Domain Transfer and Adaptation

EE807: Recent Advances in Deep Learning

Lecture 12

Slide made by

Sangwoo Mo

KAIST EE

Table of Contents

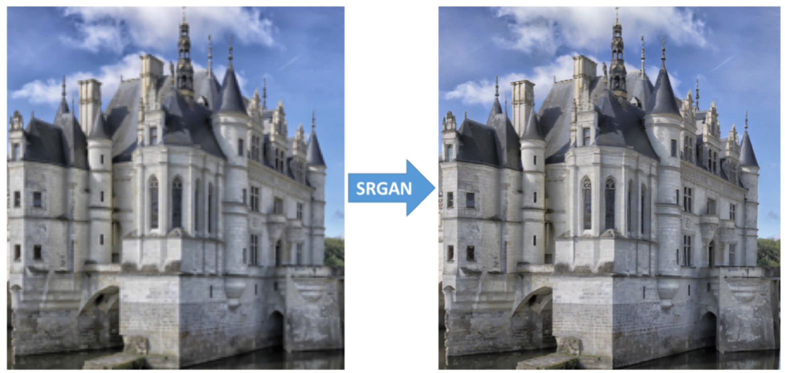
Table of Contents

What is Domain Transfer?

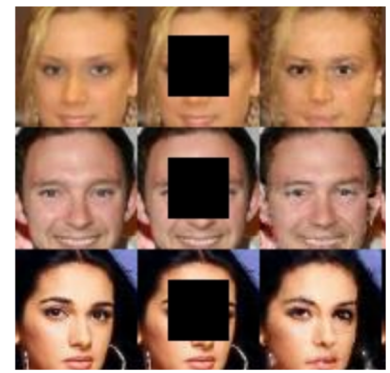
- Learning a mapping **between two (or more) domains**
  - Each domain is typically, described by a set of data samples.
  - Given X_S and X_T, learn a mapping $f: X_S \rightarrow X_T$

Colorization[1]

Super-resolution[2]



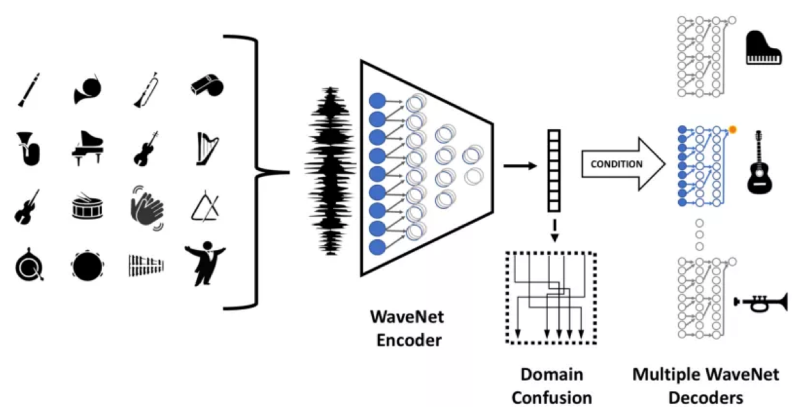
Inpainting[3]



Machine Translation[4]

Source	Proposed system (full)
Une fusillade a eu lieu à l'aéroport international de Los Angeles.	A shooting occurred at Los Angeles International Airport.
Cette controverse croissante autour de l'agence a provoqué beaucoup de spéculations selon lesquelles l'incident de ce soir était le résultat d'une cyber-opération ciblée.	This growing scandal around the agency has caused much speculation about how this incident was the outcome of a targeted cyber operation.

Music Style Transfer[5]



1. [Zhang et al., 2016]; 2. [Ledig et al., 2017]; 3. [Yeh et al., 2017]; 4. [Artetxe et al., 2018]; 5. [Mor et al., 2018]

What is Domain Adaptation?

- Learning a mapping **between two (or more) domains with labels**
  - A special case of transfer learning
  - Given (X_S, Y_S) and X_T, learn a mapping $f: X_T \to Y_T$

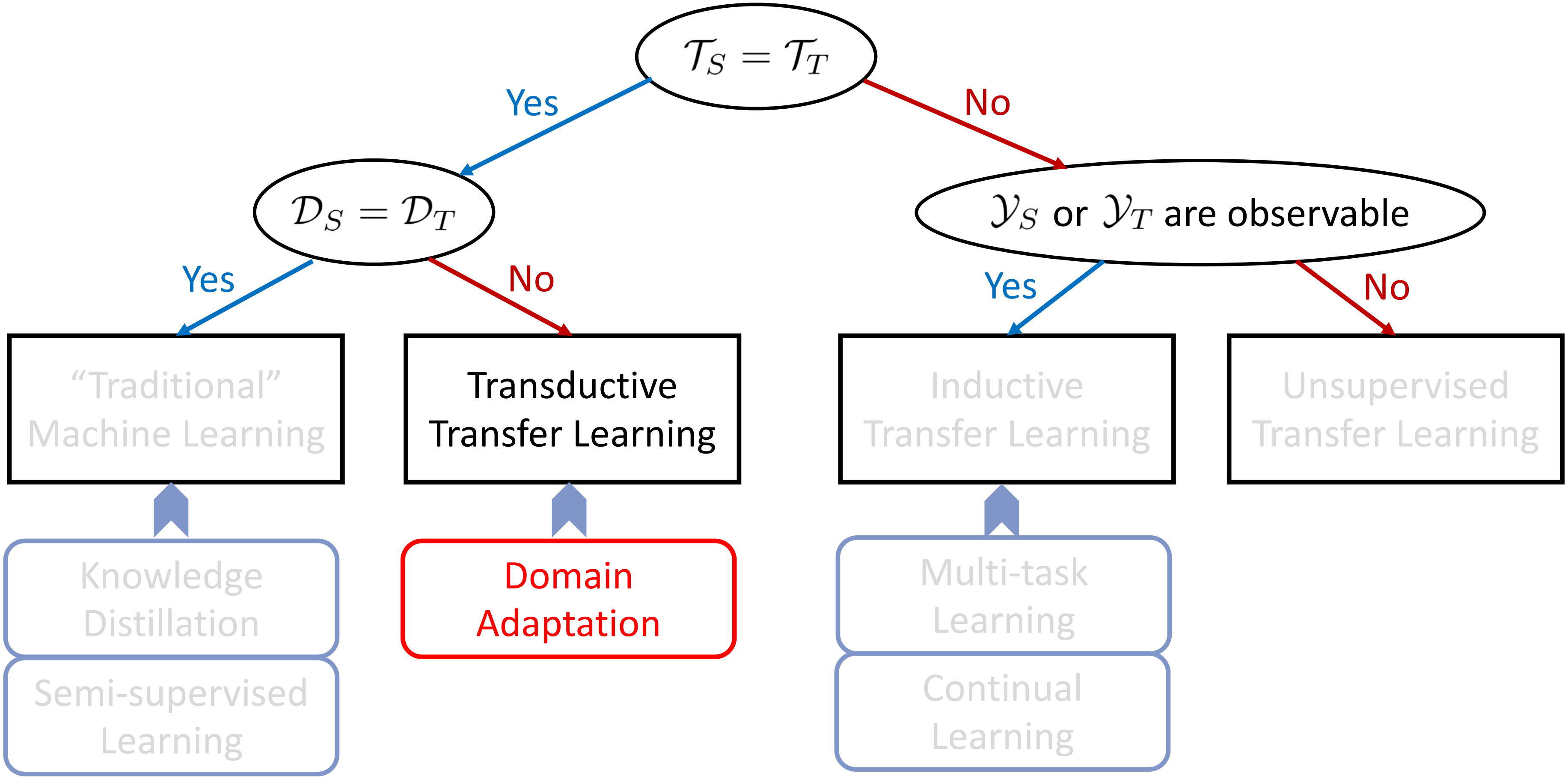


*Figure made by Jongjin Park

Table of Contents

General Approaches for Domain Transfer

- **Domain (or style) transfer** aims to
  - Keep content of source data
  - Change style to match with target data (or domain)

- **General optimizing objective for producing outputs:** $\mathcal{L}_{\text{content}} + \lambda \mathcal{L}_{\text{style}}$
  - **Domain transfer** research is about designing content & style losses

*Source: Ulyanov et al. "Instance Normalization: The Missing Ingredient for Fast Stylization", arXiv 2016

Neural Style Transfer [Gatys et al., 2016]

- **Idea:** Use a well pretrained (e.g., by the ImageNet dataset) neural network for content & style losses
  - **Goal:** Given inputs x_c and x_s, generate $\hat{x}$ having content of x_c and style of x_s
  - **Content loss:**
    - High layer of NN contains (abstract) content information
    - Match feature maps of **high** layers

$$\mathcal{L}_{\text{content}}(x_c, x) = \frac{1}{2}\sum_{i,k}(F_{ik}^{l}(x_c) - F_{ik}^{l}(x))^2$$

where F_{ijk}^{l} denotes feature of $l \in \{1, \dots, L\}$-th layer, $i \in \{1, \dots, H^l \times W^l\}$ denotes spatial location, and $k \in \{1, \dots, C^l\}$ denotes channel

Neural Style Transfer [Gatys et al., 2016]

- **Idea:** Use a well pretrained (e.g., by the ImageNet dataset) neural network for content & style losses
  - **Goal:** Given inputs x_c and x_s, generate $\hat{x}$ having content of x_c and style of x_s

  - **Style loss:**
    - It has been observed that **feature statistics** contains style information
      - Feature statistics of $\mathbb{P}^l(z;x)$ where

$$z_{i;x} := (F^l_{i1}(x), \ldots, F^l_{iC^l}(x)) \in \mathbb{R}^{C^l} \sim \mathbb{P}^l(z;x)$$

    - Match features statistics (or underlying p.d.f.) of **low** layers

$$\mathcal{L}_{\text{style}}(x_s, x) = D(\mathbb{P}^l(z;x_s) \| \mathbb{P}^l(z;x))$$

*D is some function distance, e.g., maximum mean discrepancy (MMD) or moment matching

Neural Style Transfer [Gatys et al., 2016]

- **Idea:** Use a well pretrained (e.g., by the ImageNet dataset) neural network for content & style losses
  - **Goal:** Given inputs x_c and x_s, generate $\hat{x}$ having content of x_c and style of x_s
  - **Style loss:**
    - Match features statistics (or underlying p.d.f.) of **low** layers

$$\mathcal{L}_{\text{style}}(x_s, x) = D(\mathbb{P}^l(z; x_s) || \mathbb{P}^l(z; x))$$

    - For example, one can minimize a Frobenius norm of **Gramian matrix** G^l, where (i, j)-th element of G^l is given by

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l$$

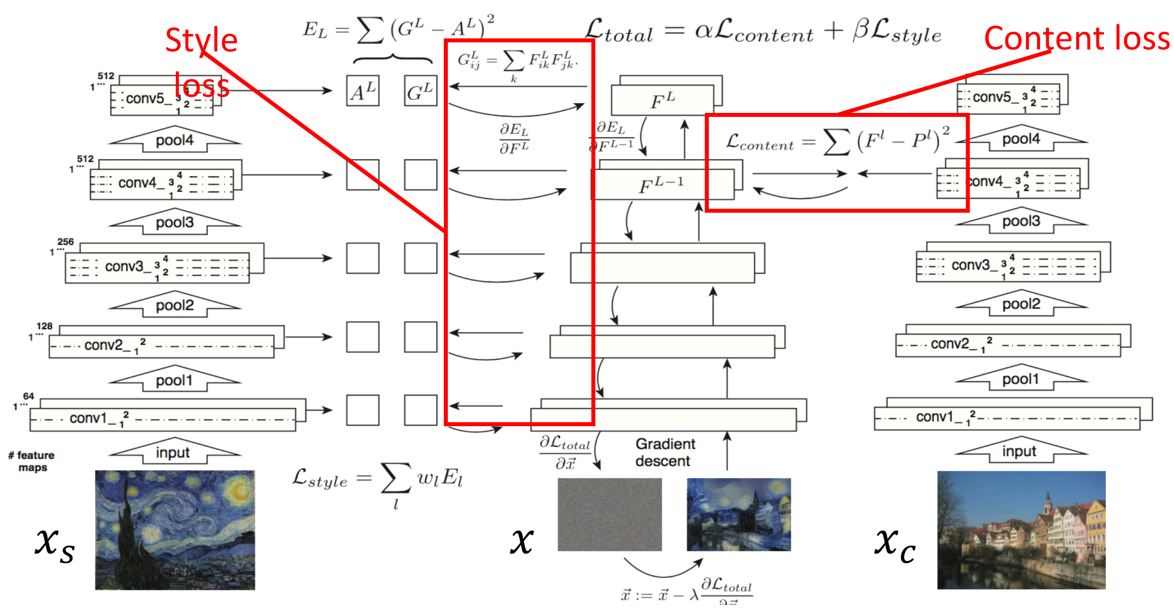
hence,

$$\mathcal{L}_{\text{style}}(x_s, x) = \sum_{i,j} (G_{ij}^l(x_s) - G_{ij}^l(x))^2$$

*Frobenius norm of Gramian matrix identical to minimize MMD with kernel $k(x, y) = (x^T y)^2$ [Li et al., 2017]

Neural Style Transfer [Gatys et al., 2016]

- **Idea:** Use a well pretrained (e.g., by the ImageNet dataset) neural network for content & style losses
  - **Goal:** Given inputs x_c and x_s, generate $\hat{x}$ having content of x_c and style of x_s
  - Used **4th** layers for **content** loss and **[1-5]th** layers for **style** loss



- **Inference:** Solve the following optimization problem (for each (x_c, x_s))

$$\hat{x} = \min_x \mathcal{L}_{\text{content}}(x_c, x) + \lambda \mathcal{L}_{\text{style}}(x_s, x)$$

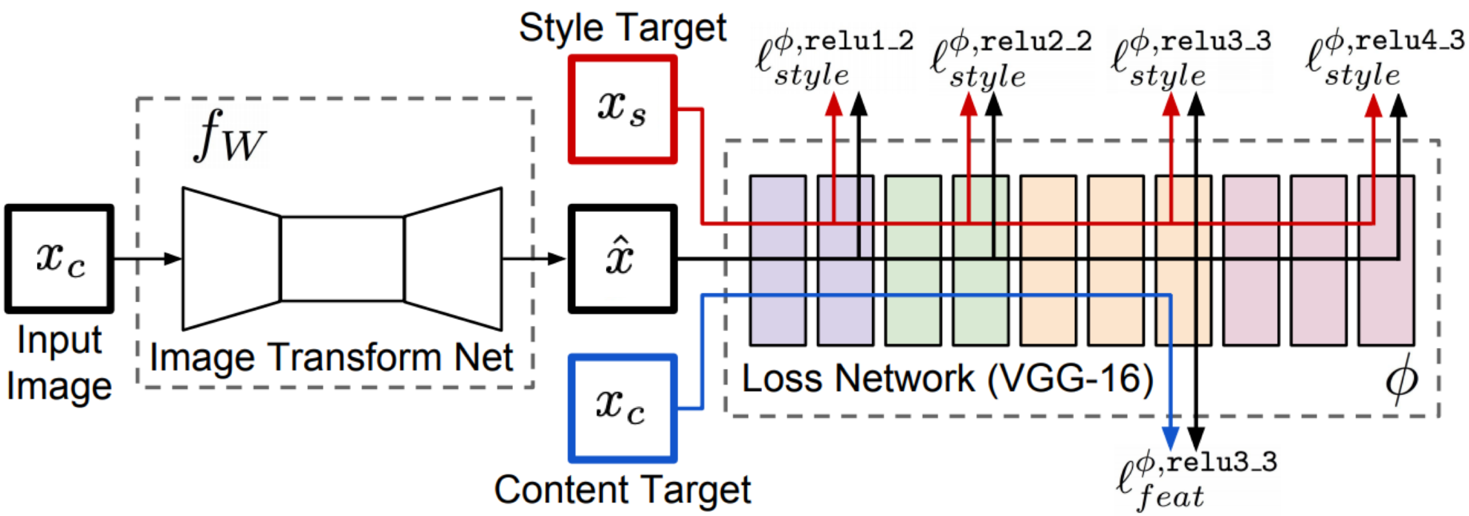
Neural Style Transfer [Gatys et al., 2016]

- **Experimental results:** Neural Style succeeds in translating styles of images

Fast Neural Style [Johnson et al., 2016]

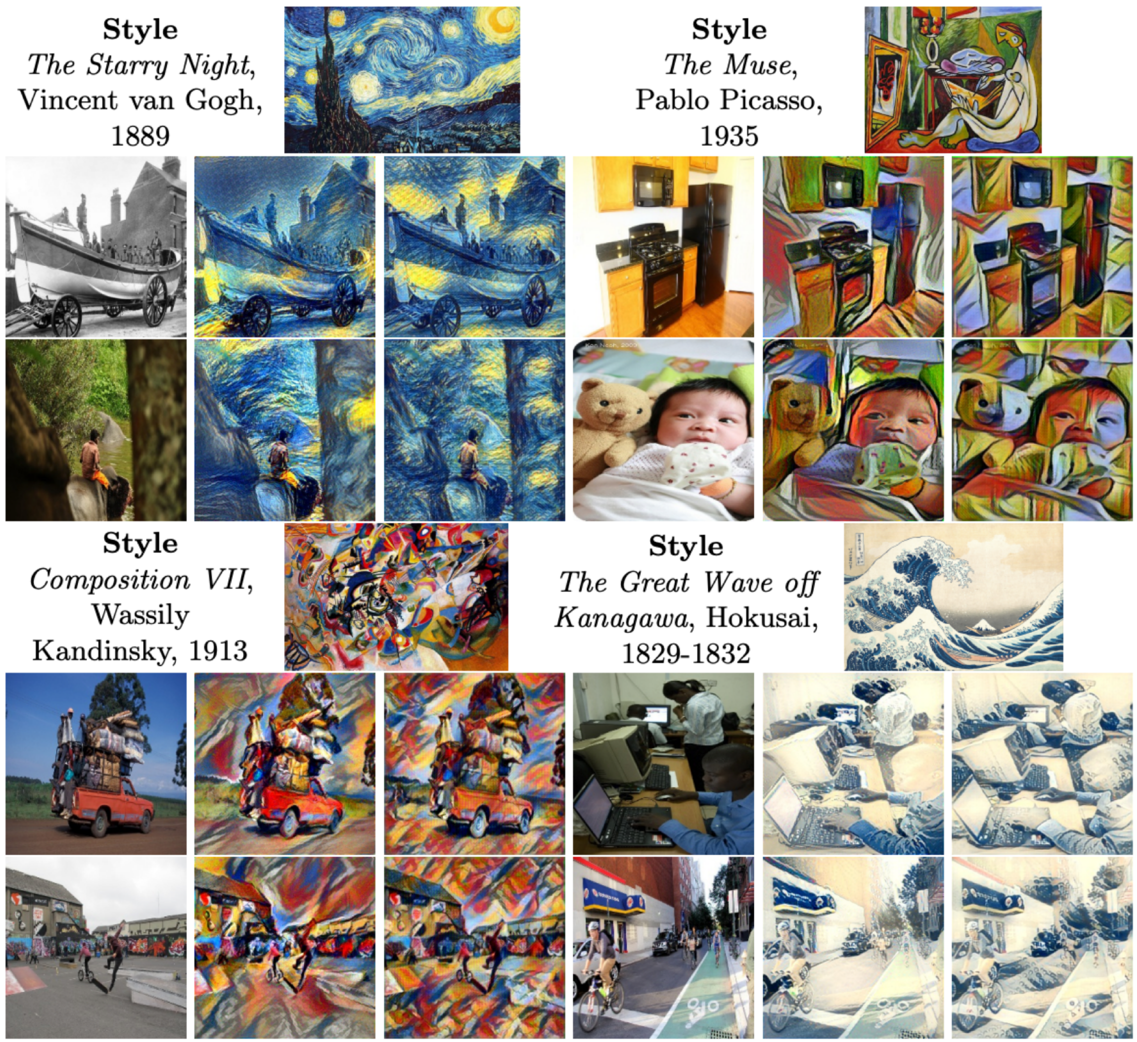
- **Motivation:** Inference of Neural Style is too slow!
  - Namely, one has to solve some optimization per each inference
  - **Idea:** Instead of solving optimization problems, train a neural network f_W which translates input x_c to have the style of x_s (style is fixed for a single network)

  - Now, inference is a **single forward step** $\hat{x} = f_W(x_c)$, which is **100x faster** than Neural Style (solving some optimization problems)
  - The loss function is identical to Neural Style (defined by a pretrained network)



Fast Neural Style [Johnson et al., 2016]

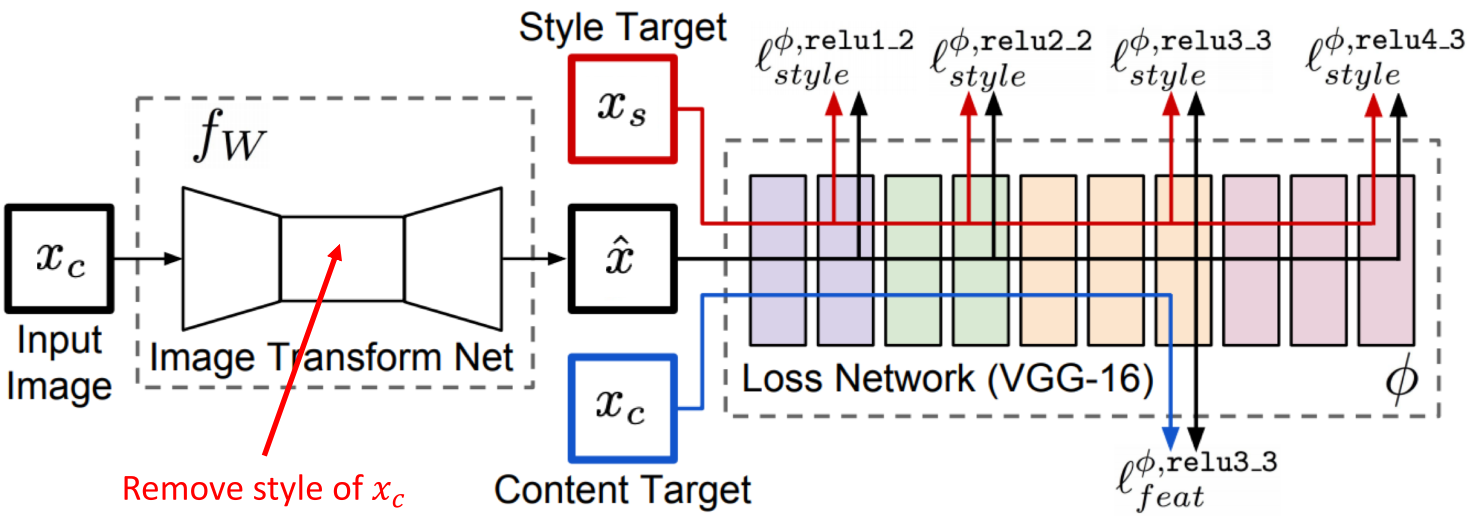
- **Experimental results:** Fast Neural Style is often as good as Neural Style



Left: original
Middle: Gatys et al., 2016
Right: Johnson et al., 2016

Instance Normalization [Ulyanov et al., 2016]

- **Motivation:** Can we design a better network architecture for domain transfer?
  - **Idea:** Removing the original style of x_c would make **restyling** easier



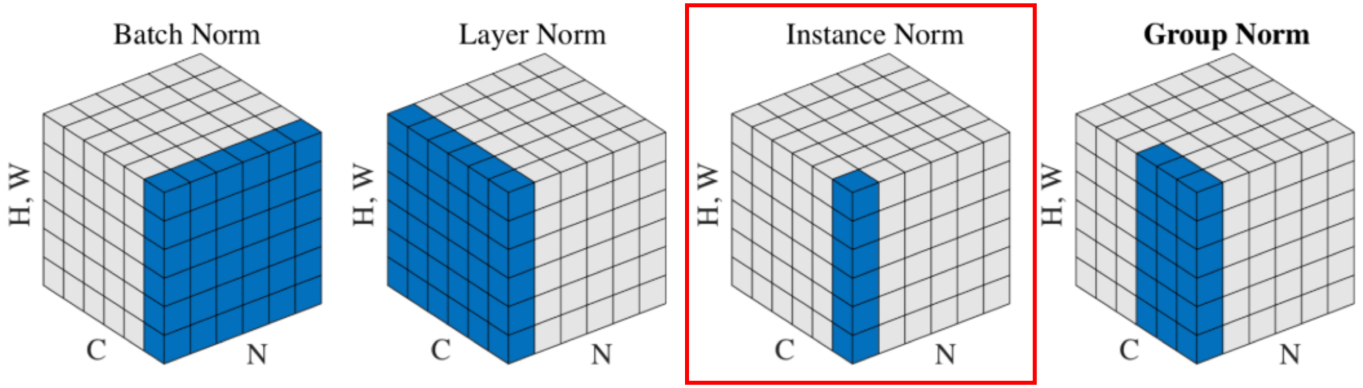
  - **More idea:** As observed in Neural Style, **feature statistics** represents the style
  - Hence, normalizing feature statistics will remove the original style

*Original motivation of IN was to normalize contrast, but recent studies [Li et al., 2017] suggest the real reason of improvement is normalizing feature statistics

Instance Normalization [Ulyanov et al., 2016]

- **Motivation:** Can we design a better network architecture for domain transfer?
  - **Idea:** normalizing feature statistics to remove the original style
  - In particular, normalize **1st & 2nd moments** (i.e., mean & variance)

  - Similar to batch normalization (BN), but for a **single instance**



H: height
W: width
C: channel
N: batch size

Instance Normalization [Ulyanov et al., 2016]

- **Motivation:** Can we design a better network architecture for domain transfer?
  - **Idea:** normalizing feature statistics to remove the original style
  - In particular, normalize **1st & 2nd moments** (i.e., mean & variance)
  - Formally, for features $z_{ik} \in \mathbb{R}^{H^l \times W^l} \times \mathbb{R}^{C^l}$, **Instance Normalization (IN)** is

$$\mathrm{IN}(z_{ik}) = \gamma \left(\frac{z_{ik} - \mu_k}{\sigma_k} \right) + \beta$$

where

$$\mu_k = \frac{1}{H^l \times W^l} \sum_{i=1}^{H^l \times W^l} z_{ik}, \qquad \sigma_k = \frac{1}{H^l \times W^l} \sum_{i=1}^{H^l \times W^l} (z_{ik} - \mu_k)^2$$

  - One can also learn affine parameters **separately** for each domain (CIN [Dumoulin et al., 2017]) or **adaptively** from target style y (AdaIN [Huang et al., 2017])

*Source: Wu et al. "Group Normalization", ECCV 2018

Instance Normalization [Ulyanov et al., 2016]

- **Experimental results**: Instance Normalization helps for Fast Neural Style

- **Motivation:** Can removing style also improve performance of classification?
  - Naïvely applying IN **hurts** performance as style contains some useful info.
  - **Idea:** use linear combination of BN and IN **(learn weight ρ)**

  - For simplicity, let $z^{(B)}$ and $z^{(I)}$ are normalized features of BN and IN, that

$$z_{ik}^{(B)} = \frac{z_{ik} - \mu_k^{(B)}}{\sigma_k^{(B)}}, \qquad z_{ik}^{(I)} = \frac{z_{ik} - \mu_k^{(I)}}{\sigma_k^{(I)}},$$

    and μ, σ are defined as before (normalize for $H^l \times W^l \times B$ with batch size B for BN, and normalize for $H^l \times W^l$ for IN)

  - Here, **Batch-Instance Normalization (BIN)** is

$$\mathrm{BIN}(z_{ik}) = \gamma \left(\rho \cdot z_{ik}^{(B)} + (1 - \rho) \cdot z_{ik}^{(I)}\right) + \beta$$

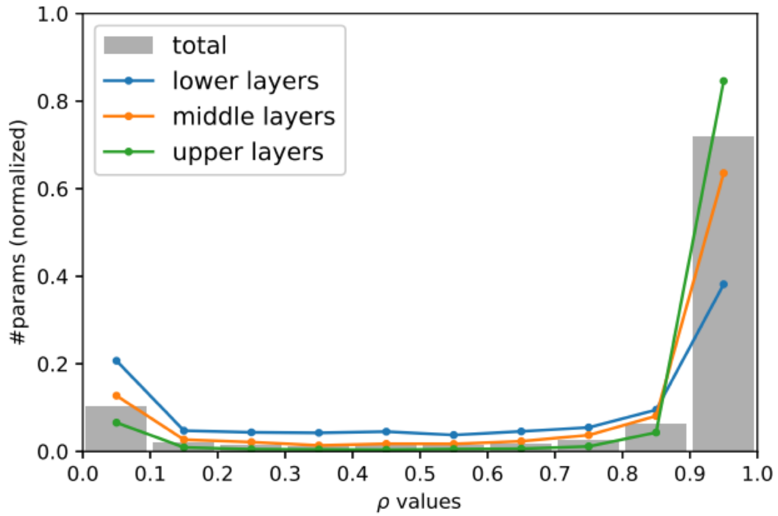
Batch-Instance Normalization [Nam et al., 2018]

- **Motivation:** Can removing style also improve performance of classification?
  - Naïvely applying IN **hurts** performance as style contains some useful info.
  - **Idea:** use linear combination of BN and IN **(learn parameter ρ)**

  - **Batch-Instance Normalization (BIN)** is linear combination of BN and IN

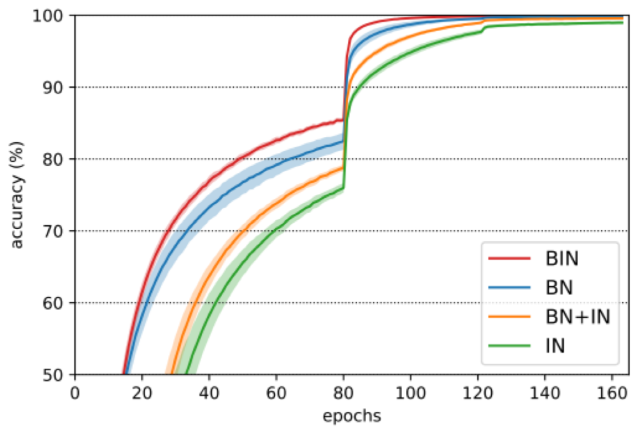
$$\mathrm{BIN}(x) = \gamma\left(\rho \cdot x^{(B)} + (1-\rho)\cdot x^{(I)}\right) + \beta$$

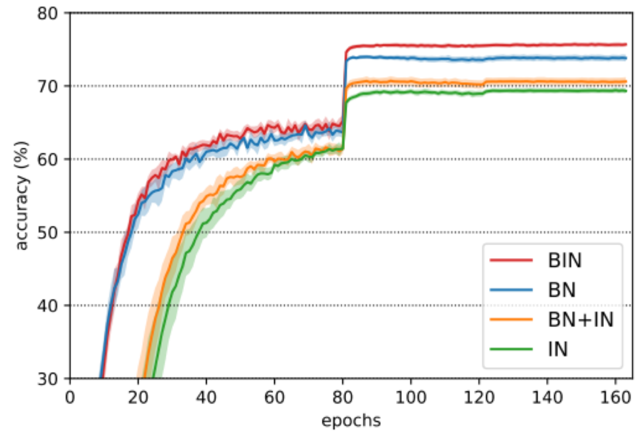
  - After training, **low** layers use **IN** more, and **high** layers use **BN** more



Batch-Instance Normalization [Nam et al., 2018]

- **Experimental results:** BIN shows better performance than BN





Left: train accuracy
Right: test accuracy
CIFAR-100 results

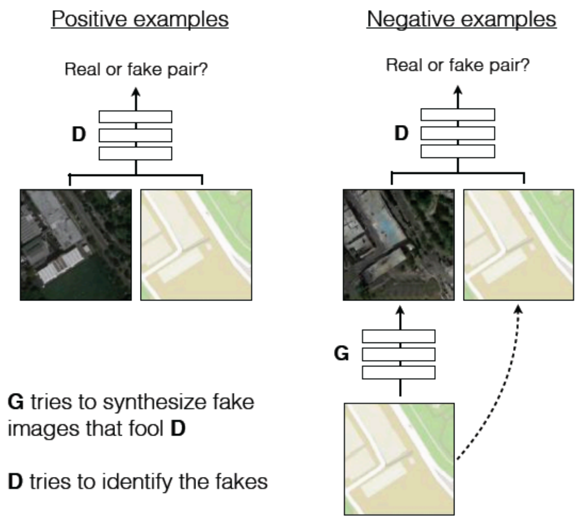
	AlexNet	VGG-19	ResNet-56	ResNet-110
BN	50.62	72.29	72.92	74.26
BIN	**51.00**	**72.50**	**75.05**	**75.88**
	PreResNet-110	WRN-28-10	ResNeXt-29, 8×64d	DenseNet-BC (L100, k12)
BN	76.49	80.92	80.50	76.93
BIN	**77.84**	**81.48**	**81.57**	**77.80**

Top-1 accuracy for CIFAR-100

*Source: Nam et al. "Batch-Instance Normalization for Adaptively Style-Invariant Neural Networks", NIPS 2018

pix2pix [Isola et al., 2017]

- **Motivation:** Neural Style shows good performance for *artistic styles*, but often fails to generate realistic outputs for more complex domain transfer
  - **Idea:** Use GAN (which known to produce realistic images) for style loss
  - **Goal:** Given source domain X_S and target domain X_T, learn a mapping $f: X_S \rightarrow X_T$
    - In prior terminology, generate x_t' with content of $x_s \in X_S$ and style of X_T
  - **Style loss:**
    - Generator G **fools** discriminator D which guesses if data is in **target domain**



$$\mathcal{L}_{\text{style}} = \mathbb{E}_{x_t}[\log D(x_t)] + \mathbb{E}_{x_s}[\log(1 - D(G(x_s)))]$$

pix2pix [Isola et al., 2017]

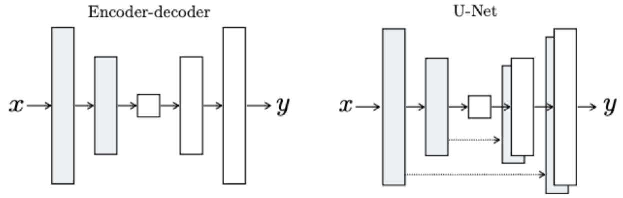
- **Motivation:** Neural Style shows good performance for *artistic styles*, but often fails to generate realistic outputs for more complex domain transfer
  - **Idea:** Use GAN (which known to produce realistic images) for style loss
  - **Goal:** Given source domain X_S and target domain X_T, learn a mapping $f: X_S \to X_T$
    - In prior terminology, generate x_t' with content of $x_s \in X_S$ and style of X_T

  - **Content loss:**
    - Similar to Neural Style, one can use **neural network-defined** content loss
    - However, if we have paired data (x_s, x_t), we don't need such a network, but can simply apply $\boldsymbol{L_1}$ **loss**

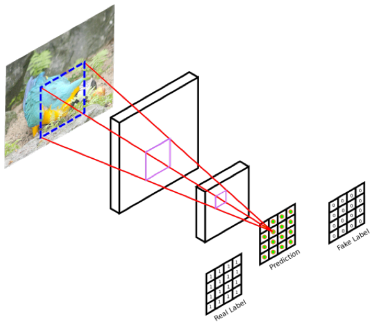
$$\mathcal{L}_{\text{content}} = \mathbb{E}_{x_s}[\|x_t - G(x_s)\|_1]$$

- **Motivation:** Neural Style shows good performance for *artistic styles*, but often fails to generate realistic outputs for more complex domain transfer
  - **Idea:** Use GAN (which known to produce realistic images) for style loss
  - **Goal:** Given source domain X_S and target domain X_T, learn a mapping $f: X_S \rightarrow X_T$
    - In prior terminology, generate x_t' with content of $x_s \in X_S$ and style of X_T

  - In addition, pix2pix propose **novel architectures**
    - **Skip-connection generator** (e.g., U-Net [Ronneberger et al., 2015])



    - **PatchGAN discriminator** (output is $N \times N$ matrix rather than a single scalar)



*Source: https://www.groundai.com/project/patch-based-image-inpainting-with-generative-adversarial-networks/

pix2pix [Isola et al., 2017]

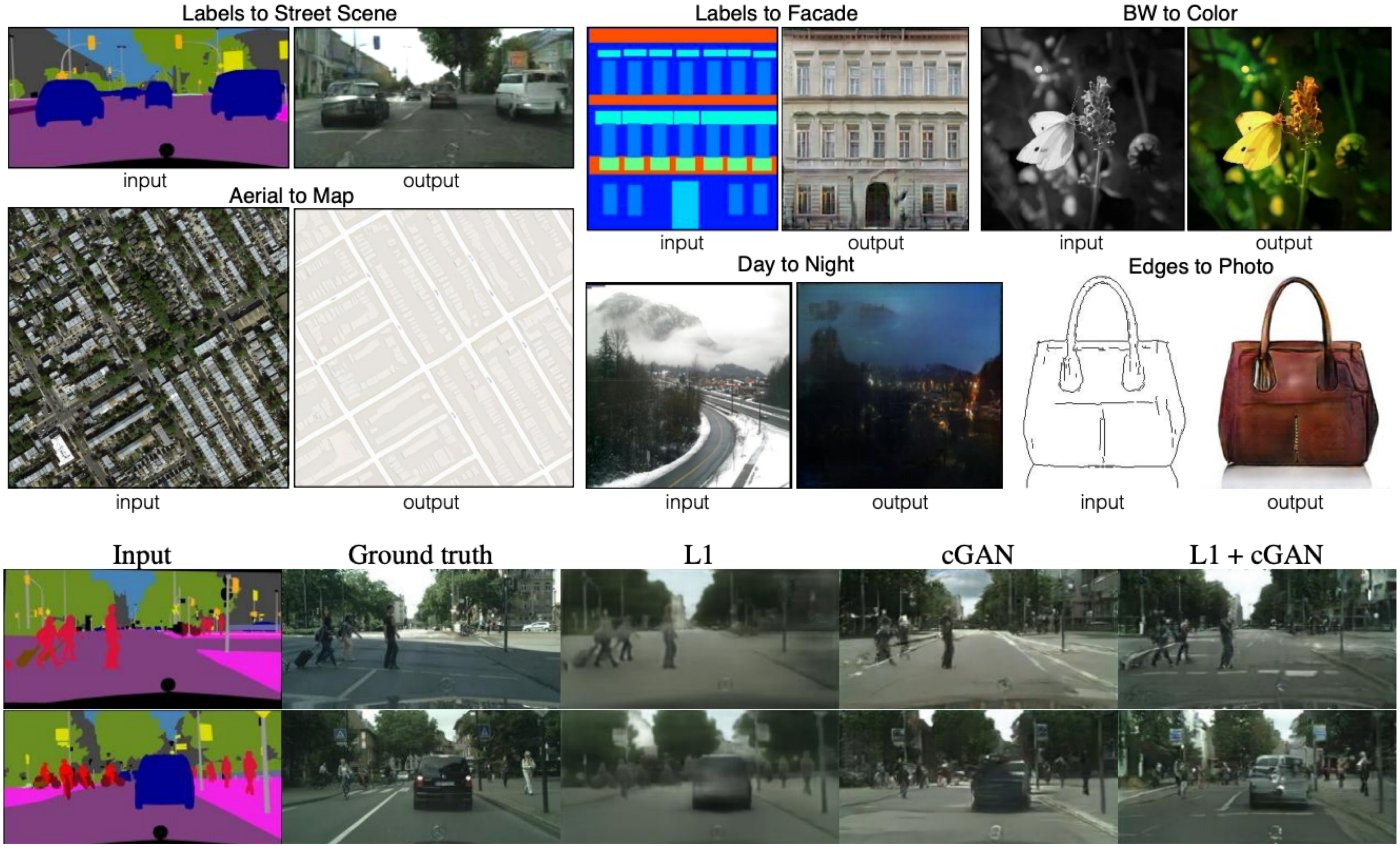
- **Motivation:** Neural Style shows good performance for *artistic styles*, but often fails to generate realistic outputs for more complex domain transfer
  - **Idea:** Use GAN (which known to produce realistic images) for style loss
  - **Goal:** Given source domain X_S and target domain X_T, learn a mapping $f: X_S \rightarrow X_T$
    - In prior terminology, generate x_t' with content of $x_s \in X_S$ and style of X_T

  - **Pros & Cons** (between GAN-based methods and Neural Style)**:**

    - (+) Generates **realistic** images (neural style is mostly for *artistic* styles)
    - (+) Does not rely on a **pretrained network** (can be applied to non-images)
    - (+) Theoretically **sound** (neural style relies on feature statistics heuristic)

    - (-) Need a **dataset** of target style, not a single data
    - (-) Training is **less stable** (alternative optimization)

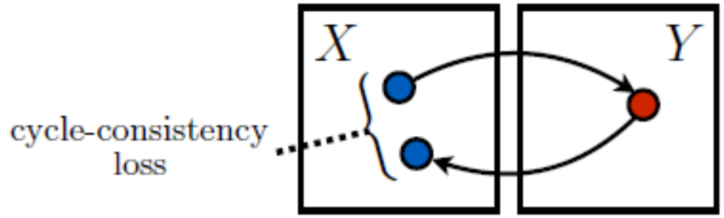
- **Experimental results:** pix2pix can do more complex domain transfer



CycleGAN [Zhu et al., 2017a]

- **Motivation:** pix2pix requires **paired** data of two domains in training (for content = L_1 loss). Can we extend it to unpaired (unsupervised setting)?

  - **Idea:** data translated **from source domain to target domain**, and translated back **to source domain from target domain** should be identical to the original image



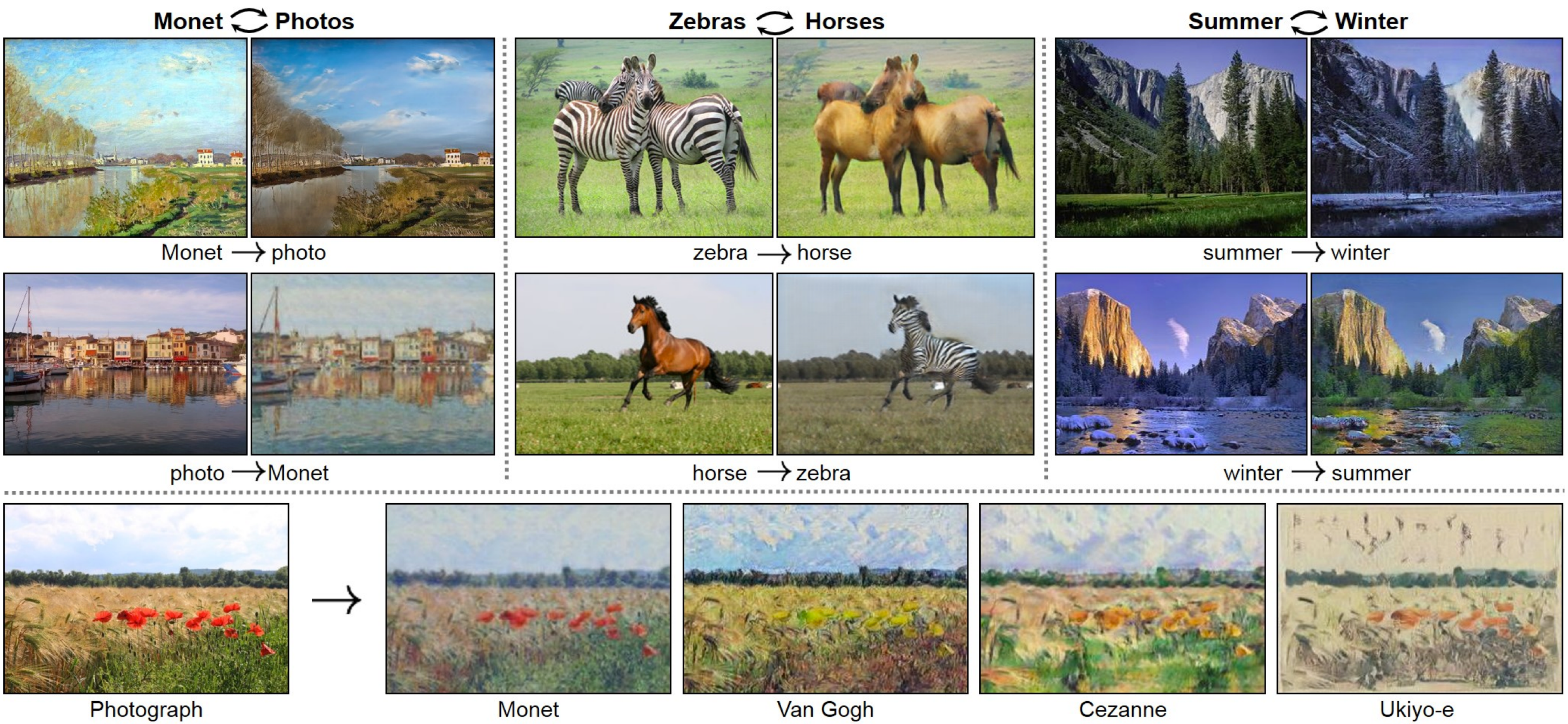
  - **Content loss:**
    - For source→target generator G_{ST} and target→source generator G_{TS}, give cycle-consistency loss, that

$$\mathcal{L}_{\text{content}} = \mathbb{E}_{x_s}[\|x_s - G_{TS}(G_{ST}(x_s))\|_1]$$

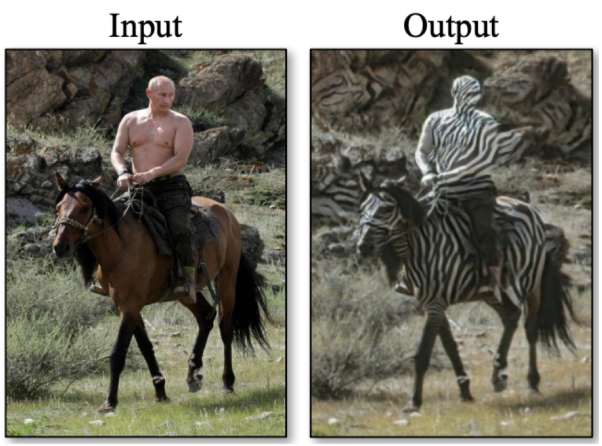
*There are other methods, e.g., isometry constraints [Benaim et al., 2017] or complexity constraints [Galanti et al., 2018] too

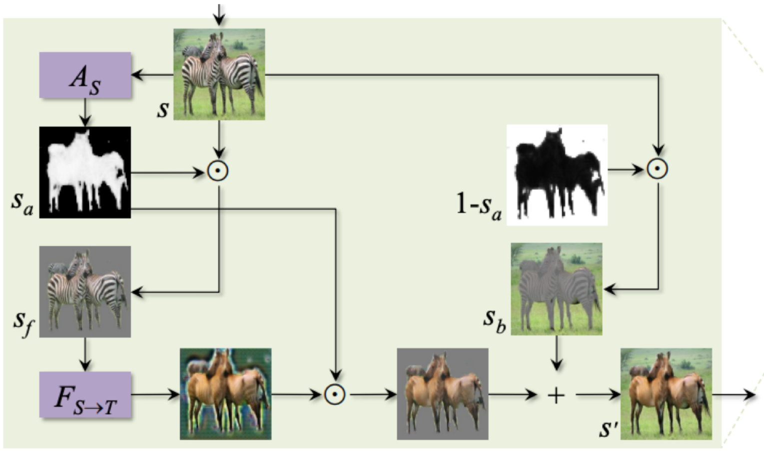
CycleGAN [Zhu et al., 2017a]

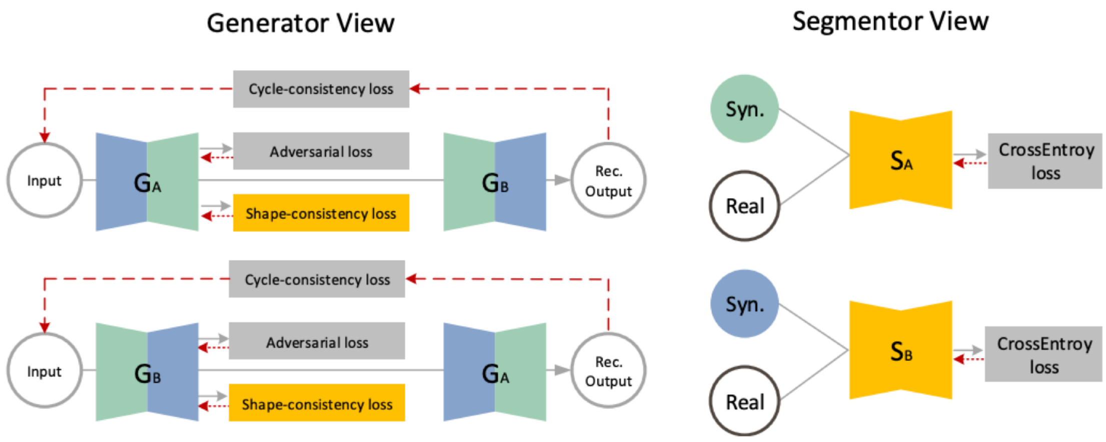
- **Experimental results**



- **Results (Failure cases & Solutions)**
  - CycleGAN suffers from **false positive/negative** problems
  - To relax this issue, one can use segmentation [Liang et al., 2017] or predicted attention [Mejjati et al., 2018] to **hardly mask** instances
  - Or train additional **segmentors** to provide shape-consistency loss [Zhang et al., 2018]





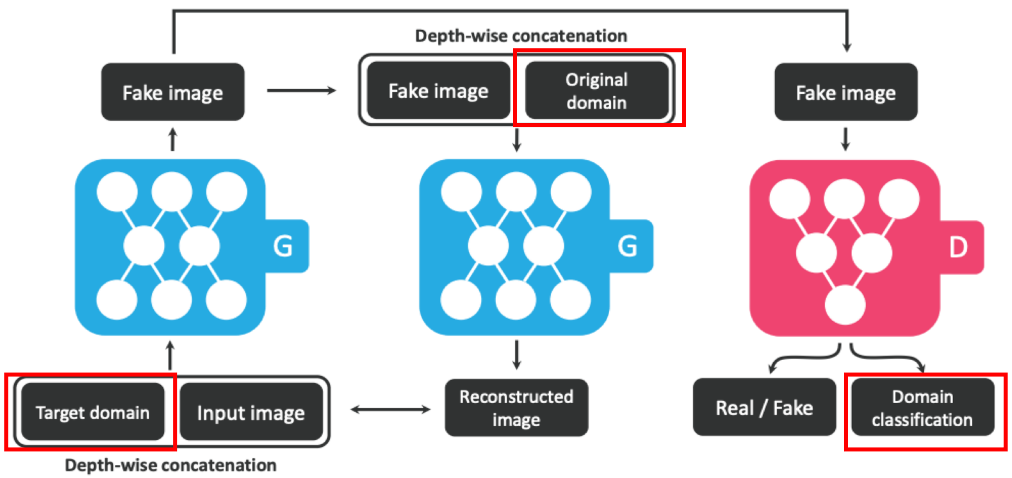


*Source: Mejjati et al. "Unsupervised Attention-guided Image to Image Translation", NIPS 2018
Zhang et al. "Translating and Segmenting Multimodal Medical Volumes with Cycle- and Shape-Consistency Generative Adversarial Network", CVPR 2018

StarGAN [Choi et al., 2018]

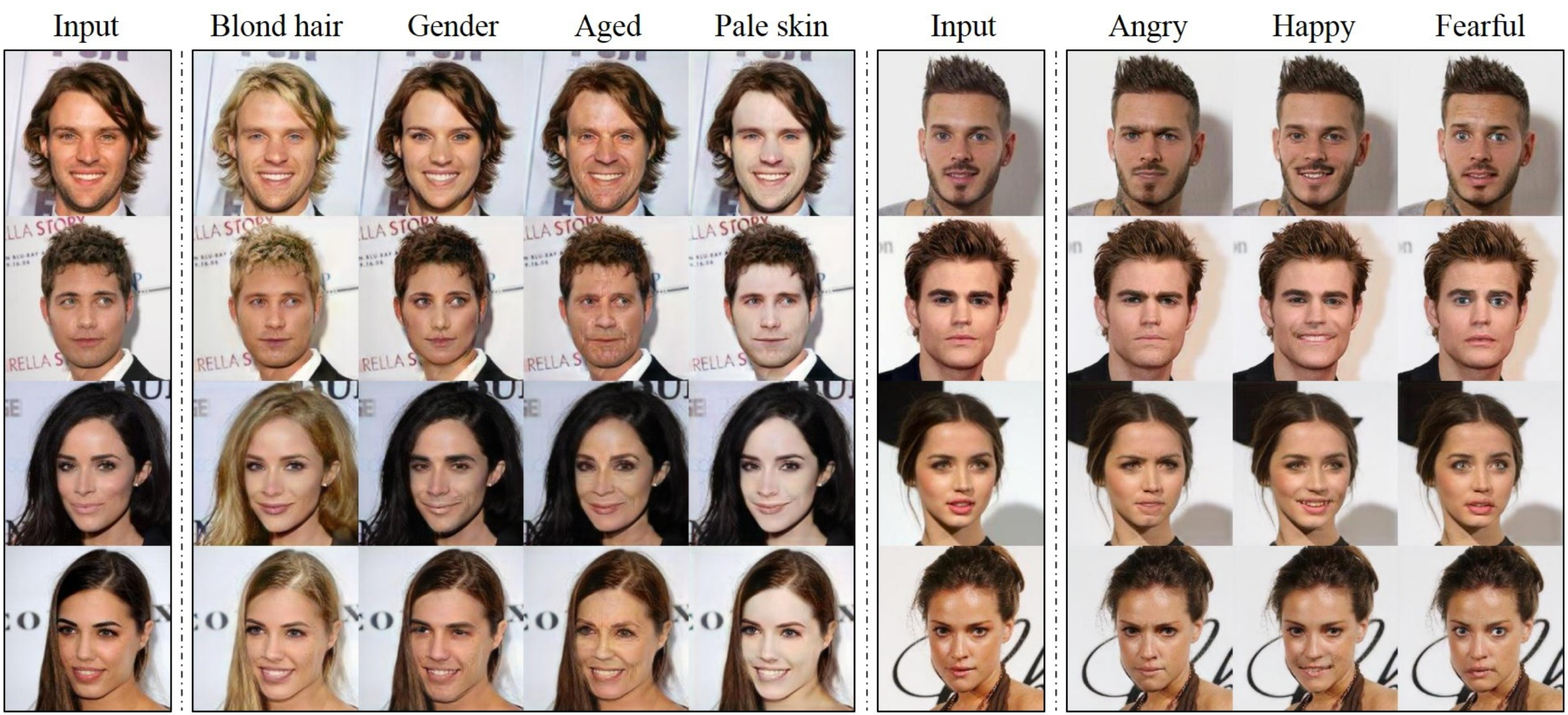
- **Motivation:** Can we extend domain transfer to multi-domain settings?
  - **Idea:** Provide domain conditional vector c (one-hot encoded) as input

  - For **translation**, give **target** domain vector, and for **reconstruction**, give **original** domain vector, hence comes back to the original image
  - Discriminator **classifies domain** in addition to real/fake

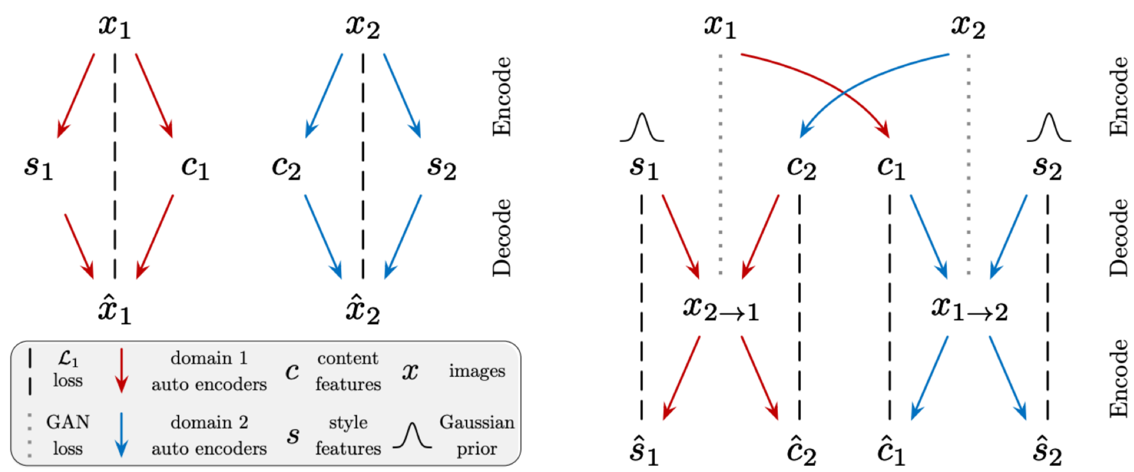


  - This idea also can be applied to multi-modal settings (e.g., BicycleGAN [Zhu et al., 2017b], AugCGAN [Almahairi et al., 2018]) by using random vector z
  - In this case, one should maximize **mutual information** between $G(x_s, z)$ and z to avoid **mode collapse**, i.e., single output with regardless of z

- **Experimental results**

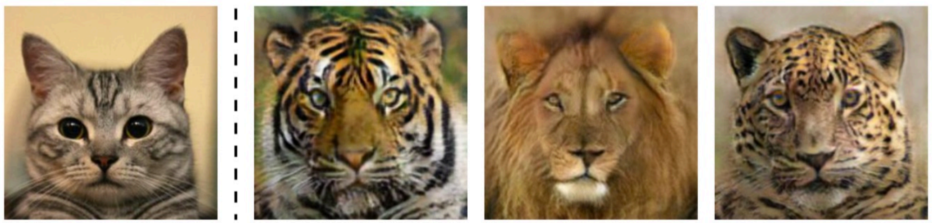


- **Motivation:** Can we do multi-modal domain transfer?
  - **Idea:** Disentangle content & style, and restyle with random style
  - To this end, train a **content encoder** $E_C: x \mapsto c$ and a **style encoder** $E_S: x \mapsto s$
  - Also, train a **decoder** $D: (c, s) \mapsto x$
  - In addition to the original **reconstruction** (= cycle-consistency) loss (Fig a), use **cross-domain reconstruction** loss (Fig b)
  - At inference, one can apply **arbitrary style** (randomly sampled) to the given content

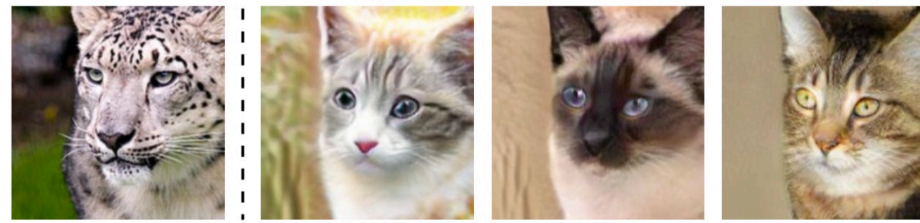


(a) Within-domain reconstruction (b) Cross-domain translation

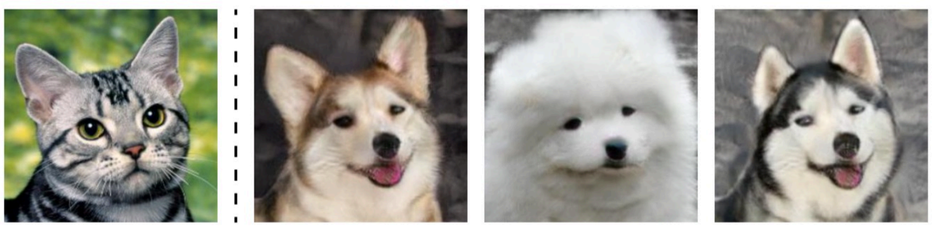
- **Experimental results**



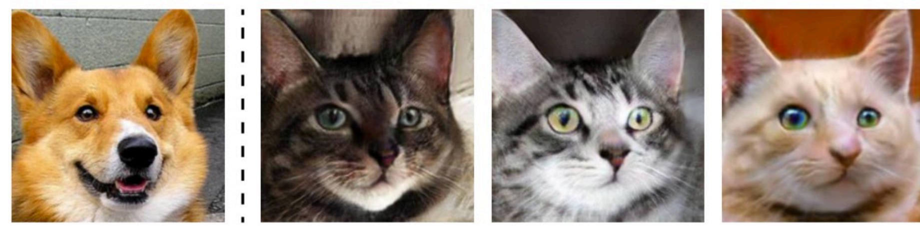
(a) house cats → big cats



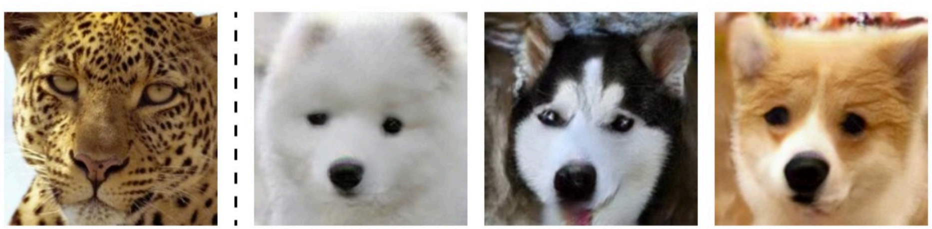
(b) big cats → house cats



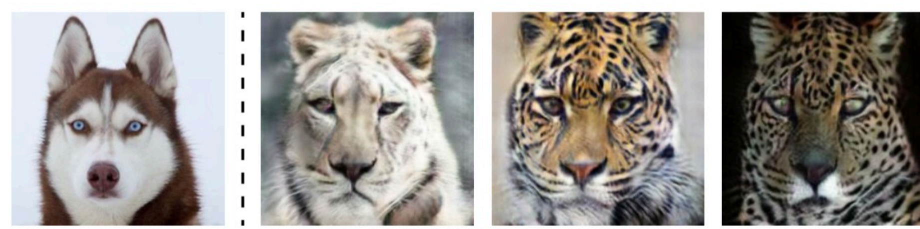
(c) house cats → dogs



(d) dogs → house cats



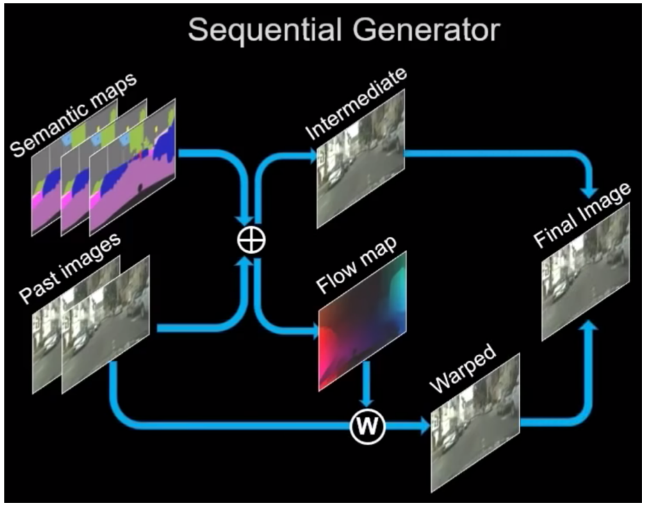
(e) big cats → dogs

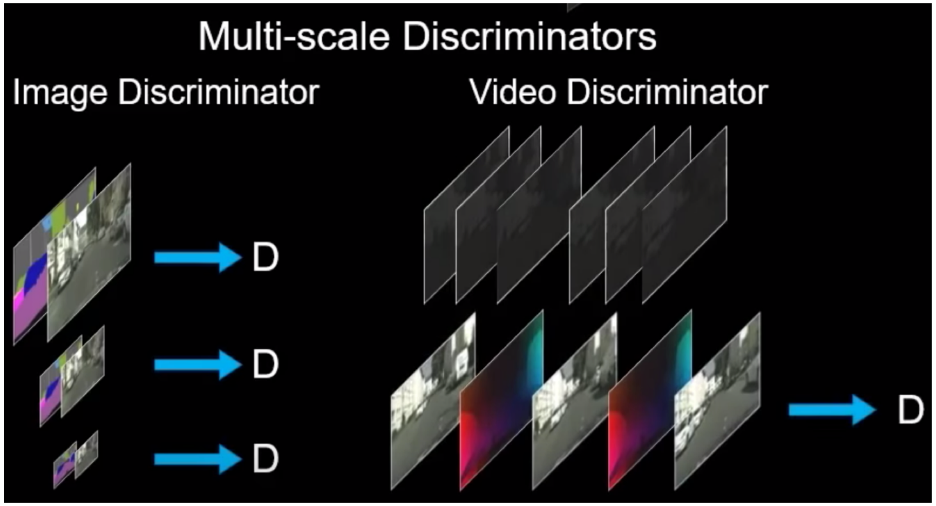


(f) dogs → big cats

vid2vid [Wang et al., 2018]

- **Motivation:** Can we extend domain transfer to video translation?
  - **Issue:** In video generation, one should consider temporal coherence, that successive images should be smoothly varied
  - **Idea:** design an additional recurrent structure in a model
    - Train a **sequential generator** $G: (x_{1:t+1}, y_{1:t}) \mapsto y_{t+1}$
    - In addition to image discriminator D_I, train a **video discriminator** D_V, which compares the ***real* sequences** and ***generated* sequences**
    - **Caveat:** We need paired source/target sequences for this approach

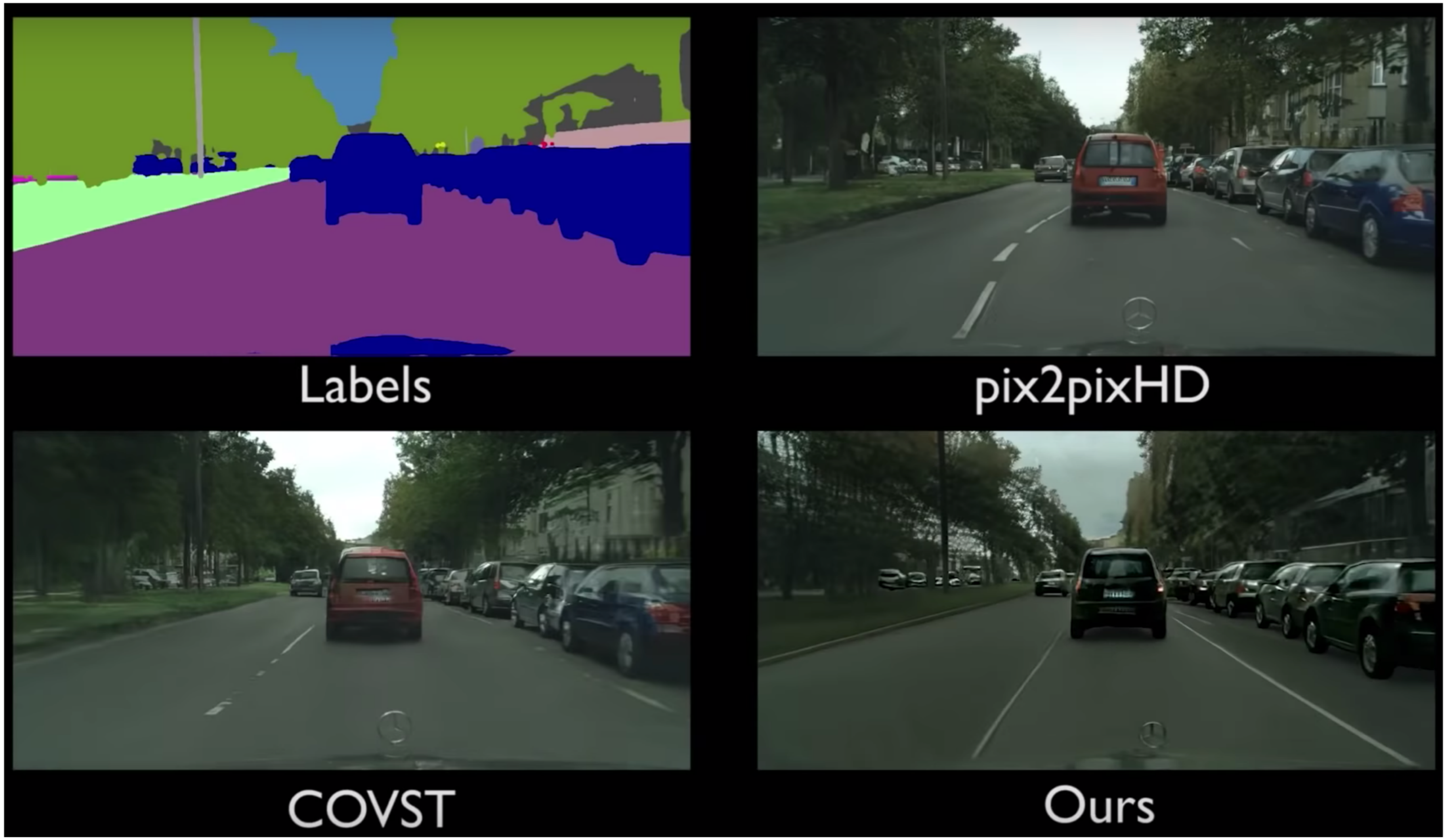




*Source: https://www.youtube.com/watch?v=GrP_aOSXt5U

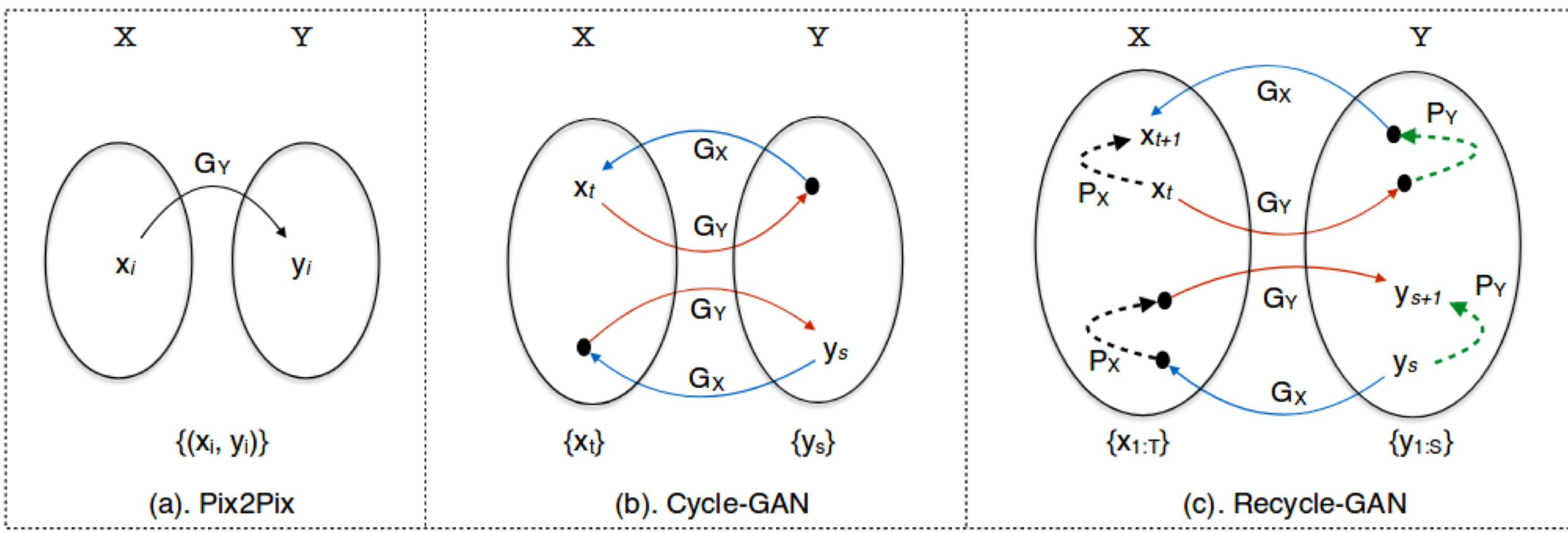
vid2vid [Wang et al., 2018]

- **Results**
  - https://www.youtube.com/watch?v=GrP_aOSXt5U



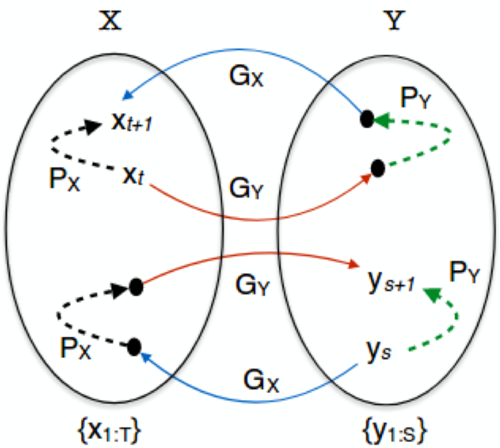
Recycle-GAN [Bansal et al., 2018]

- **Motivation:** Can we extend domain transfer to unpaired video translation?
  - **Problem:** We don't have *real target* sequences
  - **Idea:** Train prediction models $P_X: x_{1:t} \mapsto x_{t+1}, P_Y: y_{1:t} \mapsto y_{t+1}$



Recycle-GAN [Bansal et al., 2018]

- **Motivation:** Can we extend domain transfer to unpaired video translation?
  - **Idea:** Train prediction models $P_X: x_{1:t} \mapsto x_{t+1}$, $P_Y: y_{1:t} \mapsto y_{t+1}$
  - In addition to **GAN** loss and **cycle-consistency** loss, use



(c). Recycle-GAN

- **Recurrent loss** (for training P_X, P_Y):

$$\mathcal{L}_{\text{recurrent}}(P_X) = \sum_t \|x_{t+1} - P_X(x_{1:t})\|^2$$

- **Recycle loss** (for training G_X, G_Y, using P_X, P_Y):

$$\mathcal{L}_{\text{recycle}}(G_X, G_Y, P_X) = \sum_t \|x_{t+1} - G_X(P_Y(G_Y(x_{1:t})))\|^2$$

Recycle-GAN [Bansal et al., 2018]

- **Results**
  - https://www.youtube.com/watch?v=UXjWWy6iTVo

Table of Contents

General Approaches for Domain Adaptation

- **Domain adaptation** aims to learn $f: X_T \rightarrow Y_T$ only using (X_S, Y_S) and X_T

- There are **two general** approaches:
  - **Source/target feature matching:** Make features of X_S and X_T be similar
  - **Target data augmentation:** Generate target data (X_T', Y_T') using domain transfer

*Source: Ganin et al. "Unsupervised Domain Adaptation by Backpropagation", ICML 2015

General Approaches for Domain Adaptation

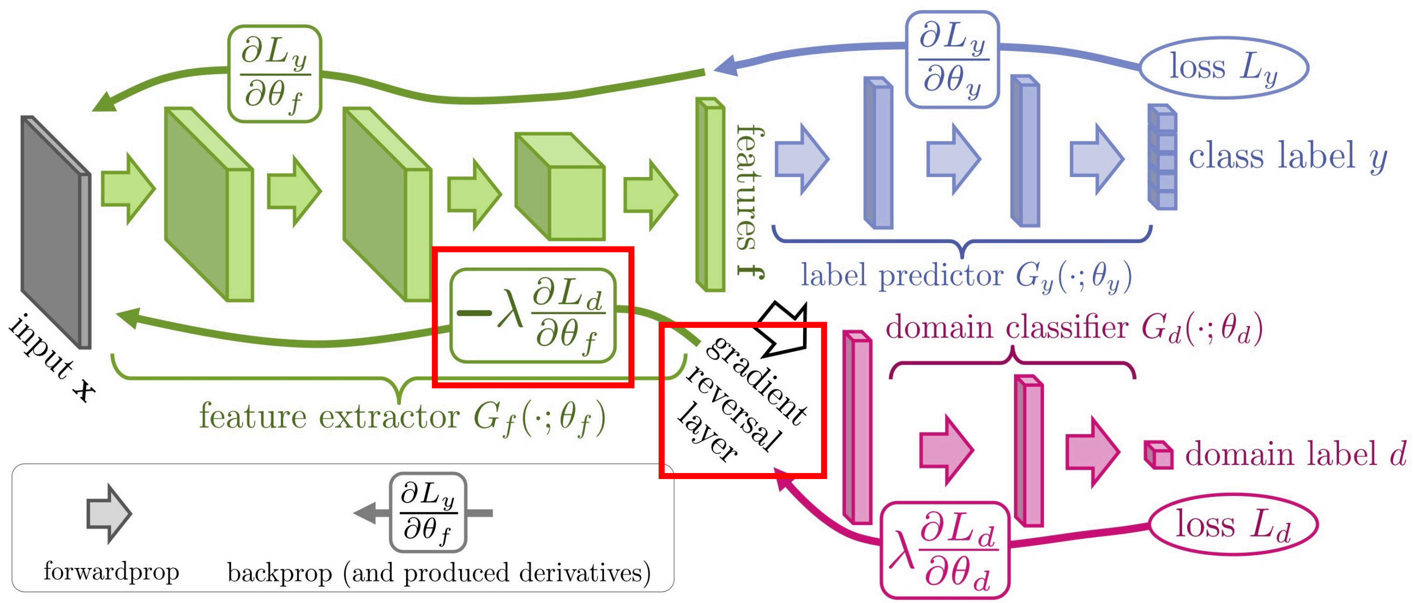
- **Domain adaptation** aims to learn $f: X_T \rightarrow Y_T$ only using (X_S, Y_S) and X_T

- There are **two general** approaches:
  - **Source/target feature matching:** Make features of X_S and X_T be similar
  - **Target data augmentation:** Generate target data (X_T', Y_T') using domain transfer

*Source: Ganin et al. "Unsupervised Domain Adaptation by Backpropagation", ICML 2015

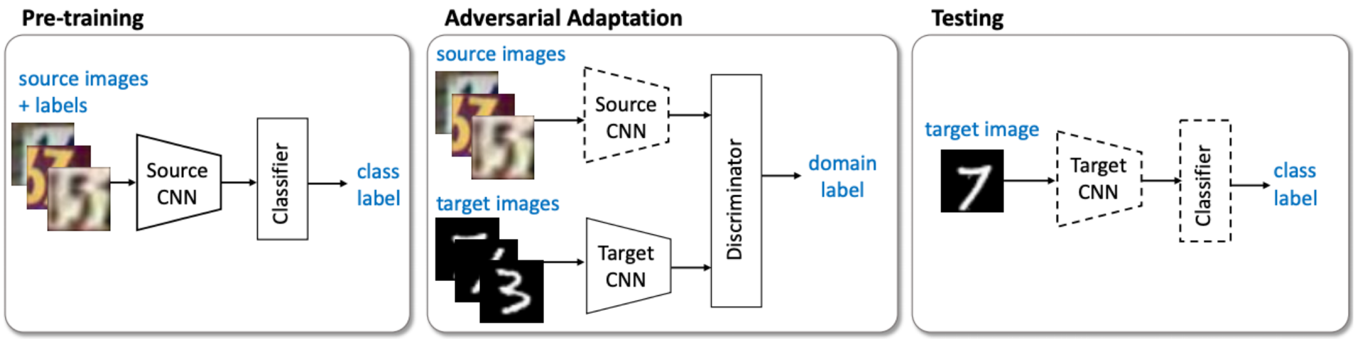
Domain adversarial neural network (DANN) [Ganin et al., 2015]

- **Goal:** Make features of source data X_S and target data X_T be similar
  - **Idea:** Train discriminator D which classifies domain label, and adversarially train network to fool discriminator fail to distinguish source/target feature
  - To this end, gradient from domain classifier is **reversely applied** for the network



Adversarial discriminative domain adaptation (ADDA) [Tzeng et al., 2017]

- **Goal:** Make features of source data X_S and target data X_T be similar
  - Instead, one can alternatively update discriminator, similar to GAN scheme
  - Also, one can train separate feature extractors for source/target domain

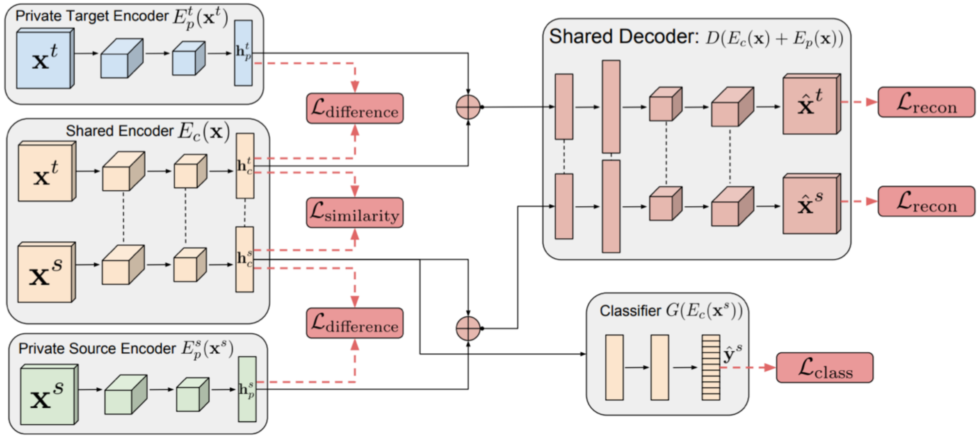


- It is less **stable for train**, but shows **better performance** than gradient reversal

Method	MNIST → USPS	USPS → MNIST	SVHN → MNIST
Source only	0.752 ± 0.016	0.571 ± 0.017	0.601 ± 0.011
Gradient reversal	0.771 ± 0.018	0.730 ± 0.020	0.739 [16]
Domain confusion	0.791 ± 0.005	0.665 ± 0.033	0.681 ± 0.003
CoGAN	0.912 ± 0.008	0.891 ± 0.008	did not converge
ADDA (Ours)	0.894 ± 0.002	0.901 ± 0.008	0.760 ± 0.018

Domain Separation Network (DSN) [Bousmalis et al., 2016]

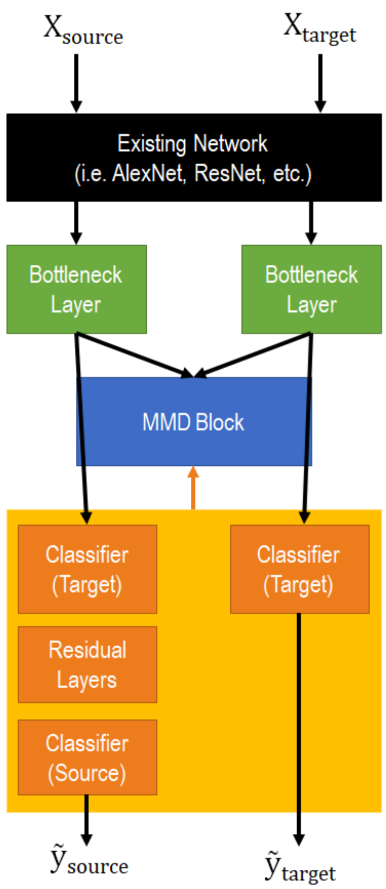
- **Motivation:** Is it rational to exactly match features for source/target data?
  - **Idea:** Consider style of each domain in addition to the **shared content**
  - To this end, train **shared content encoder** E_C and **private style encoders** E_S^S, E_S^T
  - Classifier ignores styles but only use **shared content** as an input



Model	MNIST to MNIST-M	Synth Digits to SVHN	SVHN to MNIST	Synth Signs to GTSRB
Source-only	56.6 (52.2)	86.7 (86.7)	59.2 (54.9)	85.1 (79.0)
CORAL [26]	57.7	85.2	63.1	86.9
MMD [29, 17]	76.9	88.0	71.1	91.1
DANN [8]	77.4 (76.6)	90.3 (91.0)	70.7 (73.8)	92.9 (88.6)
DSN w/ MMD (ours)	80.5	88.5	72.2	92.6
DSN w/ DANN (ours)	**83.2**	**91.2**	**82.7**	**93.1**
Target-only	98.7	92.4	99.5	99.8

Residual Transfer Network (RTN) [Long et al., 2016]

- **Motivation:** Is it rational to exactly match classifiers for source/target data?
  - **Idea:** Define source classifier as a residual function of target classifier



$$f_S(x) = f_T(x) + \Delta f(x)$$

$$\|\Delta f(x)\| \ll |f_T(x)| \approx |f_S(x)|$$

- To ensure that f_T learns structure of target domain, minimize entropy for target data, which is popular method for **semi-supervised** learning [Grandvalet & Bengio, 2004]

- Hence, in addition to (supervised) **classification loss** L and **feature matching loss** $D(X_S, X_T)$ (e.g., GAN loss), use (unsupervised) entropy loss H on target dataset

$$\mathcal{L} = \mathbb{E}_{x_s}[L(f_S(x_s), y_s)] + \gamma \mathbb{E}_{x_t}[H(f_T(x_t))] + \lambda D(X_S, X_T)$$

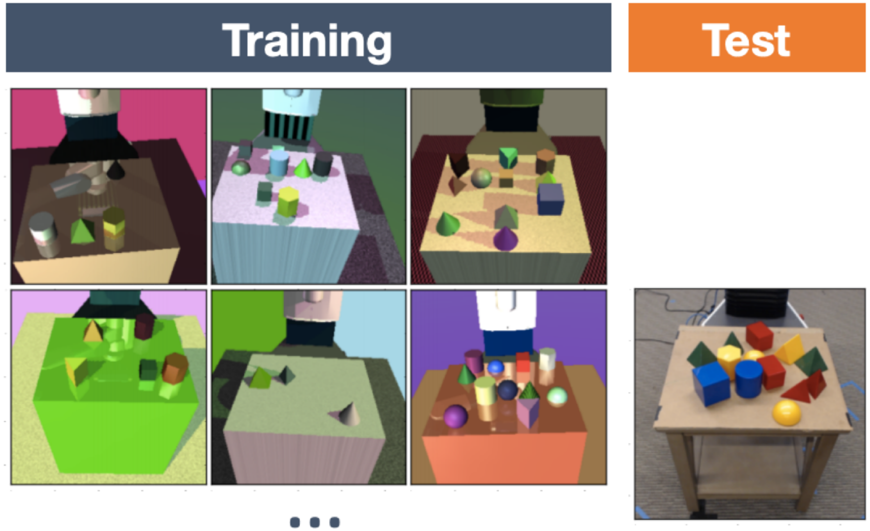
*Source: https://wiki.math.uwaterloo.ca/statwiki/index.php?title=Unsupervised_Domain_Adaptation_with_Residual_Transfer_Networks

Domain Randomization [Tobin et al., 2017]

- **Motivation:** Source/target feature matching can be viewed as disentangling content and style (remove style of each domain but only keep common content)

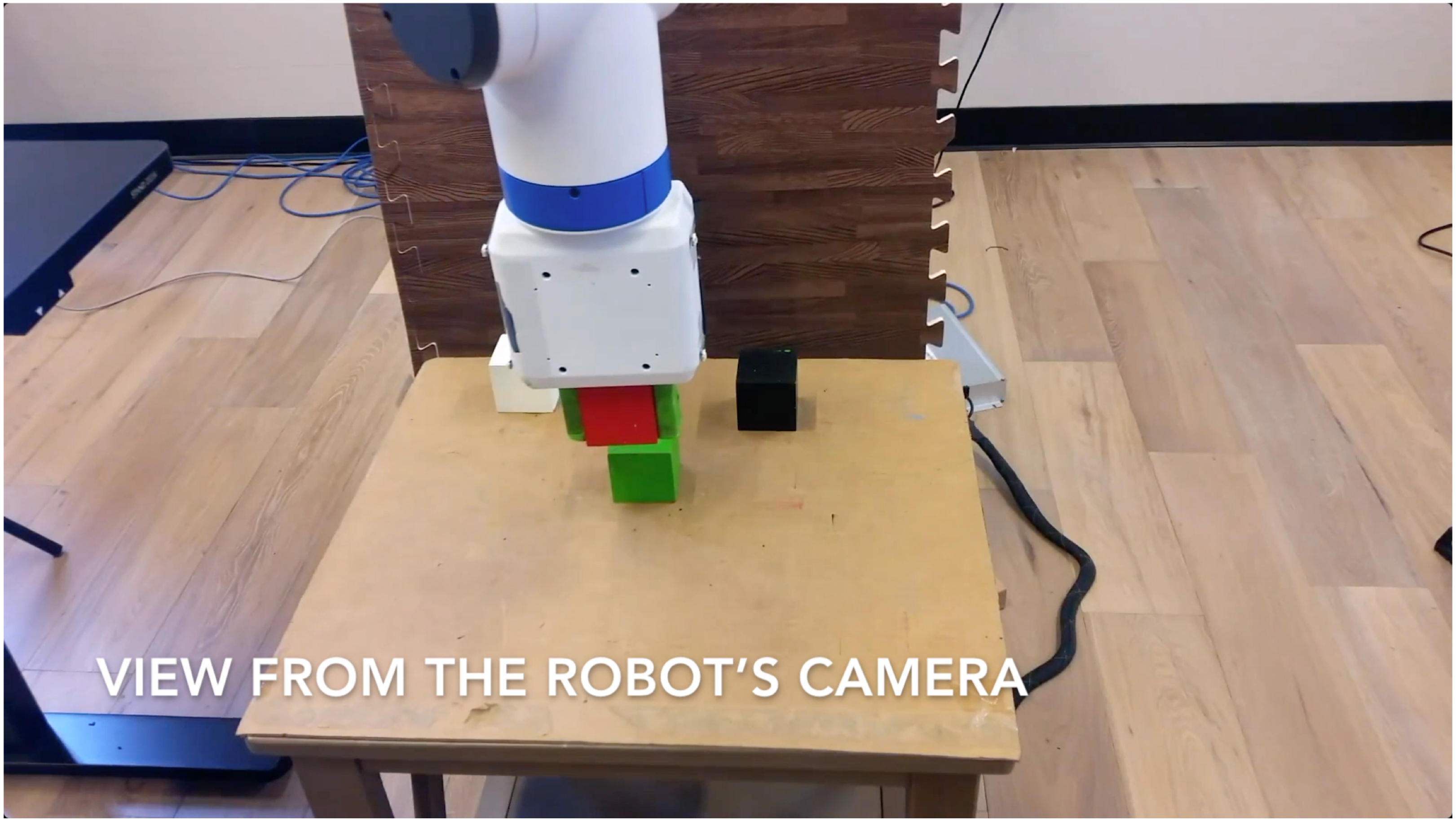
  - **Idea:** In simulation-to-real (sim2real) setting, we can disentangle content by domain augmentation

  - Train NN on simulations with randomly generated styles
    $\Rightarrow$ style sums up, and only content remains



Domain Randomization [Tobin et al., 2017]

- **Results**
  - https://blog.openai.com/generalizing-from-simulation/



General Approaches for Domain Adaptation

- **Domain adaptation** aims to learn $f: X_T \rightarrow Y_T$ only using (X_S, Y_S) and X_T

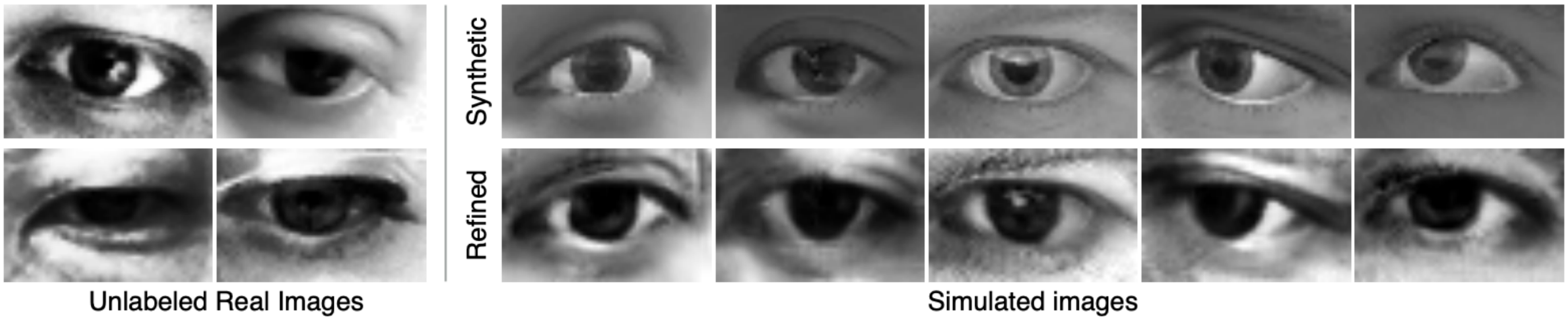
- There are **two general** approaches:
  - **Source/target feature matching:** Make features of X_S and X_T be similar
  - **Target data augmentation**: Generate target data (X'_T, Y'_T) using domain transfer

*Source: Ganin et al. "Unsupervised Domain Adaptation by Backpropagation", ICML 2015

SimGAN [Shrivastava et al., 2017]

- **Idea:** Generate target data with domain transfer model $G: X_S \rightarrow X_T$
  - Given source data (x_s, y_s) and transfer model G, we can generate **labeled target data** $(x_t', y_t') = (G(x_s), y_s)$, and use it to train target network

  - Popular application is **augmenting real images** from **synthetic** images



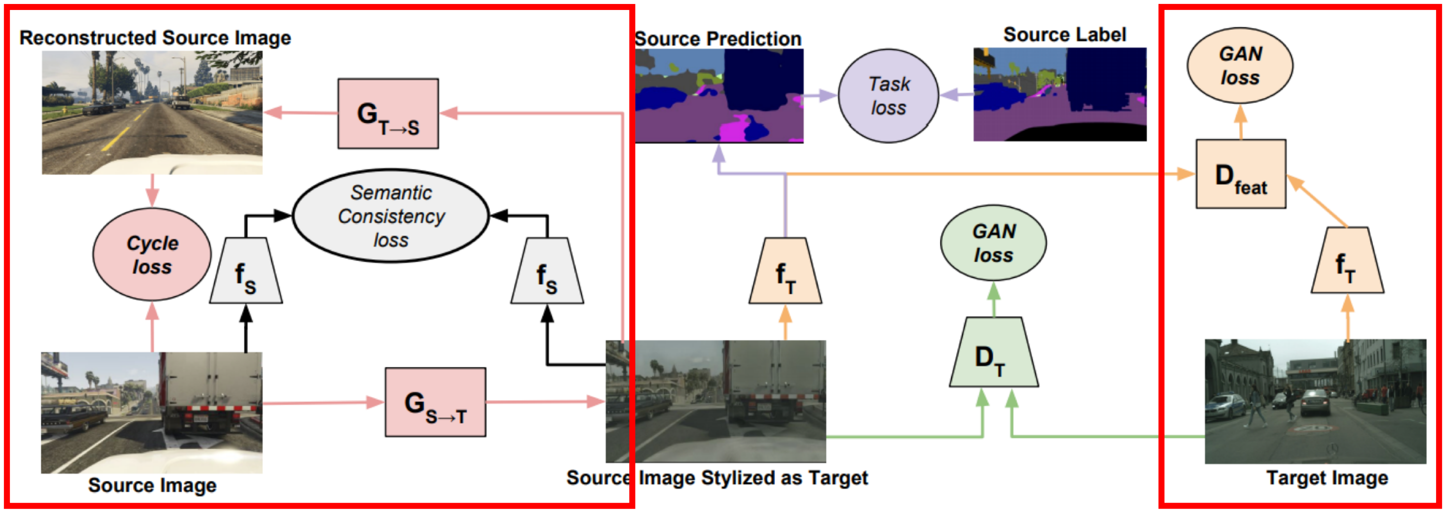
Training data	% of images within d
Synthetic Data	69.7
Refined Synthetic Data	72.4
Real Data	74.5
Synthetic Data 3x	77.7
Refined Synthetic Data 3x	**83.3**

*Source: Shrivastava et al. "Learning from Simulated and Unsupervised Images through Adversarial Training", CVPR 2017

CyCADA [Hoffman et al., 2018]

- **Motivation:** Bridging gap between two approaches: source/target feature matching and target data augmentation?
    - Combine **ADDA** (feature matching via GAN) and **CycleGAN** (domain transfer)

target data augmentation (CycleGAN)



source/target feature matching

Source image (GTA5)

Adapted source image (**Ours**)

Target image (CityScapes)

Pixel accuracy on target

Source-only:	54.0%
Adapted (**ours**):	**83.6%**

Conclusion

- **Domain transfer** is about **generating data** match with given content and style
  - Hence, we should design two losses: **content** loss and **style** loss

- **Domain adaptation** is about **transferring knowledge** for different domains
  - To match source/target features, we apply **adversarial** or **randomization** schemes
  - We can also apply **domain transfer** algorithms to generate target data

- **The research is still ongoing**
  - Dozens of papers exist.
  - Lots of variants not covered in this slide
  - There would be many interesting research directions

References

Introduction

- [Zhang et al., 2016] Colorful Image Colorization. ECCV 2016.
link : https://arxiv.org/abs/1603.08511
- [Ledig et al., 2017] Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial… CVPR 2017.
link : https://arxiv.org/abs/1609.04802
- [Yeh et al., 2017] Semantic Image Inpainting with Deep Generative Models. CVPR 2017.
link : https://arxiv.org/abs/1607.07539
- [Artetxe et al., 2018] Unsupervised Neural Machine Translation. ICLR 2018.
link : https://arxiv.org/abs/1710.11041
- [Mor et al., 2018] A Universal Music Translation Network. arXiv 2018.
link : https://arxiv.org/abs/1805.07848

Network Architecture

- [Ronneberger et al., 2015] U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI 2015.
link : https://arxiv.org/abs/1505.04597
- [He et al., 2016] Deep Residual Learning for Image Recognition. CVPR 2016.
link : https://arxiv.org/abs/1512.03385

References

Domain Transfer

- [Gatys et al., 2016] Image Style Transfer Using Convolutional Neural Networks. CVPR 2016.
  link : https://ieeexplore.ieee.org/document/7780634/
- [Johnson et al., 2016] Perceptual Losses for Real-Time Style Transfer and Super-Resolution. ECCV 2016.
  link : https://arxiv.org/abs/1603.08155
- [Li et al., 2017] Demystifying Neural Style Transfer. ICJAI 2017.
  link : https://arxiv.org/abs/1701.01036
- [Taigman et al., 2017] Unsupervised Cross-Domain Image Generation. ICLR 2017.
  link : https://arxiv.org/abs/1611.02200
- [Dumoulin et al., 2017] A Learned Representation For Artistic Style. ICLR 2017.
  link : https://arxiv.org/abs/1610.07629
- [Isola et al., 2017] Image-to-Image Translation with Conditional Adversarial Networks. CVPR 2017.
  link : https://arxiv.org/abs/1611.07004
- [Zhu et al., 2017a] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial... ICCV 2017.
  link : https://arxiv.org/abs/1703.10593
- [Liu et al., 2017] Unsupervised Image-to-Image Translation Networks. NIPS 2017.
  link : https://arxiv.org/abs/1703.00848
- [Benaim et al., 2017] One-Sided Unsupervised Domain Mapping. NIPS 2017.
  link : https://arxiv.org/abs/1706.00826
- [Galanti et al., 2018] The Role of Minimal Complexity Functions in Unsupervised Learning of... ICLR 2018.
  link : https://arxiv.org/abs/1709.00074

References

Domain Transfer - Extensions

- [Zhu et al., 2017b] Toward Multimodal Image-to-Image Translation. NIPS 2017.
  link : https://arxiv.org/abs/1711.11586
- [Liang et al., 2017] Generative Semantic Manipulation with Contrasting GAN. arXiv 2017.
  link : https://arxiv.org/abs/1708.00315
- [Choi et al., 2018] StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image... CVPR 2018.
  link : https://arxiv.org/abs/1711.09020
- [Zhang et al., 2018] Translating and Segmenting Multimodal Medical Volumes with Cycle- and... CVPR 2018.
  link : https://arxiv.org/abs/1802.09655
- [Almahairi et al., 2018] Augmented CycleGAN: Learning Many-to-Many Mappings from Unpaired… ICML 2018.
  link : https://arxiv.org/abs/1802.10151
- [Huang et al., 2018] Multimodal Unsupervised Image-to-Image Translation. ECCV 2018.
  link : https://arxiv.org/abs/1804.04732
- [Bansal et al., 2018] Recycle-GAN: Unsupervised Video Retargeting. ECCV 2018.
  link : https://arxiv.org/abs/1808.05174
- [Wang et al., 2018] Video-to-Video Synthesis. NIPS 2018.
  link : https://arxiv.org/abs/1808.06601
- [Mejjati et al., 2018] Unsupervised Attention-guided Image to Image Translation. NIPS 2018.
  link : https://arxiv.org/abs/1806.02311
- [Chan et al., 2018] Everybody Dance Now. arXiv 2018.
  link : https://arxiv.org/abs/1808.07371

References

Instance Normalization

- [Ioffe et al., 2016] Batch Normalization: Accelerating Deep Network Training by Reducing Internal... ICML 2015.
  link : https://arxiv.org/abs/1502.03167
- [Ulyanov et al., 2016] Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv 2016.
  link : https://arxiv.org/abs/1607.08022
- [Dumoulin et al., 2017] A Learned Representation For Artistic Style. ICLR 2017.
  link : https://arxiv.org/abs/1610.07629
- [Huang et al., 2017] Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. ICCV 2017.
  link : https://arxiv.org/abs/1703.06868
- [Wu et al., 2018] Group Normalization. ECCV 2018.
  link : https://arxiv.org/abs/1803.08494
- [Nam et al., 2018] Batch-Instance Normalization for Adaptively Style-Invariant Neural Networks. NIPS 2018.
  link : https://arxiv.org/abs/1805.07925

References

Domain Adaptation

- [Grandvalet & Bengio, 2004] Semi-supervised Learning by Entropy Minimization. NIPS 2004.
  link : https://papers.nips.cc/paper/2740-semi-supervised-learning-by-entropy-minimization
- [Ganin et al., 2015] Unsupervised Domain Adaptation by Backpropagation. ICML 2015.
  link : http://proceedings.mlr.press/v37/ganin15.html
- [Bousmalis et al., 2016] Domain Separation Networks. NIPS 2016.
  link : https://arxiv.org/abs/1608.06019
- [Long et al., 2016] Unsupervised Domain Adaptation with Residual Transfer Networks. NIPS 2016.
  link : https://arxiv.org/abs/1602.04433
- [Tzeng et al., 2017] Adversarial Discriminative Domain Adaptation. CVPR 2017.
  link : https://arxiv.org/abs/1702.05464
- [Shrivastava et al., 2017] Learning from Simulated and Unsupervised Images through Adversarial... CVPR 2017.
  link : https://arxiv.org/abs/1612.07828
- [Bousmalis et al., 2017] Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial... CVPR 2017.
  link : https://arxiv.org/abs/1612.05424
- [Tobin et al., 2017] Domain Randomization for Transferring Deep Neural Networks from Simulation... IROS 2017.
  link : https://arxiv.org/abs/1703.06907
- [Hoffman et al., 2018] CyCADA: Cycle-Consistent Adversarial Domain Adaptation. ICML 2018.
  link : https://arxiv.org/abs/1711.03213