# Neural Style Transfer in Text


Zhiping (Patricia) Xiao

University of California, Los Angeles

February 23, 2021

# Outline

# Overview

Summary:

- Style Transfer in Text: Exploration and Evaluation (AAAI'18)

Papers:

- Style Transfer Through Back-Translation (ACL'18) (code)
- CycleGAN-based Emotion Style Transfer as Data Augmentation for Speech Emotion Recognition (INTERSPEECH'19) (code)
- Cycle-Consistent Adversarial Autoencoders for Unsupervised Text Style Transfer (COLING'20)

# Challenges

The progress in language style transfer is lagged behind other domains (e.g. CV).

Special Challenges in **Text** Style Transfer:

- Lack of parallel data;
  - Solution #1: find more data sets
  - Solution #2: focus on **unpaired** approaches (i.e. don't need samples like "a in style A is b in style B")

# Challenges

The progress in language style transfer is lagged behind other domains (e.g. CV).

Special Challenges in **Text** Style Transfer:

- Lack of parallel data;
  - Solution #1: find more data sets
  - Solution #2: focus on **unpaired** approaches (i.e. don't need samples like "a in style A is b in style B")
- Lack of reliable evaluation metrics.
  - Solution #1: human evaluations
  - Solution #2: design metrics to evaluate some important properties (e.g. style difference & content preservation)

# Selected Related Works

Style Transfer Through Back-Translation (ACL'18)

Why back-translation:

- serves as the *Encoder*, represents the **meaning** of the input sentence;
- weakens the style attributes. [1]

How to do style-transfer:

1. pre-trained (supervised training) style classifier
2. multi-task decoder

[1] Discussed in `https://arxiv.org/abs/1610.05461`

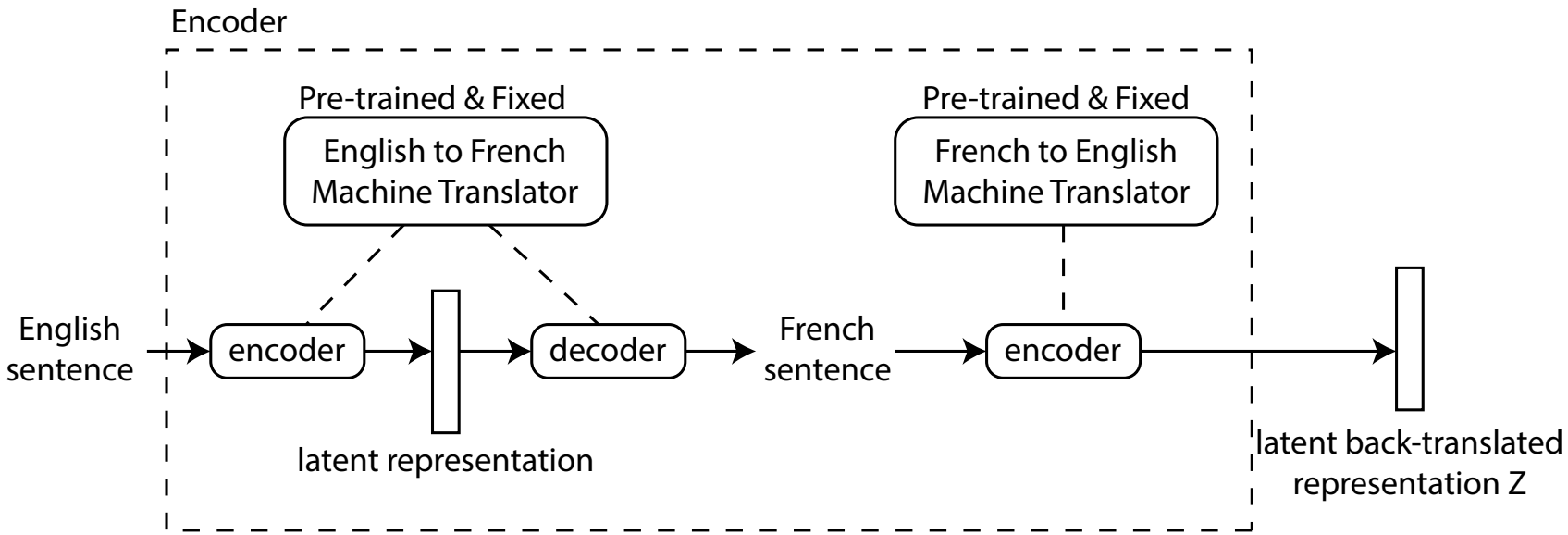


Figure: The Encoder. Machine Translation Models are fixed.

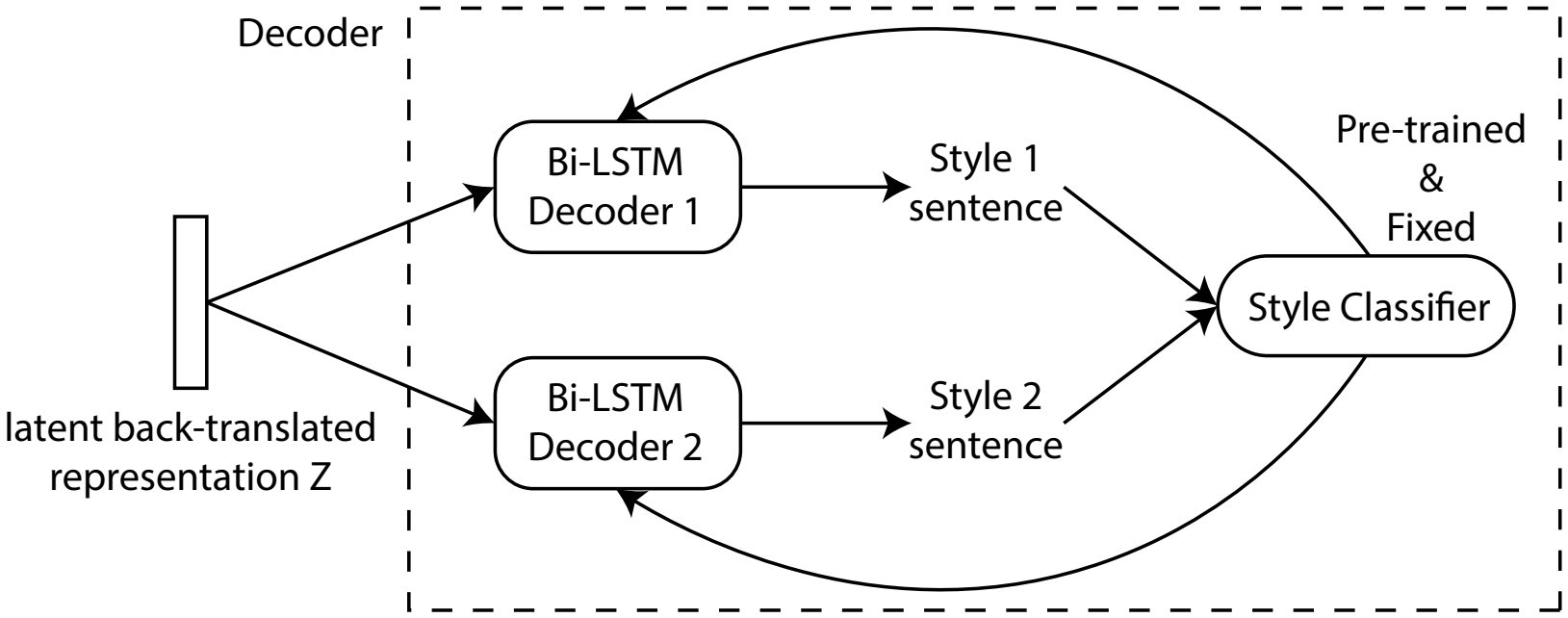


Figure: The Decoder. Could be regarded as a multi-task decoder. The style classifier is a convolutional neural network (CNN) trained in a supervised manner, on held-out training data (never used to train style transfer decoders later on).

How the challenges are solved:

- Lack of parallel data: train against style classifier;
- Lack of reliable evaluation metrics:
  1. Style transfer accuracy: the proportion of our generated sentences of the desired style (according to the pre-trained style classifiers).
  2. Preservation of meaning: conducted human evaluations.
  3. Fluency (the readability and the naturalness): conducted human evaluations.

Components (decoders and classifier) are *somewhat* "adversarial", but are not trained end-to-end.

# Adversarial Components' Necessity

By design, the style-transfer models are trying to generate "just-as-good" fake samples, such that the fake ones are hard to be distinguished from genuine ones.

It is natural to come up with a style-classifier, and a generator to work against it.

# Quick Review: GAN [2]

A framework: estimating generative models via an adversarial process.

- simultaneously train two models:
  - a generative model **G**: captures the data distribution, generates synthetic data that looks like real;
  - a discriminative model **D**: tell the probability that a sample came from the training data rather than **G**.
- training procedure: corresponds to a minimax two-player game
  - for **G**: to maximize the probability of **D** making a mistake;
  - for **D**: to minimize the mistake that **D** makes regarding the current **G**.

[2] https://arxiv.org/abs/1406.2661

A unique solution exists:

- **G**: recovering the training data distribution;
- **D**: equal to $\frac{1}{2}$ everywhere.

in the case where **G** and **D** are both multilayer perceptrons (MLP), the entire system can be trained end-to-end.

# Quick Review: GAN

Objective of vanilla GAN is:

$$\min_G \max_D \mathbb{E}_{x_{real}}\big(\log(D(x_{real}))\big) + \mathbb{E}_{x_{fake}}\big(\log(1 - D(G(x_{fake})))\big)$$

**D** (discriminator, output 1 for real data and 0 for fake data) is trained first and then **G** in each iteration. While training **D** we optimize:

$$\max_D \mathbb{E}_{x_{real}}\big(\log(D(x_{real}))\big) + \mathbb{E}_{x_{fake}}\big(\log(1 - D(G(x_{fake})))\big)$$

which is minimizing:

$$\ell_D = -\mathbb{E}_{x_{real}}\big(\log(D(x_{real}))\big) - \mathbb{E}_{x_{fake}}\big(\log(1 - D(G(x_{fake})))\big)$$

and while training **G** we minimize:

$$\ell_G = \log(1 - D(G(x_{fake})))$$

# Quick Review: GAN

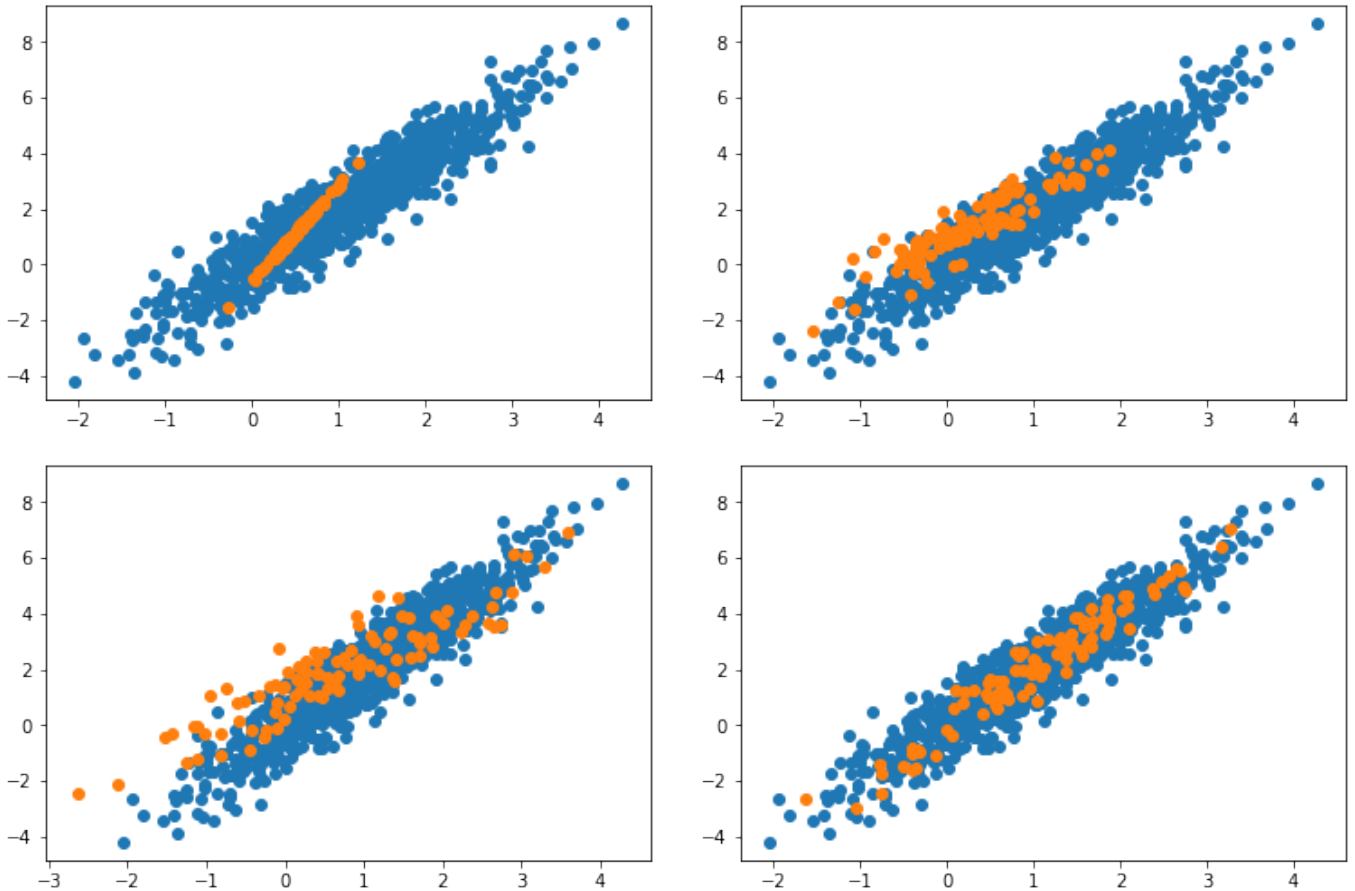

Figure: Example. Noise distribution: normal. [3]

[3] URL of tutorial online (with code).

# Quick Review: GAN

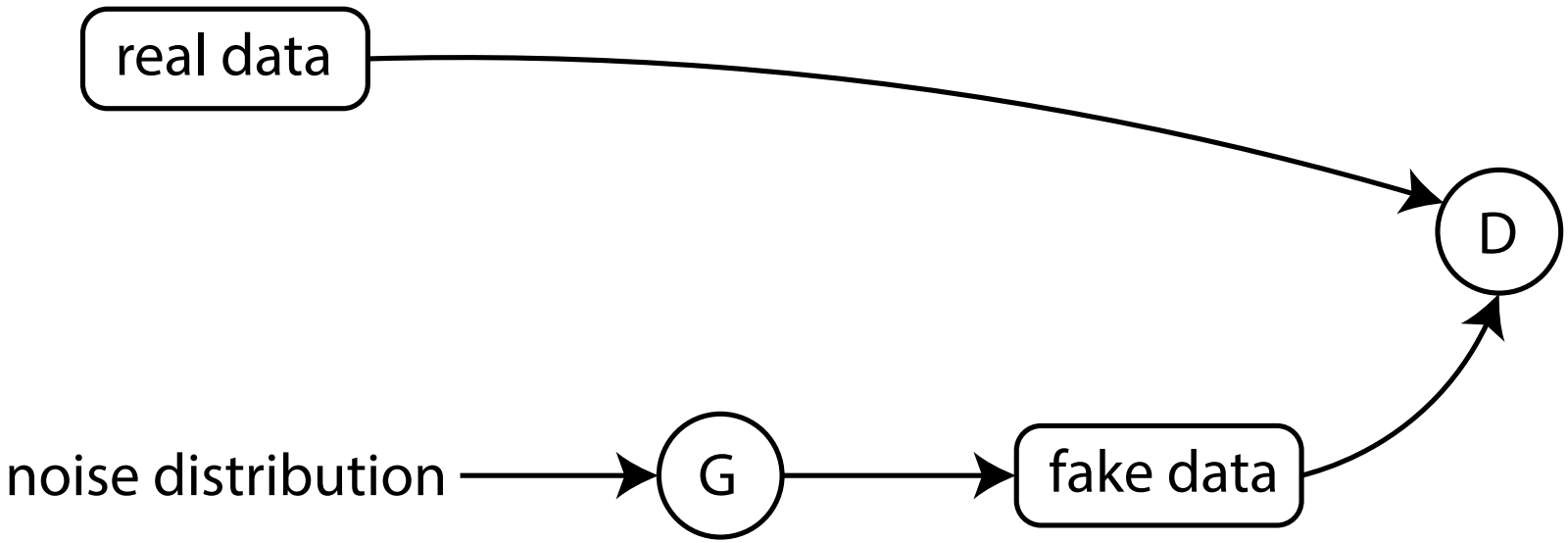


Figure:
$\min_G \max_D \mathbb{E}_{x_{real}} \big( \log(D(x_{real})) \big) + \mathbb{E}_{x_{fake}} \big( \log(1 - D(G(x_{fake}))) \big)$

# Quick Review: Conditional GAN [4]



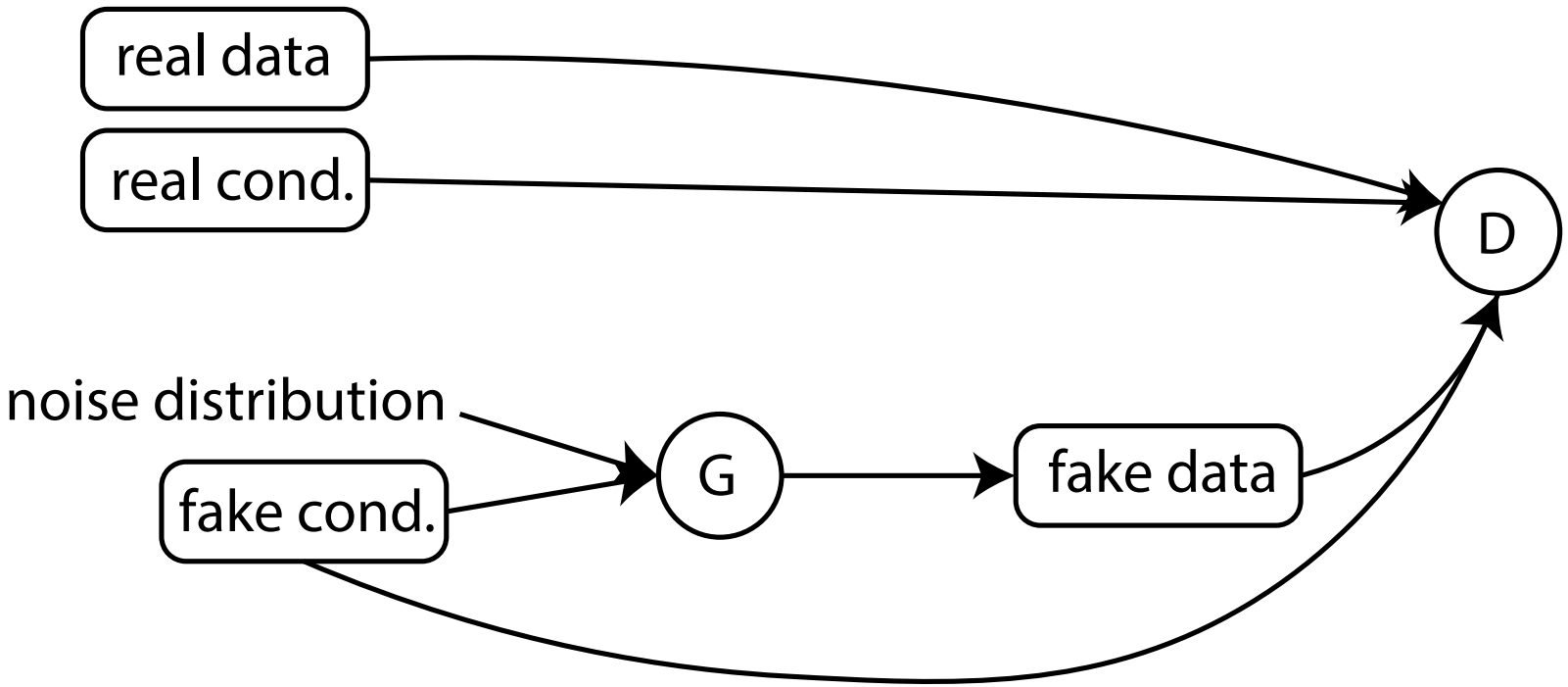


Figure: For a condition $y$,
$\min_G \max_D \mathbb{E}_{x_{real}} \big( \log(D(x_{real}|y)) \big) + \mathbb{E}_{x_{fake}} \big( \log(1 - D(G(x_{fake}|y))) \big)$

[4] https://arxiv.org/abs/1411.1784

# Quick Review: Conditional GAN

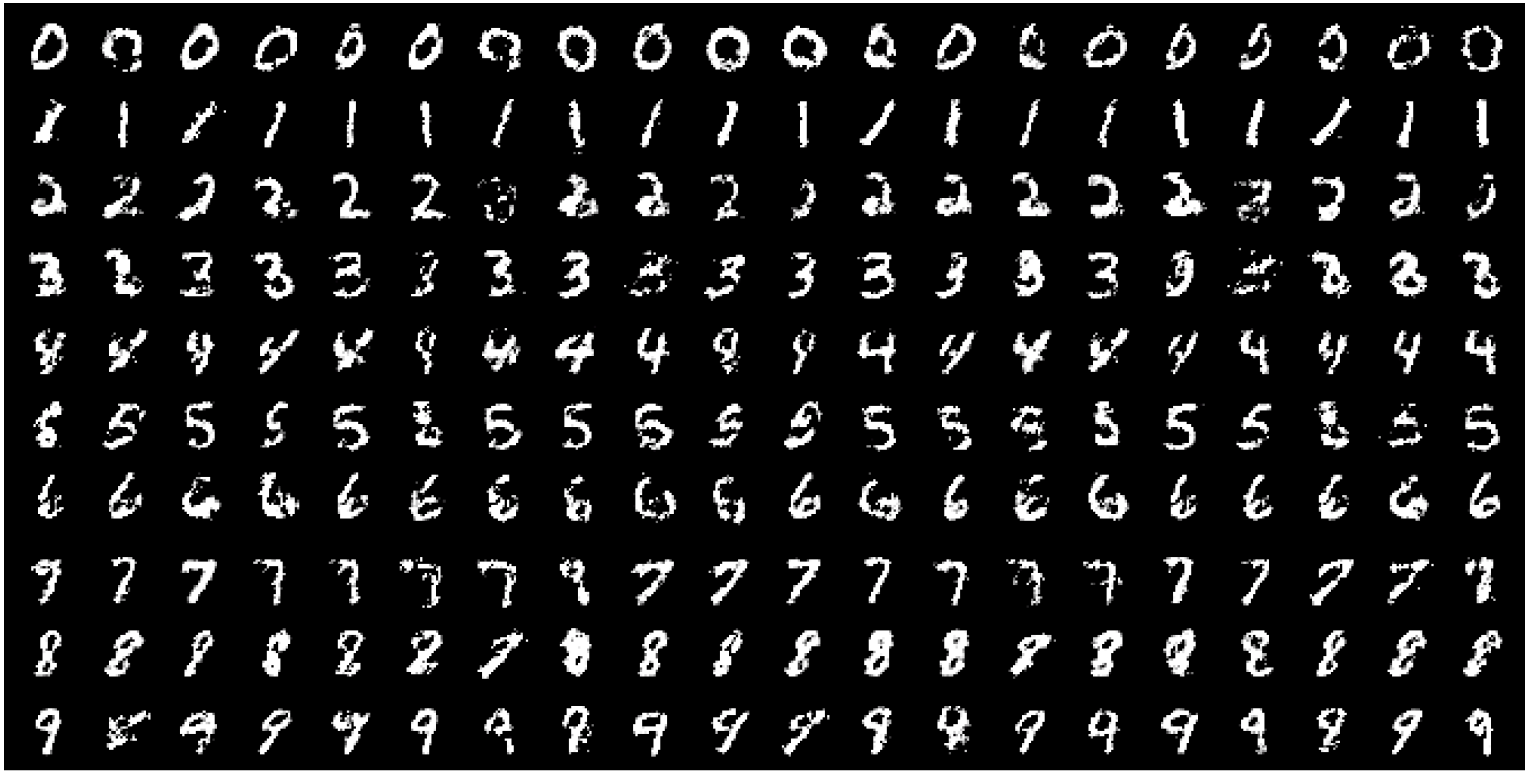

Figure: Generated MNIST digits, each row conditioned on one label.

For more on applications, see pix2pix [5] (condition: outline of shapes).

[5] https://arxiv.org/abs/1611.07004

# Quick Review: Cycle-GAN [6]



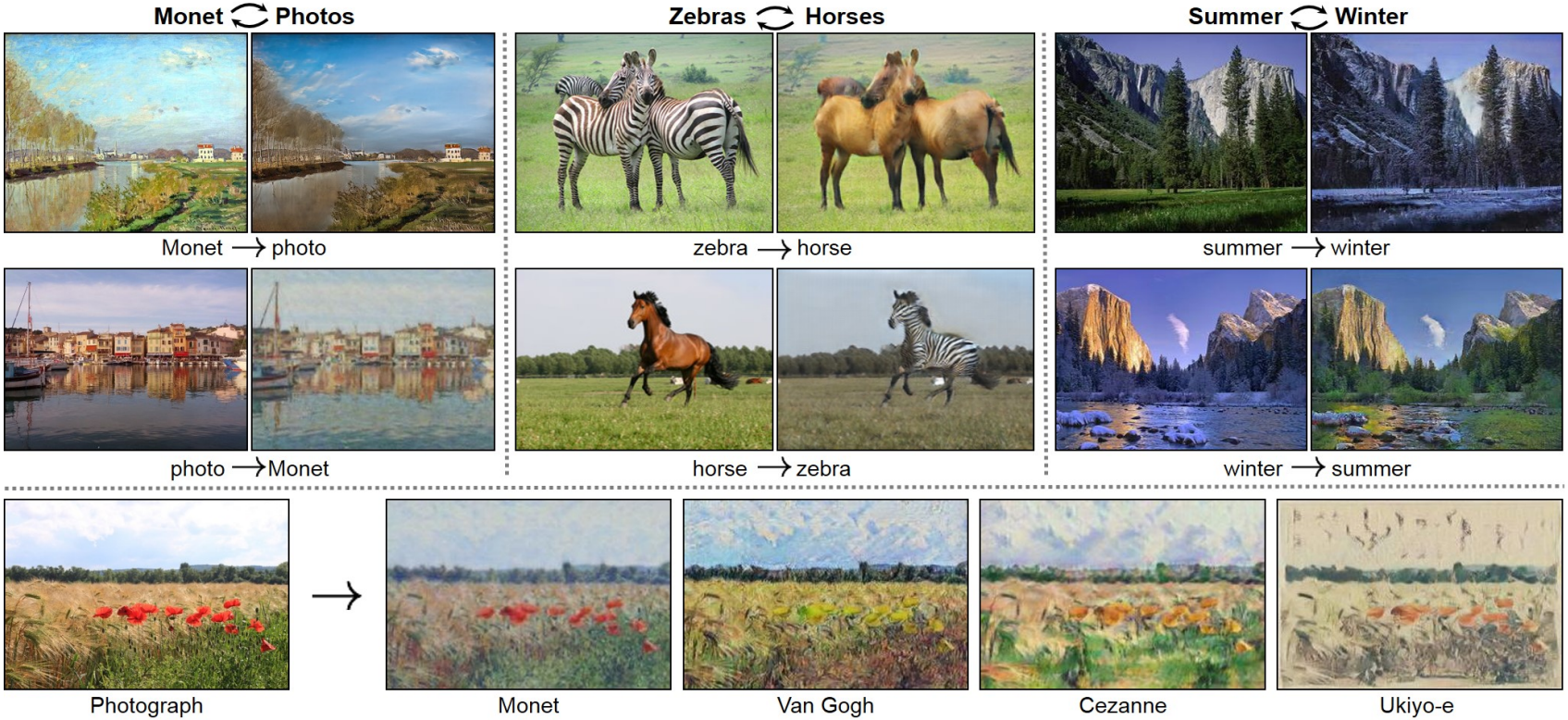


Figure: CycleGAN example. Cycle refers to the bi-directional transfer. There are two generators indeed.

[6] https://junyanz.github.io/CycleGAN/

# Quick Review: CycleGAN

CycleGAN framework solved the "**lack of parallel data**" problem, by not requiring paired samples from different styles.

- Input data: data from domain $X = \{x_i\}_{i=1}^{N}$, and domain $Y = \{y_j\}_{j=1}^{M}$. Two sets of data, two different styles.
- Two generator / translators: $G : X \to Y$ and $F : Y \to X$.
- Associated adversarial discriminators $D_X$ and $D_Y$.
- $G$, $F$, $D_X$, $D_Y$ are trained together, end-to-end.
- To enhance cycle consistency, encourage $F(G(x)) \approx x$ and $G(F(y)) \approx y$.

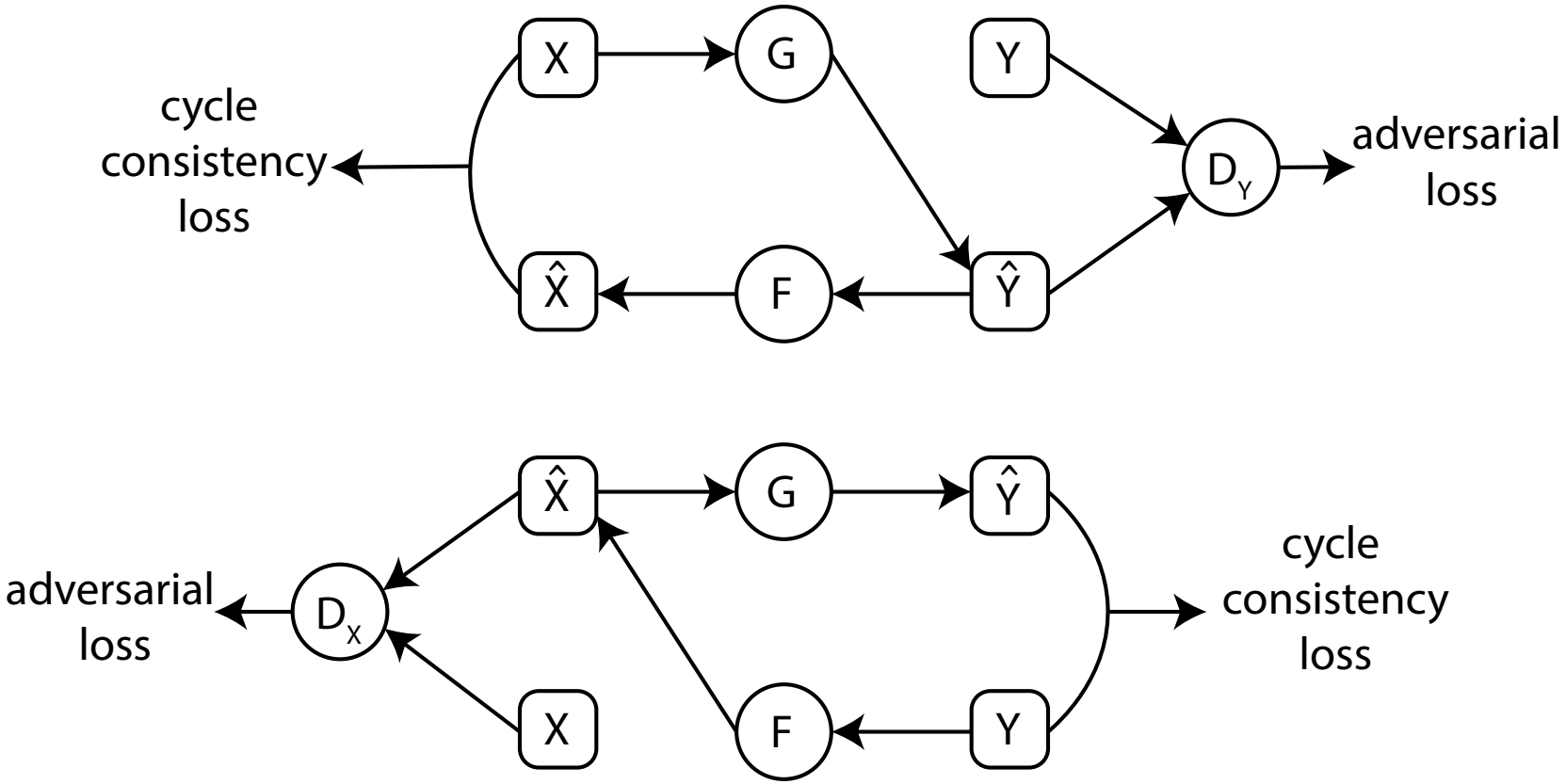


Figure: CycleGAN training pipeline.

# Quick Review: CycleGAN

Adversarial losses:

$$\min_G \max_{D_Y} \mathcal{L}_{GAN}(G, D_Y, X, Y)$$
$$= \min_G \max_{D_Y} \Big( \mathbb{E}_x\big( \log(1 - D_Y(G(x))) \big) + \mathbb{E}_y\big( \log(D_Y(y)) \big) \Big)$$

$$\min_F \max_{D_X} \mathcal{L}_{GAN}(F, D_X, X, Y)$$
$$= \min_F \max_{D_X} \Big( \mathbb{E}_x\big( \log(D_X(x)) \big) + \mathbb{E}_y\big( \log(1 - D_X(F(y))) \big) \Big)$$

Cycle-consistency loss:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_x\big( \|F(G(x)) - x\|_1 \big) + \mathbb{E}_y\big( \|G(F(y)) - y\|_1 \big)$$

Full objective:

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = &\mathcal{L}_{\mathrm{GAN}}(G, D_Y, X, Y) \\ &+ \mathcal{L}_{\mathrm{GAN}}(F, D_X, X, Y) \\ &+ \lambda \mathcal{L}_{cyc}(G, F)\end{aligned}$$

Aim to solve:

$$G^*, F^* = \arg\min_{G,F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y)$$

Full objective:

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = &\mathcal{L}_{\mathrm{GAN}}(G, D_Y, X, Y) \\ &+ \mathcal{L}_{\mathrm{GAN}}(F, D_X, X, Y) \\ &+ \lambda \mathcal{L}_{cyc}(G, F)\end{aligned}$$

Aim to solve:

$$G^*, F^* = \arg\min_{G,F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y)$$

Can be viewed as training two autoencoders jointly:

- $F \circ G : X \to X$
- $G \circ F : Y \to Y$

# CycleGAN-based Data Augmentation

CycleGAN-based Emotion Style Transfer as Data Augmentation for Speech Emotion Recognition (INTERSPEECH'19)

It is not transferring the whole sentence into another sentence of another style (i.e. emotion). Instead, it is transferring the features into another style's features.

- Where is CycleGAN used: generate synthetic feature vectors.
- Why CycleGAN: to obtain a better classifier
  - classifier trained on the combination of real and synthetic feature vectors achieves better classification performance than those rely on the real features.

# CycleGAN-based Data Augmentation

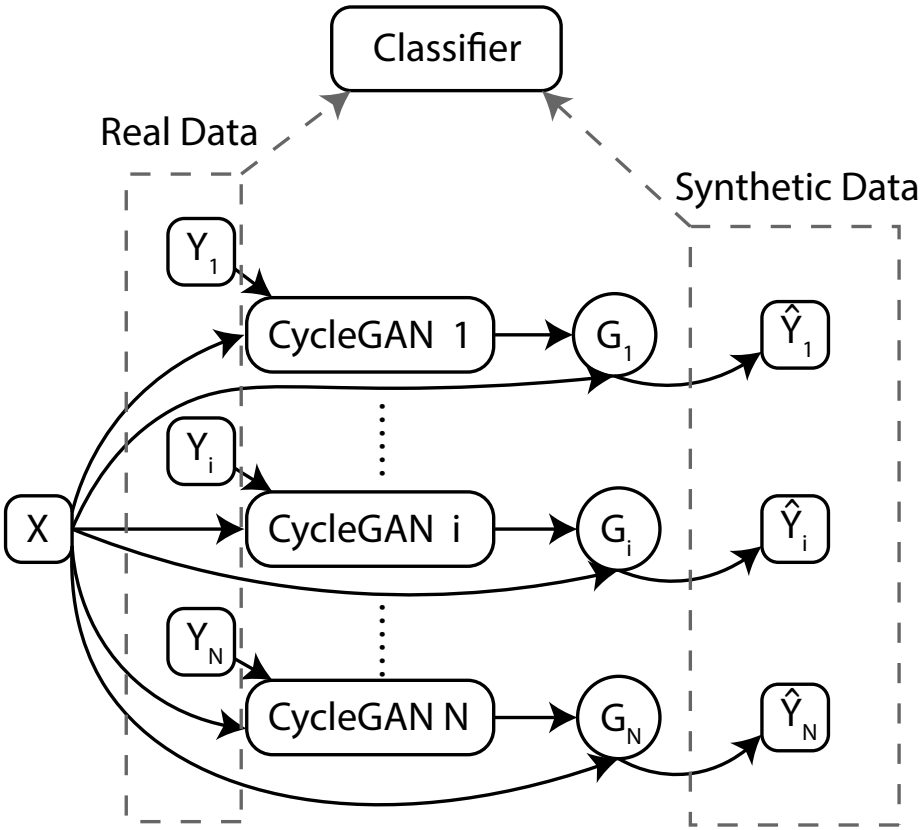


Figure: The data-augmented emotion classifier. $X$ is an external unlabeled dataset, and $Y_i$ represents the features of emotion-$i$ samples in the labeled dataset.

Structure:

- Encoder (sentences $\rightarrow$ features) is not needed: Directly use the "emobase2010" reference feature set, which is based on the Interspeech 2010 Paralinguistic Challenge feature set, consisting of 1,582 features.
- $N$ CycleGANs: each corresponds to an emotion type;
- A domain classifier: classifying $N$ emotion types.

Result: classifier performance gets improved.

# Cycle-Consistent Adversarial Autoencoders (CAE)

Cycle-Consistent Adversarial Autoencoders for Unsupervised Text Style Transfer (COLING'20)

Components:

- Encoder: vanilla LSTM autoencoders (1997 ver.), *one for each style*, sequence $\rightarrow$ feature, or feature $\rightarrow$ sequence;
- Transfer Nets: transforming features from one style to the other;
- Cycle-consistent Constraints: $F(G(x)) \approx x$, $G(F(y)) \approx y$.

# CAE: Encoder & Decoder of Sequences

The encoded representation of style

- $X$: $Z_X = enc_X(X)$;
- $Y$: $Z_Y = enc_Y(Y)$.

where $enc_X$ and $enc_Y$ are the LSTM autoencoders for style $X$ and style $Y$ respectively.

There are corresponding decoders $dec_X$ and $dec_Y$. Objective of the autoencoders is defined as (assuming $X = \{x_i\}_{i=1}^N$ and $Y = \{y_j\}_{j=1}^M$, $x_i$, $y_j$ are sequences):

$$\begin{aligned}
&\mathcal{L}_R(enc_X, dec_X, enc_Y, dec_Y) \\
&= -\frac{1}{N}\sum_{i=1}^{N} \log p(dec_X(enc_X(x_i)) = x_i) \\
&\quad -\frac{1}{M}\sum_{j=1}^{M} \log p(dec_Y(enc_Y(y_j)) = y_j)
\end{aligned}$$

# Recall: CycleGAN

Adversarial losses:

$$\min_G \max_{D_Y} \mathcal{L}_{GAN}(G, D_Y, X, Y)$$

$$= \min_G \max_{D_Y} \Big( \mathbb{E}_x \big( \log(1 - D_Y(G(x))) \big) + \mathbb{E}_y \big( \log(D_Y(y)) \big) \Big)$$

$$\min_F \max_{D_X} \mathcal{L}_{GAN}(F, D_X, X, Y)$$

$$= \min_F \max_{D_X} \Big( \mathbb{E}_x \big( \log(D_X(x)) \big) + \mathbb{E}_y \big( \log(1 - D_X(F(y))) \big) \Big)$$

Cycle-consistency loss:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_x \big( \|F(G(x)) - x\|_1 \big) + \mathbb{E}_y \big( \|G(F(y)) - y\|_1 \big)$$

# CAE: losses

Adversarial losses:

$$\min_{G} \max_{D_Y} \mathcal{L}_{GAN}(G, D_Y)$$

$$= \min_{G} \max_{D_Y} \Big( \mathbb{E}_{z_x} \big( \log(1 - D_Y(G(z_x))) \big) + \mathbb{E}_{z_y} \big( \log(D_Y(z_y)) \big)$$

$$+ \mathbb{E}_{x} \big( \log(1 - D_Y(G(enc_X(x)))) \big) + \mathbb{E}_{y} \big( \log(D_Y(enc_Y(y))) \big) \Big)$$

$$\min_{F} \max_{D_X} \mathcal{L}_{GAN}(F, D_X)$$

$$= \min_{F} \max_{D_X} \Big( \mathbb{E}_{z_x} \big( \log(D_X(z_x)) \big) + \mathbb{E}_{z_y} \big( \log(1 - D_X(F(z_y))) \big)$$

$$+ \mathbb{E}_{x} \big( \log(D_X(enc_X(x))) \big) + \mathbb{E}_{y} \big( \log(1 - D_X(F(enc_Y(y)))) \big)$$

# CAE: losses

Cycle-consistency loss:

$$\begin{aligned}\mathcal{L}_{cyc}(G,F) =& \mathbb{E}_{z_x}\big(\|F(G(z_x)) - z_x\|_1\big) + \mathbb{E}_{z_y}\big(\|G(F(z_y)) - z_y\|_1\big) \\ &+ \mathbb{E}_{x}\big(\|F(G(enc_X(x))) - enc_X(x)\|_1\big) \\ &+ \mathbb{E}_{y}\big(\|G(F(enc_Y(y))) - enc_Y(y)\|_1\big)\end{aligned}$$

The full objective:

$$\begin{aligned}\mathcal{L}_{\mathrm{C}AE} &= \lambda_1 \mathcal{L}_R(enc_X, \mathrm{d}ec_X, enc_Y, \mathrm{d}ec_Y) \\ &+ \lambda_2\big(\mathcal{L}_{\mathrm{G}AN}(G, D_Y) + \mathcal{L}_{\mathrm{G}AN}(F, D_X)\big) \\ &+ \lambda_3 \mathcal{L}_{cyc}(G, F)\end{aligned}$$

Aim to solve ($k = \{X, Y\}$):

$$G^*, F^*, enc_k^*, \mathrm{d}ec_k^* = \arg \min_{G,F,enc_k,\mathrm{d}ec_k} \max_{D_X,D_Y} \mathcal{L}_{\mathrm{C}AE}$$

Now that we have learned:

$$G, F, \mathrm{enc}_X, \mathrm{dec}_X, \mathrm{enc}_Y, \mathrm{dec}_Y$$

we transfer sequence $x_i$ into style $Y$ sequence $\hat{y}_i$ as:

$$\hat{y}_i = \mathrm{dec}_Y(z_{\hat{y}_i}) = \mathrm{dec}_Y(G(z_{x_i})) = \mathrm{dec}_Y\big(G(\mathrm{enc}_X(x_i))\big)$$

Automatic metrics:

- Transfer: style-transfer success rate, a classifier in *fastText* library;
- BLEU: evaluate the content preservation;
- PPL (perplexity): evaluate the fluency of the transferred sequence;
- RPPL (reverse perplexity): evaluate representativeness with respect to the underlying data distribution, detect the mode collapse for generative models, etc.

Human evaluation: let human grade generated sentences with scores from 1 to 5 for style transfer, content preservation and fluency.

# A Glance at Other Possibilities

Reinforcement-learning Based approaches:

- Reinforcement Learning Based Text Style Transfer without Parallel Training Corpus (NAACL-HLT'19) (code)
  - Adversarial training involved.
- A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer (IJCAI'19) (code)

Domain adaptation:

- Domain Adaptive Text Style Transfer (EMNLP'19) (code)
  - Assume that the target domain only has limited non-parallel data; but source domain can be unknown.

# Summary

# Summary

Components of a generator:
(sequence →) encoder (→ feature →) decoder (→ sequence)
The encoder / decoder are not necessarily LSTMs.

- encoder: can be either style-specific or not; can be as simple as RNNs, or as complex as neural machine translation.
- decoder:
  - Type #1: multi-decoder model (one decoder per style)
  - Type #2: style-embedding model (style as parameter)

Adversarial framework is typically used for separating *style* from *content*, and avoid the *lack-of-parallel-data* problem.

Thank you! ☺