CVPR 2021 Tutorial

Normalization Techniques in Deep Learning: Methods, Analyses and Applications

Lei Huang

Beihang University, Beijing, China

北京航空航天大学人工智能研究院
Institute of Artificial Intelligence, Beihang University

软件开发环境国家重点实验室
State Key Laboratory of Software Development Environment

2021-06-19

Outline

01. Motivations of Normalization Techniques

02. Introduction of Normalization Methods

03. Analyses of Normalization

04. Applications of Normalization

Normalization for Application

- The general idea of learning invariant property and editing the distribution

	Statistics of a set of images (domain)	Statistics of one image (style)
Discriminative model	Distribution alignment (Domain invariant learning)	Contrast removal (Style invariant learning)
Generative model	Edit domain information 	Edit style information 

Outline

04. Applications of Normalization

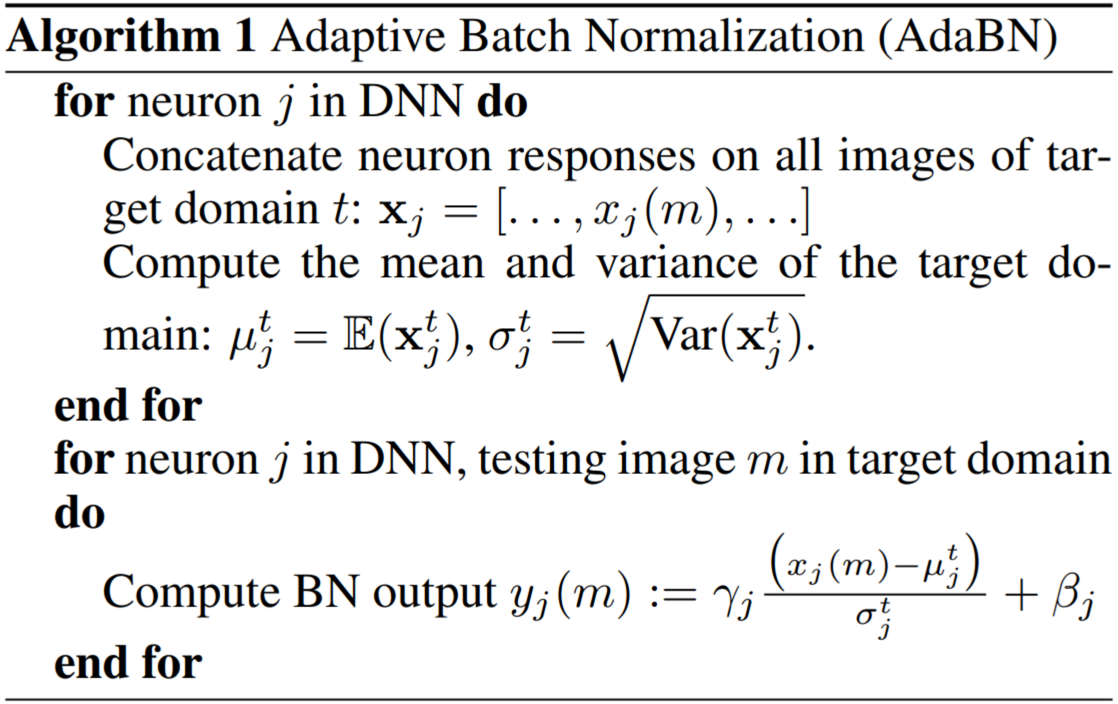
- Control domain
- Control style
- Training GANs
- Efficient model

Normalization for Domain Adaptation

- Adaptive Batch Normalization

(a) Shallow layer distributions (b) Deep layer distributions

t-SNE visualization of the mini-batch BN feature vector distributions in both shallow and deep layers, across different datasets. Each point represents the BN statistics in one mini-batch.



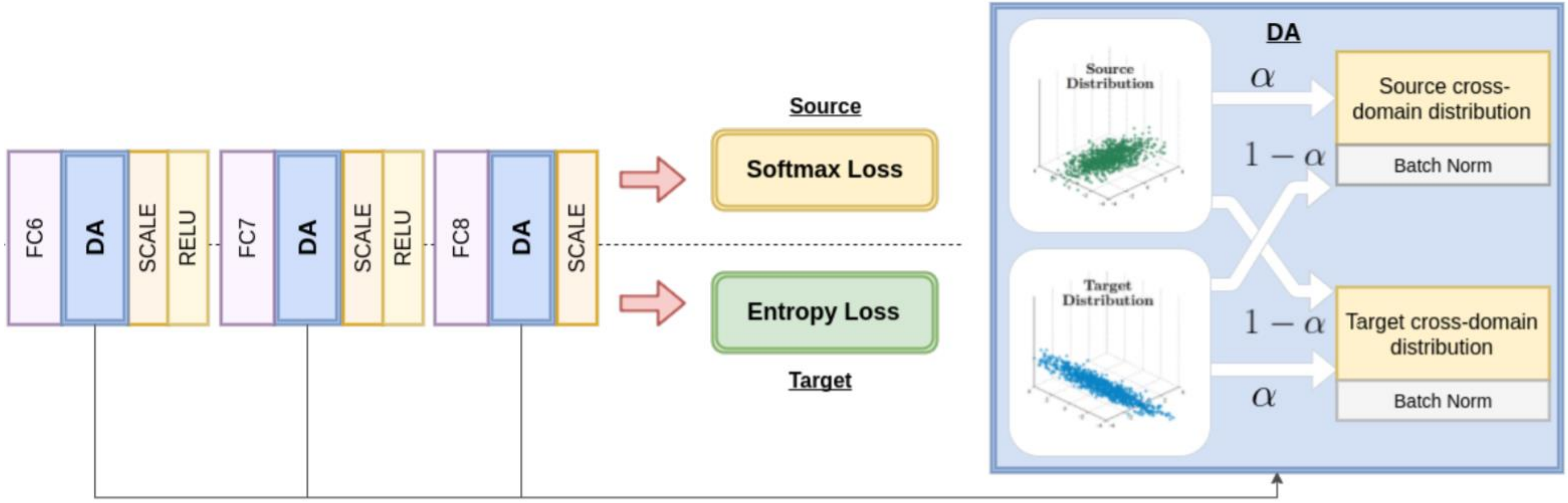
Algorithm 1 Adaptive Batch Normalization (AdaBN)

```
for neuron $j$ in DNN do
    Concatenate neuron responses on all images of target domain $t$: $\mathbf{x}_j = [\ldots, x_j(m), \ldots]$
    Compute the mean and variance of the target domain: $\mu_j^t = \mathbb{E}(\mathbf{x}_j^t)$, $\sigma_j^t = \sqrt{\mathrm{Var}(\mathbf{x}_j^t)}$.
end for
for neuron $j$ in DNN, testing image $m$ in target domain do
    Compute BN output $y_j(m) := \gamma_j \frac{(x_j(m) - \mu_j^t)}{\sigma_j^t} + \beta_j$
end for
```

Revisiting Batch Normalization For Practical Domain Adaptation [Li et al, 2016]

Normalization for Domain Adaptation

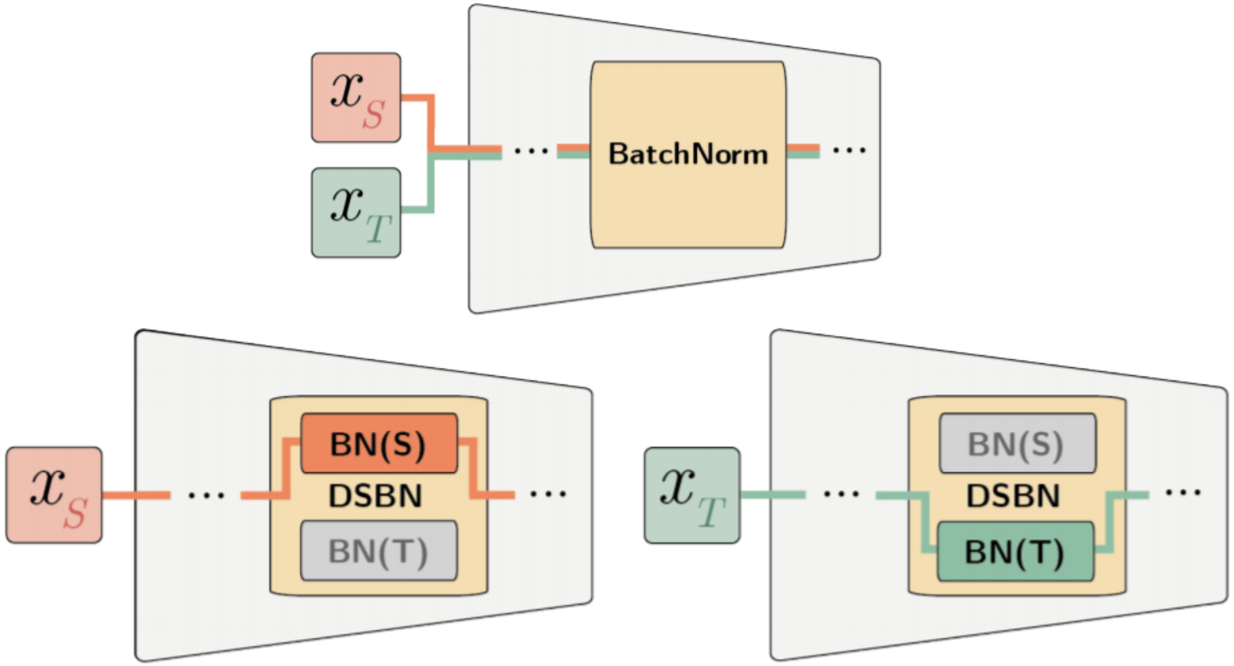
- Domain alignment
  - – Align during training,
  - – Combining semi-supervised loss (uncertainty on target)

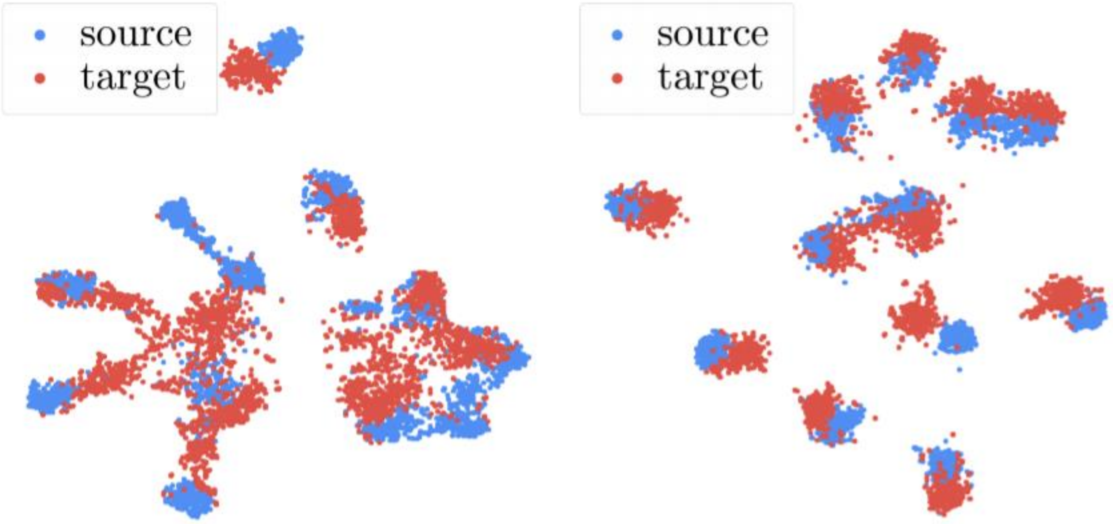


AutoDIAL: Automatic DomaIn Alignment Layers [Carlucci et al, 2017]

Normalization for Domain Adaptation

- Domain Specific BN
  - Align during training, specific to each domain
  - Pseudo label

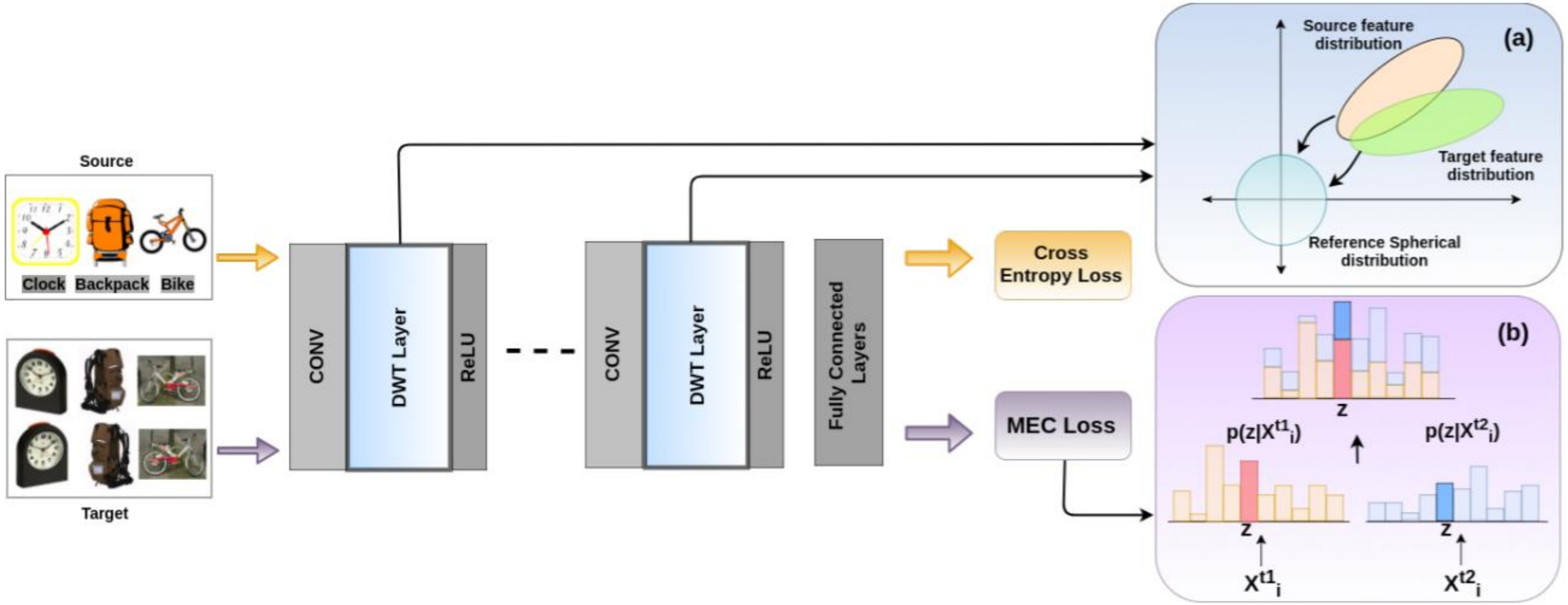




Domain-Specific Batch Normalization for Unsupervised Domain Adaptation [Chang et al, 2019]

Normalization for Domain Adaptation

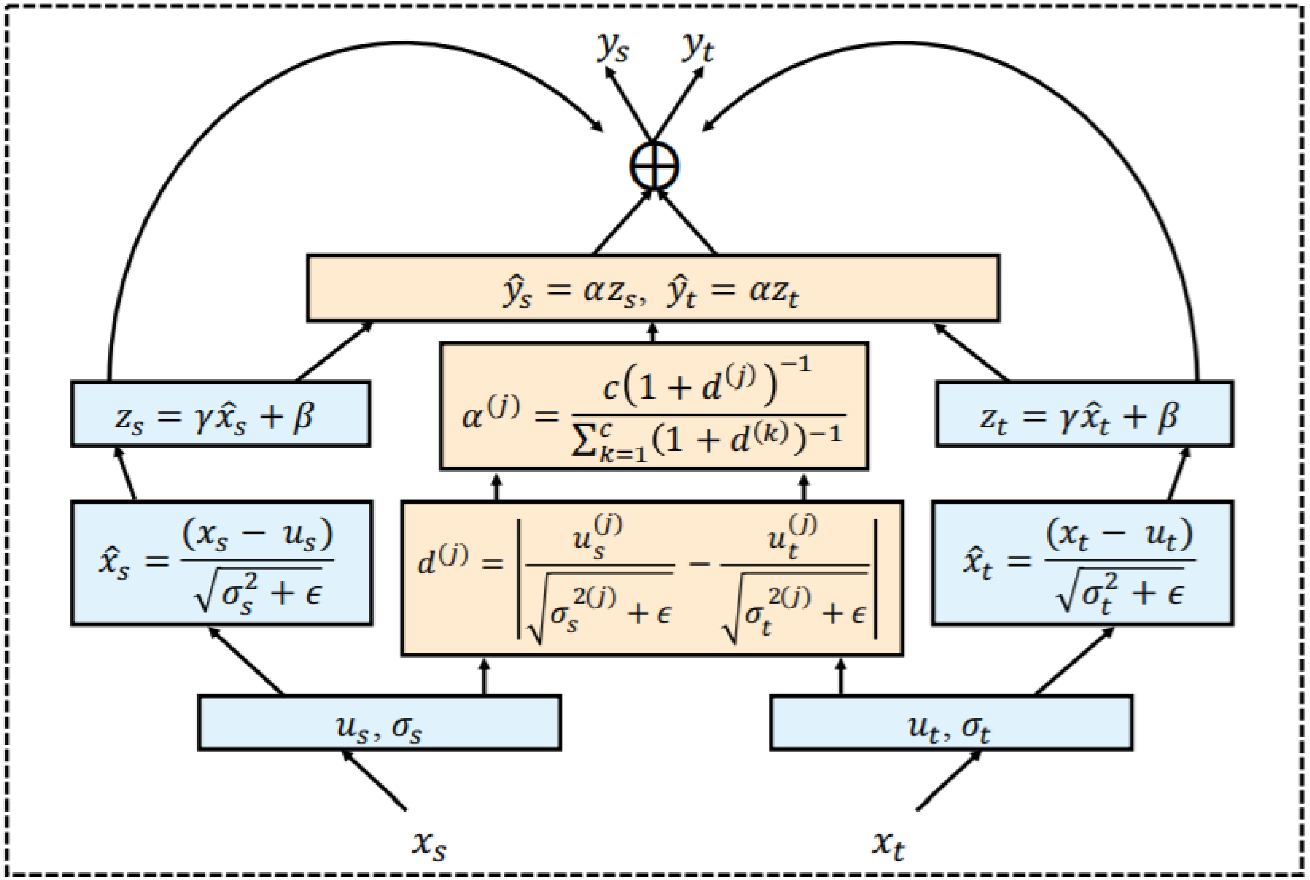
- Whitening operation for alignment
  - – Align during training, specific to each domain
  - – Using whitening for align



Unsupervised Domain Adaptation Using Feature-Whitening and Consensus Loss [Roy et al, CVPR 2019]

Normalization for Domain Adaptation

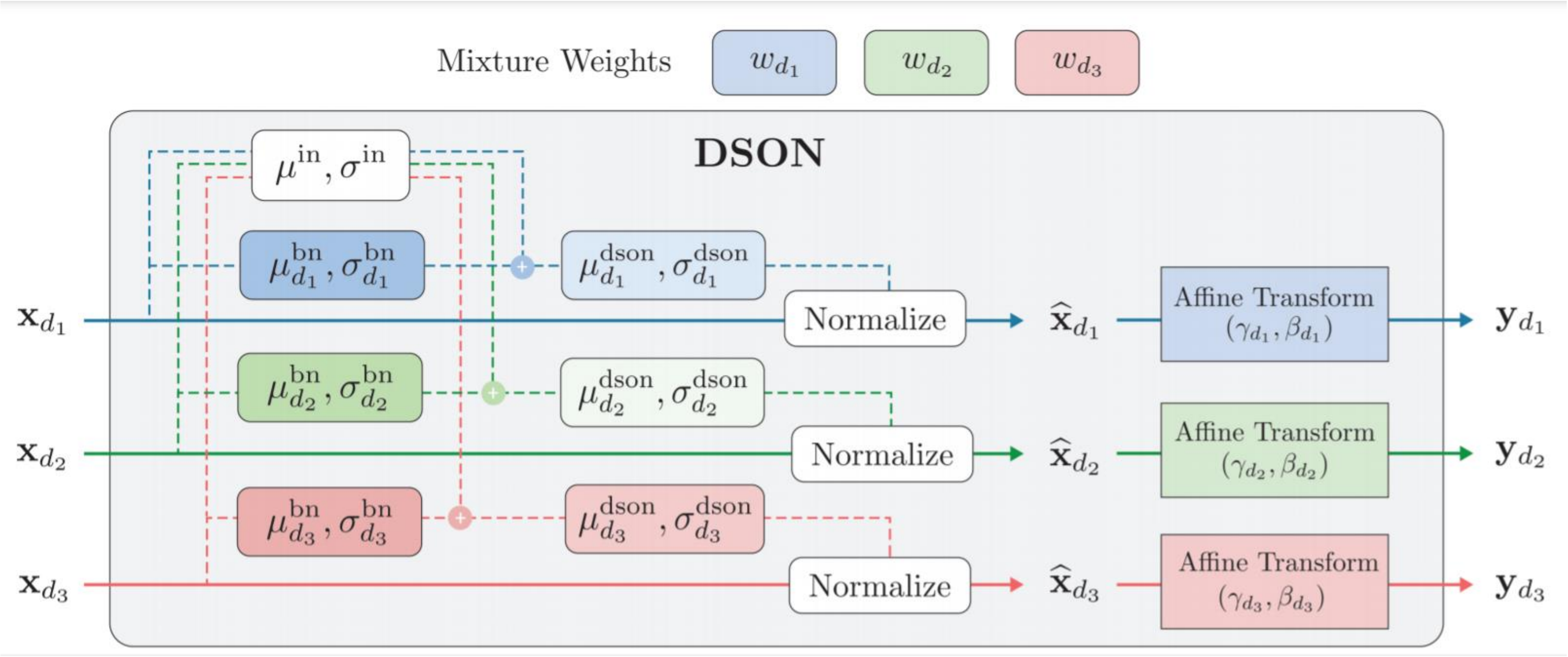
- Transferable normalization
  - Align during training, specific each domain
  - Get the information for transfer from other target statistics



Transferable Normalization: Towards Improving Transferability of Deep Neural Networks [Wang et al, 2019]

Normalization for Domain Generalization

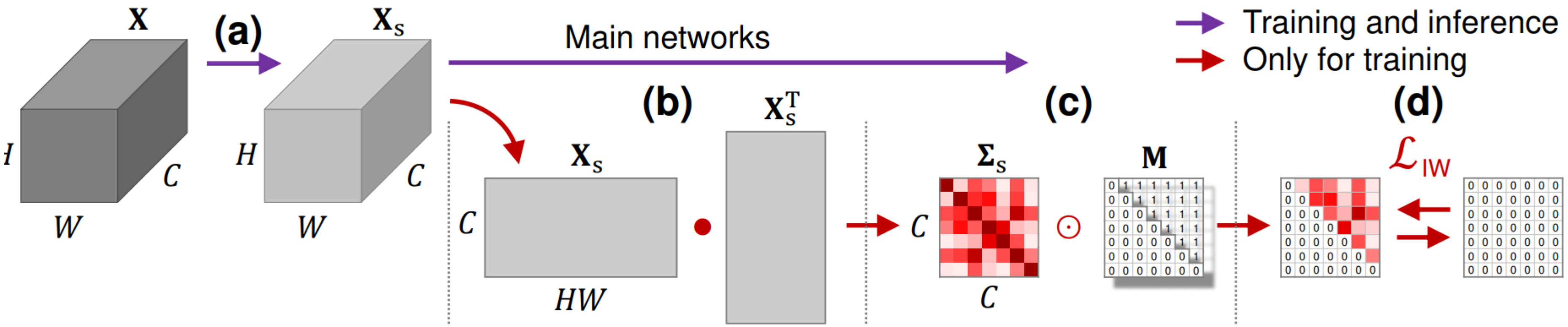
- Domain-specific optimized normalization (DSON)
  - – Normalized by a weighted average of multiple normalization statistics

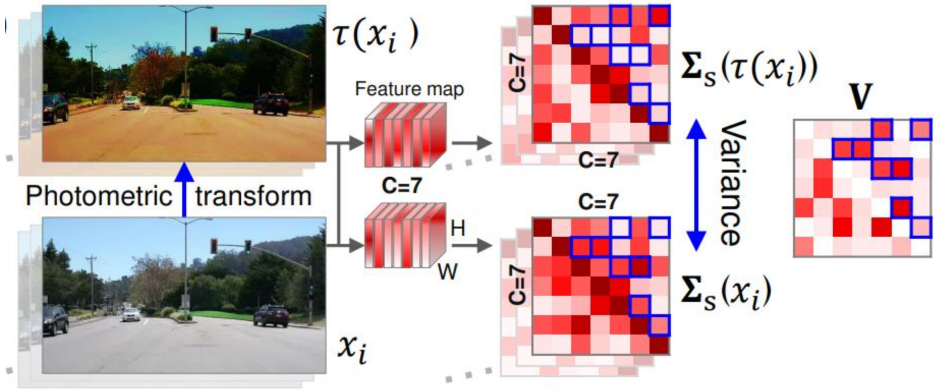


Learning to Optimize Domain Specific Normalization for Domain Generalization [Seo et al, ECCV 2020]

Normalization for Domain Generalization

- For domain generalization
  - Instance selective whitening

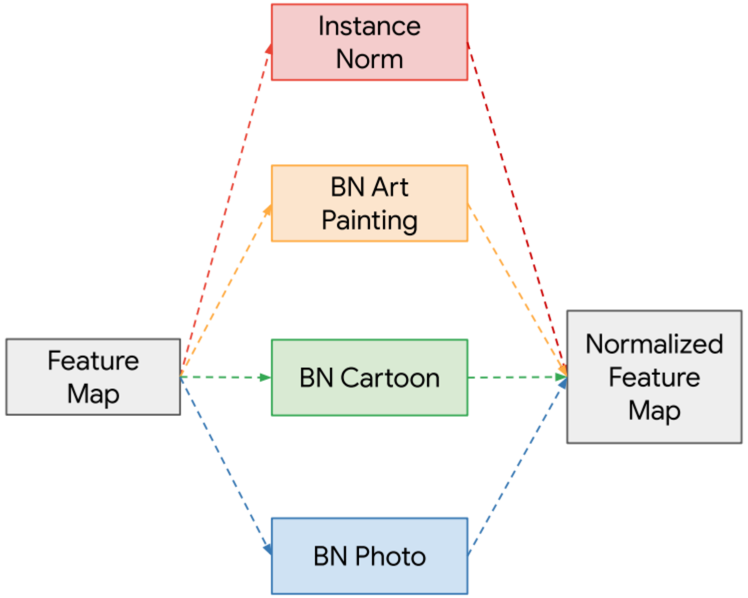




RobustNet: Improving Domain Generalization in Urban-Scene Segmentation via Instance Selective Whitening [Tsai et al, CVPR 2021]

Normalization for Domain Generalization

- Batch normalization embeddings



Instance Norm

BN Art Painting

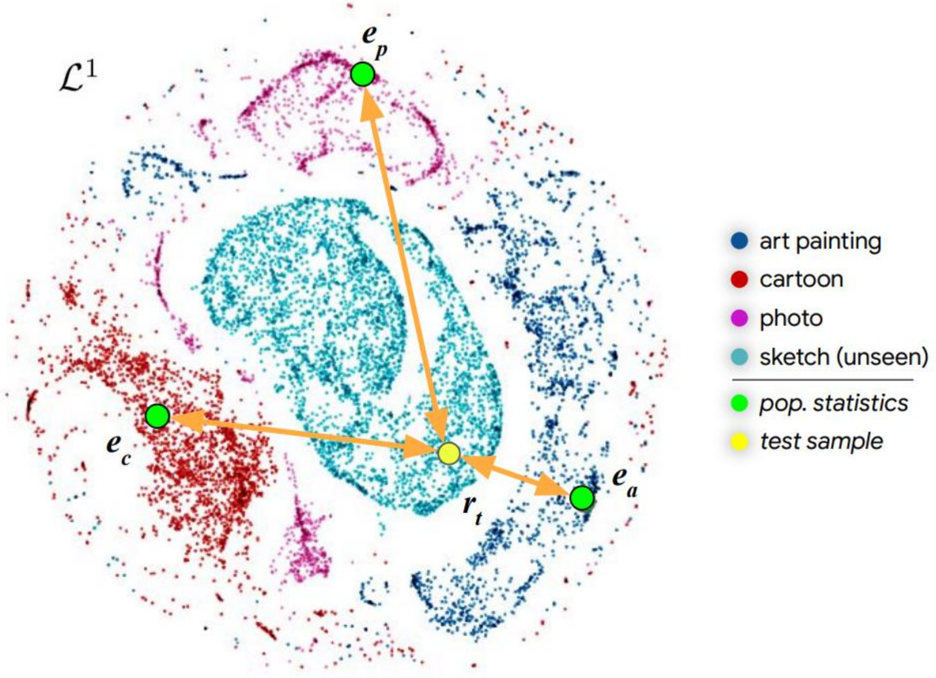
Feature Map

BN Cartoon

Normalized Feature Map

BN Photo

Multi-Source Domain Alignment Layer



Batch Normalization Embeddings for Deep Domain Generalization [Segu et al, 2020]

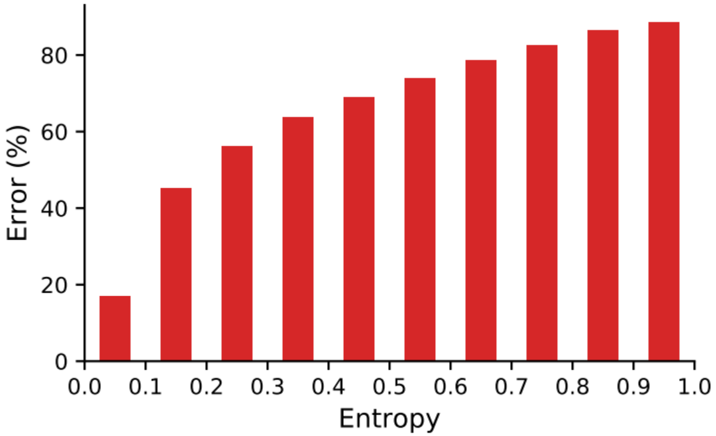
Normalization for Corruption Robust

- Estimate BN statistics during test to improve performance for corruption
  - Improving robustness against common corruptions by covariate shift adaptation [Schneider et al, 2020]
  - Revisiting Batch Normalization for Improving Corruption Robustness [Benz et al, 2020]

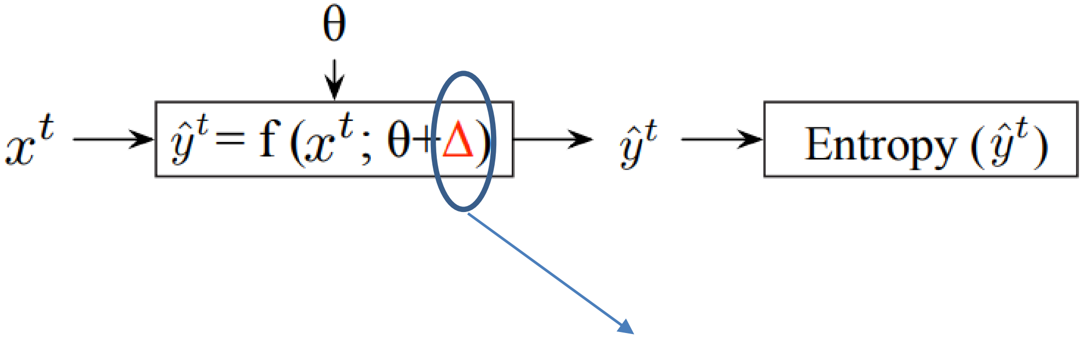
- BN’s train mode for test (mini-batch test)
  - Evaluating prediction-time batch normalization for robustness under covariate shift [Nado et al, 2020]

Normalization for Corruption Robust

- Tent: fully test-time adaptation
  - Source-free adaptation,
  - Entropy minimization on test data



Predictions with lower entropy have lower error rates on corrupted CIFAR-100-C. Certainty can serve as supervision during testing.



$$x^t \xrightarrow{} \boxed{\hat{y}^t = f(x^t; \theta + \Delta)} \xrightarrow{} \hat{y}^t \xrightarrow{} \boxed{\text{Entropy}(\hat{y}^t)}$$

θ

Optimize γ, β of BN

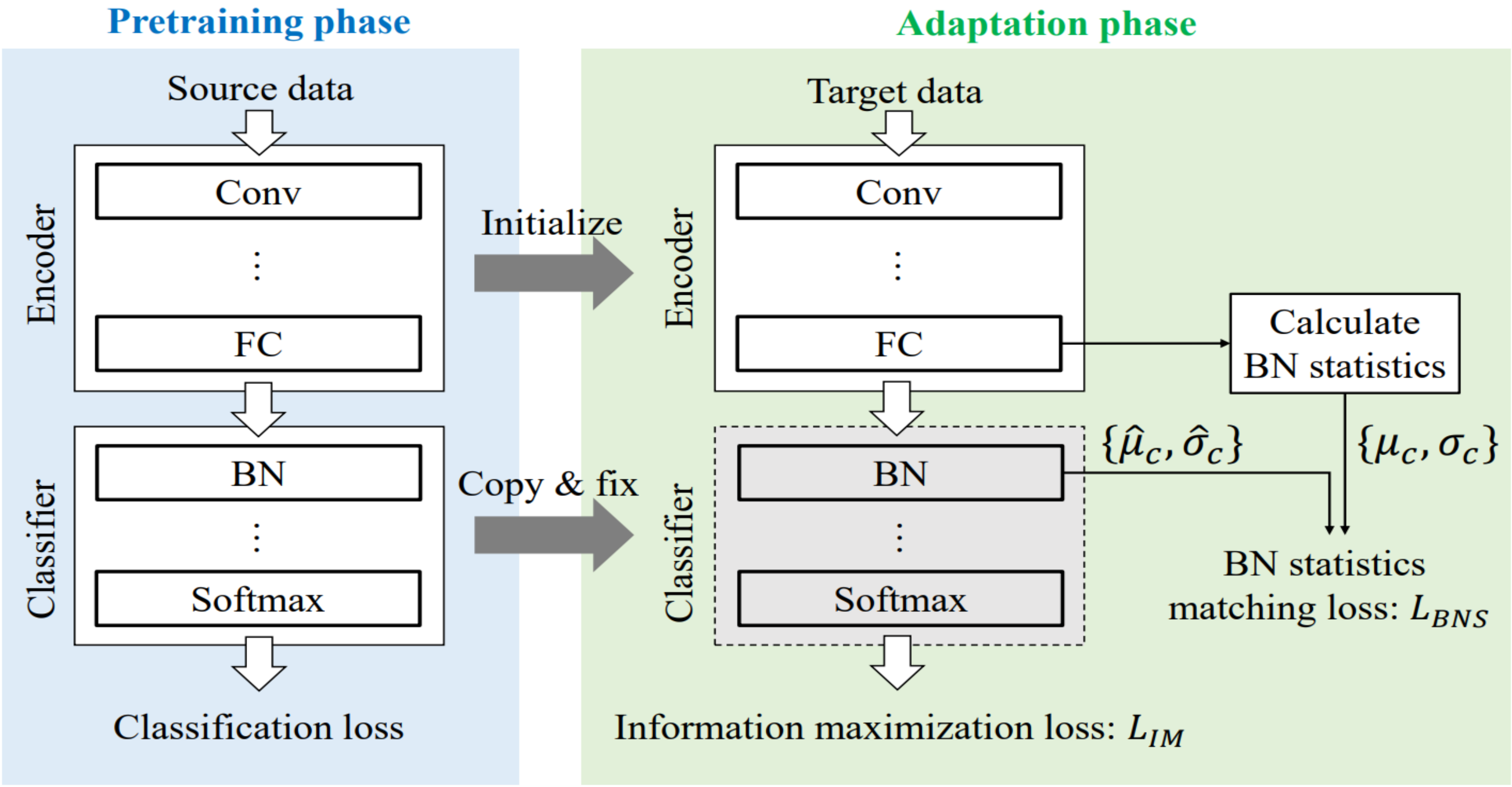
+

Estimate μ, σ of BN

Tent: Fully Test-Time Adaptation by Entropy Minimization [Wang et al, ICLR 2021]

Normalization for Corruption Robust

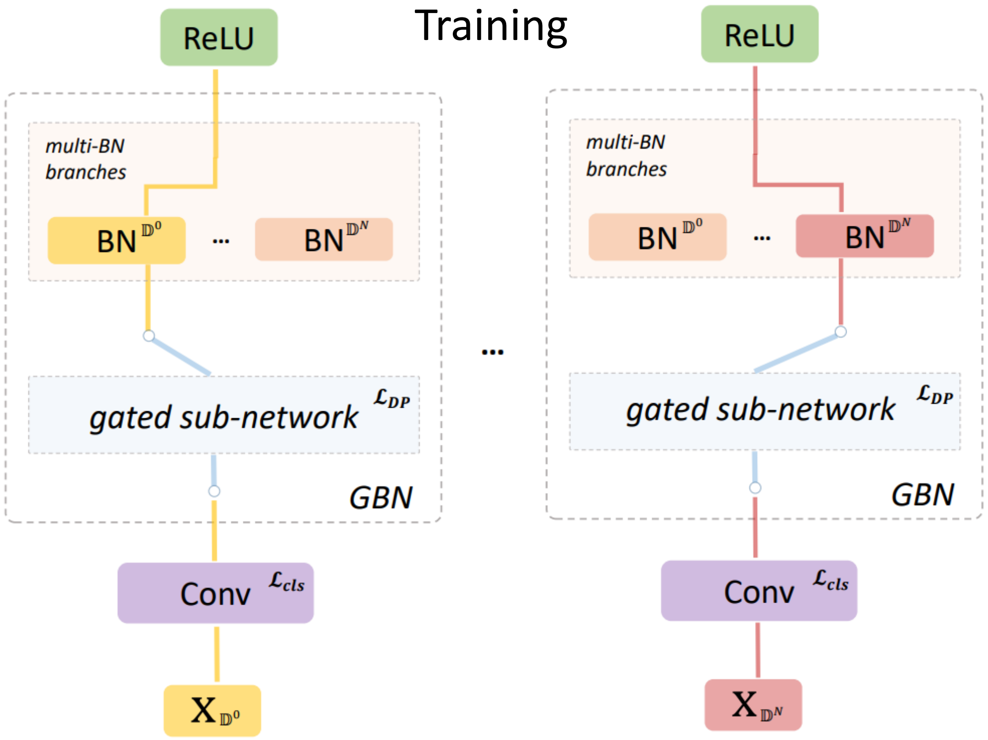
- Matching batch normalization statistics

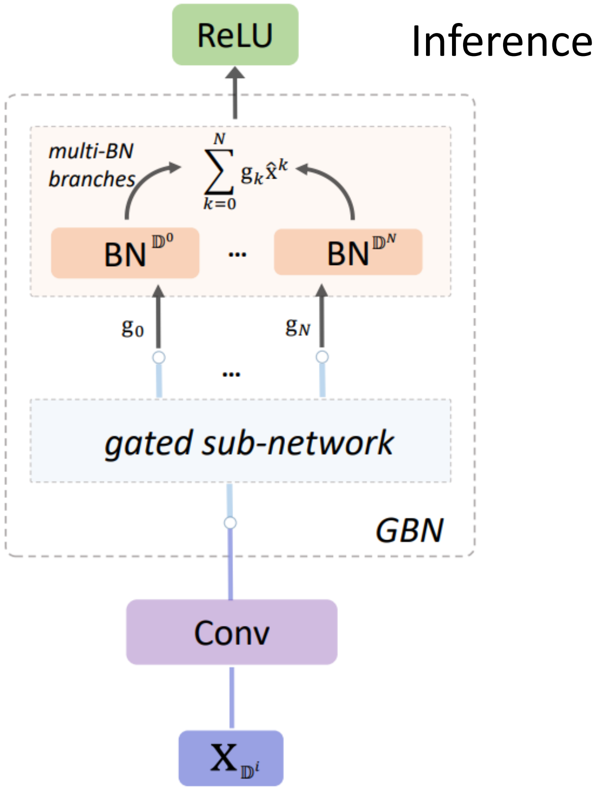


Source-free Domain Adaptation via Distributional Alignment by Matching Batch Normalization Statistics [Ishii et al, 2021]

Normalization for Adversarial Robust

- Gated Batch Normalization (GBN): defending multiple adversarial perturbations



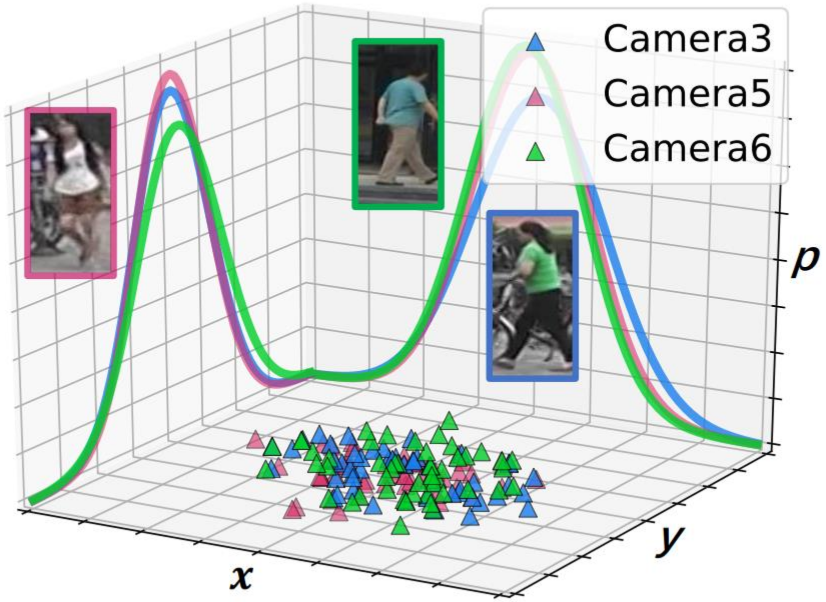


Towards Defending Multiple Adversarial Perturbations via Gated Batch Normalization [Liu et al, 2020]

Normalization for Person Re-identification

- Camera-based Batch Normalization
  - View each camera as a “domain

- Modality Batch Normalization
  - View each modality as a “domain”

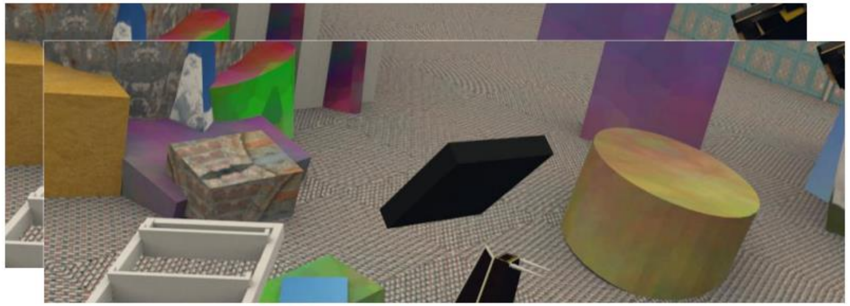


Rethinking the Distribution Gap of Person Re-identification with Camera-based Batch Normalization [Zhuang et al 2020]

Bridging the Distribution Gap of Visible-Infrared Person Re-identification with Modality Batch Normalization [Li et al, 2021]

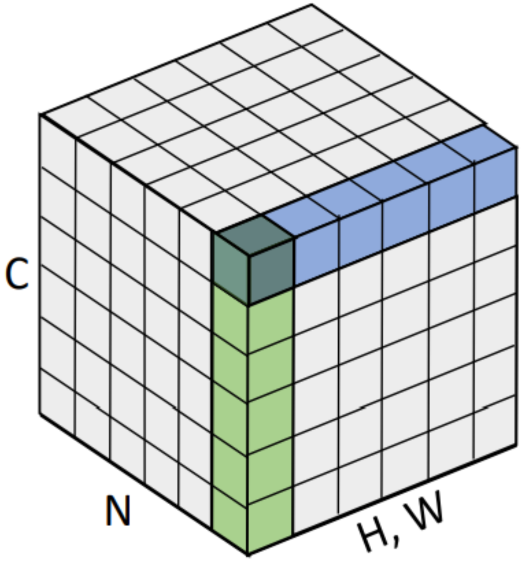
Normalization for Stereo Matching

- Domain-invariant stereo matching networks [Zhang et al, CVPR 2020]
  - Domain generalization: for learning domain invariant stereo matching
  - Domain-invariant normalization: IN + Position L2-Norm



(a) Training Scenes

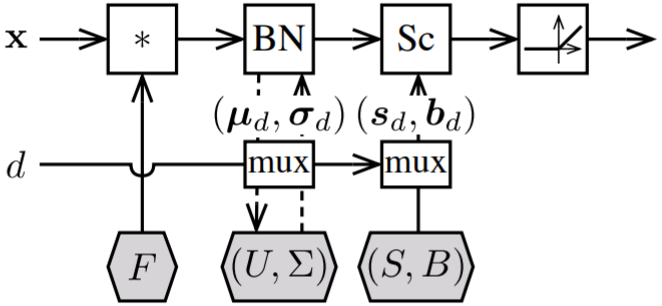
(b) Test Scenes



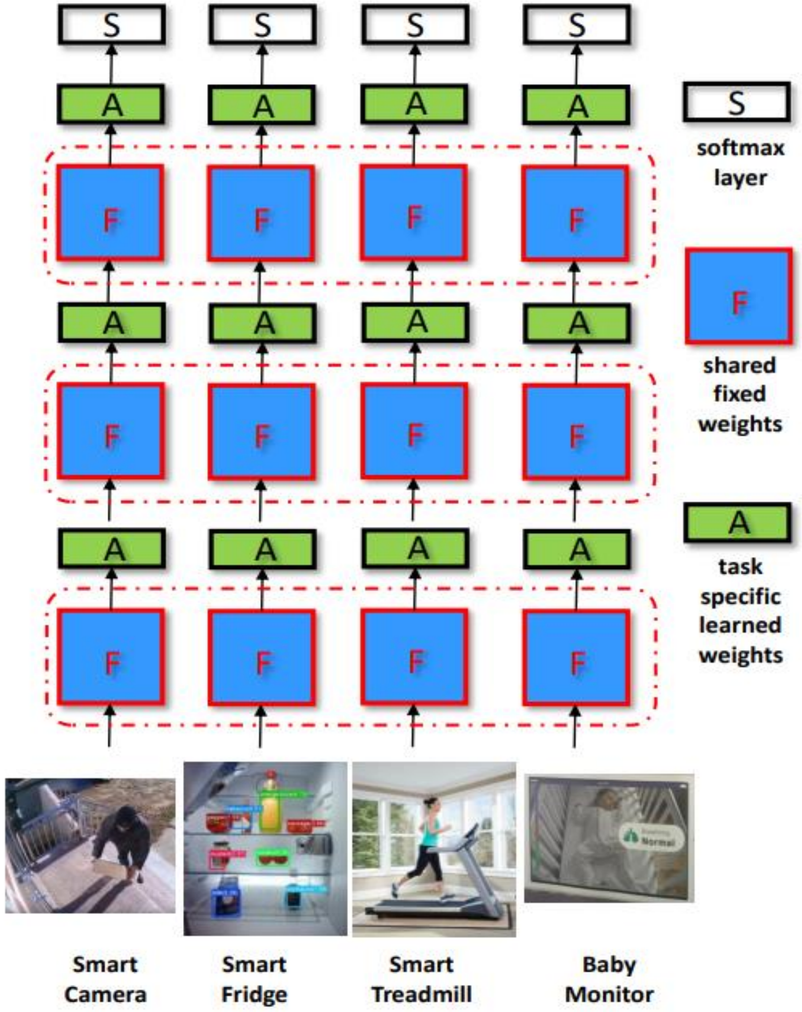
Domain Norm

Normalization for Multi-task Learning

- Share parameters + specific learned weights



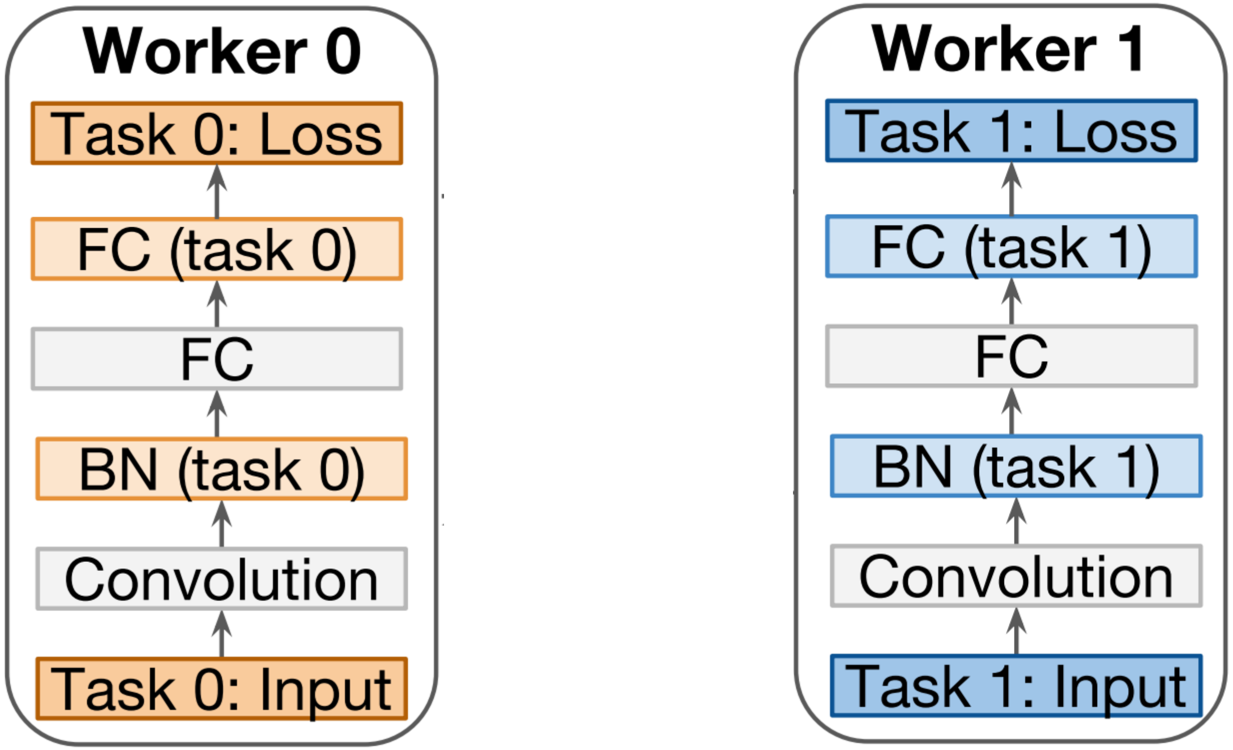
Universal representations: The missing link between faces, text, planktons, and cat breeds [Bilen and Vedaldi, 2017]



Efficient Multi-Domain Learning by Covariance Normalization [Li and Vasconcelos, 2019]

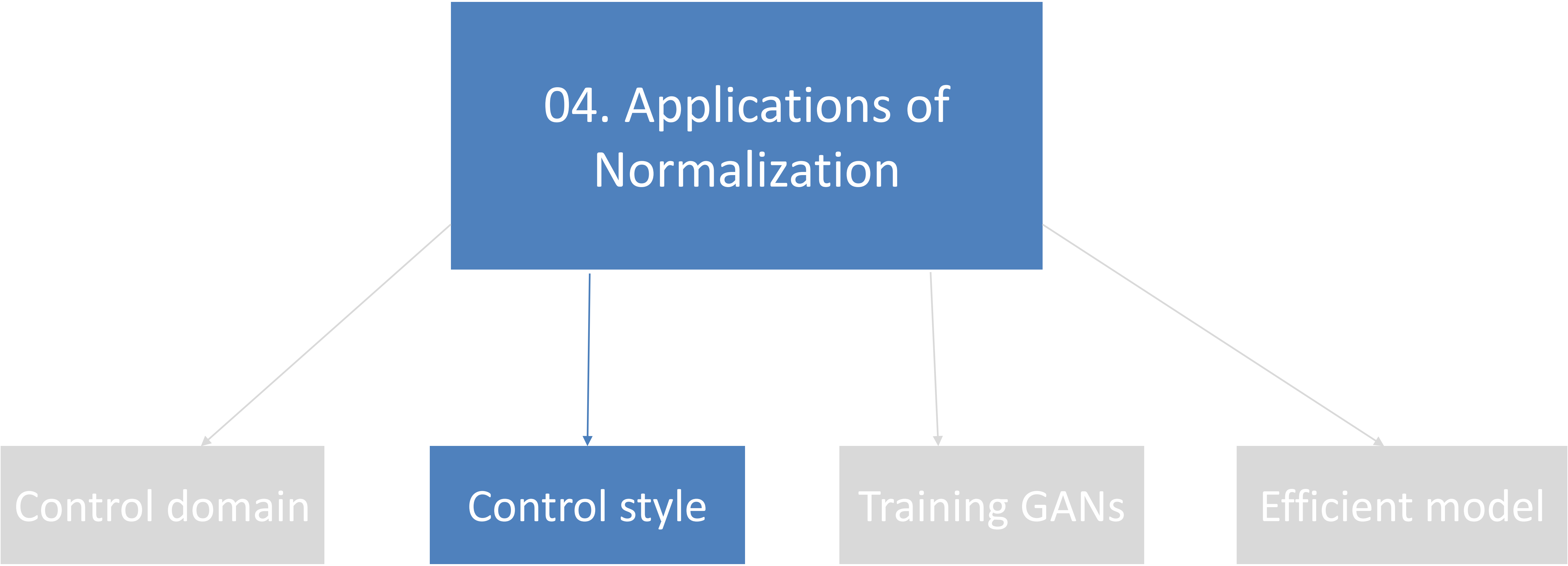
Normalization for Multi task Learning

- BN’s mean/variance + scalar/bias as per task



K for the price of 1: Parameter efficient multi-task and transfer learning [Mudrakarta et al, 2018]

Outline



04. Applications of Normalization

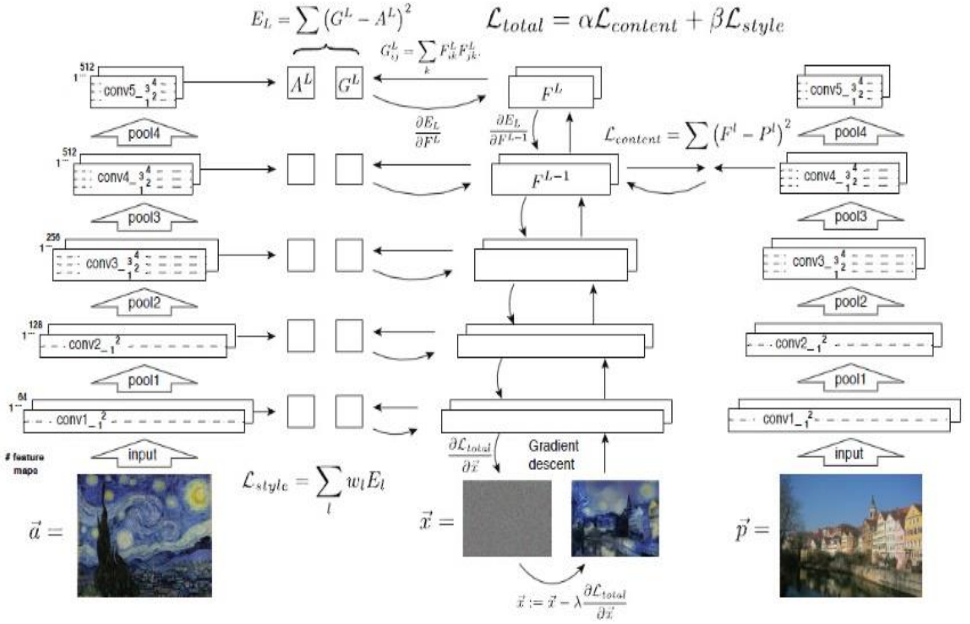
- Control domain
- Control style
- Training GANs
- Efficient model

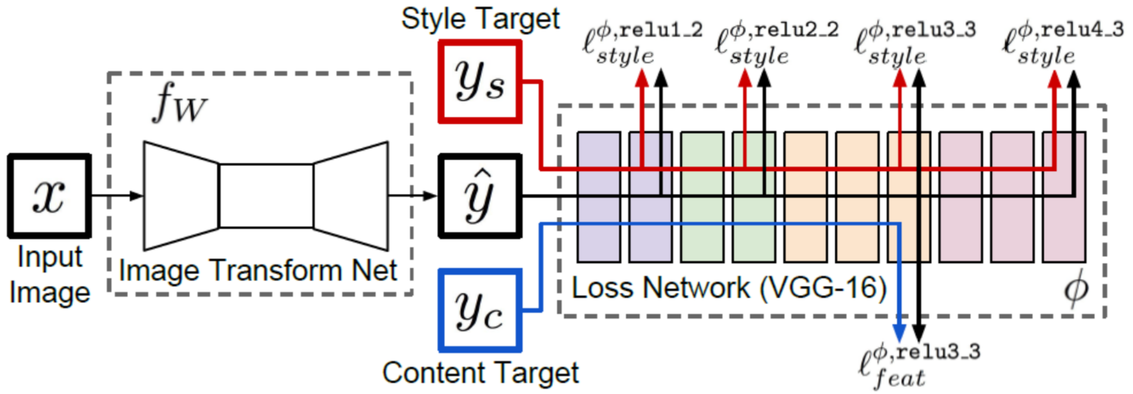
Neural Style Transfer

- Content representation
  - Higher layers activations in neural networks

- Style representation
  - Lower layers activation in neural networks
  - Gram matrix

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}.$$

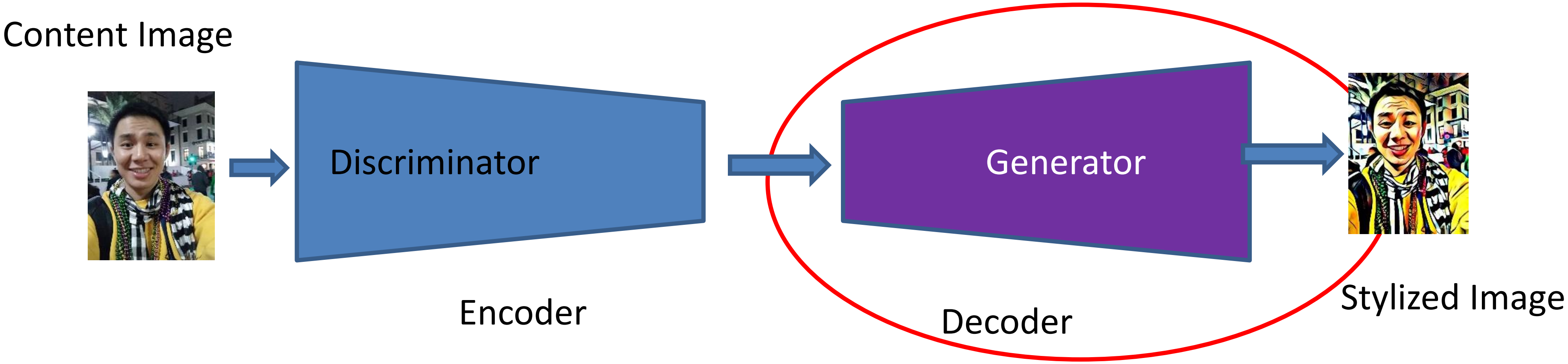




[Gatys et al, CVPR 2016; Johnson et al, ECCV 2016]

Neural Style Transfer

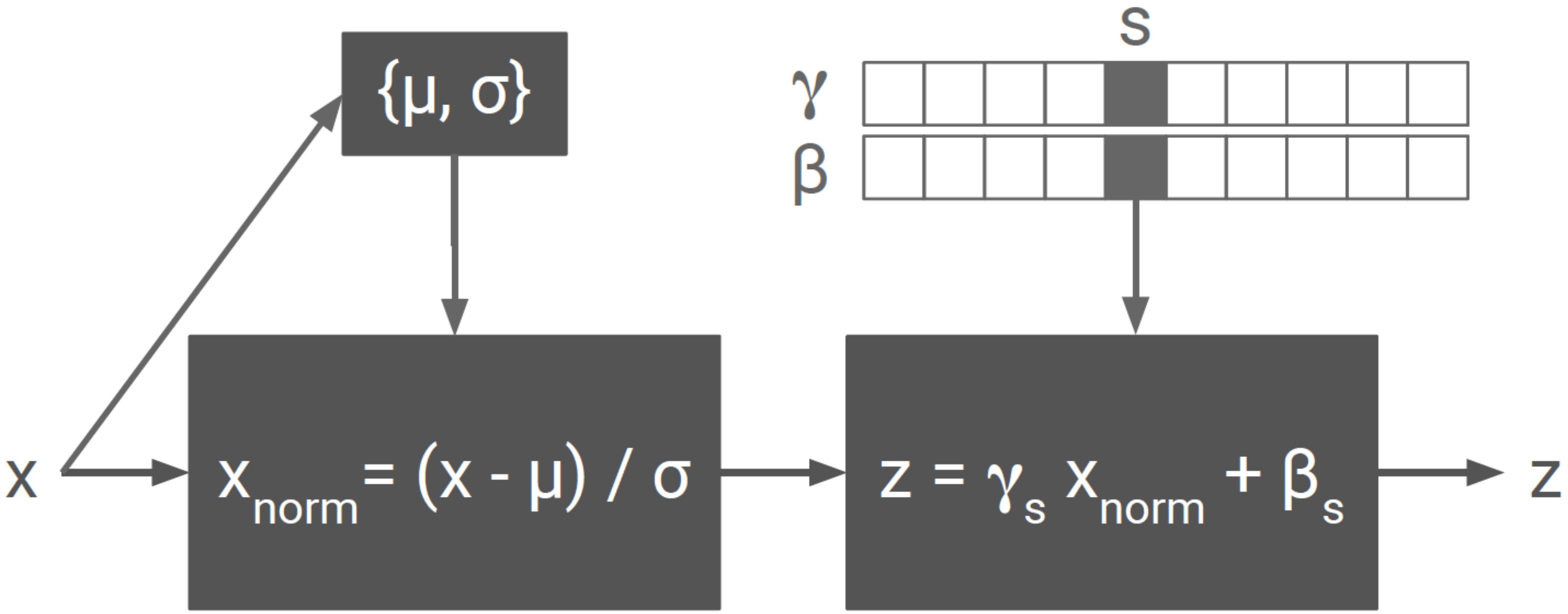
- Instance Normalization
  - – No data dependency
  - – Remove instance-specific contrast information from the content image



Instance Normalization: The Missing Ingredient for Fast Stylization [Ulyanov et al, 2016]

Neural Style Transfer

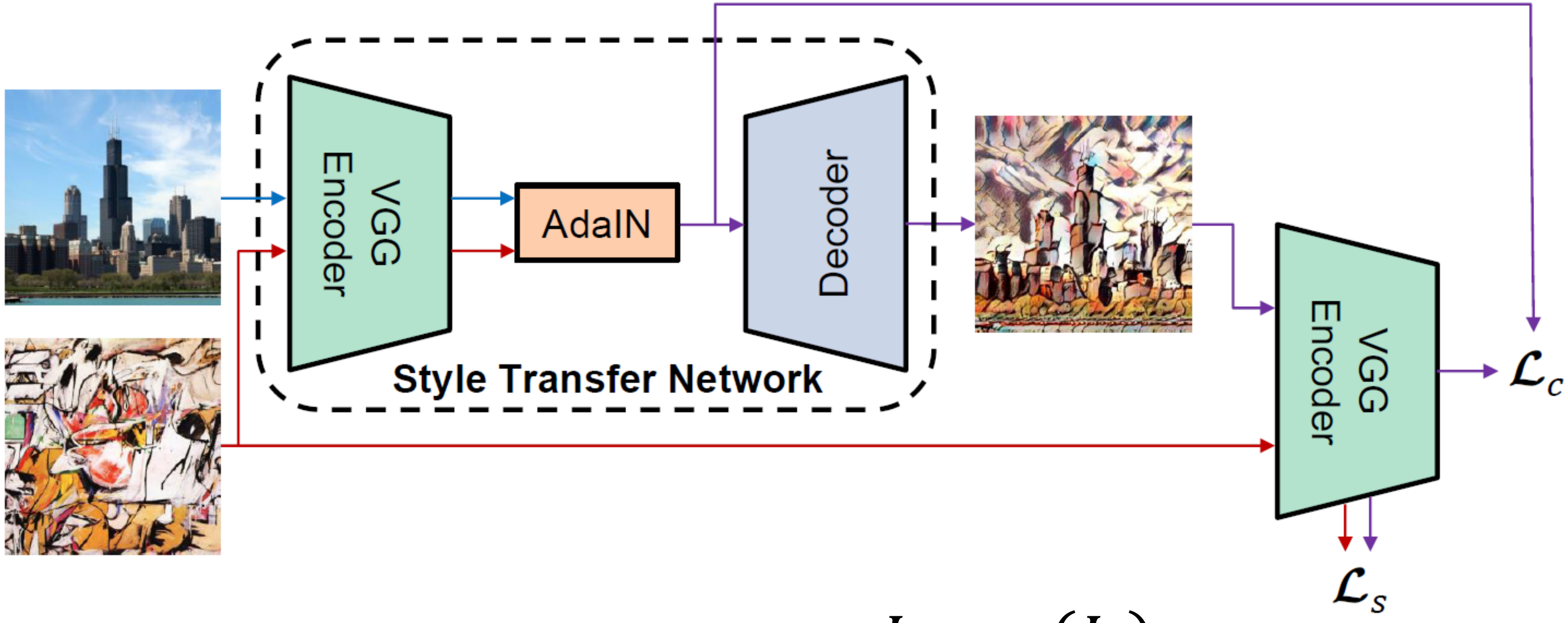
- Condition Instance Normalization (CIN)



A Learned Representation For Artistic Style [Dumoulin et al, ICLR 2017)]

Neural Style Transfer

- Adaptive instance normalization (Huang et al, ICCV 2017)



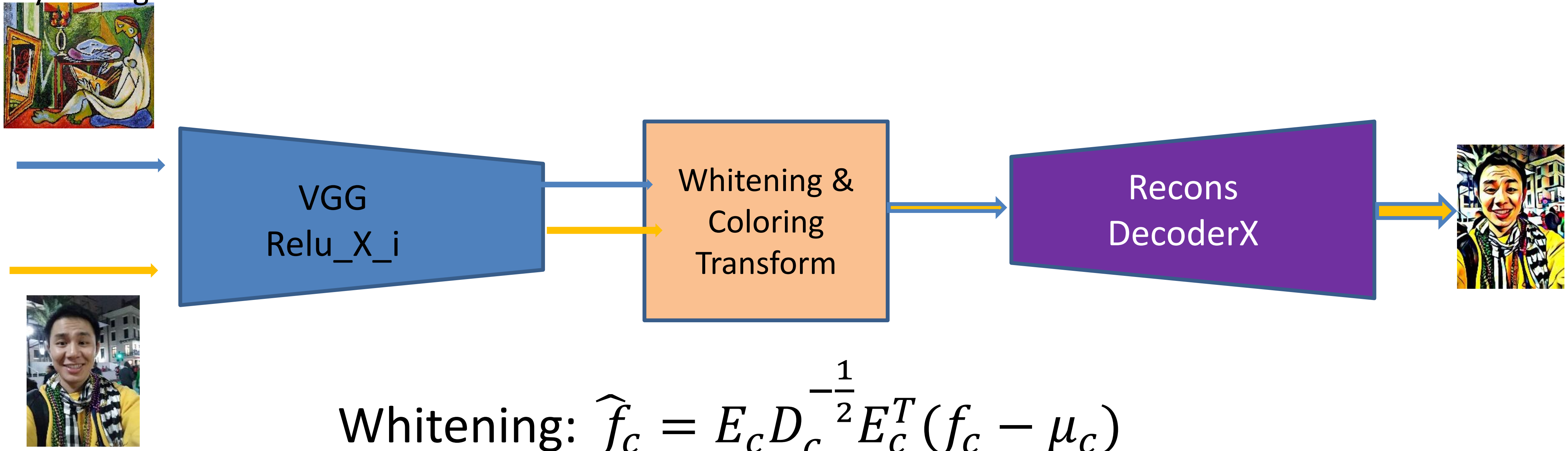
$$AdaIN(I_c, I_s) = \sigma(I_s) * \frac{I_c - \mu(I_c)}{\sigma(I_c)} + \mu(I_s)$$

Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization [Huang et al, ICCV 2017]

Neural Style Transfer

- Whitening Instance Norm (WIN)

Style Image



Content Image

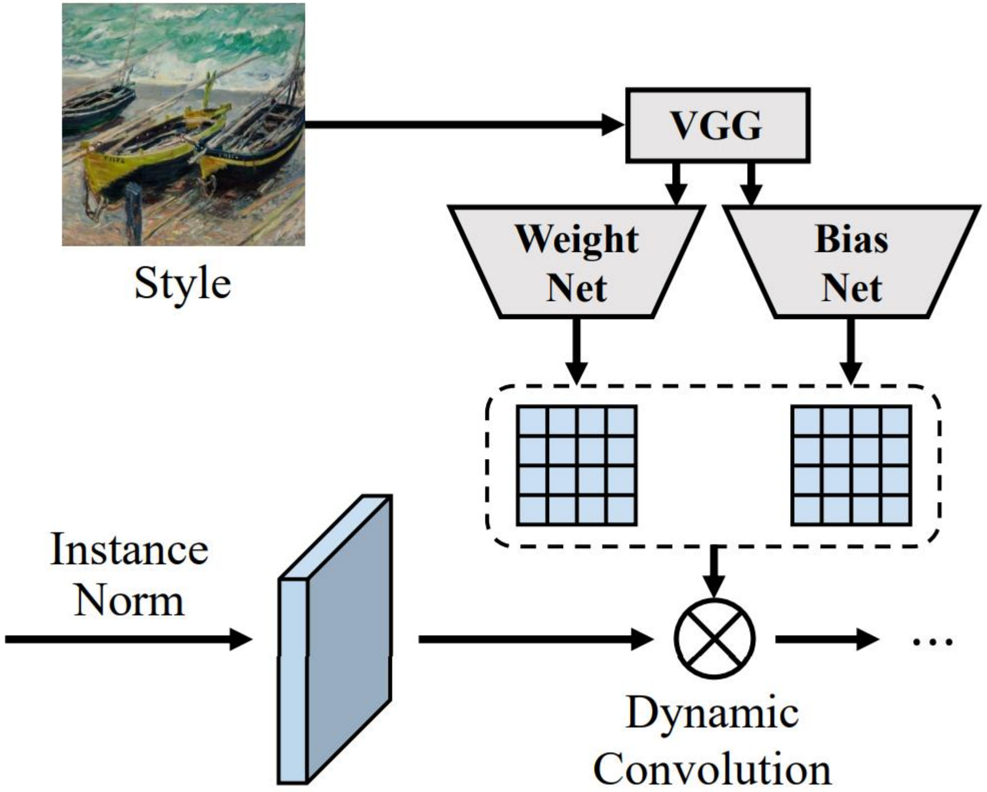
Whitening: $\widehat{f}_c = E_c D_c^{-\frac{1}{2}} E_c^T (f_c - \mu_c)$

Coloring: $\widehat{f}_{cs} = E_s D_s^{\frac{1}{2}} E_s^T \widehat{f}_c + \mu_s$

Universal style transfer via feature transforms [Li et al, NeurIPS 2017]

Neural Style Transfer

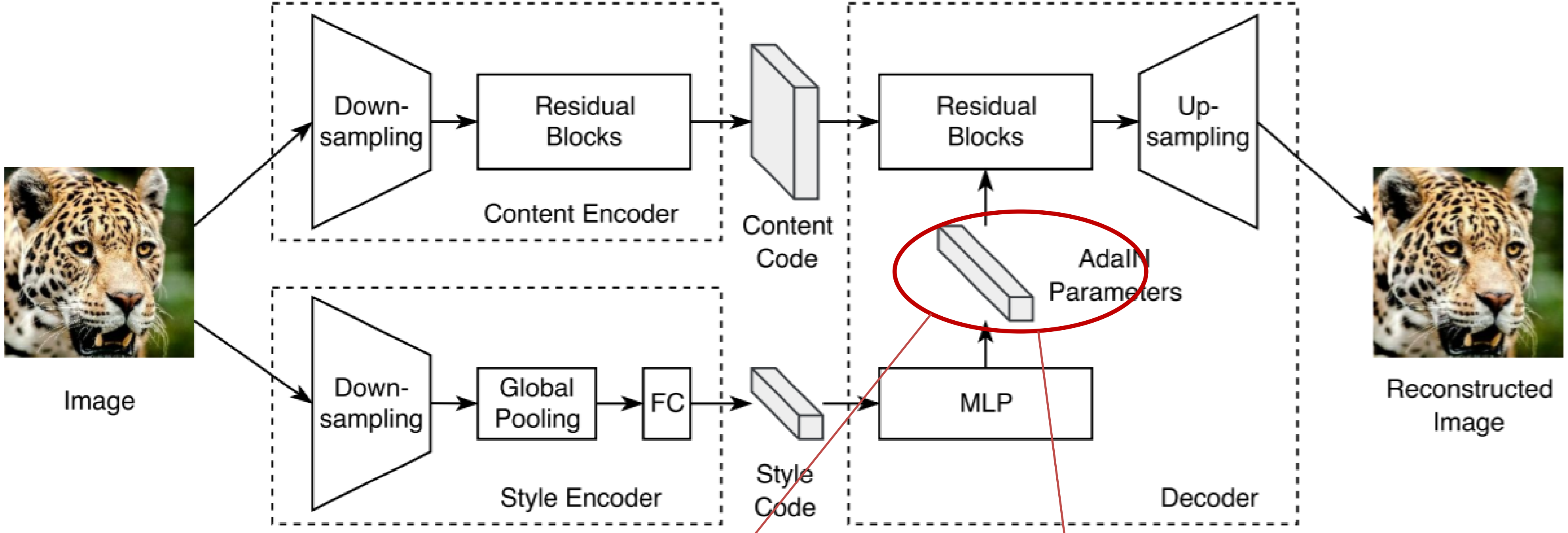
- Dynamic Instance Normalization (DIN)



Dynamic Instance Normalization for Arbitrary Style Transfer [Jing et al, AAAI 2020]

Image Translation

- Adaptive instance normalization



$$\mathrm{AdaIN}(z,\gamma,\beta)=\gamma\left(\frac{z-\mu(z)}{\sigma(z)}\right)+\beta$$

Multimodal Unsupervised Image-to-Image Translation [Huang et al, ECCV 2018]

Image Translation

- group-wise whitening-and-coloring transformation

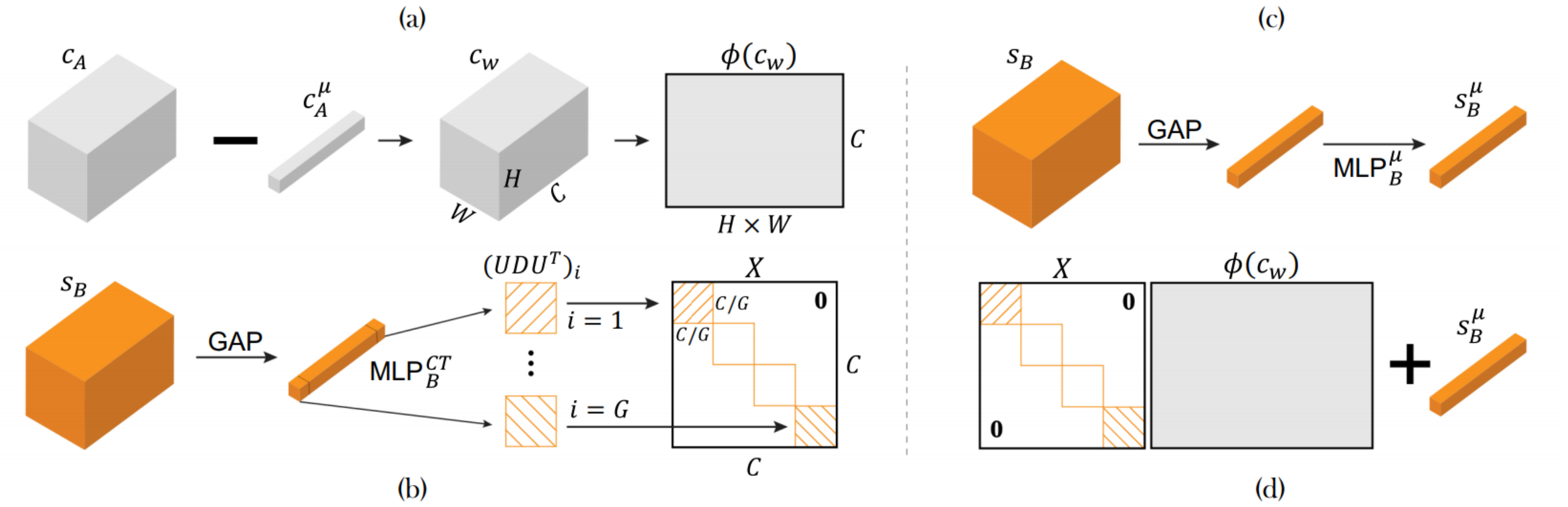
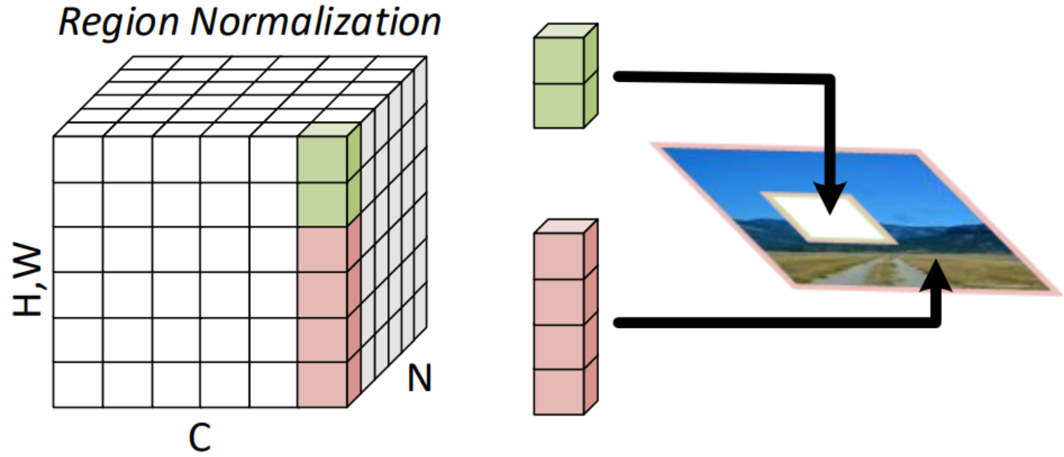
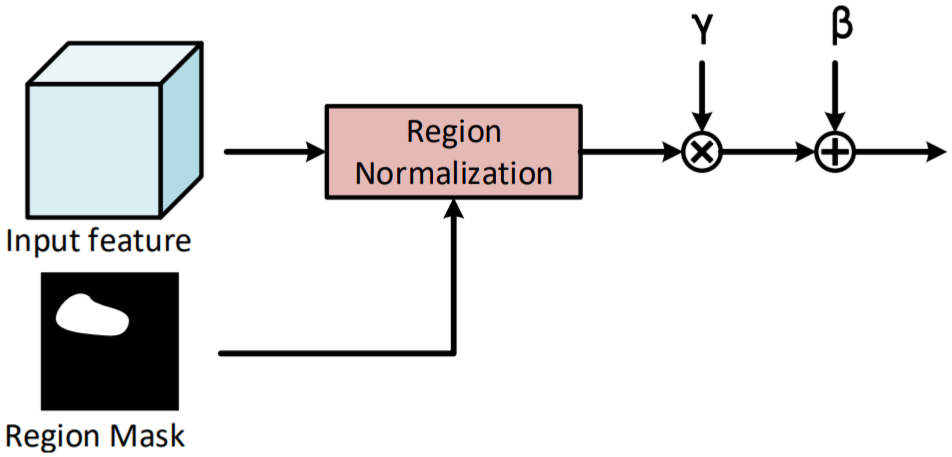


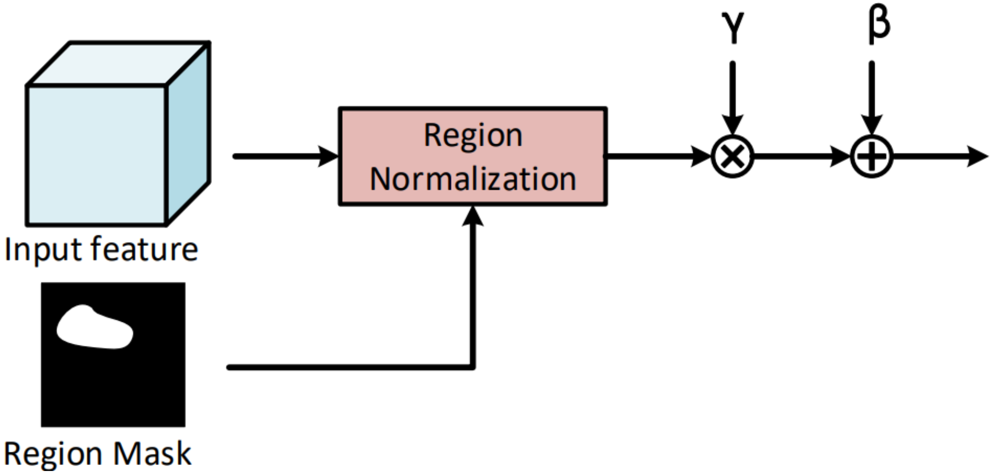
Image-to-image translation via group-wise deep whitening-and-coloring transformation[Cho et al, CVPR 2019]

Image Inpainting

- Region Normalization







Region Normalization for Image Inpainting [Yu et al, AAAI 2020]

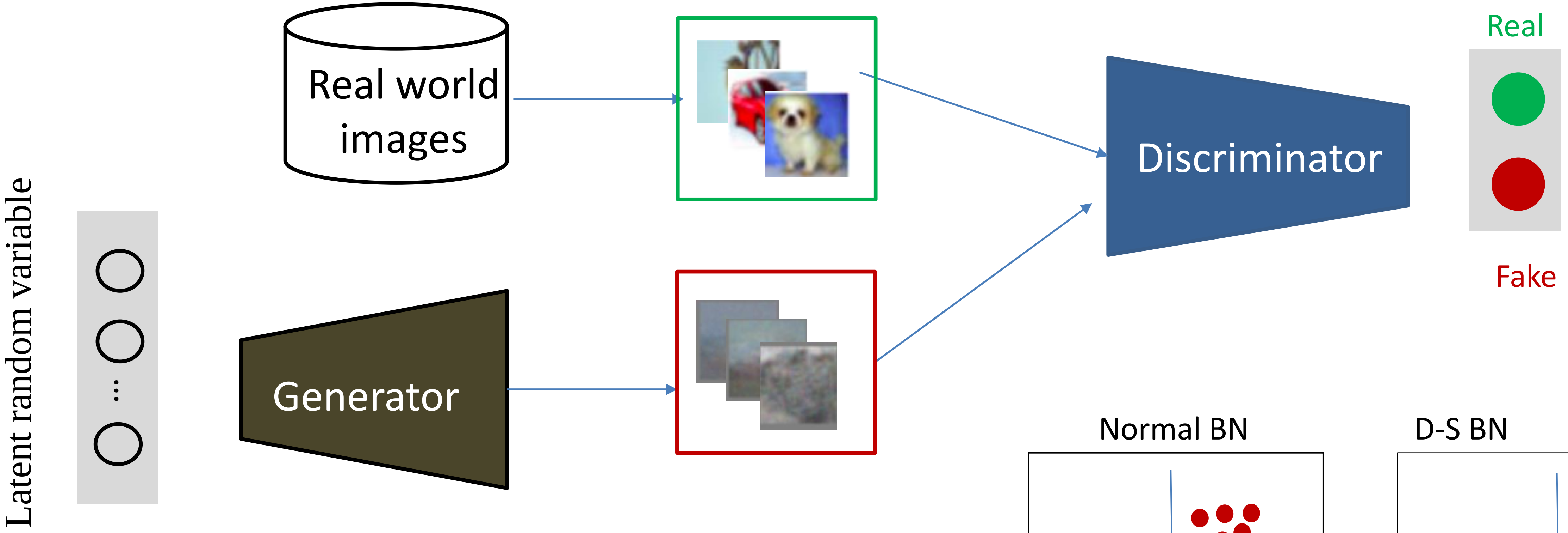
Outline

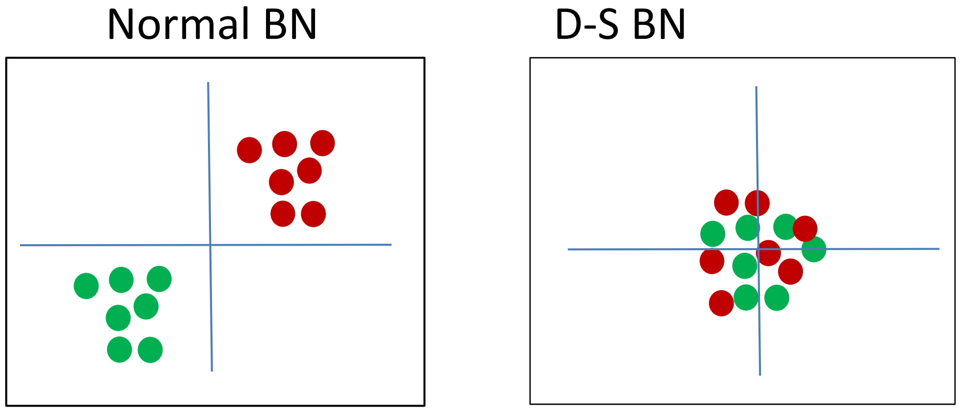
04. Applications of Normalization

- Control domain
- Control style
- Training GANs
- Efficient model

Training GANs

- Control the pace of learning of discriminator

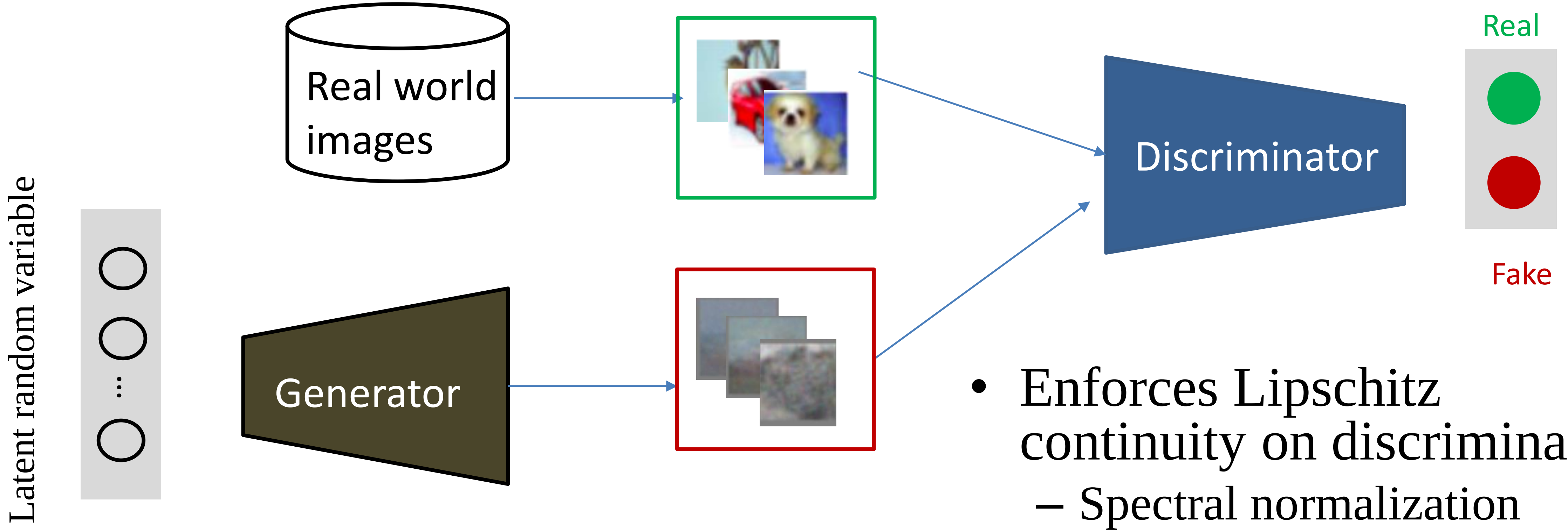




- Domain-specific BN

Training GANs

- Control the pace of learning of discriminator



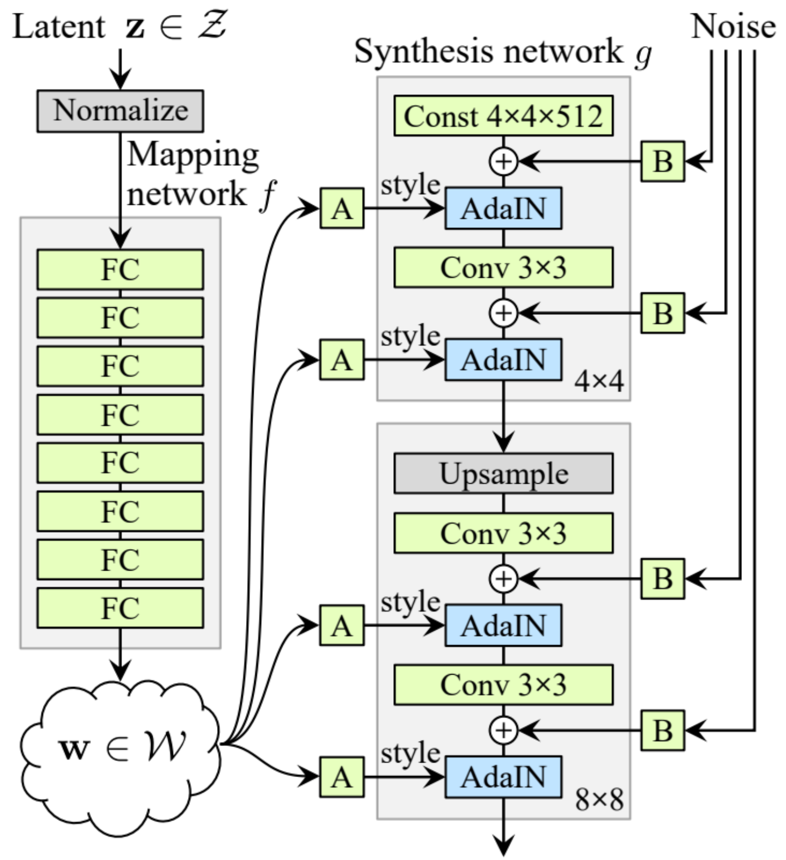
- Enforces Lipschitz continuity on discriminator
  - Spectral normalization
  - Orthogonalization

Spectral normalization for generative adversarial networks [Miyato et al 2018]

Controllable Orthogonalization in Training DNNs [Huang et al, 2020]

Conditional GANs

- A style-based generator architecture for generative adversarial networks [Karras et al, 2019]

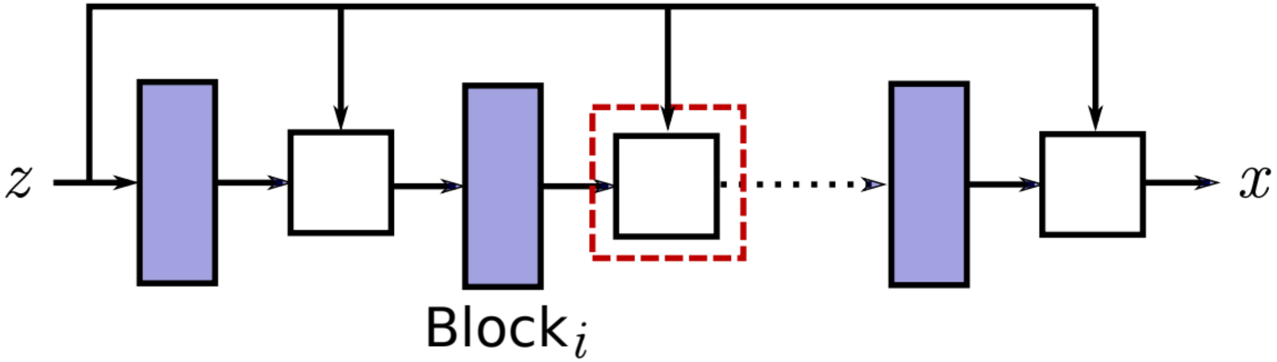


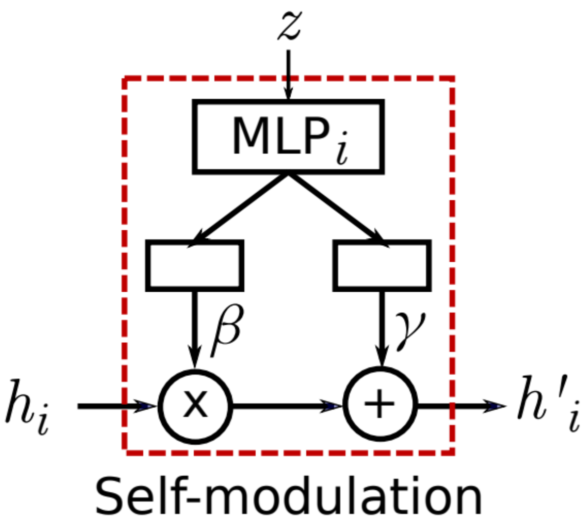
$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}$$

A Style-Based Generator Architecture for Generative Adversarial Networks [Karras et al, CVPR 2019]

Conditional GANs

- Self modulation for generative adversarial networks





$$\boldsymbol{h}'_\ell = \boldsymbol{\gamma}_\ell(\boldsymbol{z}) \odot \frac{\boldsymbol{h}_\ell - \boldsymbol{\mu}}{\boldsymbol{\sigma}} + \boldsymbol{\beta}_\ell(\boldsymbol{z})$$

On Self Modulation for Generative Adversarial Networks [Chen et al, ICLR 2019]

Outline

04. Applications of Normalization

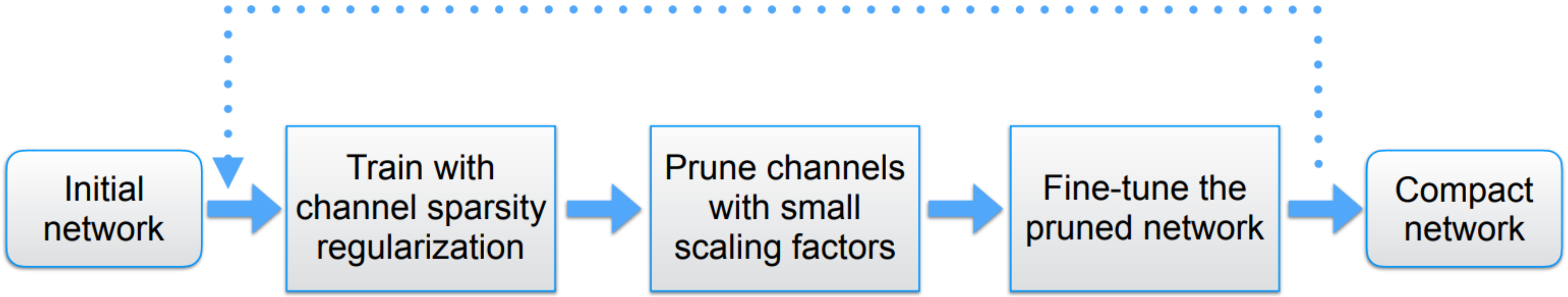
- Control domain
- Control style
- Training GANs
- Efficient model

Efficient Deep Models

- Pruning: exploit the channel-wise scaling layers in BN

Impose sparsity regularization

$$\widetilde{X} = \Psi_{AF}(\widehat{X}) = \widehat{X} \odot (\gamma \mathbf{1}^T) + (\beta \mathbf{1}^T)$$



Initial network → Train with channel sparsity regularization → Prune channels with small scaling factors → Fine-tune the pruned network → Compact network

Learning Efficient Convolutional Networks through Network Slimming [Liu et al, ICCV 2017]

Rethinking the Smaller-Norm-Less-Informative Assumption in Channel Pruning of Convolution Layers [Ye et al, ICLR 2018]

Efficient Deep Models

- Adaptive BN for Pruning

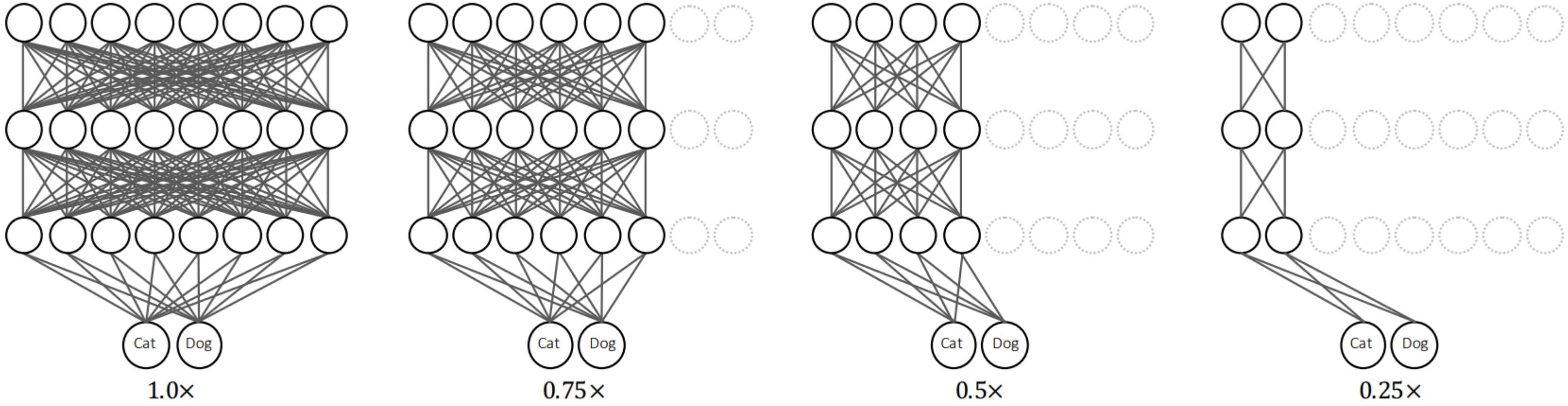
$$\begin{cases} \hat{u} = (1 - \lambda)\hat{u} + \lambda u, \\ \hat{\sigma}^2 = (1 - \lambda)\hat{\sigma}^2 + \lambda\sigma^2 \end{cases}$$

Population statistics needs to be re-calculated after pruning

EagleEye: Fast Sub-net Evaluation for Efficient Neural Network Pruning [Li etal, ECCV 2020]

Efficient Deep Models

- Slimmable network
  - Switchable batch normalization (SBN)

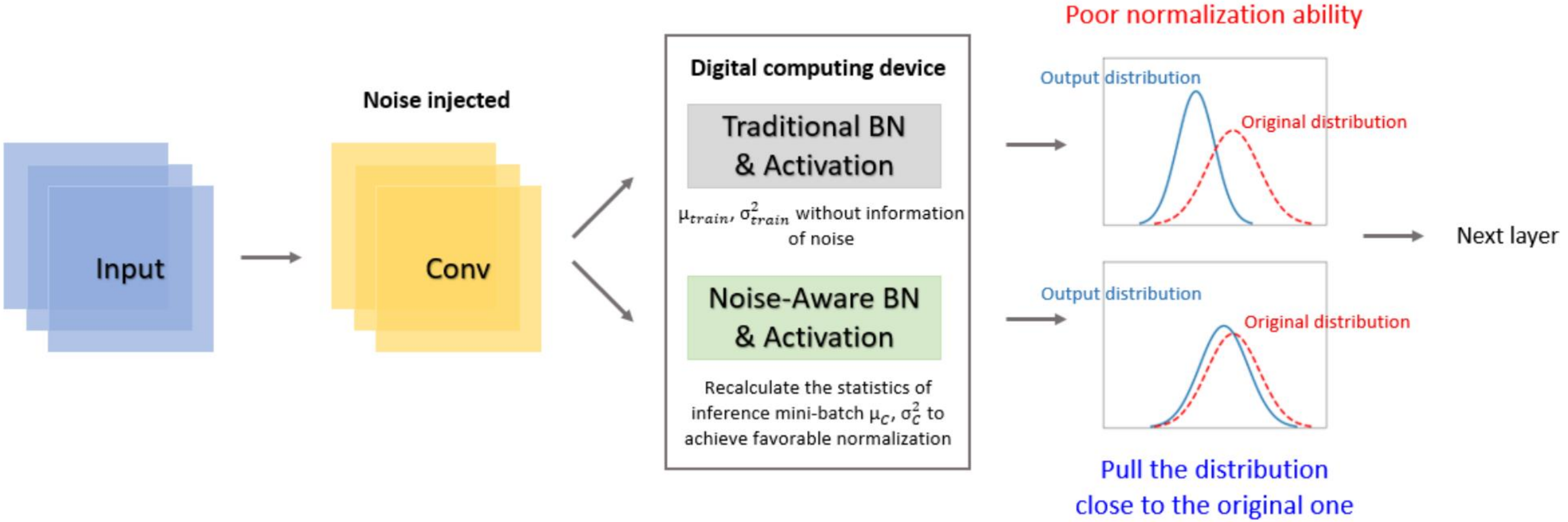


Slimmable Neural Networks [Yu etal, ICLR 2019]

Efficient Deep Models

- Network quantization
  - Noise-Aware BatchNorm

$$F^{Q}(w+\Delta w)\approx F(w)$$



Poor normalization ability

Input

Noise injected

Conv

Digital computing device

Traditional BN & Activation

$\mu_{train}, \sigma^2_{train}$ without information of noise

Noise-Aware BN & Activation

Recalculate the statistics of inference mini-batch μ_C, σ_C^2 to achieve favorable normalization

Output distribution

Original distribution

Output distribution

Original distribution

Next layer

Pull the distribution close to the original one

Robust Processing-In-Memory Neural Networks via Noise-Aware Normalization [Tsai et al, 2020]

Open Discussion

- How to validate BN/IN effectively learning the invariant representation?
- The statistical mechanism of normalization can be used in NLP tasks?
- Why BN/GN in CNN/CV, while LN in Transformer/NLP?
- Possible to make network architecture larger in deep reinforcement learning, by using normalization?

Q&A