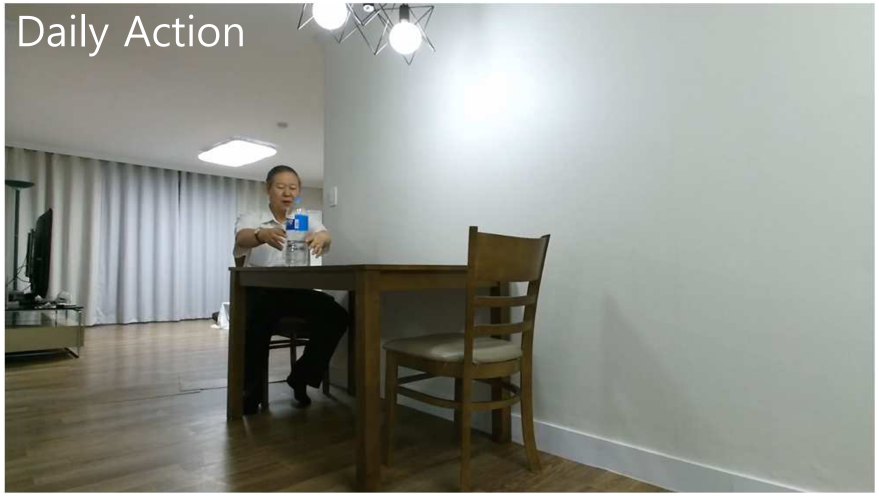
UST seminar

Towards domain-agnostic Video action recognition

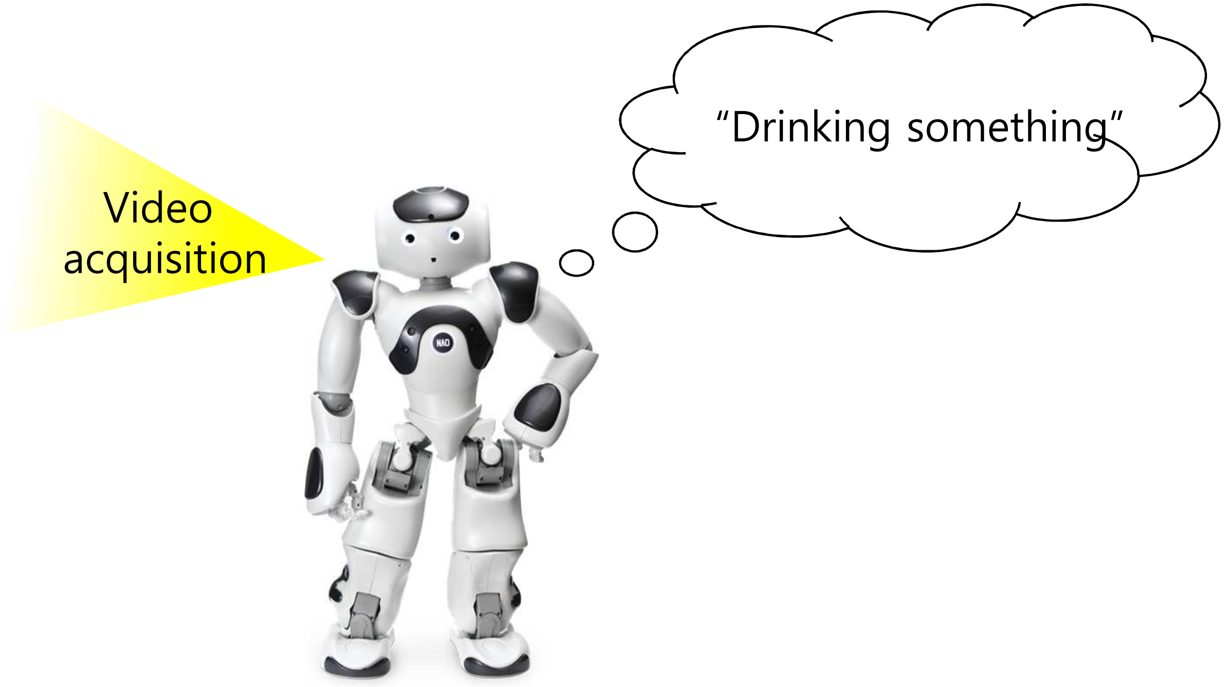
Hyungmin Kim
UST-ETRI
Ph.D. student
khm159@etri.re.kr/ust.ac.kr

26th Oct, 2023

Human action recognition in Human-Robot Interaction



Video from [1]

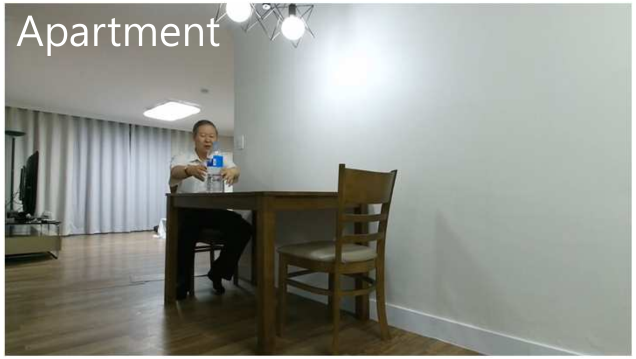


- **Problem definition :**

  Given input video v_i
  Model predicts corresponding action label y_i

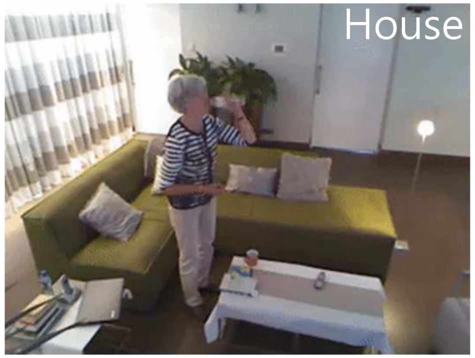
[1] Jang, Jinhyeok, et al. "ETRI-activity3D: A large-scale RGB-D dataset for robots to recognize daily activities of the elderly." *IROS*, 2020.

Service robot encounters numerous domains



Video from [1]

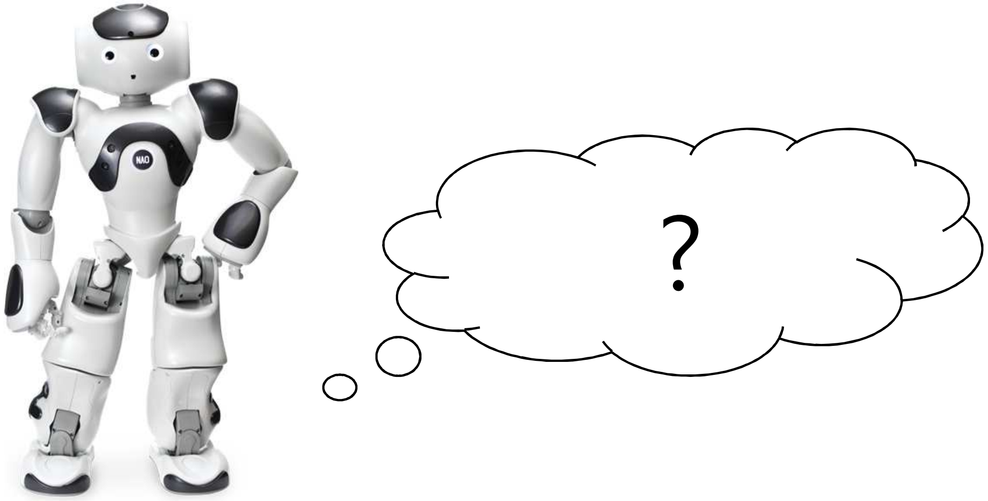
Video from [2]



Video from [3]

- Care-robot encounters numerous domains

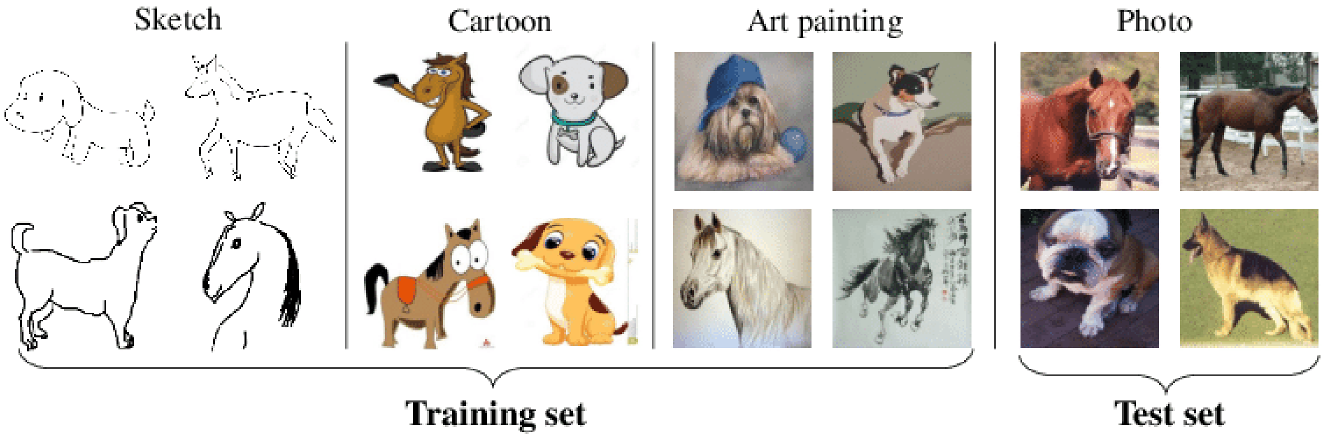
  Apartment, House, Building, Office ....



[1] Jang, Jinhyeok, et al. "ETRI-activity3D: A large-scale RGB-D dataset for robots to recognize daily activities of the elderly." *IROS*, 2020.
[2] Shahroudy, Amir, et al. "Ntu rgb+ d: A large scale dataset for 3d human activity analysis." *CVPR*. 2016.
[3] Das, Srijan, et al. "Toyota smarthome: Real-world activities of daily living." *ICCV*. 2019.

What is Domain Generalization?

Domain Generalization?



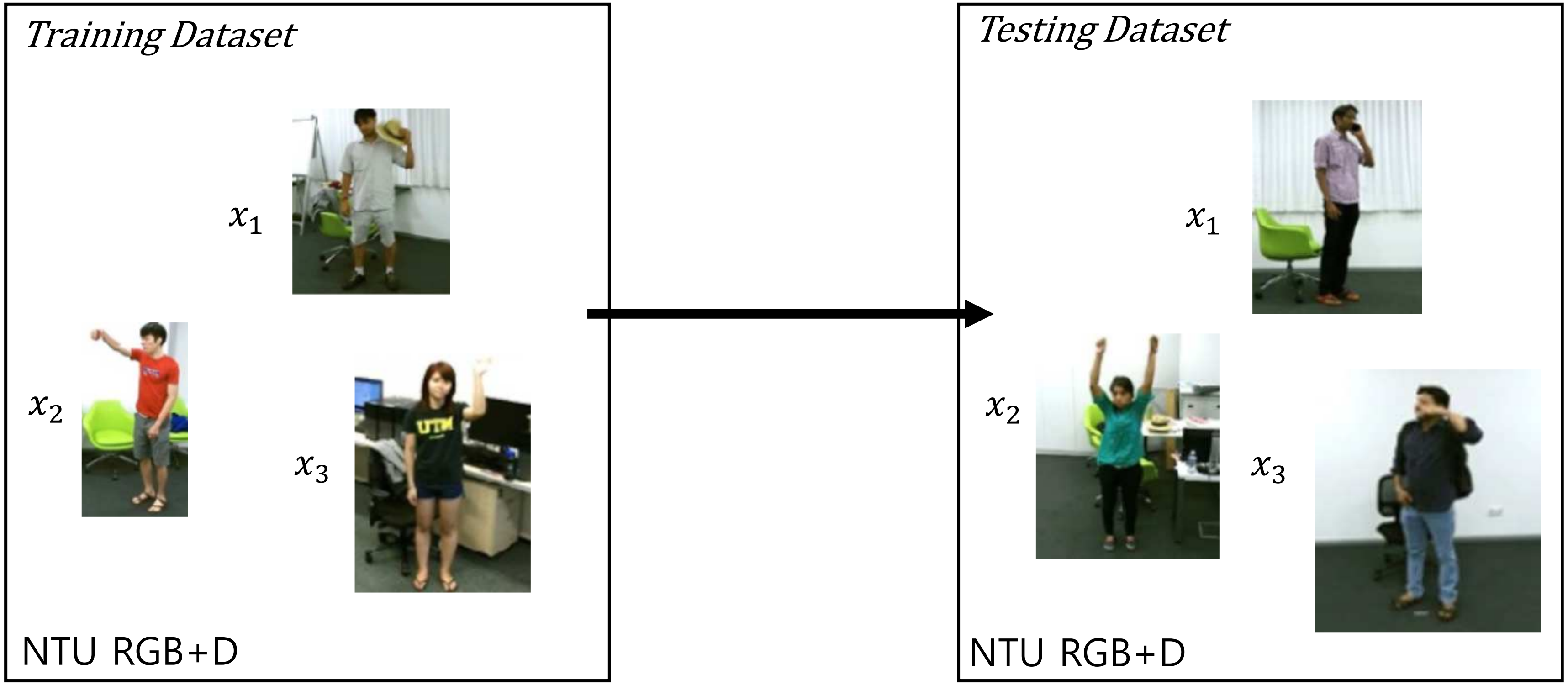
Train a generalized model with datasets from different domains that share the same semantic to improve performance on Un-seen domain.

[4] Wang, Jindong, et al. "Generalizing to unseen domains: A survey on domain generalization." *IEEE Transactions on Knowledge and Data Engineering* (2022).

In i.i.d. assumption…

- Usually, the training set and test set are obtained in the same or similar way.
  Each sample is Independent Identically distributed.

Similar back ground, similar camera view point, etc…



Training Dataset

x_1

x_2

x_3

NTU RGB+D

Testing Dataset

x_1

x_2

x_3

NTU RGB+D

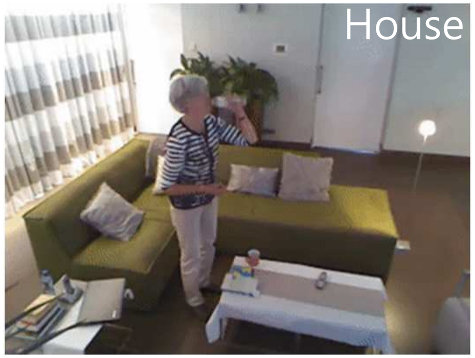
i.i.d. assumption is easily violated in real world

- Care-robot encounters numerous domains

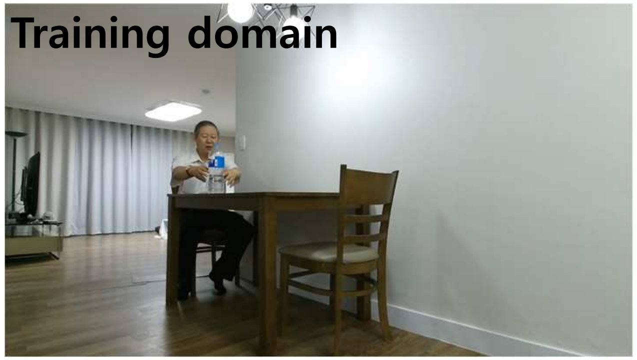
  Apartment, House, Building, Office ....

Inferenced domains

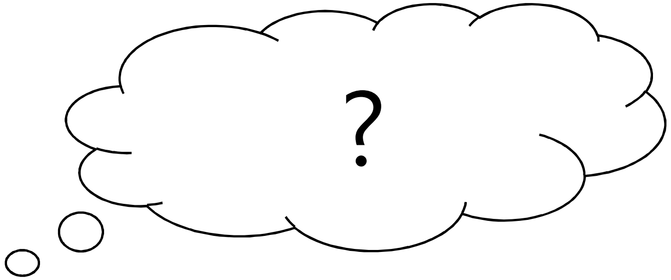
Video from [2]



Video from [3]



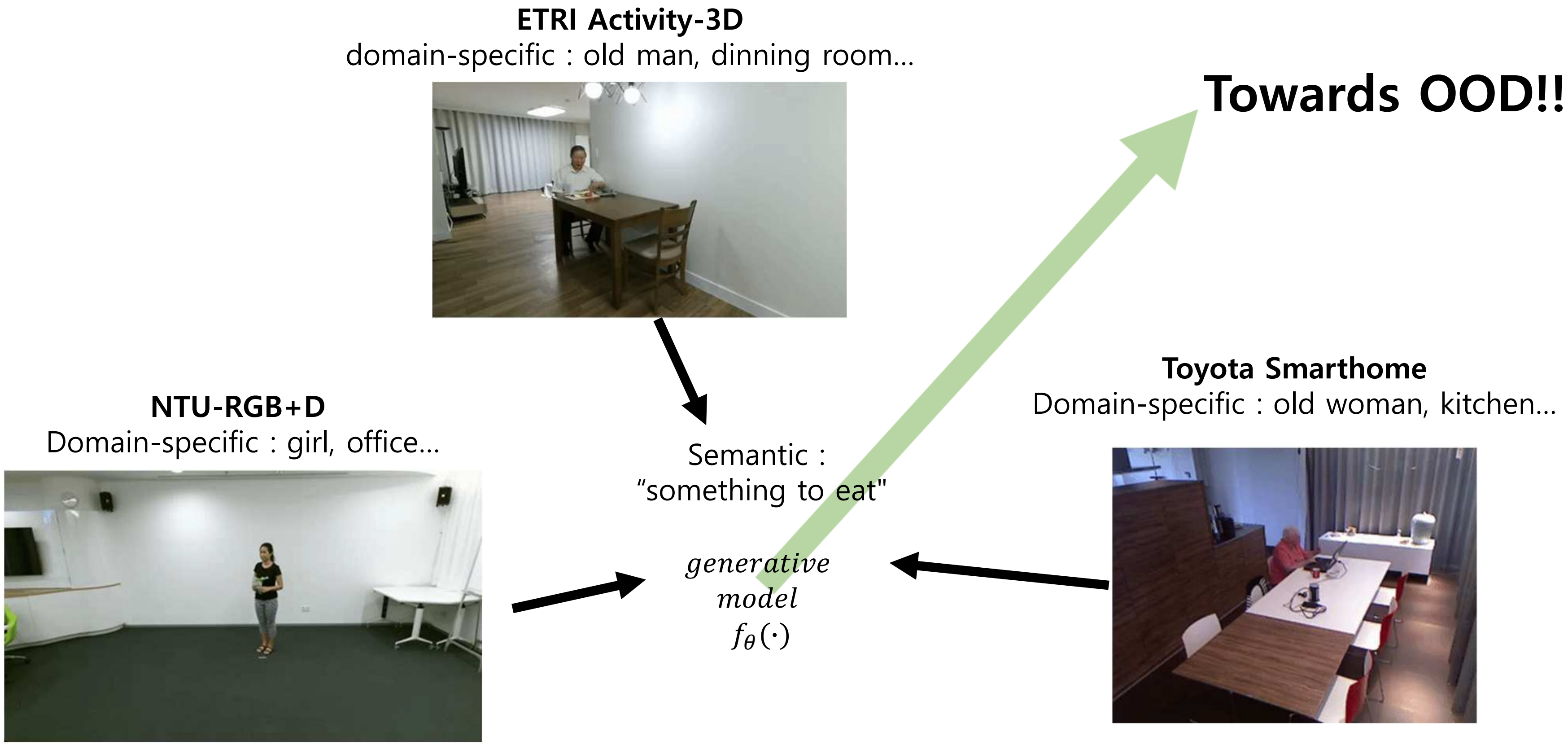
Video from [1]



[1] Jang, Jinhyeok, et al. "ETRI-activity3D: A large-scale RGB-D dataset for robots to recognize daily activities of the elderly." *IROS*, 2020.
[2] Shahroudy, Amir, et al. "Ntu rgb+ d: A large scale dataset for 3d human activity analysis." *CVPR*. 2016.
[3] Das, Srijan, et al. "Toyota smarthome: Real-world activities of daily living." *ICCV*. 2019.

The goal of DG

Using many training domains well
to train models that are **robust to domain differences**.



ETRI Activity-3D
domain-specific : old man, dinning room…

Towards OOD!!

NTU-RGB+D
Domain-specific : girl, office…

Semantic :
“something to eat"

$generative$
$model$
$f_\theta(\cdot)$

Toyota Smarthome
Domain-specific : old woman, kitchen…

Evaluation protocol

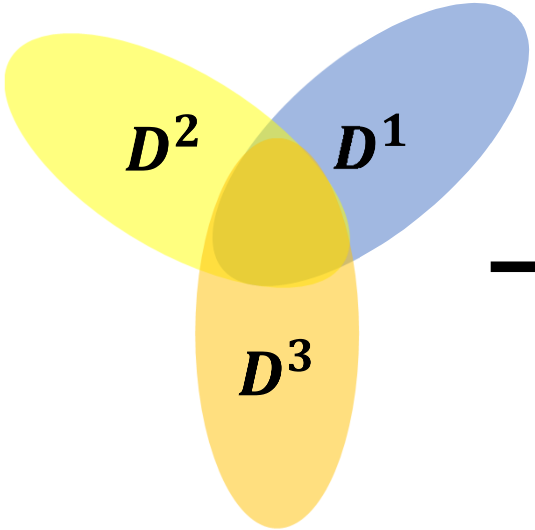
- **Leave-one-domain-out protocol**

  Set of domains $D = \{D^i \mid i = 1, \dots M\}$

  Each domain $D^i = \{(x^{i,j}, y^{i,j})\}_{j=1}^{n_i}$

  one domain from D is excluded or left out during training,
  while the remaining domains of D are used for training the model.

  - n_j : number of samples of D^i
  - M : number of domains



→

If $D = \{D^i \mid i = 1, \dots 4\}$

Evaluate D^4

Domain Generalization

1. **Data Augmentation**

   Transform or generate input data to **make it domain-agnostic**.
   This is primarily achieved through the use of generative models or augmentation techniques

2. **Feature Alignment**

   A research field that **utilizes domain priors (domain numbers)**
   to align features, primarily leveraging **contrastive learning.**

3. **Meta-learning**

   Domain generalization using **meta-learning** techniques, primarily
   involving the **creation of diverse pseudo domain sets**

4. **Style Augmentation**

   The **feature augmentation method** using **style representation and normalization.**

Domain Generalization

1. **Data Augmentation**

   Transform or generate input data to **make it domain-agnostic**.
   This is primarily achieved through the use of generative models or augmentation techniques

2. **Feature Alignment**

   A research field that **utilizes domain priors (domain numbers)**
   to align features, primarily leveraging **contrastive learning.**

3. **Meta-learning**

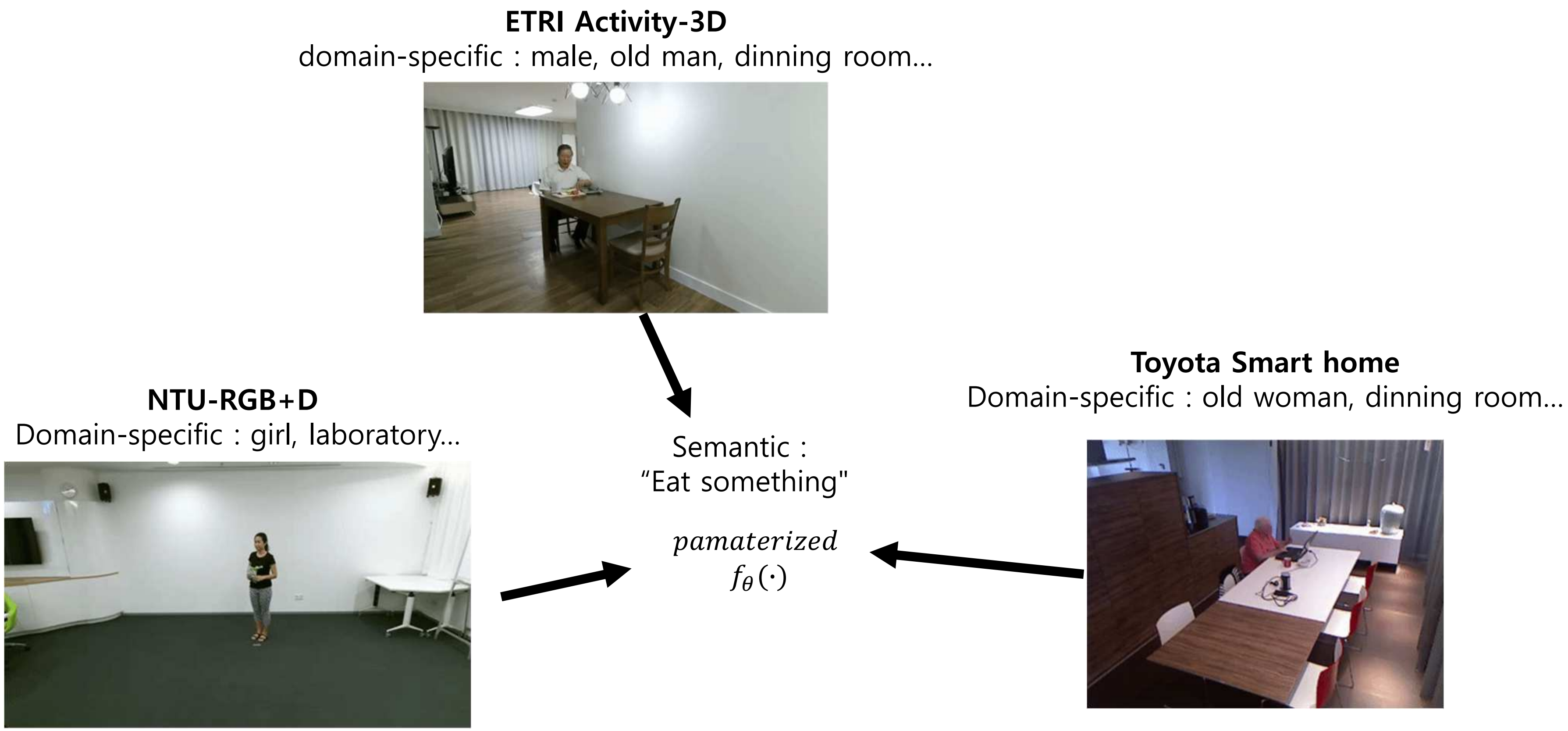
   Domain generalization using **meta-learning** techniques, primarily
   involving the **creation of diverse pseudo domain sets**

4. **Style Augmentation (Today's highlight)**

   The **feature augmentation method** using **style representation and normalization.**

Style and Domain

- **Domain is closely related to the style!**



ETRI Activity-3D
domain-specific : male, old man, dinning room…

NTU-RGB+D
Domain-specific : girl, laboratory…

Toyota Smart home
Domain-specific : old woman, dinning room…

Semantic :
“Eat something"

$pamaterized$
$f_\theta(\cdot)$

How to model the style information?

- **Instance Normalization? Style information modeling!**

왜 효과가 있는가? Back to the style-transfer

$$BN(X) = \gamma\Big(\frac{X - \mu_{batch}}{\sigma_{batch}}\Big) + \beta$$

$$\mathrm{IN}(x) = \gamma \frac{x - \mu(x)}{\sigma(x)} + \beta,$$

$$\mathrm{AdaIN}(x) = \sigma(y)\frac{x - \mu(x)}{\sigma(x)} + \mu(y).$$

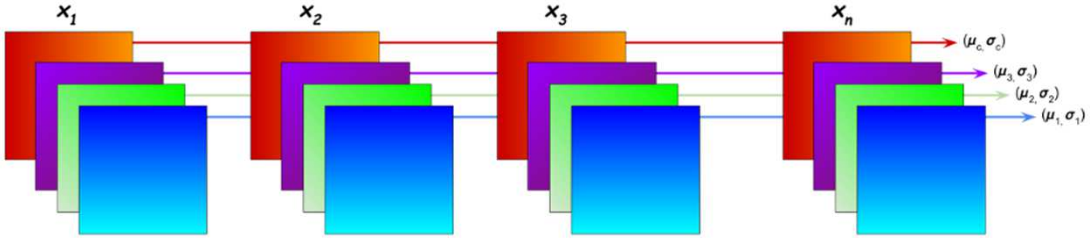
For reasons unknown (empirically), the addition of Batch Normalization resulted in improved image generation performance [5]

It indicates that through Instance Normalization, the second-order statistics of each instance can model style information and, by utilizing this for normalization, it implies that style information can be removed (normalized) [6].

So, if we use Instance Normalization to remove style information and then extract second-order statistics from another image and transfer them, we should achieve style transfer, right? [7]

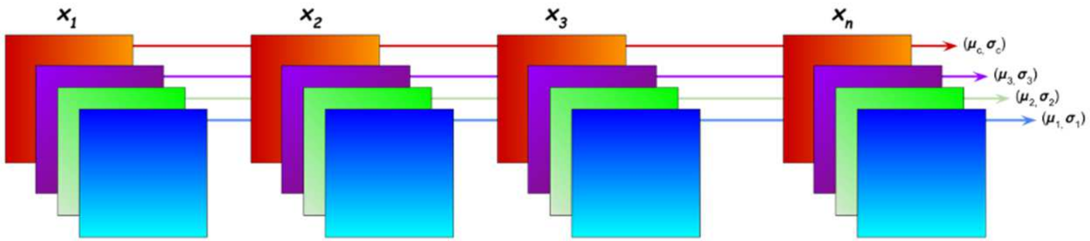
[5] Radford, A., et al., Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv,* 2015
[6] Ulyanov, D., et al., Instance normalization: The missing ingredient for fast stylization. *arXiv 2016.*
[7] Huang, Xun, et al. ,"Arbitrary style transfer in real-time with adaptive instance normalization." *ICCV,* 2017.

Batch normalization vs Instance normalization



Batch Normalization

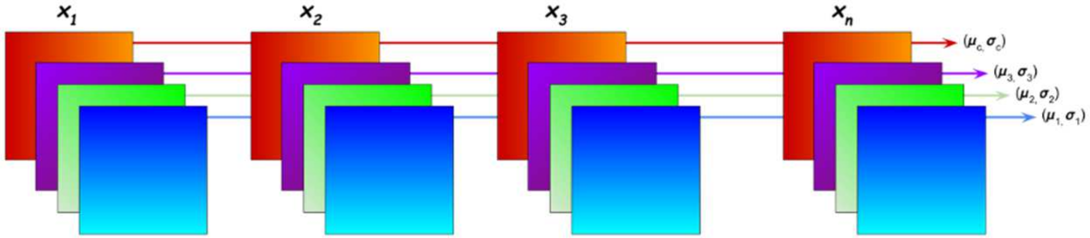
Batch normalization vs Instance normalization



Batch Normalization

Calculate mean and standard deviation across all samples

Batch normalization vs Instance normalization



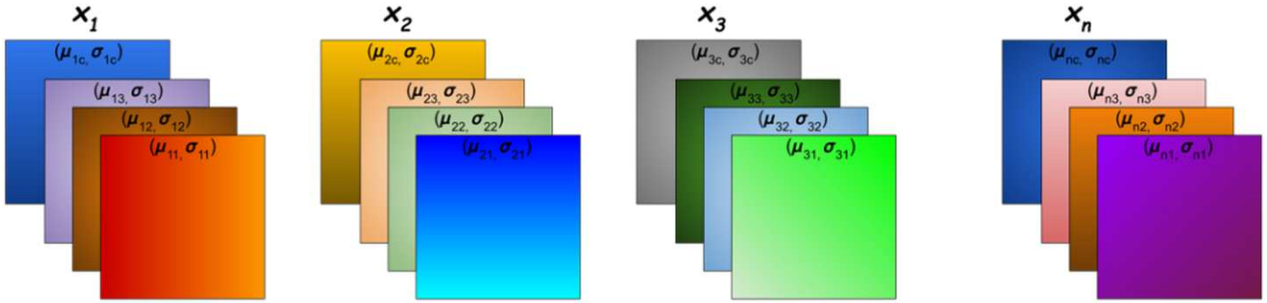
Batch Normalization
Calculate mean and standard deviation across all samples

$$\mu_c = \frac{1}{NHW} \boxed{\sum_{i=1}^{N}} \sum_{j=1}^{H} \sum_{k=1}^{W} x_{icjk}$$

$$\sigma_c^2 = \frac{1}{NHW} \boxed{\sum_{i=1}^{N}} \sum_{j=1}^{H} \sum_{k=1}^{W} (x_{icjk} - \mu_c)^2$$

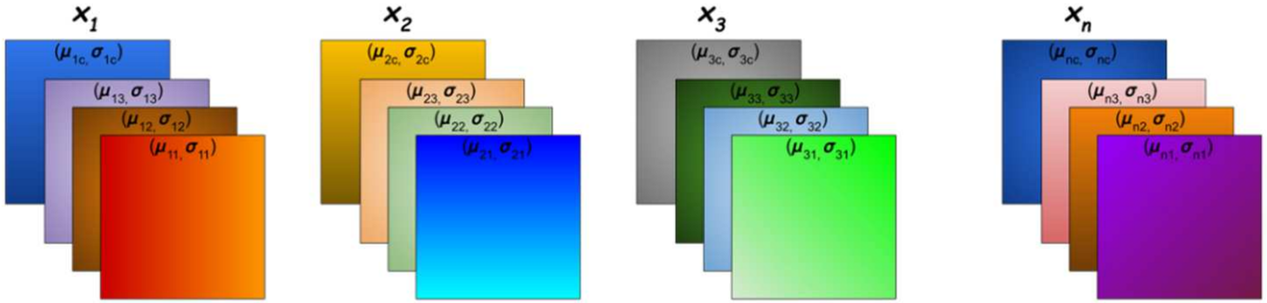
$$\hat{x} = \frac{x - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

Batch normalization vs Instance normalization


x_1
(μ_{1c}, σ_{1c})
(μ_{13}, σ_{13})
(μ_{12}, σ_{12})
(μ_{11}, σ_{11})
x_2
(μ_{2c}, σ_{2c})
(μ_{23}, σ_{23})
(μ_{22}, σ_{22})
(μ_{21}, σ_{21})
x_3
(μ_{3c}, σ_{3c})
(μ_{33}, σ_{33})
(μ_{32}, σ_{32})
(μ_{31}, σ_{31})
x_n
(μ_{nc}, σ_{nc})
(μ_{n3}, σ_{n3})
(μ_{n2}, σ_{n2})
(μ_{n1}, σ_{n1})

Instance Normalization

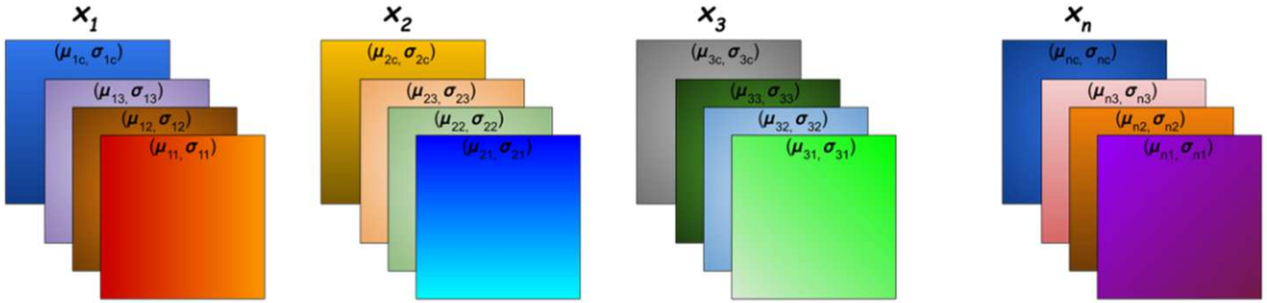
Batch normalization vs Instance normalization


x_1
(μ_{1c}, σ_{1c})
(μ_{13}, σ_{13})
(μ_{12}, σ_{12})
(μ_{11}, σ_{11})
x_2
(μ_{2c}, σ_{2c})
(μ_{23}, σ_{23})
(μ_{22}, σ_{22})
(μ_{21}, σ_{21})
x_3
(μ_{3c}, σ_{3c})
(μ_{33}, σ_{33})
(μ_{32}, σ_{32})
(μ_{31}, σ_{31})
x_n
(μ_{nc}, σ_{nc})
(μ_{n3}, σ_{n3})
(μ_{n2}, σ_{n2})
(μ_{n1}, σ_{n1})

Instance Normalization

Calculate mean and standard deviation for each individual channel for each individual samples

Batch normalization vs Instance normalization


x_1
(μ_{1c}, σ_{1c})
(μ_{13}, σ_{13})
(μ_{12}, σ_{12})
(μ_{11}, σ_{11})
x_2
(μ_{2c}, σ_{2c})
(μ_{23}, σ_{23})
(μ_{22}, σ_{22})
(μ_{21}, σ_{21})
x_3
(μ_{3c}, σ_{3c})
(μ_{33}, σ_{33})
(μ_{32}, σ_{32})
(μ_{31}, σ_{31})
x_n
(μ_{nc}, σ_{nc})
(μ_{n3}, σ_{n3})
(μ_{n2}, σ_{n2})
(μ_{n1}, σ_{n1})

Instance Normalization

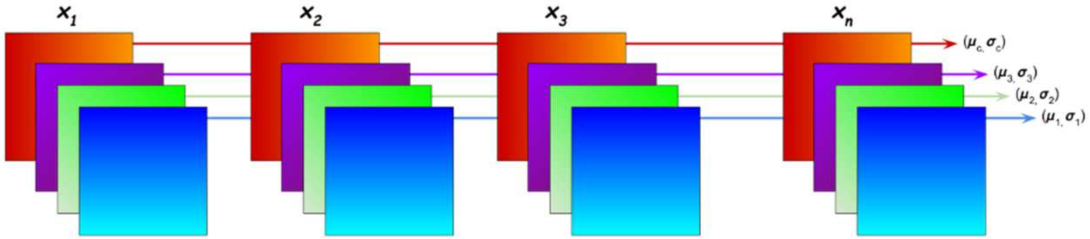
Calculate mean and standard deviation for each individual channel for each individual samples

$$\mu_{nc} = \boxed{\frac{1}{HW}\sum_{j=1}^{H}\sum_{k=1}^{W}} x_{ncjk}$$

$$\sigma_{nc}^2 = \boxed{\frac{1}{HW}\sum_{j=1}^{H}\sum_{k=1}^{W}} (x_{ncjk} - \mu_{nc})^2$$

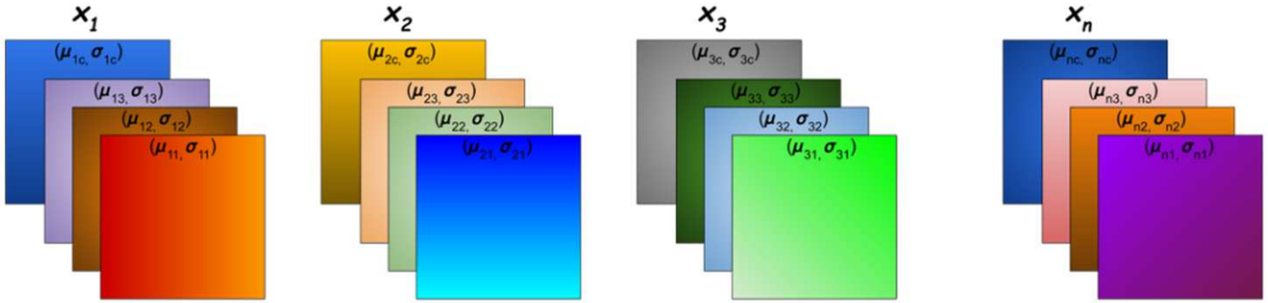
$$\hat{x} = \frac{x - \mu_{nc}}{\sqrt{\sigma_{nc}^2 + \epsilon}}$$

Batch normalization vs Instance normalization



Batch Normalization

Calculate mean and standard deviation across all samples


x_1
(μ_{1c}, σ_{1c})
(μ_{13}, σ_{13})
(μ_{12}, σ_{12})
(μ_{11}, σ_{11})
x_2
(μ_{2c}, σ_{2c})
(μ_{23}, σ_{23})
(μ_{22}, σ_{22})
(μ_{21}, σ_{21})
x_3
(μ_{3c}, σ_{3c})
(μ_{33}, σ_{33})
(μ_{32}, σ_{32})
(μ_{31}, σ_{31})
x_n
(μ_{nc}, σ_{nc})
(μ_{n3}, σ_{n3})
(μ_{n2}, σ_{n2})
(μ_{n1}, σ_{n1})

Instance Normalization

Calculate mean and standard deviation for each individual channel for each individual samples

AdaBN (Domain Adaptation)

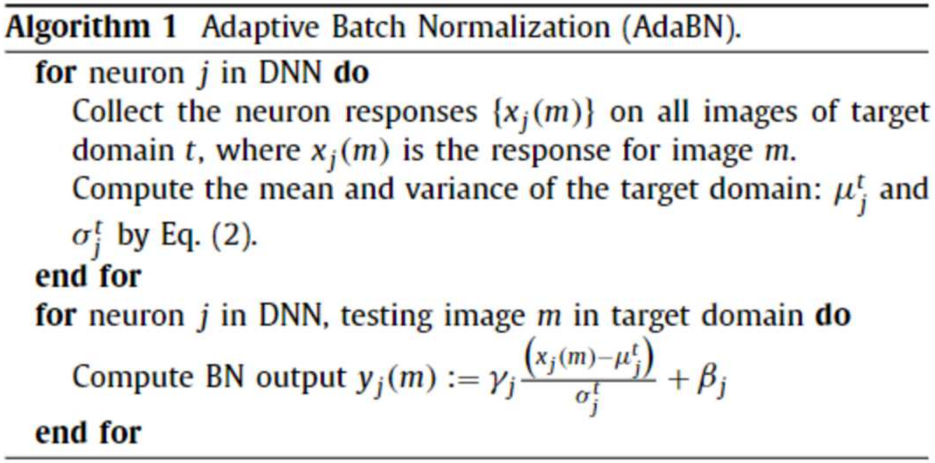
- **AdaBN**

Domain adaptation technique using Batch Normalization.

The parameters of Batch Norm are trained based on the training set.
There is a possibility of them being affected when the domain changes

As an early related work, a **method updates the parameters of Batch Normalization with data from the target domain**.

Let's remember that **domain shift is closely associated with style**



```
Algorithm 1  Adaptive Batch Normalization (AdaBN).
for neuron $j$ in DNN do
    Collect the neuron responses $\{x_j(m)\}$ on all images of target
    domain $t$, where $x_j(m)$ is the response for image $m$.
    Compute the mean and variance of the target domain: $\mu_j^t$ and
    $\sigma_j^t$ by Eq. (2).
end for
for neuron $j$ in DNN, testing image $m$ in target domain do
    Compute BN output $y_j(m) := \gamma_j \frac{\left(x_j(m)-\mu_j^t\right)}{\sigma_j^t} + \beta_j$
end for
```

Batch normalization

$$BN(X) = \gamma\left(\frac{X - \mu_{batch}}{\sigma_{batch}}\right) + \beta$$

Moving update

$$d = \mu - \mu_j,$$

$$\mu_j \leftarrow \mu_j + \frac{dk}{n_j},$$

$$\sigma_j^2 \leftarrow \frac{\sigma_j^2 n_j}{n_j + k} + \frac{\sigma^2 k}{n_j + k} + \frac{d^2 n_j k}{(n_j + k)^2},$$

$$n_j \leftarrow n_j + k,$$

[8] Li, Y., et al., Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, *2018.*

DSON

Insight from [9] related to the Batch Normalization

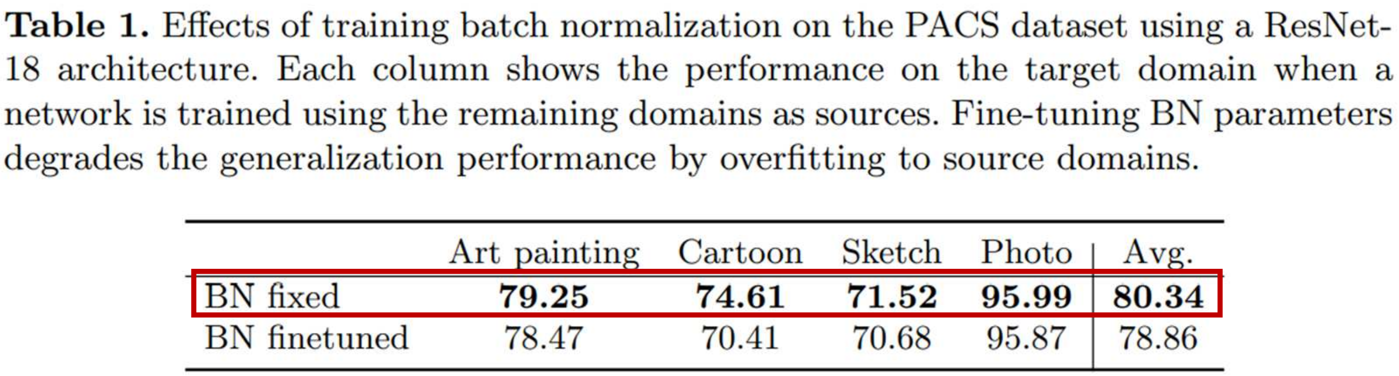


Table 1. Effects of training batch normalization on the PACS dataset using a ResNet-18 architecture. Each column shows the performance on the target domain when a network is trained using the remaining domains as sources. Fine-tuning BN parameters degrades the generalization performance by overfitting to source domains.

	Art painting	Cartoon	Sketch	Photo	Avg.
BN fixed	**79.25**	**74.61**	**71.52**	**95.99**	**80.34**
BN finetuned	78.47	70.41	70.68	95.87	78.86

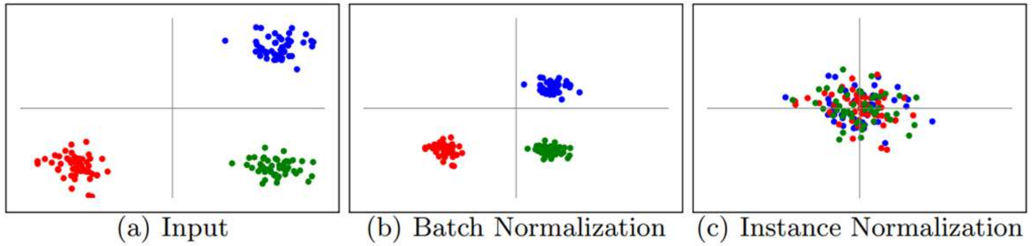
When fixing the Batch Normalization parameters in a pre-trained ResNet-18 on ImageNet, there was an improvement in generalization performance in the domain generalization task.

“This is because the batch statistics overfit to the particular training domains, resulting in poor generalization performance in unseen domains.”

In the context of the earlier mentioned materials, this is a compelling

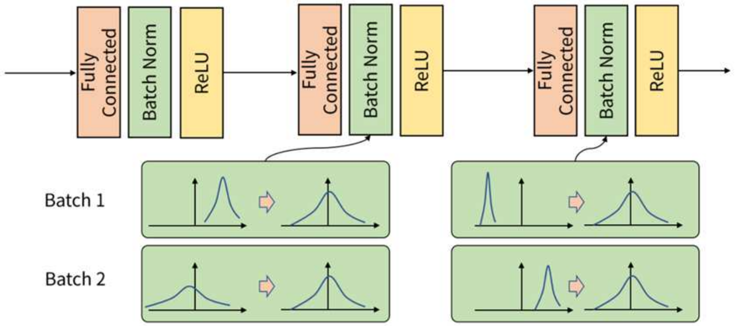
[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

DSON



(a) Input (b) Batch Normalization (c) Instance Normalization

Fig. 2. Comparing feature distributions of three classes, where color represents the class label and each dot represents a feature map with two channels. where each axis corresponds to one channel. For given (a) input activations, (c) instance normalization makes the features less discriminative over classes when compared to (b) batch normalization. Although instance normalization loses discriminability, it makes the normalized representations less overfit to a particular domain and eventually improves the quality of features when combined with batch normalization. (Best viewed in color.)



1. Batch-based → There is a probability of having data from multiple classes together

$$BN(X) = \gamma\Big(\frac{X - \mu_{batch}}{\sigma_{batch}}\Big) + \beta$$

[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

DSON

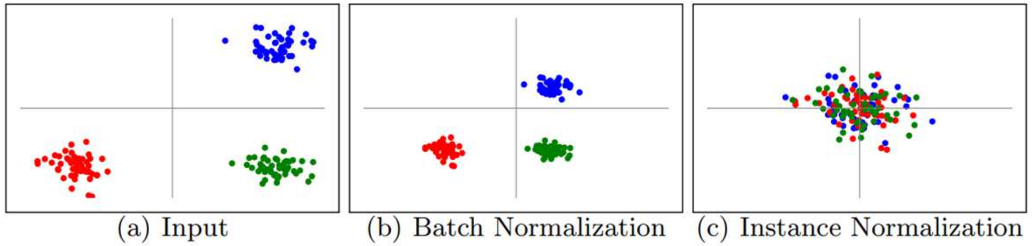
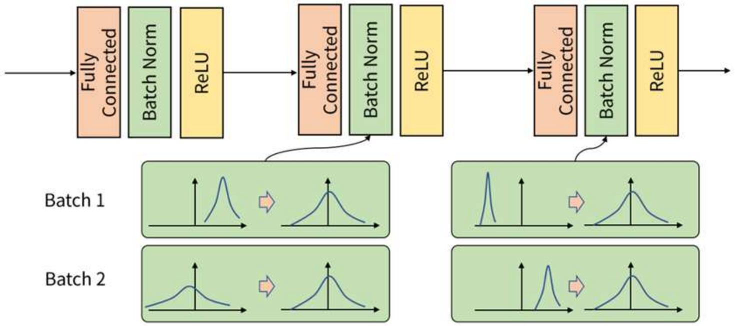

(a) Input (b) Batch Normalization (c) Instance Normalization

Fig. 2. Comparing feature distributions of three classes, where color represents the class label and each dot represents a feature map with two channels. where each axis corresponds to one channel. For given (a) input activations, (c) instance normalization makes the features less discriminative over classes when compared to (b) batch normalization. Although instance normalization loses discriminability, it makes the normalized representations less overfit to a particular domain and eventually improves the quality of features when combined with batch normalization. (Best viewed in color.)



1. Batch-based → There is a probability of having data from multiple classes together

mean/var encapsulates inter-class relationships
γ and β can manipulate inter-class relation ships, enabling adaptation to the original training domain even with slightly different data

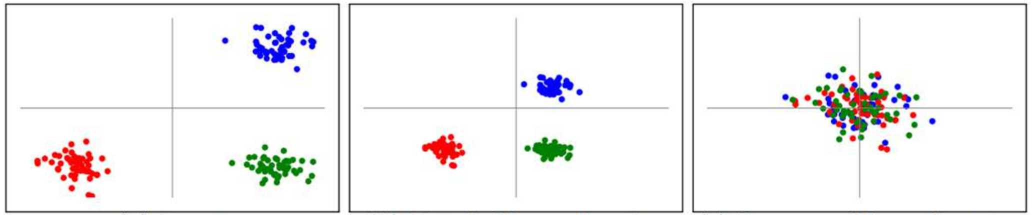
$$BN(X) = \gamma\Big(\frac{X - \mu_{batch}}{\sigma_{batch}}\Big) + \beta$$

Sampling-based method

Close to the Clustering

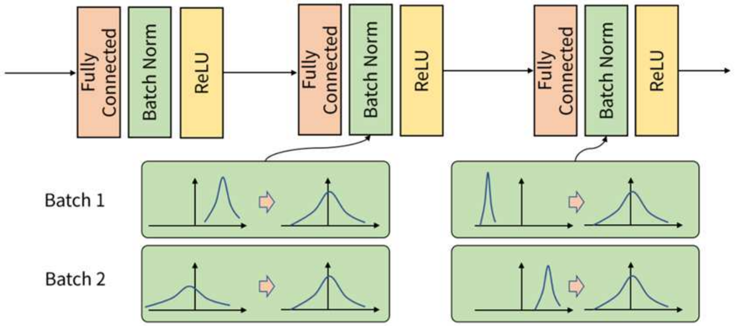
[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

DSON



(a) Input (b) Batch Normalization (c) Instance Normalization

Fig. 2. Comparing feature distributions of three classes, where color represents the class label and each dot represents a feature map with two channels. where each axis corresponds to one channel. For given (a) input activations, (c) instance normalization makes the features less discriminative over classes when compared to (b) batch normalization. Although instance normalization loses discriminability, it makes the normalized representations less overfit to a particular domain and eventually improves the quality of features when combined with batch normalization. (Best viewed in color.)



1. Batch-based → There is a probability of having data from multiple classes together

mean/var encapsulates inter-class relationships
γ and β can manipulate inter-class relation ships, enabling adaptation to the original training domain even with slightly different data

$$BN(X) = \gamma\left(\frac{X - \mu_{batch}}{\sigma_{batch}}\right) + \beta \qquad \mathrm{IN}(x) = \gamma\frac{x - \mu(x)}{\sigma(x)} + \beta,$$

Sampling-based method

Close to the Clustering

2. Instance-based → Only the channel-wise statistics within a single image (feature map) are considered.

[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

DSON

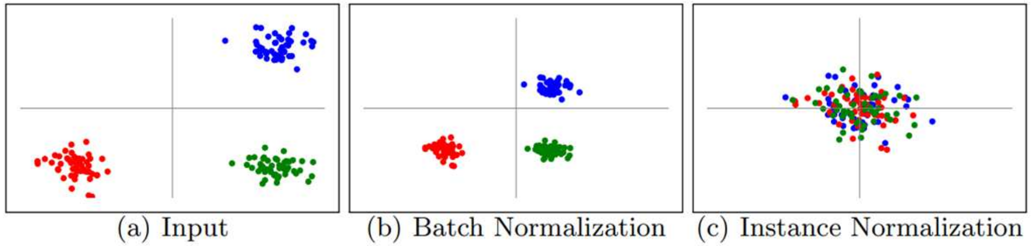
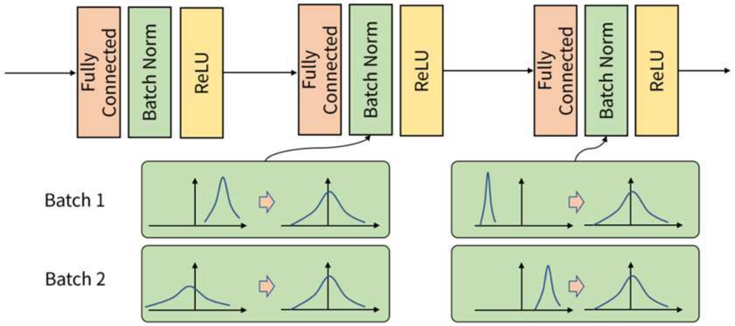


Fig. 2. Comparing feature distributions of three classes, where color represents the class label and each dot represents a feature map with two channels. where each axis corresponds to one channel. For given (a) input activations, (c) instance normalization makes the features less discriminative over classes when compared to (b) batch normalization. Although instance normalization loses discriminability, it makes the normalized representations less overfit to a particular domain and eventually improves the quality of features when combined with batch normalization. (Best viewed in color.)



1. Batch-based → There is a probability of having data from multiple classes together

mean/var encapsulates inter-class relationships
γ and β can manipulate inter-class relation ships,
enabling adaptation to the original training domain
even with slightly different data

$$BN(X) = \gamma\Big(\frac{X - \mu_{batch}}{\sigma_{batch}}\Big) + \beta$$

Sampling-based method

Close to the Clustering

$$\mathrm{IN}(x) = \gamma\frac{x - \mu(x)}{\sigma(x)} + \beta,$$

Pixel-relationship

Close to the Whitening

2. Instance-based → Only the channel-wise statistics within a single image (feature map) are considered.

mean/var represents not inter-class relationships, but
The average tendencies that constitute an image,
In other words, modeling the style of the dataset.
γ and β have the ability to manipulate certain
Tendencies(style) inherent to the dataset itself.

[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV*, 2020

DSON

- **Domain-specific Normalization : DSON (BN+IN)**

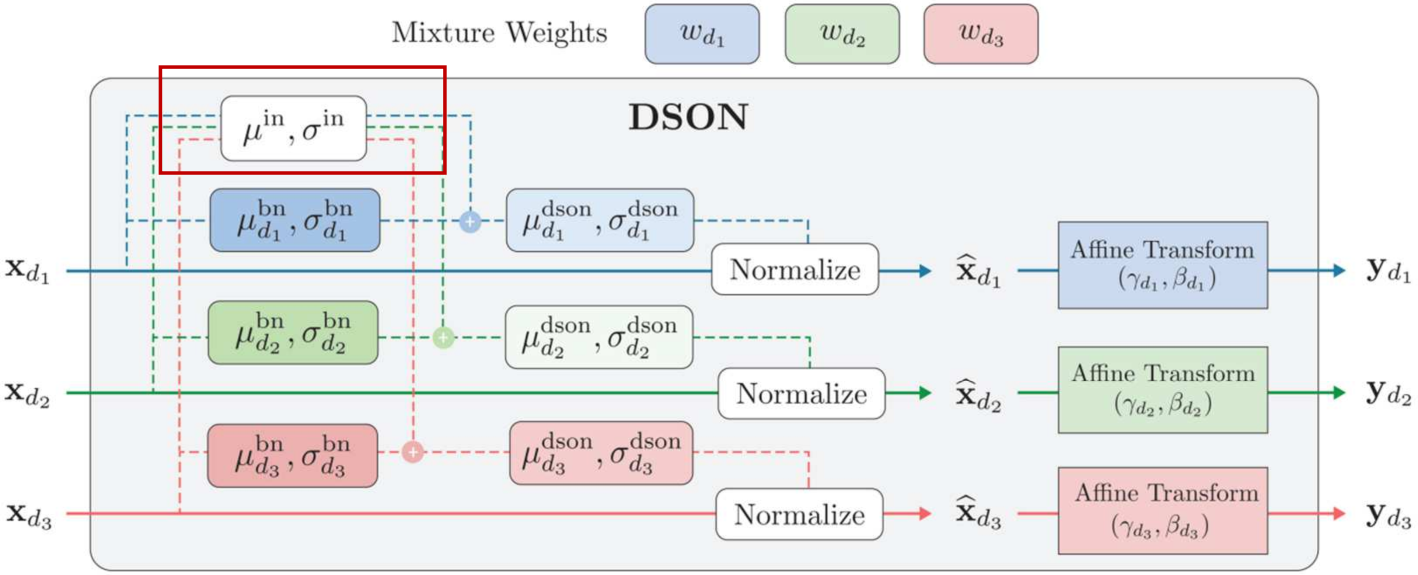

Mixture Weights
w_{d_1}
w_{d_2}
w_{d_3}
DSON
$\mu^{\text{in}}, \sigma^{\text{in}}$
$\mu^{\text{bn}}_{d_1}, \sigma^{\text{bn}}_{d_1}$
$\mu^{\text{dson}}_{d_1}, \sigma^{\text{dson}}_{d_1}$
$\mathbf{x}_{d_1}$
Normalize
$\hat{\mathbf{x}}_{d_1}$
Affine Transform $(\gamma_{d_1}, \beta_{d_1})$
$\mathbf{y}_{d_1}$
$\mu^{\text{bn}}_{d_2}, \sigma^{\text{bn}}_{d_2}$
$\mu^{\text{dson}}_{d_2}, \sigma^{\text{dson}}_{d_2}$
$\mathbf{x}_{d_2}$
Normalize
$\hat{\mathbf{x}}_{d_2}$
Affine Transform $(\gamma_{d_2}, \beta_{d_2})$
$\mathbf{y}_{d_2}$
$\mu^{\text{bn}}_{d_3}, \sigma^{\text{bn}}_{d_3}$
$\mu^{\text{dson}}_{d_3}, \sigma^{\text{dson}}_{d_3}$
$\mathbf{x}_{d_3}$
Normalize
$\hat{\mathbf{x}}_{d_3}$
Affine Transform $(\gamma_{d_3}, \beta_{d_3})$
$\mathbf{y}_{d_3}$

Fig. 1. Illustration of Domain Specific Optimized Normalization (DSON). Each domain maintains domain-specific batch normalization statistics and affine parameters, as well as mixture weights.

1. Calculate domain-specific Instance Normalization mean variance

[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

DSON

- **Domain-specific Normalization : DSON (BN+IN)**

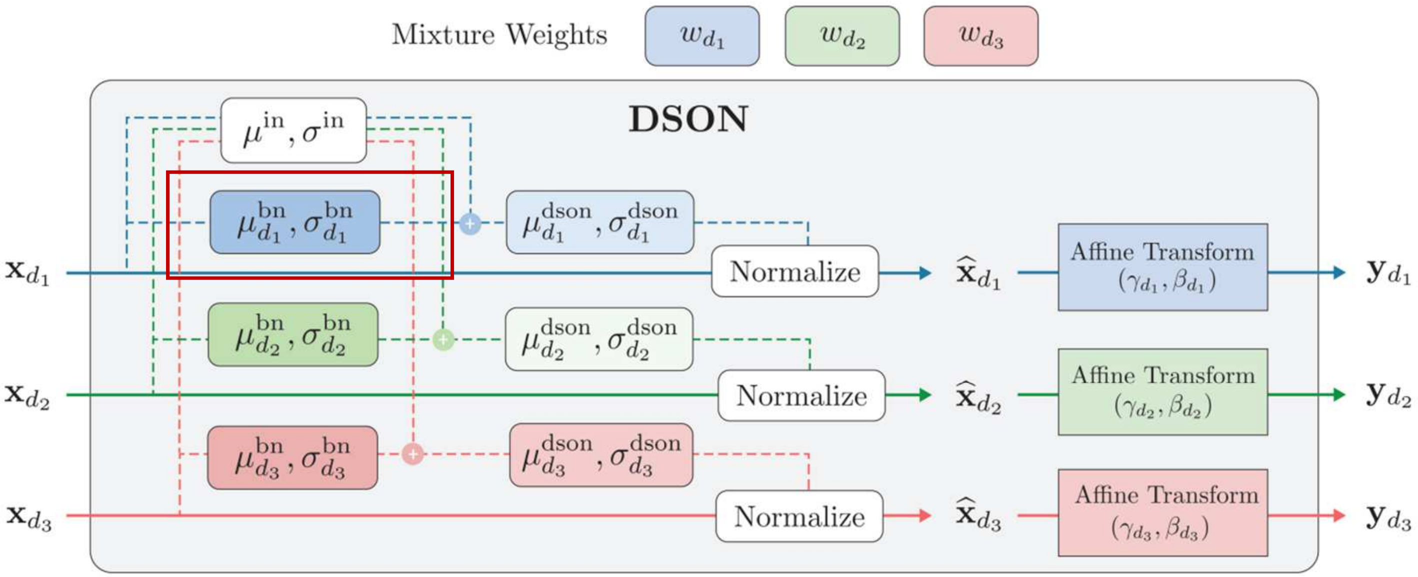

Mixture Weights
w_{d_1}
w_{d_2}
w_{d_3}
DSON
$\mu^{\text{in}}, \sigma^{\text{in}}$
$\mu_{d_1}^{\text{bn}}, \sigma_{d_1}^{\text{bn}}$
$\mu_{d_1}^{\text{dson}}, \sigma_{d_1}^{\text{dson}}$
$\mathbf{x}_{d_1}$
Normalize
$\widehat{\mathbf{x}}_{d_1}$
Affine Transform $(\gamma_{d_1}, \beta_{d_1})$
$\mathbf{y}_{d_1}$
$\mu_{d_2}^{\text{bn}}, \sigma_{d_2}^{\text{bn}}$
$\mu_{d_2}^{\text{dson}}, \sigma_{d_2}^{\text{dson}}$
$\mathbf{x}_{d_2}$
Normalize
$\widehat{\mathbf{x}}_{d_2}$
Affine Transform $(\gamma_{d_2}, \beta_{d_2})$
$\mathbf{y}_{d_2}$
$\mu_{d_3}^{\text{bn}}, \sigma_{d_3}^{\text{bn}}$
$\mu_{d_3}^{\text{dson}}, \sigma_{d_3}^{\text{dson}}$
$\mathbf{x}_{d_3}$
Normalize
$\widehat{\mathbf{x}}_{d_3}$
Affine Transform $(\gamma_{d_3}, \beta_{d_3})$
$\mathbf{y}_{d_3}$

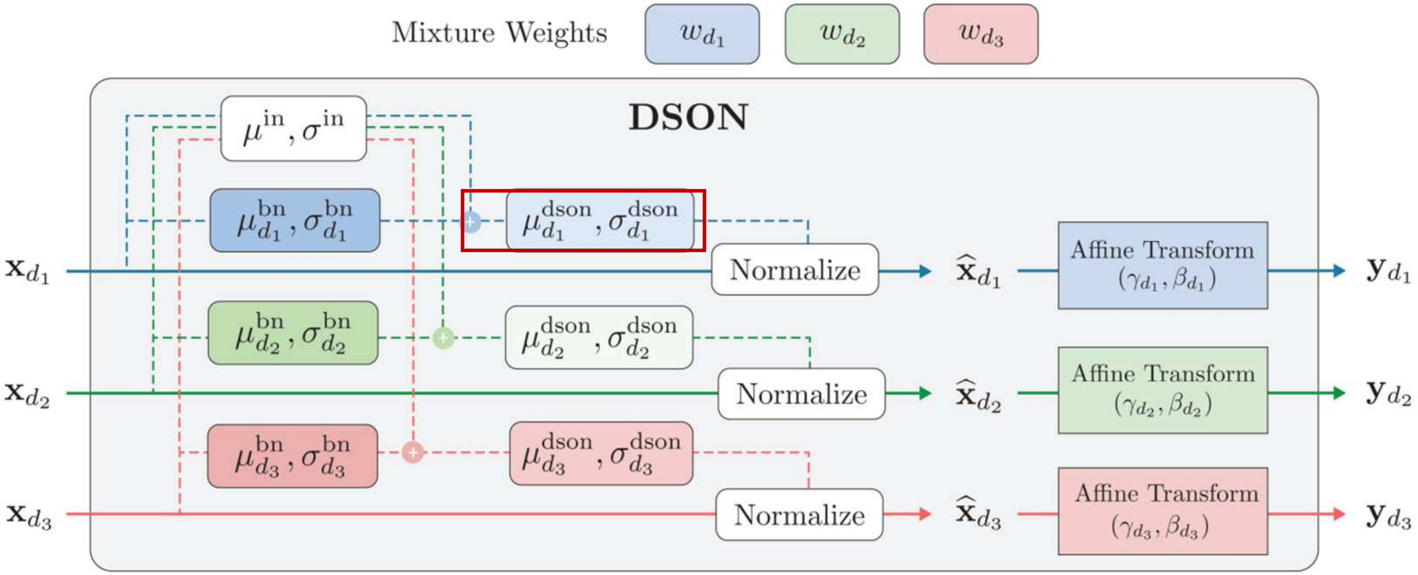
Fig. 1. Illustration of Domain Specific Optimized Normalization (DSON). Each domain maintains domain-specific batch normalization statistics and affine parameters, as well as mixture weights.

2. Calculate domain-specific Batch Normalization mean variance

[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

DSON

- **Domain-specific Normalization : DSON (BN+IN)**



Mixture Weights
w_{d_1}
w_{d_2}
w_{d_3}
DSON
$\mu^{\text{in}}, \sigma^{\text{in}}$
$\mu^{\text{bn}}_{d_1}, \sigma^{\text{bn}}_{d_1}$
$\mu^{\text{dson}}_{d_1}, \sigma^{\text{dson}}_{d_1}$
$\mathbf{x}_{d_1}$
Normalize
$\hat{\mathbf{x}}_{d_1}$
Affine Transform $(\gamma_{d_1}, \beta_{d_1})$
$\mathbf{y}_{d_1}$
$\mu^{\text{bn}}_{d_2}, \sigma^{\text{bn}}_{d_2}$
$\mu^{\text{dson}}_{d_2}, \sigma^{\text{dson}}_{d_2}$
$\mathbf{x}_{d_2}$
Normalize
$\hat{\mathbf{x}}_{d_2}$
Affine Transform $(\gamma_{d_2}, \beta_{d_2})$
$\mathbf{y}_{d_2}$
$\mu^{\text{bn}}_{d_3}, \sigma^{\text{bn}}_{d_3}$
$\mu^{\text{dson}}_{d_3}, \sigma^{\text{dson}}_{d_3}$
$\mathbf{x}_{d_3}$
Normalize
$\hat{\mathbf{x}}_{d_3}$
Affine Transform $(\gamma_{d_3}, \beta_{d_3})$
$\mathbf{y}_{d_3}$

Fig. 1. Illustration of Domain Specific Optimized Normalization (DSON). Each domain maintains domain-specific batch normalization statistics and affine parameters, as well as mixture weights.

$$\mu_{dn} = w_d \mu_d^{\text{bn}} + (1 - w_d)\mu_n^{\text{in}},$$
$$\sigma_{dn}^2 = w_d {\sigma_d^{\text{bn}}}^2 + (1 - w_d){\sigma_n^{\text{in}}}^2,$$

3. Convex combination IN and BN parameters

[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

DSON

- **Domain-specific Normalization : DSON (BN+IN)**

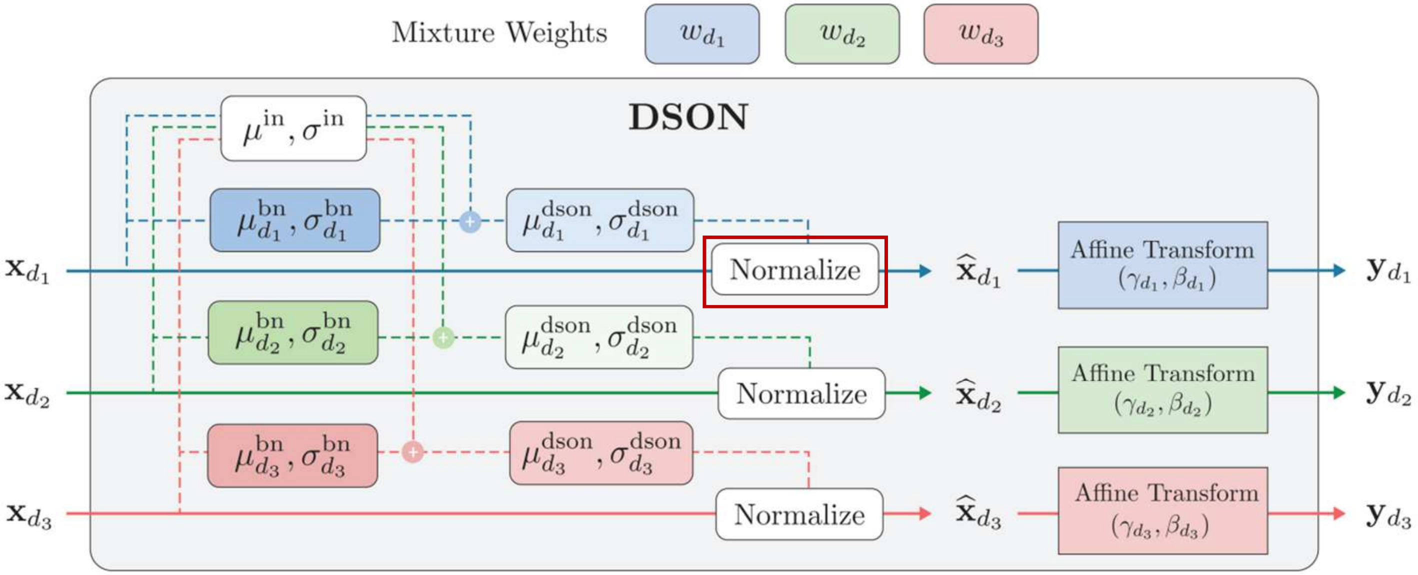

Mixture Weights
w_{d_1}
w_{d_2}
w_{d_3}
DSON
$\mu^{\text{in}}, \sigma^{\text{in}}$
$\mu^{\text{bn}}_{d_1}, \sigma^{\text{bn}}_{d_1}$
$\mu^{\text{dson}}_{d_1}, \sigma^{\text{dson}}_{d_1}$
$\mu^{\text{bn}}_{d_2}, \sigma^{\text{bn}}_{d_2}$
$\mu^{\text{dson}}_{d_2}, \sigma^{\text{dson}}_{d_2}$
$\mu^{\text{bn}}_{d_3}, \sigma^{\text{bn}}_{d_3}$
$\mu^{\text{dson}}_{d_3}, \sigma^{\text{dson}}_{d_3}$
$\mathbf{x}_{d_1}$
$\mathbf{x}_{d_2}$
$\mathbf{x}_{d_3}$
Normalize
Normalize
Normalize
$\hat{\mathbf{x}}_{d_1}$
$\hat{\mathbf{x}}_{d_2}$
$\hat{\mathbf{x}}_{d_3}$
Affine Transform $(\gamma_{d_1}, \beta_{d_1})$
Affine Transform $(\gamma_{d_2}, \beta_{d_2})$
Affine Transform $(\gamma_{d_3}, \beta_{d_3})$
$\mathbf{y}_{d_1}$
$\mathbf{y}_{d_2}$
$\mathbf{y}_{d_3}$

Fig. 1. Illustration of Domain Specific Optimized Normalization (DSON). Each domain maintains domain-specific batch normalization statistics and affine parameters, as well as mixture weights.

$$\hat{\mathbf{x}}_d[i,j,n] = \frac{\mathbf{x}_d[i,j,n] - \mu_{dn}}{\sqrt{\sigma^2_{dn} + \epsilon}}.$$

4. Domain-specific whitening

[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

DSON

- **Domain-specific Normalization : DSON (BN+IN)**

$$\mathrm{DSON}_d(\mathbf{x}_d[i,j,n];\gamma_d,\beta_d) = \gamma_d \cdot \hat{\mathbf{x}}_d[i,j,n] + \beta_d,$$

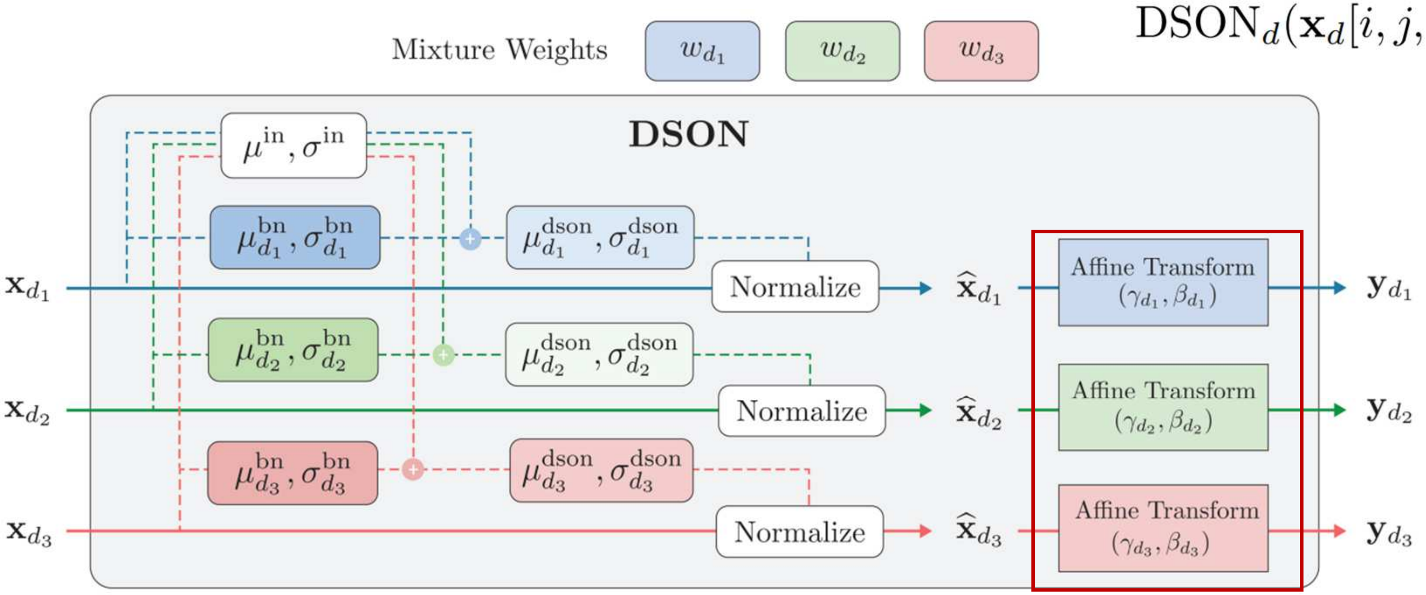

Mixture Weights
w_{d_1}
w_{d_2}
w_{d_3}
DSON
$\mu^{\text{in}}, \sigma^{\text{in}}$
$\mu^{\text{bn}}_{d_1}, \sigma^{\text{bn}}_{d_1}$
$\mu^{\text{dson}}_{d_1}, \sigma^{\text{dson}}_{d_1}$
$\mathbf{x}_{d_1}$
Normalize
$\hat{\mathbf{x}}_{d_1}$
Affine Transform ($\gamma_{d_1}, \beta_{d_1}$)
$\mathbf{y}_{d_1}$
$\mu^{\text{bn}}_{d_2}, \sigma^{\text{bn}}_{d_2}$
$\mu^{\text{dson}}_{d_2}, \sigma^{\text{dson}}_{d_2}$
$\mathbf{x}_{d_2}$
Normalize
$\hat{\mathbf{x}}_{d_2}$
Affine Transform ($\gamma_{d_2}, \beta_{d_2}$)
$\mathbf{y}_{d_2}$
$\mu^{\text{bn}}_{d_3}, \sigma^{\text{bn}}_{d_3}$
$\mu^{\text{dson}}_{d_3}, \sigma^{\text{dson}}_{d_3}$
$\mathbf{x}_{d_3}$
Normalize
$\hat{\mathbf{x}}_{d_3}$
Affine Transform ($\gamma_{d_3}, \beta_{d_3}$)
$\mathbf{y}_{d_3}$

Fig. 1. Illustration of Domain Specific Optimized Normalization (DSON). Each domain maintains domain-specific batch normalization statistics and affine parameters, as well as mixture weights.

5. Affine Transform(domain-specific learnable parameter)

[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

DSON

- **Domain-specific Normalization : DSON (BN+IN)**

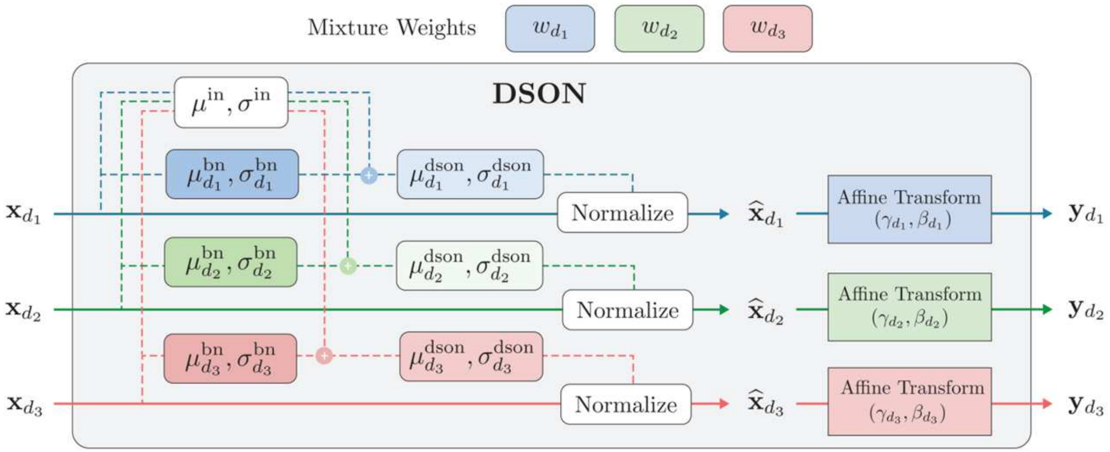

Mixture Weights
w_{d_1}
w_{d_2}
w_{d_3}
DSON
$\mu^{\mathrm{in}}, \sigma^{\mathrm{in}}$
$\mu^{\mathrm{bn}}_{d_1}, \sigma^{\mathrm{bn}}_{d_1}$
$\mu^{\mathrm{dson}}_{d_1}, \sigma^{\mathrm{dson}}_{d_1}$
$\mu^{\mathrm{bn}}_{d_2}, \sigma^{\mathrm{bn}}_{d_2}$
$\mu^{\mathrm{dson}}_{d_2}, \sigma^{\mathrm{dson}}_{d_2}$
$\mu^{\mathrm{bn}}_{d_3}, \sigma^{\mathrm{bn}}_{d_3}$
$\mu^{\mathrm{dson}}_{d_3}, \sigma^{\mathrm{dson}}_{d_3}$
$\mathbf{x}_{d_1}$
$\mathbf{x}_{d_2}$
$\mathbf{x}_{d_3}$
Normalize
Normalize
Normalize
$\hat{\mathbf{x}}_{d_1}$
$\hat{\mathbf{x}}_{d_2}$
$\hat{\mathbf{x}}_{d_3}$
Affine Transform $(\gamma_{d_1}, \beta_{d_1})$
Affine Transform $(\gamma_{d_2}, \beta_{d_2})$
Affine Transform $(\gamma_{d_3}, \beta_{d_3})$
$\mathbf{y}_{d_1}$
$\mathbf{y}_{d_2}$
$\mathbf{y}_{d_3}$

Fig. 1. Illustration of Domain Specific Optimized Normalization (DSON). Each domain maintains domain-specific batch normalization statistics and affine parameters, as well as mixture weights.

1. Domain-specific mean/var whitening

$$\hat{\mathbf{x}}_d[i,j,n] = \frac{\mathbf{x}_d[i,j,n] - \mu_{dn}}{\sqrt{\sigma^2_{dn} + \epsilon}}$$

2. BN+IN

$$\mu_{dn} = w_d \mu^{\mathrm{bn}}_d + (1 - w_d)\mu^{\mathrm{in}}_n,$$

$$\sigma^2_{dn} = w_d {\sigma^{\mathrm{bn}}_d}^2 + (1 - w_d){\sigma^{\mathrm{in}}_n}^2,$$

While maintaining the advantages of IN, it complements the inter-class variance minimization problem with BN.
BN is used for learning domain-specific statistics, which are subsequently employed for whitening (domain-specific removal).
Then, IN+BN is employed for domain-agnostic feature learning.

This process **preserves the 'semantic' while eliminating the 'style' (style whitening, domain-agnostic).**

[9] Seo, S., et al., Learning to optimize domain specific normalization for domain generalization., *ECCV,* 2020

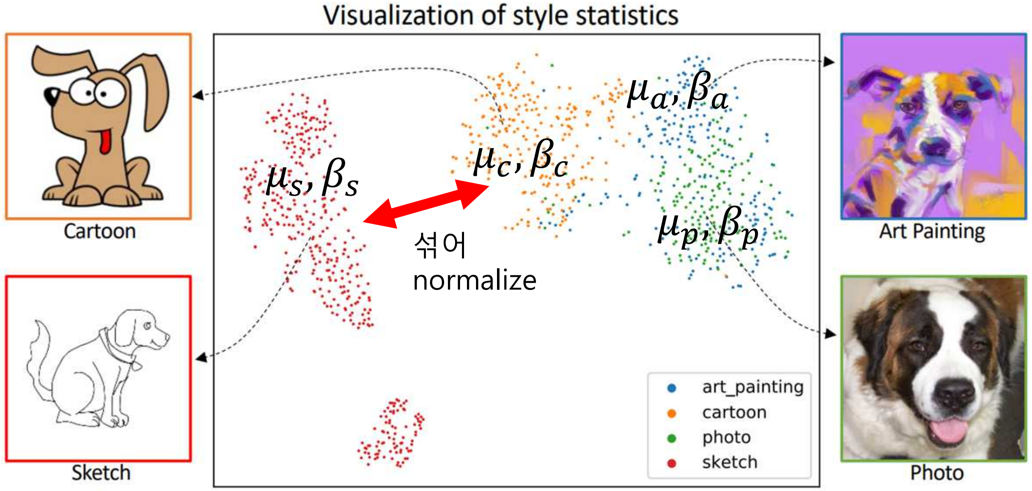
Mixstyle

1. Mix Style representation

$$\gamma_{mix} = \lambda\sigma(x) + (1-\lambda)\sigma(\tilde{x}),$$
$$\beta_{mix} = \lambda\mu(x) + (1-\lambda)\mu(\tilde{x}),$$

2. Whitening and affine transform using Mixed mean and standard deviation

$$\mathrm{MixStyle}(x) = \gamma_{mix}\frac{x-\mu(x)}{\sigma(x)} + \beta_{mix}.$$

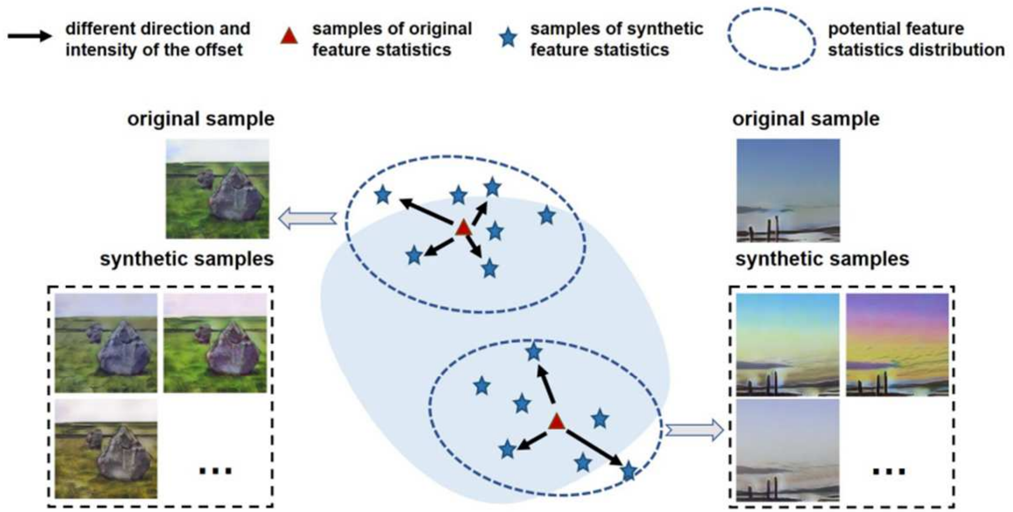


- Easy to implement
- Plug-in-play manner
- No additional computational cost in test time

[10] Kaiyang Z., et al., Domain Generalization with Mixstyle, *ICLR,* 2021

DSU

Deterministic feature statistics is not enough!



different direction and intensity of the offset
samples of original feature statistics
samples of synthetic feature statistics
potential feature statistics distribution
original sample
synthetic samples
...
original sample
synthetic samples
...

Figure 1: The visualization of reconstructed samples with synthesized feature statistics, using a pre-trained style transfer auto-encoder (Huang & Belongie, 2017). The illustration of the feature statistics shifts, which may vary in both intensity and direction (*i.e.*, different offsets in the vector space of feature statistics). We also show images of "new" domains generated by manipulating feature statistic shifts with different direction and intensity. Note these images are for visualization only, rather than feeding into the network for training.

Extracted Mean/variance is deterministic
But the target is unknown. Should be probabilistic

[11] Li, Xiaotong, et al. "Uncertainty Modeling for Out-of-Distribution Generalization." *ICLR,* 2022.

DSU

- **Previously, the style representation(domain-specific information) is modeled**
- **By deterministic values (channel-wise mean/std)**

$$\mu(x) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{b,c,h,w},$$

$$\sigma^2(x) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(x_{b,c,h,w} - \mu(x))^2.$$

$$\mathrm{IN}(x) = \gamma\frac{x - \mu(x)}{\sigma(x)} + \beta,$$

$$\gamma_{mix} = \lambda\sigma(x) + (1-\lambda)\sigma(\tilde{x}),$$
$$\beta_{mix} = \lambda\mu(x) + (1-\lambda)\mu(\tilde{x}),$$

$$\mathrm{MixStyle}(x) = \gamma_{mix}\frac{x - \mu(x)}{\sigma(x)} + \beta_{mix}.$$

Mean, std → **Deterministic!!!**
Linear manipulation

Mean, std → **Deterministic!!!**
Simply mix deterministic values

[11] Li, Xiaotong, et al. "Uncertainty Modeling for Out-of-Distribution Generalization." *ICLR,* 2022.

DSU

- **limitations**

$$\gamma_{mix} = \lambda\sigma(x) + (1-\lambda)\sigma(\tilde{x}),$$
$$\beta_{mix} = \lambda\mu(x) + (1-\lambda)\mu(\tilde{x}),$$

$$\text{MixStyle}(x) = \gamma_{mix}\frac{x-\mu(x)}{\sigma(x)} + \beta_{mix}.$$

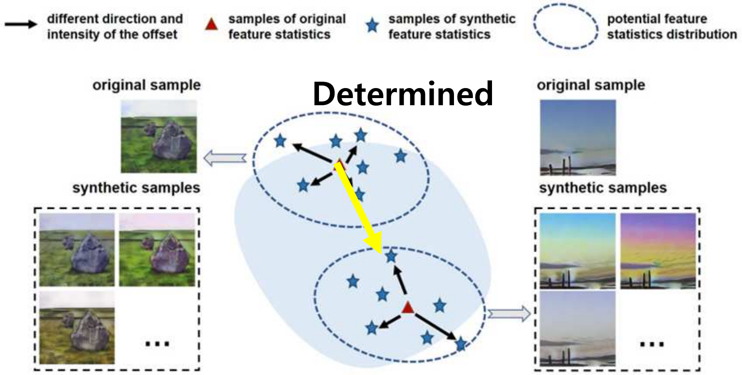

different direction and intensity of the offset
samples of original feature statistics
samples of synthetic feature statistics
potential feature statistics distribution
original sample
synthetic samples
Determined
...
original sample
synthetic samples
...

Figure 1: The visualization of reconstructed samples with synthesized feature statistics, using a pre-trained style transfer auto-encoder (Huang & Belongie, 2017). The illustration of the feature statistics shifts, which may vary in both intensity and direction (*i.e.*, different offsets in the vector space of feature statistics). We also show images of "new" domains generated by manipulating feature statistic shifts with different direction and intensity. Note these images are for visualization only, rather than feeding into the network for training.

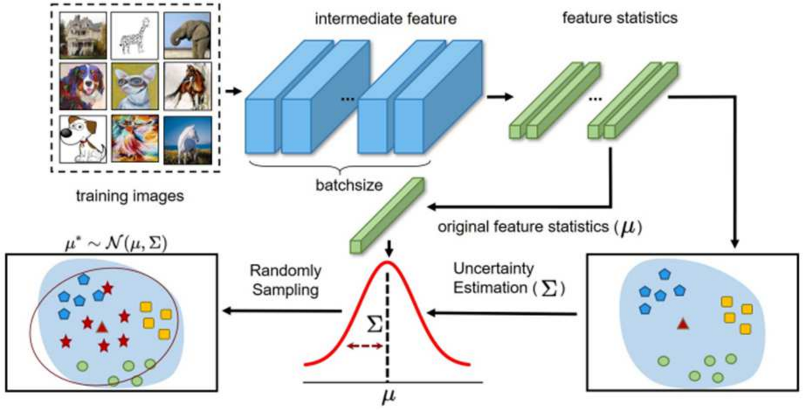
The direction of their variants is determined **By the chosen reference sample**
Their variety is sub-optimal!
(because we performed Instance Normalization)

Li, Xiaotong, et al. "Uncertainty Modeling for Out-of-Distribution Generalization." *ICLR,* 2022.

DSU

- **Uncertainty estimation**

[1] Suppose the sample's mean and std. are sampled from some normal distribution $\mathcal{N}(\mu, \Sigma_{\mu}^{2})$ $\mathcal{N}(\sigma, \Sigma_{\sigma}^{2})$



training images
intermediate feature
batchsize
feature statistics
original feature statistics (μ)
Uncertainty Estimation (Σ)
Randomly Sampling
Σ
μ
$\mu^* \sim \mathcal{N}(\mu, \Sigma)$

Figure 2: Illustration of the proposed method. Feature statistic is assumed to follow a multi-variate Gaussian distribution during training. When passed through this module, the new feature statistics randomly drawn from the corresponding distribution will replace the original ones to model the diverse domain shifts.

[11] Li, Xiaotong, et al. "Uncertainty Modeling for Out-of-Distribution Generalization." *ICLR,* 2022.

DSU

- **Uncertainty estimation**

[1] Suppose the sample's mean and std. are sampled from some normal distribution $\mathcal{N}(\mu, \Sigma_\mu^2)$ $\mathcal{N}(\sigma, \Sigma_\sigma^2)$

[2] Each variance of $\mathcal{N}(\mu, \Sigma_\mu^2)$ $\mathcal{N}(\sigma, \Sigma_\sigma^2)$ can be seen that the "uncertainty" or "variety".

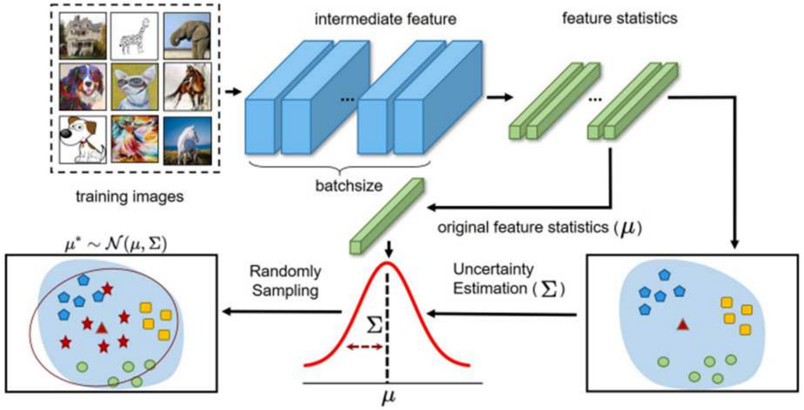

training images
intermediate feature
feature statistics
...
batchsize
original feature statistics (μ)
$\mu^* \sim \mathcal{N}(\mu, \Sigma)$
Randomly Sampling
Uncertainty Estimation (Σ)
Σ
μ

Figure 2: Illustration of the proposed method. Feature statistic is assumed to follow a multi-variate Gaussian distribution during training. When passed through this module, the new feature statistics randomly drawn from the corresponding distribution will replace the original ones to model the diverse domain shifts.

[11] Li, Xiaotong, et al. "Uncertainty Modeling for Out-of-Distribution Generalization." *ICLR,* 2022.

DSU

- **Uncertainty estimation**

[1] Suppose the sample's mean and std. are sampled from some normal distribution $\mathcal{N}(\mu, \Sigma_\mu^2)$ $\mathcal{N}(\sigma, \Sigma_\sigma^2)$

[2] Each variance of $\mathcal{N}(\mu, \Sigma_\mu^2)$ $\mathcal{N}(\sigma, \Sigma_\sigma^2)$ can be seen that the "uncertainty" or "variety".

$$\Sigma_\mu^2(x) = \frac{1}{B}\sum_{b=1}^{B}(\mu(x) - \mathbb{E}_b[\mu(x)])^2,$$

$$\Sigma_\sigma^2(x) = \frac{1}{B}\sum_{b=1}^{B}(\sigma(x) - \mathbb{E}_b[\sigma(x)])^2.$$

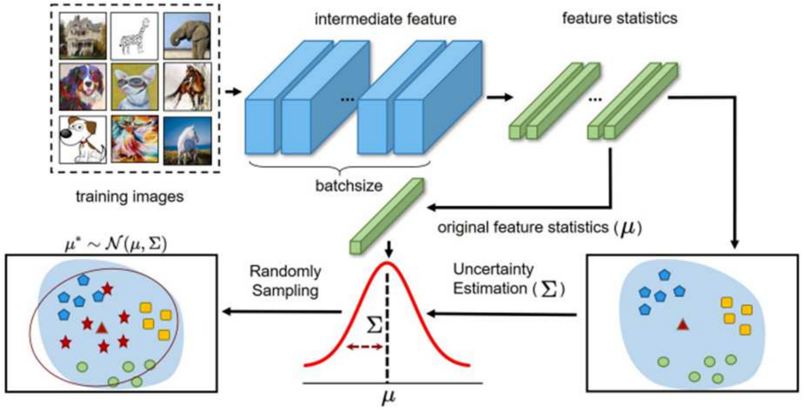

intermediate feature
feature statistics
...
...
training images
batchsize
original feature statistics (μ)
$\mu^* \sim \mathcal{N}(\mu, \Sigma)$
Randomly Sampling
Uncertainty Estimation (Σ)
Σ
μ

Figure 2: Illustration of the proposed method. Feature statistic is assumed to follow a multi-variate Gaussian distribution during training. When passed through this module, the new feature statistics randomly drawn from the corresponding distribution will replace the original ones to model the diverse domain shifts.

[11] Li, Xiaotong, et al. "Uncertainty Modeling for Out-of-Distribution Generalization." *ICLR,* 2022.

DSU

- **Uncertainty estimation**

[1] Suppose the sample's mean and std. are sampled from some normal distribution $\mathcal{N}(\mu, \Sigma_\mu^2)$ $\mathcal{N}(\sigma, \Sigma_\sigma^2)$

[2] Each variance of $\mathcal{N}(\mu, \Sigma_\mu^2)$ $\mathcal{N}(\sigma, \Sigma_\sigma^2)$ can be seen that the "uncertainty" or "variety".

$$\Sigma_\mu^2(x) = \frac{1}{B}\sum_{b=1}^{B}(\mu(x) - \mathbb{E}_b[\mu(x)])^2,$$

$$\Sigma_\sigma^2(x) = \frac{1}{B}\sum_{b=1}^{B}(\sigma(x) - \mathbb{E}_b[\sigma(x)])^2.$$

We can estimate the variances
Using Batch statistics

$$\mathrm{Var}(X) = \mathrm{E}\left[(X-\mu)^2\right]$$

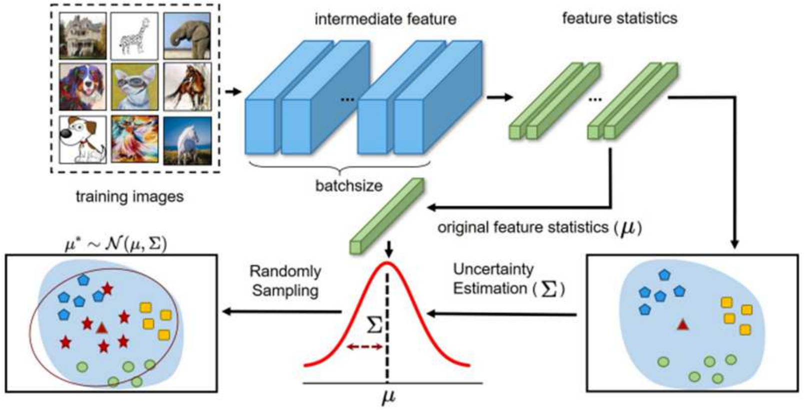

training images
intermediate feature
feature statistics
...
batchsize
original feature statistics (μ)
$\mu^* \sim \mathcal{N}(\mu, \Sigma)$
Randomly Sampling
Uncertainty Estimation (Σ)
Σ
μ

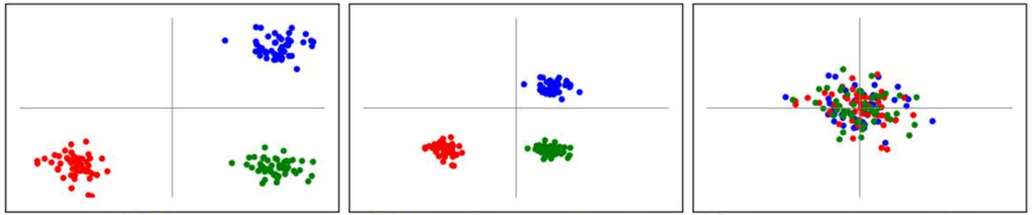
Figure 2: Illustration of the proposed method. Feature statistic is assumed to follow a multi-variate Gaussian distribution during training. When passed through this module, the new feature statistics randomly drawn from the corresponding distribution will replace the original ones to model the diverse domain shifts.

[11] Li, Xiaotong, et al. "Uncertainty Modeling for Out-of-Distribution Generalization." *ICLR,* 2022.

DSU

- **Uncertainty estimation**

The variance between features contain implicit semantic meaning and the directions with larger variances can imply potentials of more valuable semantic changes (Wang et al. 2019b)



(a) Input (b) Batch Normalization (c) Instance Normalization

Fig. 2. Comparing feature distributions of three classes, where color represents the class label and each dot represents a feature map with two channels. where each axis corresponds to one channel. For given (a) input activations, (c) instance normalization makes the features less discriminative over classes when compared to (b) batch normalization. Although instance normalization loses discriminability, it makes the normalized representations less overfit to a particular domain and eventually improves the quality of features when combined with batch normalization. (Best viewed in color.)

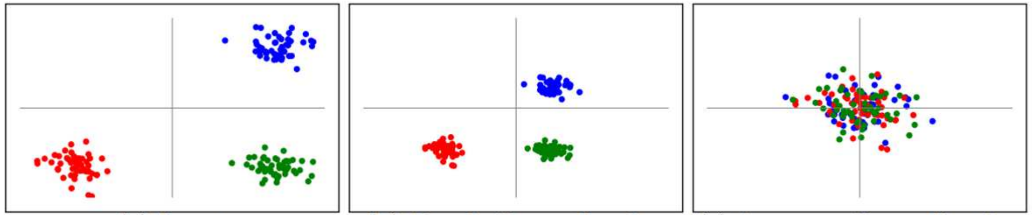
[12] Wang, Yulin, et al. "Implicit semantic data augmentation for deep networks." *Advances in Neural Information Processing Systems* 32 (2019).

DSU

- **Uncertainty estimation**

The variance between features contain implicit semantic meaning and the directions with larger variances can imply potentials of more valuable semantic changes (Wang et al. 2019b)
Although the underlying distribution of the domain shifts is unpredictable,
The uncertainty estimation captured from the mini batch can provide an appropriate and **meaningful variation rage for each feature channel**

Batch mean → class-wise mean (more semantic)
Instance mean → channel-wise mean (more domain-specific)



(a) Input (b) Batch Normalization (c) Instance Normalization

Fig. 2. Comparing feature distributions of three classes, where color represents the class label and each dot represents a feature map with two channels. where each axis corresponds to one channel. For given (a) input activations, (c) instance normalization makes the features less discriminative over classes when compared to (b) batch normalization. Although instance normalization loses discriminability, it makes the normalized representations less overfit to a particular domain and eventually improves the quality of features when combined with batch normalization. (Best viewed in color.)

[12] Wang, Yulin, et al. "Implicit semantic data augmentation for deep networks." *Advances in Neural Information Processing Systems* 32 (2019).

DSU

- DSU

Add perturbations with certain range ($\mathcal{N}(\mu, \Sigma_\mu^2)$ $\mathcal{N}(\sigma, \Sigma_\sigma^2)$)

$$\beta(x) = \mu(x) + \boxed{\epsilon_\mu \Sigma_\mu(x)}, \qquad \epsilon_\mu \sim \mathcal{N}(\mathbf{0}, \mathbf{1}),$$

$$\gamma(x) = \sigma(x) + \boxed{\epsilon_\sigma \Sigma_\sigma(x)}, \qquad \epsilon_\sigma \sim \mathcal{N}(\mathbf{0}, \mathbf{1}).$$

Now we can sample the
Mean and std from the
Estimated gaussian distribution!!!

$$\mathrm{DSU}(x) = \underbrace{(\sigma(x) + \epsilon_\sigma \Sigma_\sigma(x))}_{\gamma(x)} \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \underbrace{(\mu(x) + \epsilon_\mu \Sigma_\mu(x))}_{\beta(x)}.$$

DSU

- DSU vs Mixstyle

$$\mathrm{DSU}(x) = \boxed{\underbrace{(\sigma(x) + \epsilon_\sigma \Sigma_\sigma(x))}_{\gamma(x)}} \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \boxed{\underbrace{(\mu(x) + \epsilon_\mu \Sigma_\mu(x))}_{\beta(x)}}.$$

sampling

$$\mathrm{MixStyle}(x) = \boxed{\gamma_{mix}} \frac{x - \mu(x)}{\sigma(x)} + \boxed{\beta_{mix}}.$$

deterministic

Following presentation

- Set up the experiment
- Set up the previous approach's disadvantages
- Acquire baseline experimental result
- Set up the future work

UST seminar

Towards domain-agnostic Video action recognition

Hyungmin Kim
UST-ETRI
Ph.D. student
khm159@etri.re.kr/ust.ac.kr

26th Oct, 2023