Reducing Domain Gap by Reducing Style Bias

(CVPR 2021)

Nam et al.

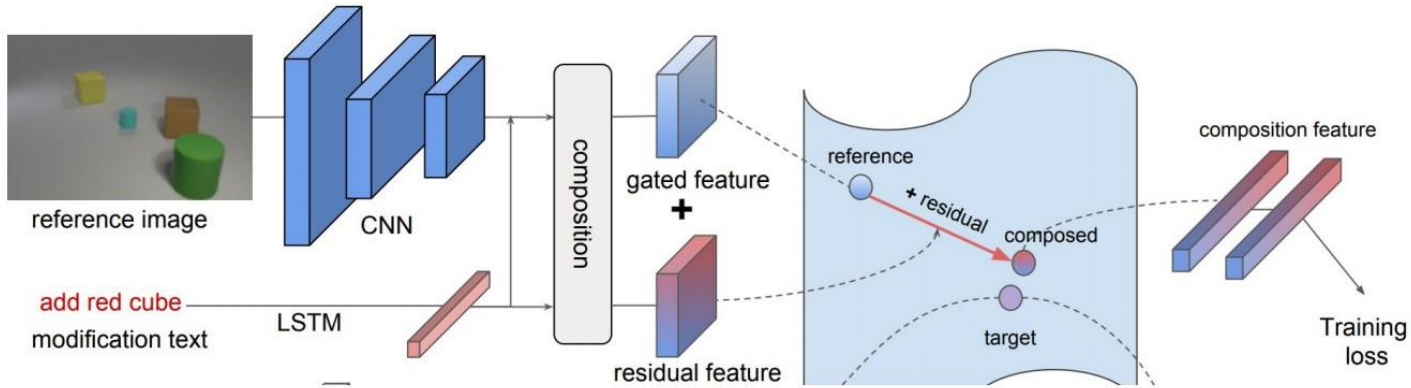
Presenter : Dongjun Kim

Contents

Review

Composing Text and Image for Image Retrieval

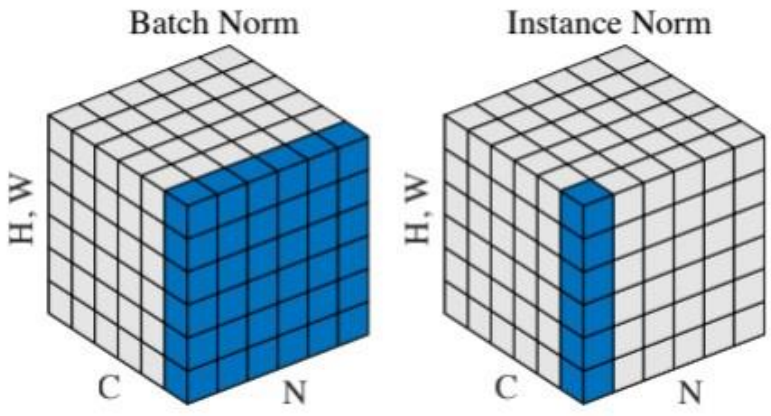
- **Aim :** To retrieve better image with modification text of a given image
- Used TIRG (Text Image Residual Gating) to learn the composition of feature with text + image
- With **gating connection** and **residual connection**, it retains the image feature with modified text and learns similarity between gated features and target image features
- Used **classification loss** from deep metric learning

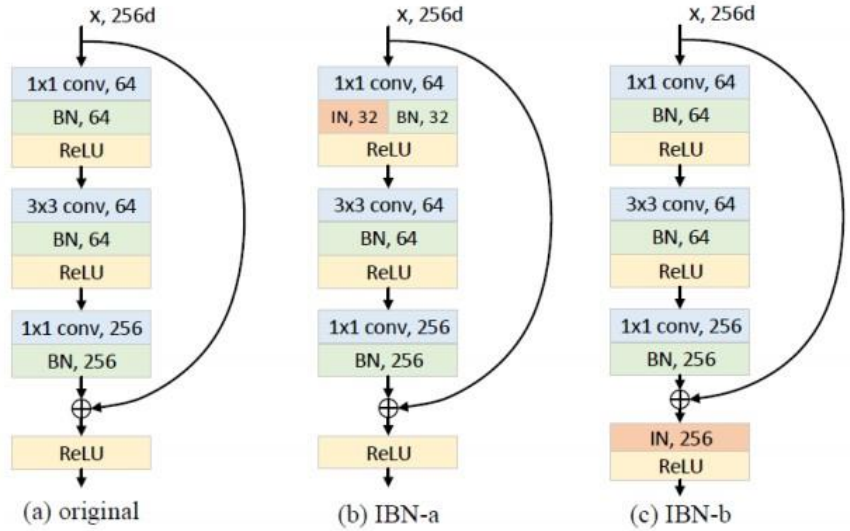


Review

Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net

- **Aim :** Want to have better learning and generalization in unseen domain with no additional target domain data
- They thought appearance variance causes the domain gap, therefore by **Batch Norm(BN)** and **Instance Norm(IN)** in network can control the style variances





Introduction & Motivation

Introduction & Motivation

- CNN architectures often fail to maintain their performance when they confront new test domains, which is known as **domain shift**
- **Main cause :** CNN’s strong bias towards image styles (i.e. textures)

ImageNet-trained CNNs are biased towards texture (ICLR 2019)

Introduction & Motivation

Introduction & Motivation

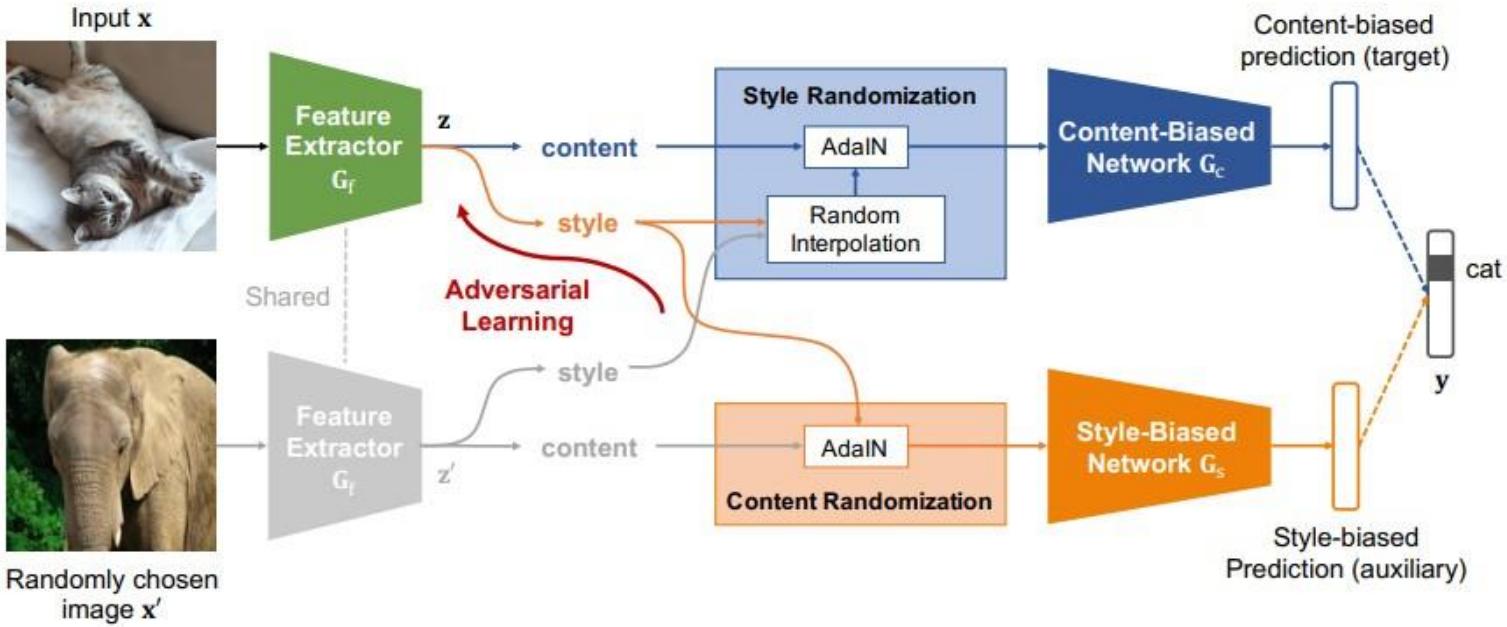
To overcome the domain gap,

- Learn a shared feature space across multiple source domains
- Split the source domains into meta-train and meta-test (Meta Learning)

Introduction & Motivation

Introduction & Motivation

- Propose to reduce the intrinsic style bias of CNNs to close the gap between domains
- Presents **Style-Agnostic Networks (SagNets)** that disentangle style encodings from class categories to prevent style biased predictions



Introduction & Motivation

AdaIN

- In style transfer methods, many of approaches have been introduced
- But the main problem was that the speed of optimization is **too slow**
- Although IN can perform style normalization, can’t tell what specific style could be transferred

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

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

Introduction & Motivation

AdaIN

- With content x and style y, we can adaptively stylize input x with the style y
- We call the normalization process as whitening, and the shifting process as coloring

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

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

Method

Content-biased learning

- Enforce the model to learn content-biased features by introducing a style randomization (SR) module
- Constructs randomize style by interpolating between the styles of z and z', then replaces the style of the input with AdaIN

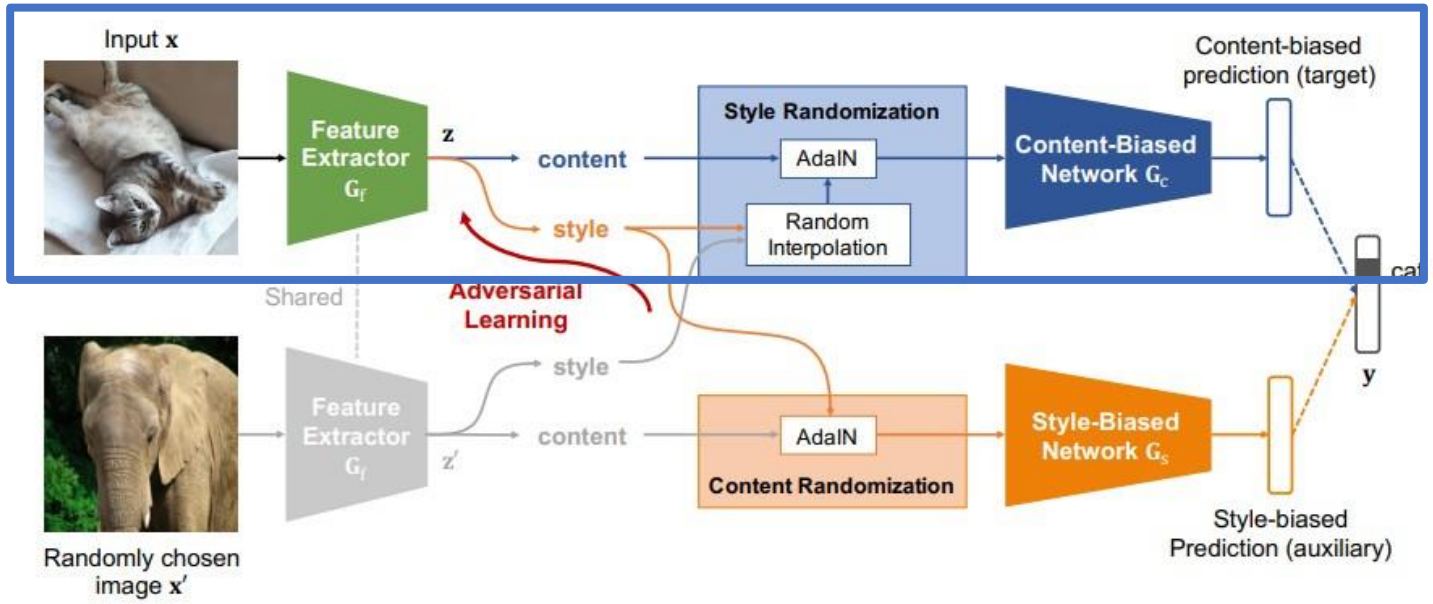
$$\mu(\mathbf{z}) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\mathbf{z}_{hw},$$

$$\sigma(\mathbf{z}) = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(\mathbf{z}_{hw}-\mu(\mathbf{z}))^2+\epsilon}.$$

$$\hat{\mu} = \alpha\cdot\mu(\mathbf{z}) + (1-\alpha)\cdot\mu(\mathbf{z}')$$

$$\hat{\sigma} = \alpha\cdot\sigma(\mathbf{z}) + (1-\alpha)\cdot\sigma(\mathbf{z}')$$

$$\mathrm{SR}(\mathbf{z},\mathbf{z}') = \hat{\sigma}\cdot\left(\frac{\mathbf{z}-\mu(\mathbf{z})}{\sigma(\mathbf{z})}\right)+\hat{\mu},$$

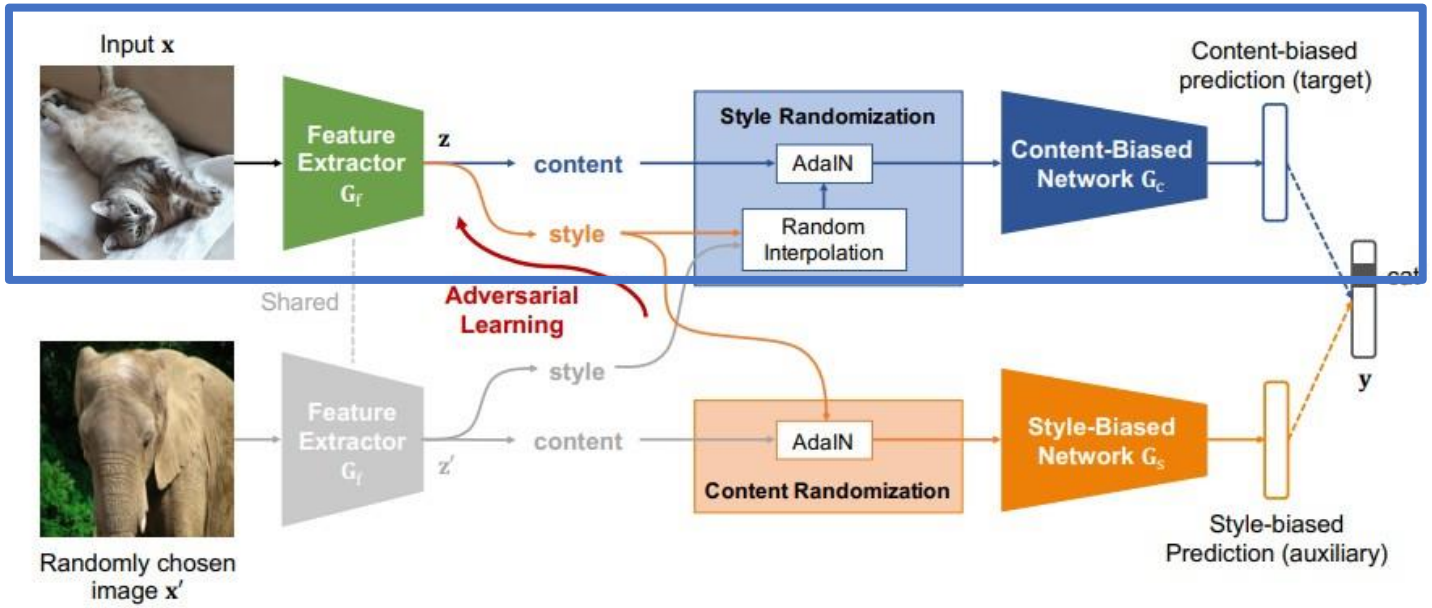


Method

Content-biased learning

- Then the representation is fed into the content-biased network
- Jointly optimizes the both feature extractor and the content-biased network
- K = Number of class categories

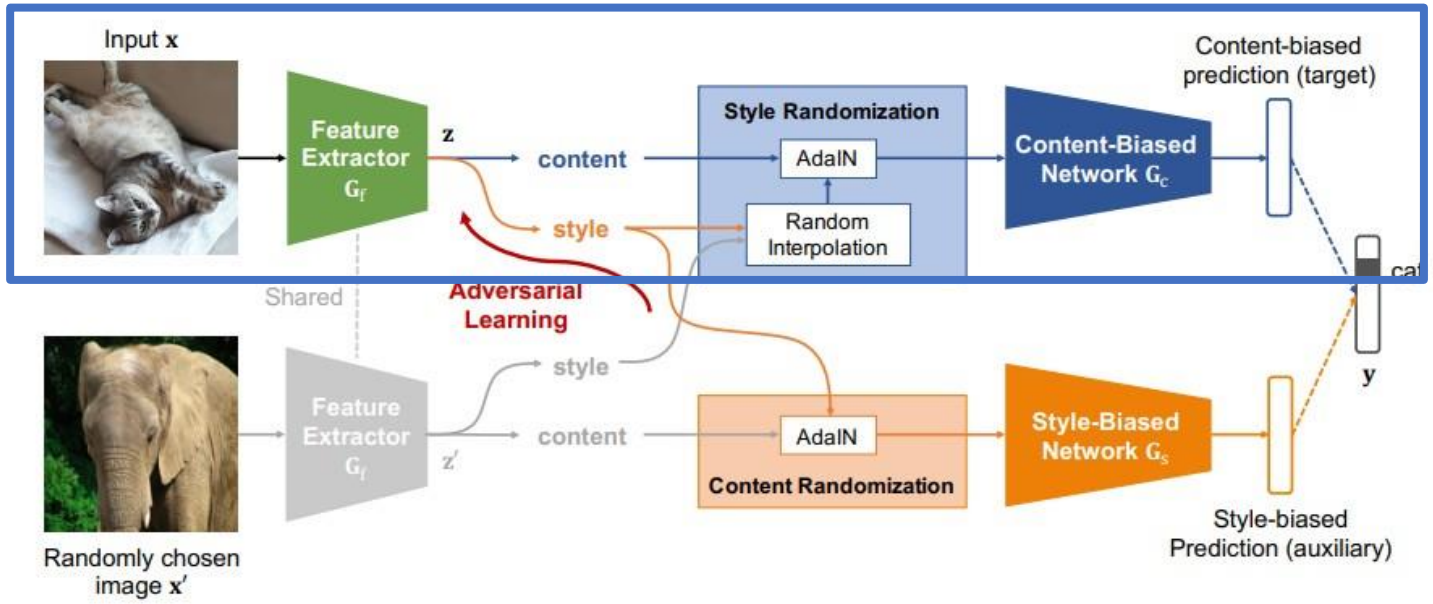
$$\min_{\mathbf{G}_\mathrm{f},\mathbf{G}_\mathrm{c}} L_\mathrm{c} = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\in S} \sum_{k=1}^{K} \mathbf{y}_k \log \mathbf{G}_\mathrm{c}(\mathrm{SR}(\mathbf{G}_\mathrm{f}(\mathbf{x}), \mathbf{z}'))_k$$



Method

Adversarial Style-Biased learning

- Constrain the feature extractor from learning style-biased representation by adopting an adversarial learning framework
- Build an auxiliary style-biased network as a discriminator to make style-biased predictions



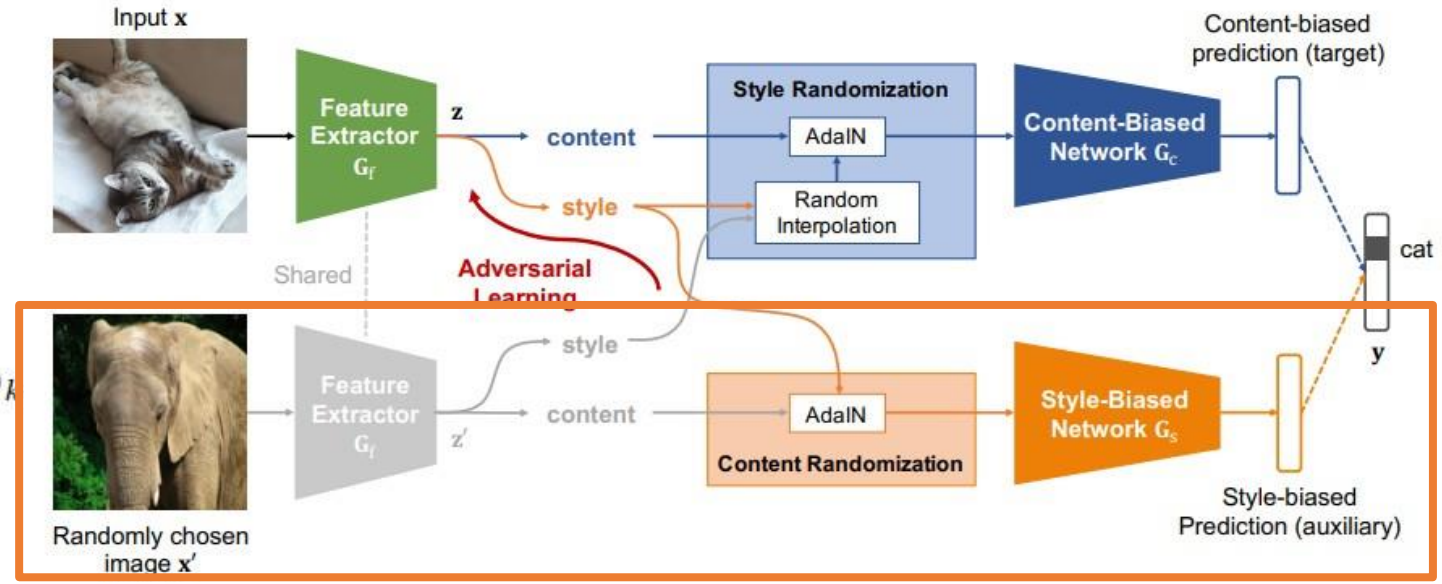
Method

Adversarial Style-Biased learning

- Contrary to SR which leaves the content of the input and randomizes its style, the CR module does **the opposite**
- Apply AdaIN to the content of z' with the style of z
- It is taken as an input to the style-biased network to make a style biased prediction

$$\mathrm{CR}(\mathbf{z}, \mathbf{z}') = \sigma(\mathbf{z}) \cdot \left(\frac{\mathbf{z}' - \mu(\mathbf{z}')}{\sigma(\mathbf{z}')} \right) + \mu(\mathbf{z})$$

$$\min_{\mathbf{G}_s} L_s = -\mathbb{E}_{(\mathbf{x},\mathbf{y}) \in S} \sum_{k=1}^{K} \mathbf{y}_k \log \mathbf{G}_s(\mathrm{CR}(\mathbf{G}_f(\mathbf{x}), \mathbf{z}'))_k$$

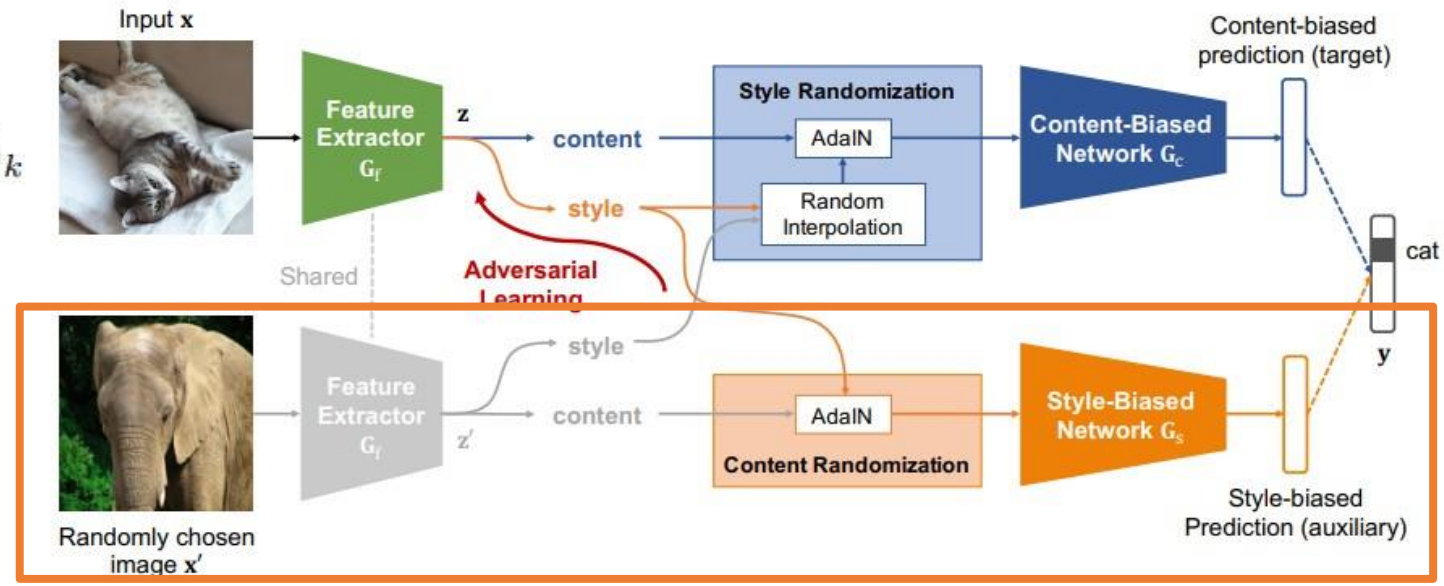


Method

Adversarial Style-Biased learning

- The feature extractor G_f is then trained to fool G_s by minimizing an adversarial loss computed by the cross entropy
- Efficiently control the trade-off between style and content biases by adjusting the coefficient λ_{adv}

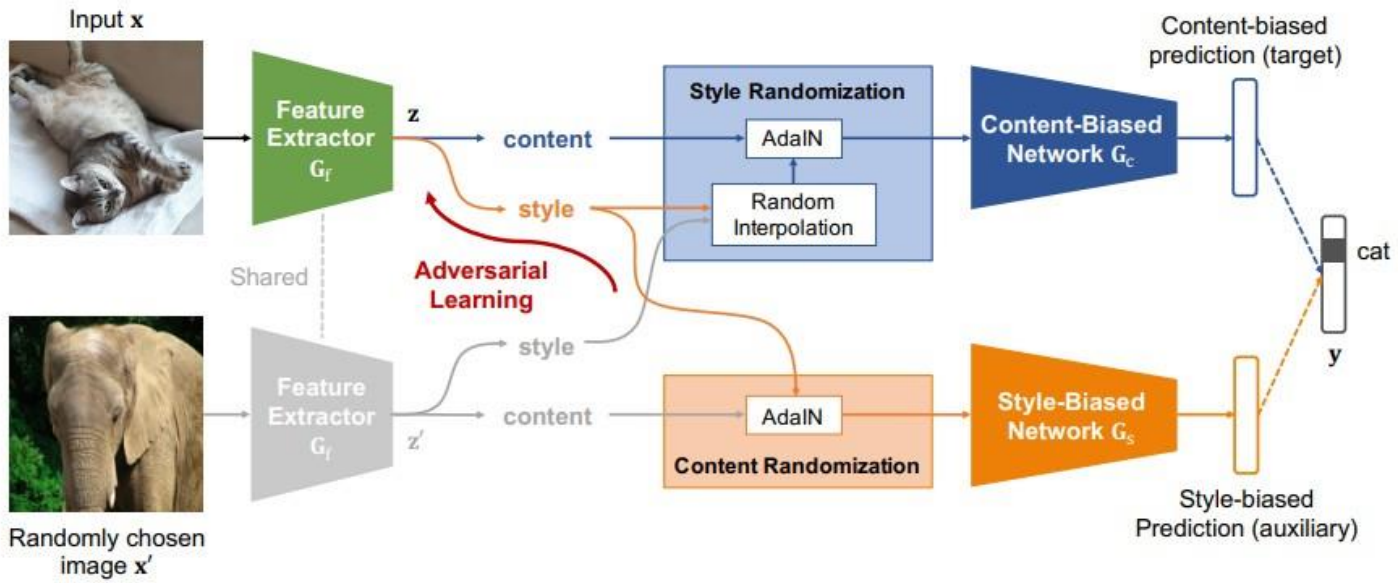
$$\min_{\mathbf{G_f}} L_{\text{adv}} = -\lambda_{\text{adv}} \mathbb{E}_{(\mathbf{x},\cdot)\in S} \sum_{k=1}^{K} \frac{1}{K} \log \mathbf{G}_{\text{s}}(\text{CR}(\mathbf{G}_{\text{f}}(\mathbf{x}), \mathbf{z}'))_k$$



Method

Implementation

- The style-biased network forms the same structure with the content-biased network
- Meaning that, no overheads and same network architectures as CNN
- Trained end-to-end



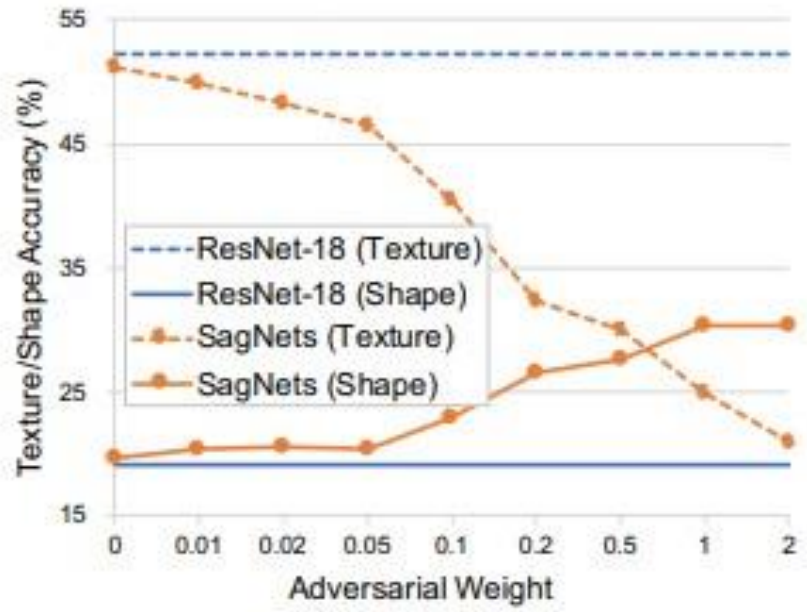
Experiments

Cue conflict stimuli

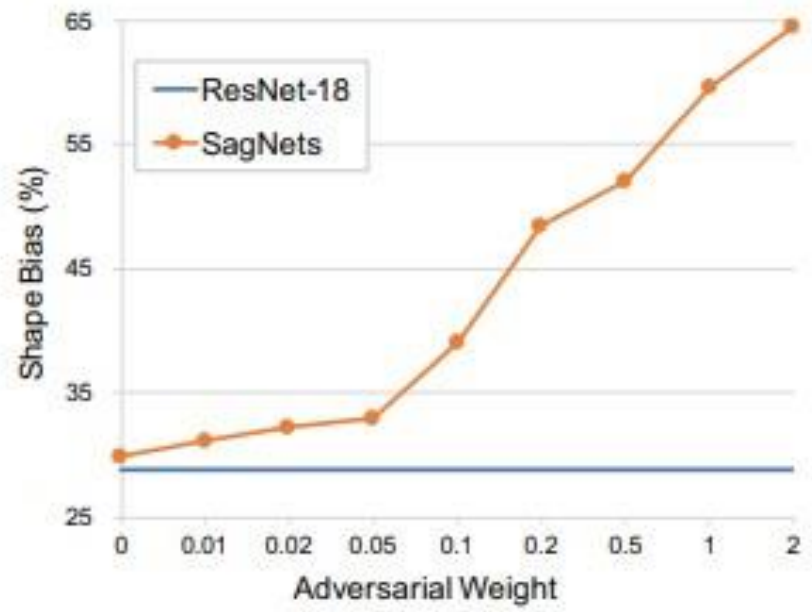
Experiments

Texture/Shape bias

- Quantify the texture and shape biases by evaluating them on the cue conflict stimuli and counting the number of predictions



(a) Texture/shape accuracy



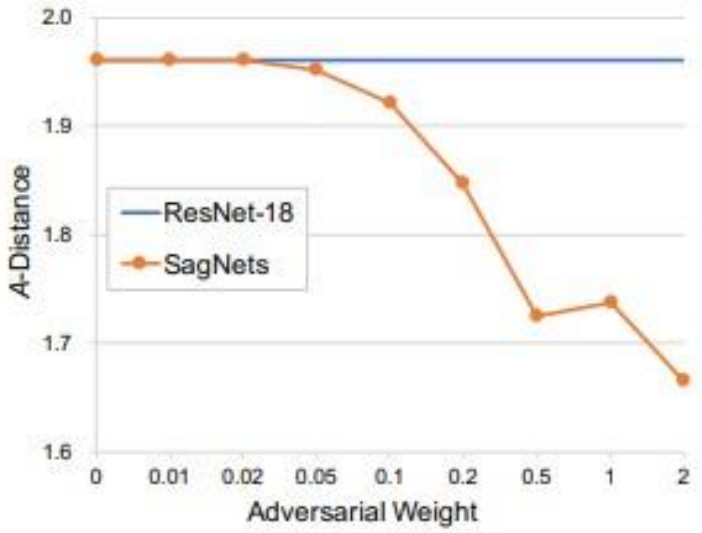
(b) Shape bias

Experiments

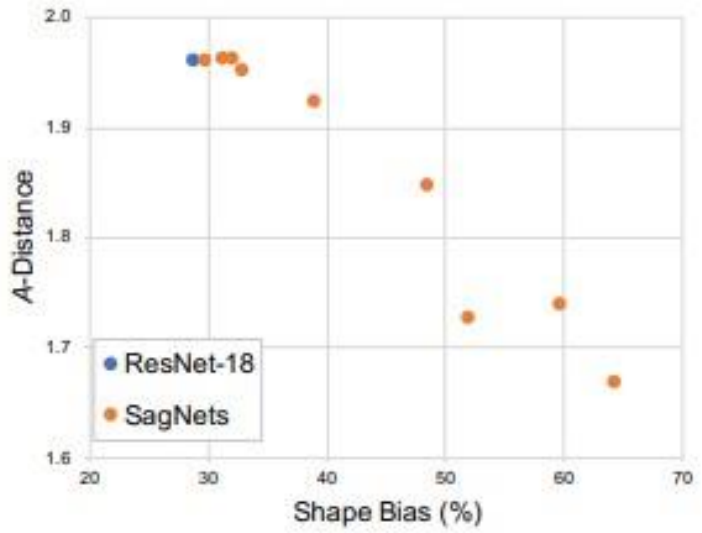
Domain Gap

- Measure distance between the two domains (ImageNet, Conflict Cue)
- Where the epsilon, is a generalization error of SVM classifier
- Shape-biased representation generalizes better across domains

$\mathcal{A}$-distance $d_{\mathcal{A}} = 2(1 - \epsilon)$



(c) Domain discrepancy



(d) Domain discrepancy *vs* shape bias

Experiments

Domain Generalization

- Generalize from multiple source domains to a novel target domain (unseen)

	Art paint.	Cartoon	Sketch	Photo	Avg.
AlexNet					
D-SAM	63.87	70.70	64.66	85.55	71.20
JiGen	67.63	71.71	65.18	89.00	73.38
Epi-FCR	64.7	72.3	65.0	86.1	72.0
MASF	70.35	72.46	67.33	**90.68**	75.21
MMLD	69.27	**72.83**	66.44	88.98	74.38
DeepAll	65.19	67.83	63.75	90.08	71.71
SagNet	**71.01**	70.78	**70.26**	90.04	**75.52**
ResNet-18					
D-SAM	77.33	72.43	**77.83**	95.30	80.72
JiGen	79.42	75.25	71.35	96.03	80.51
Epi-FCR	82.1	77.0	73.0	93.9	81.5
MASF	80.29	77.17	71.69	94.99	81.04
MMLD	81.28	77.16	72.29	**96.09**	81.83
DeepAll	78.12	75.10	68.43	95.37	79.26
SagNet^{-CBL}	78.86	77.05	73.28	95.43	81.15
SagNet^{-ASBL}	82.94	76.73	74.74	95.07	82.37
SagNet	**83.58**	**77.66**	76.30	95.47	**83.25**

Conclusion

Conclusion

- Presents Style-Agnostic Networks (SagNets) that are robust against domain shift caused by style variability

- By randomizing styles in a latent feature space, SagNets rely more on contents rather than styles

Thank you !