Domain Generalization using Whitening Transform

곽 재 호
Vision and Display System Lab.
Sogang University

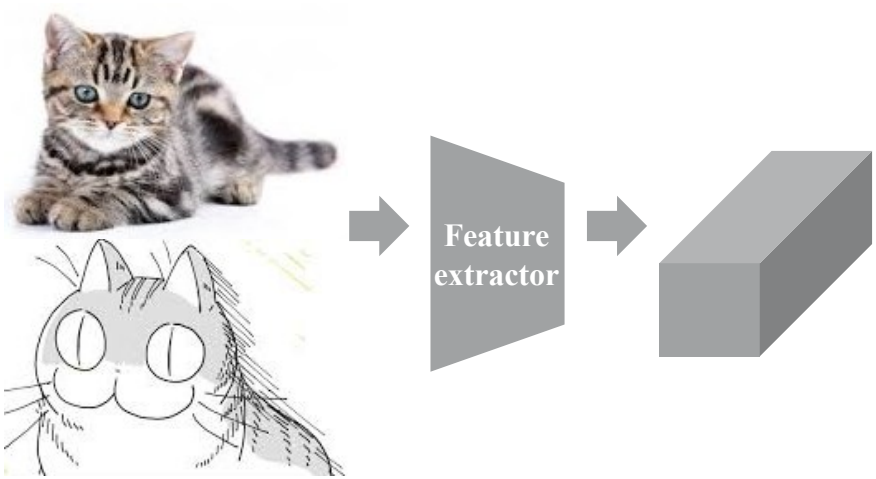
Outline

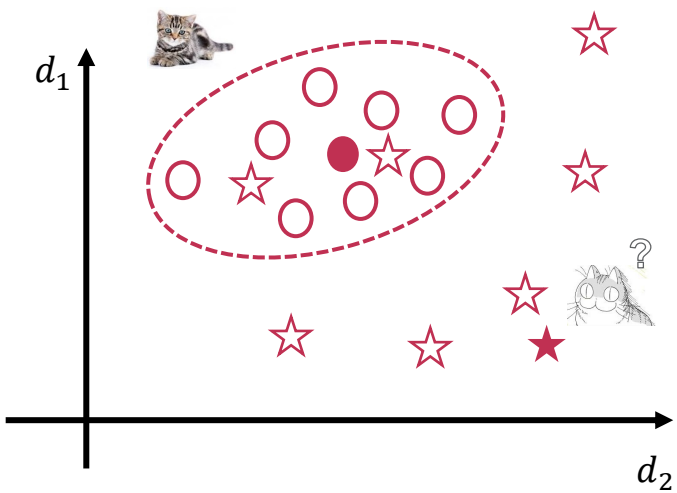
- Background
  - Domain generalization
  - Domain generalization methods
  - Whitening transform

- RobustNet
  - RobustNet: Improving Domain Generalization in Urban-Scene Segmentation via Instance Selective Whitening (CVPR 2021)

Background

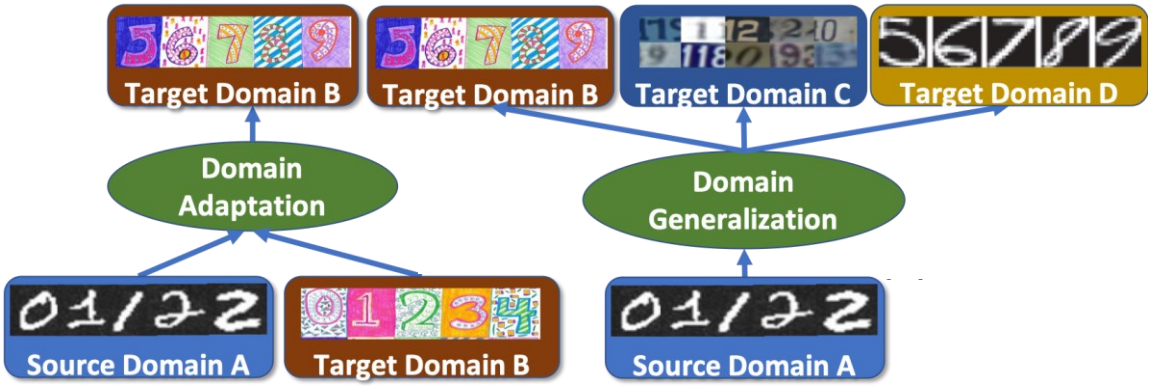
- Domain generalization
  - Domain shift
    - Change in the data distribution between an algorithm's training dataset, and a dataset it encounters when deployed
    - The domain used to train the model is called ‘source domain’, and the domain to be applied is called ‘target domain’
    - It is impossible to model the distribution of all data in the world with a limited dataset





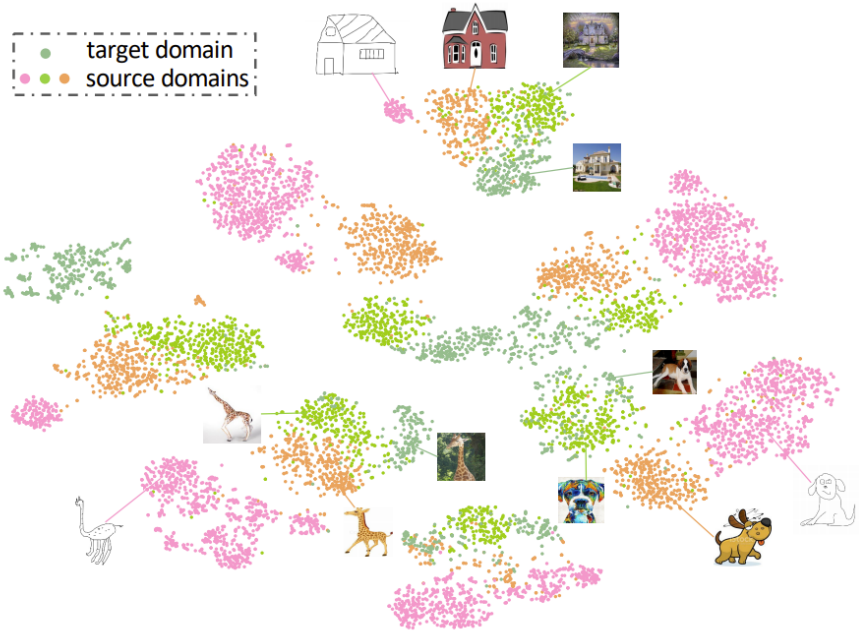
Background

- Domain generalization
  - Domain adaptation
    - DA focuses on adapting the source domain distribution to that of the specific target domain
      - Such as GTAV to Cityscapes or real world to cartoon etc..
      - It requires access to the samples in the target domain
  - Domain generalization
    - DG task sets the entire real world as a target domain
      - It does not require access to the samples in the target domain



Background

- Domain generalization methods
  - [1] Learning a shared representation across multiple source domains
    - Collecting such multi-domain datasets is costly and labor-intensive
    - The performance highly depends on the number of source datasets



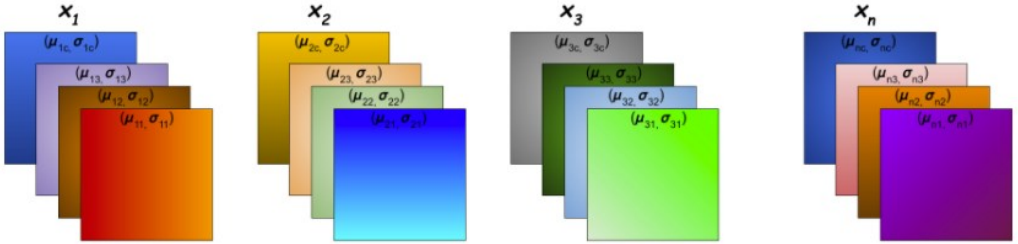
Background

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

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

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

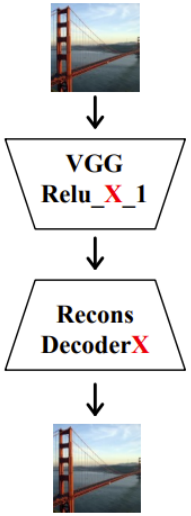
- Domain generalization methods
  - Instance normalization
    - Simple appearance variations such as color or brightness shift could simply be eliminated by normalizing each RGB channel of an image with its mean and standard deviation
    - [1] IBN-Net shows a significant performance improvement with incorporating the IN layers through training on a single source domain
    - Instance normalization just standardizes features while not considering the correlation between channels

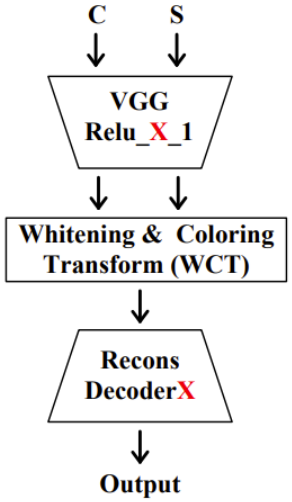


[1] Texture synthesis using convolutional neural networks. In Advances in Neural Information Processing Systems (NeurIPS), 2015
[2] Image style transfer using convolutional neural networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016
[3] Universal style transfer via feature transforms. In Advances in Neural Information Processing Systems (NeurIPS), 2017
[4] RobustNet: Improving Domain Generalization in Urban-Scene Segmentation via Instance Selective Whitening. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021

Background

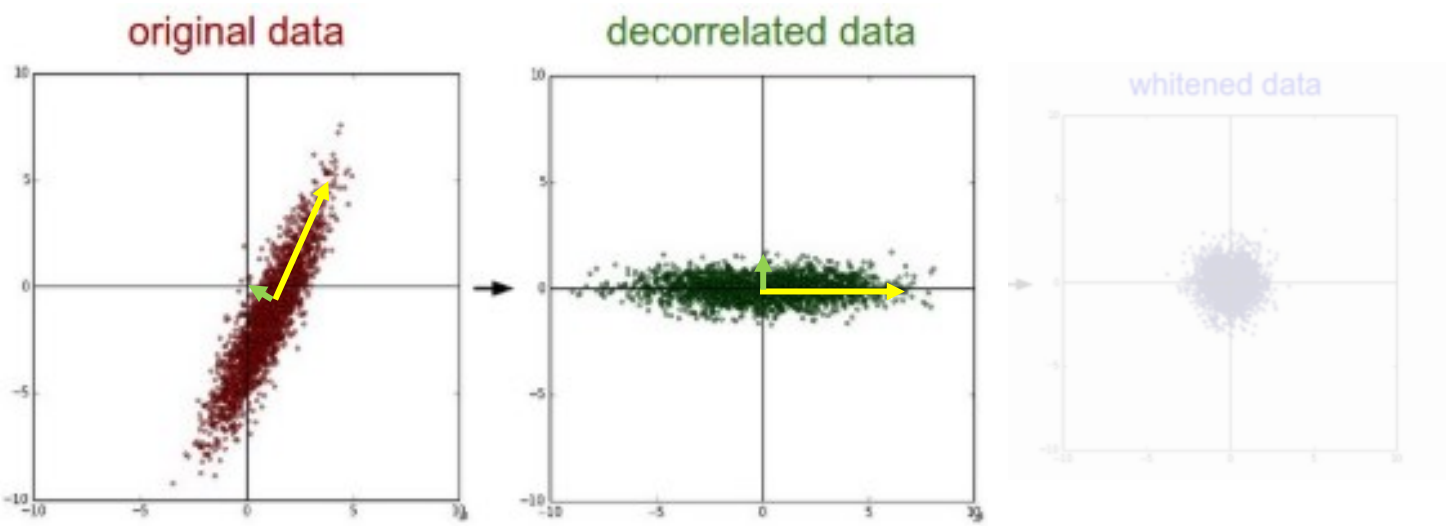
- Domain generalization methods
  - Feature whitening
    - [1, 2] Feature correlation contains domain-specific style information such as texture and color
    - [3] Whitening transform effectively eliminates domain-specific style information
    - **[4] Decoupling style and content and selectively removing the domain-specific style**





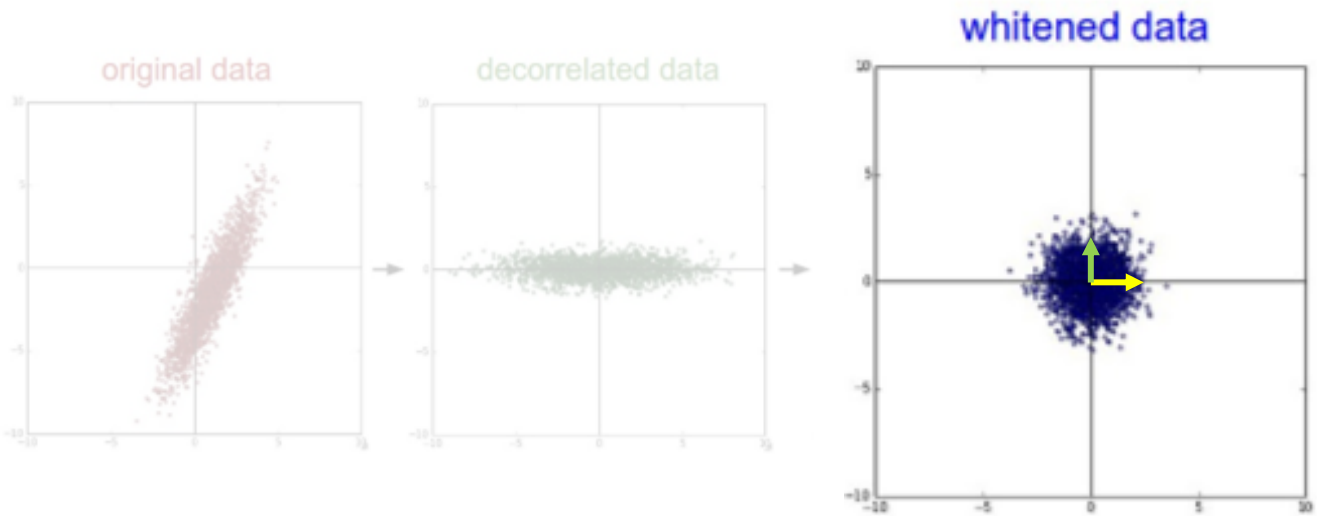
Background

- Whitening transform
  - Principal component analysis
    - Perform eigenvalue decomposition on the original data to find the eigen-basis
      - Each is a green and yellow arrow
    - Project all data into a space composed of the eigen-basis
      - All features are decorrelated



Background

- Whitening transform
  - Whitening
    - Divide each eigenvalue corresponding to the basis vector of all data samples
      - The covariance of whitened data distribution becomes **I**
      - Each feature will have the same influence



Background

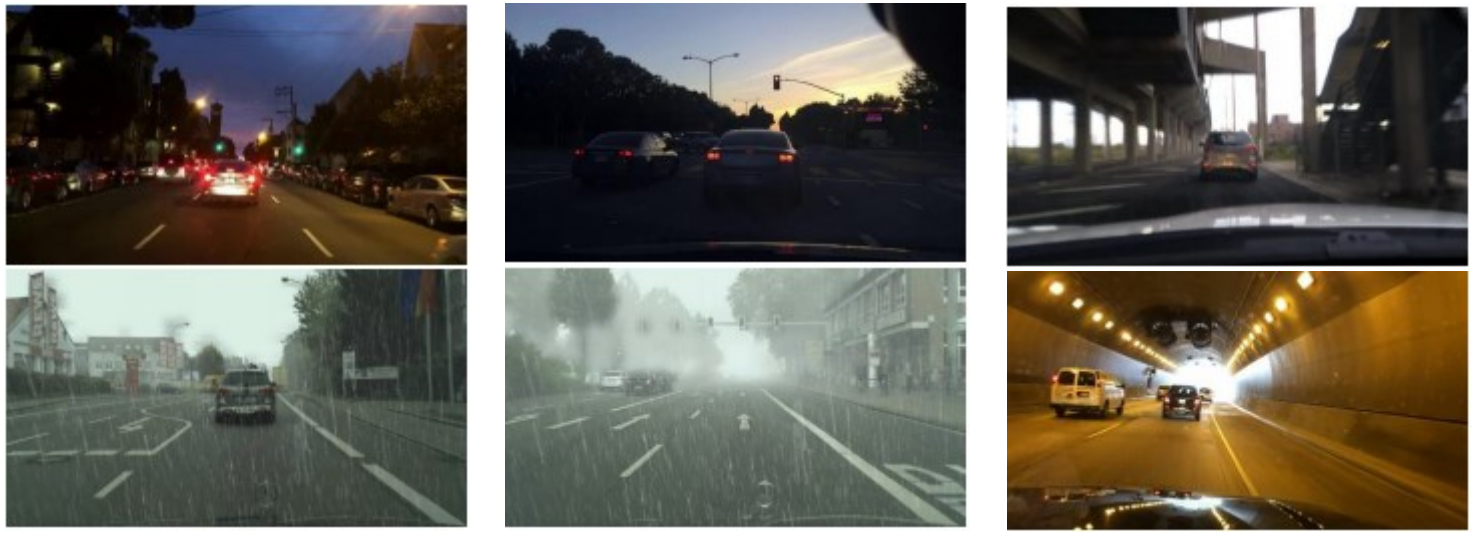
- Whitening transform
  - Limitation of WT
    - Eigenvalue decomposition is computationally expensive
      - Leading to slow training and inference speed
    - Eigenvalue decomposition is also difficult to backpropagate the gradient signal
    - The goal of WT can be achieved without the eigen-decomposition
      - [1] Through the whitening loss

$$\mathcal{L}_{\mathrm{DWT}} = \mathbb{E}[\|\mathbf{\Sigma}_{\mu} - \mathbf{I}\|_1]$$

        - ✓ Deep whitening transformation let the network naturally encode the whitened feature
      - [2] Through approximating the whitening transformation matrix using Newton's iteration

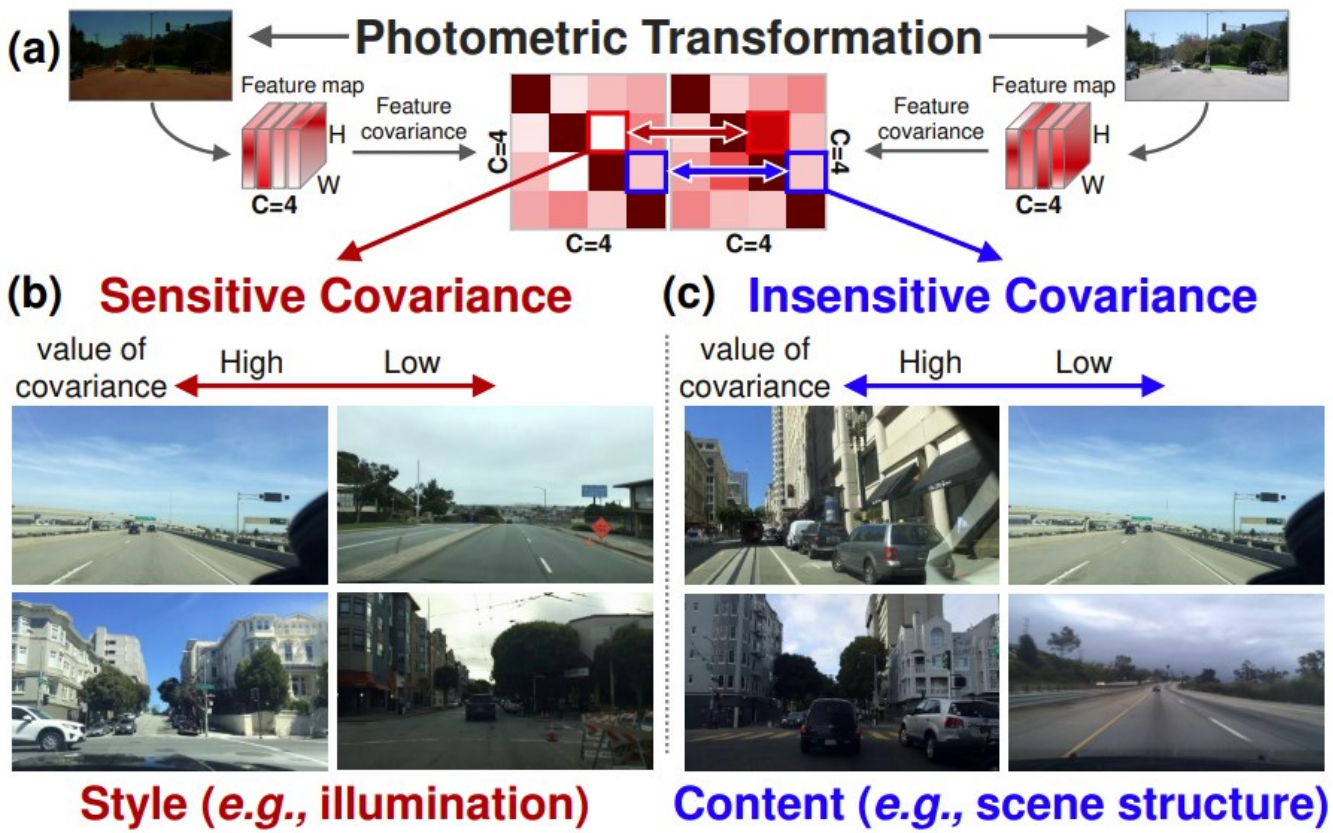
RobustNet

- RobustNet: Improving Domain Generalization in Urban-Scene Segmentation via Instance Selective Whitening (CVPR 2021 oral)
  - Provides reasonable predictions for unseen domain images
    - Low-illuminated, rainy, unseen structures
  - **Decouples the domain specific style and domain invariant content feature covariance**
  - **Selectively removes only the style information causing domain shift**



RobustNet

- Proposed method



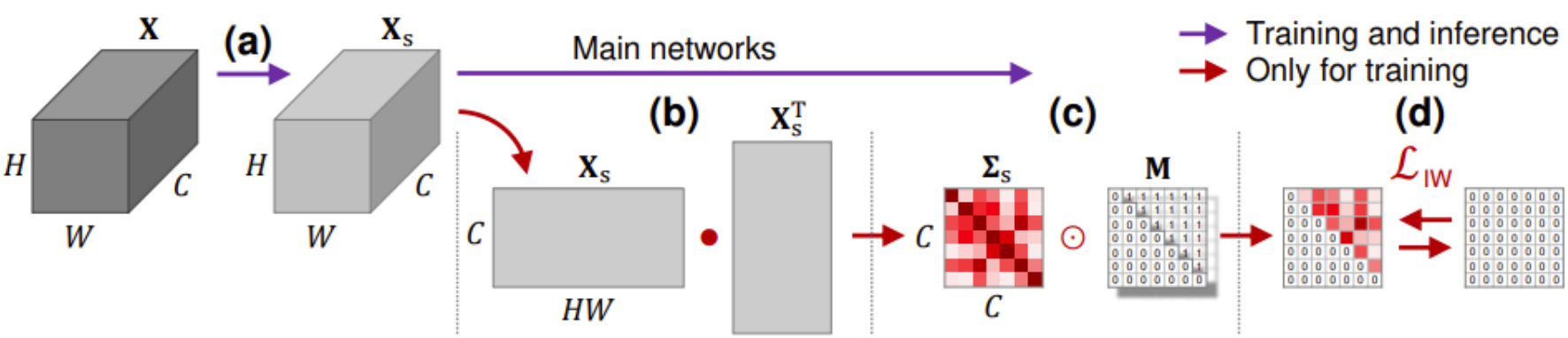
RobustNet

$$\boldsymbol{\mu} = \frac{1}{HW}\mathbf{X}\cdot\mathbf{1} \in \mathbb{R}^{C\times 1}$$
$$\boldsymbol{\Sigma}_\mu = \frac{1}{HW}\left(\mathbf{X}-\boldsymbol{\mu}\cdot\mathbf{1}^\top\right)\left(\mathbf{X}-\boldsymbol{\mu}\cdot\mathbf{1}^\top\right)^\top \in \mathbb{R}^{C\times C}$$

- Proposed method

$$\mathcal{L}_{\mathrm{DWT}} = \mathbb{E}[\|\boldsymbol{\Sigma}_\mu - \mathbf{I}\|_1]$$

  - Instance whitening loss
    - For diagonal element $\|\boldsymbol{\Sigma}_{\mu\,(i,i)}-1\|_1 = \|\frac{\mathbf{x}_i^\top\cdot\mathbf{x}_i}{HW}-1\|_1 = \|\frac{|\mathbf{x}_i||\mathbf{x}_i|\cos 0^\circ}{HW}-1\|_1$ (1)
    - For off-diagonal element $\|\boldsymbol{\Sigma}_{\mu\,(i,j)}\|_1 = \|\frac{\mathbf{x}_i^\top\cdot\mathbf{x}_j}{HW}\|_1 = \|\frac{|\mathbf{x}_i||\mathbf{x}_j|\cos\theta}{HW}\|_1$ (2)
    - Both loss terms conflict with each other
      - Diagonal element (1) makes $x_i \to \sqrt{HW}$, off-diagonal element (2) makes $x_i \to 0$



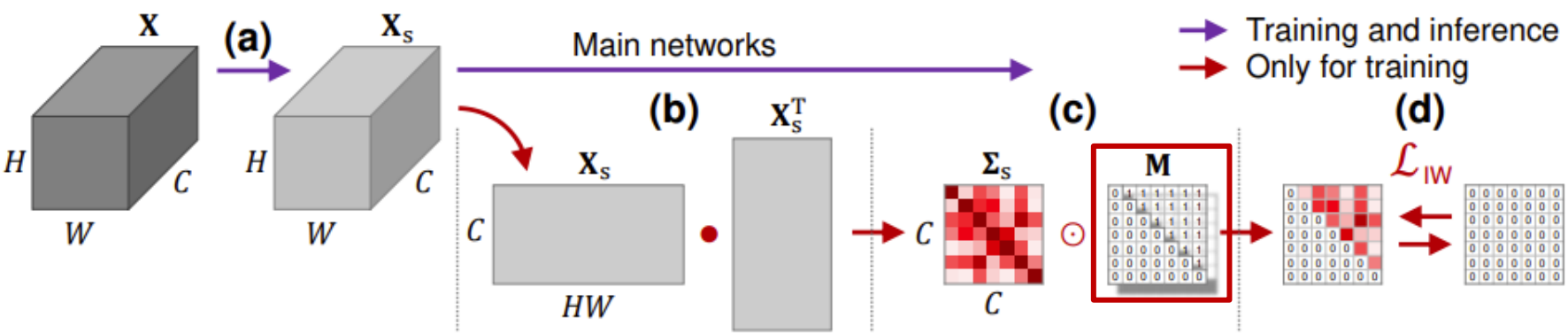
RobustNet

- Proposed method
  - Instance whitening loss
    - [1] Instance normalization could address this issue

$$\mathbf{X_s} = (\mathrm{diag}(\mathbf{\Sigma}_\mu))^{-\frac{1}{2}} \odot (\mathbf{X} - \boldsymbol{\mu} \cdot \mathbf{1}^\top)$$

    - After the standardization process, diagonal elements of the covariance matrix are already set as 1
      - We just need to make off-diagonals of the covariance matrix close to zero
    - Since the covariance matrix is symmetric, the loss can be applied only to the strict upper triangular part

$$\mathcal{L}_{\mathrm{IW}} = \mathbb{E}[\|\mathbf{\Sigma_s} \odot \mathbf{M}\|_1]$$



RobustNet

- Proposed method
  - Margin-based relaxation of whitening loss
    - The instance whitening loss suppresses all covariance elements to zero

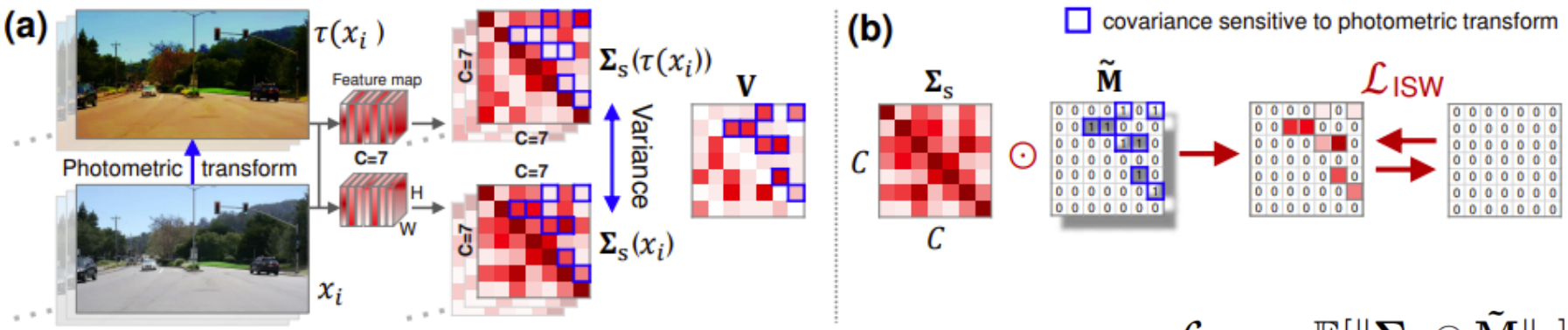
$$\mathcal{L}_{\mathrm{IW}} = \mathbb{E}[\|\mathbf{\Sigma_s} \odot \mathbf{M}\|_1]$$

    - It can adversely affect the discriminative power of features within DNNs
      - Instance-relaxed whitening (IRW) loss maintains the discriminative power

$$\mathcal{L}_{\mathrm{IRW}} = \max(\mathbb{E}[\|\mathbf{\Sigma_s} \odot \mathbf{M}\|_1] - \delta,\ 0)$$

RobustNet

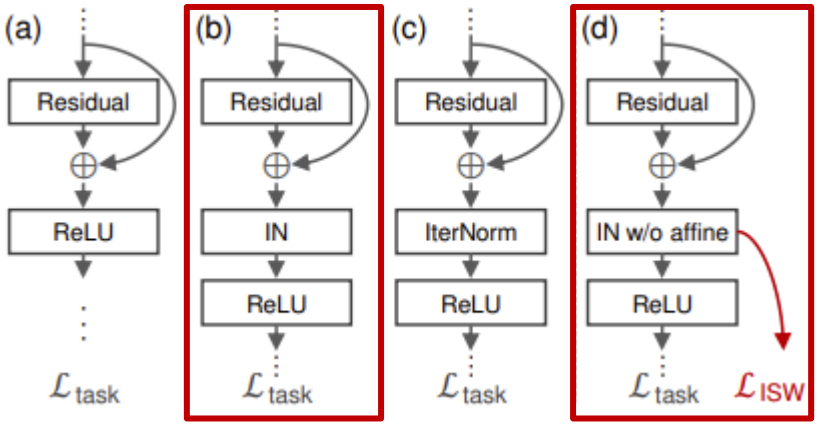
- Proposed method
  - Separating Covariance Elements
    - Compute the variance matrix V out of the covariance matrices of the two images
      - Each is i -th image x_i and its photometric transformed image τ (x_i)
      - Photometric transformation τ consists of changes in style, such as color jitter and gaussian blur
    - Identify those elements sensitive to the transformation (blue boxes)
    - Mask the covariance matrix $\mathbf{\Sigma}_s$ by the matrix $\widetilde{M}$ to suppress style-sensitive covariances by $\mathcal{L}_{\text{ISW}}$

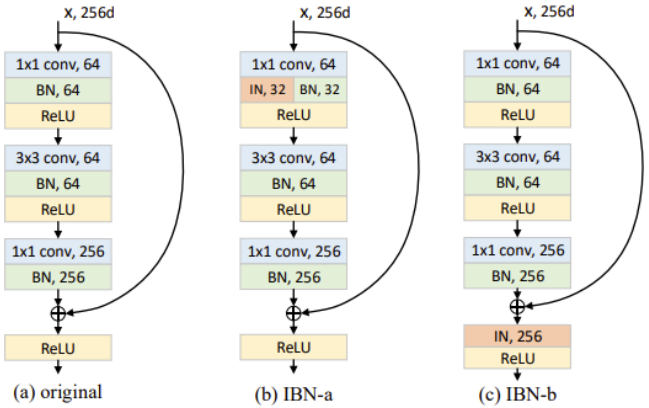


$$\mathcal{L}_{\text{ISW}} = \mathbb{E}[\|\mathbf{\Sigma}_s \odot \tilde{\mathbf{M}}\|_1]$$

RobustNet

- Proposed method
  - Network architecture with proposed ISW loss
    - Followed the architecture of [1] IBN-b which combines instance normalization with batch normalization
      - Adds an instance normalization layer right after the addition operation of a residual block
      - add three instance normalization layers after the first three convolution groups
    - loss in total is described as $$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda\left(\frac{1}{L}\sum_{i}^{L}\mathcal{L}_{\text{ISW}}^{i}\right)$$
    - λ denotes the weight of $\mathcal{L}_{\text{ISW}}$ and is empirically set to 0.6





RobustNet

- Experiments
  - Set DeepLabv3+ as a baseline model
  - Each table shows the generalization performance of the models trained on GTAV dataset, Cityscapes dataset
    - ISW shows a significant improvement on real-world datasets such as Cityscapes, BDD-100K, Mapillary
    - ISW sacrifices the performance on the source domains, but shows good generalizability
    - Baseline, SW, IBN-Net tend to overfit the source domain

Models (GTAV)	C	B	M	S	G
Baseline	28.95	25.14	28.18	26.23	73.45
†SW [43]	29.91	27.48	29.71	27.61	**73.50**
†IBN-Net [42]	33.85	32.30	37.75	27.90	72.90
†IterNorm [21]	31.81	32.70	33.88	27.07	73.19
Ours (IW)	33.21	32.67	37.35	27.57	72.06
Ours (IRW)	33.57	33.18	38.42	27.29	71.96
Ours (ISW)	**36.58**	**35.20**	**40.33**	**28.30**	72.10

Models (Cityscapes)	B	M	G	S	C
Baseline	44.96	51.68	42.55	23.29	**77.51**
†SW [43]	48.49	55.82	44.87	26.10	77.30
†IBN-Net [42]	48.56	57.04	45.06	26.14	76.55
†IterNorm [21]	49.23	56.26	**45.73**	25.98	76.02
Ours (IW)	48.19	58.90	45.21	25.81	76.06
Ours (IRW)	48.67	**59.20**	45.64	26.05	76.13
Ours (ISW)	**50.73**	58.64	45.00	**26.20**	76.41

* Cityscapes (C), BDD-100K (B), Mapillary (M), SYNTHIA (S), GTAV (G)

RobustNet

- Experiments
  - Table left shows the wide applicability of the proposed method
    - First group (top 3 rows) is reported by adopting ShuffleNetV2, and the second group (bottom 3 rows) is using MobileNetV2 as backbone networks
    - In both cases, ISW outperforms the baseline and IBN-Net on real-world datasets
  - Table right presents the comparison with baselines trained on multiple synthetic domains
    - The backbone is ResNet-50 with an output stride of 16

Models (GTAV)	C	B	M	S	G
Baseline	25.56	22.17	28.60	23.33	**66.47**
†IBN-Net [42]	27.10	31.82	34.89	**25.56**	65.44
Ours (ISW)	**30.98**	**32.06**	**35.31**	24.31	64.99
Baseline	25.92	25.73	26.45	24.03	**68.12**
†IBN-Net [42]	30.14	27.66	27.07	**24.98**	67.66
Ours (ISW)	**30.86**	**30.05**	**30.67**	24.43	67.48

Models (G + S)	C	B	M	G	S
Baseline	35.46	25.09	31.94	68.48	67.99
IBN-Net	35.55	32.18	38.09	**69.72**	66.90
Ours	**37.69**	**34.09**	**38.49**	68.26	**68.77**

* Cityscapes (C), BDD-100K (B), Mapillary (M), SYNTHIA (S), GTAV (G)

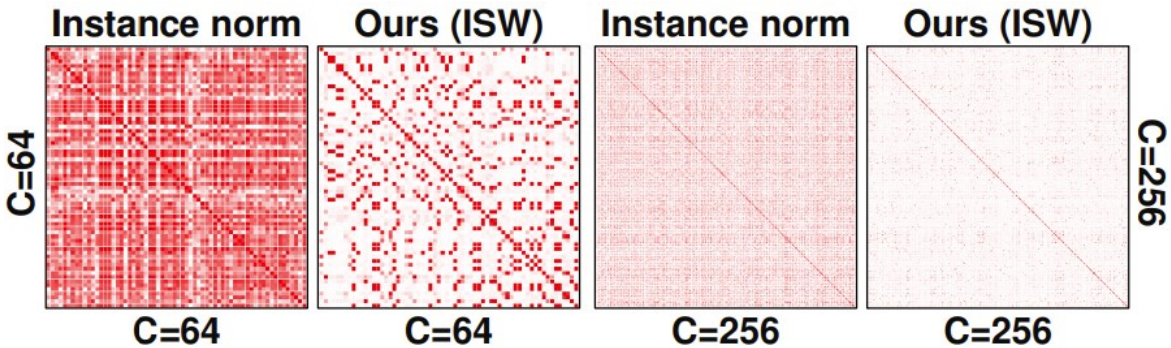
RobustNet

- Experiments
  - Table below shows the comparison of computational cost
    - Tested with the image size of 2048×1024 on NVIDIA A100 GPU
    - Proposed method performs a whitening transformation without additional computational cost

Models	# of Params	GFLOPS	Inference Time (ms)
Baseline	45.082M	554.31	10.48
†IBN-Net [42]	45.083M	554.31	10.51
†IterNorm [21]	45.081M	554.31	40.31
Ours	45.081M	554.31	10.43

RobustNet

- Experiments
  - Figure below shows comparison of covariance matrices
    - Visualization of the covariance matrix of intermediate feature maps from IBN-Net and model with ISW
    - The first pair of covariance matrices are from the first convolution layer and the others are from the second convolution layer
    - Style information mainly exists in the early layers of the network as pointed out in IBN-Net
      - The covariance matrices are sparser at the second pair, compared to the first ones
    - The ones from the model with ISW are whitened but a small number of covariance elements remain large, showing ISW selectively eliminates the covariance

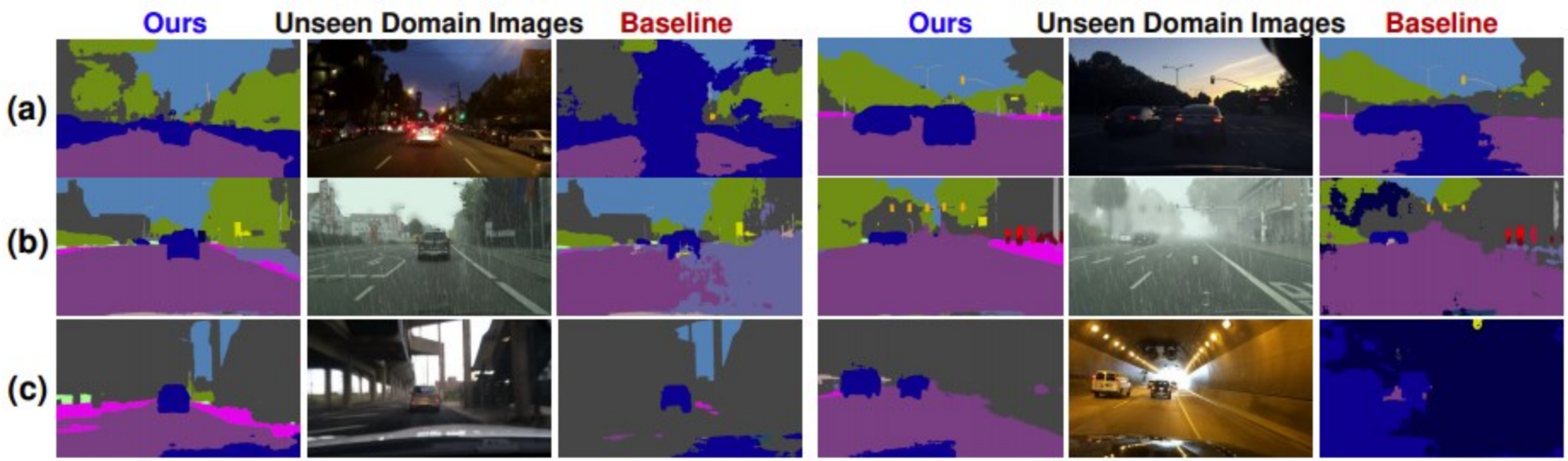


RobustNet

- Experiments
  - Figure below shows reconstructing images with whitened features
    - Reconstructed images from ISW-whitened feature maps using U-Net
      - The first row: a baseline backbone, the second row: an ISW model backbone
    - The image contents are properly maintained while the style such as illumination and colors vanish
    - Tested with the image size of 2048×1024 on NVIDIA A100 GPU
    - Proposed method performs a whitening transformation without additional computational cost

RobustNet

- Results
  - Figure below shows segmentation results on unseen domains
    - i.e., BDD-100K and RainCityscapes
    - (a) low-illuminated, (b) rainy, and (c) unexpected scenes



Thank you for listening