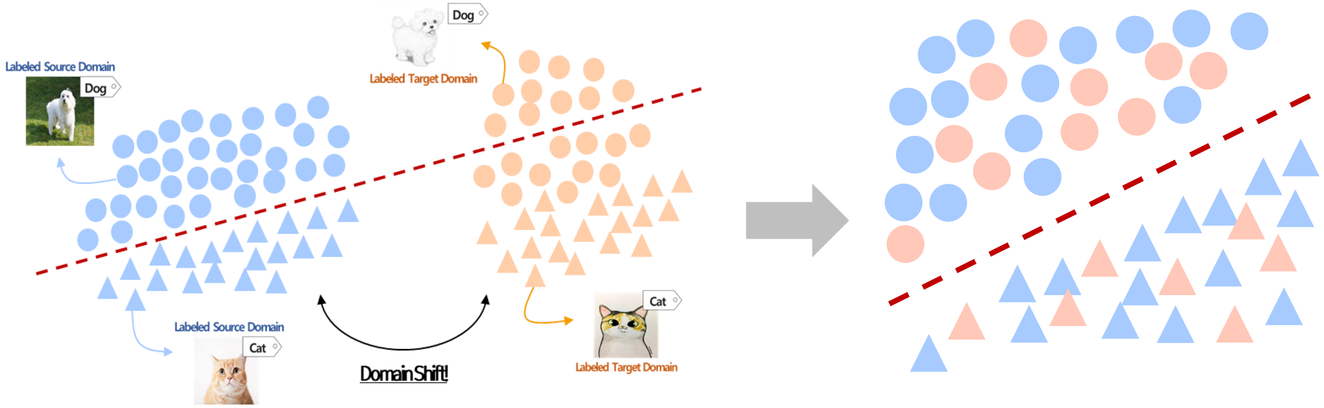
Introduction

Background

도메인 적응 방법론 (Domain Adaptation)

→ **서로 다른 도메인** 데이터셋간 **일반화 성능**을 높이는 방법론

특징 : 유사하지만 분포가 서로 약간 다름



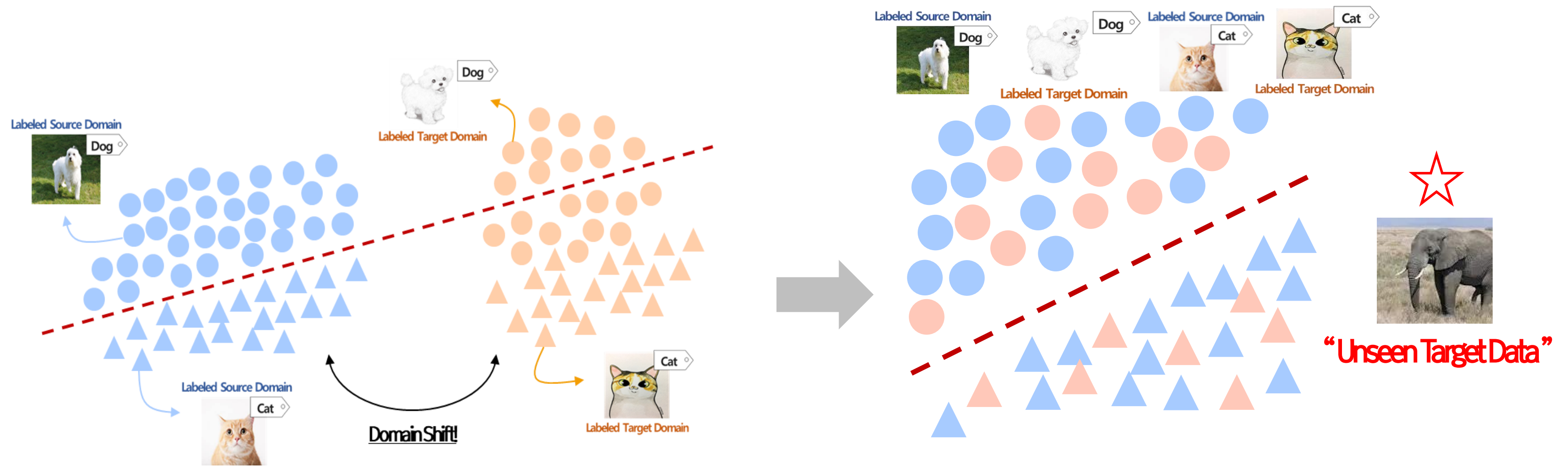
✓ 두 도메인을 잘 Alignment해서 모두 잘 분류되도록 만들자!

Introduction

Background

테스트 시점 적응 방법론 (Test-Time Adaptation)

→ **Source Domain의 도움 없이 테스트 시점의 새로운 Target Domain** 데이터를 잘 일반화



✓ 테스트 시점에서 학습된 모델에 **학습 단계에서 보지 못한 분포의 데이터**가 투입된다면?

Introduction

Background

테스트 시점 적응 방법론 (Test-Time Adaptation)

→ Source Domain의 도움 없이 테스트 시점의 새로운 Target Domain 데이터를 잘 일반화

ex. 제품 Recipe 정보(Categorical Variables)

	Sample No.	Step 1	Step 2	Step 3	· · ·	Step 100	Y
Train (Source Domain)	1	A	K	AA	· · ·	Z	0.59
	2	B	L	AB	· · ·	Z	0.66
	3	B	K	AA	· · ·	Y	0.58
	4	A	K	AC	· · ·	Y	0.61
	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·	· · ·
Test (Target Domain)	998	A	K	AC	· · ·	M	?
	999	C	L	AB	· · ·	Z	?
	1,000	B	O	AD	· · ·	M	?

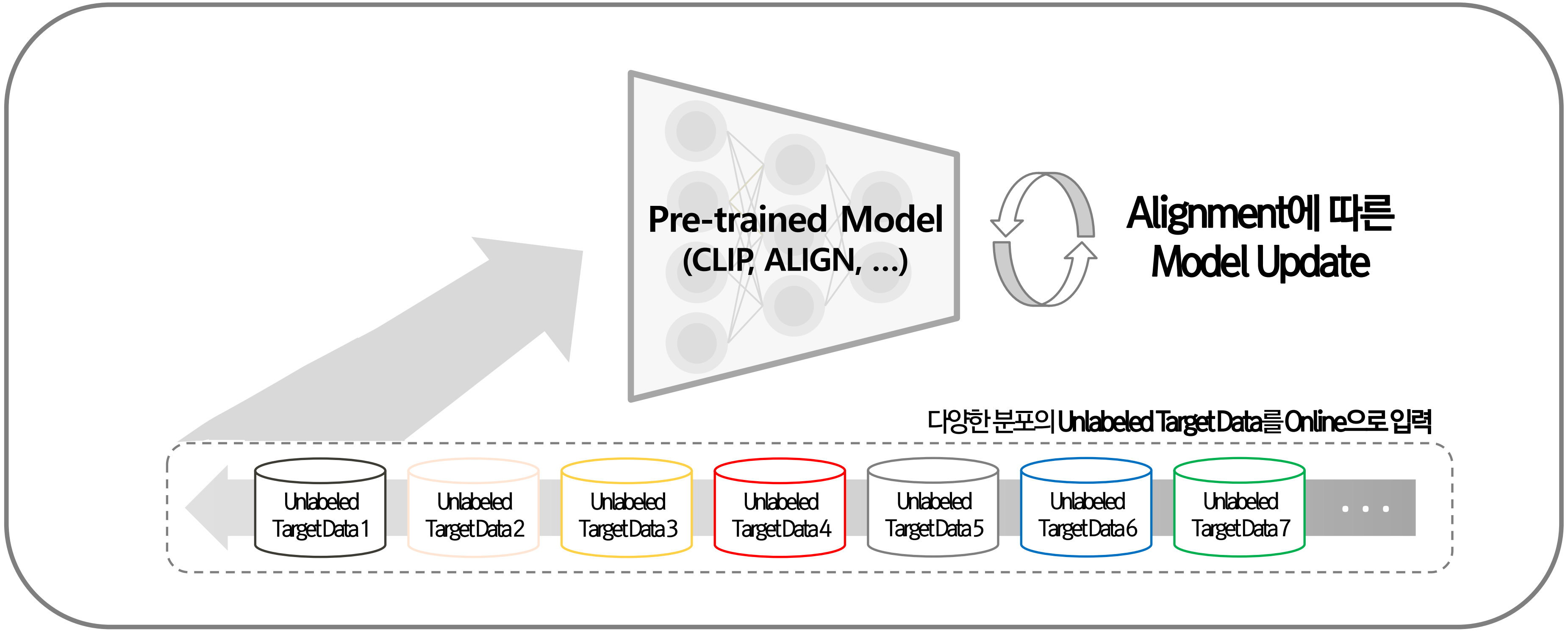
✓ 테스트 시점에서 학습된 모델에 학습 단계에서 보지 못한 분포의 데이터가 투입된다면?

Introduction

Background

테스트 시점 적응 방법론 (Test-Time Adaptation)

→ Source Domain의 도움 없이 테스트 시점의 새로운 Target Domain 데이터를 잘 일반화

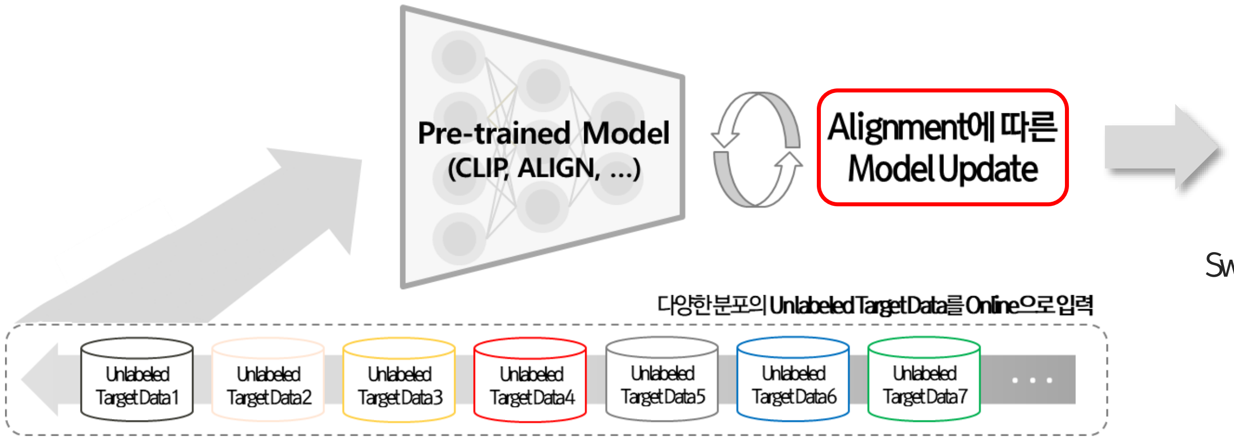


Introduction

Background

테스트 시점 적응 방법론 (Test-Time Adaptation)

→ Source Domain의 도움 없이 테스트 시점의 새로운 Target Domain 데이터를 잘 일반화



Self-Training Strategy

Improved Self-Training for Test-Time Adaptation
(CVPR 2024)

Self-Supervised Strategy

SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models
(CVPR 2024)

Batch Normalization Strategy

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization
(AAAI 2024)

Improved Self-Training for Test-Time Adaptation
(2024 CVPR)

Methods

Improved Self-Training for Test-Time Adaptation

❖ Improved Self-Training for Test-Time Adaptation [1]

- 2024년에 제안된 Self-training 기반 Test-time adaptation 방법론(CVPR, 2025년 2월 기준 3회 인용)
- 고도화된 Self-training 방법론을 통한 고품질의 Pseudo-labeling 및 Adaptation process 안정화

Improved Self-Training for Test-Time Adaptation

Jing Ma
Huazhong University of Science and Technology
jingma0011@gmail.com

Abstract

Test-time adaptation (TTA) is a technique to improve the performance of a pre-trained source model on a target distribution without using any labeled data. However, existing self-trained TTA methods often face the challenges of unreliable pseudo-labels and unstable model optimization. In this paper, we propose an Improved Self-Training (IST) approach, which addresses these challenges by enhancing the pseudo-label quality and stabilizing the adaptation process. Specifically, we use a simple augmentation strategy to generate multiple views of each test sample, and construct a graph structure to correct the pseudo-labels based on the similarity of the latent features. Moreover, we adopt a parameter moving average scheme to smooth the model updates and prevent catastrophic forgetting. Instead of using

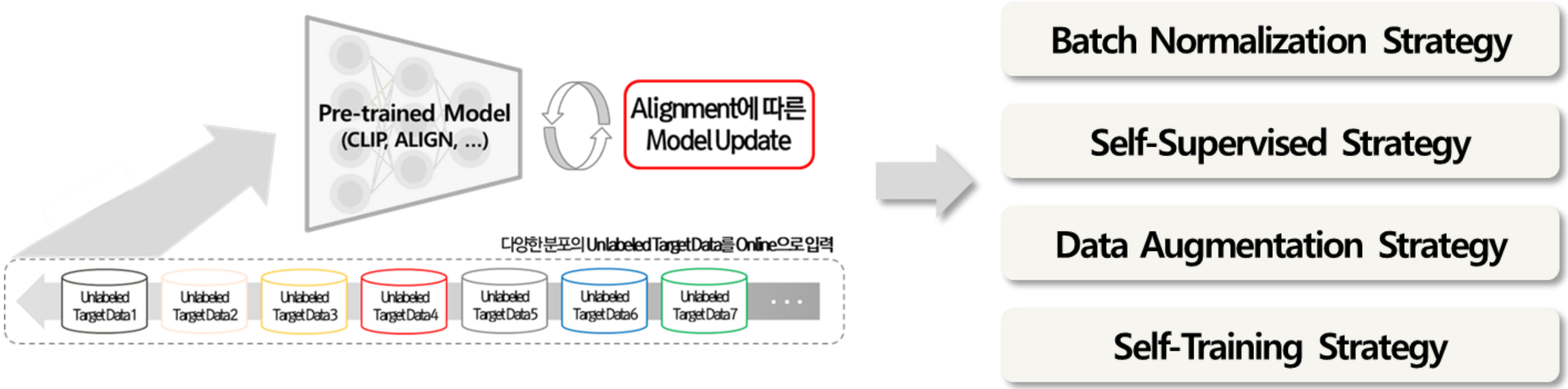
Figure 1. An illustration of test-time adaptation. The approach employs source models from a model zoo, where the source data is unknown. Source models adapt to target domains using one batch of test samples at a time. Self-training is a key component for updating models and providing more accurate online predictions.

Methods

Improved Self-Training for Test-Time Adaptation

❖ Improved Self-Training for Test-Time Adaptation [1]

- **Motivation** : 분포 변화가 존재하는 테스트 데이터에서 1) 신뢰성 높은 Pseudo-labeling 생성
  2) Adaptation 과정의 안정성 향상을 통한 모델 성능 개선
- Test-time Adaptation은 크게 **1) BN, 2) Self-supervised, 3) Data Augmentation** 방식이 있으나 각각의 한계점이 존재함

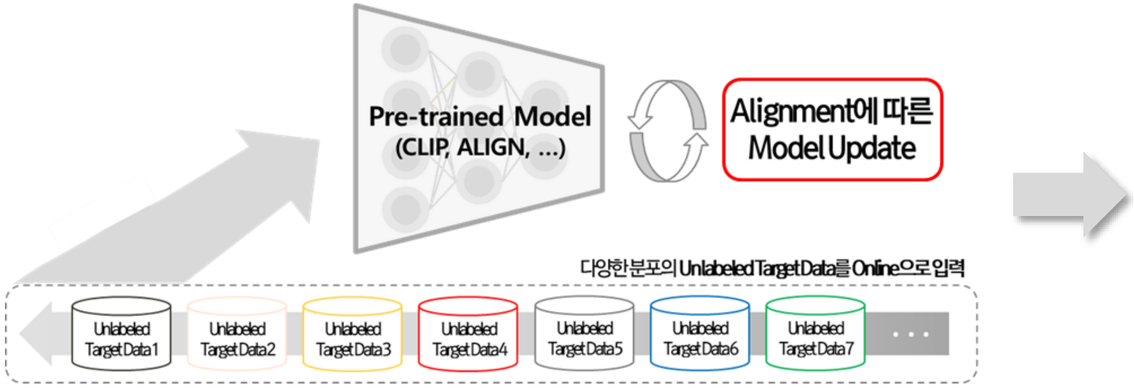


Methods

Improved Self-Training for Test-Time Adaptation

❖ Improved Self-Training for Test-Time Adaptation [1]

- **Motivation** : 분포 변화가 존재하는 테스트 데이터에서 1) 신뢰성 높은 Pseudo-labeling 생성
  2) Adaptation 과정의 안정성 향상을 통한 모델 성능 개선
- Test-time Adaptation은 크게 1) BN, 2) Self-supervised, 3) Data Augmentation 방식이 있으나 각각의 한계점이 존재함



Batch Normalization Strategy

BN 통계값을 업데이트하여 분포 변화 완화 시도
특정 분포 이동 범위에서만 효과적
일반화 성능이 제한됨

Self-Supervised Strategy

Data Augmentation Strategy

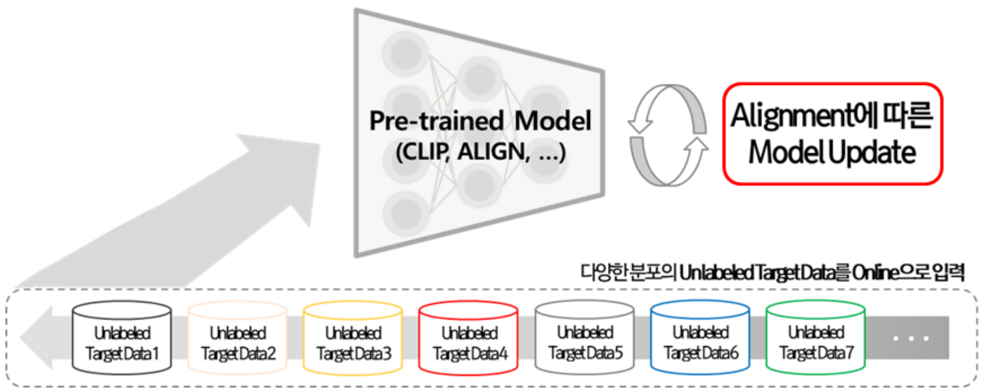
Self-Training Strategy

Methods

Improved Self-Training for Test-Time Adaptation

❖ Improved Self-Training for Test-Time Adaptation [1]

- Motivation : 분포 변화가 존재하는 테스트 데이터에서 1) 신뢰성 높은 Pseudo-labeling 생성
  2) Adaptation 과정의 안정성 향상을 통한 모델 성능 개선
- Test-time Adaptation은 크게 1) BN, 2) Self-supervised, 3) Data Augmentation 방식이 있으나 각각의 한계점이 존재함



Batch Normalization Strategy

BN 통계값을 업데이트하여 분포 변화 완화 시도
특정 분포 이동 범위에서만 효과적
일반화 성능이 제한됨

Self-Supervised Strategy

부적절한 Pretext-Task는 오히려
분포 변화 완화에 부정적인 영향을 미침

Data Augmentation Strategy

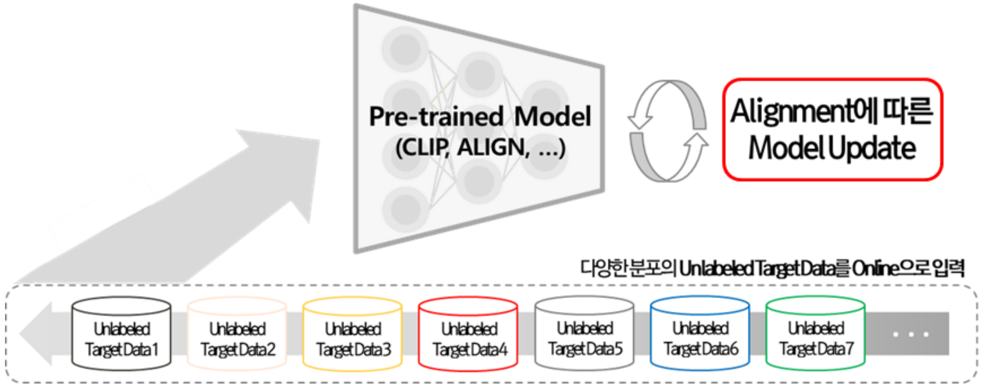
Self-Training Strategy

Methods

Improved Self-Training for Test-Time Adaptation

❖ Improved Self-Training for Test-Time Adaptation [1]

- **Motivation** : 분포 변화가 존재하는 테스트 데이터에서 1) 신뢰성 높은 Pseudo-labeling 생성
  2) Adaptation 과정의 안정성 향상을 통한 모델 성능 개선
- Test-time Adaptation은 크게 1) BN, 2) Self-supervised, 3) Data Augmentation 방식이 있으나 각각의 한계점이 존재함



Batch Normalization Strategy

BN 통계값을 업데이트하여 분포 변화 완화 시도
특정 분포 이동 범위에서만 효과적
일반화 성능이 제한됨

Self-Supervised Strategy

부적절한 Pretext-Task는 오히려
분포 변화 완화에 부정적인 영향을 미침

Data Augmentation Strategy

밝기, 대비, 채도 등 복잡한 데이터 증강은
클래스 간 구분에 중요한 속성을 손상시킴

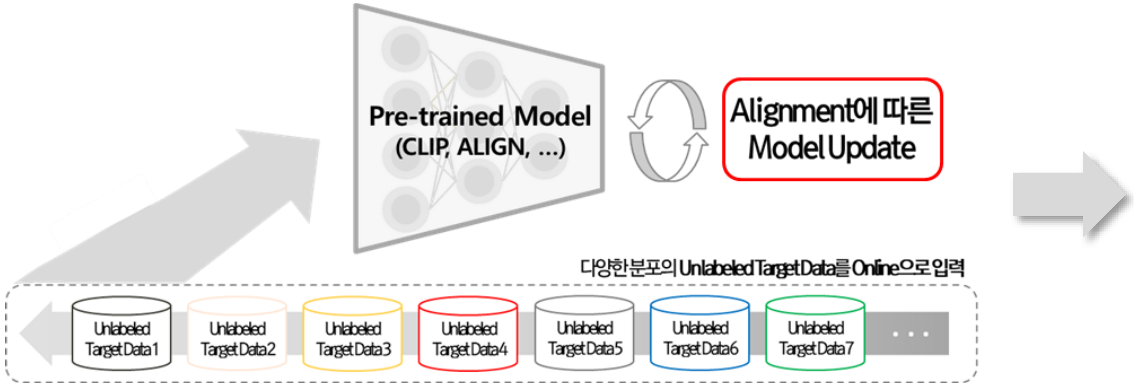
Self-Training Strategy

Methods

Improved Self-Training for Test-Time Adaptation

❖ Improved Self-Training for Test-Time Adaptation [1]

- **Motivation** : 분포 변화가 존재하는 테스트 데이터에서 1) 신뢰성 높은 Pseudo-labeling 생성
  2) Adaptation 과정의 안정성 향상을 통한 모델 성능 개선
- Test-time Adaptation은 크게 1) BN, 2) Self-supervised, 3) Data Augmentation 방식이 있으나 각각의 한계점이 존재함



Batch Normalization Strategy

BN 통계값을 업데이트하여 분포 변화 완화 시도
특정 분포 이동 범위에서만 효과적
일반화 성능이 제한됨

Self-Supervised Strategy

부적절한 Pretext-Task는 오히려
분포 변화 완화에 부정적인 영향을 미침

Data Augmentation Strategy

밝기, 대비, 채도 등 복잡한 데이터 증강은
클래스 간 구분에 중요한 속성을 손상시킴

Self-Training Strategy

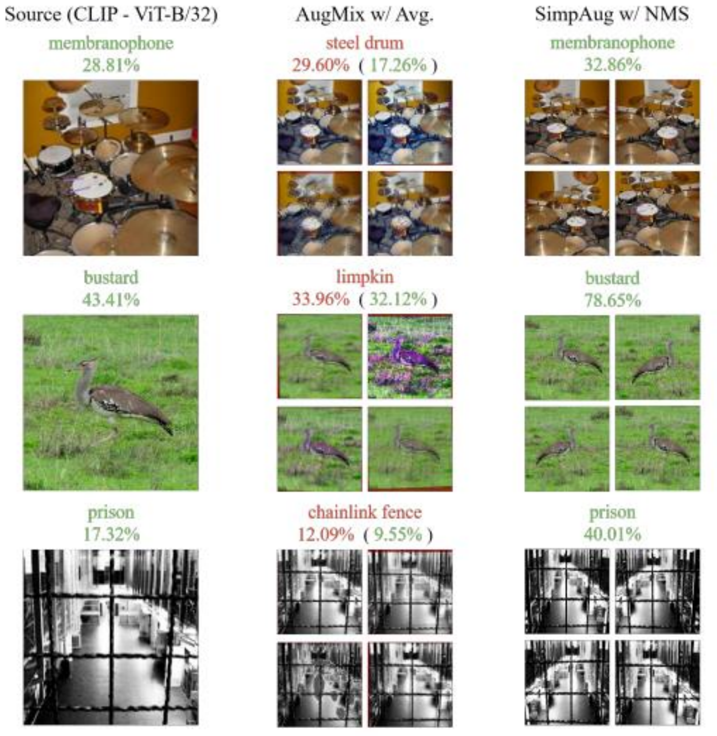
고품질의 Pseudo-labeling 기반 강력한 일반화 성능
낮은 모델 복잡도 및 빠른 데이터 처리 속도

Methods

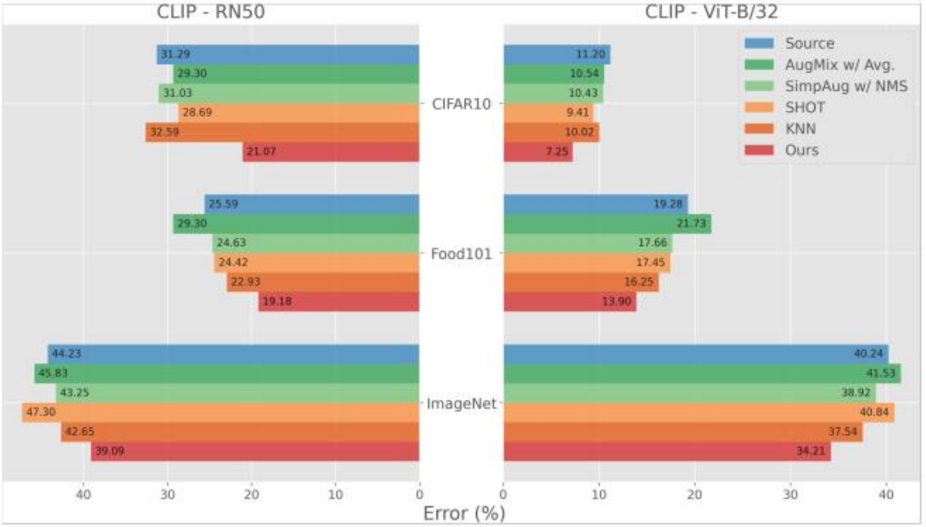
Improved Self-Training for Test-Time Adaptation

❖ ① Simple Augmentation Strategy

- 기존 Data Augmentation Strategy 기반의 Test-time Adaptation은 중요 속성을 손상시킬 우려가 있는 복잡한 증강 기법을 적용
- 본 논문은 Pre-trained Model인 CLIP에 보다 간단한 증강 기법을 적용했을 때 유의미한 성능 향상을 보였음을 실험적으로 증명함



(a)



(b)

Usage Time (seconds)	SHOT	KNN	Ours
CLIP - RN50	7.501	1.528	3.911
CLIP - ViT-B/32	3.239	1.239	3.601

(c)

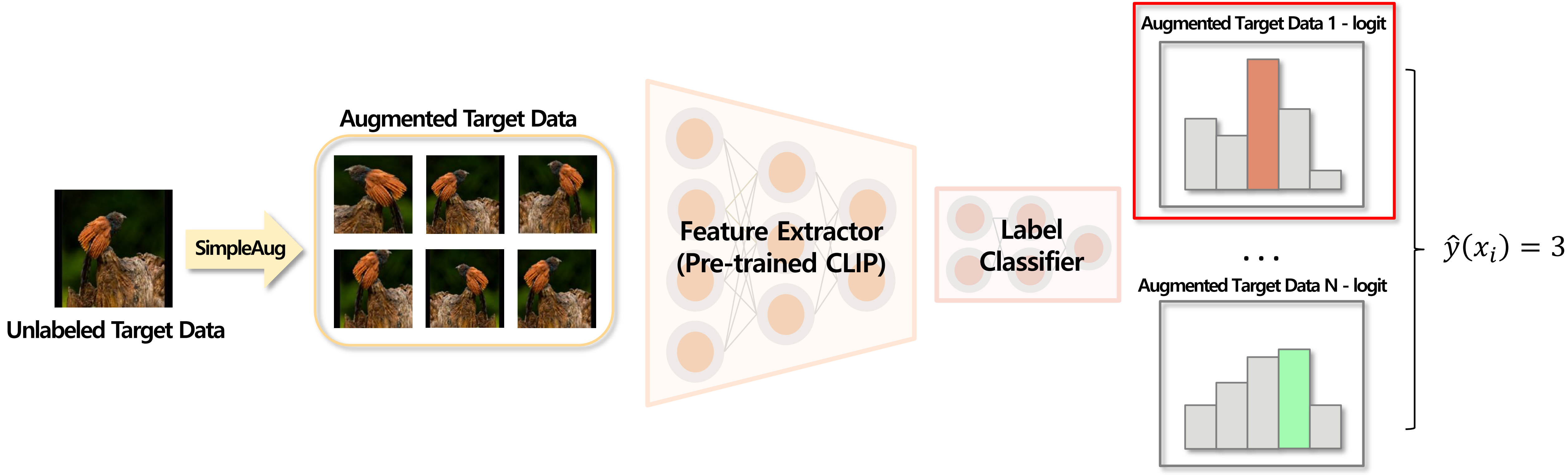
Data Mining
Quality Analytics

Methods

Improved Self-Training for Test-Time Adaptation

❖ ① Simple Augmentation Strategy

- Crops, Scales, Horizontal Flips의 비교적 간단한 데이터 증강 기법을 적용
- 데이터 증강이 적용된 다양한 Unlabeled Target Data의 Class prediction logit값 중 max값을 갖는 class로 pseudo-label 설정



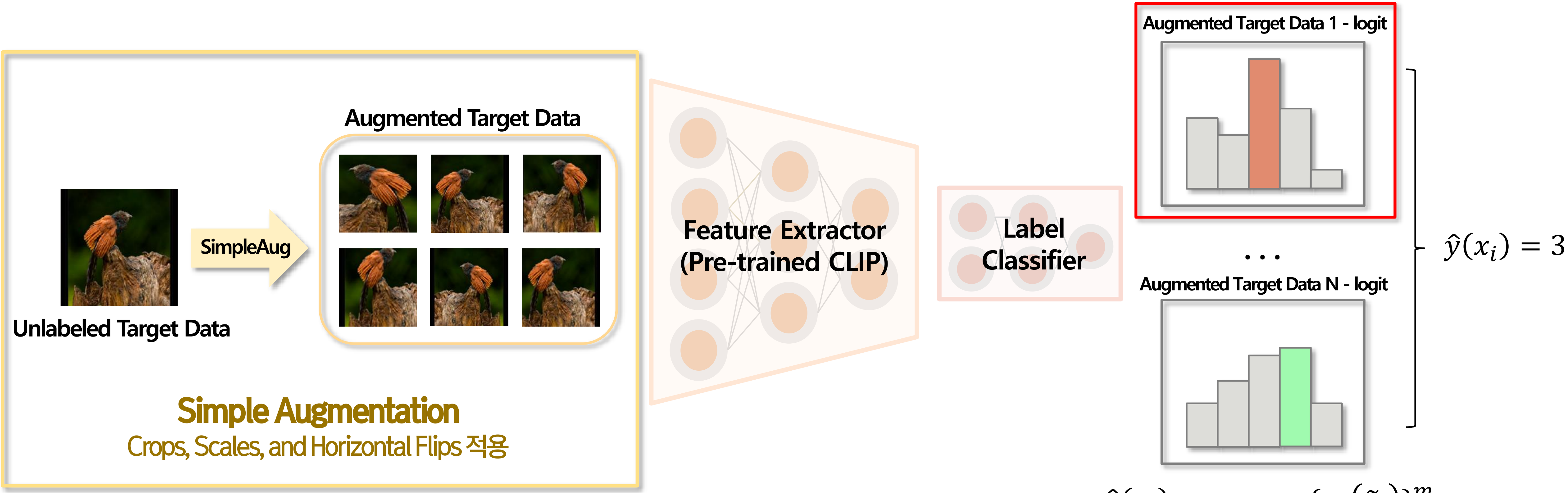
$$\hat{y}(x_i) = \arg\max_c \{p_c(\tilde{x}_j)\}_{j=1}^{m}$$

Methods

Improved Self-Training for Test-Time Adaptation

❖ ① Simple Augmentation Strategy

- Crops, Scales, Horizontal Flips의 비교적 간단한 데이터 증강 기법을 적용
- 데이터 증강이 적용된 다양한 Unlabeled Target Data의 Class prediction logit값 중 max값을 갖는 class로 pseudo-label 설정



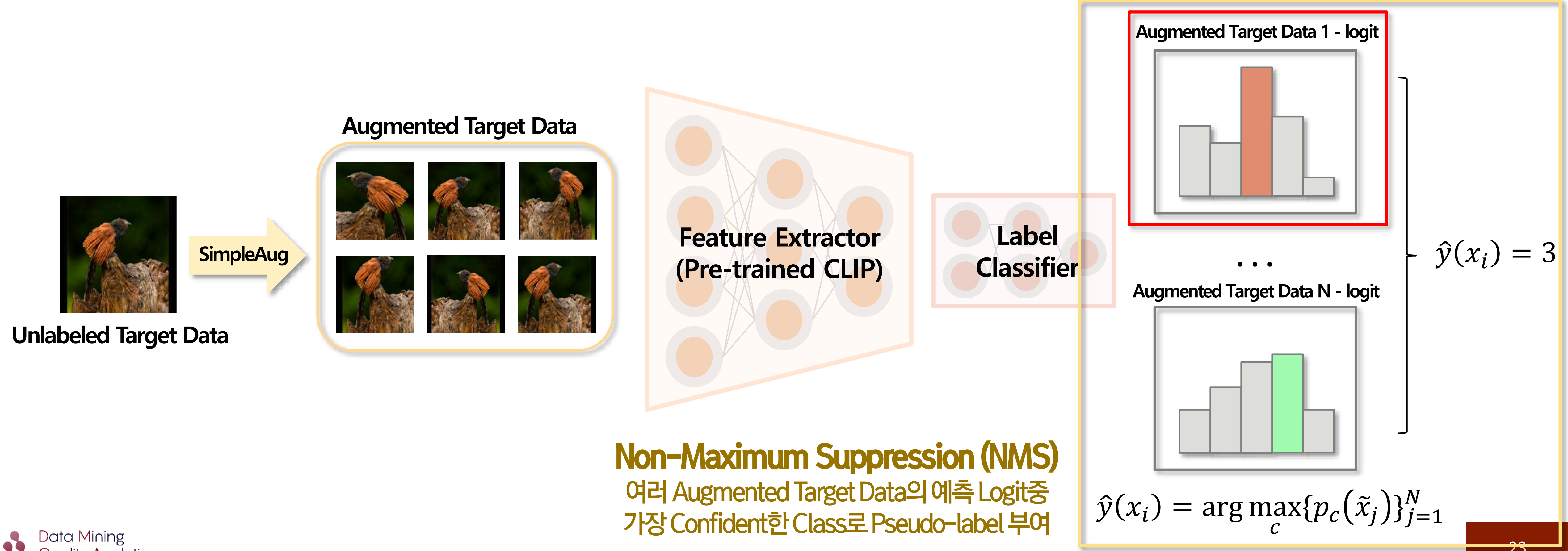
$$\hat{y}(x_i) = \arg\max_{c}\{p_c(\tilde{x}_j)\}_{j=1}^{m}$$

Methods

Improved Self-Training for Test-Time Adaptation

❖ ① Simple Augmentation Strategy

- Crops, Scales, Horizontal Flips의 비교적 간단한 데이터 증강 기법을 적용
- 데이터 증강이 적용된 다양한 Unlabeled Target Data의 Class prediction logit값 중 max값을 갖는 class로 pseudo-label 설정

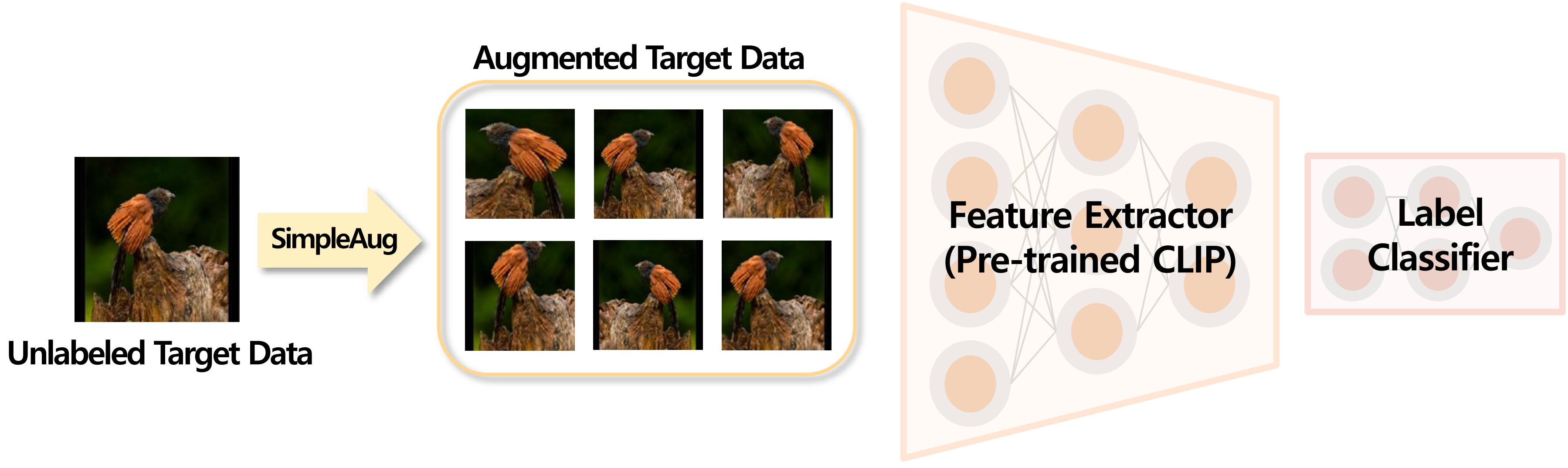


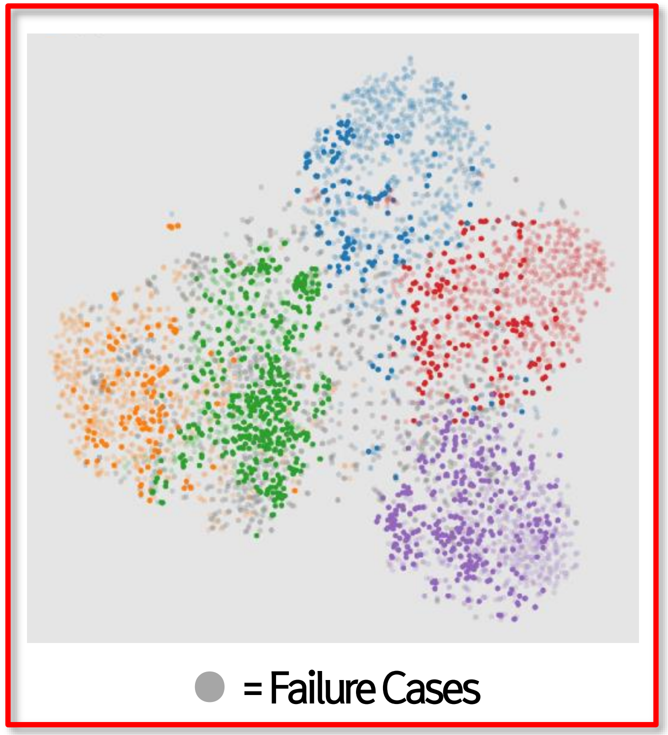
Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm

- 기존 사전학습된 CLIP 모델에 ① Simple Augmentation Strategy만 적용할 경우 오분류 케이스가 다수 확인
- 이는 단순 증강 기법에 의존한 Pseudo-labeling 적용은 한계가 있음을 보여줌
  - ➢ 이러한 저품질 Pseudo-label 문제를 ② Pseudo-Label Correction Algorithm으로 해결



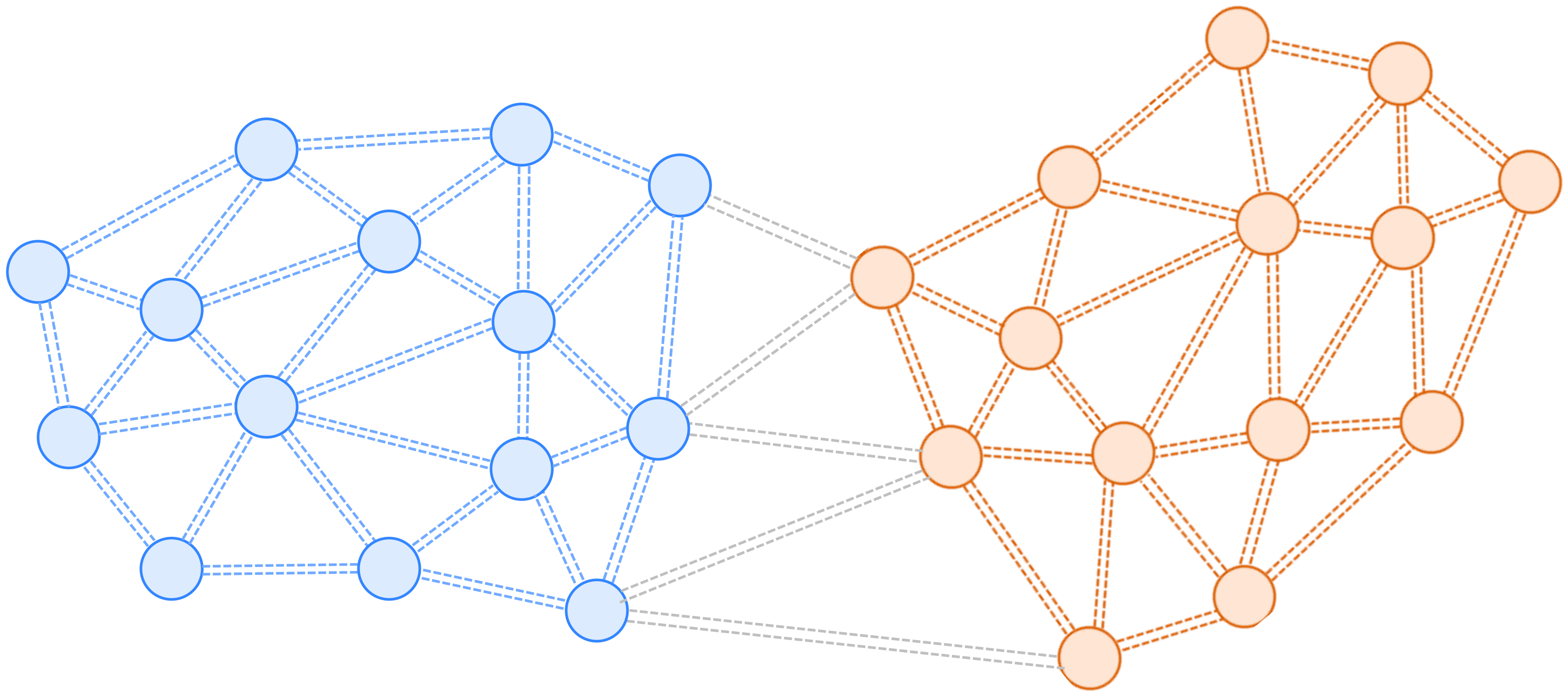


Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 1) Graph Structure Definition

- Target Domain 데이터의 잠재 공간 상 Feature Vector를 그래프 구조로 정의
  - ➢ Vertices(노드) : 특징 벡터, Edge(엣지) : 노드 간의 거리(가중치)

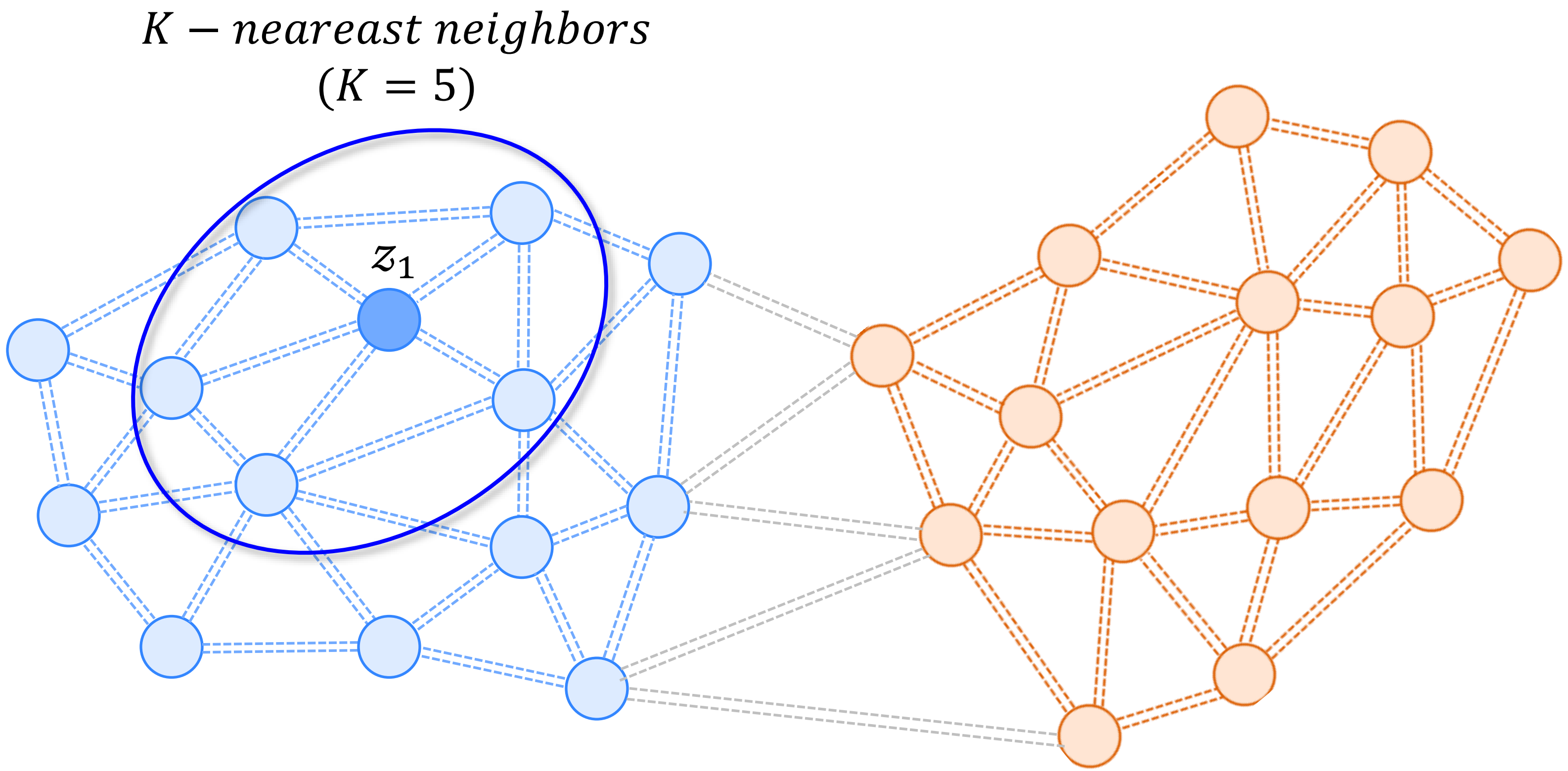


Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 1) Graph Structure Definition

- 특정 노드(벡터)를 기준으로 K-neareast neighbors 적용을 통한 인접 노드 산정



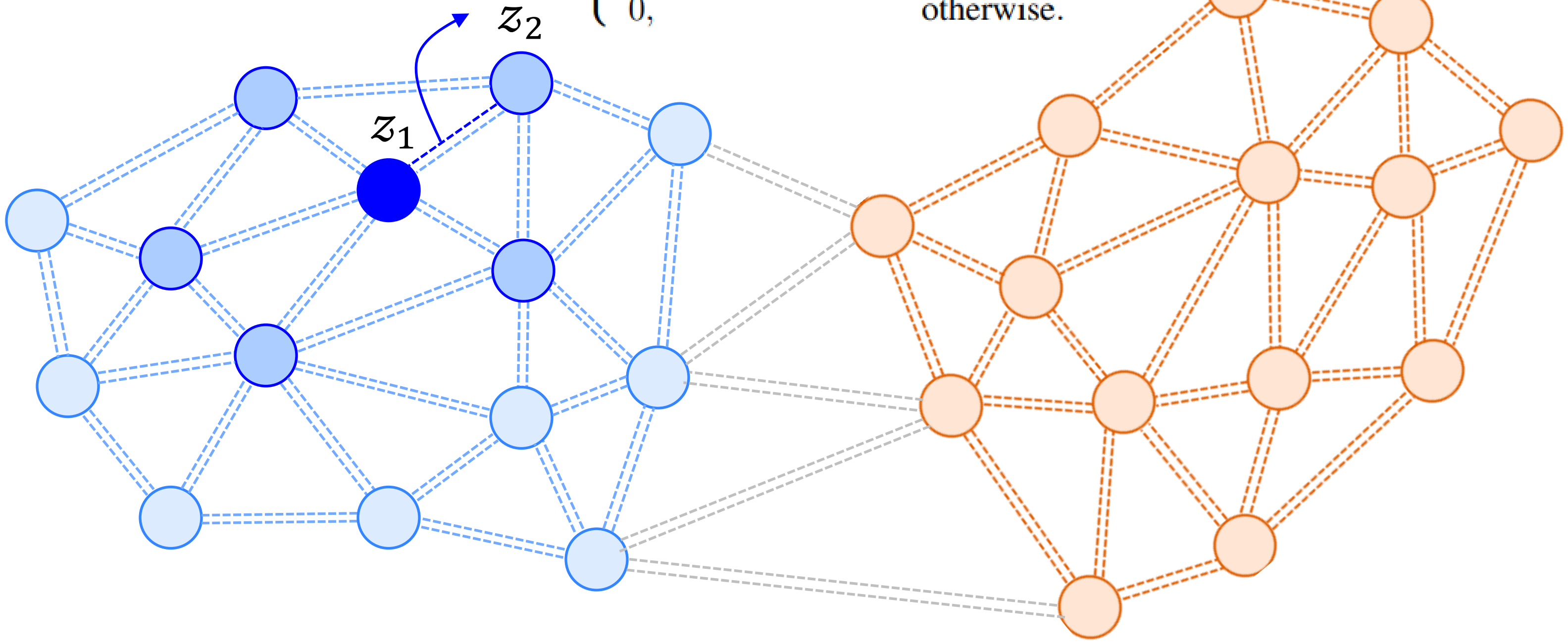
Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 1) Graph Structure Definition

- 노드 간 거리를 반영하는 가중치 w_{ij} 공식을 적용하여 모든 노드 간 가중치 산정

$$w_{ij} = \begin{cases} 1, & i = j, \\ 1 - \frac{\|z_i - z_j\|_2}{\max_j \|z_i - z_j\|_2}, & z_j \in \mathcal{N}_K(z_i), \\ 0, & \text{otherwise.} \end{cases}$$



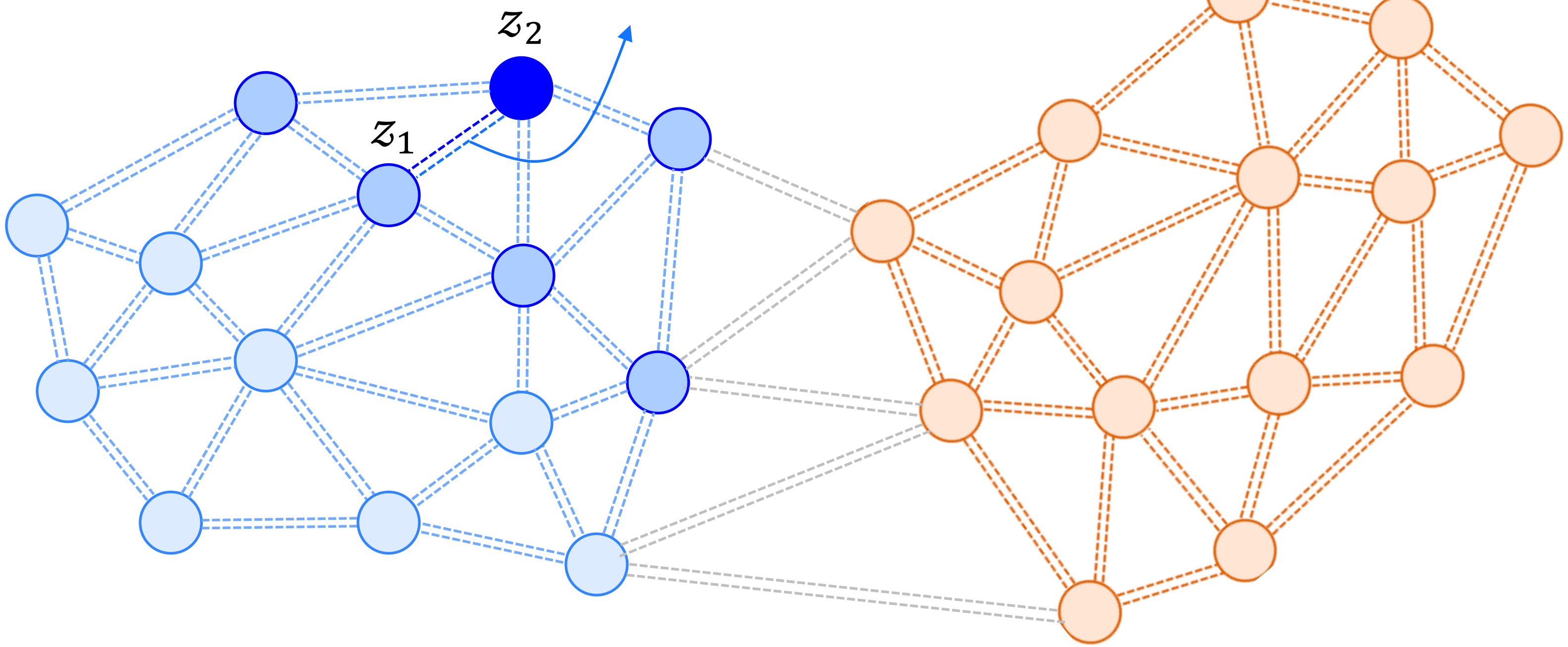
Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 1) Graph Structure Definition

- 노드 간 거리를 반영하는 가중치 w_{ij} 공식을 적용하여 모든 노드 간 가중치 산정

$$w_{ij} = \begin{cases} 1, & i = j, \\ 1 - \frac{\|z_i - z_j\|_2}{\max_j \|z_i - z_j\|_2}, & z_j \in \mathcal{N}_K(z_i), \\ 0, & \text{otherwise}. \end{cases}$$



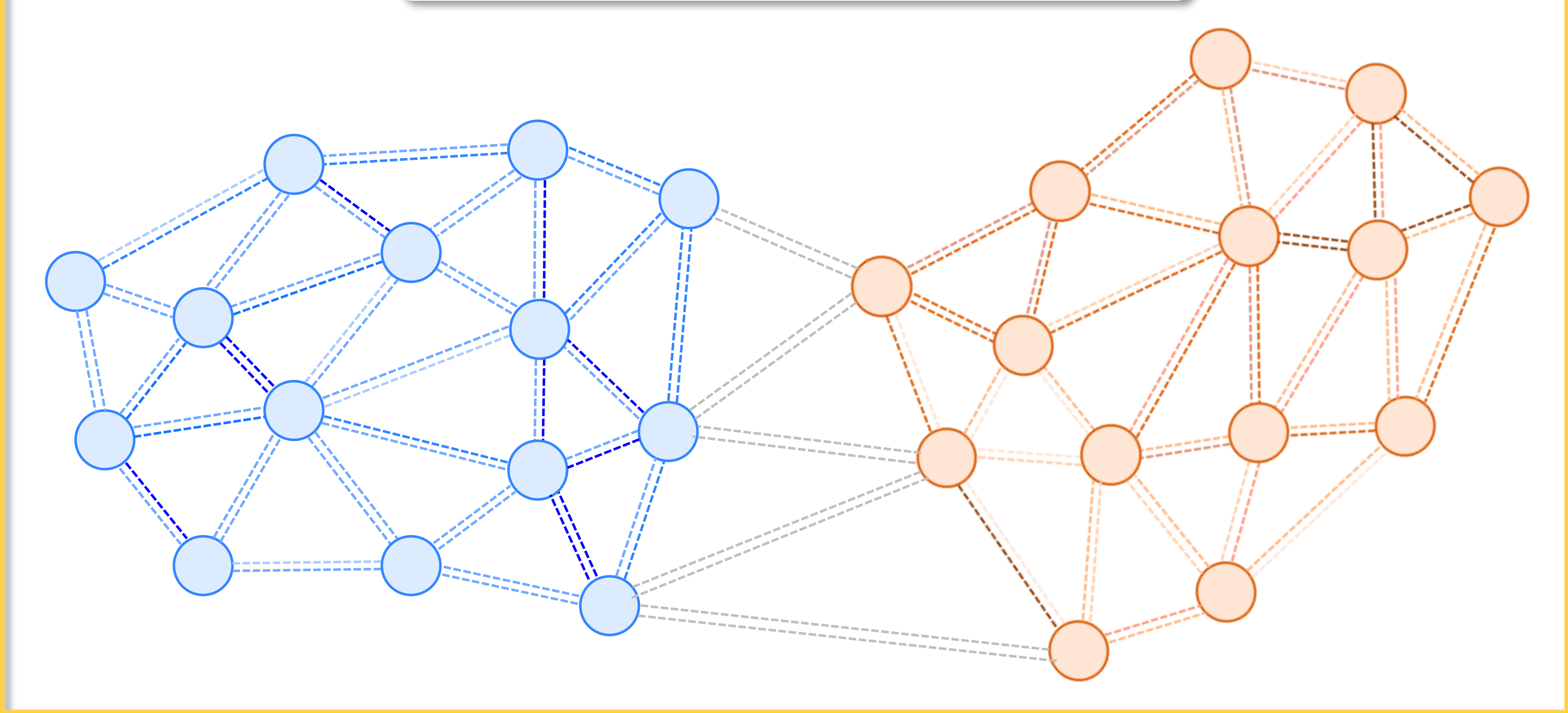
Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 1) Graph Structure Definition

- 노드 간 거리를 반영하는 가중치 w_{ij} 공식을 적용하여 모든 노드 간 가중치 산정

Feature Vectors Graph Structure



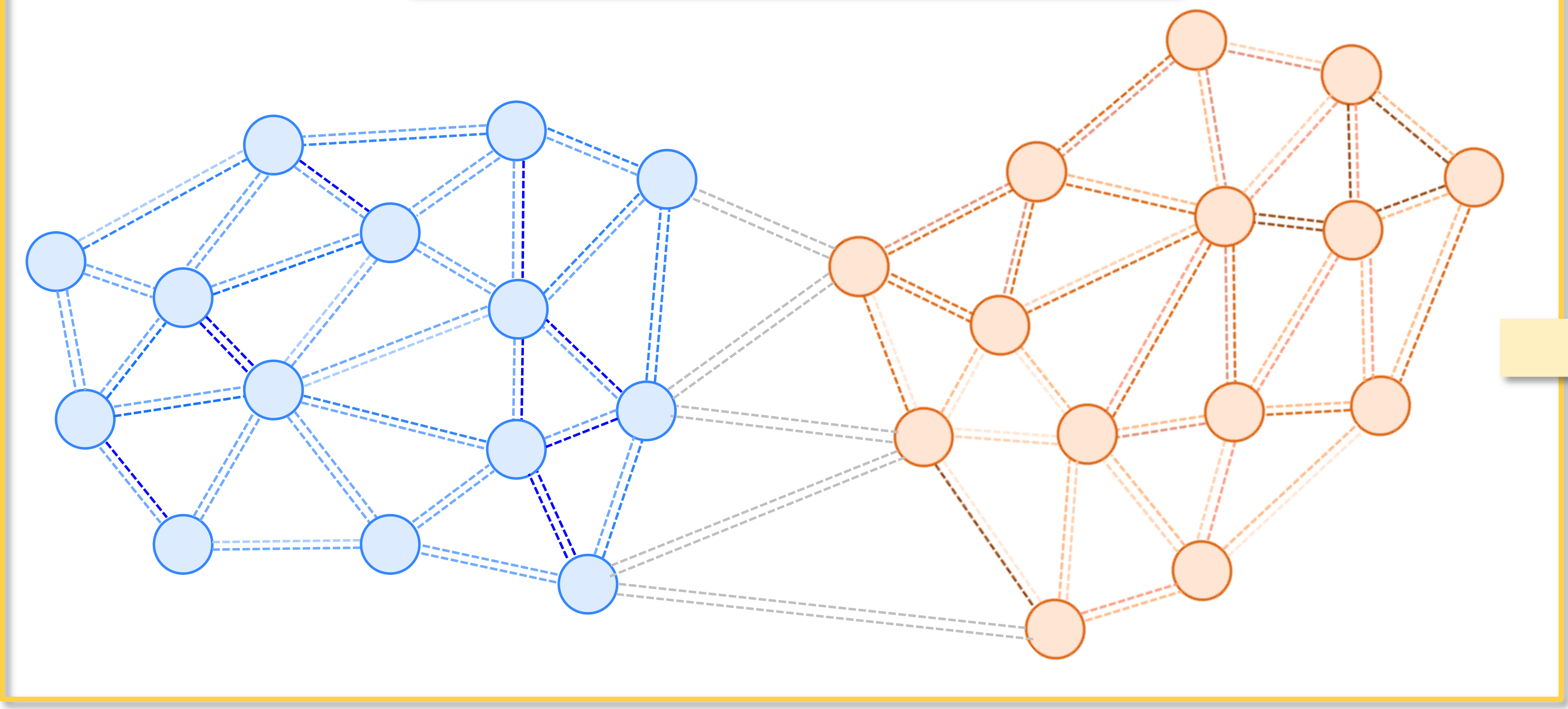
Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 1) Graph Structure Definition

- 노드 간 거리를 반영하는 가중치 w_{ij} 공식을 적용하여 모든 노드 간 가중치 산정

Feature Vectors Graph Structure



대칭 가중치 행렬 W

$$W = (w_{ij})_{n\times n} + (w_{ji})_{n\times n}$$

정규화 그래프 행렬 S

$$D_{ii} = \sum_j W_{ij}$$

$$\boxed{S = D^{-1/2} W D^{-1/2}}$$

Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 2) Label Propagation Process

- 초기 레이블 행렬을 정의하고 이를 기점으로 레이블 전파 과정을 거쳐 고품질의 Pseudo-label 형성

$$Y(0) = (y_{ij})_{n \times m}, \qquad where\ y_{ij} = p_j(x_i)$$

$$Y(t+1) = \alpha S Y(t) + (1-\alpha) Y(0)$$

$$Y(T) = \alpha^T S^T Y(0) + (1-\alpha) \sum_{t=0}^{T-1} \alpha^t S^t Y(0)$$

$$Y^* = \lim_{T \to \infty} Y(T) = (1-\alpha)\,(I - \alpha S)^{-1} Y(0)$$

Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 2) Label Propagation Process

- 초기 레이블 행렬을 정의하고 이를 기점으로 레이블 전파 과정을 거쳐 고품질의 Pseudo-label 형성

$$\boxed{Y(0) = (y_{ij})_{n \times m}, \qquad where\ y_{ij} = p_j(x_i)}$$

<u>초기 Label $\boldsymbol{Y(0)}$</u>

Sample x_i가 Class j에 속할 Predictive probability

$$Y(t+1) = \alpha S Y(t) + (1-\alpha) Y(0)$$

$$Y(T) = \alpha^T S^T Y(0) + (1-\alpha) \sum_{t=0}^{T-1} \alpha^t S^t Y(0)$$

$$Y^* = \lim_{T \to \infty} Y(T) = (1-\alpha)\,(I - \alpha S)^{-1} Y(0)$$

Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 2) Label Propagation Process

- 초기 레이블 행렬을 정의하고 이를 기점으로 레이블 전파 과정을 거쳐 고품질의 Pseudo-label 형성

$$Y(0) = (y_{ij})_{n\times m}, \qquad where\ y_{ij} = p_j(x_i)$$

$$\boxed{Y(t+1) = \alpha S Y(t) + (1-\alpha) Y(0)}$$

$$Y(T) = \alpha^T S^T Y(0) + (1-\alpha) \sum_{t=0}^{T-1} \alpha^t S^t Y(0)$$

$$Y^* = \lim_{T\to\infty} Y(T) = (1-\alpha)\,(I-\alpha S)^{-1} Y(0)$$

Label Update

초기 Label $\boldsymbol{Y}(\mathbf{0})$와 샘플간 **정규화 행렬 그래프** $\boldsymbol{S}$의 정보가 반영된

현재 시점 $\boldsymbol{Y}(\boldsymbol{t})$의 정보로 Label을 Update

Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 2) Label Propagation Process

- 초기 레이블 행렬을 정의하고 이를 기점으로 레이블 전파 과정을 거쳐 고품질의 Pseudo-label 형성

$$Y(0) = (y_{ij})_{n\times m}, \qquad where\ y_{ij} = p_j(x_i)$$

$$\boxed{\begin{aligned} Y(t+1) &= \alpha S Y(t) + (1-\alpha) Y(0) \\ Y(T) &= \alpha^T S^T Y(0) + (1-\alpha) \sum_{t=0}^{T-1} \alpha^t S^t Y(0) \end{aligned}}$$

$$Y^* = \lim_{T\to\infty} Y(T) = (1-\alpha)\,(I-\alpha S)^{-1} Y(0)$$

Label Update (After *T* times)

$$\boldsymbol{Y^{(1)} = \alpha S Y^{(0)} + (1-\alpha) Y^{(0)}}$$

$$\begin{aligned} \boldsymbol{Y^{(2)}} &= \alpha S Y^{(1)} + (1-\alpha) Y^{(0)} \\ &= \alpha S\big(\alpha S Y^{(0)} + (1-\alpha) S Y^{(0)}\big) + (1-\alpha) Y^{(0)} \\ &= \boldsymbol{\alpha^2 S^2 Y^{(0)} + \alpha(1-\alpha) S Y^{(0)} + (1-\alpha) Y^{(0)}} \end{aligned}$$

$$\begin{aligned} \boldsymbol{Y^{(3)}} &= \alpha S Y^{(2)} + (1-\alpha) Y^{(0)} \\ &= \boldsymbol{\alpha^3 S^3 Y^{(0)} + \alpha^2(1-\alpha) S^2 Y^{(0)} + \alpha(1-\alpha) S Y^{(0)} + (1-\alpha) Y^{(0)}} \end{aligned}$$

$$\therefore\ \underline{\boldsymbol{Y^{(T)} = \alpha^T S^T Y(0) + (1-\alpha) \textstyle\sum_{t=0}^{T-1} \alpha^t S^t Y(0)}}$$

Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 2) Label Propagation Process

- 초기 레이블 행렬을 정의하고 이를 기점으로 레이블 전파 과정을 거쳐 고품질의 Pseudo-label 형성

$$Y(0) = (y_{ij})_{n\times m}, \qquad where\ y_{ij} = p_j(x_i)$$

$$Y(t+1) = \alpha S Y(t) + (1-\alpha) Y(0)$$

$$Y(T) = \alpha^T S^T Y(0) + (1-\alpha)\sum_{t=0}^{T-1} \alpha^t S^t Y(0)$$

$$Y^* = \lim_{T\to\infty} Y(T) = (1-\alpha)\,(I-\alpha S)^{-1} Y(0)$$

Label Update (After ∞ times)

정규화 행렬 $S = D^{-1/2} W D^{-1/2}$의 스펙트럼 $\rho(S) \le 1$

Hyperparameter $\alpha \le 1$

$$\therefore \lim_{T\to\infty} \alpha^T S^T Y(0) = 0$$

$$\left.\begin{aligned} &\lim_{T\to\infty} (1-\alpha)\textstyle\sum_{t=0}^{T-1} \alpha^t S^t Y(0) \\ &= \frac{(1-\alpha)Y(0)}{(I-\alpha S)} \\ &= \boldsymbol{(1-\alpha)\,(I-\alpha S)^{-1} Y(0)} \end{aligned}\right\} \text{무한 등비급수의 합}$$

Methods

Improved Self-Training for Test-Time Adaptation

❖ ② Pseudo-Label Correction Algorithm – 2) Label Propagation Process

- 초기 레이블 행렬을 정의하고 이를 기점으로 레이블 전파 과정을 거쳐 고품질의 Pseudo-label 형성

$$Y(0) = (y_{ij})_{n\times m}, \quad where\ y_{ij} = p_j(x_i)$$

$$Y(t+1) = \alpha S Y(t) + (1-\alpha) Y(0)$$

$$Y(T) = \alpha^T S^T Y(0) + (1-\alpha)\sum_{t=0}^{T-1} \alpha^t S^t Y(0)$$

$$Y^* = \lim_{T\to\infty} Y(T) = (1-\alpha)\,(I-\alpha S)^{-1} Y(0)$$

기존 Pseudo-labeling

$$\hat{y}(x_i) = \arg\max_c \{p_c(\tilde{x}_j)\}_{j=1}^N$$

신규 Pseudo-labeling

$$\hat{y}(x_i) = \arg\max_c \{p_c(\tilde{x}_j) \times \mathbb{I}_{\arg\max k\ y^*_{jk}=c}\}_{j=1}^m$$

Experiments

Improved Self-Training for Test-Time Adaptation

❖ Experiments Results – Test-time Adaptation

- Batch Normalization, Self-supervised, Data Augmentation Strategy의 Test-time Adaptation과 비교 실험
- 고품질의 Pseudo-labeling 기반의 Self-training(Proposed)을 적용한 Test-time Adaptation의 성능이 가장 우수함을 증명

Methods	SF	Gauss	Shot	Impul	Defcs	Gls	Mtn	Zm	Snw	Frst	Fg	Brt	Cnt	Els	Px	Jpg	Mean
Source	✓	28.8	22.9	26.2	9.5	20.6	10.6	9.3	14.2	15.3	17.5	7.6	20.9	14.7	41.3	14.7	18.3±0.00
TTT++ (Offline) [20]	✓	12.8	11.1	11.2	7.3	17.1	8.2	6.5	9.4	9.9	7.9	5.0	5.1	13.7	8.8	10.6	9.6±0.00
SHOT (Offline) [19]	✓	13.4	11.6	16.3	7.3	15.9	8.2	7.1	9.4	9.4	10.2	6.3	8.3	12.8	9.8	13.6	10.6±0.00
BN Adapt [27]	✓	15.9	13.7	18.0	7.8	18.3	8.9	8.0	10.8	9.6	12.7	6.1	9.4	13.5	14.3	14.5	12.1±0.01
TENT [30]	✓	14.5	12.4	17.7	7.7	17.7	8.8	7.9	10.3	9.6	12.0	6.1	9.0	13.4	11.3	14.5	11.5±0.02
MEMO [35]	✓	13.9	12.2	16.3	7.4	16.6	8.2	7.4	9.8	9.3	10.7	6.1	9.3	12.6	**10.0**	14.3	10.9±0.02
TTT++ (Online) [20]	✓	15.5	14.1	23.6	9.1	25.1	11.4	8.1	13.2	13.1	13.4	6.6	**6.9**	17.6	12.5	13.6	13.6±0.03
SHOT (Online) [19]	✓	14.5	12.3	17.7	7.8	17.8	8.7	7.9	10.4	9.6	12.1	6.1	9.0	13.4	11.4	14.4	11.5±0.02
Ours (Online)	✓	**12.8**	**11.4**	**14.9**	**6.7**	**15.8**	**7.7**	**6.9**	**8.9**	**8.6**	**10.1**	**5.6**	8.0	**11.9**	10.5	**12.8**	**10.2**±0.04

Methods	SF	Gauss	Shot	Impul	Defcs	Gls	Mtn	Zm	Snw	Frst	Fg	Brt	Cnt	Els	Px	Jpg	Mean
Source	✓	98.4	97.7	98.4	90.6	92.5	89.8	81.8	89.5	85.0	86.3	51.1	97.2	85.3	76.9	71.7	86.2±0.00
SHOT (Offline) [19]	✓	73.8	70.5	72.2	79.2	80.6	58.5	54.0	53.6	63.0	47.3	39.2	97.7	48.7	46.1	53.0	62.5±0.00
BN Adapt [27]	✓	87.1	90.6	89.5	87.6	93.4	80.0	71.9	70.6	81.5	65.9	46.8	89.8	73.5	63.2	67.5	77.3±0.30
TTT [29]	✓	73.7	71.4	73.1	**76.3**	93.4	71.3	66.6	64.4	81.3	52.4	41.7	**64.7**	55.7	52.2	55.7	66.3±0.00
MEMO [35]	✓	85.4	81.0	83.5	94.1	92.5	72.5	60.5	64.0	72.5	52.0	41.4	97.1	55.8	50.7	57.5	70.7±0.05
TENT [30]	✓	80.8	78.6	80.4	82.5	82.5	72.1	60.5	63.7	66.7	52.1	**39.2**	84.2	55.5	50.8	58.2	67.2±0.02
SHOT (Online) [19]	✓	83.9	82.3	83.7	83.9	83.8	72.6	61.9	65.7	68.6	54.8	39.4	85.9	58.1	53.1	62.3	69.3±0.03
Ours (Online)	✓	**72.9**	**70.8**	**73.1**	80.7	**79.7**	**69.6**	**57.4**	**59.8**	**63.1**	**50.0**	39.3	83.9	**51.8**	**48.5**	**50.8**	**63.4**±0.03

Experiments

Improved Self-Training for Test-Time Adaptation

❖ Experiments Results – Ablation Study

- Simple Augmentation, Pseudo-labeling Correction Algorithm(PLCA), off/online 상황에 따른 성능 비교
  - ➢ Offline : 모든 테스트 데이터를 한번에 접근 가능한 설정, 이전 데이터에 대한 재학습 O / Online : 테스트 데이터를 순차적으로 입력, 이전 데이터에 대한 재학습 X
- PLCA의 적용에 따른 성능 향상이 상대적으로 우수하며 Simple Augmentation과 접목했을 때 더 높은 성능 향상을 보임

Methods	Models	CIFAR10	CIFAR100	Food101	Cars	ImageNet	Models	CIFAR10	CIFAR100	Food101	Cars	ImageNet
CLIP (zero-shot) [25]	ResNet-50	68.7±0.0	37.9±0.0	74.4±0.0	53.9±0.0	55.8±0.0	ResNet-101	78.1±0.0	41.2±0.0	77.5±0.0	60.7±0.0	58.2±0.0
Ours (SimpAug)		69.0±0.1	38.7±0.1	75.4±0.0	54.0±0.0	56.8±0.0		78.8±0.1	42.0±0.1	78.7±0.0	61.0±0.0	59.2±0.0
Ours (PLCA)		78.9±0.1	49.0±0.1	80.8±0.0	62.0±0.0	**60.9**±0.0		84.2±0.1	55.0±0.1	83.9±0.0	68.8±0.0	**63.2**±0.0
Ours (Online)		83.0±0.2	52.8±0.4	79.5±0.2	56.9±0.4	58.3±0.1		89.1±0.1	59.0±0.2	82.6±0.2	64.0±0.3	60.1±0.0
Ours (Offline)		**84.6**±0.1	**54.1**±0.5	**81.0**±0.0	**65.9**±0.1	60.2±0.0		**92.1**±0.1	**63.4**±0.1	**85.1**±0.0	**71.4**±0.2	62.6±0.0
CLIP (zero-shot) [25]	ViT-B/32	88.8±0.0	61.1±0.0	80.7±0.0	58.6±0.0	59.8±0.0	ViT-B/16	89.3±0.0	64.0±0.0	86.6±0.0	63.9±0.0	64.5±0.0
Ours (SimpAug)		89.6±0.1	62.1±0.1	82.3±0.0	58.9±0.0	61.1±0.0		90.1±0.1	65.7±0.1	87.6±0.0	64.3±0.0	65.5±0.0
Ours (PLCA)		92.8±0.1	69.7±0.1	86.1±0.0	65.7±0.0	65.8±0.0		94.3±0.0	73.0±0.1	90.3±0.0	71.0±0.0	69.9±0.0
Ours (Online)		92.9±0.1	72.1±0.2	86.3±0.1	62.1±0.3	62.1±0.1		94.8±0.0	74.6±0.1	90.2±0.0	67.0±0.1	67.2±0.1
Ours (Offline)		**95.7**±0.0	**75.9**±0.1	**87.1**±0.1	**69.3**±0.0	**66.2**±0.0		**96.4**±0.0	**78.5**±0.1	**91.8**±0.0	**73.4**±0.1	**72.0**±0.1

SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models

(2024 CVPR)

Methods

SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- 2024년에 제안된 Self-supervised Learning 기반 Test-time adaptation 방법론(CVPR, 2025년 2월 기준 31회 인용)
- 대조 학습 기반의 Self-supervised Learning 학습 방식을 통해 Online으로 Prompt를 Update하는 Vision-Language Model 제안

SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models

Xiaosong Ma
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
xiaosong16.ma@connect.polyu.hk

Jie Zhang *
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
jie-comp.zhang@polyu.edu.hk

Song Guo
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China
songguo@cse.ust.hk

Wenchao Xu
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
wenchao.xu@polyu.edu.hk

Abstract

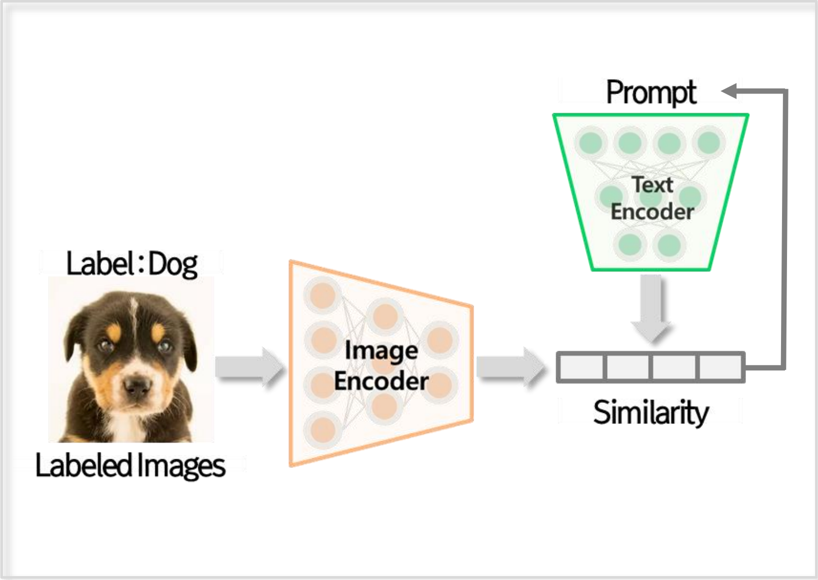
Test-time adaptation (TTA) is a special and practical setting in unsupervised domain adaptation, which allows a pre-trained model in a source domain to adapt to unlabeled test data in another target domain. To avoid the computation-intensive backbone fine-tuning process, the zero-shot generalization potentials of the emerging pre-trained vision-language models (e.g., CLIP, CoOp) are leveraged to only tune the run-time prompt for unseen test domains. However, existing solutions have yet to fully exploit the representation capabilities of pre-trained models as

Methods

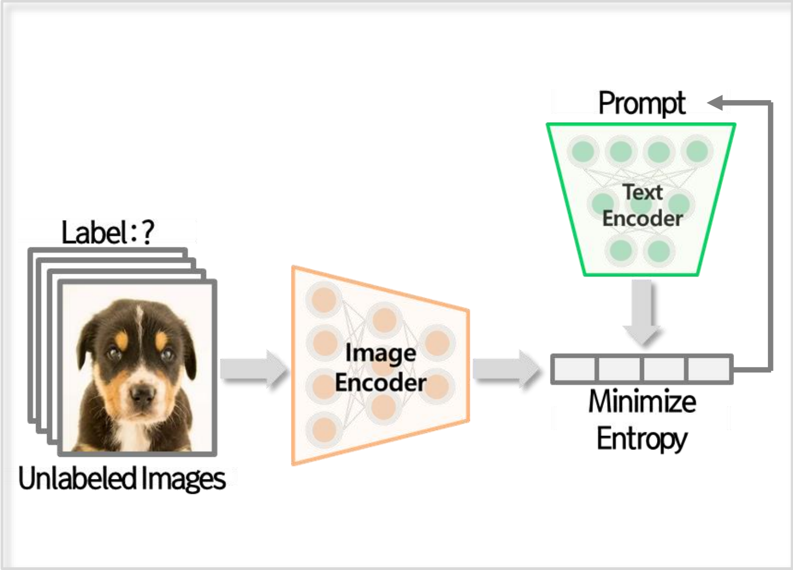
SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models

❖ 기존 Vision-Language Model의 한계점

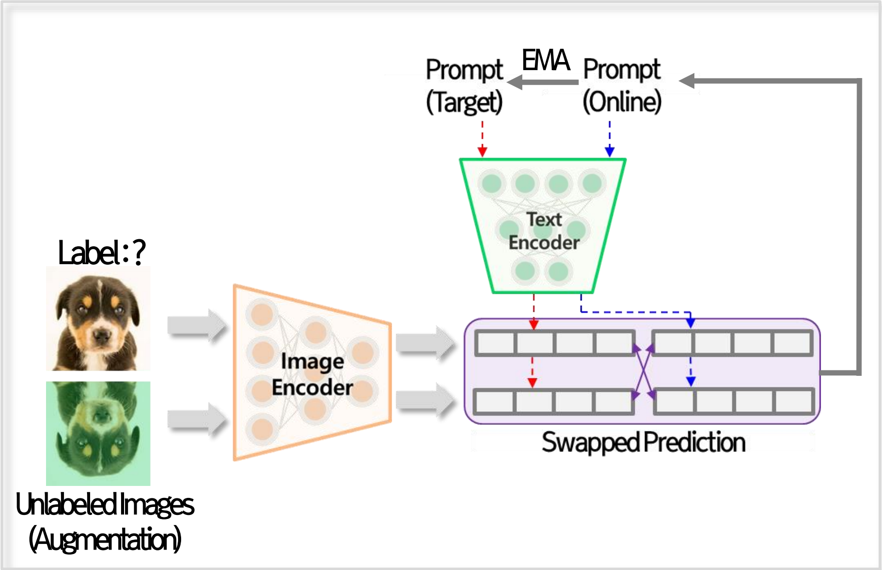
- 대표적인 Vision-Language Model로는 CoOp [3], TPT [4] 알고리즘이 있으나, 아래와 같은 한계점이 존재함
  - ➢ CoOp [1] : 프롬프트 학습을 위해 Label이 포함된 데이터가 필요하며 타겟 도메인으로의 적응력에 대한 학습 효율성 문제
  - ➢ TPT [2] : 엔트로피 최소화 방식에 의존하기 때문에 over-confidence 문제 및 새로운 도메인에 충분한 적응력을 보이지 못함



[CoOp Framework]



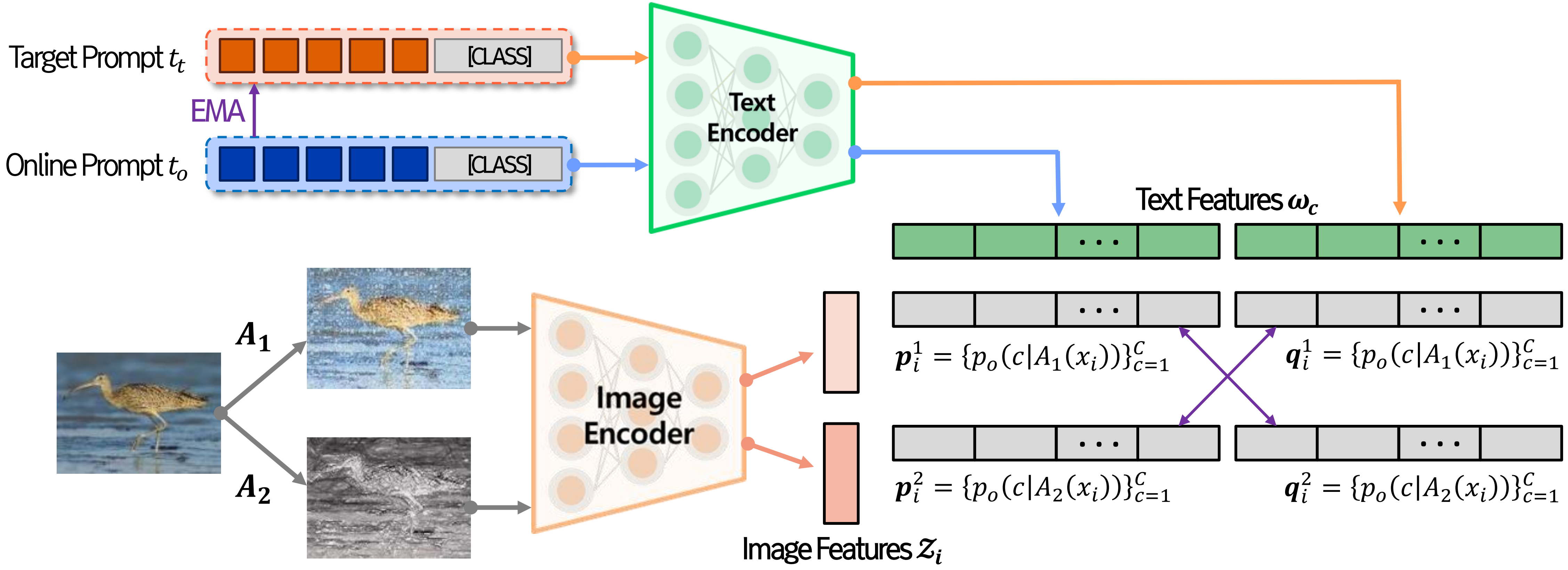
[TPT Framework]



[SwapPrompt Framework]

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- Vision-Language 모델에서의 Prompt를 “Online Prompt”와 “Target Prompt”로 역할을 분리하여 학습 효율성 확보

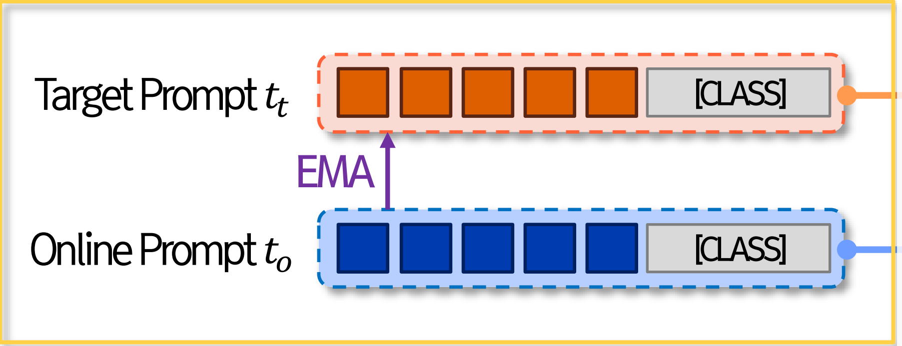


Methods

SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- Vision-Language 모델에서의 Prompt를 "Online Prompt"와 "Target Prompt"로 역할을 분리하여 학습 효율성 확보



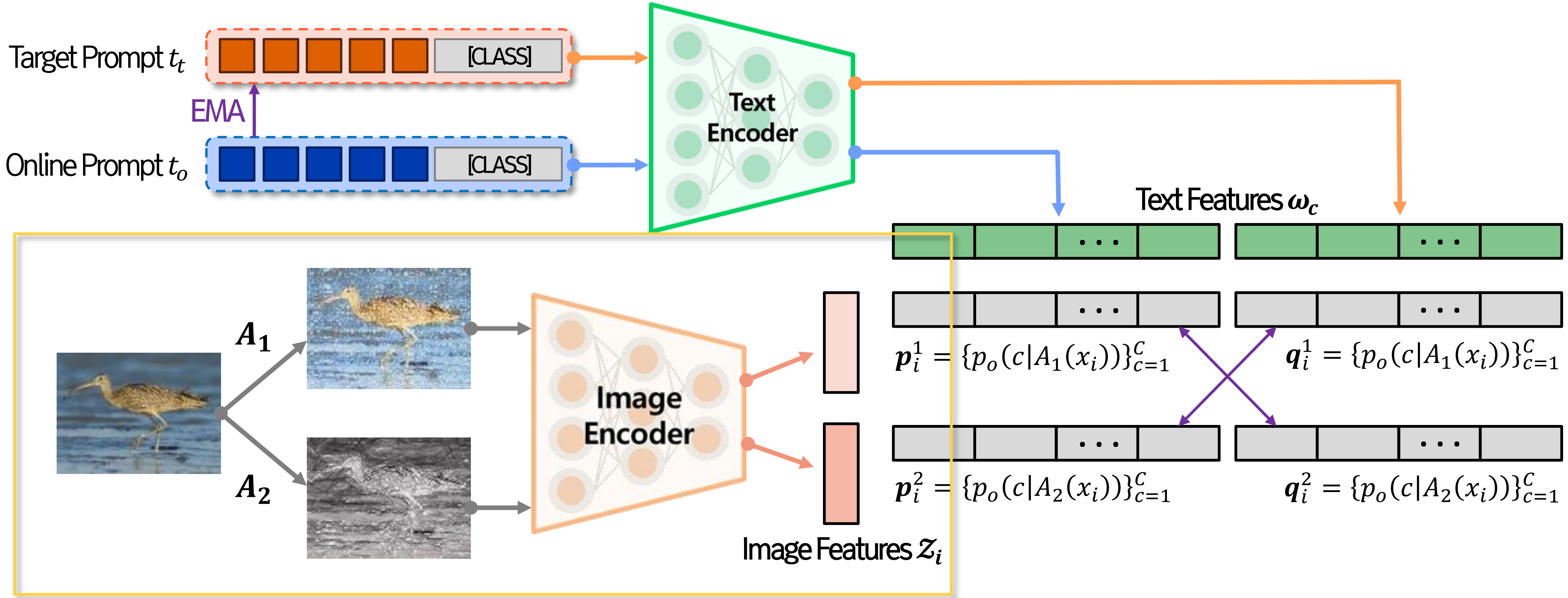
Online Prompt t_o ⟷ Target Prompt t_t

	Online Prompt t_o	**Target Prompt t_t**
정의	현재 학습되고 있는 프롬프트	Online Prompt의 과거 정보를 반영하는 EMA 버전
역할	새로운 데이터에 대해 업데이트 되는 학습 가능한 프롬프트	학습을 돕는 참조 프롬프트
업데이트 방식	손실 함수($\mathcal{L}_{swap}, \mathcal{L}_{pseudo}$) 기반으로 학습	Online Prompt의 EMA로 업데이트
변화 속도	빠르게 변하며 최신 데이터에 적응	천천히 변하며 이전 정보를 보존
활용 방식	학습 과정에서 직접 최적화	온라인 프롬프트를 보완 클래스 확률 계산에 활용

Methods

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

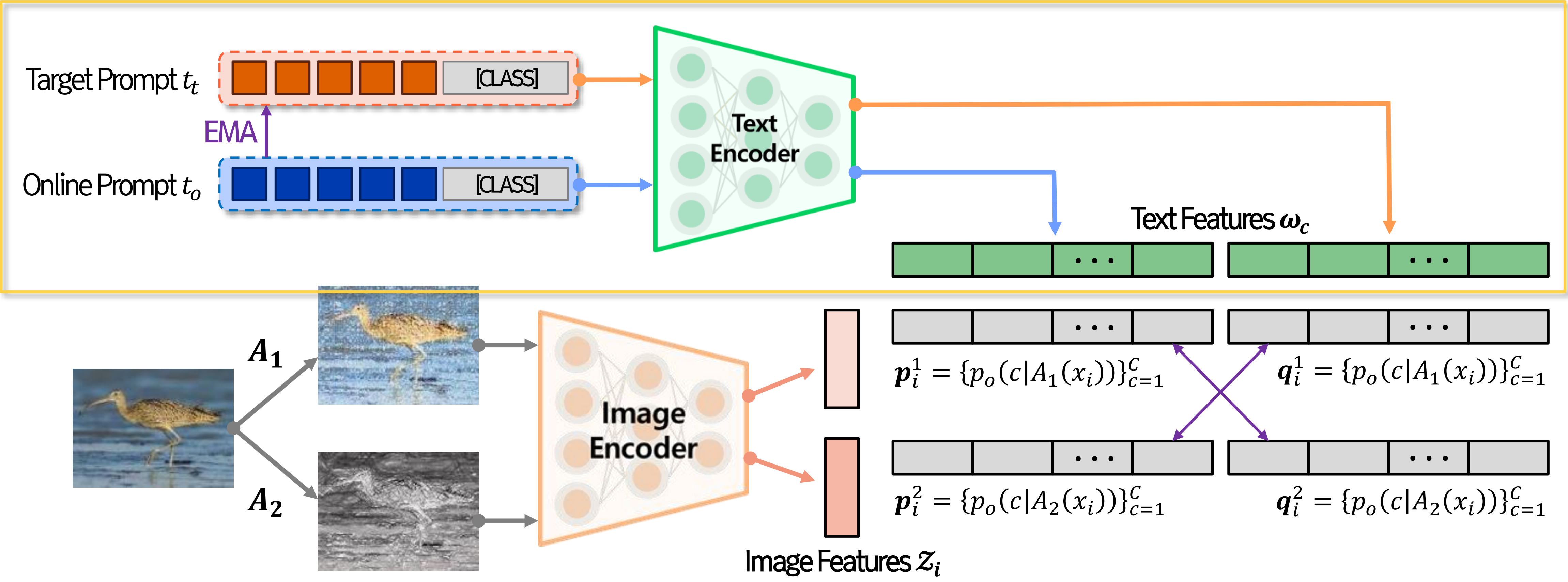
- Vision-Language 모델에서의 Prompt를 “Online Prompt”와 “Target Prompt”로 역할을 분리하여 학습 효율성 확보



Methods

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

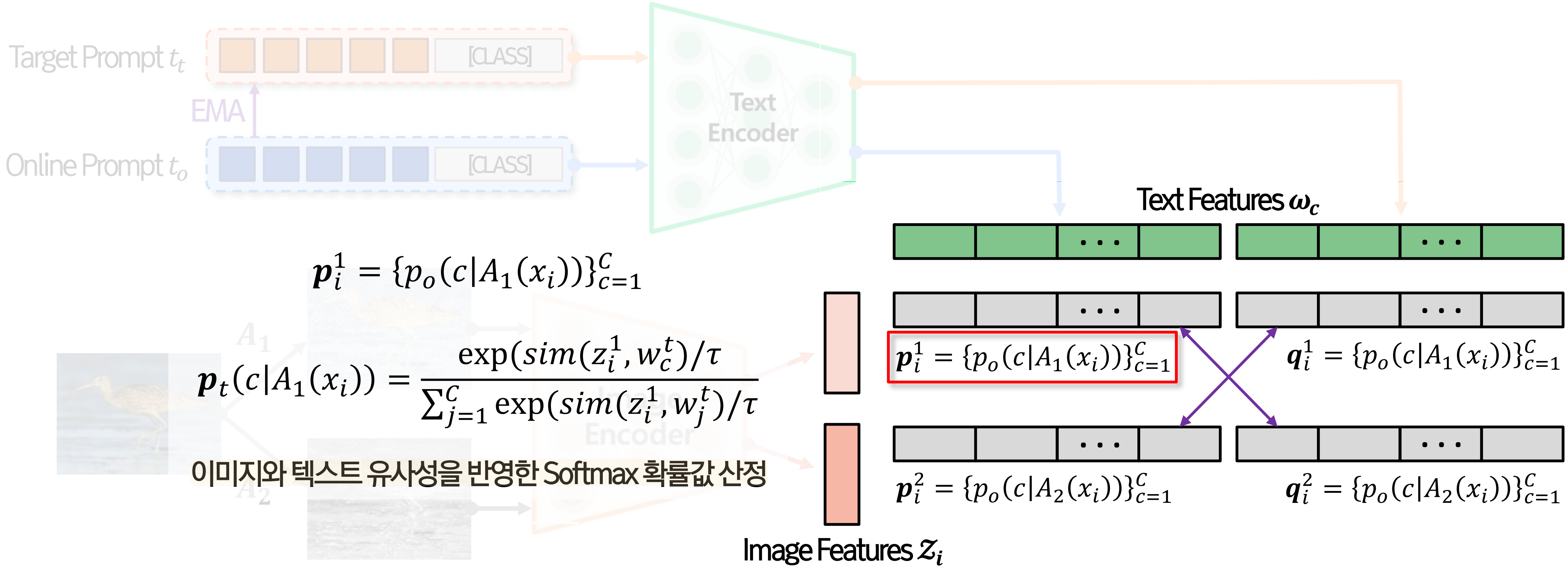
- Vision-Language 모델에서의 Prompt를 “Online Prompt”와 “Target Prompt”로 역할을 분리하여 학습 효율성 확보



Methods

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- Vision-Language 모델에서의 Prompt를 "Online Prompt"와 "Target Prompt"로 역할을 분리하여 학습 효율성 확보



$$\boldsymbol{p}_i^1 = \{p_o(c|A_1(x_i))\}_{c=1}^C$$

$$\boldsymbol{p}_t(c|A_1(x_i)) = \frac{\exp(sim(z_i^1, w_c^t)/\tau)}{\sum_{j=1}^C \exp(sim(z_i^1, w_j^t)/\tau)}$$

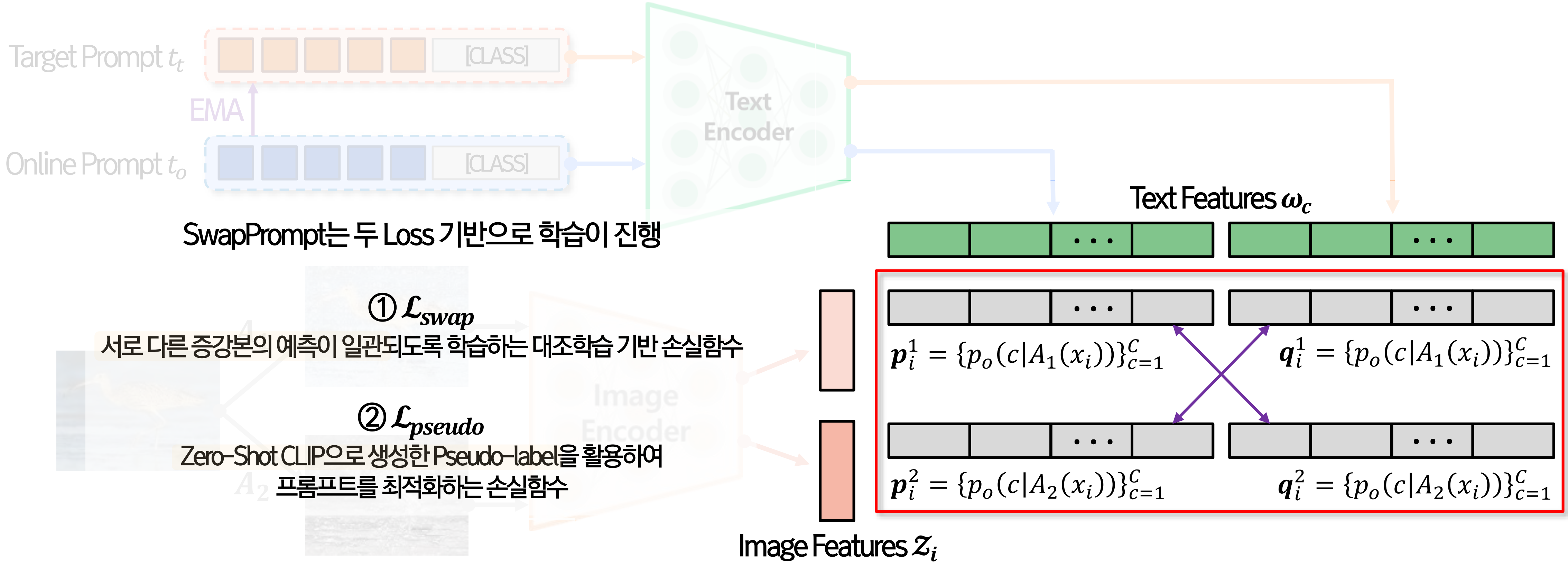
이미지와 텍스트 유사성을 반영한 Softmax 확률값 산정

Methods

SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- Vision-Language 모델에서의 Prompt를 "Online Prompt"와 "Target Prompt"로 역할을 분리하여 학습 효율성 확보



Methods

SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- Vision-Language 모델에서의 Prompt를 "Online Prompt"와 "Target Prompt"로 역할을 분리하여 학습 효율성 확보

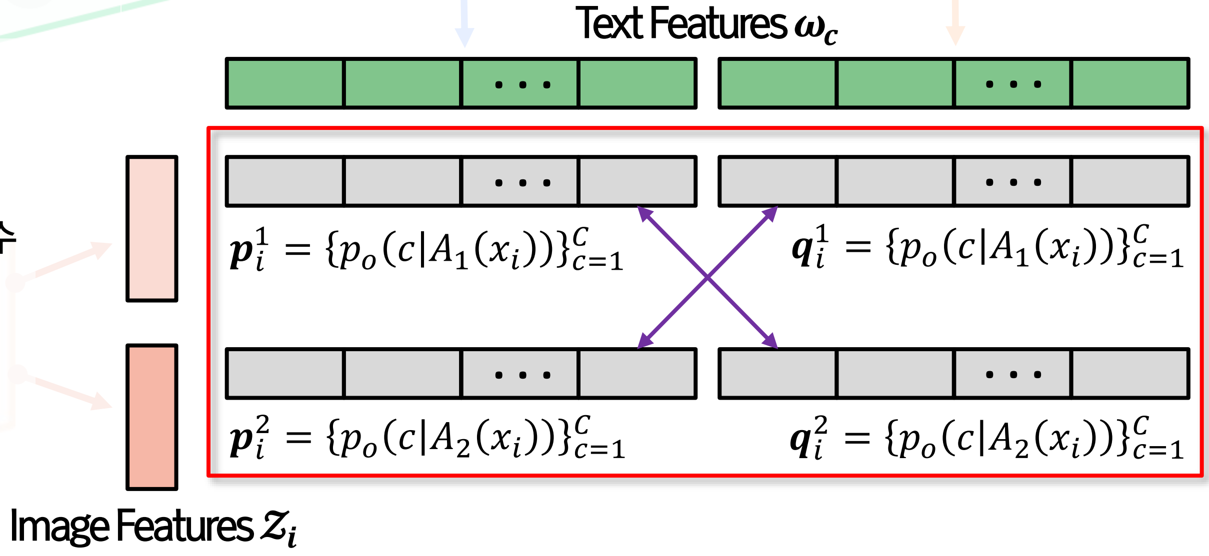
SwapPrompt는 두 Loss 기반으로 학습이 진행

① $\mathcal{L}_{swap}$

서로 다른 증강본의 예측이 일관되도록 학습하는 대조학습 기반 손실함수

$$\mathcal{L}_{swap} = \ell(p_i^1, q_i^2) + \ell(p_i^2, q_i^1)$$

$$\ell(p_i^1, q_i^2) = -\sum_{c=1}^{C} p_t(c|A_2(x_i)) \log p_o(c|A_1(x_i))$$

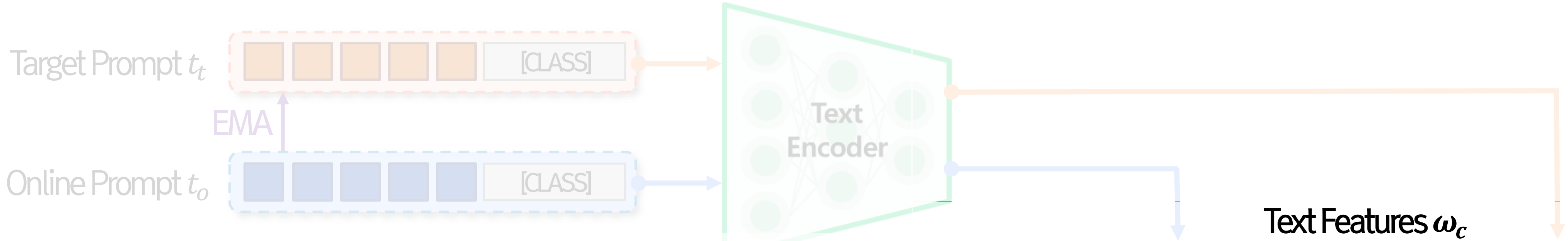


Methods

SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- Vision-Language 모델에서의 Prompt를 “Online Prompt”와 “Target Prompt”로 역할을 분리하여 학습 효율성 확보



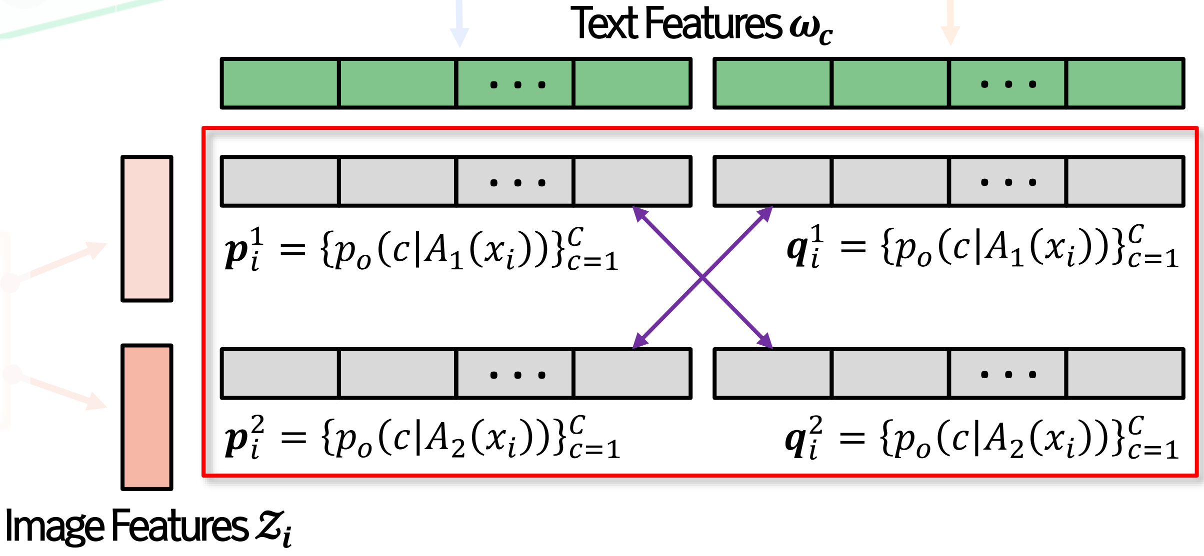
SwapPrompt는 두 Loss 기반으로 학습이 진행

② $\mathcal{L}_{pseudo}$

Zero-Shot CLIP으로 생성한 Pseudo-label을 활용하여 프롬프트를 최적화하는 손실함수

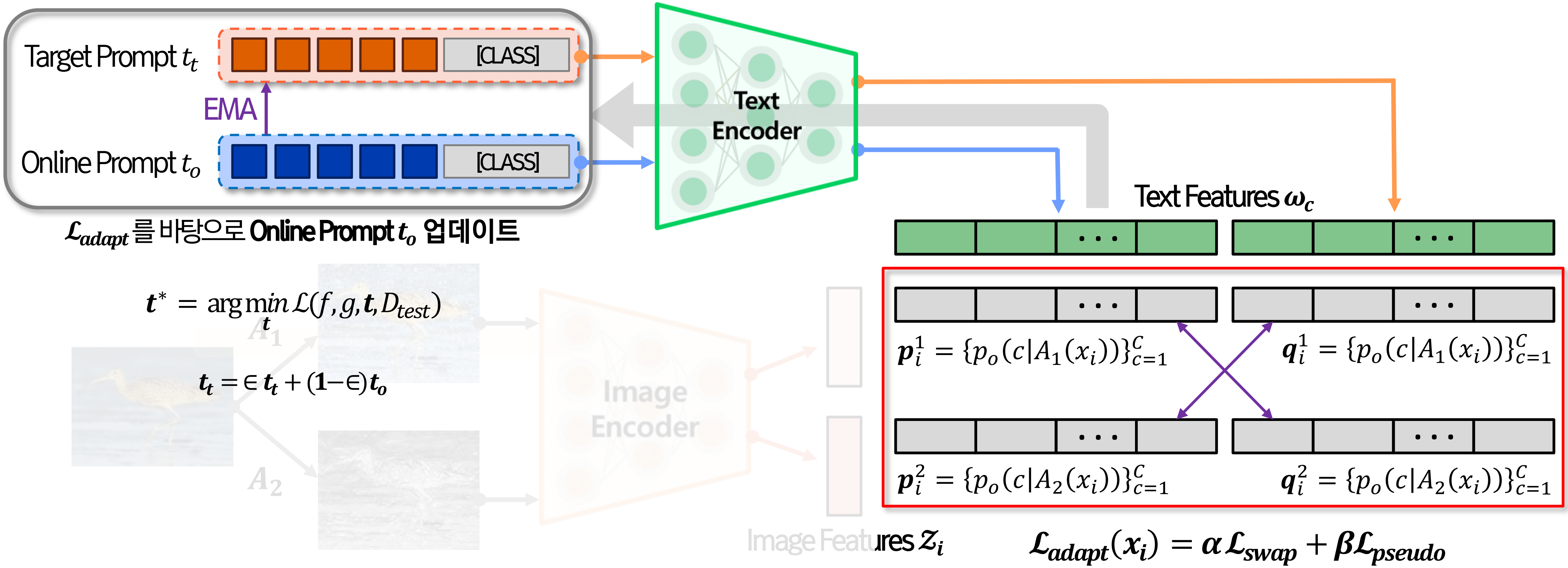
$$\mathcal{L}_{pseudo} = \ell_{ce}(p_i^1, \hat{y}_i) + \ell_{ce}(p_i^2, \hat{y}_i)$$

$$\hat{y}_i = \arg\max_c p_{CLIP}(c|x_i)$$



❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- Vision-Language 모델에서의 Prompt를 "Online Prompt"와 "Target Prompt"로 역할을 분리하여 학습 효율성 확보



Experiments

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- 14가지 데이터셋에 대하여 4가지 Vision-Language Models과의 성능 비교
  - ➢ CoOp[2] (Labeled Data 사용), UPL[3], TPT[4]
- 제안하는 SwapPrompt의 성능이 가장 우세했으며, Online 상황에서도 성능 하락폭이 크지 않는 학습 강건성을 보임

Method	Caltech101	DTD	Flowers102	Oxford-Pets	UCF101	StanfordCars	Food101	EuroSAT	SUN397	ImageNet	ImageNet-V2	ImageNet-A	ImageNet-R	ImageNet-Sketch
CoOp [17]	88.76	54.62	83.98	87.44	66.71	61.83	73.79	61.68	64.33	61.23	55.29	23.41	56.96	35.64
CLIP [16]	85.13	42.16	65.40	83.05	61.15	55.65	74.23	37.60	58.55	58.18	51.36	21.69	55.98	33.33
UPL [24]	86.37	45.04	67.11	88.53	63.63	58.46	74.38	41.40	61.07	61.19	52.07	23.59	57.09	36.40
TPT [19]	87.22	42.17	65.42	84.60	61.18	58.49	74.88	43.82	61.46	60.74	**54.35**	**26.24**	58.72	35.02
SwapPrompt	**89.90**	**47.34**	**70.22**	**89.14**	**65.66**	**59.60**	**75.08**	**46.64**	**63.93**	**61.80**	53.94	24.46	**60.88**	**38.21**
Δ	+2.68	+2.30	+3.11	+0.61	+2.03	+1.11	+0.20	+2.82	+2.47	+0.61	-0.41	-1.78	+2.16	+1.81
+ Online	89.69	46.40	68.12	88.97	64.52	58.88	75.66	42.45	63.36	61.41	52.93	24.42	60.25	38.13

Experiments

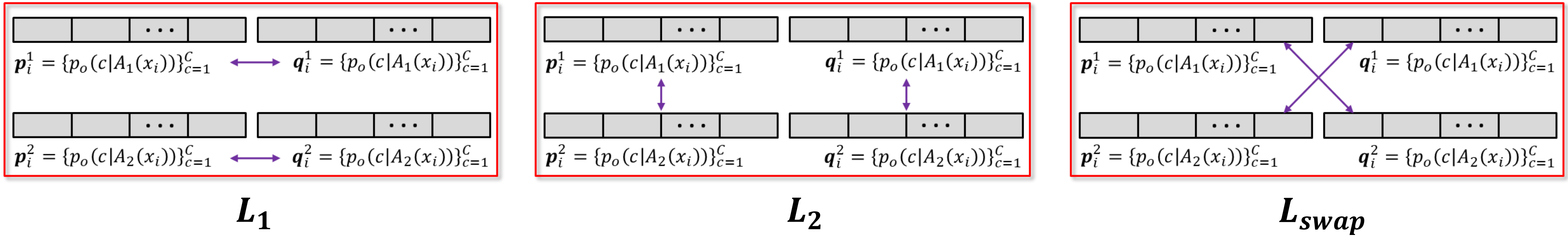
❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- Swapped Prediction의 효용성을 입증할 수 있는 Ablation Study 실험 결과
- Unsupervised Prompt Learning의 근간이 되는 UPL [3] 알고리즘을 Basis로 하여 Components를 추가하며 성능 향상 검증
- Unlabeled Data의 Augmentation 적용과 Swapped Prediction Strategy를 적용한 SwapPrompt의 성능 향상을 관찰

	ImageNet	Caltech101	DTD	Flowers102	Oxford-Pets	UCF101	Average
UPL [24]	61.19	86.37	45.04	67.11	88.53	63.63	68.65
UPL+AUG	61.30	87.75	46.04	68.43	87.67	65.15	69.39
SwapPrompt	**61.80**	**89.90**	**47.34**	**70.22**	**89.14**	**65.66**	**70.68**

❖ SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models [2]

- Swap 방식에 따른 성능 변화 비교 실험
- 프롬프트 간(L_1), 데이터 증강본 간(L_2), 그리고 프롬프트 및 데이터 증강본 간(L_{swap}) 중 제안하는 L_{swap}이 가장 높은 성능을 보임



	Caltech101	DTD	Flowers102	Oxford-Pets	UCF101	Average
L_1	87.38	47.22	69.63	87.71	64.68	71.32
L_2	88.45	46.69	69.28	87.30	64.46	71.24
$L_{\texttt{swap}}$	**89.90**	**47.34**	**70.22**	**89.14**	**65.66**	**72.45**

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization
(AAAI 2024)

Methods

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

- 2024년에 제안된 Batch Normalization 기반 Test-time adaptation 방법론(AAAI, 2025년 2월 기준 13회 인용)
- 자체적인 Batch Normalization 파라미터 조정 방법 및 메타 러닝을 활용한 보조 학습 방식의 TTA 방법론 제안

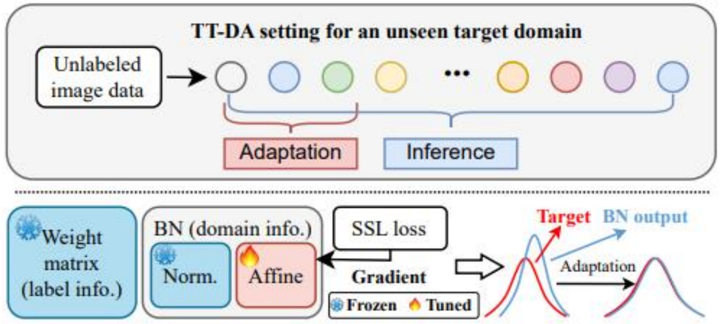
Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

Yanan Wu[1,2*], Zhixiang Chi[3*], Yang Wang[4], Konstantinos N. Plataniotis[3], Songhe Feng[1,2†]

[1]Key Laboratory of Big Data & Artificial Intelligence in Transportation,
Ministry of Education, Beijing Jiaotong University, Beijing, 100044, China
[2]School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, China
[3]The Edward S Rogers Sr. ECE Department, University of Toronto, Toronto, M5S3G8, Canada
[4]Department of Computer Science and Software Engineering, Concordia University, Montreal, H3G2J1, Canada
{ynwu0510,shfeng}@bjtu.edu.cn, zhixiang.chi@mail.utoronto.ca, yang.wang@concordia.ca, kostas@ece.utoronto.ca

Abstract

Test-time domain adaptation aims to adapt the model trained on source domains to unseen target domains using a few unlabeled images. Emerging research has shown that the label and domain information is separately embedded in the weight matrix and batch normalization (BN) layer. Previous works normally update the whole network naively without explicitly decoupling the knowledge between label and domain. As a result, it leads to knowledge interference and defective distribution adaptation. In this work, we propose to reduce

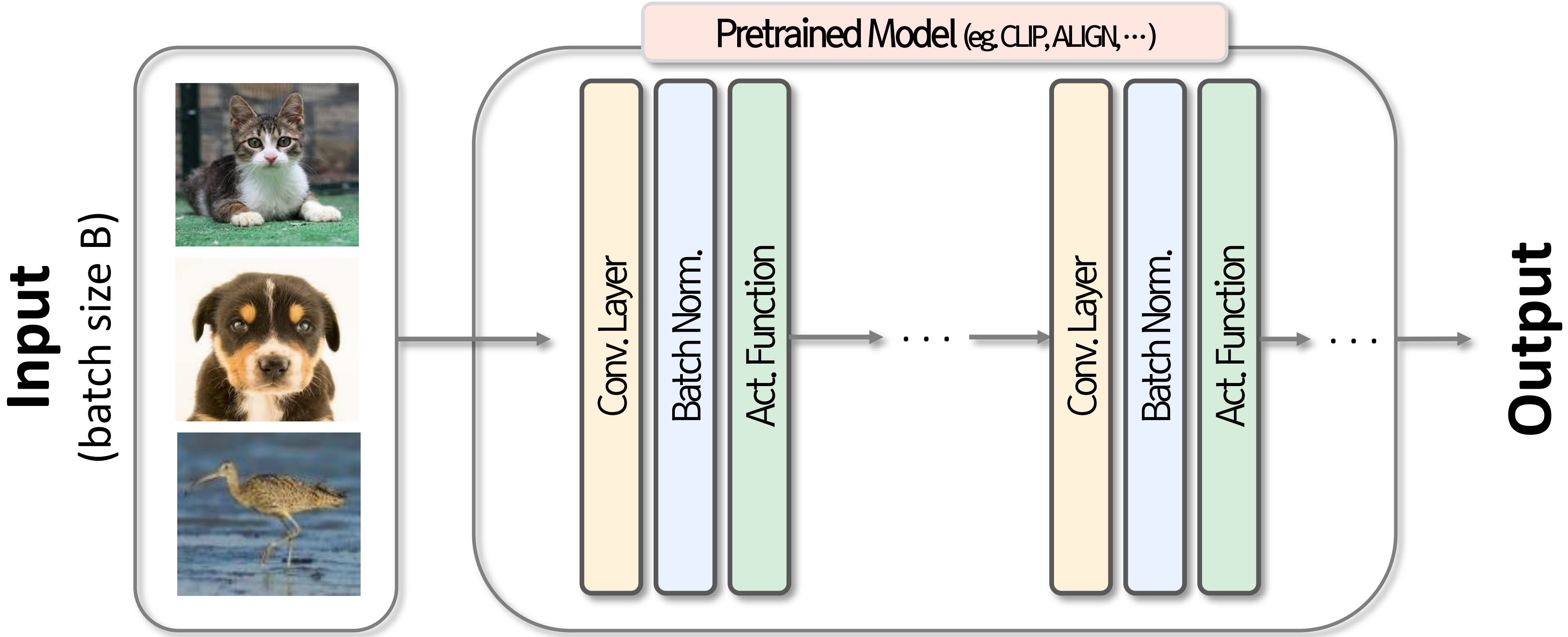


Methods

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

❖ Batch Normalization in Test-Time Adaptation

- 2024년에 제안된 Batch Normalization 기반 Test-time adaptation 방법론(AAAI, 2025년 2월 기준 13회 인용)
- 자체적인 Batch Normalization 파라미터 조정 방법 및 메타 러닝을 활용한 보조 학습 방식의 TTA 방법론 제안

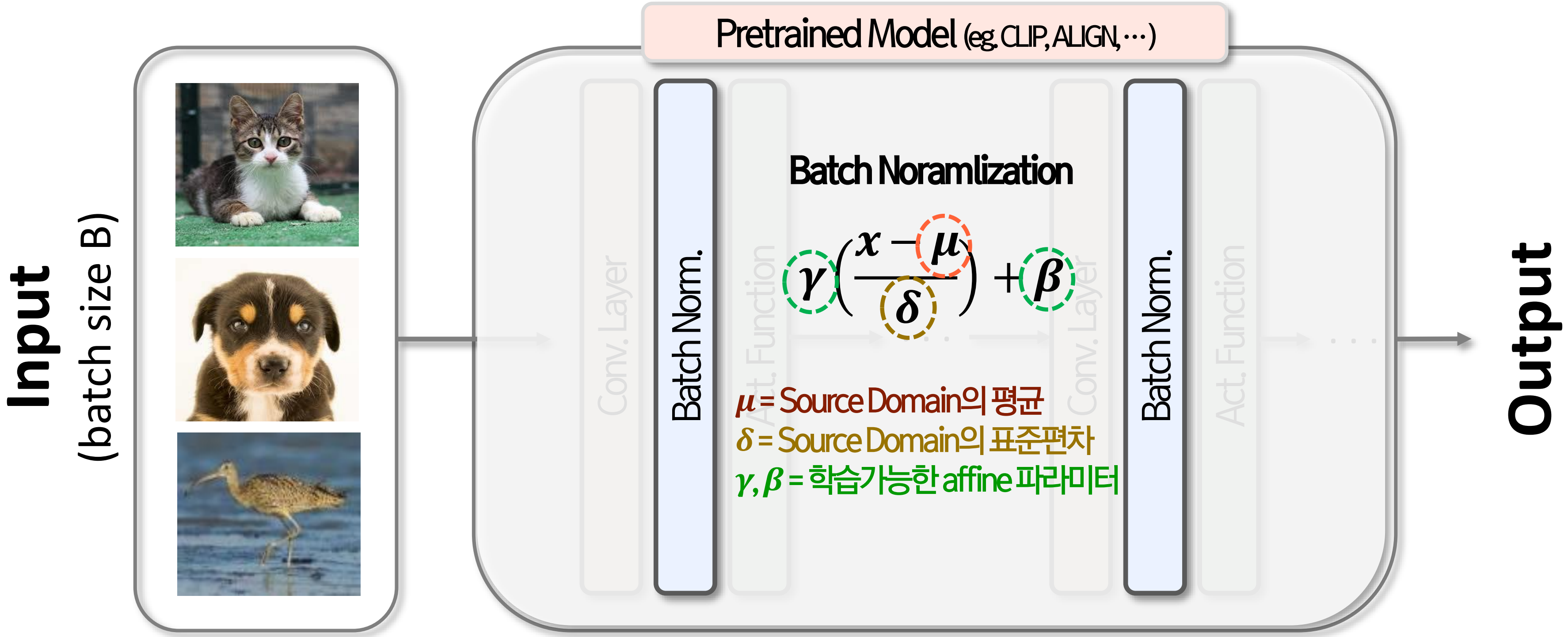


Methods

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

❖ Batch Normalization in Test-Time Adaptation

- 2024년에 제안된 Batch Normalization 기반 Test-time adaptation 방법론(AAAI, 2025년 2월 기준 13회 인용)
- 자체적인 Batch Normalization 파라미터 조정 방법 및 메타 러닝을 활용한 보조 학습 방식의 TTA 방법론 제안

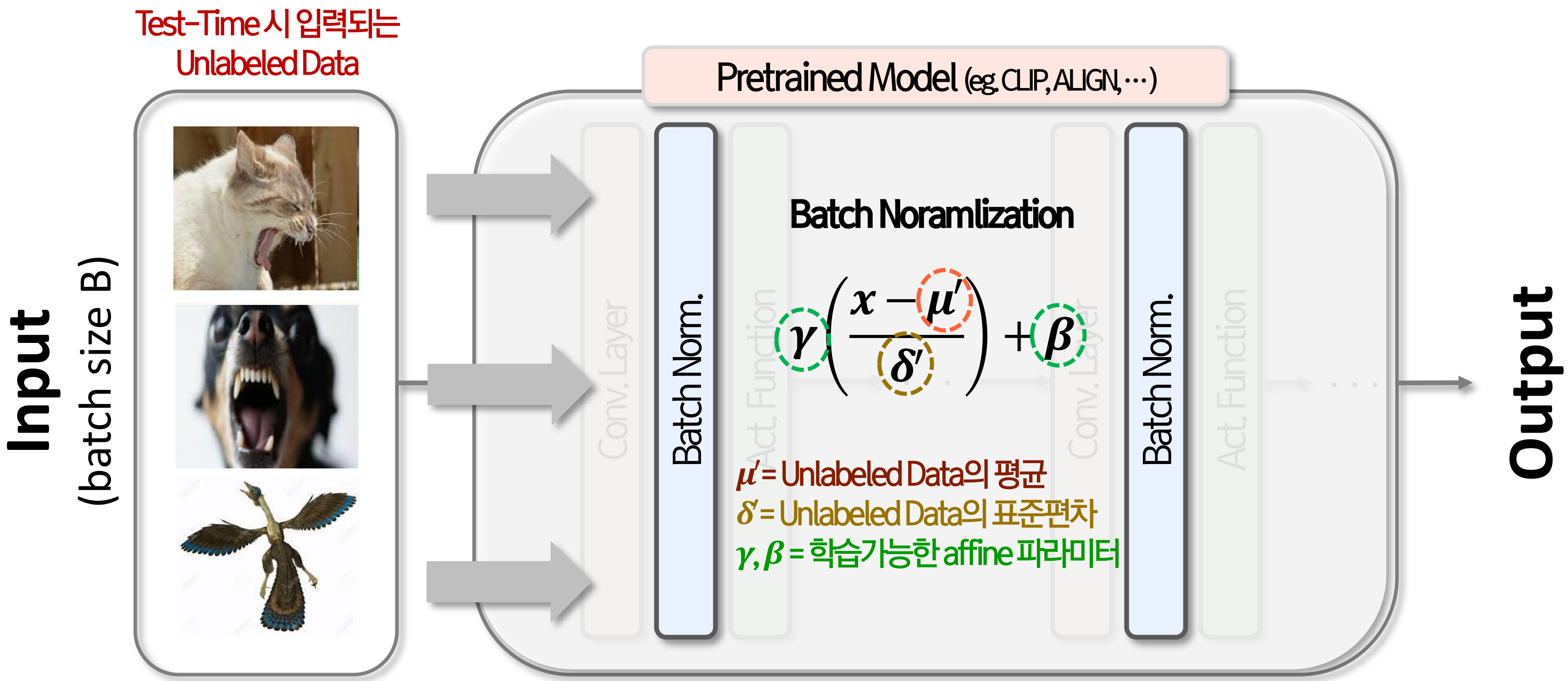


Methods

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

❖ Batch Normalization in Test-Time Adaptation

- 2024년에 제안된 Batch Normalization 기반 Test-time adaptation 방법론(AAAI, 2025년 2월 기준 13회 인용)
- 자체적인 Batch Normalization 파라미터 조정 방법 및 메타 러닝을 활용한 보조 학습 방식의 TTA 방법론 제안

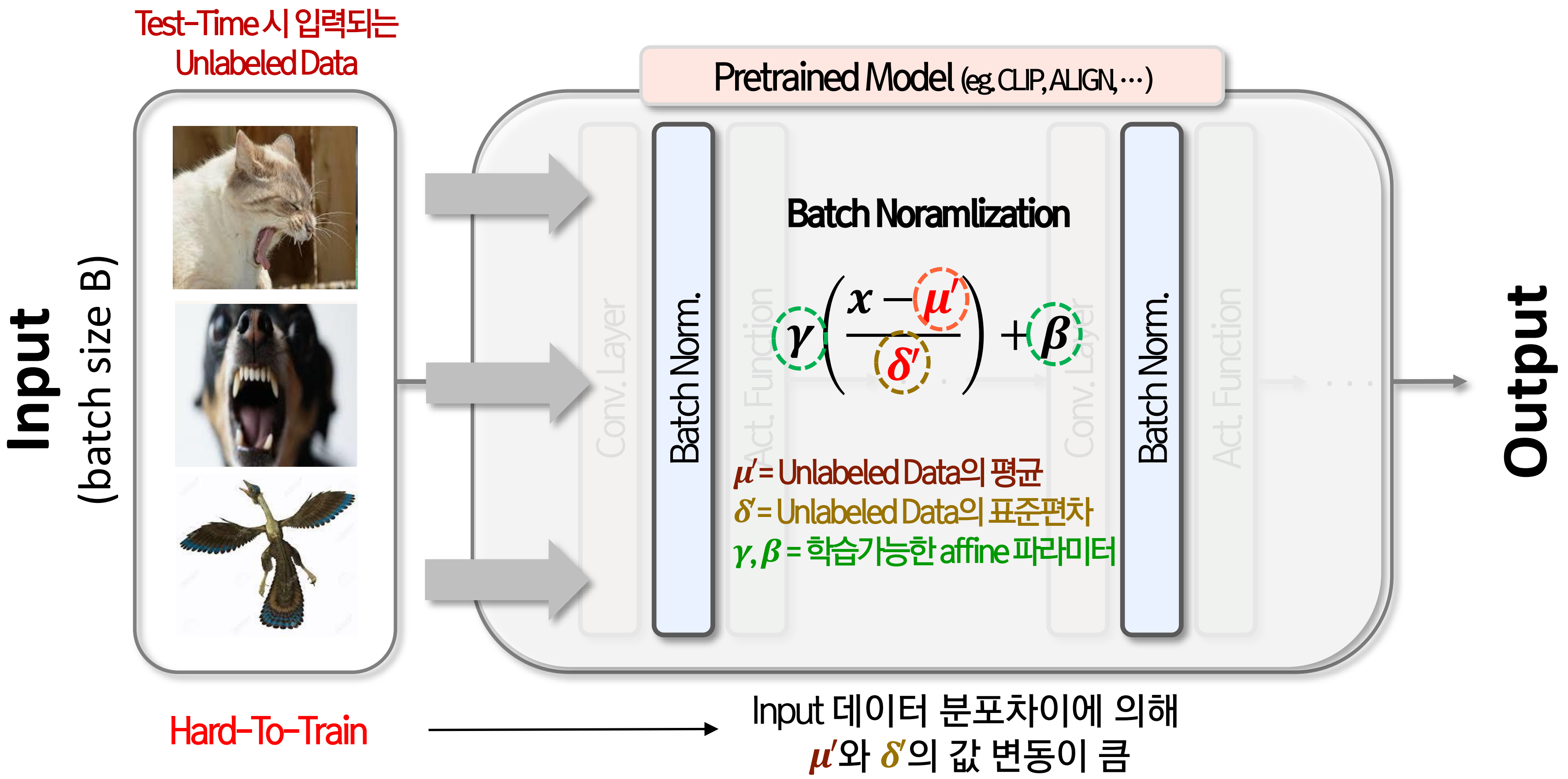


Methods

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

❖ Batch Normalization in Test-Time Adaptation

- 2024년에 제안된 Batch Normalization 기반 Test-time adaptation 방법론(AAAI, 2025년 2월 기준 13회 인용)
- 자체적인 Batch Normalization 파라미터 조정 방법 및 메타 러닝을 활용한 보조 학습 방식의 TTA 방법론 제안

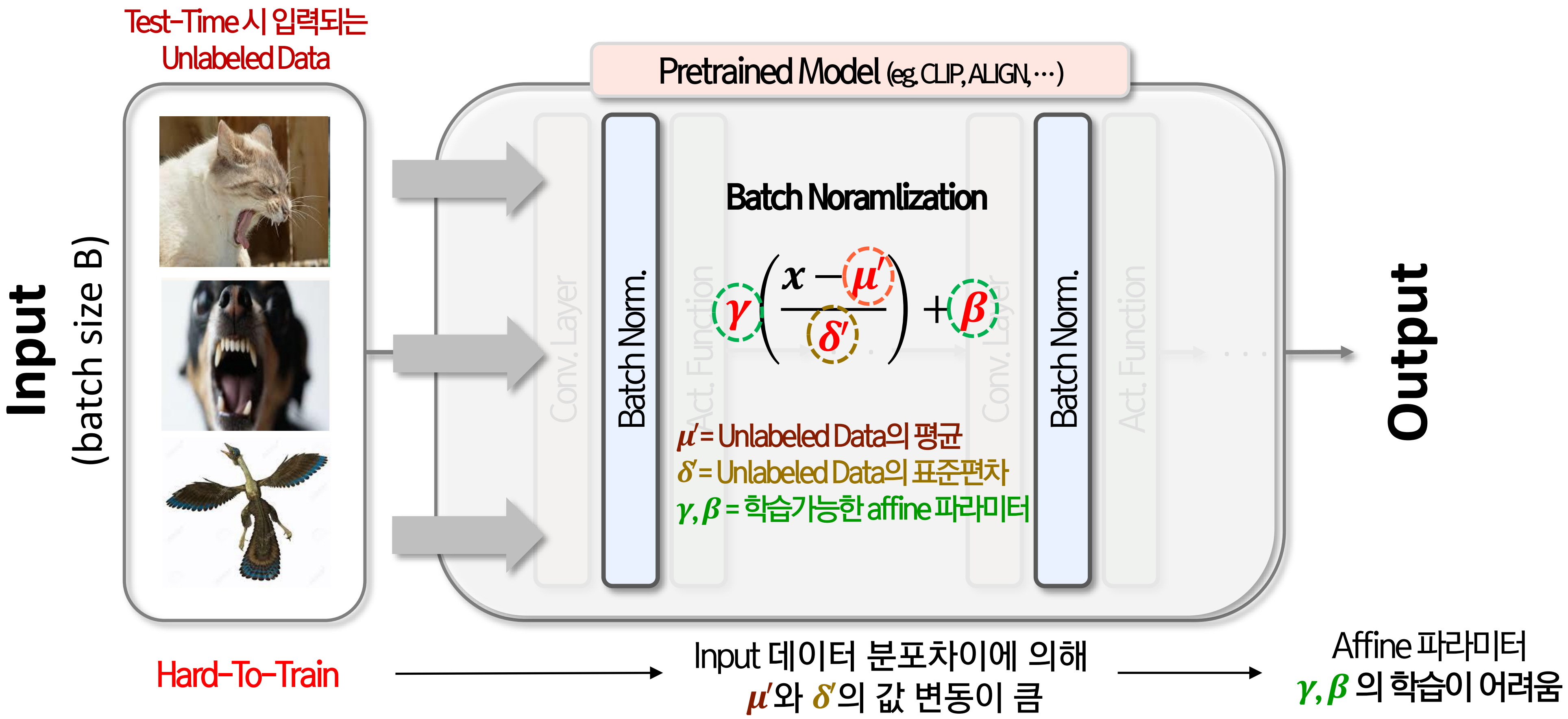


Methods

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

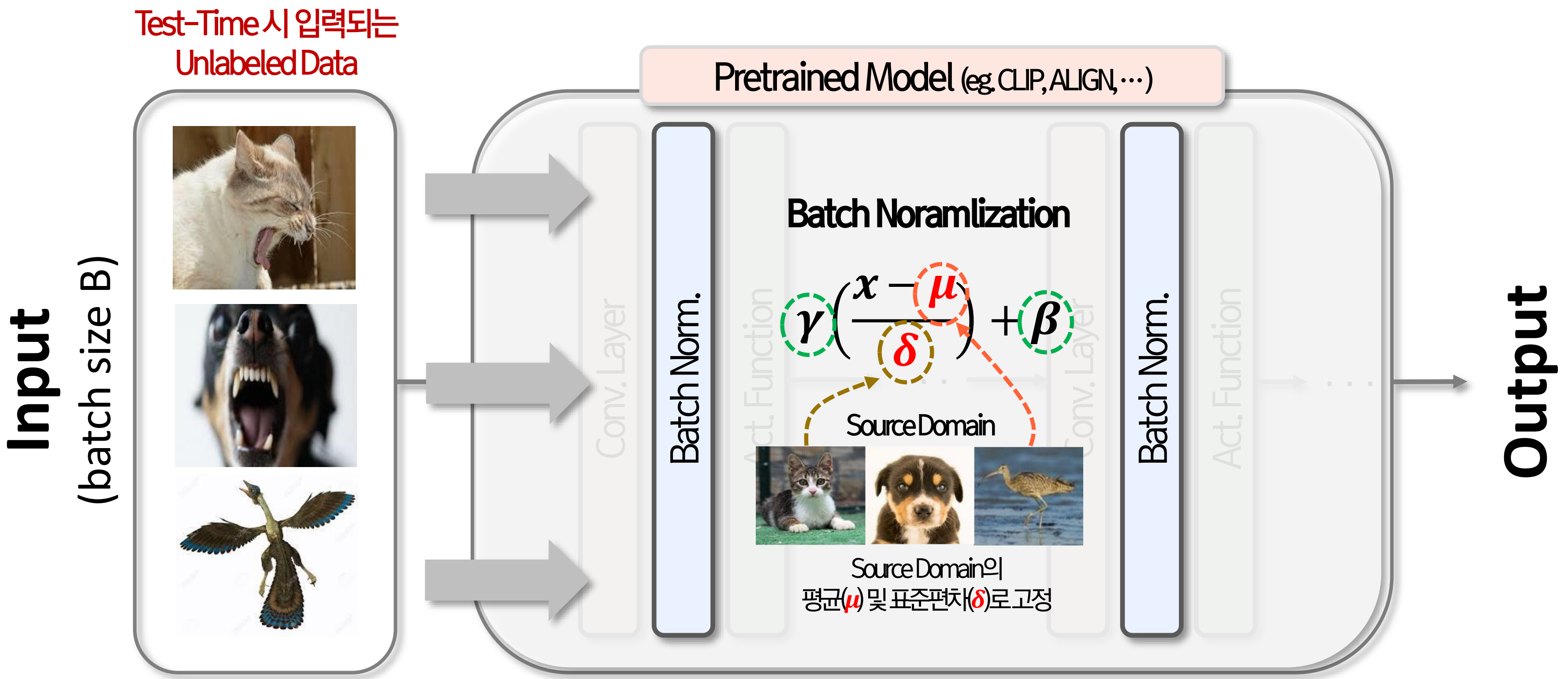
❖ Batch Normalization in Test-Time Adaptation

- 2024년에 제안된 Batch Normalization 기반 Test-time adaptation 방법론(AAAI, 2025년 2월 기준 13회 인용)
- 자체적인 Batch Normalization 파라미터 조정 방법 및 메타 러닝을 활용한 보조 학습 방식의 TTA 방법론 제안



❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

- 2024년에 제안된 Batch Normalization 기반 Test-time adaptation 방법론(AAAI, 2025년 2월 기준 13회 인용)
- 자체적인 Batch Normalization 파라미터 조정 방법 및 메타 러닝을 활용한 보조 학습 방식의 TTA 방법론 제안

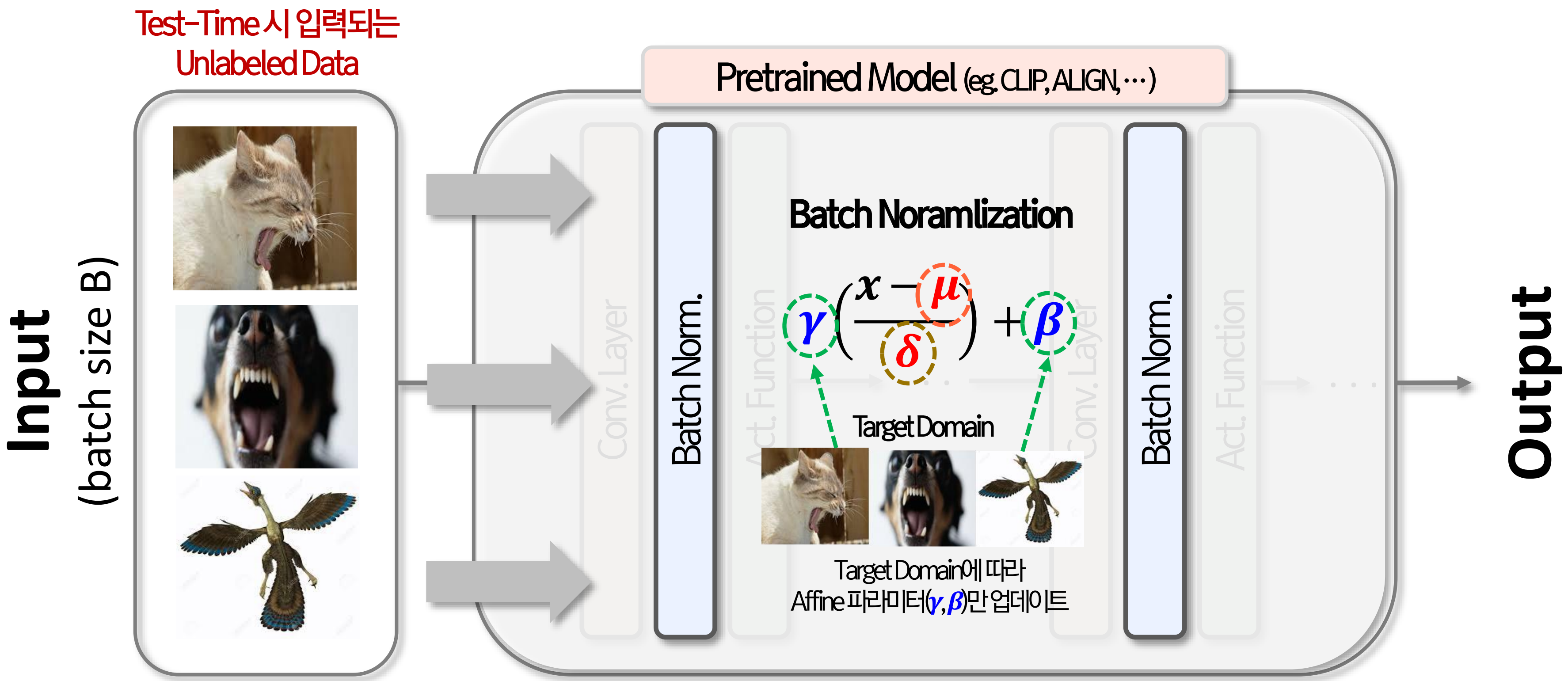


Methods

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

- 2024년에 제안된 Batch Normalization 기반 Test-time adaptation 방법론(AAAI, 2025년 2월 기준 13회 인용)
- 자체적인 Batch Normalization 파라미터 조정 방법 및 메타 러닝을 활용한 보조 학습 방식의 TTA 방법론 제안

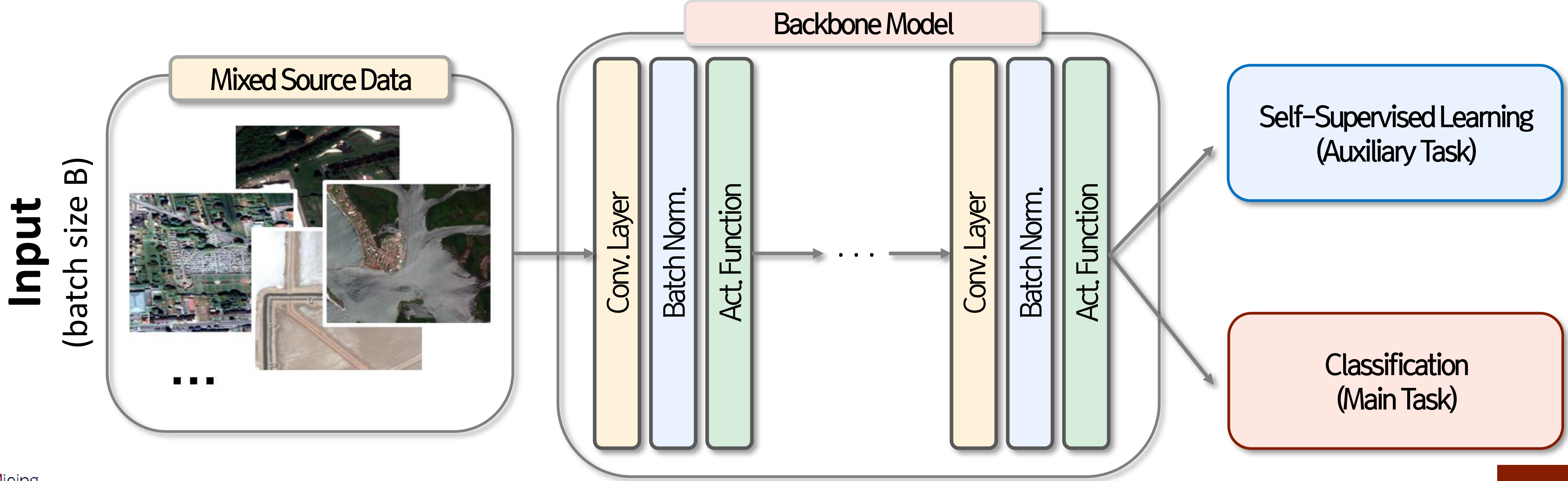


Methods

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

① Joint Training

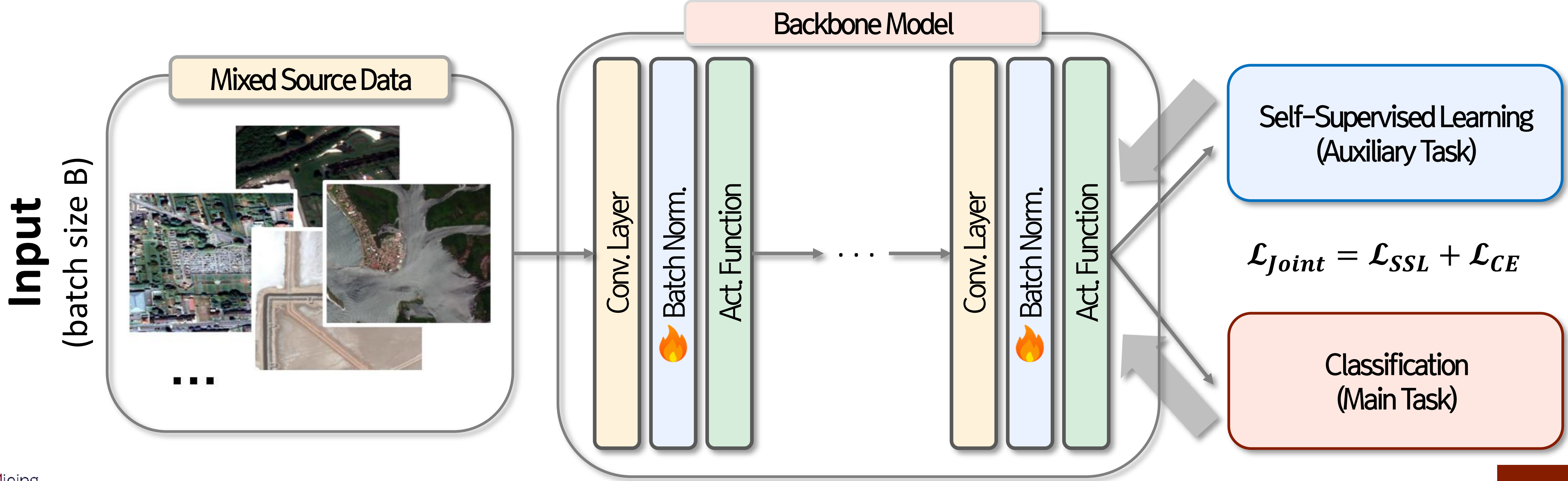
- Domain 구분이 없는 Mixed Source Data로 Auxiliary Task 및 Main Task을 통한 Backbone Model 업데이트
  - ➢ Auxiliary Task : BYOL 기반의 Self-supervised Learning
  - ➢ Main Task : 일반적인 Classification Task



❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

① Joint Training

- Domain 구분이 없는 Mixed Source Data로 Auxiliary Task 및 Main Task을 통한 Backbone Model 업데이트
  - ➢ Auxiliary Task : BYOL 기반의 Self-supervised Learning
  - ➢ Main Task : 일반적인 Classification Task

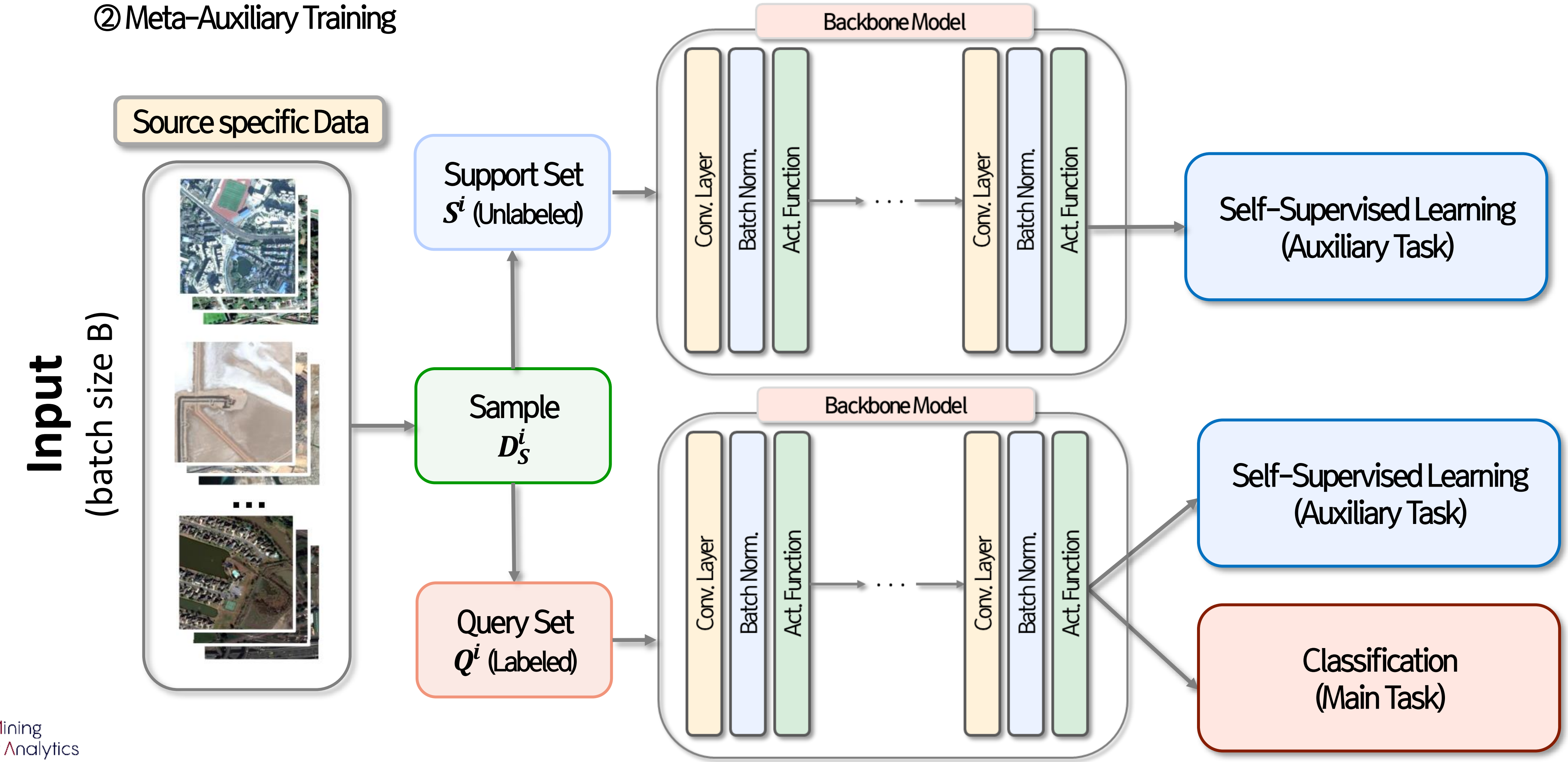


$$\mathcal{L}_{Joint} = \mathcal{L}_{SSL} + \mathcal{L}_{CE}$$

Methods

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

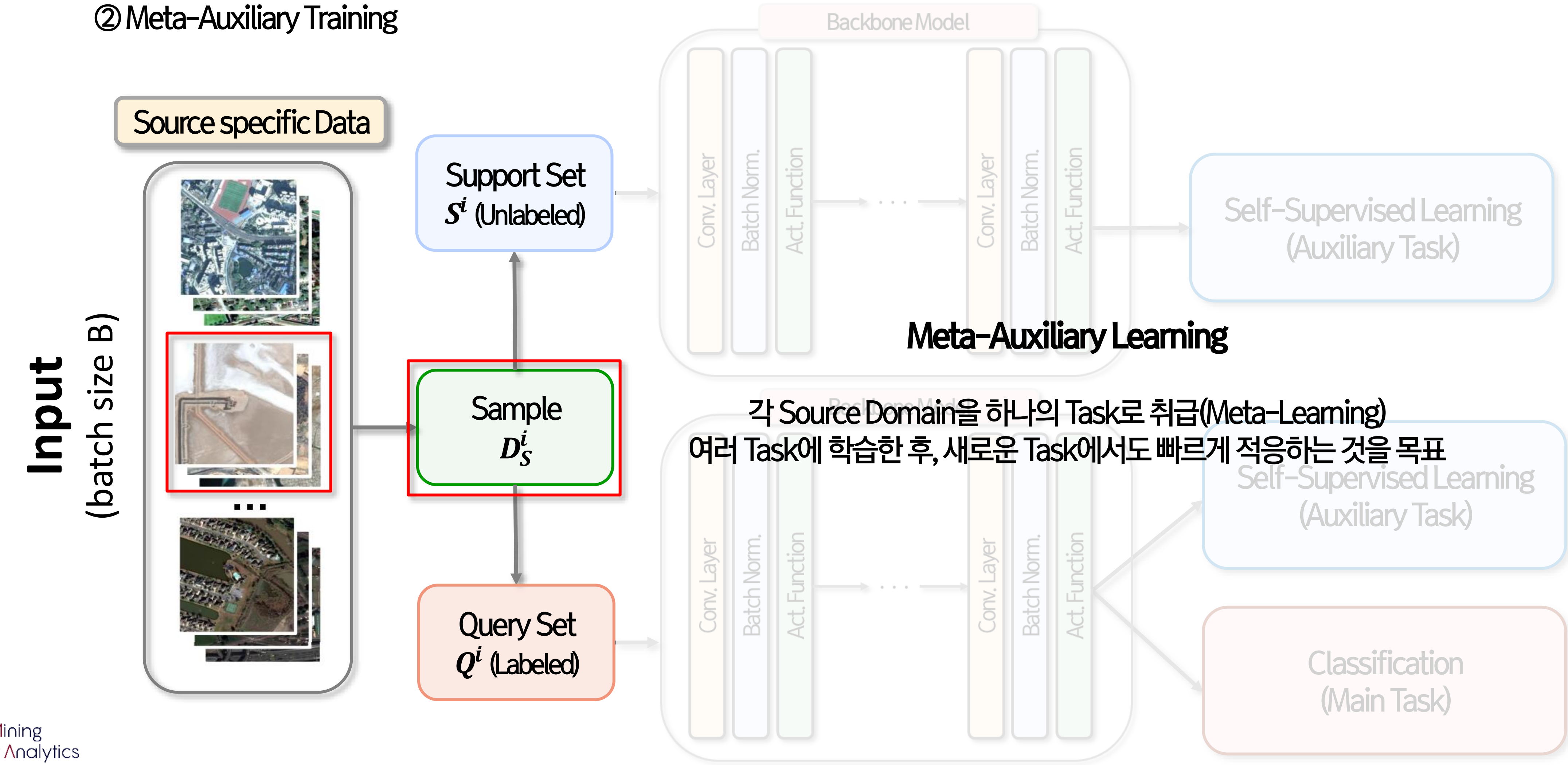
② Meta-Auxiliary Training



Methods

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

② Meta-Auxiliary Training



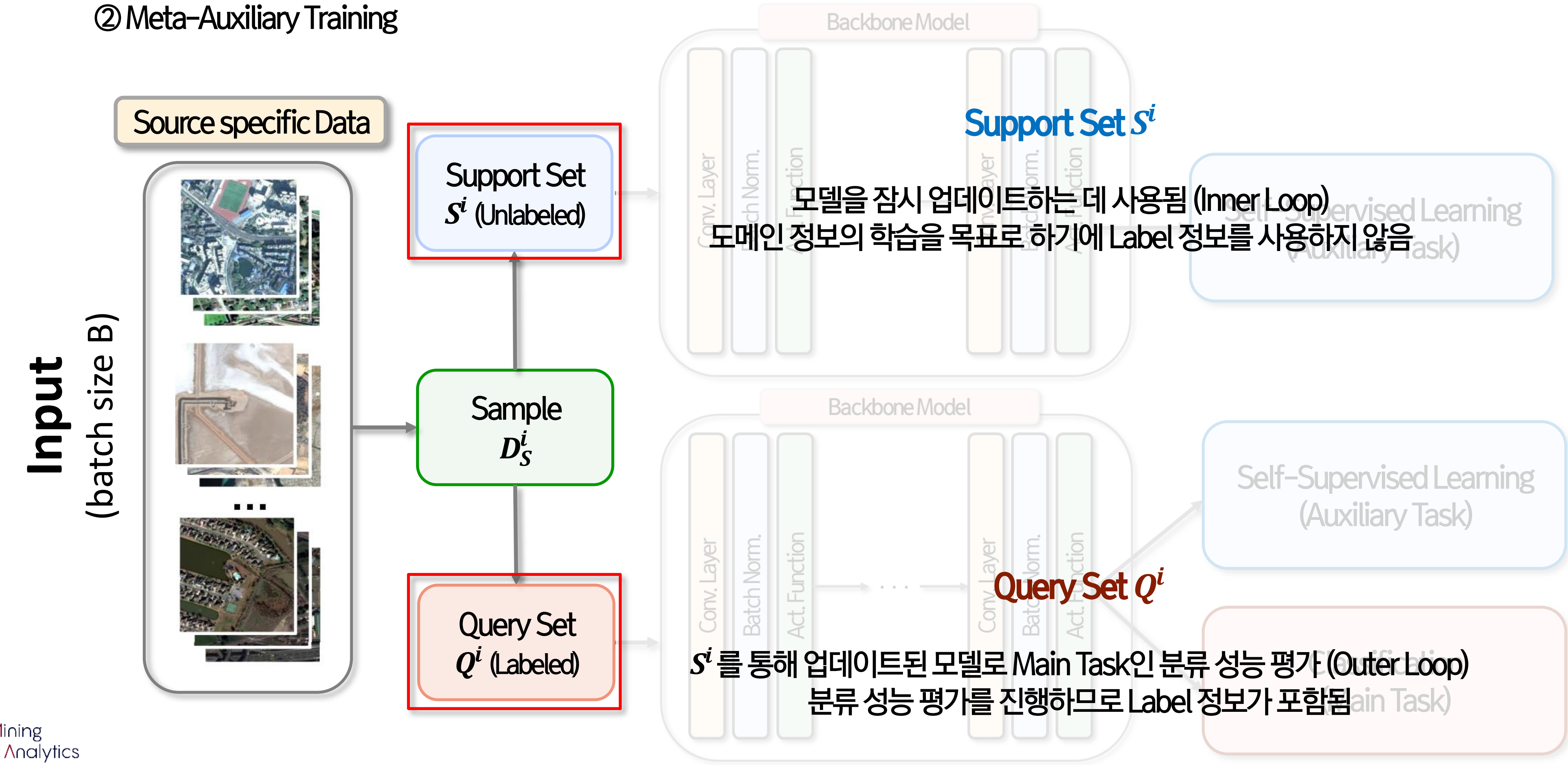
Meta-Auxiliary Learning

각 Source Domain을 하나의 Task로 취급(Meta-Learning)
여러 Task에 학습한 후, 새로운 Task에서도 빠르게 적응하는 것을 목표

Methods

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

② Meta-Auxiliary Training



Support Set S^i

모델을 잠시 업데이트하는 데 사용됨 (Inner Loop)
도메인 정보의 학습을 목표로 하기에 Label 정보를 사용하지 않음

Query Set Q^i

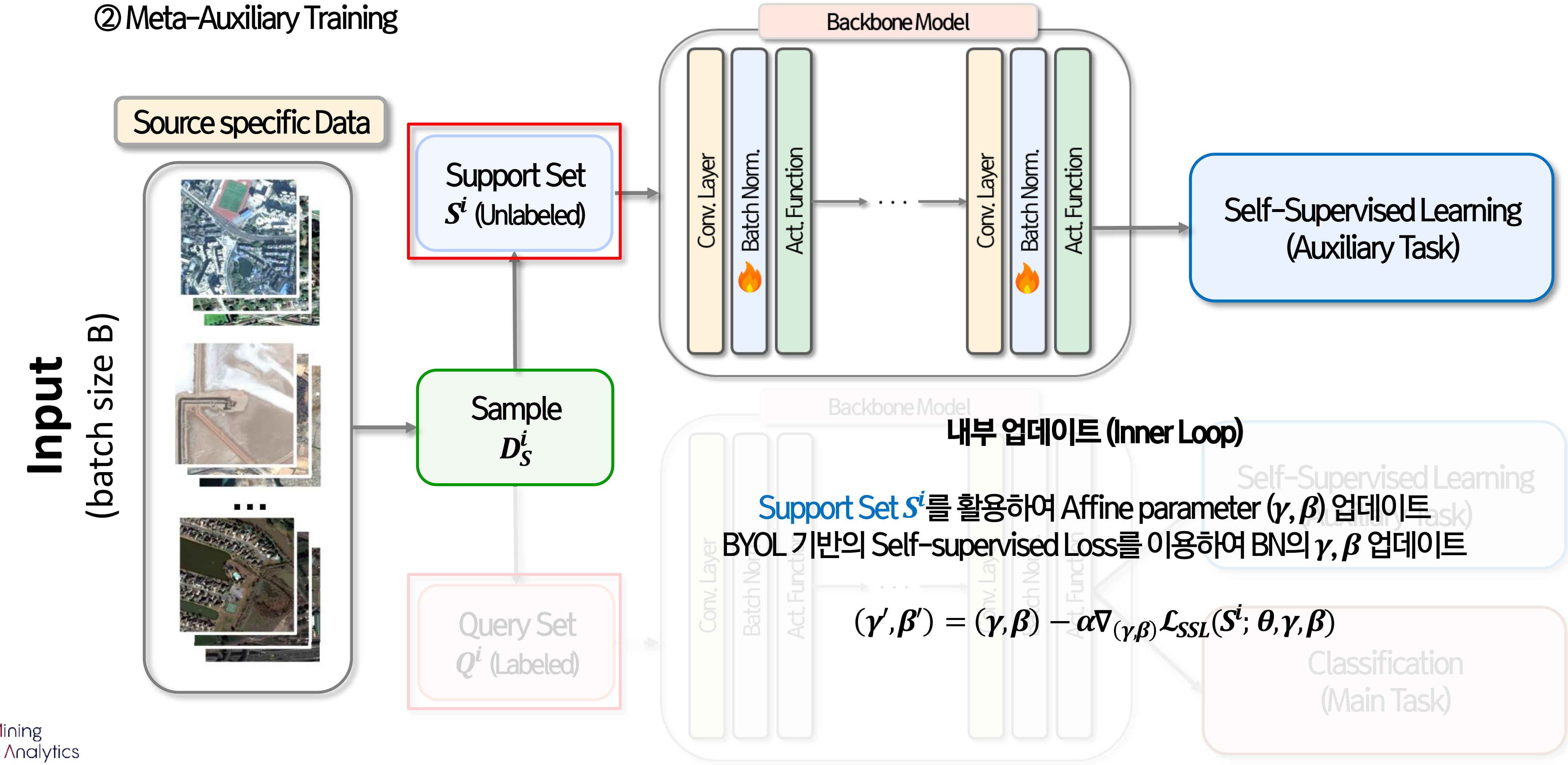
S^i 를 통해 업데이트된 모델로 Main Task인 분류 성능 평가 (Outer Loop)
분류 성능 평가를 진행하므로 Label 정보가 포함됨

Methods

Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

② Meta-Auxiliary Training



내부 업데이트 (Inner Loop)

Support Set S^i를 활용하여 Affine parameter (γ, β) 업데이트

BYOL 기반의 Self-supervised Loss를 이용하여 BN의 γ, β 업데이트

$$(\gamma', \beta') = (\gamma, \beta) - \alpha \nabla_{(\gamma,\beta)} \mathcal{L}_{SSL}(S^i; \theta, \gamma, \beta)$$

Methods

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

② Meta-Auxiliary Training

외부 업데이트 (Outer Loop)

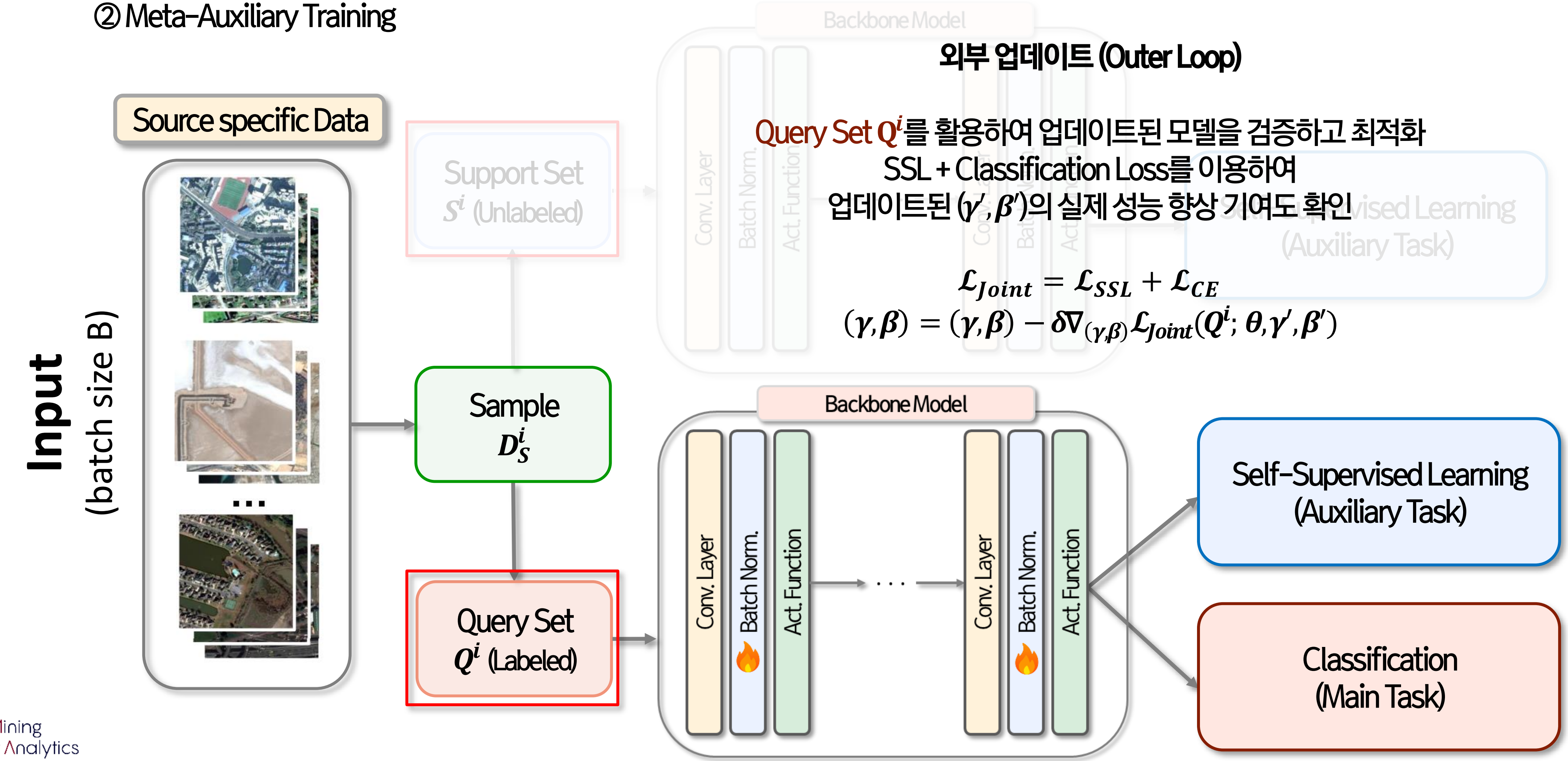
Query Set Q^i를 활용하여 업데이트된 모델을 검증하고 최적화

SSL + Classification Loss를 이용하여

업데이트된 (γ', β')의 실제 성능 향상 기여도 확인

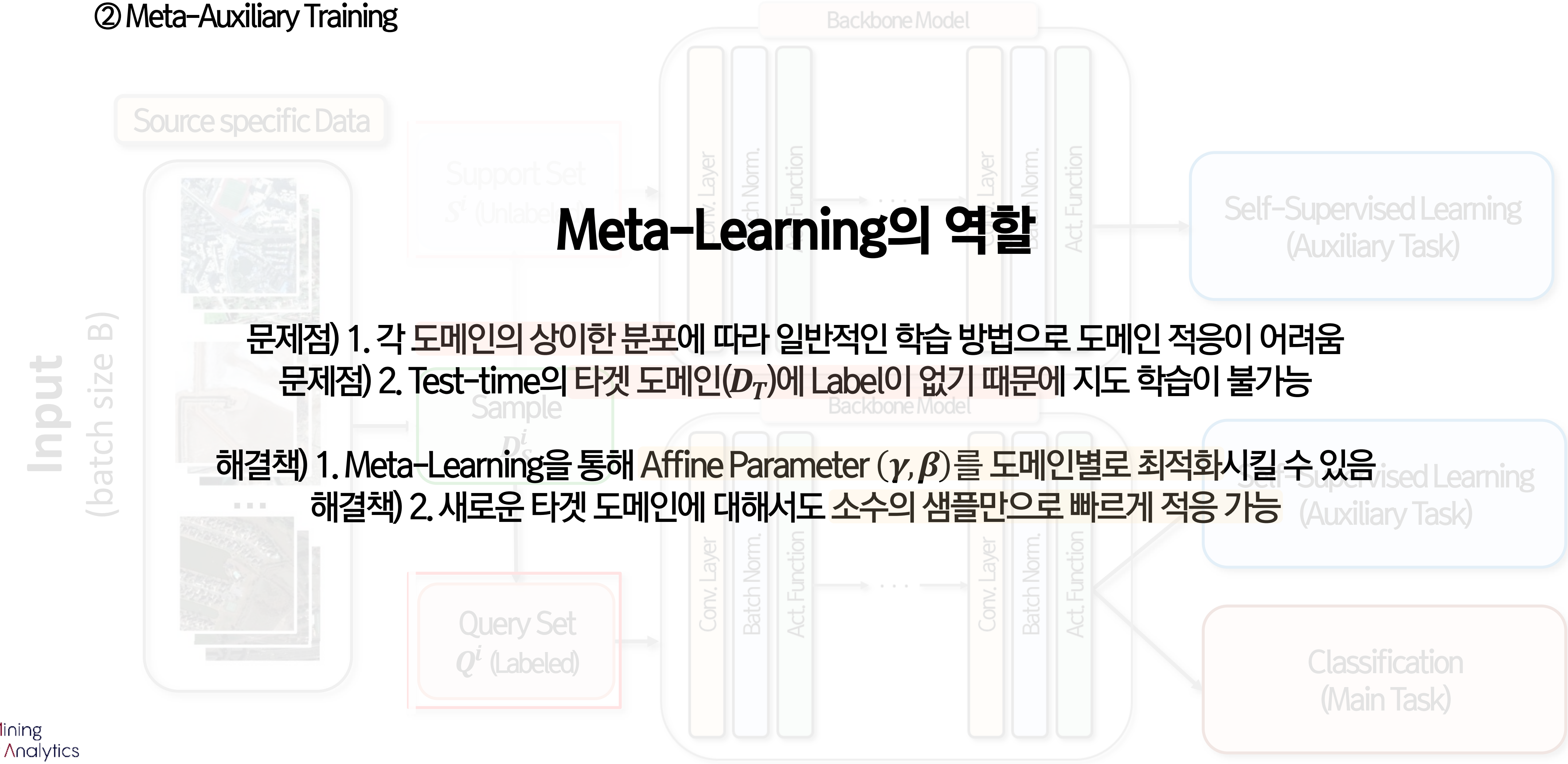
$$\mathcal{L}_{Joint} = \mathcal{L}_{SSL} + \mathcal{L}_{CE}$$

$$(\gamma, \beta) = (\gamma, \beta) - \delta\nabla_{(\gamma,\beta)}\mathcal{L}_{Joint}(Q^i; \theta, \gamma', \beta')$$



❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

② Meta-Auxiliary Training



Meta-Learning의 역할

문제점) 1. 각 도메인의 상이한 분포에 따라 일반적인 학습 방법으로 도메인 적응이 어려움

문제점) 2. Test-time의 타겟 도메인(D_T)에 Label이 없기 때문에 지도 학습이 불가능

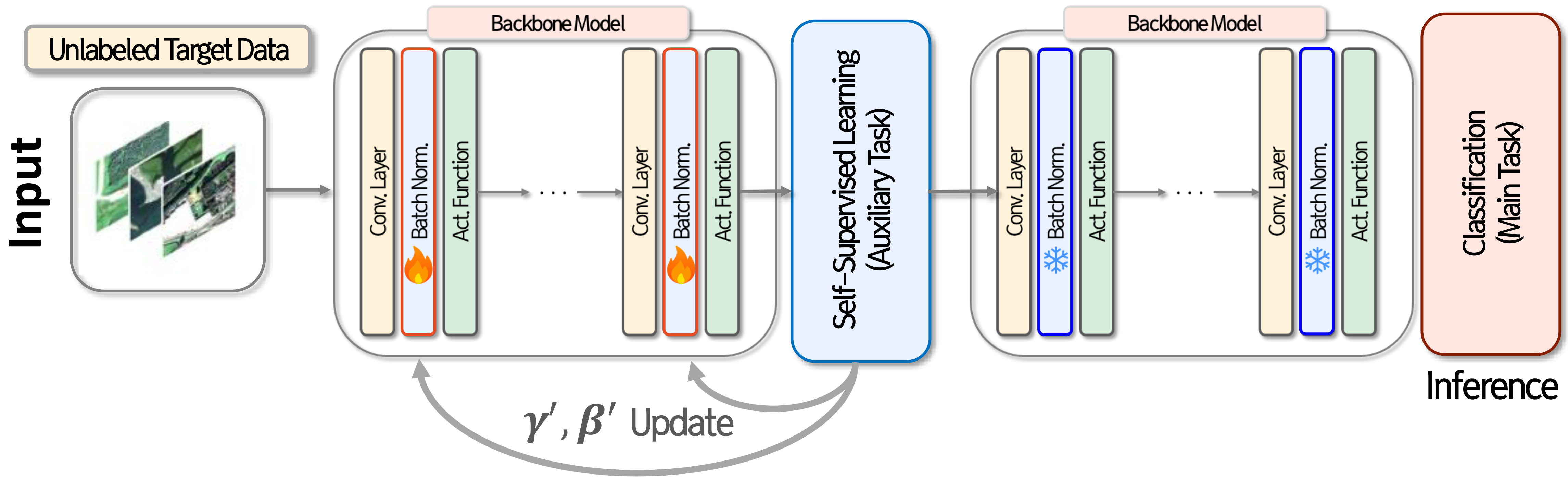
해결책) 1. Meta-Learning을 통해 Affine Parameter (γ, β)를 도메인별로 최적화시킬 수 있음

해결책) 2. 새로운 타겟 도메인에 대해서도 소수의 샘플만으로 빠르게 적응 가능

Methods

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

③ Test-Time Adaptation



Experiments

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

- WILDS 벤치마크 데이터셋에 대한 Out-of-Distribution(OOD)의 일반화 성능에 대한 실험
- WILDS 벤치마크 데이터셋 내 다섯가지 각각의 도메인에 대해 높은 분류 성능을 보임
  - ➢ Domain-aware한 Meta-auxiliary learning의 효과

Methods	iWildCam		Camelyon17	RxRx1	FMoW		PovertyMap	
	Acc	Macro F1	Acc	Acc	WC Acc	Avg Acc	WC Pearson r	Pearson r
ERM	71.6±2.5	31.0±1.3	70.3±6.4	29.9±0.4	32.3±1.25	53.0±0.55	0.45±0.06	0.78±0.04
CORAL	73.3±4.3	32.8±0.1	59.5±7.7	28.4±0.3	31.7±1.24	50.5±0.36	0.44±0.06	0.78±0.05
Group DRO	72.7±2.1	23.9±2.0	68.4±7.3	23.0±0.3	30.8±0.81	52.1±0.5	0.39±0.06	0.75±0.07
IRM	59.8±3.7	15.1±4.9	64.2±8.1	8.2±1.1	30.0±1.37	50.8±0.13	0.43±0.07	0.77±0.05
ARM-CML	70.5±0.6	28.6±0.1	84.2±1.4	17.3±1.8	27.2±0.38	45.7±0.28	0.37±0.08	0.75±0.04
ARM-BN	70.3±2.4	23.7±2.7	87.2±0.9	31.2±0.1	24.6±0.04	42.0±0.21	0.49±0.21	**0.84±0.05**
ARM-LL	71.4±0.6	27.4±0.8	84.2±2.6	24.3±0.3	22.1±0.46	42.7±0.71	0.41±0.04	0.76±0.04
Meta-DMoE	77.2±0.3	34.0±0.6	91.4±1.5	29.8±0.4	35.4±0.58	52.5±0.18	0.51±0.04	0.80±0.03
PAIR	74.9±1.1	27.9±0.9	74.0±7.2	28.8±0.0	35.4±1.30	-	0.47±0.09	-
MABN (ours)	**78.4±0.6**	**38.3±1.2**	**92.4±1.9**	**32.7±0.2**	**36.6±0.41**	**53.2±0.52**	**0.56±0.05**	**0.84±0.04**

Experiments

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

- 학습된 γ', β'이 도메인 지식을 잘 학습했는지 확인하는 실험
  - ➢ No adapt : 별다른 적응 학습 없이 Affine Parameter 그대로 사용
  - ➢ Not matched : 다른 도메인의 Parameter를 Random하게 사용
  - ➢ Matched : 해당 도메인의 Parameter를 적절하게 사용

Adapted $(\tilde{\gamma}, \tilde{\beta})$	**No adapt**	**Not-matched**	**Matched**
Accuracy	74.69	72.39	**78.40**
Macro-F1	36.77	33.32	**38.27**

❖ Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization [5]

- Ablation Study : 제안 방법론의 주요 구성 요소(SSL, Affine Parameter Update, Meta-auxiliary training) 영향 분석
  - SSL : BYOL 적용을 통한 $\mathcal{L}_{SSL}$ 적용 여부
  - Param. : 전체 파라미터(All), Batch Normalization 전체 파라미터(BN), Affine 파라미터(Aff)만 업데이트
  - TS : Main Task와 Auxiliary Task를 동시에(Joint) 학습하는 방식과 메타러닝 기반의 학습(Meta)
  - Adapt : Test-Time Adaptation 적용 여부

Index	*SSL*	*Param.*	*TS*	*Adapt*	iWildCam Acc	iWildCam F1
1	✗	All	CE	✗	68.7	31.3
2	✓	All	Joint	✗	70.5	33.2
3	✓	BN	Joint	✓	68.2	30.5
4	✓	Aff	Joint	✓	71.1	33.9
5	✓	All	Meta	✓	72.0	29.4
6	✓	Aff	Meta	✗	74.7	36.8
7	✓	Aff	Meta	✓	78.4	38.3

Conclusions

❖ Test-Time Adaptation 의 필요성과 종류

- 테스트 시점에 데이터 분포가 변화하는 상황에 적용할 수 있는 강건한 일반화 성능을 갖춘 Adaptation Model이 필요
- Test-time Adaptation의 분류는 활용하는 방식에 따라 크게 1) Self-training, 2) Self-supervised, 3) Batch Normalization로 나뉨

1) Self-training strategy – Improved Self-Training for Test-Time Adaptation

- 비교적 간단한 데이터 증강 방식(SimpleAug)의 유효성을 입증함
- 자체적인 Pseudo-labeling 정제 방식인 Pseudo-Labeling Correction Algorithm(PLCA)을 제안

2) Self-supervised strategy – SwapPrompt: Test-Time Prompt Adaptation for Vision-Language Models

- 일반적인 Prompt를 Online & Target Prompt로 분할하여 입력되는 데이터마다의 특징을 Online으로 학습
- 강건한 Contrastive Learning 기반의 Swapped Prediction 방법 제안

3) Batch Normalization strategy – Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

- Domain-aware한 Meta-learning 기반으로 도메인 마다 최적화된 파라미터 학습을 가능하게 함
- 도메인 정보를 학습하는 Auxiliary Task과 분류/회귀의 Main Task를 접목하여 높은 예측 성능을 달성

Reference

References

[1] Ma, J. (2024). Improved Self-Training for Test-Time Adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 23701-23710).

[2] Ma, X., Zhang, J., Guo, S., & Xu, W. (2024). Swapprompt: Test-time prompt adaptation for vision-language models. Advances in Neural Information Processing Systems, 36.

[3] Zhou, K., Yang, J., Loy, C. C., & Liu, Z. (2022). Learning to prompt for vision-language models. International Journal of Computer Vision, 130(9), 2337-2348.

[4] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, Advances in Neural Information Processing Systems, 2022.

[5] Wu, Y., Chi, Z., Wang, Y., Plataniotis, K. N., & Feng, S. (2024, March). Test-time domain adaptation by learning domain-aware batch normalization. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 14, pp. 15961-15969).

Thank You