GANs
+
Final practice questions

Lecture 23

CS 753

Instructor: Preethi Jyothi

Final Exam Syllabus

1. WFST algorithms/WFSTs used in ASR
2. HMM algorithms/EM/Tied state Triphone models
3. DNN-based acoustic models
4. N-gram/Smoothing/RNN language models
5. End-to-end ASR (CTC, LAS, RNN-T)
6. MFCC feature extraction
7. Search & Decoding
8. HMM-based speech synthesis models
9. Multilingual ASR
10. Speaker Adaptation
11. Discriminative training of HMMs

Questions can be asked on any of the 11 topics listed above. You will be allowed a single A-4 cheat sheet of **handwritten notes**; content on both sides permitted.

Final Project

Deliverables

- 4-5 page final report:
  - ✓ Task definition, Methodology, Prior work, Implementation Details, Experimental Setup, Experiments and Discussion, Error Analysis (if any), Summary
- Short talk summarizing the project:
  - ✓ Each team will get 8-10 minutes for their presentation and $\approx$5 minutes for Q/A
  - ✓ Clearly demarcate which team member worked on what part

Final Project Grading

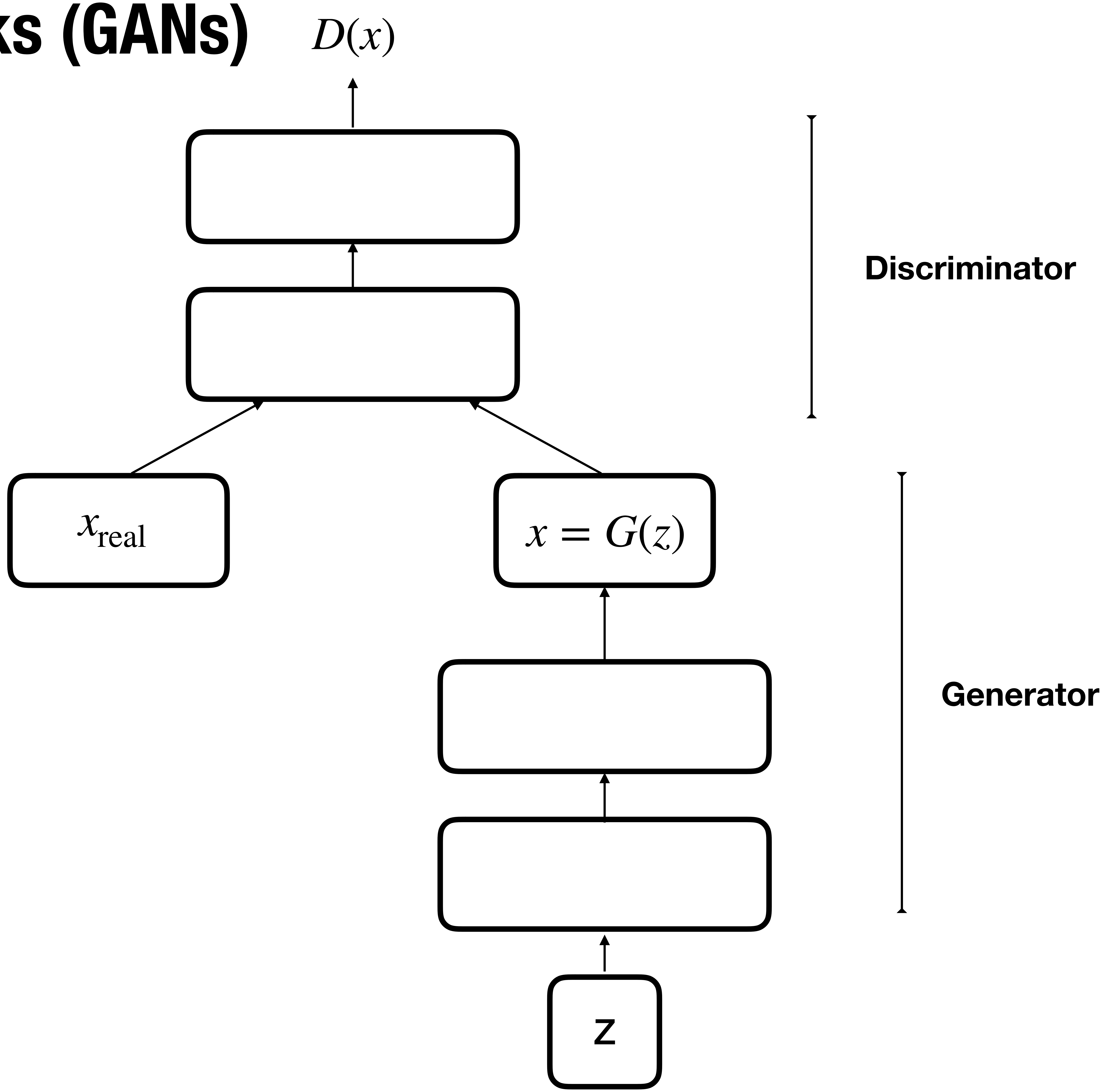
- Break-up of 20 points:
  - 6 points for the report
  - 4 points for the presentation
  - 6 points for Q/A
  - 4 points for overall evaluation of the project

Final Project Schedule

- Presentations will be held on Nov 23rd and Nov 24th
- The final report in pdf format should be sent to `pjyothi@cse.iitb.ac.in` before Nov 24th
- The order of presentations will be decided on a lottery basis and shared via Moodle before Nov 9th

Generative Adversarial Networks (GANs)

- Training process is formulated as a game between a generator network and a discriminative network
  - Objective of the generator: Create samples that seem to be from the same distribution as the training data
  - Objective of the discriminator: Examine a generated sample and distinguish between fake or real samples
- The generator tries to fool the discriminator network



Generative Adversarial Networks

$$\max_G \min_D \mathcal{L}(G, D)$$

$$\text{where } \mathcal{L}(G, D) = \mathrm{E}_{x \in D}[-\log D(x)] + \mathrm{E}_z[-\log(1 - D(G(z)))]$$

- Cost function of the generator is the opposite of the discriminator’s
- Minimax game: The generator and discriminator are playing a zero-sum game against each other

Training Generative Adversarial Networks

for number of training iterations **do**

for k steps **do**

• Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.

• Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.

• Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

• Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.

• Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Image from [Goodfellow16]: https://arxiv.org/pdf/1701.00160.pdf

Better objective for the generator

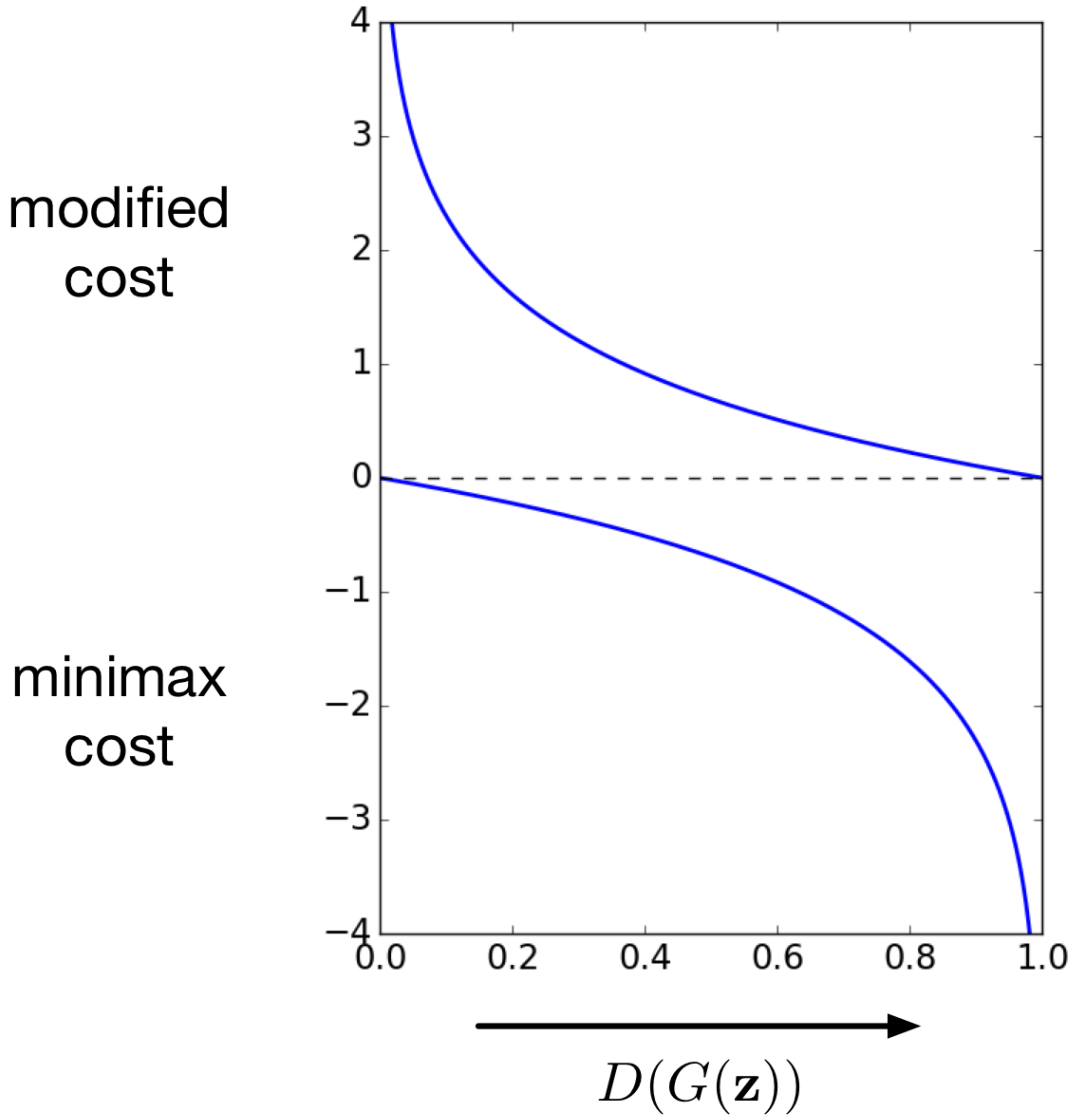
- Problem of saturation: If the generated sample is really poor, the generator's cost is relatively flat

- Original cost

$$\mathcal{L}_{\mathrm{GEN}}(G, D) = \mathrm{E}_z[\log(1 - D(G(z)))]$$

- Modified cost

$$\mathcal{L}_{\mathrm{GEN}}(G, D) = \mathrm{E}_z[-\log D(G(z))]$$



Large (& growing!) list of GANs

The GAN Zoo

- 3D-ED-GAN - Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (github)
- 3D-IWGAN - Improved Adversarial Systems for 3D Object Generation and Reconstruction (github)
- 3D-PhysNet - 3D-PhysNet: Learning the Intuitive Physics of Non-Rigid Object Deformations
- 3D-RecGAN - 3D Object Reconstruction from a Single Depth View with Adversarial Learning (github)
- ABC-GAN - ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks (github)
- ABC-GAN - GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- ACGAN - Coverless Information Hiding Based on Generative adversarial networks
- acGAN - On-line Adaptative Curriculum Learning for GANs
- ACtuAL - ACtuAL: Actor-Critic Under Adversarial Learning
- AdaGAN - AdaGAN: Boosting Generative Models
- Adaptive GAN - Customizing an Adversarial Example Generator with Class-Conditional GANs
- AdvEntuRe - AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples
- AdvGAN - Generating adversarial examples with adversarial networks
- AE-GAN - AE-GAN: adversarial eliminating with GAN

Image from https://github.com/hindupuravinash/the-gan-zoo

Conditional GANs

- Generator and discriminator receive some additional conditioning information

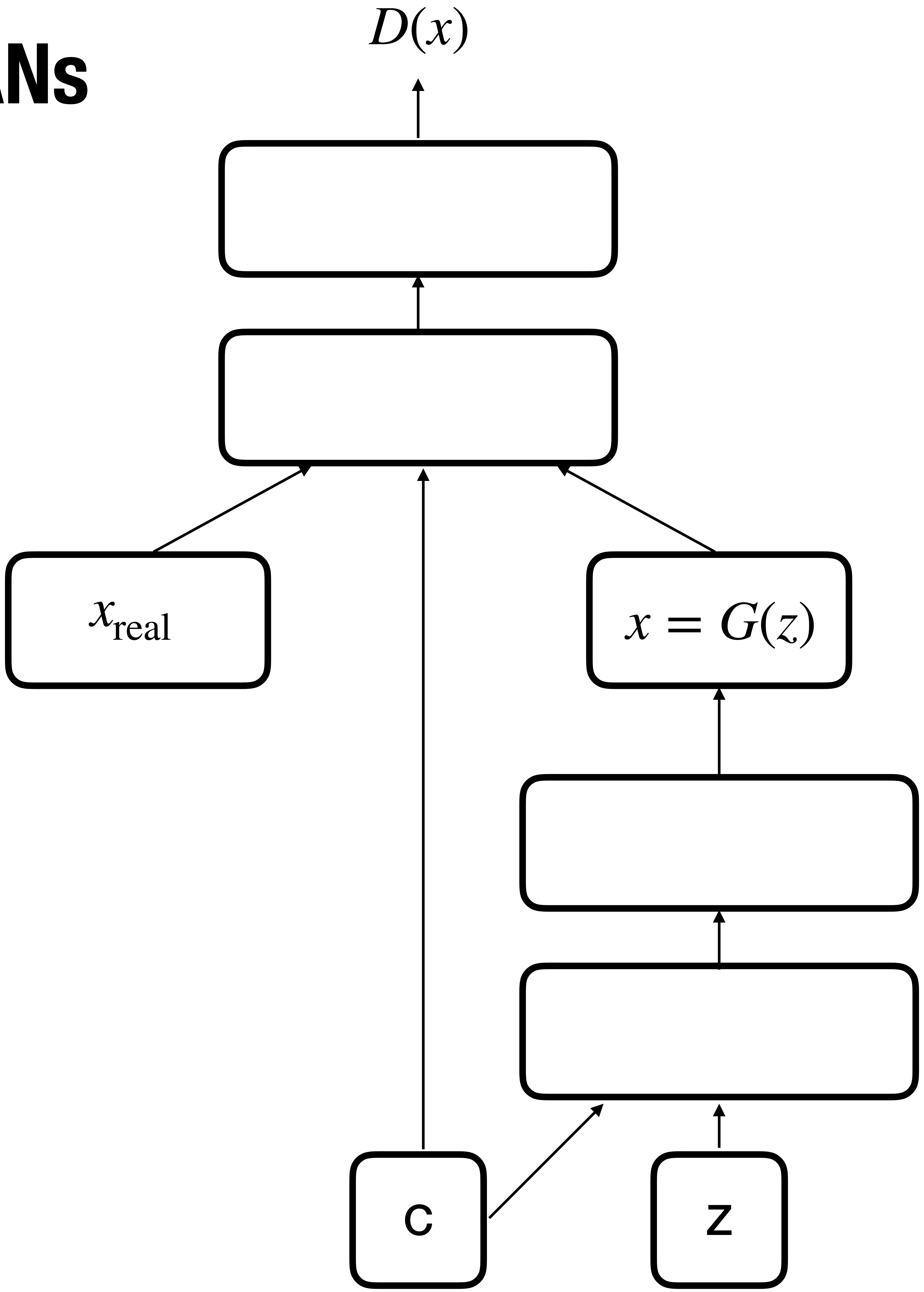


Image-to-image Translation using C-GANs

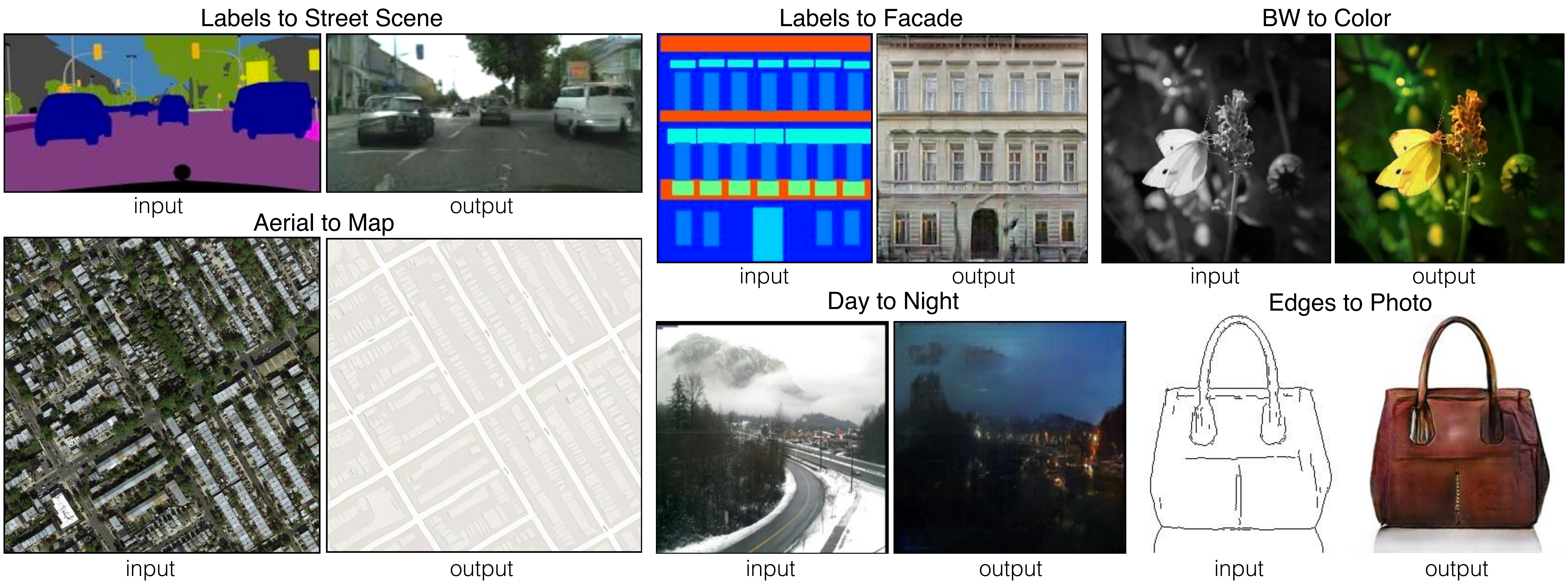


Image from Isola et al., CVPR 2017, https://arxiv.org/pdf/1611.07004.pdf

Text-to-Image Synthesis

this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

the flower has petals that are bright pinkish purple with white stigma

this white and yellow flower have thin white petals and a round yellow stamen

Image from Reed et al., ICML 2016, https://arxiv.org/pdf/1605.05396.pdf

Text-to-Image Synthesis

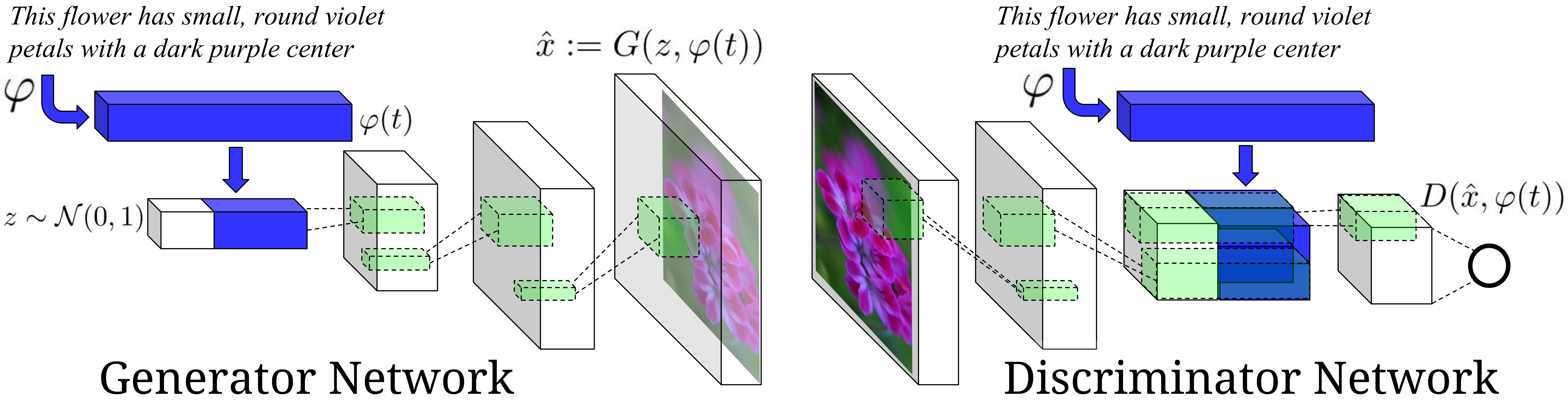


Image from Reed et al., ICML 2016, https://arxiv.org/pdf/1605.05396.pdf

Three Speech Applications of GANs

GANs for speech synthesis

- Generator produces synthesised speech which the Discriminator distinguishes from real speech
- During synthesis, a random noise + linguistic features generates speech

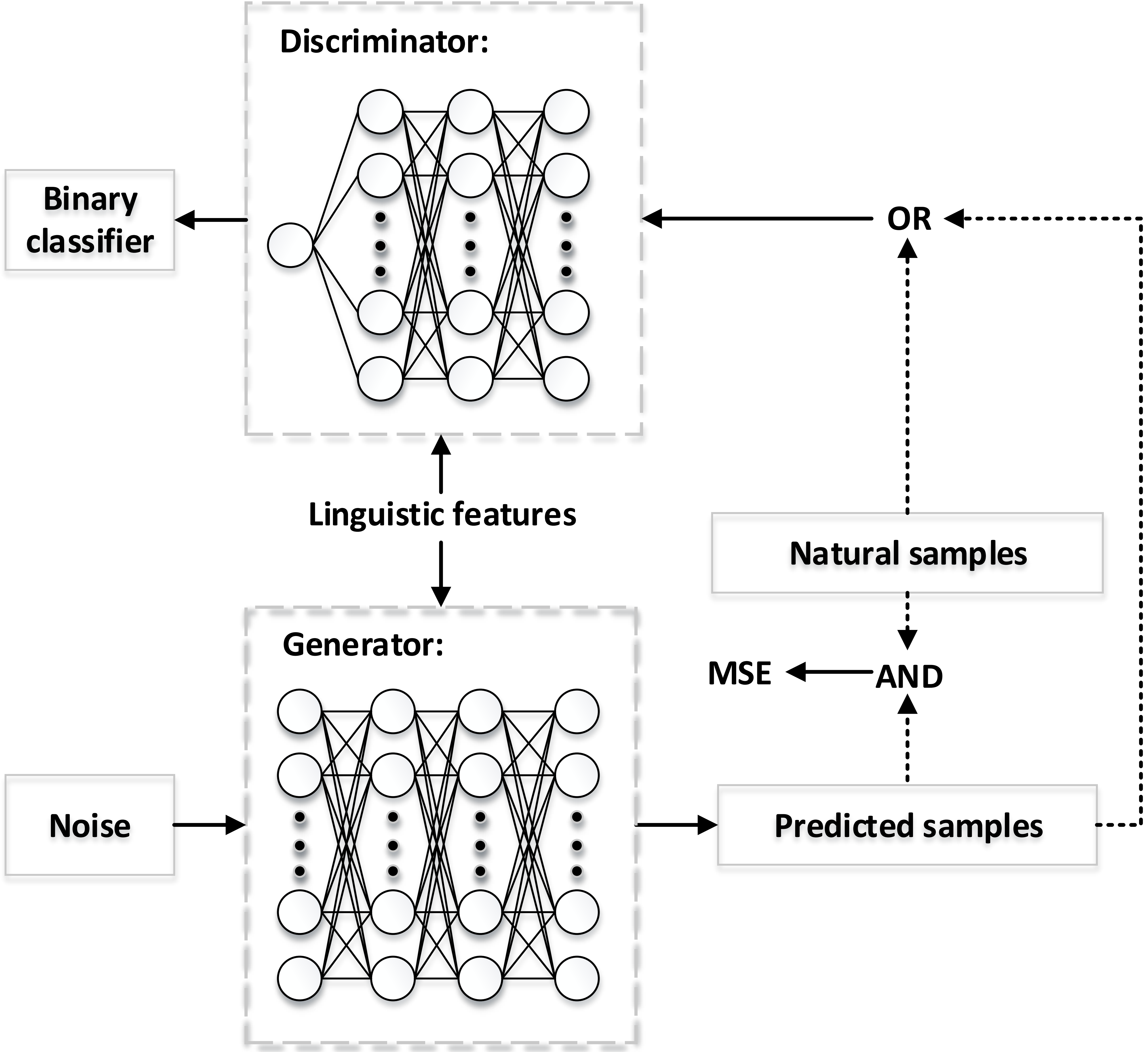
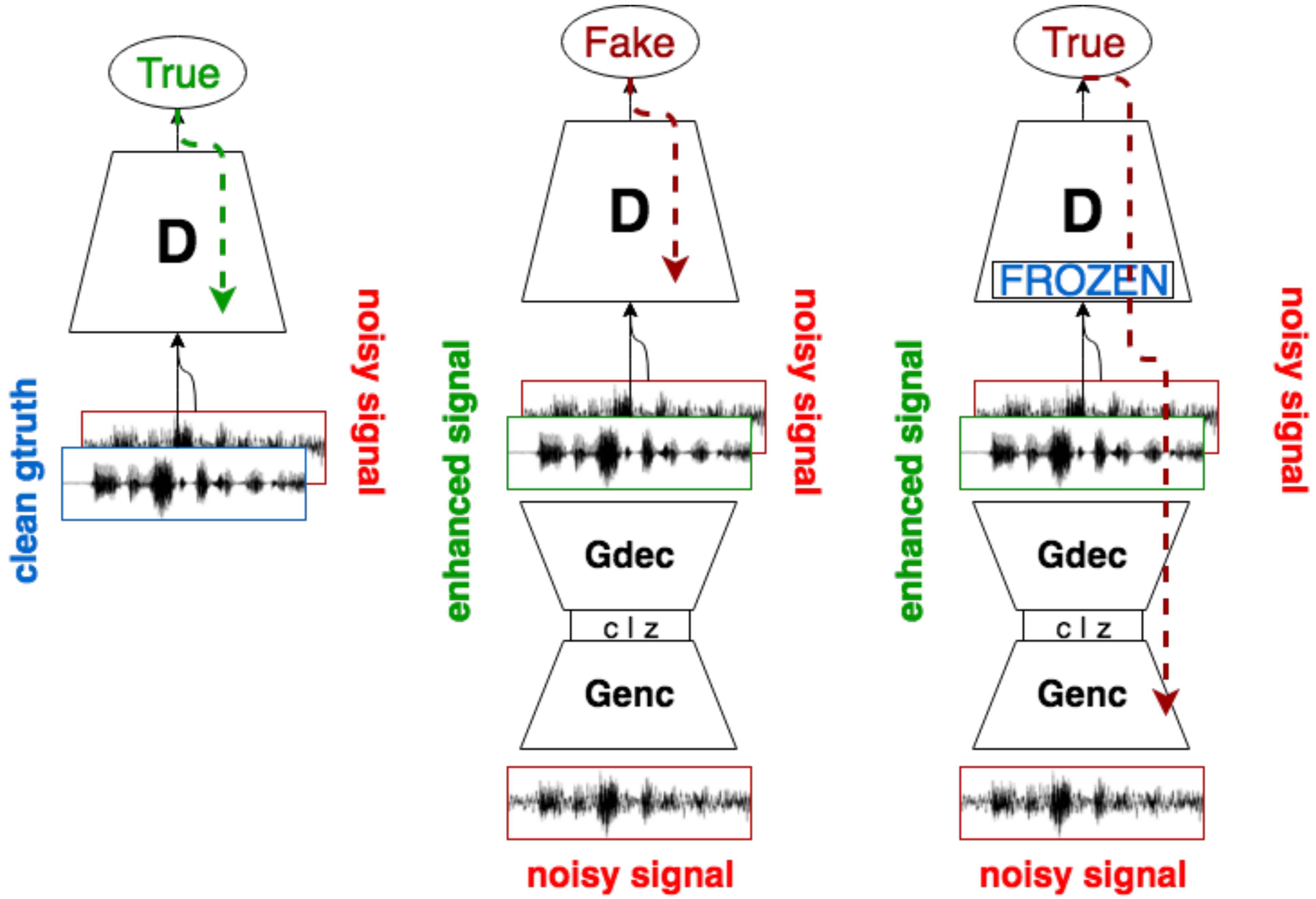


Image from Yang et al., "SPSS using GANs", 2017

SEGAN: GANs for speech enhancement

- Enhancement: Given an input noisy signal $\tilde{x}$, we want to clean it to obtain an enhanced signal x

- Generator G will take both $\tilde{x}$ and z as inputs; G is fully convolutional



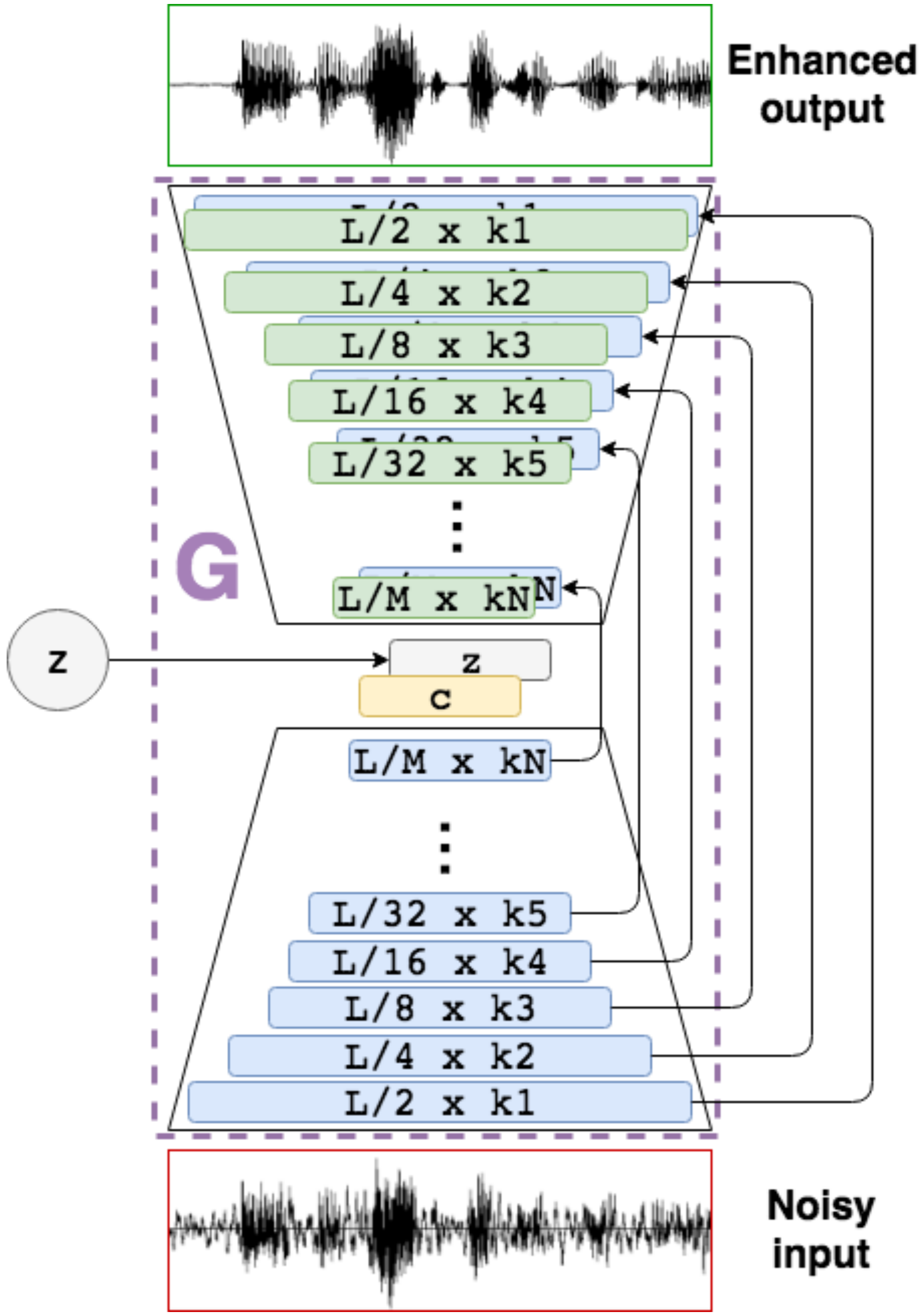


Image from https://arxiv.org/pdf/1703.09452.pdf

Voice Conversion Using Cycle-GANs

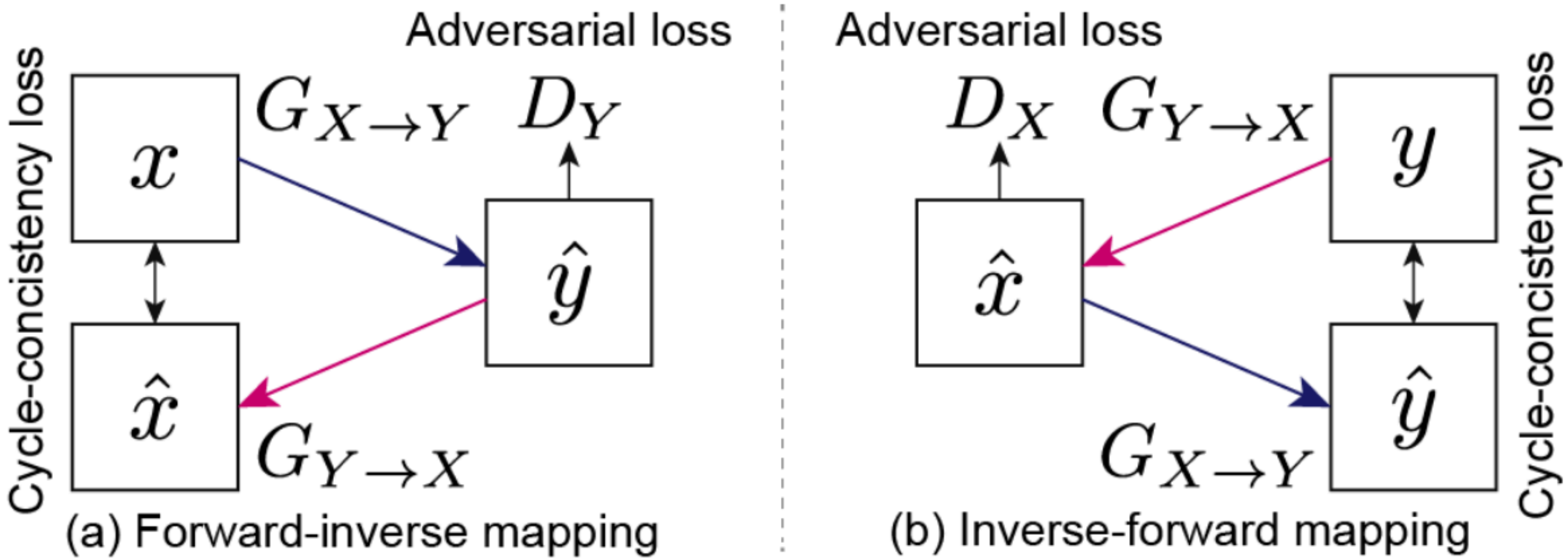
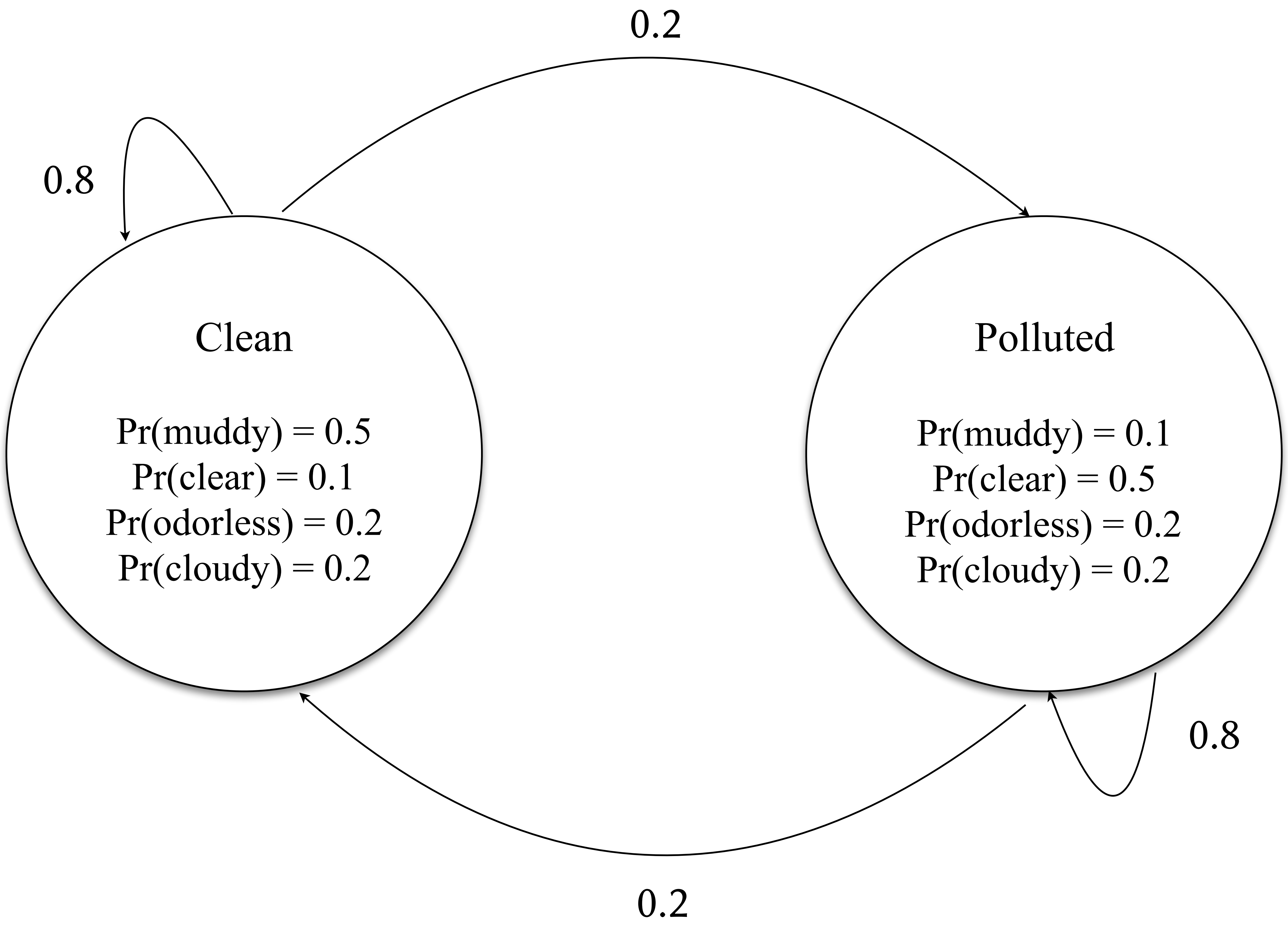


Image from https://arxiv.org/abs/1711.11293

Practice Questions

HMM 101

A water sample collected from Powai lake is either Clean or Polluted. However, this information is hidden from us and all we can observe is whether the water is muddy, clear, odorless or cloudy. We start at time step 1 in the Clean state. The HMM below models this problem. Let qt and Ot denote the state and observation at time step t, respectively.



a) What is P(O_2 = clear)?

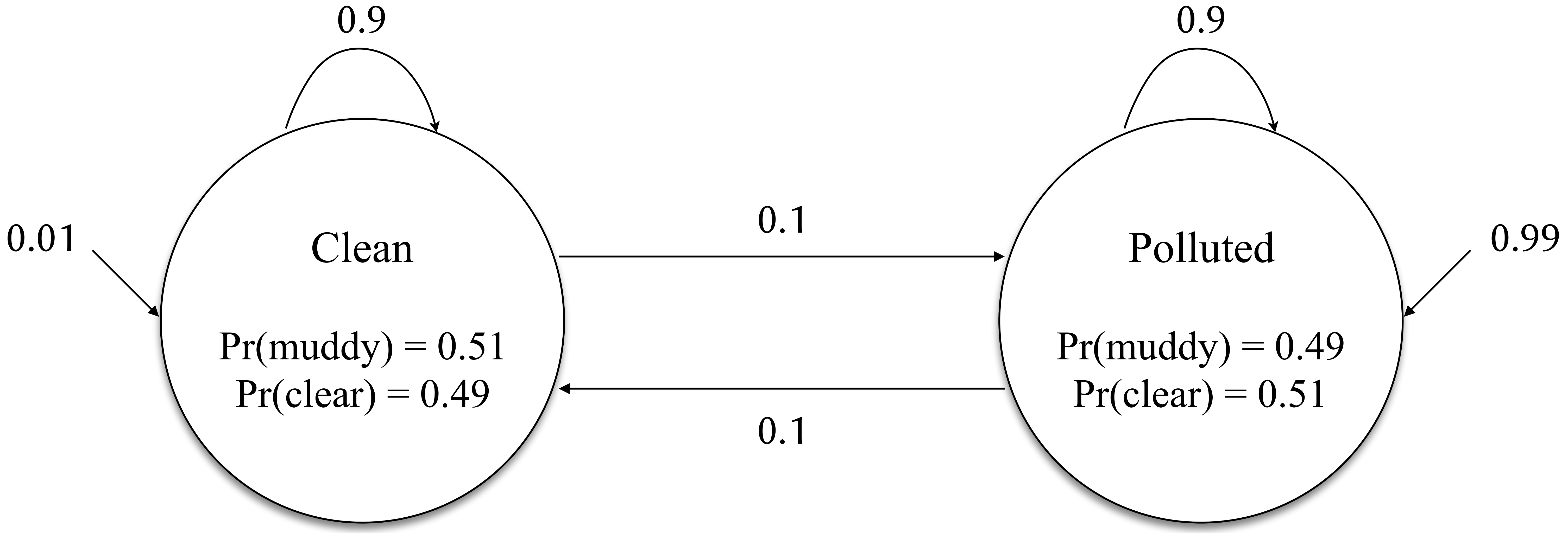
b) What is P(q_2 = Clean | O_2 = clear)?

c) What is P(O_{200} = cloudy)?

d) What's the most likely sequence of states for the following observation sequence: {O_1 = clear, O_2 = clear, O_3 = clear, O_4 = clear, O_5 = clear}?

HMM 101

Say that we are now given a modified HMM for the water samples as shown below. Initial probabilities and transition probabilities are shown next to the arcs. (Note: You do not need to use the Viterbi algorithm to answer the next two questions.)



a) What is the most likely sequence of states given a sequence of three observations: {muddy, muddy, muddy}?

b) Say we observe a very long sequence of "muddy" (e.g. 10 million "muddy" in a row). What happens to the most likely state sequence then?

Handling disfluencies in ASR

Recall that a pronunciation lexicon L maps a sequence of phones to a sequence of words. In this problem, we shall modify L in order to handle some limited forms of interruptions in speech (a.k.a. disfluencies). We will consider a dictionary of two words: W_1 with the phone sequence "a b c" and W_2 with the phone sequence "x y z".

a) Draw the state diagram of the finite-state machine L.

b) We want to modify L such that it accounts for "breaks" when the speaker stops in the middle of a word and says the word all over again. For instance, the word W_1 may be pronounced as "a b ⟨break⟩ a b c," where ⟨break⟩ is a special token produced by the acoustic model. In a valid phone sequence, breaks are allowed to appear only within a word, and not at the end or beginning of a word. Further, two consecutive ⟨break⟩ tokens are not allowed. But a word can be pronounced with an arbitrary number of breaks. E.g. W_1 can be pronounced also as "a b ⟨break⟩ a ⟨break⟩ a b ⟨break⟩ a b c". Let L_1 be an FST (obtained by modifying L from the previous part) that accepts all valid phone sequences with breaks, and outputs a corresponding sequence of words. Draw the state diagram of L_1.

Handling disfluencies in ASR

Recall that a pronunciation lexicon L maps a sequence of phones to a sequence of words. In this problem, we shall modify L in order to handle some limited forms of interruptions in speech (a.k.a. disfluencies). We will consider a dictionary of two words: W_1 with the phone sequence “a b c” and W_2 with the phone sequence “x y z”.

c) Next, we want to modify L_1 such that it can account for both “breaks” and “pauses.” A pause corresponds to when the speaker briefly stops in the middle of a word and continues. For instance, the word W_1 may be pronounced as “a b ⟨pause⟩ c”, “a ⟨break⟩ a ⟨pause⟩ b ⟨break⟩ a b c,” etc. where ⟨pause⟩ is another special token produced by the acoustic model. In a valid phone sequence, these special tokens are allowed to appear only within a word, and two consecutive special tokens are not allowed. Let L_2 be an FST (obtained by modifying L_1 from the previous part) that accepts all valid phone sequences with breaks and pauses, and outputs a corresponding sequence of words. Draw the state diagram of L_2.

Mixed Bag

An HMM-based speech synthesis system can be described using the following steps:

1. Spectral feature and excitation features are extracted from a speech database
2. Context-dependent HMMs are trained on these features
3. These HMMs are clustered using a decision tree
4. Durations of the HMM models are explicitly modeled

At synthesis time, for a given text sequence, the decision tree yields the appropriate HMM state sequence which in turn determines the output spectral and excitation features (that are passed through a synthesis filter to produce speech). Say we want to add expressivity to the synthesized speech: i.e. we want to make the voice sound happy or sad, friendly or stern. Pick one of the above-mentioned steps from (A)-(D) you would modify to add expressivity and briefly justify your choice.

Mixed Bag

Find the probability, Pr(drank|Mohan), given the following bigram counts:

Mohan drank	10
drank coffee	1
Mohan coffee	10
drank Mohan	5
Mohan ate	10
drank water	20

Pr(drank|Mohan) = ______________________________

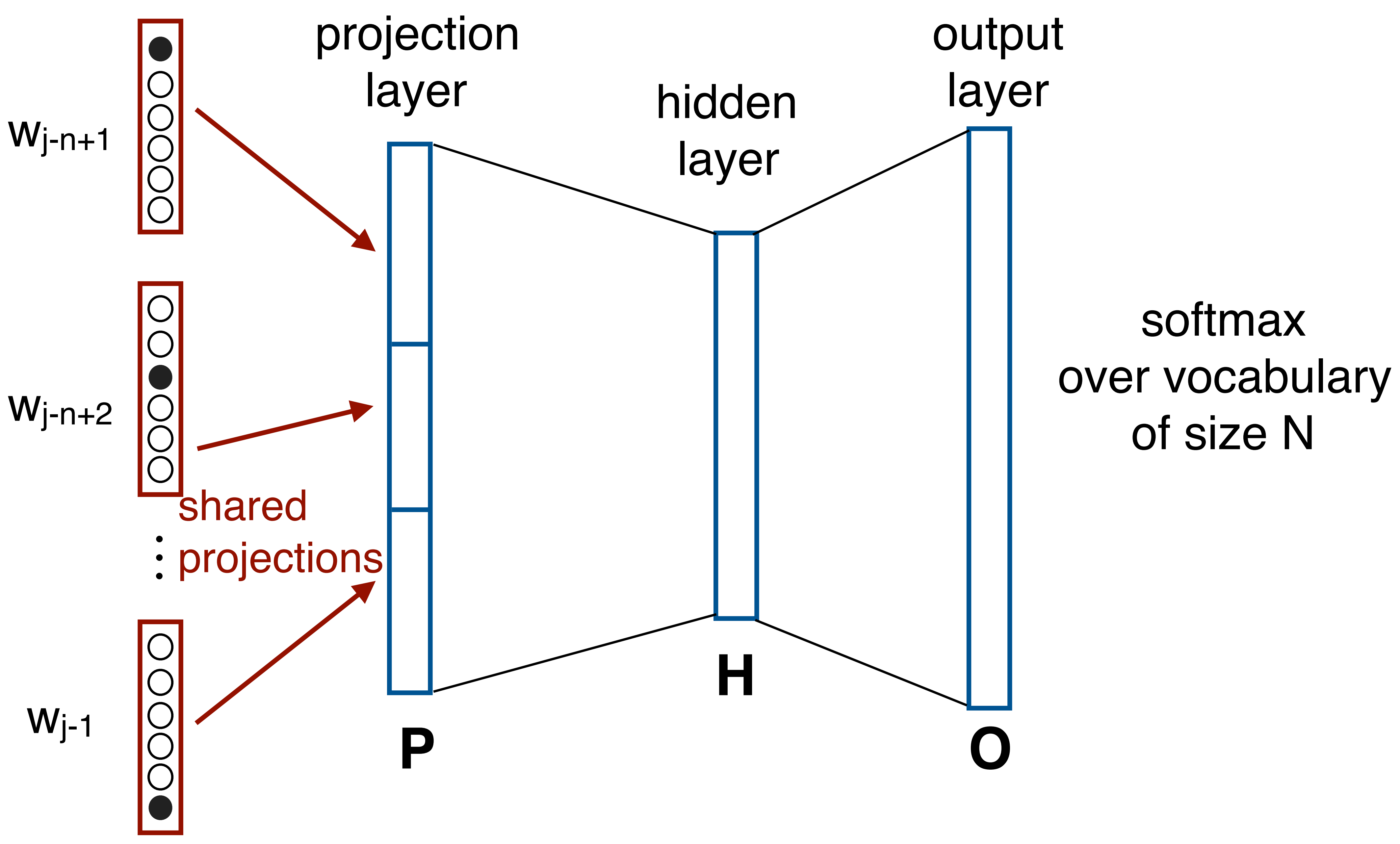
Say you have an n-gram distribution which is smoothed using add-α smoothing for some $\alpha > 0$. The entropy of the smoothed distribution is

(A) equal to (B) less than (C) greater than

the entropy of the original unsmoothed n-gram distribution. Pick one of (A), (B) or (C) and briefly justify your choice.

Mixed Bag

Recall neural network language models (NNLMs) as shown in the schematic diagram below. For a given context of fixed length, each word in the context (drawn from a vocabulary of size N) is projected onto a P dimensional projection layer using a common $N \times P$ projection matrix, that is shared across the different word positions in the context. The value of the ith node in the output layer corresponds directly to the probability of a word i given its context.



The complexity to calculate probabilities using this NNLM is quite high. Describe one main reason why this evaluation is very costly in processing time.

CTC Alignments

Given an input sequence $\mathbf{x}$ of length T and an output character sequence $\mathbf{y}$ of length N, the CTC objective function is given by:

$$P_{\text{CTC}}(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{a}:\mathcal{B}(\mathbf{a})=\mathbf{y}} P(\mathbf{a}|\mathbf{x})$$

where $\mathcal{B}$ maps a per-frame output sequence $\mathbf{a} = (a_1, \ldots, a_T)$ to a final output sequence $\mathbf{y} = (y_1, \ldots, y_N)$

Consider a different definition of $\mathcal{B}$ which first removes all occurrences of the blank symbol, and then compresses each run of an identical character to a run of length 1. Give an example of a sequence $\mathbf{y}$ such that there is no $\mathbf{a}$ with $\mathcal{B}(\mathbf{a}) = \mathbf{y}$, for this new $\mathcal{B}$. Briefly justify your answer.

CTC Alignments

Now suppose we would like to avoid the use of the blank symbol altogether. Towards this, we define a new $\mathcal{B}$ which works as follows. Given $\mathbf{a} = (a_1, \ldots, a_T)$, $\mathcal{B}$ defines the sequence $((c_1, \ell_1), (c_2, \ell_2), \ldots, (c_M, \ell_M))$ where $c_i \neq c_{i+1}$ and $\ell_i > 0$ for all i, and $\mathbf{a} = (\underbrace{c_1, \ldots, c_1}_{\ell_1 \text{ times}}, \underbrace{c_2, \ldots, c_2}_{\ell_2 \text{ times}}, \ldots, \underbrace{c_M, \ldots, c_M}_{\ell_M \text{ times}})$.

Then $\mathcal{B}$ calculates the average run length $\overline{\ell} = \frac{1}{M} \sum_{i=1}^{M} \ell_i$, and outputs

$$\mathbf{y} = (\underbrace{c_1, \ldots, c_1}_{k_1 \text{ times}}, \underbrace{c_2, \ldots, c_2}_{k_2 \text{ times}}, \ldots, \underbrace{c_M, \ldots, c_M}_{k_M \text{ times}})$$

where $k_i = \max\{1, \lfloor \ell_i / \overline{\ell} \rfloor\}$. Here, k_i is an estimate of how many times c_i needs to be repeated, depending on how ℓ_i compares with the average run length $\overline{\ell}$.

For example, $\mathcal{B}(a, a, b, b, b, b, b, b, b, b, c, c) = (a, b, b, c)$ because $\ell_1 = 2, \ell_2 = 8, \ell_3 = 2$ and therefore $k_1 = 1, k_2 = 2, k_3 = 1$.

Give an example of a sequence $\mathbf{y}$ such that there is no $\mathbf{a}$ with $\mathcal{B}(\mathbf{a}) = \mathbf{y}$, for this new $\mathcal{B}$.