CS11-711 Advanced NLP

Adversarial Methods

Graham Neubig

Carnegie Mellon University

Language Technologies Institute

Site

https://phontron.com/class/anlp2021/

With many slides by Zihang Dai & Qizhe Xie

Adversarial Methods

- Basic idea: methods that have a *model* and an *adversary* that work in tandem to learn
- **Generative adversarial networks:** the *model* generates output (e.g. text) and the *adversary* attempts to detect generated text
- **Adversarial feature learning:** the *model* generates features and the *adversary* tries to distinguish features of different types
- **Adversarial robustness:** the *adversary* tries to perturb the input to cause the *model* to fail, and the *model* may be trained to be robust to these perturbations

Generative Adversarial Networks

Generative Models

- Model a data distribution P(X) or a conditional one P(X|Y)
- Latent variable models: introduce another variable Z, and model $P(X) = \sum_{Z} P(X \mid Z) P(Z)$

A "Perfect" Generative Model Can

- Evaluate **likelihood**: P(x)
  - e.g. Perplexity in language modeling
- Generate **samples**: x ~ P(X)
  - e.g. Generate a sentence randomly from P(X) or conditioned on some other information using P(X|Y)
- Infer **latent attributes**: P(Z|X)
  - e.g. Infer the “topic” of a sentence in topic models

No Generative Model is Perfect (so far)

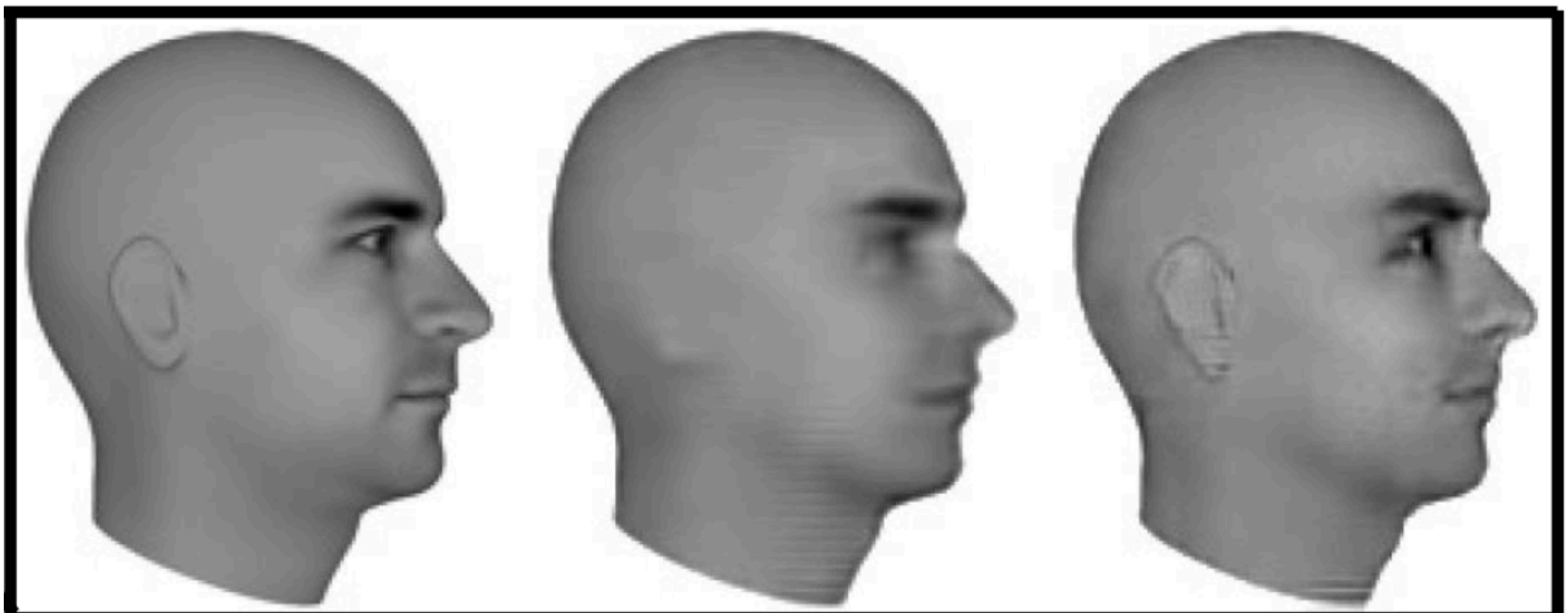
	Non-Latent	VAE	GAN
Likelihood	☆☆☆☆	☆☆	☆
Generation (image)	☆☆☆	☆☆	☆☆☆☆
Inference		☆☆☆	☆☆

- Mostly rely on **MLE** (Lower bound) based training
- **GANs** are particularly good at **generating** continuous **samples**

MLE vs. GAN

- Over-emphasis of common outputs, fuzziness

Real MLE Adversarial



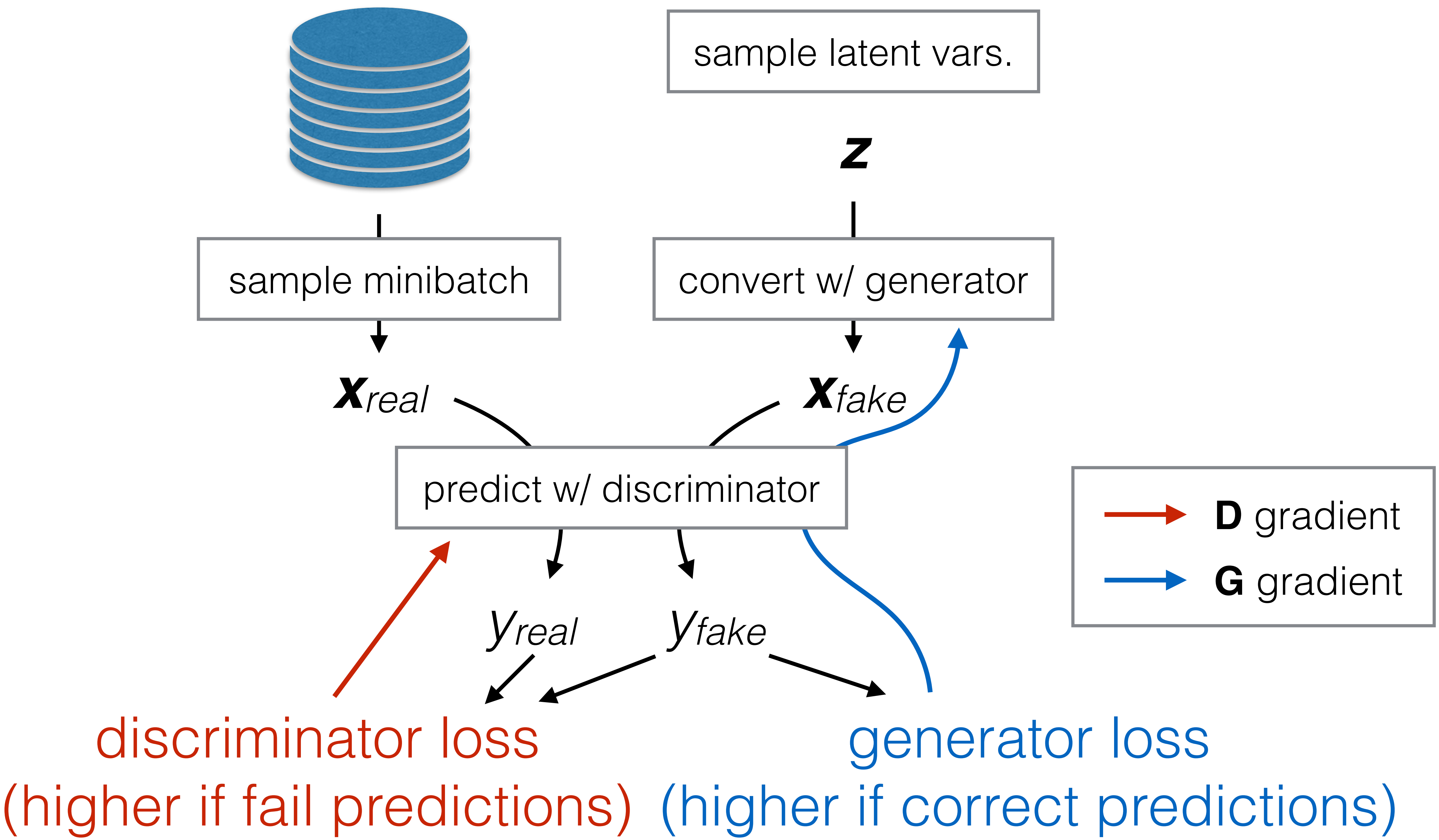
- Note: this is probably a good idea if you are doing maximum likelihood!

Image Credit: Lotter et al. 2015

Basic Paradigm

- Two models: generator and discriminator
  - **Discriminator:** given an image, try to tell whether it is real or not → P(image is real)
  - **Generator:** try to generate an image that fools the discriminator into answering “real”
- Desired result at convergence
  - Generator: generate perfect image
  - Discriminator: cannot tell the difference

Training Method



In Equations

- **Discriminator** loss function:

$$\ell_D(\theta_D, \theta_G) = -\underbrace{\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim P_{data}} \log D(\boldsymbol{x})}_{\text{Predict real for real data}} - \underbrace{\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log(\overbrace{1 - D(G(\boldsymbol{z}))}^{\text{P(fake) = 1 - P(real)}})}_{\text{Predict fake for fake data}}$$

- **Generator** loss function:
  - Make generated data **“less fake”** → Zero sum loss:

$$\ell_G(\theta_D, \theta_G) = -\ell_D(\theta_D, \theta_G)$$

  - Make generated data **“more real”** → Heuristic non-saturating loss:

$$\ell_G(\theta_D, \theta_G) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log D(G(\boldsymbol{z}))$$

  - **Latter** gives **better gradients** when discriminator accurate

Interpretation: Distribution Matching

Process

- [Step1] Z ~ P(Z), P(Z) can be any distribution
- [Step2] X = F(Z), F is a **deterministic** function

Result

- X is a random variable with an implicit distribution P(X), which decided by both P(Z) and F
- The process can produce any complicated distribution P(X) with a reasonable P(Z) and a powerful enough F

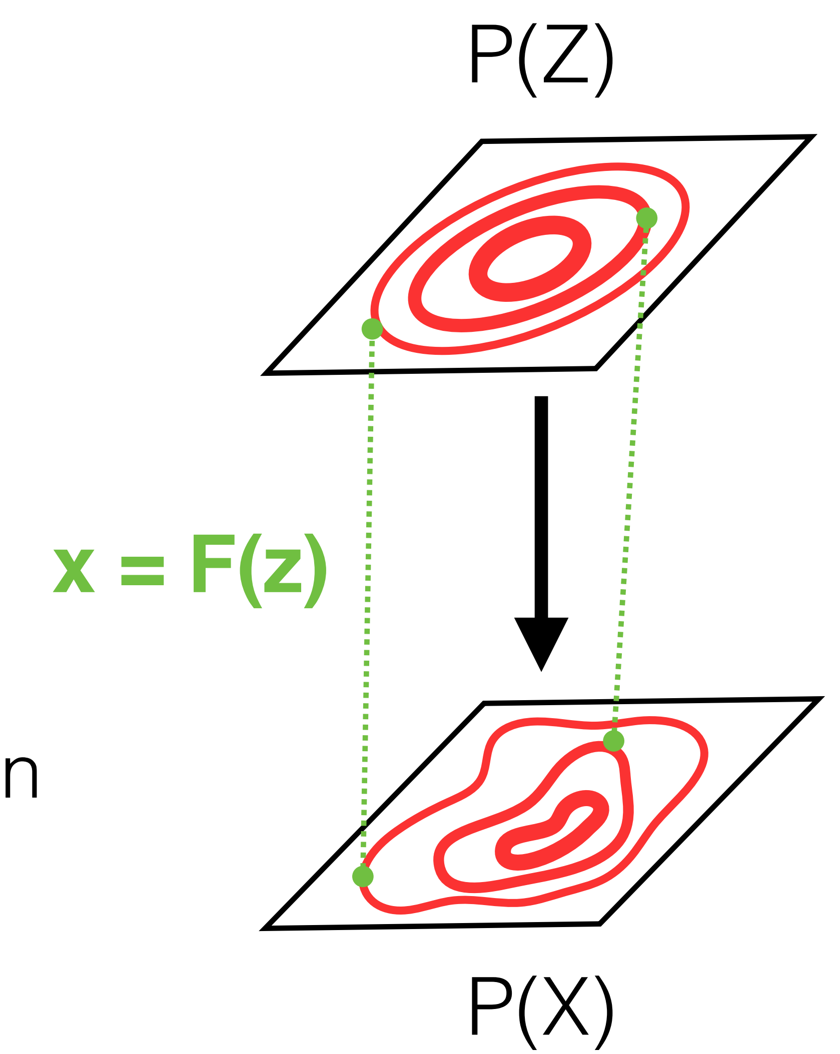


Image Credit: He et al. 2018

In Pseudo-Code

- x_{real} ~ Training data
- $z \sim P(Z)$ → Normal(0, 1) or Uniform(-1, 1)
- $x_{fake} = \mathbf{G}(z)$
- $y_{real} = \mathbf{D}(x_{real})$ → $P(x_{real}$ is real$)$
- $y_{fake} = \mathbf{D}(x_{fake})$ → $P(x_{fake}$ is real$)$
- Train **D**: $\min_D - \log y_{real} - \log(1 - y_{fake})$
- Train **G**: $\min_G - \log y_{fake}$ → non-saturating loss

Why are GANs good?

- Discriminator is a **"learned metric"** parameterized by powerful neural networks
- Can easily pick up any kind of discrepancy, e.g. blurriness, global inconsistency
- Generator has **fine-grained** (gradient) signals to inform it what and how to improve

Problems in GAN Training

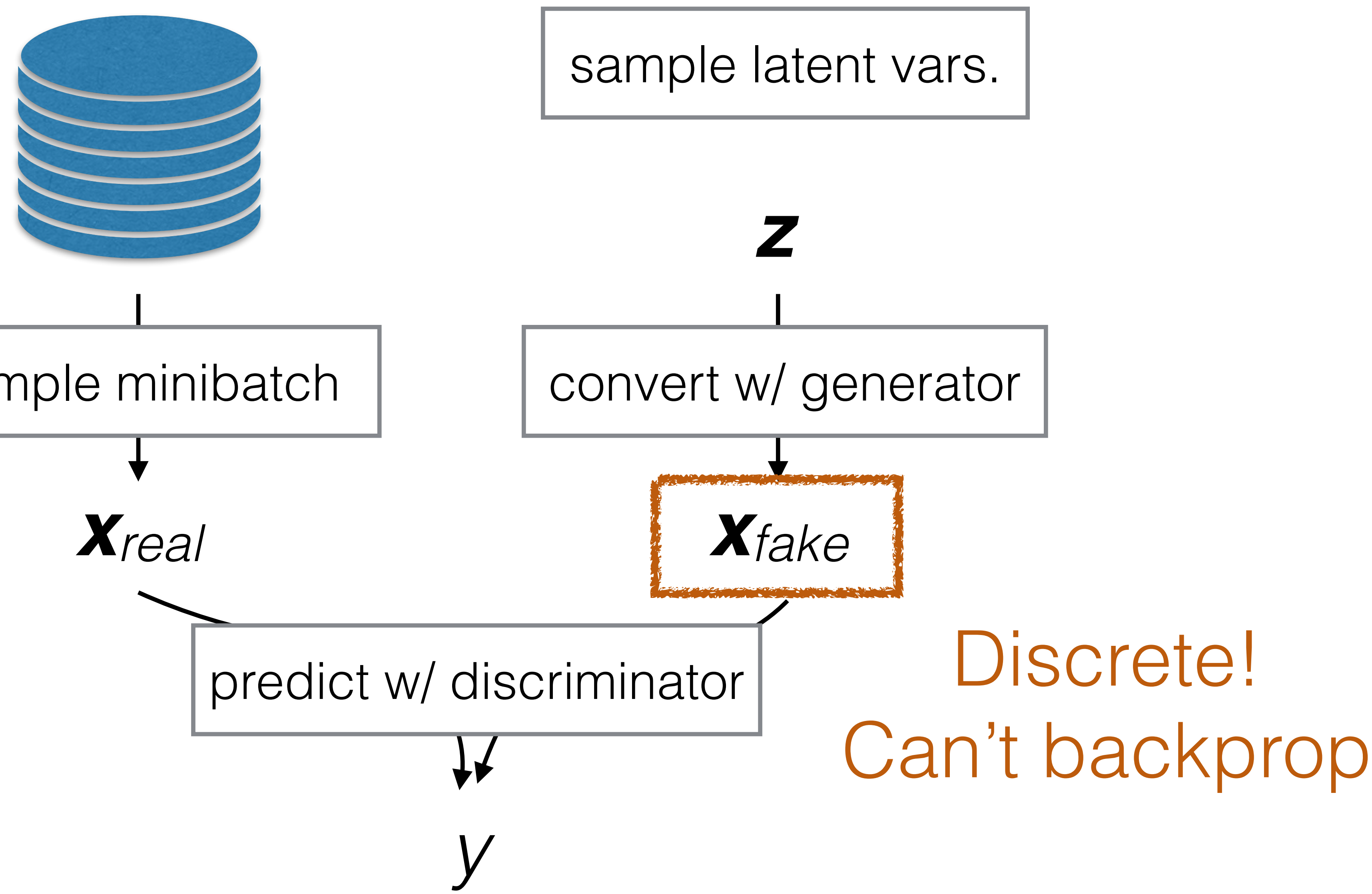
- GANs are great, but **training** is notoriously **difficult**
- Known problems
  - Convergence & Stability:
    - WGAN (Arjovsky et al., 2017)
    - Gradient-Based Regularization (Roth et al., 2017)
  - Mode collapse/dropping:
    - Mini-batch Discrimination (Salimans et al. 2016)
    - Unrolled GAN (Metz et al. 2016)
  - Overconfident discriminator:
    - One-side label smoothing (Salimans et al. 2016)

Applying GANs to Text

Applications of GAN Objectives to Language

- GANs for Language Generation (Yu et al. 2017)
- GANs for MT (Yang et al. 2017, Wu et al. 2017, Gu et al. 2017)
- GANs for Dialogue Generation (Li et al. 2016)

Problem! Can't Backprop through Sampling



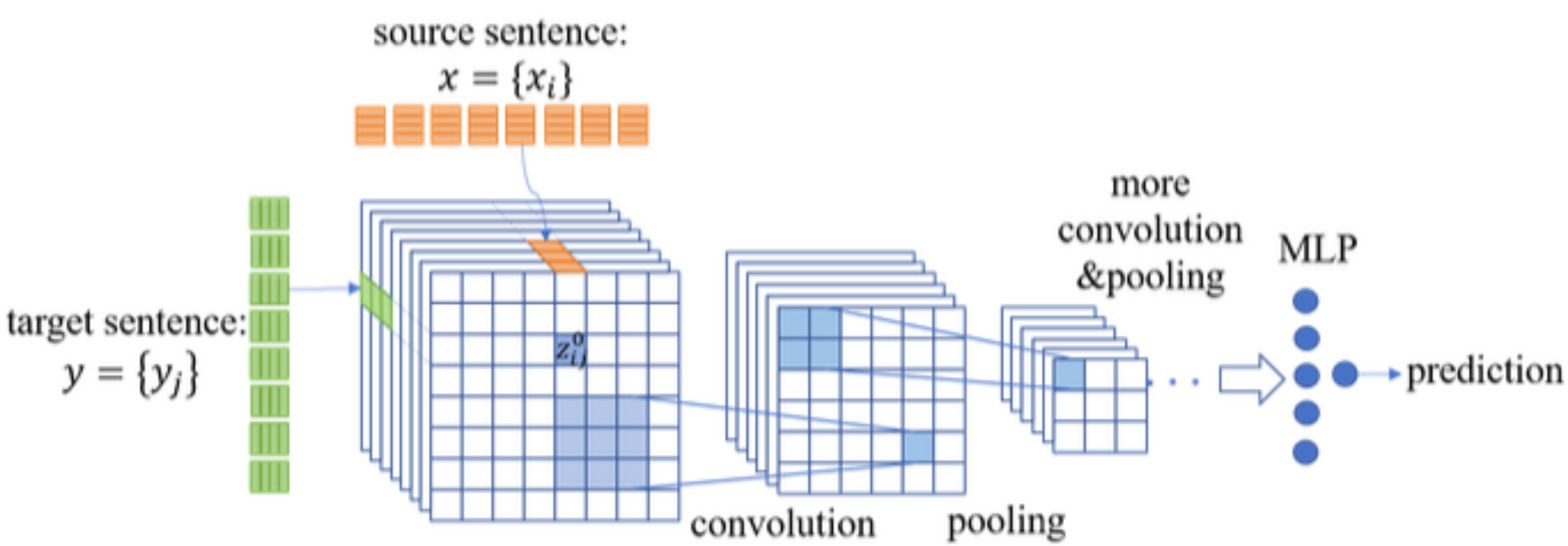
Solution: Use Learning Methods for Latent Variables

- Policy gradient reinforcement learning methods (e.g. Yu et al. 2016)

- Reparameterization trick for latent variables using Gumbel softmax (Gu et al. 2017)

Discriminators for Sequences

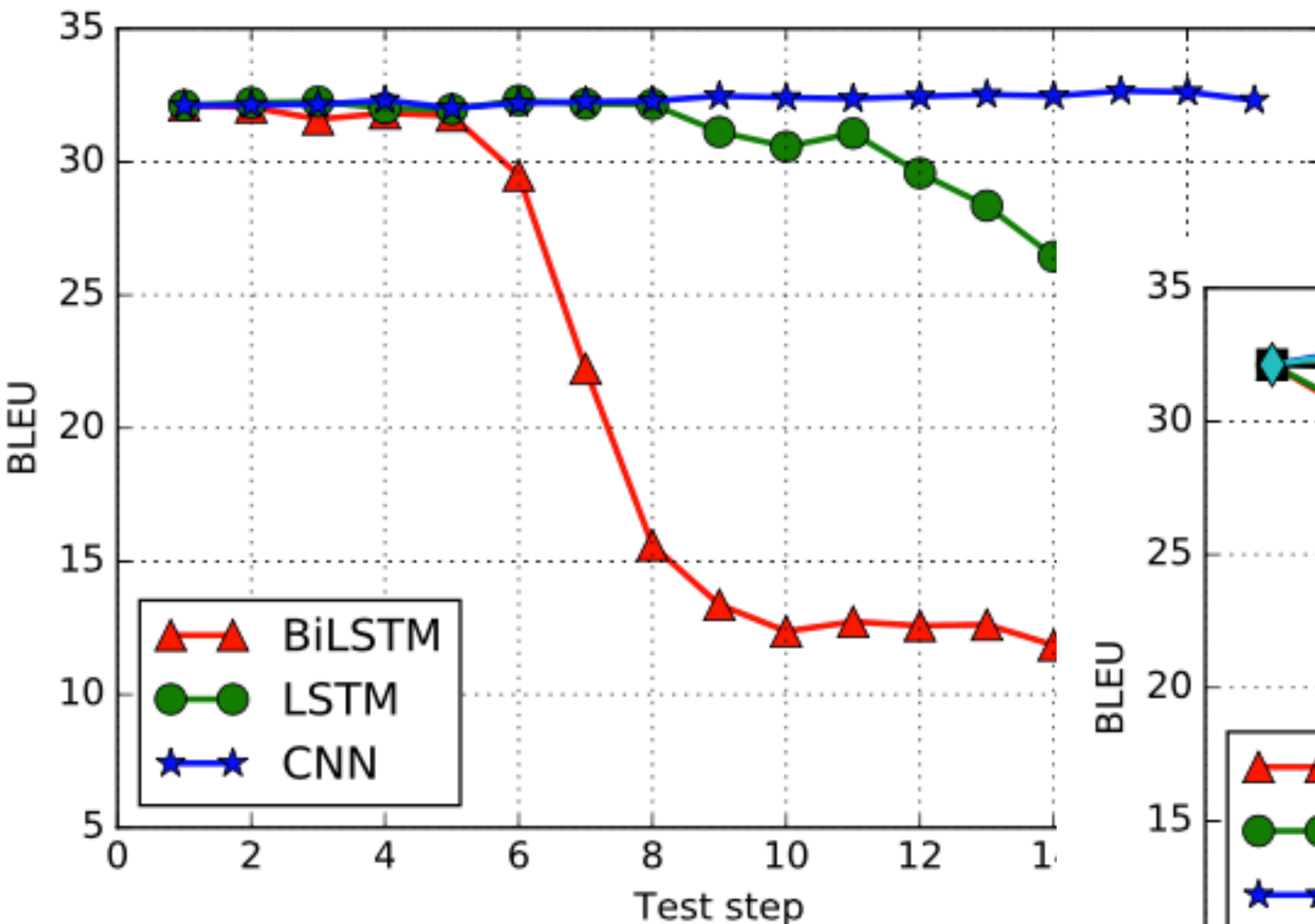
- Decide whether a particular generated output is true or not
- Commonly use CNNs as discriminators, either on sentences (e.g. Yu et al. 2017), or pairs of sentences (e.g. Wu et al. 2017)



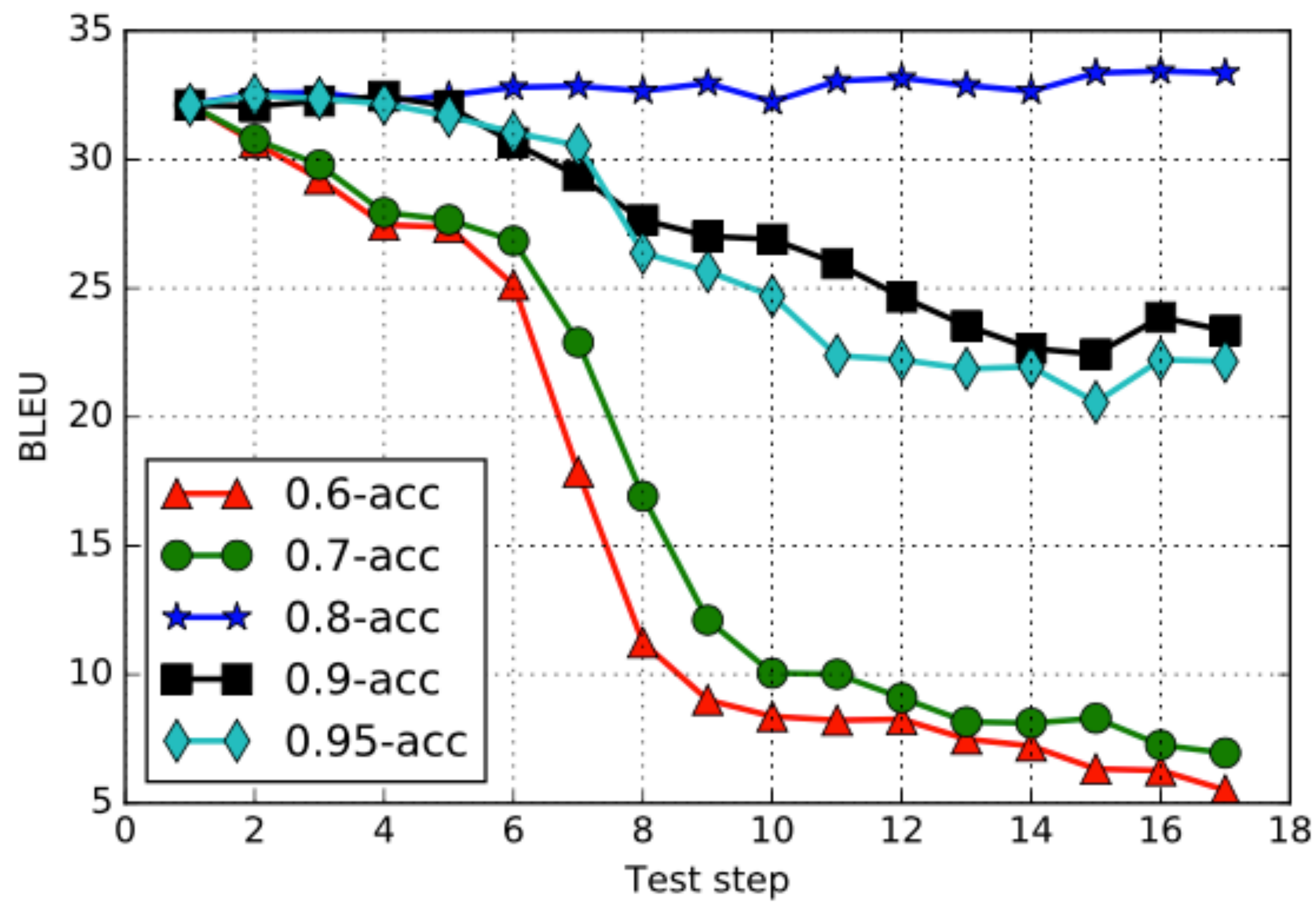
GANs for Text are Hard!

(Yang et al. 2017)

Type of Discriminator



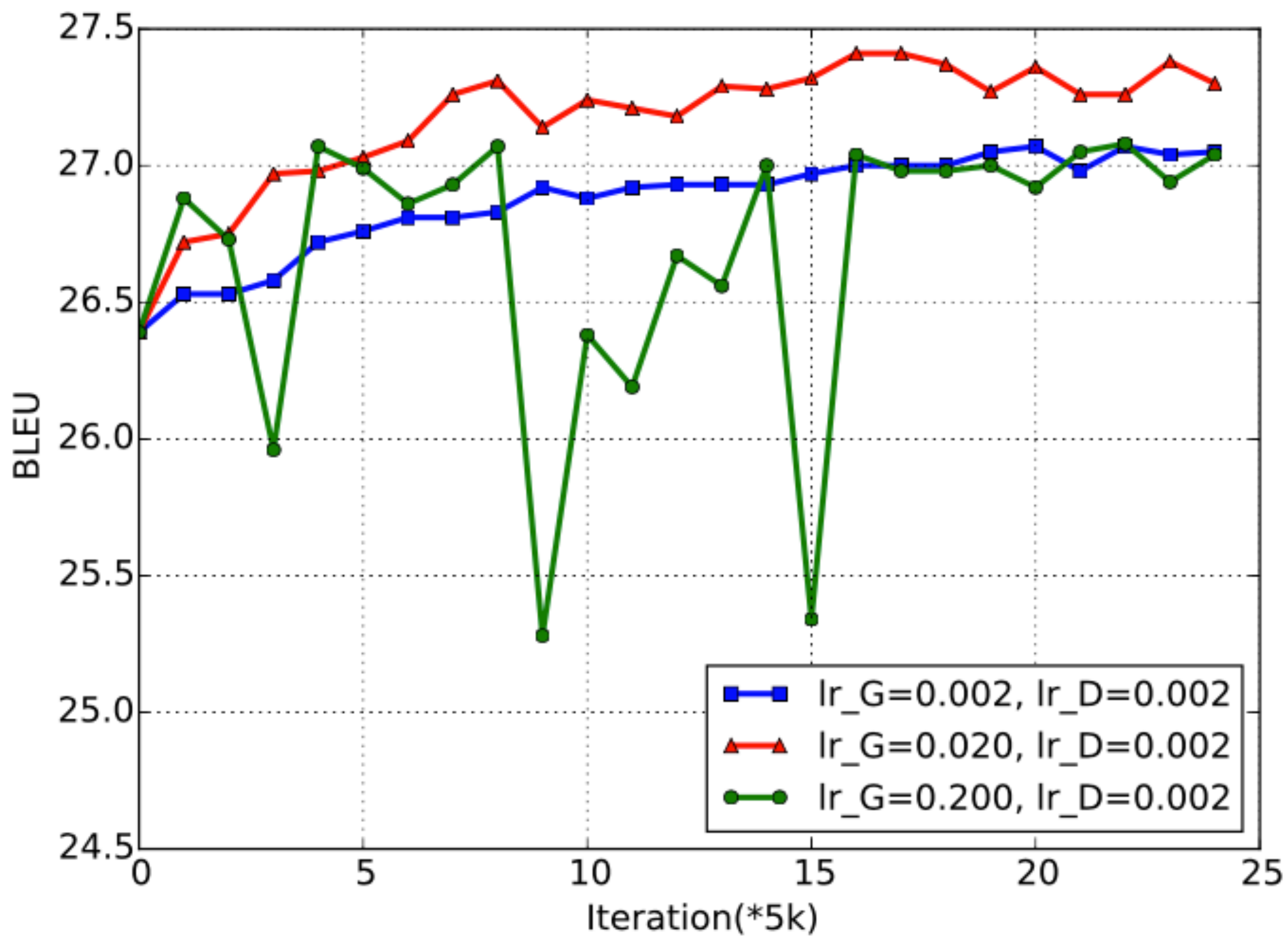
Strength of Discriminator



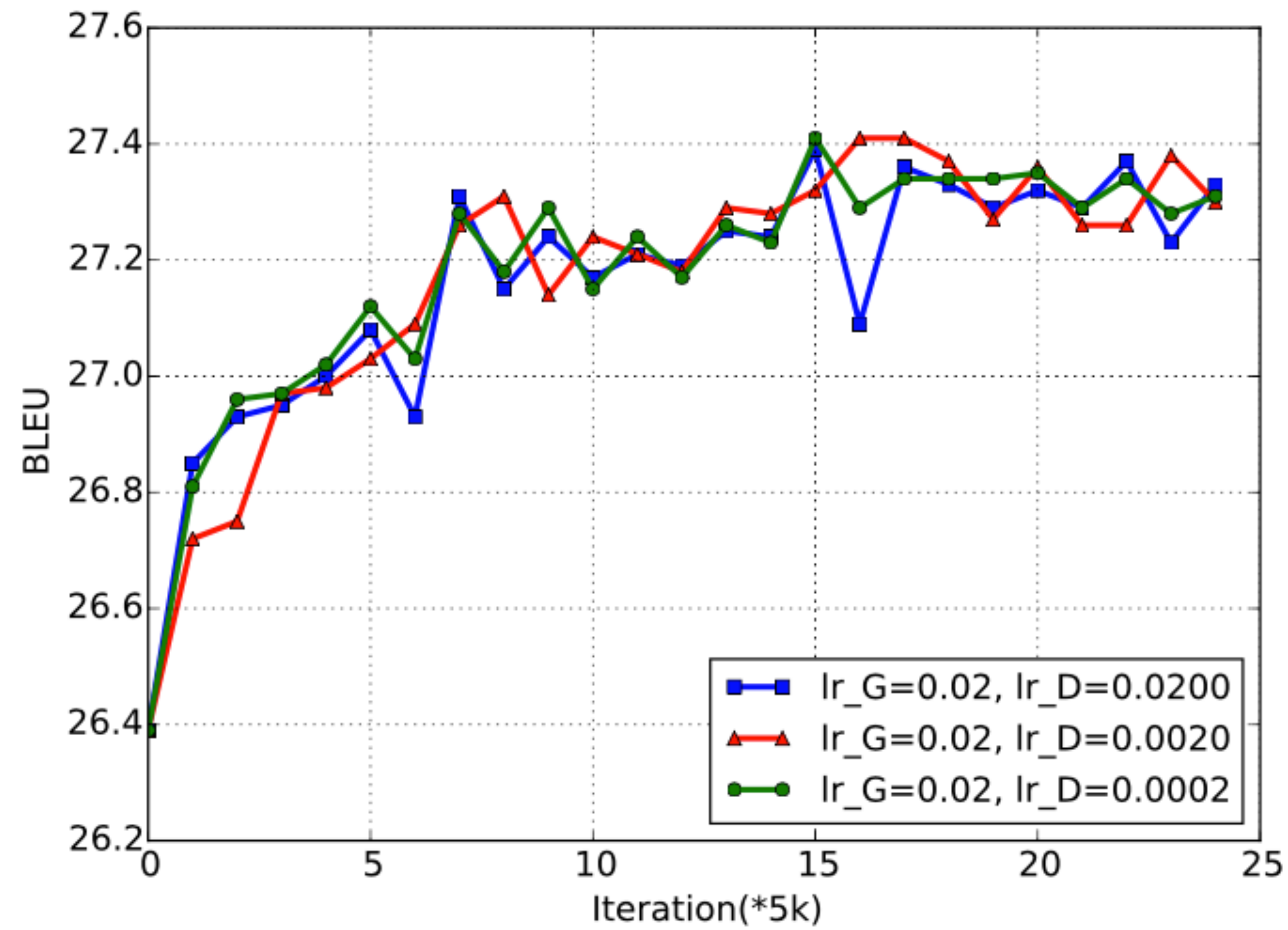
GANs for Text are Hard!

(Wu et al. 2017)

Learning Rate for Generator



Learning Rate for Discriminator



Stabilization Trick: Assigning Reward to Specific Actions

- Getting a reward at the end of the sentence gives a credit assignment problem

- Solution: assign reward for partial sequences (Yu et al. 2016, Li et al. 2017)

D(this)

D(this is)

D(this is a)

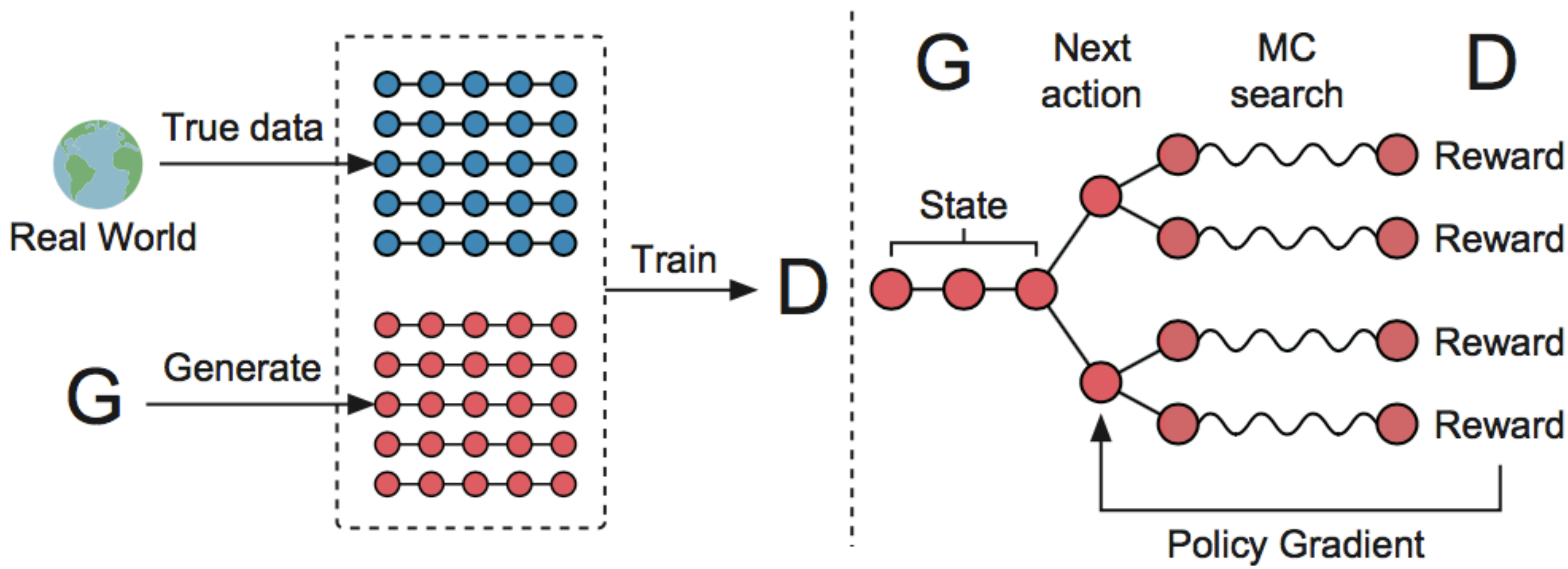
D(this is a fake)

D(this is a fake sentence)

Stabilization Tricks: Performing Multiple Rollouts

- Like other methods using discrete samples, instability is a problem

- This can be helped somewhat by doing multiple rollouts (Yu et al. 2016)



Discrimination over Softmax Results (Hu et al. 2017)

- Attempt to generate outputs with a specific trait (e.g. tense, sentiment)
- Discriminator over the softmax results

$$\boldsymbol{x} \longrightarrow \boldsymbol{h} \longrightarrow \boldsymbol{P(y)} \longrightarrow \boldsymbol{y}$$

Adversary!

Adversarial Feature Learning

Adversaries over Features vs. Over Outputs

- Generative adversarial networks

$$\boldsymbol{x} \longrightarrow \boldsymbol{h} \longrightarrow \boldsymbol{y}$$

Adversary!

- Adversarial feature learning

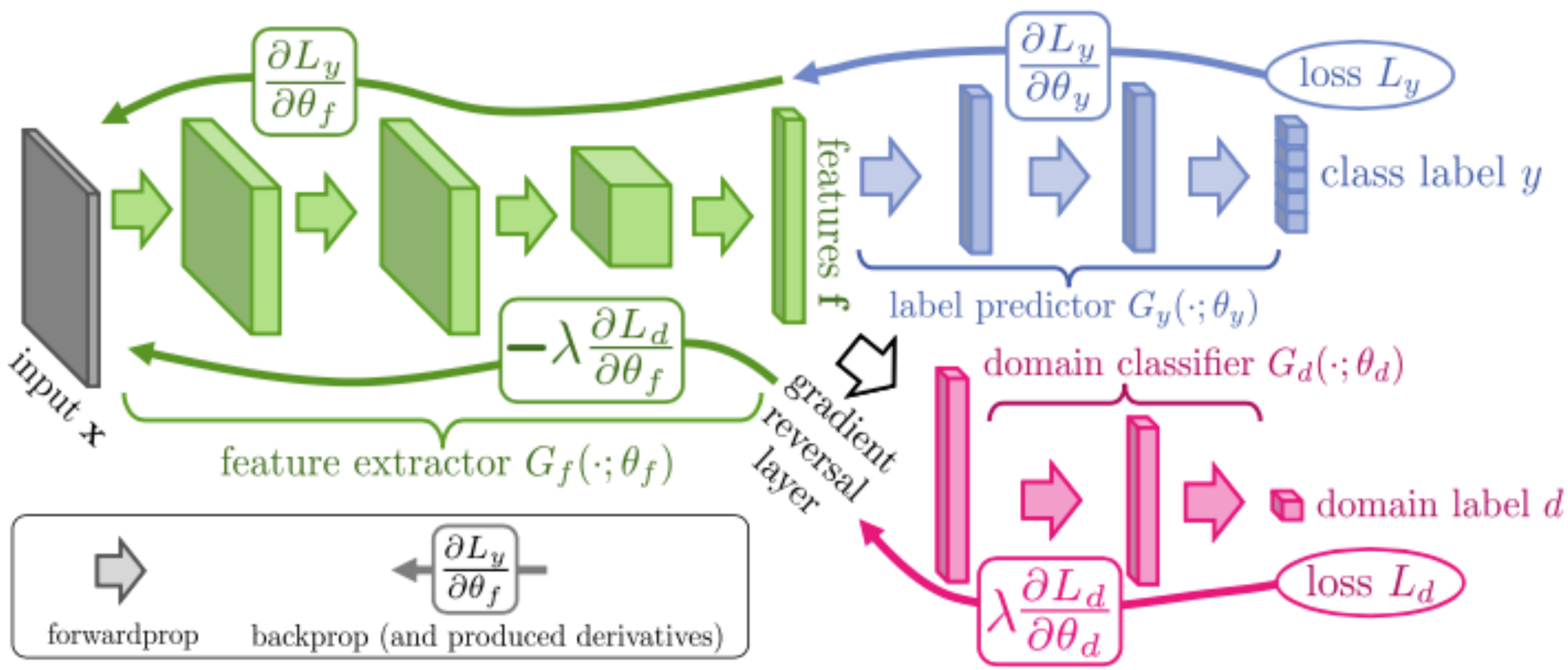
$$\boldsymbol{x} \longrightarrow \boldsymbol{h} \longrightarrow \boldsymbol{y}$$

Adversary!

- Why adversaries over features?
  - Non-generative tasks
  - Continuous features easier than discrete outputs

Learning Domain-invariant Representations (Ganin et al. 2016)

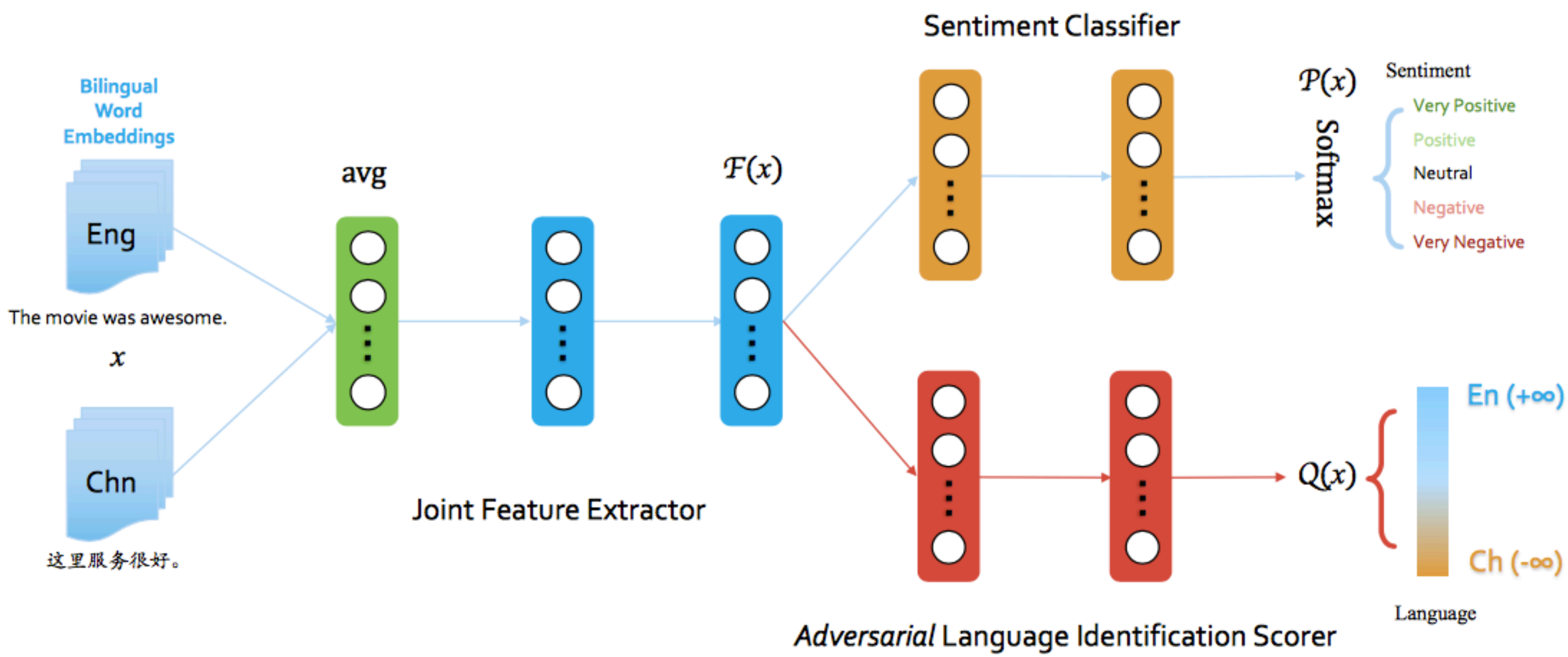
- Learn features that cannot be distinguished by domain



- Interesting application to synthetically generated or stale data (Kim et al. 2017)

Learning Language-invariant Representations

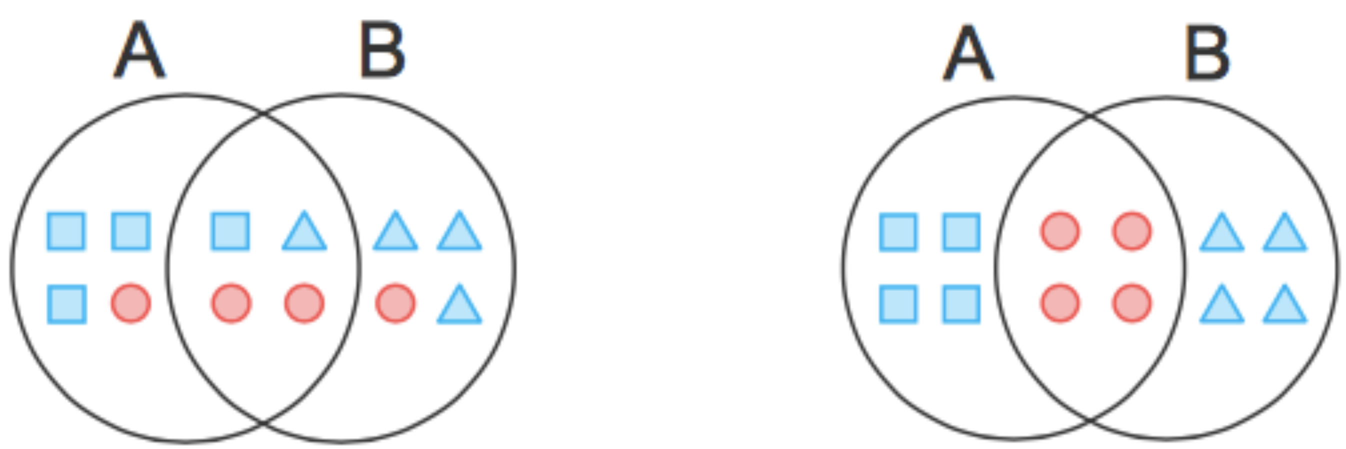
- Chen et al. (2016) learn language-invariant representations for text classification



- Also on multi-lingual machine translation (Xie et al. 2017)

Adversarial Multi-task Learning (Liu et al. 2017)

- Basic idea: want some features in a shared space across tasks, others separate



(a) Shared-Private Model (b) Adversarial Shared-Private Model

- Method: adversarial discriminator on shared features, orthogonality constraints on separate features

Professor Forcing
(Lamb et al. 2016)

- Halfway in between a discriminator on discrete outputs and feature learning
  - Generate output sequence according to model
  - But train discriminator on hidden states

(sampled or true output sequence)

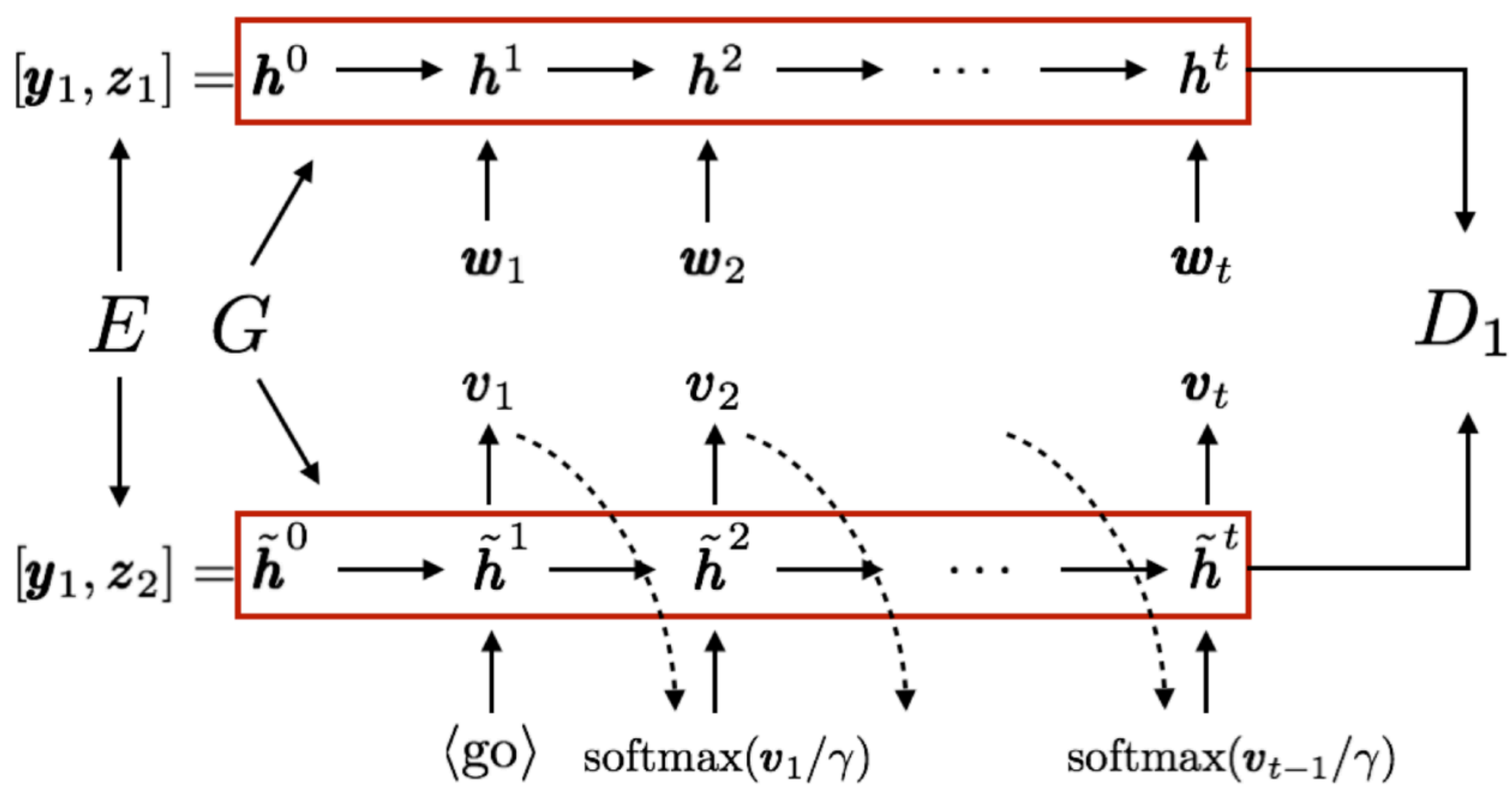
$$\boldsymbol{x} \longrightarrow \boldsymbol{h} \longrightarrow \boldsymbol{y}$$

Adversary!

Unsupervised Distribution Matching

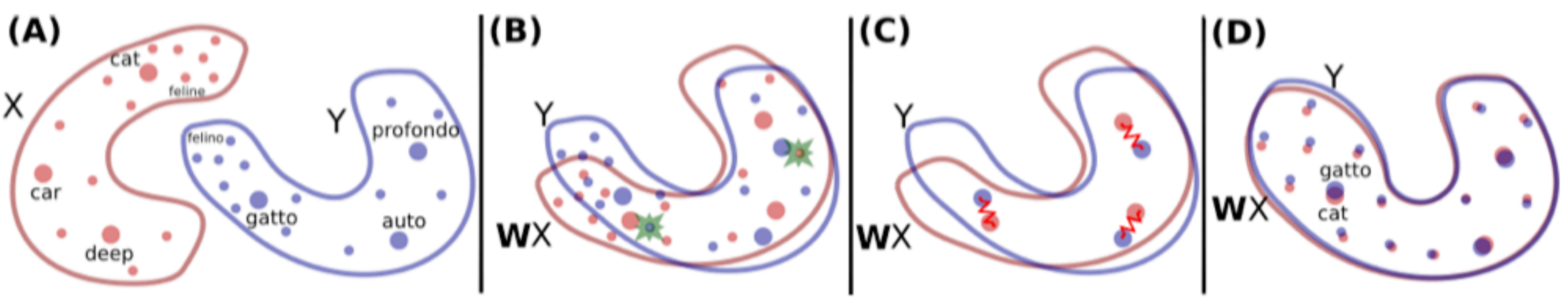
Unsupervised Style Transfer for Text (Shen et al. 2017)

- Task: transfer sentences with one style to another style
  - Decipherment: Translate ciphered sentences to natural sentences
  - Transfer sentences with positive sentiment to negative sentiment.
  - Word reordering
- Impressive performance on decipherment



Unsupervised Alignment of Word Embeddings (Lample et al. 2018)

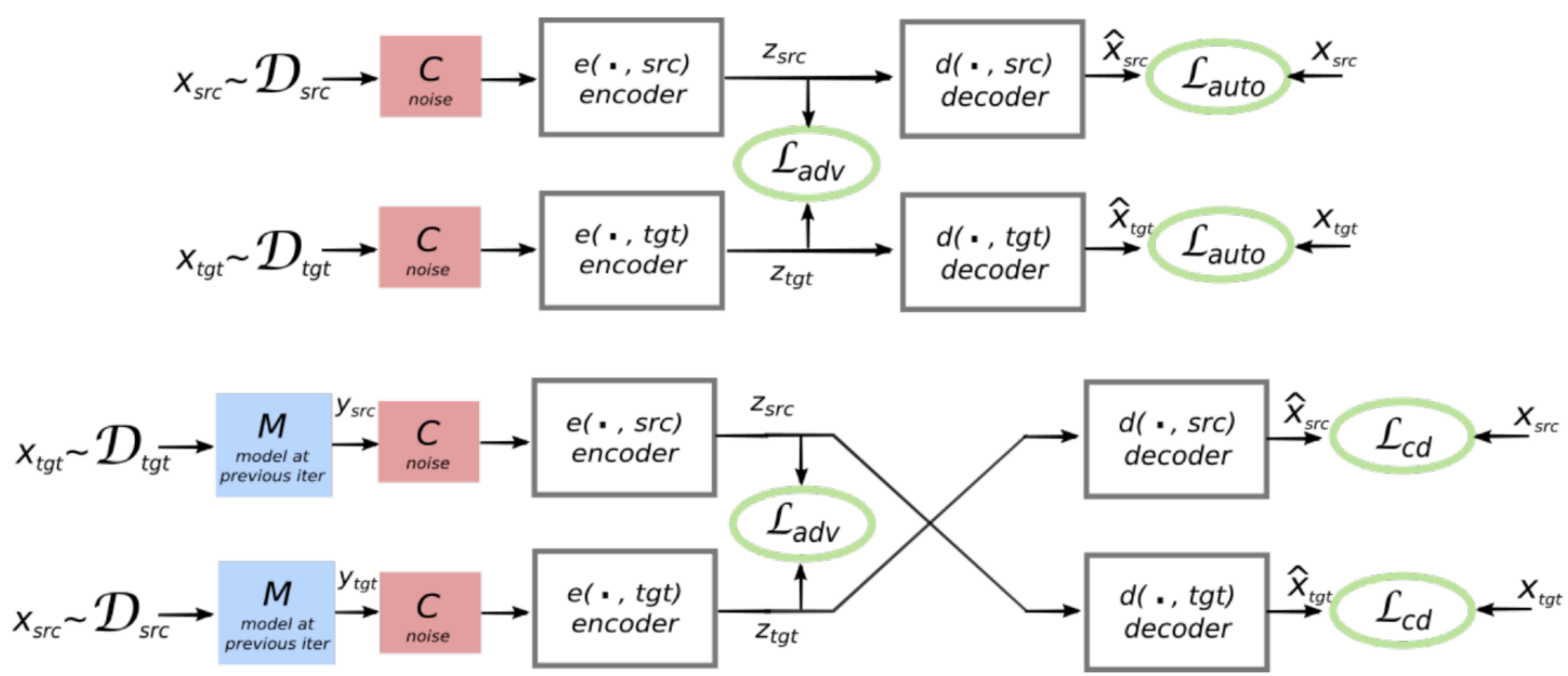
- We have two word embedding spaces (A) in different languages
- Define a function (e.g. orthogonal transform) to map between the spaces
- Use adversarial loss to try to align (B), further find closest words (C), use supervised objective (D)



Unsupervised Machine Translation

(Lample et al. 2017, Artetxe et al. 2017)

- Cycle consistency (dual learning) (He et al. 2016, Zhu et al. 2017)
- Employing denoising auto-encoder to refine translated sentence

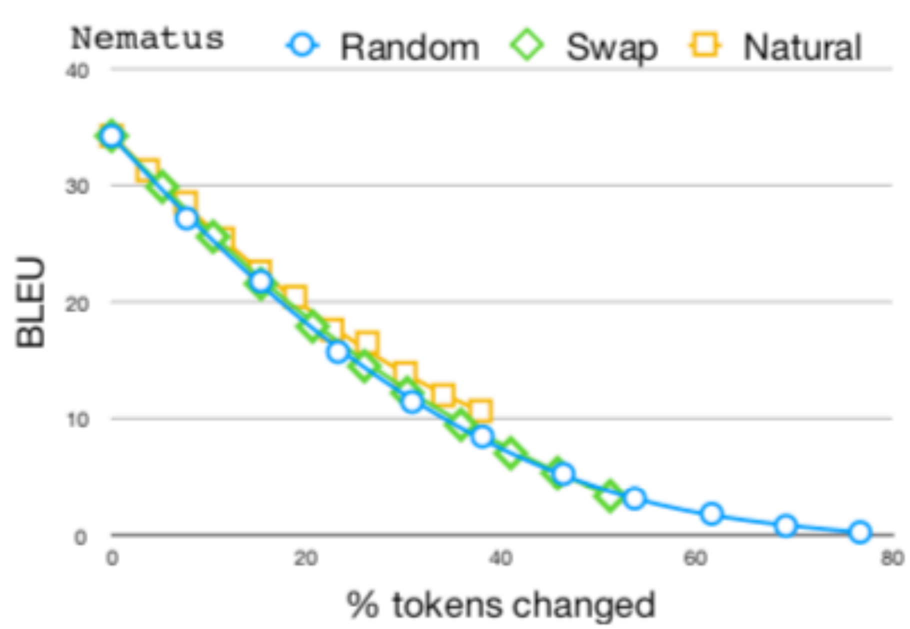


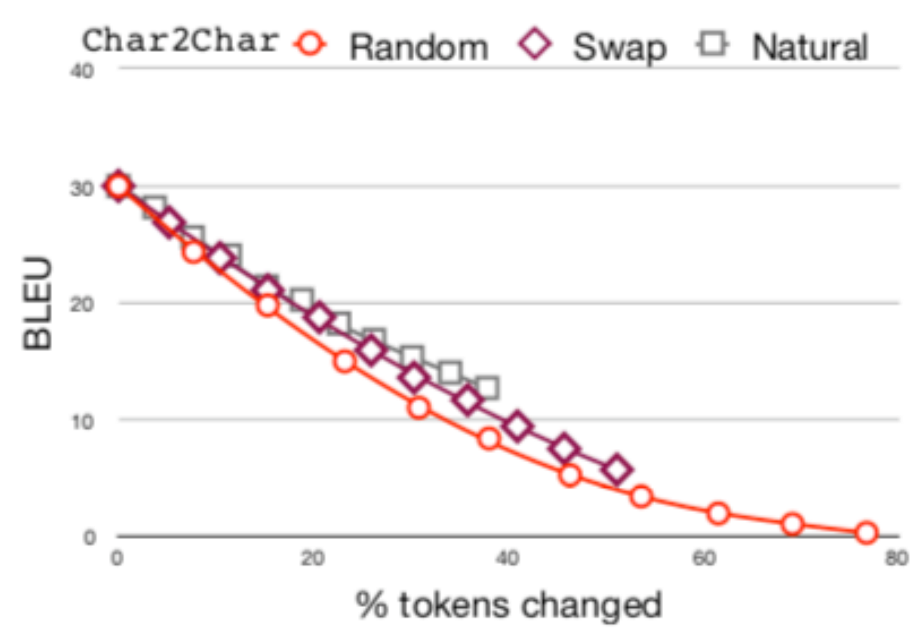
Adversarial Robustness

Problem!
Networks Sensitive to Small Perturbations
(e.g. Belinkov et al. 2018)

Table 4: An example noisy text with human and machine translations.

Input	Luat eienr Stduie der Cambrdige Unievrstit speilt es kenie Rlloe in welcehr Reiehnfogle die Buhcstbaen in eniem Wrot vorkmomen, die eingzie whctige Sahce ist, dsas der ertse und der lettze Buhcstbaen stmimt .
Human	According to a study from Cambridge university, it doesn't matter which order letters in a word are, the only important thing is that the first and the last letter appear in their correct place.
`char2char`	Cambridge Universttte is one of the most important features of the Cambridge Universttten , which is one of the most important features of the Cambridge Universttten .
`Nematus`	Luat eienr Stduie der Cambrant Unievrstilt splashed it kenie Rlloe in welcehr Reiehnfogle the Buhcstbaen in eniem Wred vorkmomen, die eingzie whcene Sahce ist, DSAs der ertse und der lettze Buhcstbaen stmimt .
`charCNN`	According to the <unk> of the Cambridge University , it 's a little bit of crude oil in a little bit of recycling , which is a little bit of a cool cap , which is a little bit of a strong cap , that the fat and the <unk> bites is consistent .





Adversarial Noise: Noise Specifically Designed to Break Systems

- Relatively simple to perform attacks on image classification systems: calculate gradient to maximize loss

- More difficult for text because input is discrete, but still some success (e.g. Ebrahimi et al. 2018)

src	1901 wurde eine Frau namens Auguste in eine medizinische Anstalt in Frankfurt gebracht.
adv	1901 wurde eine Frau namens **Afuiguste** in eine medizinische Anstalt in Frankfurt gebracht.
src-output	In 1931, a woman named **Augustine** was brought into a medical institution in France.
adv-output	In 1931, a woman named Rutgers was brought into a medical institution in France.
src	Das ist Dr. Bob Childs – er ist Geigenbauer und Psychotherapeut.
adv	Das ist Dr. Bob Childs – er ist Geigenbauer und **Psy6hothearpeiut**.
src-output	This is Dr. Bob Childs – he's a wizard maker and a **therapist**'s **therapist**.
adv-output	This is Dr. Bob Childs – he's a brick maker and a **psychopath**.

Table 1: Controlled and Targeted Attack on DE→EN NMT. In the first example, the adversary wants to suppress a person's name, and in the second example, to replace occurrences of *therapist* with *psychopath*

What is an Adversarial Example? (Michel et al. 2019)

- It should be "meaning preserving" on the source side, and "meaning destroying" on the target side

$$\boxed{1 - s_{src}(x, \hat{x})} < \boxed{d_{tgt}(y, y_M, \hat{y}_M)}$$

Source meaning destruction   Target meaning destruction

$$d_{tgt}(y, y_M, \hat{y}_M) = \begin{cases} 0 \text{ if } s_{tgt}(y, \hat{y}_M) \geq s_{tgt}(y, y_M) \\ \frac{s_{tgt}(y, y_M) - s_{tgt}(y, \hat{y}_M)}{s_{tgt}(y, y_M)} \text{ otherwise} \end{cases}$$

- Meaning defined by semantic similarity (whatever that means)

Adversarial Training

- We'd like to train our models to be robust to attacks!

- Simplest idea: sample adversarial examples at training time and make sure that they are also classified correctly

- Lots of theory, but little for NLP tasks

https://adversarial-ml-tutorial.org

Questions?