2022 하계 세미나

Multi-Target Domain Adaptation

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented by
정성훈

Outline

- Background
  - Domain Adaptation

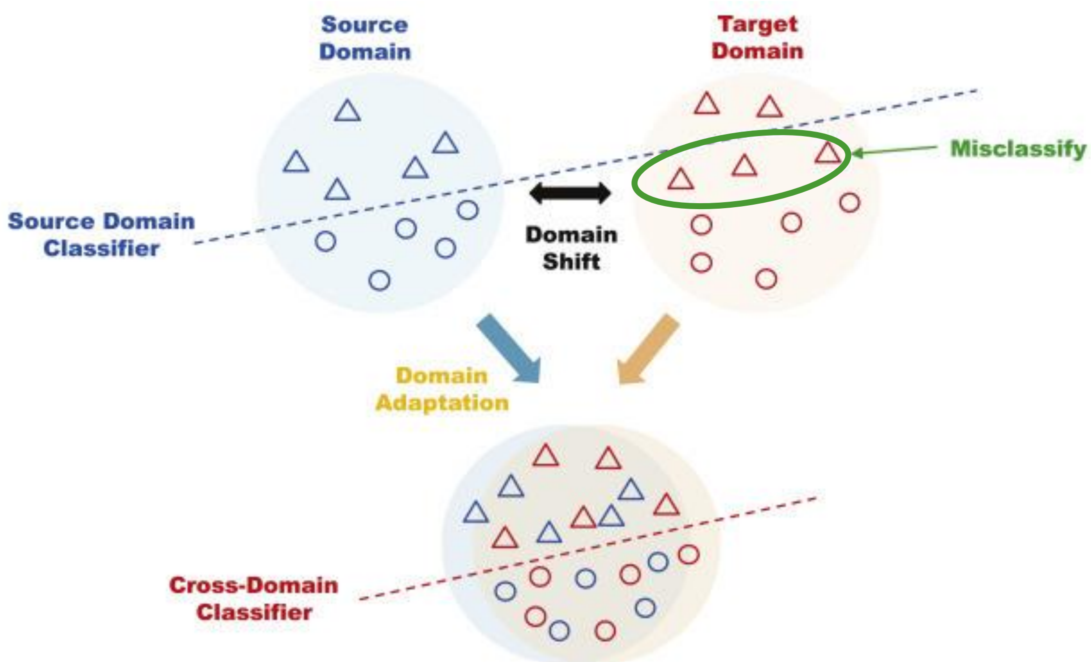
- Multi-Target Domain Adaptation
  - General methods
  - Multi-Target Adversarial Frameworks for Domain Adaptation in Semantic Segmentation (ICCV 2021)
  - ADAS: A Direct Adaptation Strategy for Multi-Target Domain Adaptive Semantic Segmentation (CVPR 2022)

Background

- Domain Adaptation
  - Why was the Domain Adaptation created?
    - Limitation
      - Model training을 위해서 기본적으로 training data / labeled data가 필요함
      - Labeling에 대한 cost가 비싸거나 training data를 만드는 것이 불가능한 경우가 존재함
    - Transfer learning
      - 이미 알고 있는 지식을 이용해서 새로운 상황을 학습시킴
      - Transfer learning 기법 중 domain adaptation이 존재

Background

- Domain Adaptation
  - What is Domain Adaptation
    - Source domain으로 pretrained된 decision boundary는 target domain에 대해서 유효하지 못함
    - Source domain으로 학습된 네트워크가 target domain에서도 유효하게 사용 가능하도록 학습시키는 방법



Background

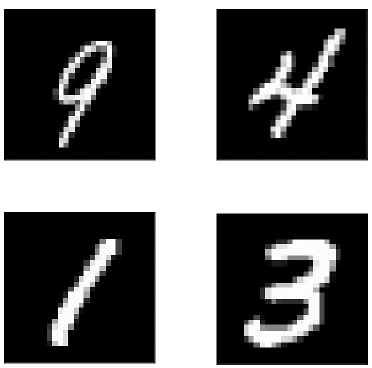
- Domain Adaptation
  - What is Domain Adaptation
    - Source domain으로 pretrained된 decision boundary는 target domain에 대해서 유효하지 못함
    - Source domain으로 학습된 네트워크가 target domain에서도 유효하게 사용 가능하도록 학습시키는 방법

Source domain

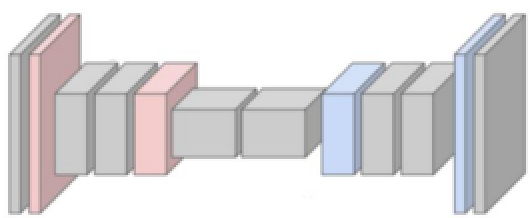
Target domain

Background

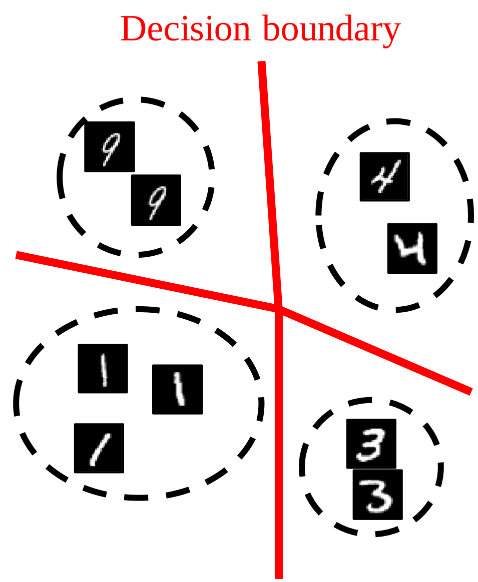
- Domain Adaptation
  - What is Domain Adaptation
    - Source domain으로 pretrained된 decision boundary는 target domain에 대해서 유효하지 못함
    - Source domain으로 학습된 네트워크가 target domain에서도 유효하게 사용 가능하도록 학습시키는 방법



MNIST
Source domain



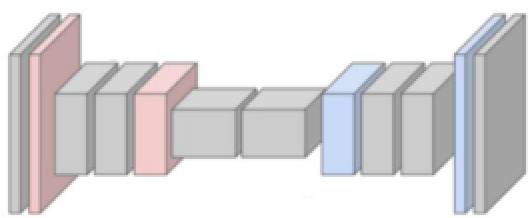
Pre-trained network
with Source domain



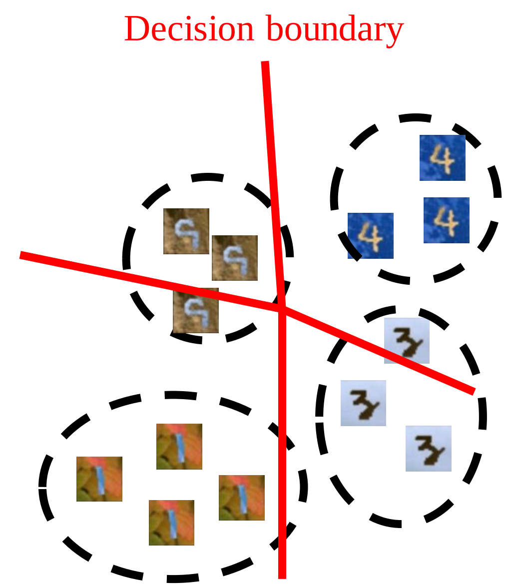
Background

- Domain Adaptation
  - What is Domain Adaptation
    - Source domain으로 pretrained된 decision boundary는 target domain에 대해서 유효하지 못함
    - Source domain으로 학습된 네트워크가 target domain에서도 유효하게 사용 가능하도록 학습시키는 방법

MNIST-M
Target domain

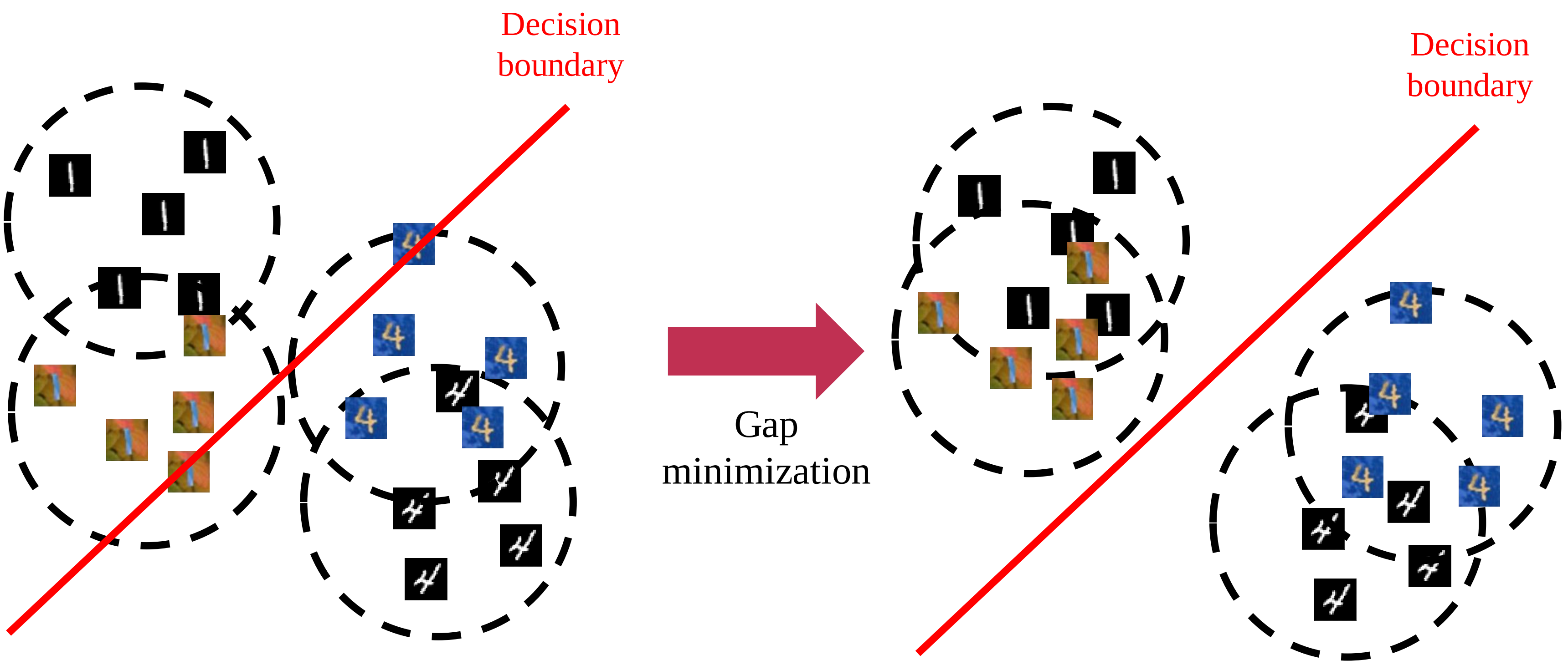


Pre-trained network
with Source domain



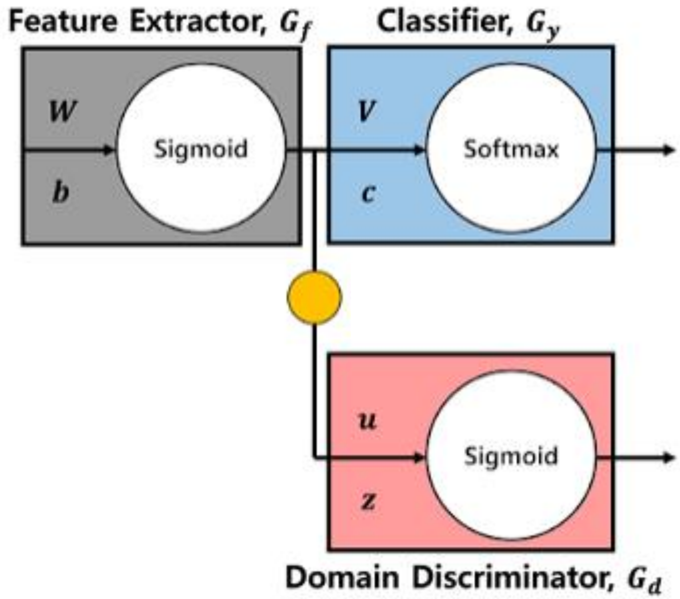
Background

- Domain Adaptation
  - Gap minimization
    - Source domain과 target domain과의 domain gap을 줄이는 방법



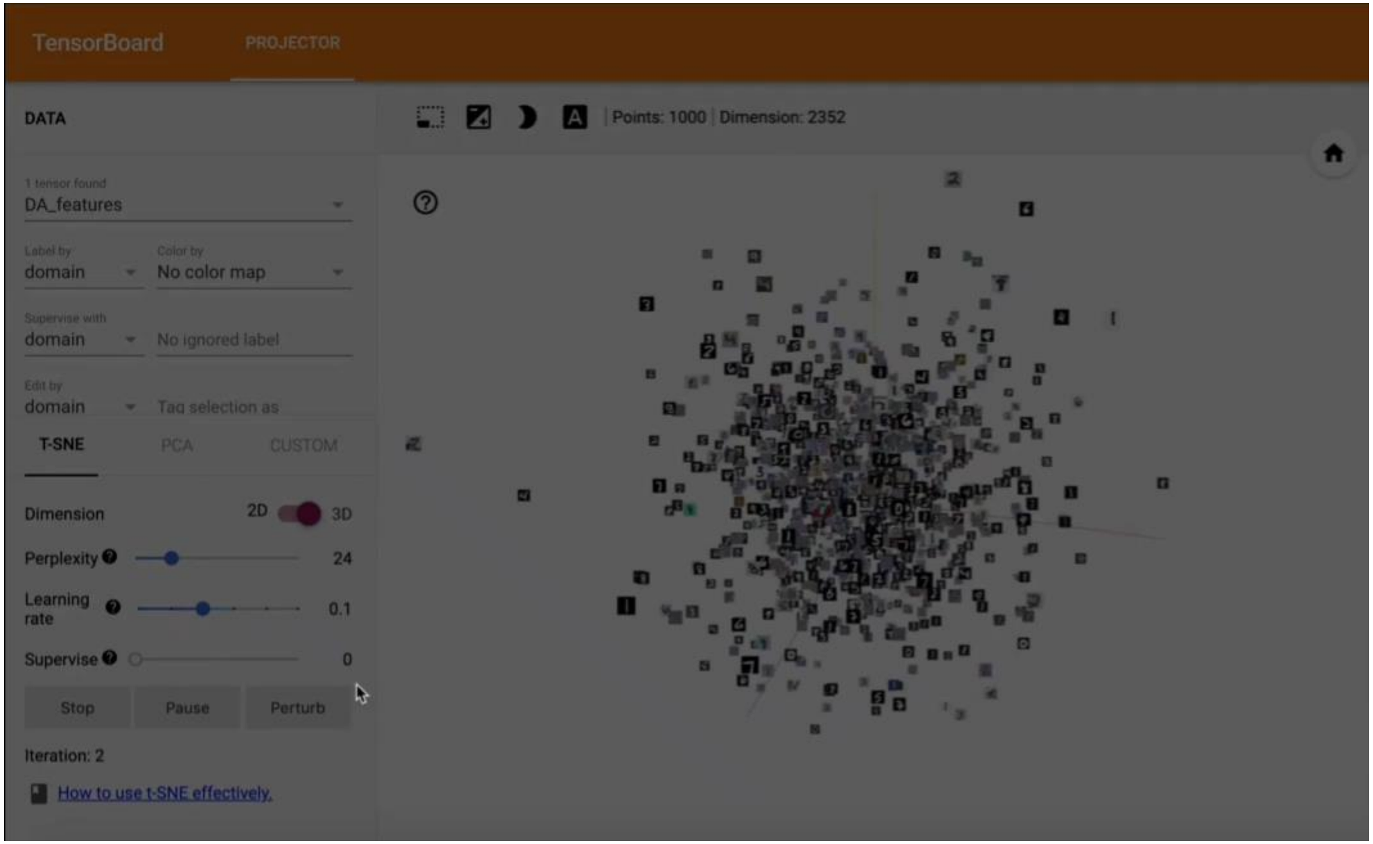
Background

- Domain Adaptation
  - Gap minimization
    - Adversarial training
      - Source dataset에 대한 classification error는 최소화
      - Target과 source를 구분하는 domain discrimination의 성능 저하시켜 domain에 상관없이 robust한 결과를 나타내도록 학습



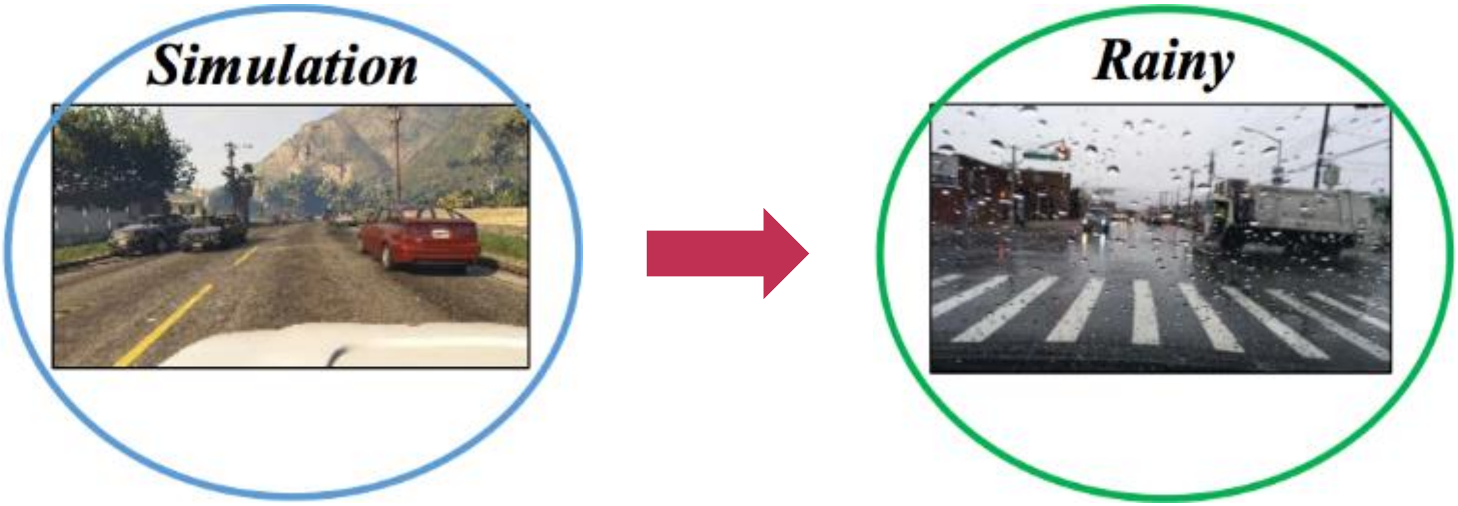
Background

- Domain Adaptation
  - Visualization



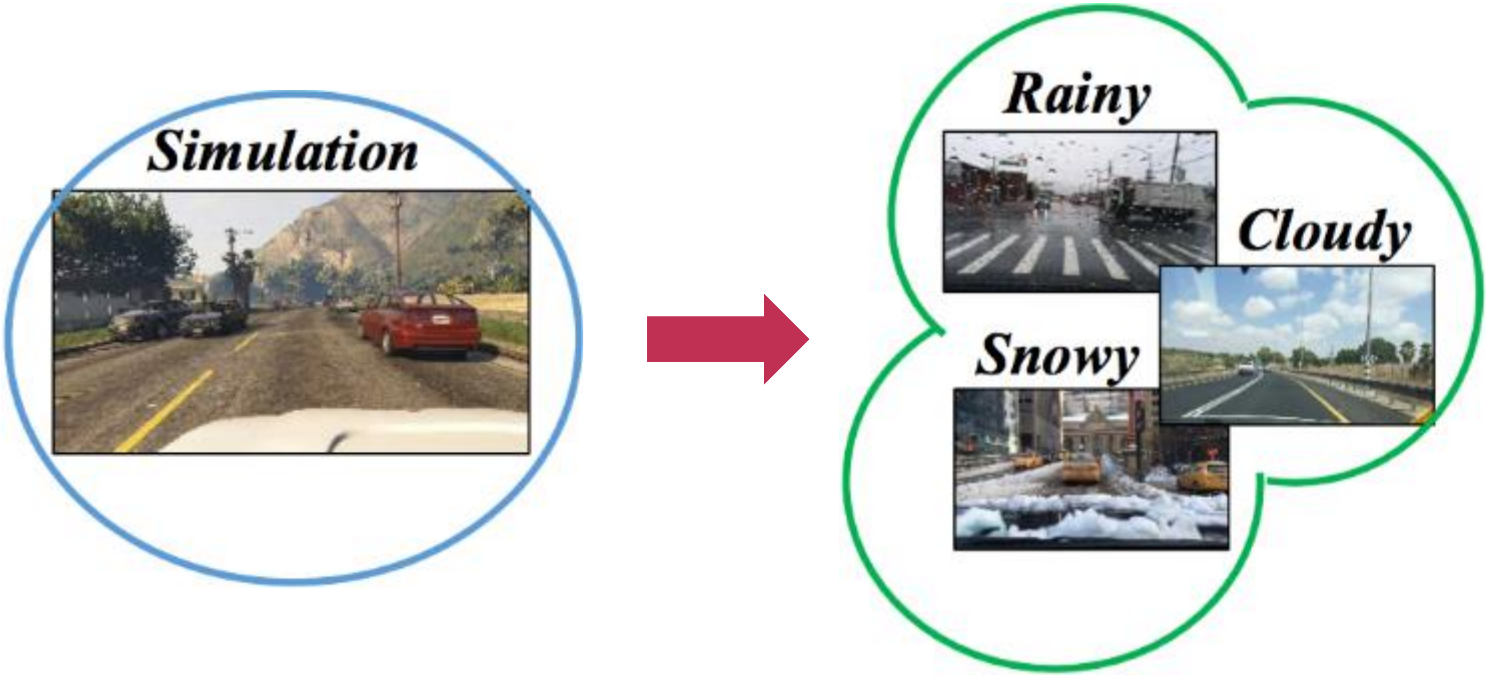
Multi-Target Domain Adaptation

- Single-Target Domain Adaptation vs. Multi-Target Domain Adaptation
  - Single-Target Domain Adaptation
    - 1개의 source domain으로 학습된 네트워크가 1개의 target domain에서도 유효하게 사용 가능하도록 학습시키는 방법
    - 다양한 상황이 존재하는 real-world 조건에서 적용 불가능



Multi-Target Domain Adaptation

- Single-Target Domain Adaptation vs. Multi-Target Domain Adaptation
  - Multi-Target Domain Adaptation
    - 1개의 source domain으로 학습된 네트워크가 N개의 target domain에서도 유효하게 사용 가능하도록 학습시키는 방법

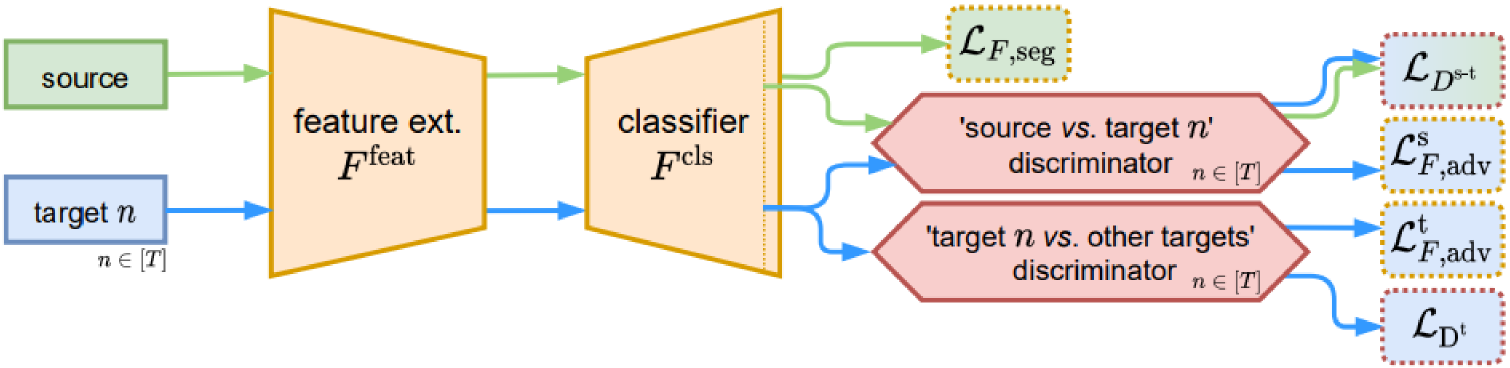


Multi-Target Domain Adaptation

- General Methods
  - Training multiple single target domain adaptation network
    - N개의 target domain에 대해 N개의 single-target domain adaptation model을 학습하여 1개의 multi-target domain adaptation model으로 knowledge distillation 수행
      - Pretrained single target domain adaptation model의 성능에 따라 model 성능 제한
        - ✓각각의 Single target domain adaptation model의 label predictions이 부정확할수록 model performance가 저하됨
  - Merging target domain
    - N개의 target data에 대해 single target으로 합쳐서 single-target domain adaptation model을 학습시키는 방법
      - Target domain 간의 distribution shifts를 고려하지 않음
        - ✓Source-target 간의 distribution shift도 존재하지만 target 간 다른 dataset을 사용하므로 target-target distribution shift도 존재함

Multi-Target Domain Adaptation

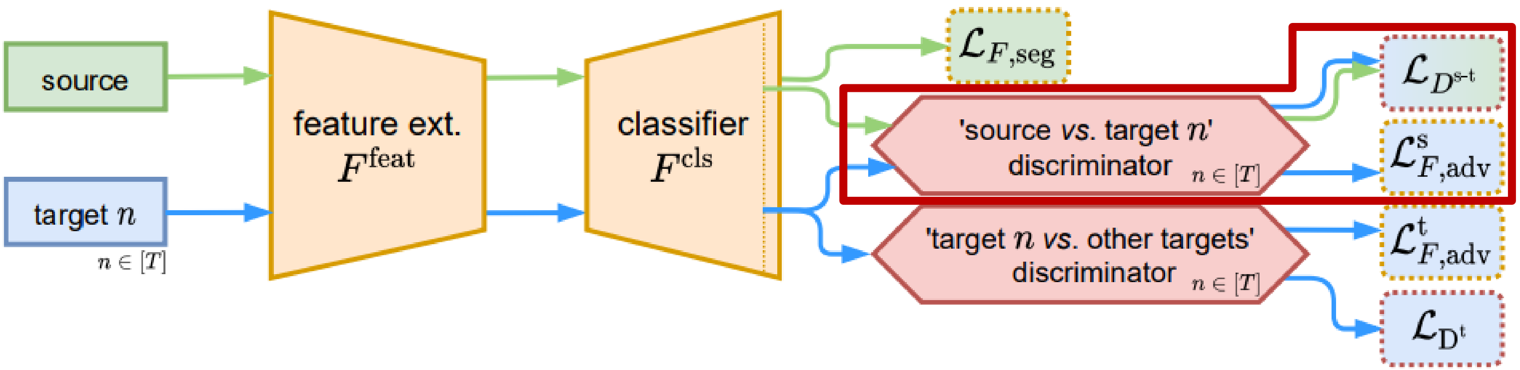
- MTAF
  - Multi-Discriminator



- Merging target domain에 기반한 framework
- Two types의 discriminator를 통해 alignment 수행
  - Source-target discriminator : source와 target 간의 distribution align 수행
  - Target-target discriminator : target간의 distribution align 수행

Multi-Target Domain Adaptation

- MTAF
  - Multi-Discriminator



- Source-target adversarial alignment

$$\mathcal{L}_{D_n^{s\text{-}t}}(\phi) = \langle \mathcal{L}_{BCE}(D(\mathbf{Q_x}), 1)\rangle_{\mathcal{X}_s} + \langle \mathcal{L}_{BCE}(D(\mathbf{Q_x}), 0)\rangle_{\mathcal{X}_{t,n}} \quad \mathcal{L}_{D^{\text{s-t}}}(\phi^{\text{s-t}}_{1:T}) = \frac{1}{T}\sum_{n\in[T]} \mathcal{L}_{D_n^{\text{s-t}}}(\phi_n^{\text{s-t}})$$

※ Source set의 경우 class 1(source), target set의 경우 class 0(target)으로 분류하도록 학습하여 source set으로부터 target set을 discriminate

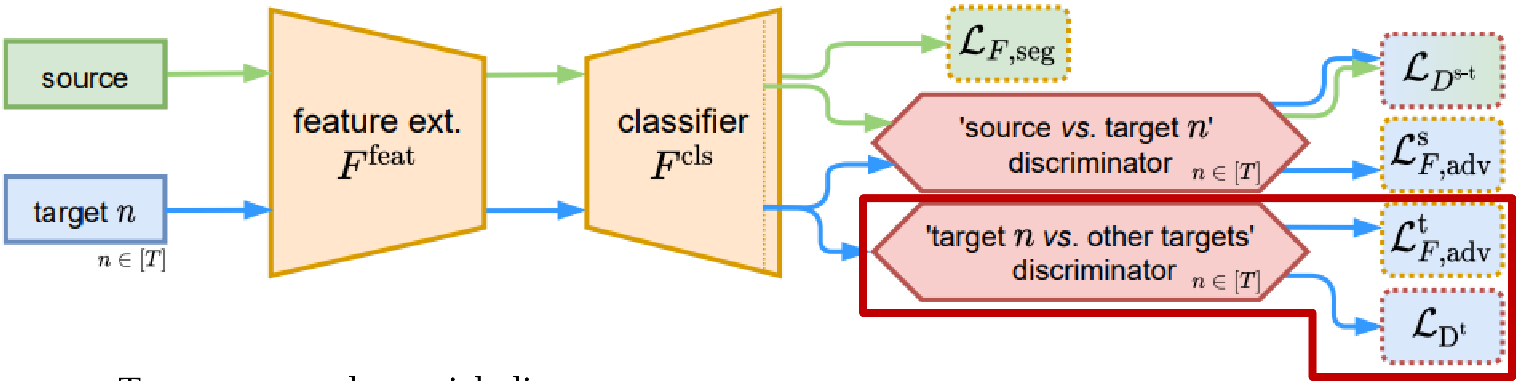
$$\mathcal{L}^{\text{s}}_{F,\text{adv}}(\theta) = \frac{1}{T}\sum_{n\in[T]} \left\langle \mathcal{L}_{\text{BCE}}(D_n^{\text{s-t}}(\mathbf{Q_x}), 1)\right\rangle_{\mathcal{X}_{t,n}}$$

※ 각 target set에 대한 segmenter output을 class 1(source)로 분류

※ Target dataset이 source dataset으로 distribution alignment 수행

Multi-Target Domain Adaptation

- MTAF
  - Multi-Discriminator



– Target-target adversarial alignment

$$\mathcal{L}_{D^{\text{t}}_n}(\phi^{\text{t}}_n) = \left\langle \mathcal{L}_{\text{BCE}}(D^{\text{t}}_n(\mathbf{Q}_{\mathbf{x}}), 1) \right\rangle_{\mathcal{X}_{\text{t},n}} + \left\langle \mathcal{L}_{\text{BCE}}(D^{\text{t}}_n(\mathbf{Q}_{\mathbf{x}}), 0) \right\rangle_{\bigcup_{k \neq n} \mathcal{X}_{\text{t},k}} \qquad \mathcal{L}_{D^{\text{t}}}(\phi^{\text{t}}_{1:T}) = \frac{1}{T} \sum_{n \in [T]} \mathcal{L}_{D^{\text{t}}_n}(\phi^{\text{t}}_n)$$

※ N번째 target set은 class 1로 분류, 다른 target set은 class 0으로 분류하도록 학습시킴

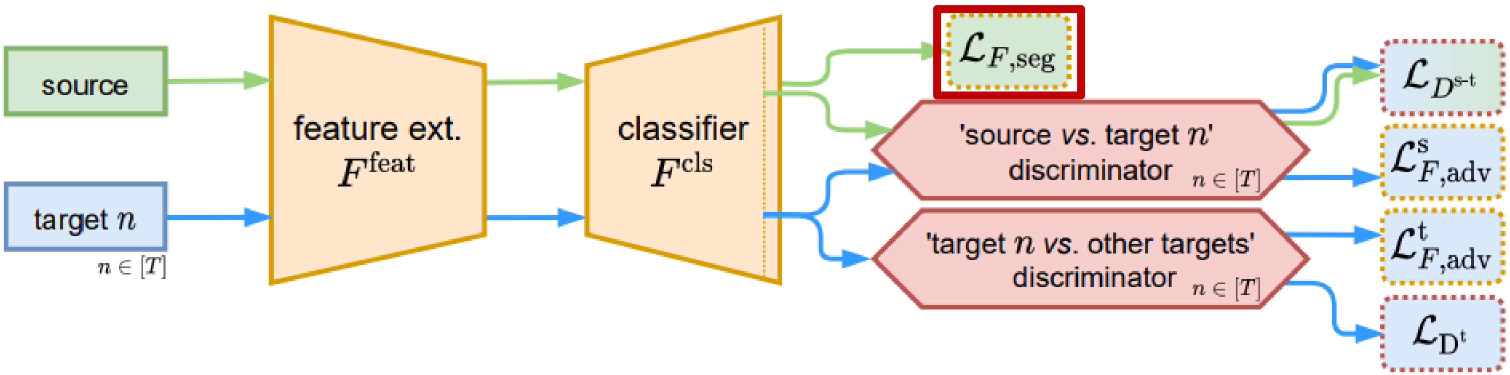
$$\mathcal{L}^{\text{t}}_{F,\text{adv}}(\theta) = \frac{1}{T} \sum_{n \in [T]} \left\langle \mathcal{L}_{\text{BCE}}(D^{\text{t}}_n(\mathbf{Q}_{\mathbf{x}}), 1) \right\rangle_{\bigcup_{k \neq n} \mathcal{X}_{\text{t},k}}$$

※ N번째가 아닌 target set의 segmenter output을 class 1로 분류하도록 학습시킴

※ Target dataset간의 distribution alignment 수행

Multi-Target Domain Adaptation

- MTAF
  - Multi-Discriminator



- Segmenter

$$\mathcal{L}_{F,\text{seg}}(\theta) = \langle \mathcal{L}_{\text{CE}}(\mathbf{x}, \mathbf{y}) \rangle_{\mathcal{X}_s}$$

※ Source dataset의 prediction이 GT label image와 같아지도록 segmenter를 학습시킴

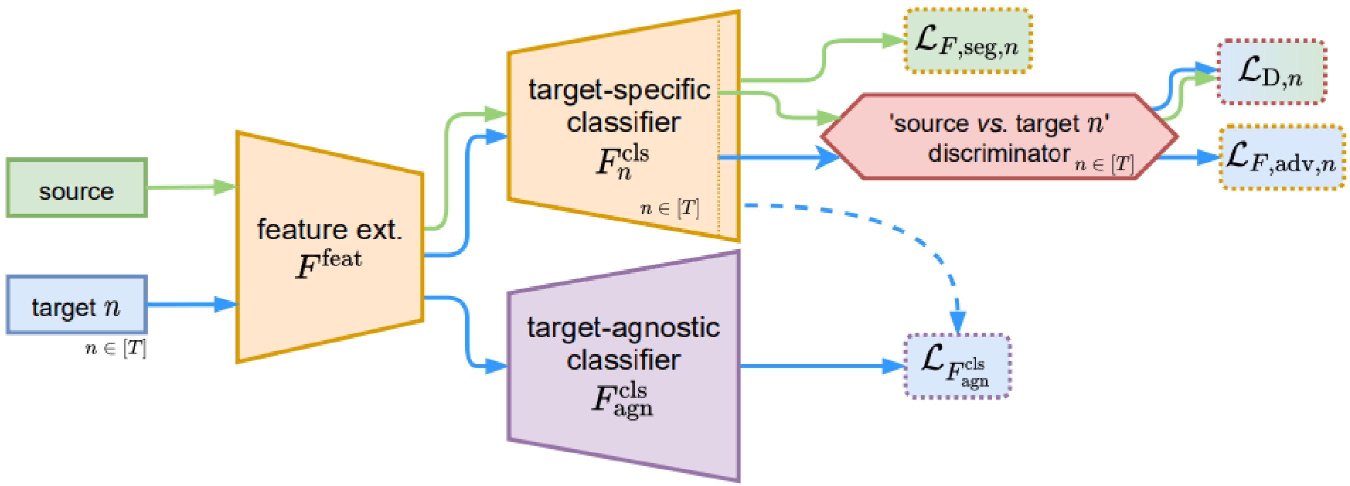
$$\mathcal{L}_F = \mathcal{L}_{F,\text{seg}} + \lambda^{\text{s}}_{\text{adv}} \mathcal{L}^{\text{s}}_{F,\text{adv}} + \lambda^{\text{t}}_{\text{adv}} \mathcal{L}^{\text{t}}_{F,\text{adv}}$$

※ Target datasets을 source로 구별하도록 discriminator를 학습함과 동시에 source dataset에 대한 정확도를 높여 multi target에 대한 segmenter의 성능 향상

※ 기존의 Merging target domain의 target간의 distribution shifts 문제를 해결

Multi-Target Domain Adaptation

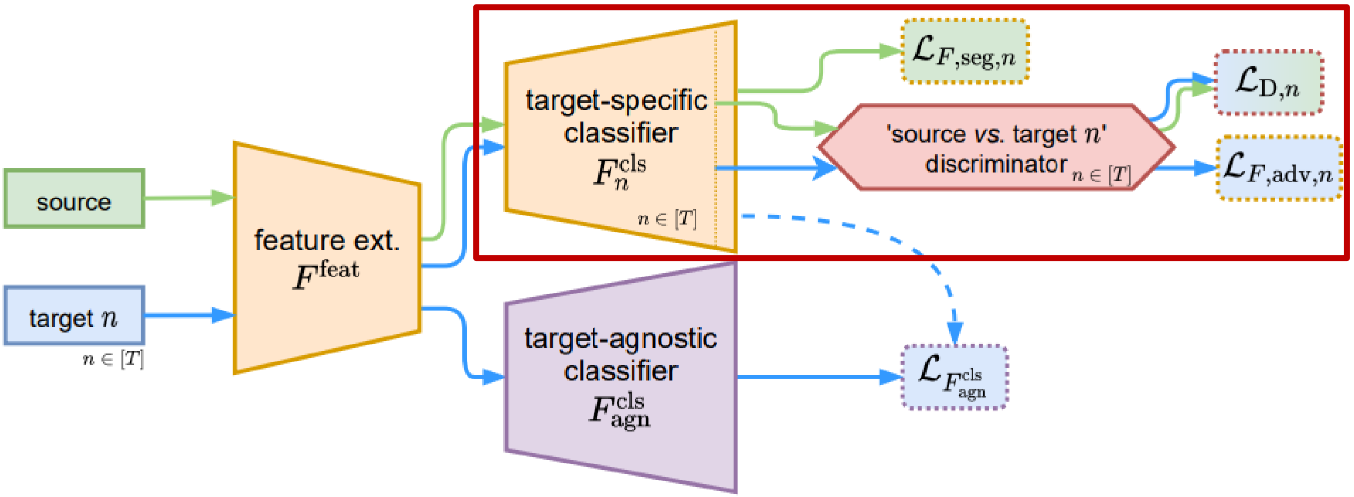
- MTAF
  - Multi-Target Knowledge Transfer



- Training multiple single target domain adaptation network에 기반한 framework
  - Adversarial training의 instability를 main challenge로 설정
  - 여러 개의 adversarial loss가 joint minimize되면 target prediction이 noisy한 training phase 초기 단계에서 instability의 문제가 발생함
  - Target간의 direct alignment를 수행하지 않고 knowledge distillation을 이용함으로써 문제점 해결

Multi-Target Domain Adaptation

- MTAF
  - Multi-Target Knowledge Transfer



– Target-specific classifier

$$\mathcal{L}_D(\phi) = \langle \mathcal{L}_{\text{BCE}}(D(\mathbf{Q_x}), 1) \rangle_{\mathcal{X}_s} + \langle \mathcal{L}_{\text{BCE}}(D(\mathbf{Q_x}), 0) \rangle_{\mathcal{X}_t}$$

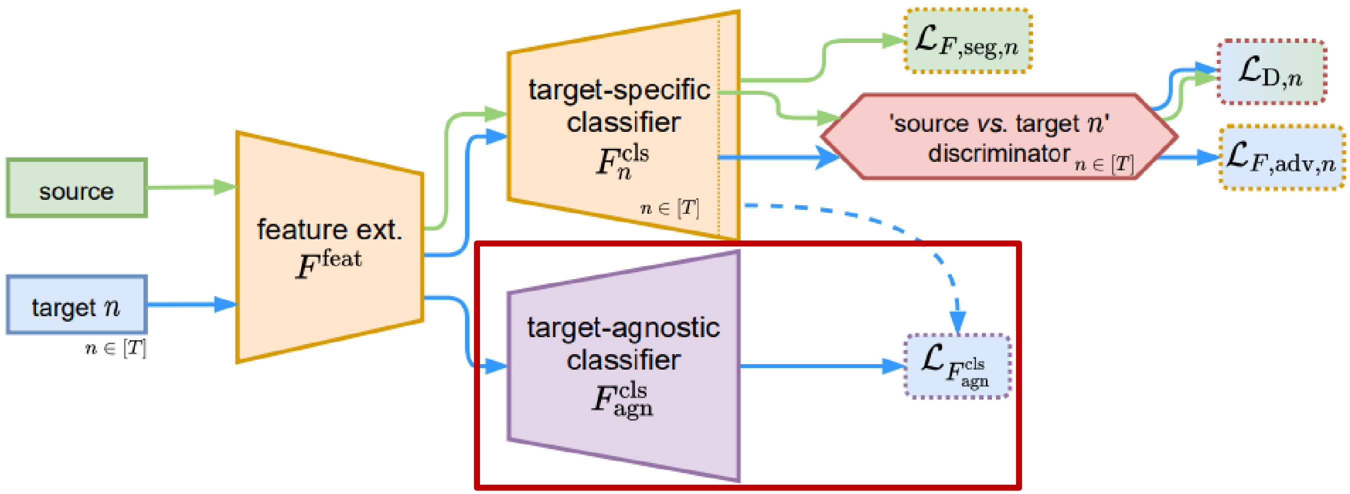
※ Source set의 경우 class 1 (source), N번째 target set의 경우 class 0 (target)으로 분류하도록 학습 진행

$$\mathcal{L}_F(\theta) = \underbrace{\langle \mathcal{L}_{\text{CE}}(\mathbf{x}, \mathbf{y}) \rangle_{\mathcal{X}_s}}_{\mathcal{L}_{F,\text{seg}}(\theta)} + \lambda_{\text{adv}} \underbrace{\langle \mathcal{L}_{\text{BCE}}(D(\mathbf{Q_x}), 1) \rangle_{\mathcal{X}_t}}_{\mathcal{L}_{F,\text{adv}}(\theta)}$$

※ N번째 target set을 class 1으로 분류하면서 source prediction이 label과 같도록 학습

Multi-Target Domain Adaptation

- MTAF
  - Multi-Target Knowledge Transfer



– Target-agnostic classifier

$$\mathcal{L}_{\text{KL},n}(\mathbf{x}) = \sum_{\boldsymbol{k} \in [H] \times [W] \times [C]} \mathbf{P}_{n,\mathbf{x}}(\boldsymbol{k}) \log \frac{\mathbf{P}_{n,\mathbf{x}}(\boldsymbol{k})}{\mathbf{P}_{\mathbf{x}}(\boldsymbol{k})} \qquad \mathcal{L}_{F_{\text{agn}}^{\text{cls}}}(\theta) = \frac{1}{T} \sum_{n \in [T]} \left\langle \mathcal{L}_{\text{KL},n}(\mathbf{x}) \right\rangle_{\mathcal{X}_{\text{t},n}}$$

- N개의 target-specific classifier를 각 target set에 대한 teacher model, target-agnostic을 student model으로 설정하고 prediction은 각각 $P_{n,x}(k)$, $P_x(k)$ 정의
- 각 target dataset에 대해서 두 개의 prediction이 같아지도록 학습 (KD)

서강대학교 SOGANG UNIVERSITY

VDS LAB

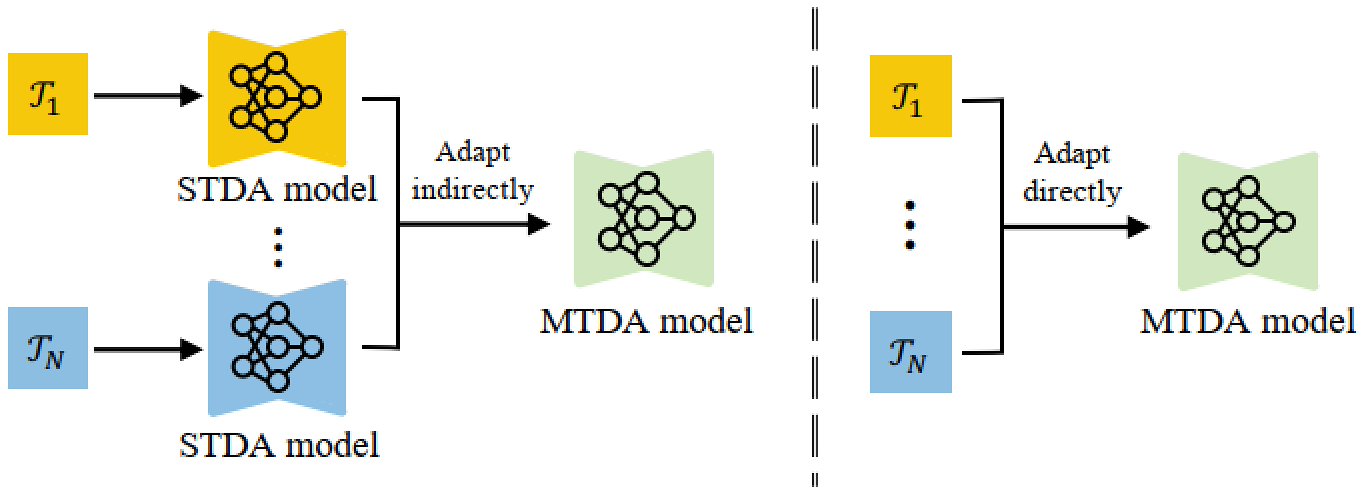
Multi-Target Domain Adaptation

- MTAF
  - Experiments

GTA5 → Cityscapes + Mapillary + IDD											
Method	Target	Train	flat	constr.	object	nature	sky	human	vehicle	mIoU	mIoU Avg.
Single-Target Baselines [25]	Cityscapes	✓	93.5	80.5	26.0	78.5	78.5	55.1	76.4	69.8 (*)	65.5
	Mapillary	-	86.8	69.0	30.2	71.2	91.5	35.3	59.5	63.3 ↓6.3	
	IDD	-	91.3	52.3	13.3	76.1	88.7	46.7	74.8	63.3 ↓1.8	
	Cityscapes	-	89.3	79.3	19.5	76.9	84.6	47.7	63.0	65.8 ↓4.0	66.7
	Mapillary	✓	89.5	72.6	31.0	75.3	94.1	50.7	73.8	69.6 (*)	
	IDD	-	91.7	54.3	13.0	77.3	92.3	47.4	76.8	64.7 ↓0.4	
	Cityscapes	-	78.6	79.2	24.8	77.6	83.6	48.7	44.8	62.5 ↓7.3	65.5
	Mapillary	-	88.5	71.2	32.4	72.8	92.8	51.3	73.7	69.0 ↓0.6	
	IDD	✓	91.2	53.1	16.0	78.2	90.7	47.9	78.9	65.1 (*)	
Multi-Target Baseline [25]	Cityscapes	✓	93.6	80.6	26.4	78.1	81.5	51.9	76.4	69.8 –	67.8
	Mapillary	✓	89.2	72.4	32.4	73.0	92.7	41.6	74.9	68.0 ↓1.6	
	IDD	✓	92.0	54.6	15.7	77.2	90.5	50.8	78.6	65.6 ↑0.5	
Multi-Dis.	Cityscapes	✓	94.6	80.0	20.6	79.3	84.1	44.6	78.2	68.8 ↓1.0	68.2
	Mapillary	✓	89.0	72.5	29.3	75.5	94.7	50.3	78.9	70.0 ↑0.4	
	IDD	✓	91.6	54.2	13.1	78.4	93.1	49.6	80.3	65.8 ↑0.7	
MTKT	Cityscapes	✓	94.6	80.7	23.8	79.0	84.5	51.0	79.2	**70.4** ↑0.6	**69.1**
	Mapillary	✓	90.5	73.7	32.5	75.5	94.3	51.2	80.2	**71.1** ↑1.5	
	IDD	✓	91.7	55.6	14.5	78.0	92.6	49.8	79.4	**65.9** ↑0.8	

Multi-Target Domain Adaptation

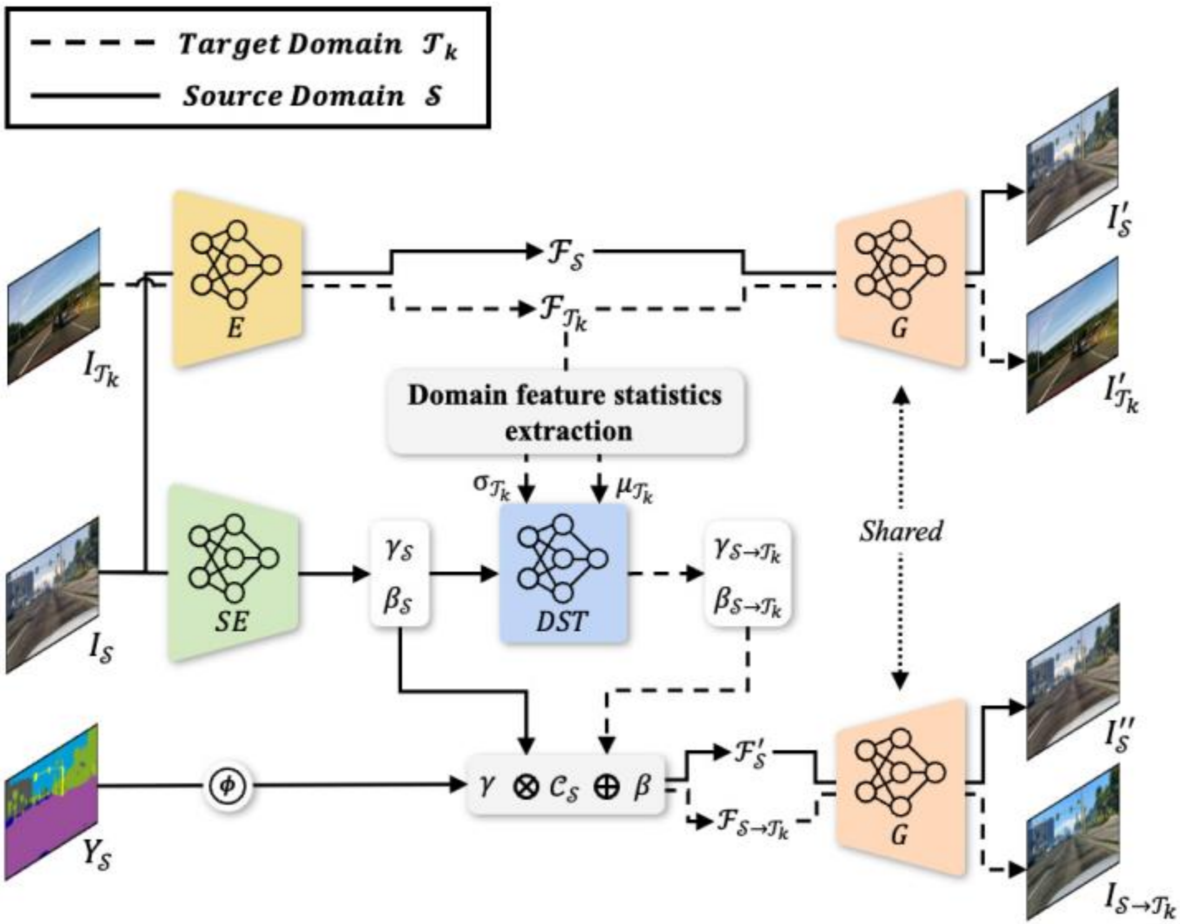
- ADAS
  - N개의 Single-Target Domain Adaptation model을 훈련하는 방식이 아닌 하나의 Multi-Target Domain Adaptation network로 학습
  - 두 개의 sub-modules으로 구성
    - MTDT-Net : 각 target의 attribute를 source domain으로 transfer시키는 module
    - BARS : Feature statics가 동일한 pixel을 선택하는 module



Existing MTDA method ADAS

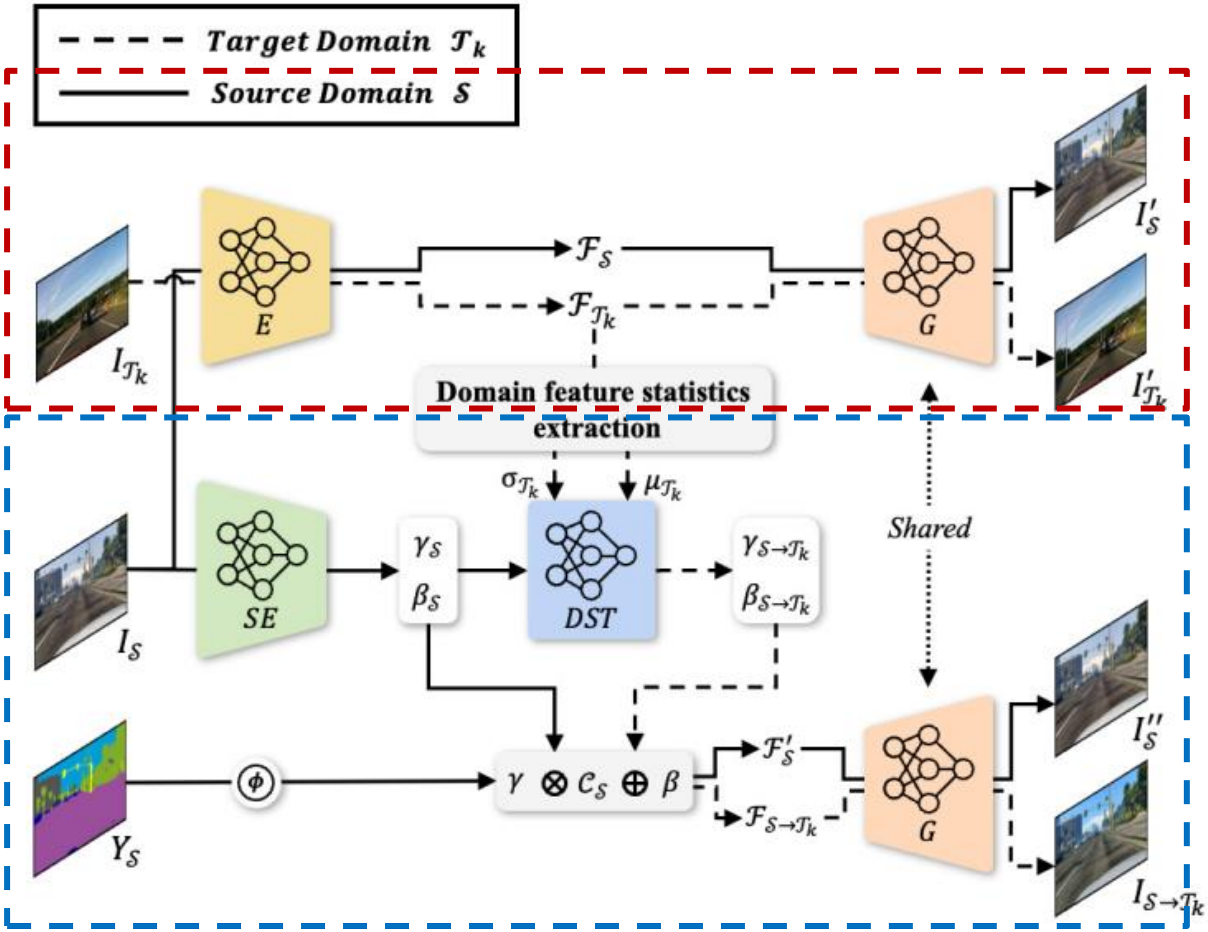
Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network



Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network



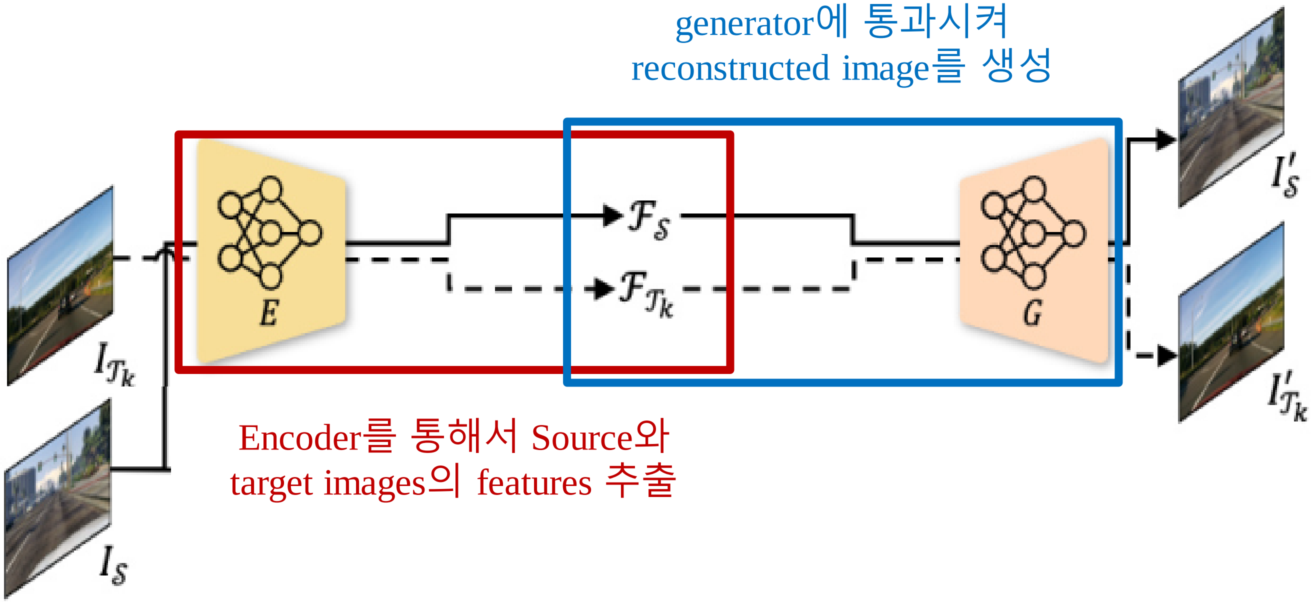
서강대학교 SOGANG UNIVERSITY

VDS LAB

Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network

Source와 target features을
generator에 통과시켜
reconstructed image를 생성



Encoder를 통해서 Source와
target images의 features 추출

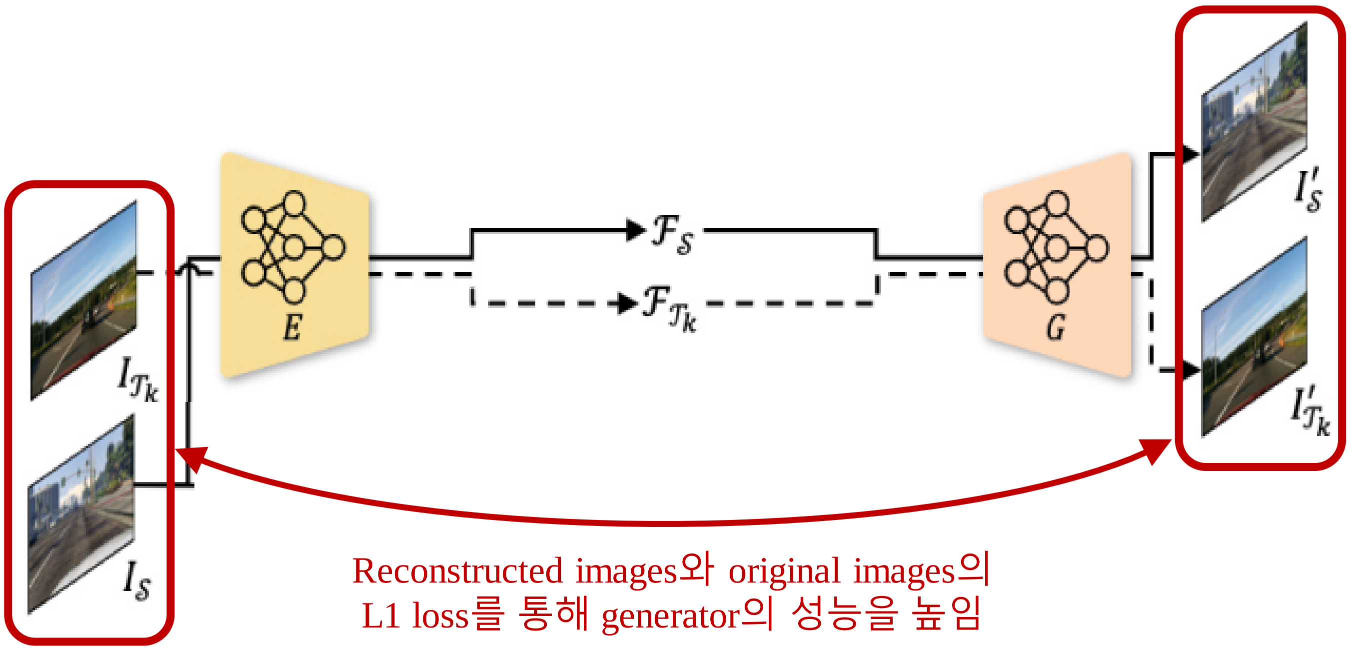
$$\mathcal{L}_{rec} = \mathcal{L}_1(I_{\mathcal{S}}, I'_{\mathcal{S}}) + \mathcal{L}_1(I_{\mathcal{S}}, I''_{\mathcal{S}}) + \sum_{k=1}^{N} \mathcal{L}_1(I_{\mathcal{T}_k}, I'_{\mathcal{T}_k})$$

서강대학교 SOGANG UNIVERSITY

VDS LAB

Multi-Target Domain Adaptation

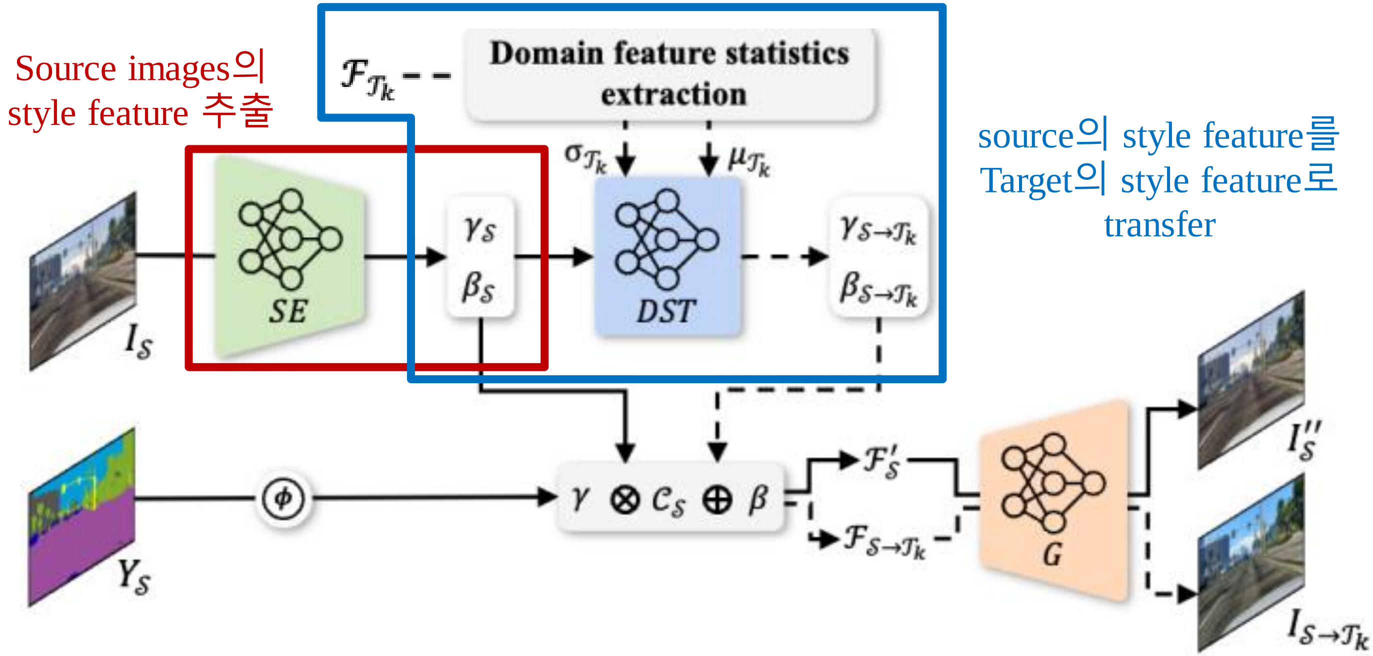
- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network



$$\mathcal{L}_{rec} = \underline{\mathcal{L}_1(I_{\mathcal{S}}, I'_{\mathcal{S}})} + \mathcal{L}_1(I_{\mathcal{S}}, I''_{\mathcal{S}}) + \sum_{k=1}^{N} \underline{\mathcal{L}_1(I_{\mathcal{T}_k}, I'_{\mathcal{T}_k})}$$

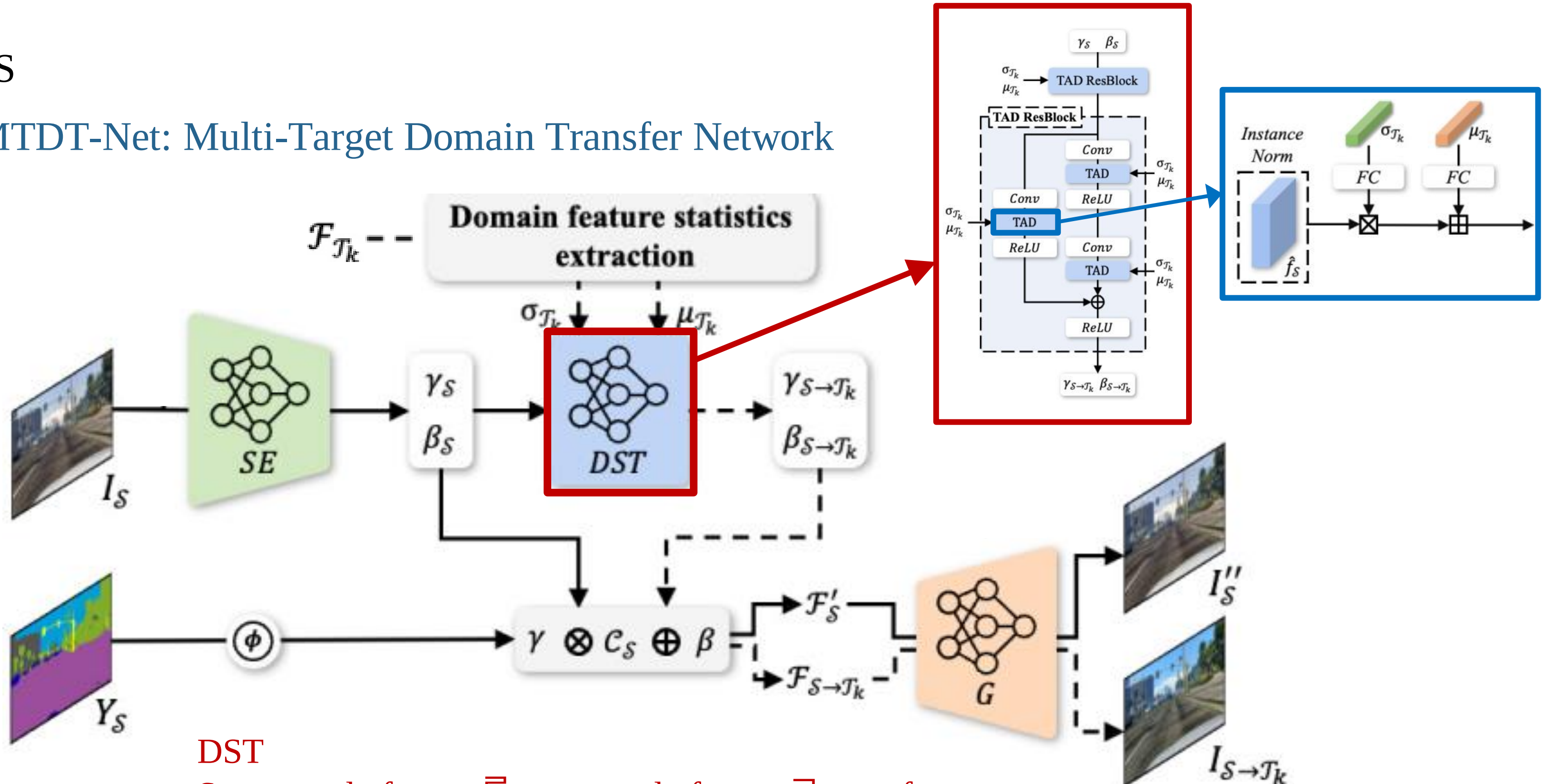
Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network



Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network

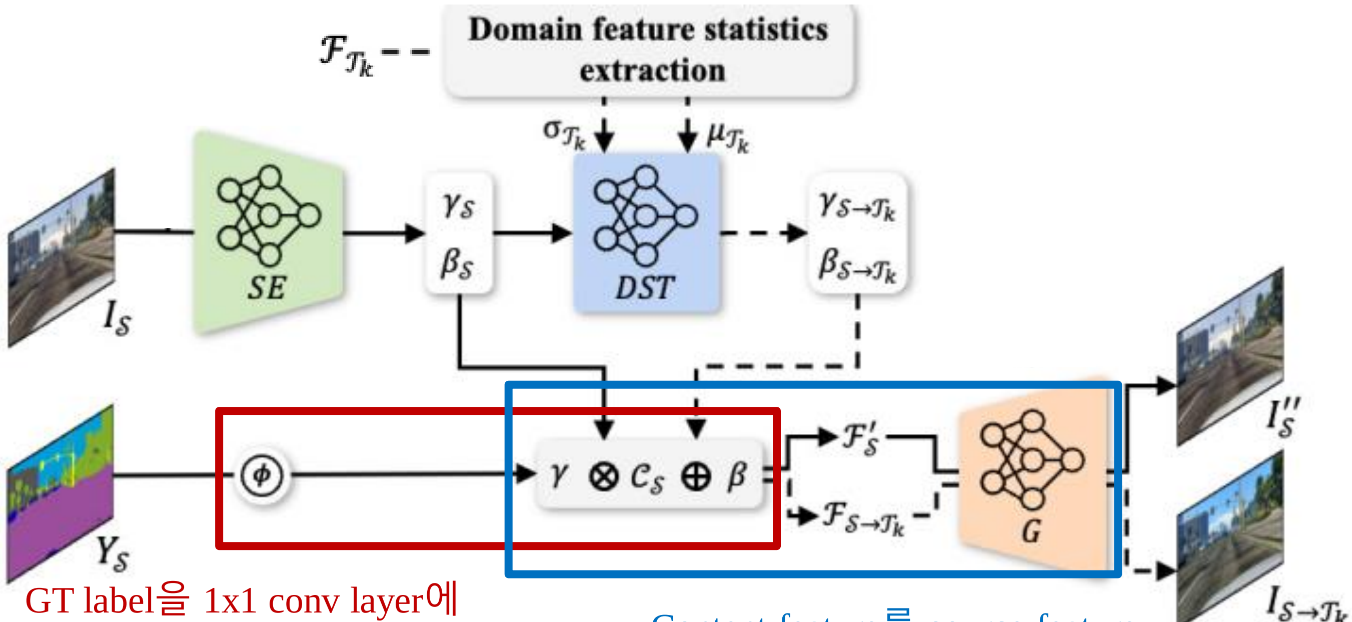


DST
Source style feature를 target style feature로 transfer
두 개의 TAD ResBlock (Conv, TAD, ReLU 조합)으로 구성

TAD
Target의 mean과 variance을 FC layer를 통과시켜 scale과 bias으로 사용해 normalized input을 modulate시킴

Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network

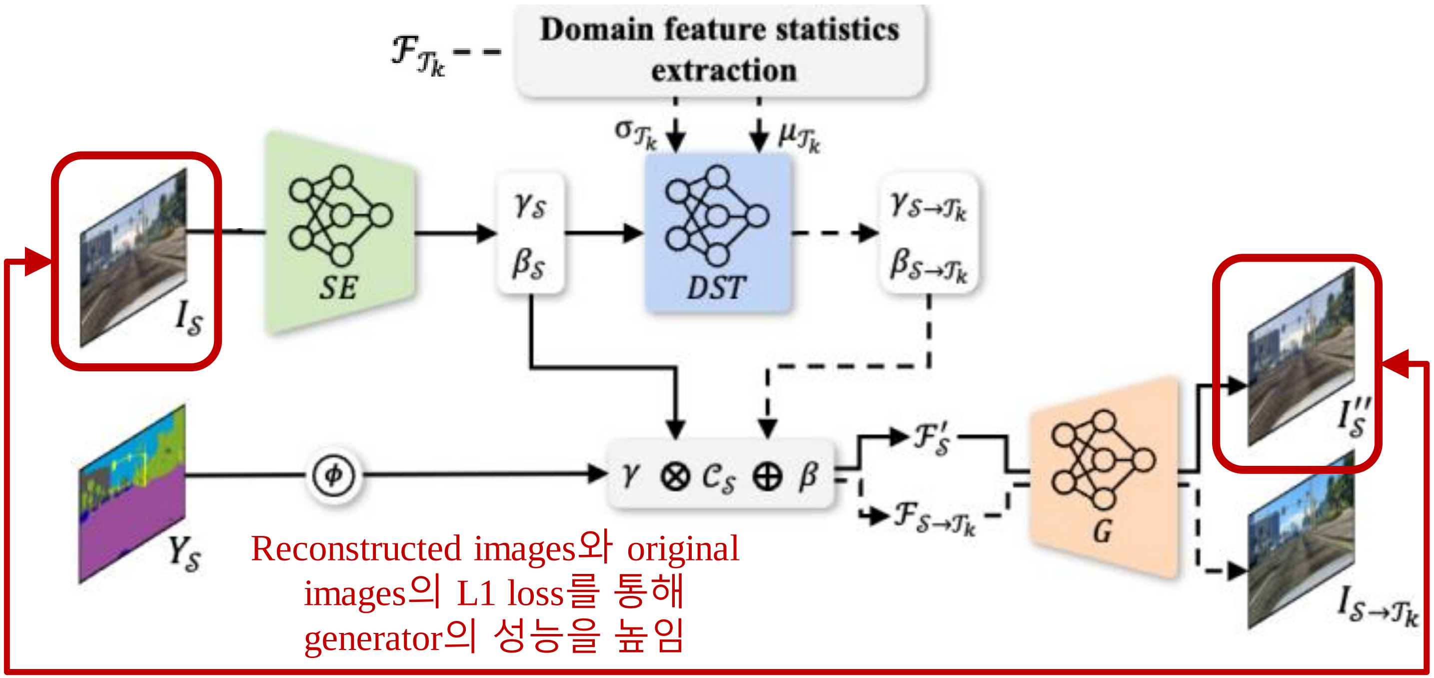


GT label을 1x1 conv layer에 통과시켜 content feature C_S 추출

Content feature를 source feature, transferred style feature와 합성하여 각각 F'_S, $F_{S\to T_k}$ feature 추출하며 reconstructed image 생성

Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network



$$\mathcal{L}_{rec} = \mathcal{L}_1(I_S, I'_S) + \underline{\mathcal{L}_1(I_S, I''_S)} + \sum_{k=1}^{N} \mathcal{L}_1(I_{\mathcal{T}_k}, I'_{\mathcal{T}_k})$$

Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network

$$\mathcal{L}^{\mathcal{G}} = \mathcal{L}_{rec} + \mathcal{L}_{per} + \mathcal{L}_{adv} + \mathcal{L}^{\mathcal{G}}_{cls}$$

$$\mathcal{L}_{rec} = \mathcal{L}_1(I_{\mathcal{S}}, I'_{\mathcal{S}}) + \mathcal{L}_1(I_{\mathcal{S}}, I''_{\mathcal{S}}) + \sum_{k=1}^{N} \mathcal{L}_1(I_{\mathcal{T}_k}, I'_{\mathcal{T}_k})$$

※ Reconstructed image가 original와 L1 loss상 차이가 없도록 학습

$$\mathcal{L}_{adv} = \sum_{k=1}^{N} \Big(\mathbb{E}_{I_{\mathcal{T}_k}} [\log D_{adv}(I_{\mathcal{T}_k})] + \mathbb{E}_{I_{\mathcal{S}\to\mathcal{T}_k}} [1 - \log D_{adv}(I_{\mathcal{S}\to\mathcal{T}_k})] \Big)$$

※ Reconstructed transferred image가 target image와 구별이 안되게 생성되도록 학습

$$\mathcal{L}^{\mathcal{G}}_{cls} = -\sum_{k=1}^{N} t_k \log D_{cls}(I_{\mathcal{S}\to\mathcal{T}_k})$$

※ Reconstructed transferred image가 target class label과 일치하도록 학습

$$\mathcal{L}_{per} = \sum_{k=1}^{N} \sum_{l \in L} \|P_l(I_{\mathcal{S}}) - P_l(I_{\mathcal{S}\to\mathcal{T}_k})\|_2^2$$

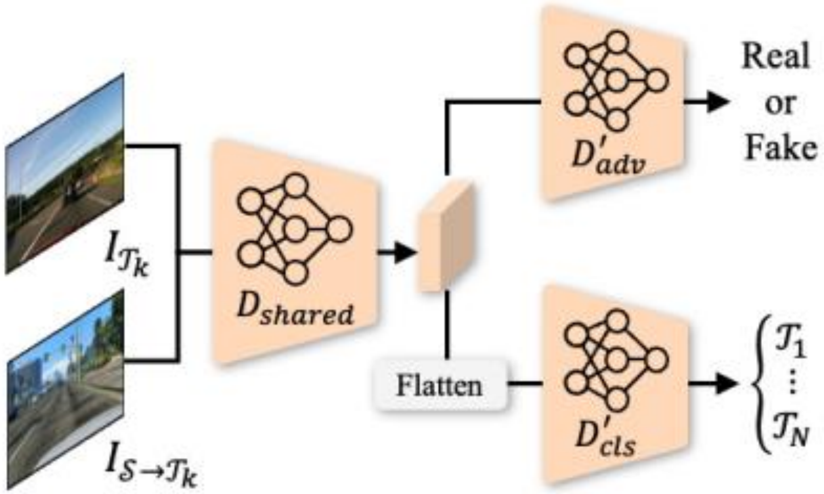
※ Perceptual network의 hidden layer에서 source image와 transferred image의 feature space간 차이를 줄여 source의 전반적인 구조를 유지하도록 학습

서강대학교 SOGANG UNIVERSITY

VDS LAB

Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network



– Adversarial discriminator와 domain classifier로 구성된 multi-head discriminator 사용

  - Adversarial discriminator : image가 real인지 fake인지 구분
  - Domain classifier : Input images의 domain이 어떤 domain에 해당하는지 prediction

Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network

$$\mathcal{L}^{\mathcal{D}} = -\mathcal{L}_{adv} + \mathcal{L}_{cls}^{\mathcal{D}}$$

$$\mathcal{L}_{adv} = \sum_{k=1}^{N} \left(\mathbb{E}_{I_{\mathcal{T}_k}} [\log D_{adv}(I_{\mathcal{T}_k})] + \mathbb{E}_{I_{\mathcal{S}\to\mathcal{T}_k}} [1 - \log D_{adv}(I_{\mathcal{S}\to\mathcal{T}_k})] \right)$$

※ Reconstructed transferred image와 target image를 잘 구별하도록 학습

$$\mathcal{L}_{cls}^{\mathcal{D}} = -\sum_{k=1}^{N} t_k \log D_{cls}(I_{\mathcal{T}_k})$$

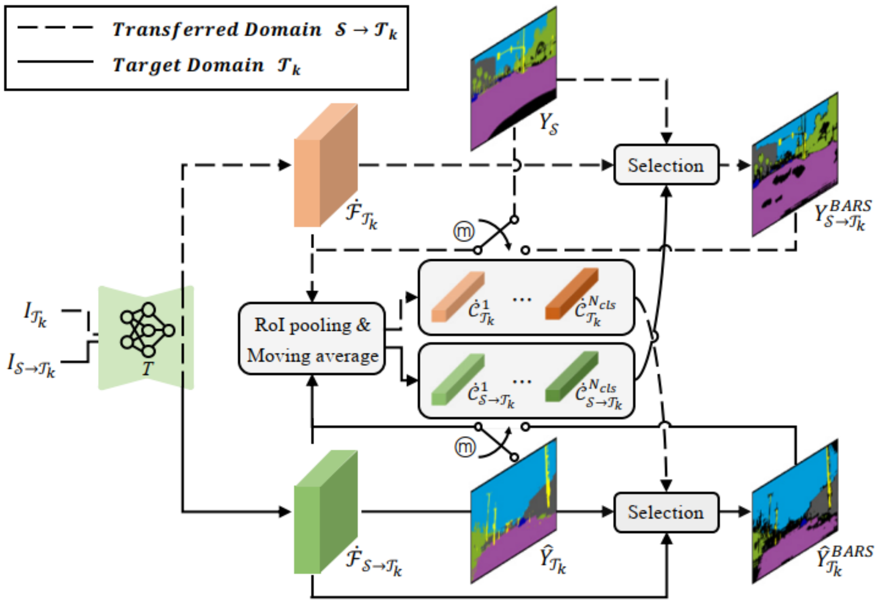
※ Target dataset의 target class label을 잘 구별하도록 학습

Multi-Target Domain Adaptation

- ADAS
  - MTDT-Net: Multi-Target Domain Transfer Network
    - Multi-target domain transfer 결과

Multi-Target Domain Adaptation

- ADAS
  - BARS: Bi-directional Adaptive Region Selection

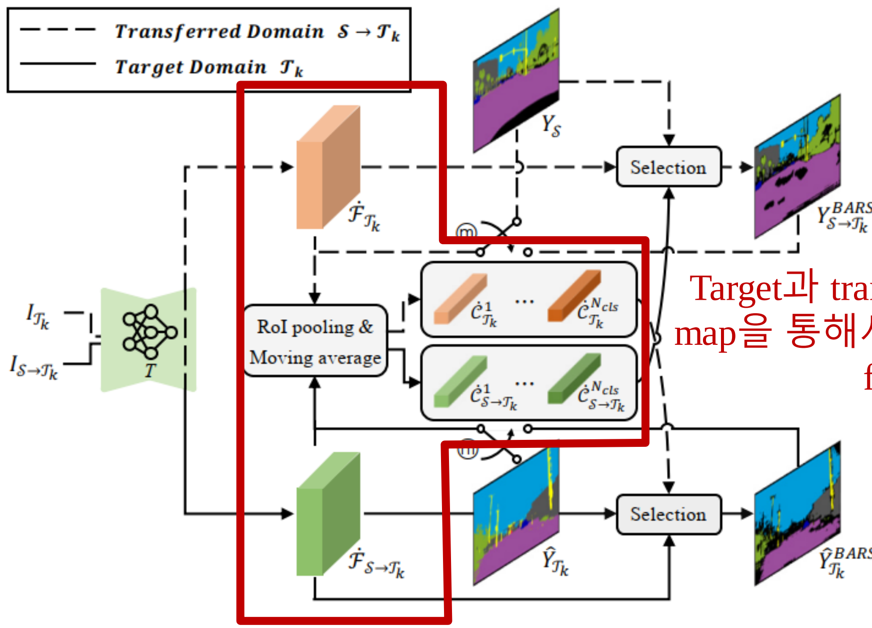


– Ambiguous 또는 noisy label region은 model performance을 제한

– 해당 영역에 대해서 filtering 작업 수행하여 정확한 criterion을 얻게 됨

Multi-Target Domain Adaptation

- ADAS
  - BARS: Bi-directional Adaptive Region Selection



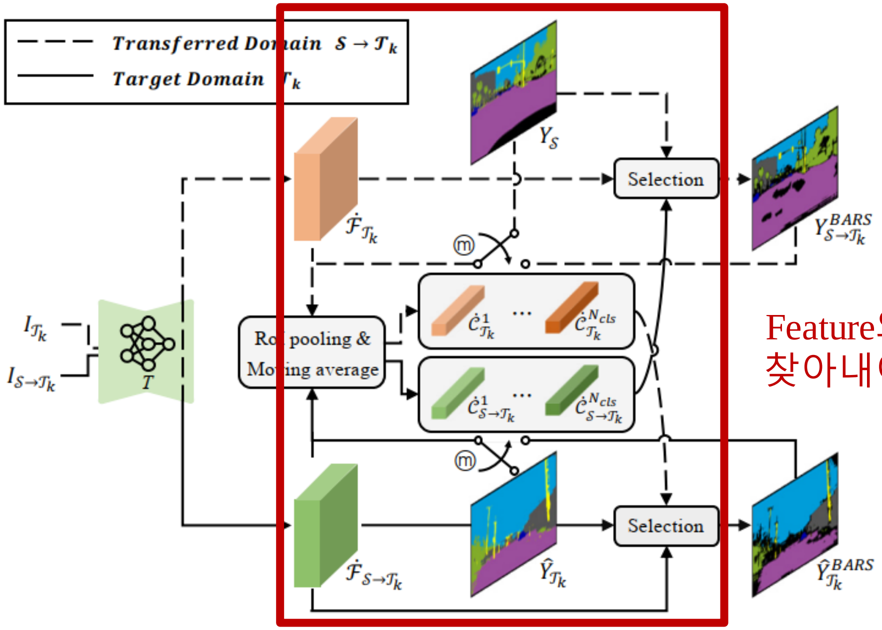
Target과 transferred domain의 feature map을 통해서 각 class에 대한 centroid feature를 추출

$$\dot{\mathcal{C}}^c_{\mathcal{S}\rightarrow\mathcal{T}_k} = \frac{1}{N_c}\sum_i\sum_j \mathbb{1}(Y_{\mathcal{S}}(i,j)=c)\dot{\mathcal{F}}_{\mathcal{S}\rightarrow\mathcal{T}_k}(i,j),$$

$$\dot{\mathcal{C}}^c_{\mathcal{T}_k} = \frac{1}{N_c}\sum_i\sum_j \mathbb{1}(\hat{Y}_{\mathcal{T}_k}(i,j)=c)\dot{\mathcal{F}}_{\mathcal{T}_k}(i,j),$$

Multi-Target Domain Adaptation

- ADAS
  - BARS: Bi-directional Adaptive Region Selection



Feature와 가장 가까운 centroid를
찾아내어 label과 다를 시 filtering

$$\dot{c}_{S \rightarrow T_k}(i,j) = \underset{c}{\text{argmin}} \, ||\dot{\mathcal{F}}_{S \rightarrow T_k}(i,j) - \dot{\mathcal{C}}^c_{T_k}||_2$$

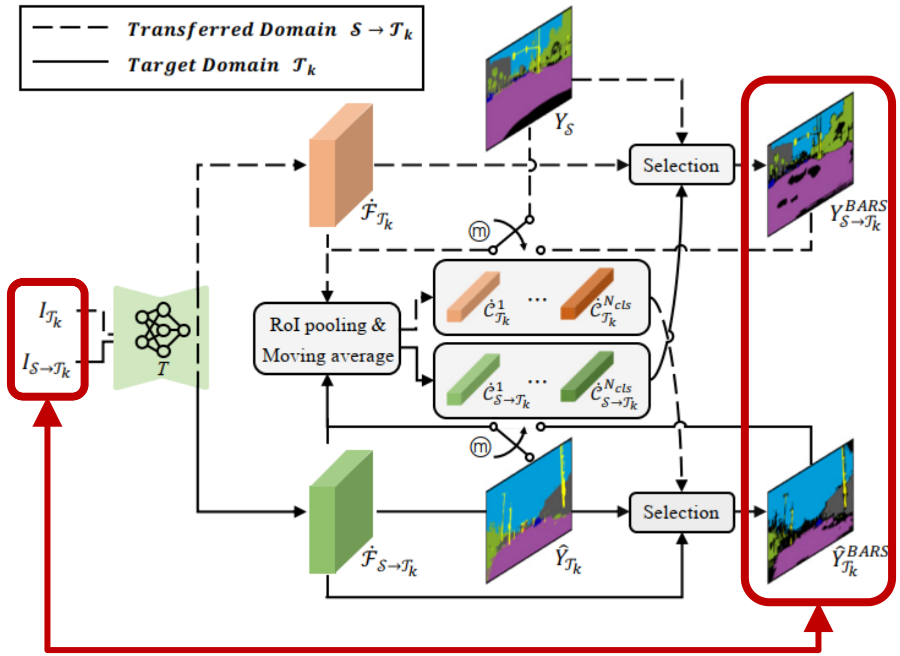
$$\dot{c}_{T_k}(i,j) = \underset{c}{\text{argmin}} \, ||\dot{\mathcal{F}}_{T_k}(i,j) - \dot{\mathcal{C}}^c_{S \rightarrow T_k}||_2$$

$$Y^{BARS}_{S \rightarrow T_k}(i,j) = \begin{cases} Y_S(i,j) & \text{if } \dot{c}_{S \rightarrow T_k}(i,j) = Y_S(i,j) \\ \emptyset & \text{otherwise} \end{cases}$$

$$\hat{Y}^{BARS}_{T_k}(i,j) = \begin{cases} \hat{Y}_{T_k}(i,j) & \text{if } \dot{c}_{T_k}(i,j) = \hat{Y}_{T_k}(i,j) \\ \emptyset & \text{otherwise} \end{cases}$$

Multi-Target Domain Adaptation

- ADAS
  - BARS: Bi-directional Adaptive Region Selection



$$\min_{T} \left(\mathcal{L}_{Task}(I_{\mathcal{S}\rightarrow\mathcal{T}_k}, Y^{BARS}_{\mathcal{S}\rightarrow\mathcal{T}_k}) + \mathcal{L}_{Task}(I_{\mathcal{T}_k}, \hat{Y}^{BARS}_{\mathcal{T}_k}) \right)$$

Multi-Target Domain Adaptation

- ADAS
  - Experiments

	Method	Target	flat	constr.	object	nature	sky	human	vehicle	mIoU	Avg.
G →C, I	ADVENT [55]	C	93.9	80.2	26.2	79.0	80.5	52.5	78.0	70.0	67.4
		I	**91.8**	54.5	14.4	76.8	90.3	47.5	78.3	64.8	
	MTKT [48]	C	94.5	82.0	23.7	80.1	**84.0**	51.0	77.6	70.4	68.2
		I	91.4	56.6	13.2	**77.3**	**91.4**	51.4	**79.9**	65.9	
	Ours	C	**95.1**	**82.6**	**39.8**	**84.6**	81.2	**63.6**	**80.7**	**75.4**	**71.2**
		I	90.5	**63.0**	**22.2**	73.7	87.9	**54.3**	76.9	**66.9**	
G →C, M	ADVENT [55]	C	93.1	80.5	24.0	77.9	81.0	52.5	75.0	69.1	68.9
		M	90.0	71.3	31.1	73.0	92.6	46.6	76.6	68.7	
	MTKT [48]	C	95.0	81.6	23.6	80.1	83.6	53.7	79.8	71.1	70.9
		M	**90.6**	73.3	31.0	75.3	**94.5**	52.2	**79.8**	70.8	
	Ours	C	**96.4**	**83.5**	**35.1**	**83.8**	**84.9**	**62.3**	**81.3**	**75.3**	**73.9**
		M	88.6	**73.7**	**41.0**	**75.4**	93.4	**58.5**	77.2	**72.6**	
G →C, I, M	ADVENT [55]	C	93.6	80.6	26.4	78.1	81.5	51.9	76.4	69.8	67.8
		I	**92.0**	54.6	15.7	77.2	90.5	50.8	78.6	65.6	
		M	89.2	72.4	32.4	73.0	92.7	41.6	74.9	68.0	
	MTKT [48]	C	94.6	80.7	23.8	79.0	84.5	51.0	79.2	70.4	69.1
		I	91.7	**55.6**	14.5	78.0	**92.6**	49.8	**79.4**	65.9	
		M	**90.5**	**73.7**	32.5	75.5	**94.3**	51.2	**80.2**	71.1	
	Ours	C	**95.8**	**82.4**	**38.3**	**82.4**	**85.0**	**60.5**	**80.2**	**74.9**	**71.3**
		I	89.9	52.7	**25.0**	**78.1**	92.1	**51.0**	77.9	**66.7**	
		M	89.2	71.5	**45.2**	**75.8**	92.3	**56.1**	75.4	**72.2**	

Table 1. Quantitative comparison between our method and state-of-the-art methods on GTA5 (G) to Cityscapes (C), IDD (I), and Mapilary (M) with 7 classes setting. **Bold**: Best score among all the methods.