Carnegie Mellon University

GENERATIVE ADVERSARIAL NETWORKS - PART II

11785- Introduction to Deep Learning

AKSHAT GUPTA

Spring 2021

Slides Inspired by Benjamin Striner

CONTENTS

Carnegie Mellon University

CONTENTS

Carnegie Mellon University

- GANS RECAP
- Understanding Training Issue in GANs
- GAN Training and Stabilization
- Wasserstein GANs
- GANstory - GAN Architectures

WHAT ARE GANS?

Carnegie Mellon University

Generative Adversarial Networks

Generative Models

We try to learn the underlying the distribution
from which our dataset comes from.
Eg: Variational AutoEncoders (VAE)

Neural Networks

Adversarial Training

GANS are made up of two competing networks (adversaries)
that are trying beat each other.

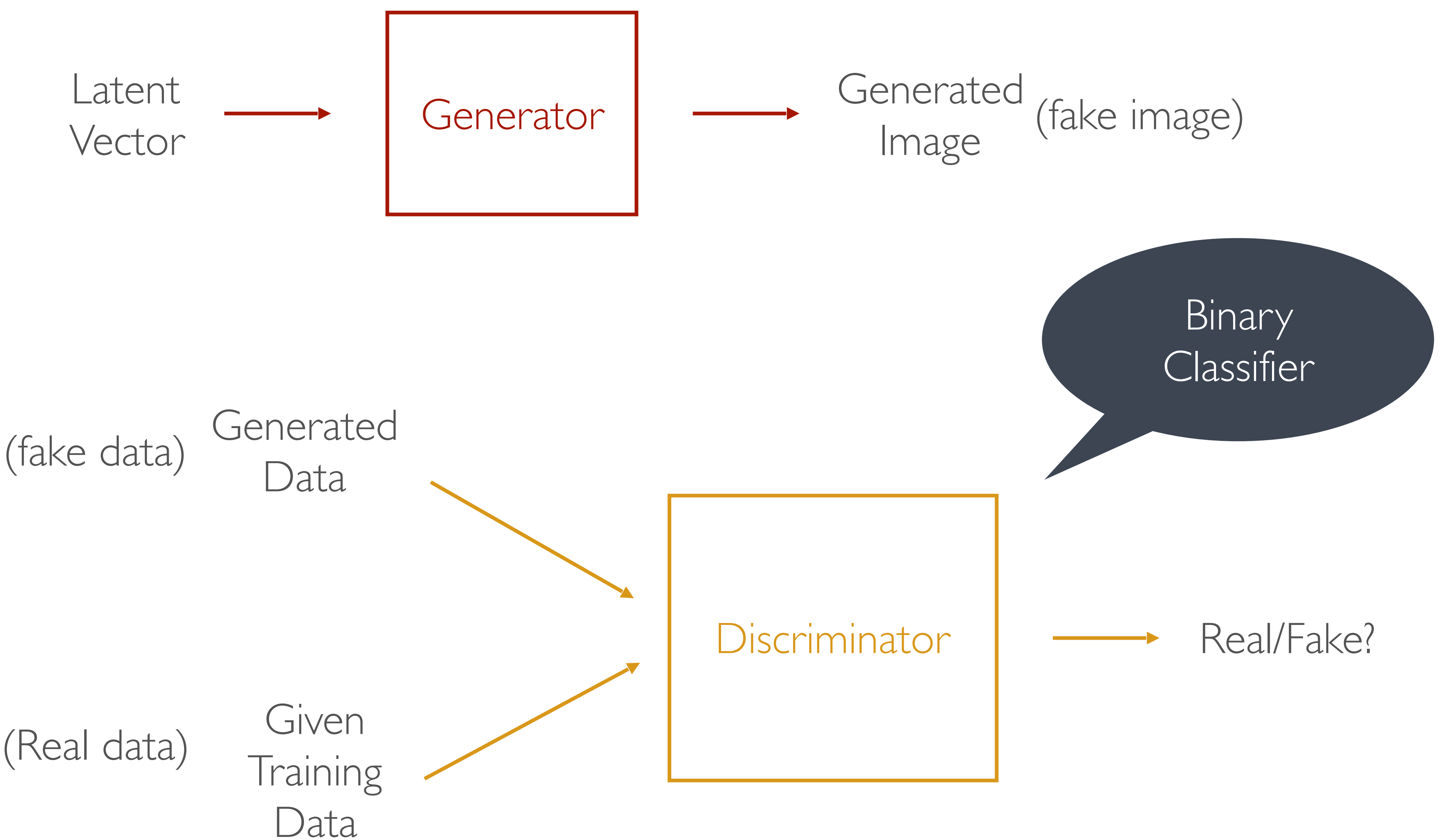
GOAL: Generate data from an unlabelled distribution.

WHAT CAN GANS DO?

- Data Augmentation
- Image-to-Image Translation
- Text-to-Image Synthesis
- Single Image Super Resolution

HOW TO TRAIN A GAN?

At t = 0,



HOW TO TRAIN A GAN?

Which network should I train first?

HOW TO TRAIN A GAN?

Which network should I train first?

Discriminator!

HOW TO TRAIN A GAN?

Which network should I train first?

Discriminator!

But with what training data?

HOW TO TRAIN A GAN?

Which network should I train first?

Discriminator!

But with what training data?

The Discriminator is a Binary classifier.
The Discriminator has two class - Real and Fake.
The data for Real class if already given: THE TRAINING DATASET
The data for Fake class? -> generate from the Generator

HOW TO TRAIN A GAN?

What's next? -> Train the Generator

But how? What's our training objective?

HOW TO TRAIN A GAN?

What's next? -> Train the Generator

But how? What's our training objective?

Generate images from the Generator
such that they are classified incorrectly by the Discriminator!

HOW TO TRAIN A GAN?

Discriminator

Generator

Step 1:
Train the Discriminator
<u>using the current ability</u>
of the Generator.

Step 2:
Train the Generator
<u>to beat</u>
the Discriminator.

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$

We represent the generator by $G(Z; \theta)$

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X \sim P_D$, what should happen to the value of $D(X)$?

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X \sim P_D$, what should happen to the value of $D(X)$?

It should be maximized!

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X \sim P_D$, what should happen to the value of $D(X)$?

It should be maximized!

$\Rightarrow D(X)$ should be maximized

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X \sim P_D$, what should happen to the value of $D(X)$?

It should be maximized!

$\Rightarrow D(X)$ should be maximized

$\Rightarrow \log(D(X))$ should be maximized

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X \sim P_D$, what should happen to the value of $D(X)$?

It should be maximized!

$\Rightarrow D(X)$ should be maximized

$\Rightarrow \log(D(X))$ should be maximized

$\Rightarrow E_{X \sim P_D}[\log(D(X))]$ should be maximized

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X \sim P_D$, what should happen to the value of $D(X)$?

It should be maximized!

$\Rightarrow D(X)$ should be maximized

$\Rightarrow \log(D(X))$ should be maximized

$\Rightarrow E_{X\sim P_D}[\log(D(X))]$ should be maximized

Chances of real data being called real.

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?

It should be minimized!

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?

It should be minimized!

$\Rightarrow D(X)$ should be minimized

$\Rightarrow \log(D(X))$ should be minimized

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?

It should be minimized!

$\Rightarrow D(X)$ should be minimized

$\Rightarrow \log(D(X))$ should be minimized

$\Rightarrow \log(1-D(X))$ should be maximized

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?

It should be minimized!

$\Rightarrow D(X)$ should be minimized

$\Rightarrow \log(D(X))$ should be minimized

$\Rightarrow \log(1-D(X))$ should be maximized

$\Rightarrow E_{X \sim P_G}[\log(1-D(X))]$ should be maximized

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?

It should be minimized!

$\Rightarrow D(X)$ should be minimized

$\Rightarrow \log(D(X))$ should be minimized

$\Rightarrow \log(1-D(X))$ should be maximized

$\Rightarrow E_{X\sim P_G}[\log(1-D(X))]$ should be maximized

$\Rightarrow E_{Z\sim P_z}[\log(1-D(G(Z)))]$ should be maximized

Chances of fake data being called fake.

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X \sim P_D$, what should happen to the value of $D(X)$?

$\Rightarrow E_{X \sim P_D}[\log(D(X))]$ should be maximized

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?

$\Rightarrow E_{Z \sim P_z}[\log(1 - D(G(Z)))]$ should be maximized

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

If $X \sim P_D$, what should happen to the value of $D(X)$?

$\Rightarrow E_{X\sim P_D}[\log(D(X))]$ should be maximized

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?

$\Rightarrow E_{Z\sim P_z}[\log(1-D(G(Z)))]$ should be maximized

⇒ The discriminator should maximize this sum:

$$V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$$

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

The discriminator maximizes this sum:

$$V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$$

Chances of real data being called real.

Chances of fake data being called fake.

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

Given that the discriminator maximizes this sum: $V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$

What should the generator do?

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

Given that the discriminator maximizes this sum: $V(D,G) = E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_Z}[\log(1 - D(G(Z)))]$

What should the generator do?

Generate images from the Generator
such that they are classified incorrectly by the Discriminator!

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

Given that the discriminator maximizes this sum: $V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$

What should the generator do?

Generate images from the Generator such that they are classified incorrectly by the Discriminator!

$\Rightarrow D(G(Z)))$ should be maximized

HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

Given that the discriminator maximizes this sum: $V(D,G) = E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_z}[\log(1 - D(G(Z)))]$

What should the generator do?

Generate images from the Generator such that they are classified incorrectly by the Discriminator!

$\Rightarrow D(G(Z)))$ should be maximized

$\Rightarrow \log(D(G(Z))))$ should be maximized

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

Given that the discriminator maximizes this sum: $V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$

What should the generator do?

Generate images from the Generator
such that they are classified incorrectly by the Discriminator!

$\Rightarrow D(G(Z)))$ should be maximized

$\Rightarrow \log(D(G(Z))))$ should be maximized

$\Rightarrow \log(1- \ D(G(Z))))$ should be minimized

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X;\theta)$
We represent the generator by $G(Z;\theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

Given that the discriminator maximizes this sum: $V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$

What should the generator do?

Generate images from the Generator
such that they are classified incorrectly by the Discriminator!

$\Rightarrow D(G(Z)))$ should be maximized

$\Rightarrow \log(D(G(Z))))$ should be maximized

$\Rightarrow \log(1- \; D(G(Z))))$ should be minimized

$\Rightarrow E_{Z\sim P_z}[\log(1-D(G(Z)))]$ should be minimized

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

Given that the discriminator maximizes this sum: $V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$

What should the generator do?

Generate images from the Generator
such that they are classified incorrectly by the Discriminator!

$\Rightarrow D(G(Z)))$ should be maximized

$\Rightarrow \log(D(G(Z))))$ should be maximized

$\Rightarrow \log(1- D(G(Z))))$ should be minimized

$\Rightarrow E_{Z\sim P_z}[\log(1-D(G(Z)))]$ should be minimized

Chances of fake data being called fake.

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

Given that the discriminator maximizes this sum: $V(D,G) = E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_z}[\log(1 - D(G(Z)))]$

What should the generator do?

Generate images from the Generator such that they are classified incorrectly by the Discriminator!

$\Rightarrow D(G(Z)))$ should be maximized

$\Rightarrow \log(D(G(Z))))$ should be maximized

$\Rightarrow \log(1 - D(G(Z))))$ should be minimized

$\Rightarrow E_{Z \sim P_z}[\log(1 - D(G(Z)))]$ should be minimized

$\Rightarrow E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_z}[\log(1 - D(G(Z)))]$ should be minimized

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

So, in your GAN formulation:

The discriminator maximizes this sum: $V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$

The generator minimizes this sum: $V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$

HOW TO TRAIN A GAN?

Carnegie Mellon University

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

P_D = actual data distribution
P_G = generated data distribution
P_Z = chosen prior in latent vector space

$D(X)$: Output of the discriminator / Probability that X came from actual data distribution P_D

$G(Z)$: Output of the generator/A point from the generated data distribution P_G

So, in your GAN formulation:

The discriminator maximizes this sum: $V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$

The generator minimizes this sum: $V(D,G) = E_{X\sim P_D}[\log(D(X))] + E_{Z\sim P_z}[\log(1-D(G(Z)))]$

Chances of real data being called real.

Chances of fake data being called fake.

ORIGINAL GAN FORMULATION

ORIGINAL GAN FORMULATION

The original GAN formulation is the following min-max game

$$\min_G \max_D V(D, G) = \mathbb{E}_X \log D(X) + \mathbb{E}_Z \log(1 - D(G(Z)))$$

- D wants $D(X) = 1$ and $D(G(Z)) = 0$
- G wants $D(G(Z)) = 1$

Carnegie Mellon University

THE OPTIMAL DISCRIMINATOR

P_D = actual data distribution
P_G = generated data distribution

D(X) = discriminator output

Objective: $\min_G \max_D V(D,G) = \mathbb{E}_X \log D(X) + \mathbb{E}_Z \log(1 - D(G(Z)))$

What is the optimal discriminator?

$$\begin{aligned} f &:= \mathbb{E}_{X\sim P_D} \log D(X) + \mathbb{E}_{X\sim P_G} \log(1 - D(X)) \\ &= \int_X [P_D(X)\log D(X) + P_G(X)\log(1 - D(X))]\, dX \end{aligned}$$

$$\frac{\partial f}{\partial D(X)} = \frac{P_D(X)}{D(X)} - \frac{P_G(X)}{1 - D(X)} = 0$$

$$\frac{P_D(X)}{D(X)} = \frac{P_G(X)}{1 - D(X)}$$

$$(1 - D(X))P_D(X) = D(X)P_G(X)$$

$$D(X) = \frac{P_D(X)}{P_G(X) + P_D(X)}$$

Carnegie Mellon University

THE OPTIMAL GENERATOR

P_D = actual data distribution
P_G = generated data distribution

D(X) = discriminator output
G(Z) = generator output

Objective: $\min_G \max_D V(D, G) = \mathbb{E}_X \log D(X) + \mathbb{E}_Z \log(1 - D(G(Z)))$

Generator wants to minimize this!

$$:= \mathbb{E}_{X \sim P_D} \log D(X) + \mathbb{E}_{X \sim P_G} \log(1 - D(X))$$

$$= \mathbb{E}_{P_D} \log \frac{P_D(X)}{P_G(X) + P_D(X)} + \mathbb{E}_{P_G} \log \frac{P_D(X)}{P_G(X) + P_D(X)}$$

$$= JSD(P_D | P_G) - \log 4$$

THE OPTIMAL GENERATOR

What is the optimal generator?

$$\min_G JSD(P_D \| P_G) - \log 4$$

Minimize the Jensen-Shannon divergence between the real and generated distributions (make the distributions similar)

MIN-MAX STATIONARY POINT

- Stationary points need not be stable (depends on the exact GANs formulation and other factors)

CONTENTS

Carnegie Mellon University

Carnegie Mellon University

WHY IS THERE NO STATIC OPTIMAL DISCRIMINATOR?

$$\min_G \max_D V(D, G) = \mathbb{E}_X \log D(X) + \mathbb{E}_Z \log(1 - D(G(Z)))$$

- Discriminator indicates the direction in which generator should move relative to the current generator
- For a given fixed discriminator, the optimal generator outputs argmax D(X) for all z~Z
- Cannot train generator without training discriminator first

CAUSES OF OPTIMIZATION ISSUES

- Simultaneous updates require a careful balance between players
- Stationary point exists but there's no guarantee of reaching it
- If discriminator is undertrained, it guides the generator in the wrong direction
- If discriminator is overtrained, it is too hard and generator cannot make much progress

FACTORS AFFECTING ADVERSARIAL BALANCE

- Different optimizers, learning rates, batch size
- Different architectures, depths, number of parameters
- Training discriminator and generator for different number of iterations

Carnegie Mellon University

ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

CASE - 1: I play rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

- What is your best strategy?
- What is your probability of winning?

ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

CASE - II: I play rock-paper-scissors with a probability of

(0.33, 0.33, 0.33)

- What is your optimal strategy?
- What is your probability of winning?

ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

- GLOBAL OPTIMUM : Both players play uniformly with (0.33, 0.33, 0.33)

ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

- If player B optimizes all the way, its optimal strategy is always paper (0, 1, 0)

ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

- If player B optimizes all the way, its optimal strategy is always paper (0,1,0)
- Now player A should play only scissors (0,0,1)

ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

- If player B optimizes all the way, its optimal strategy is always paper (0,1,0)
- Now player A should play only scissors (0,0,1)
- Now player B should only play rock (1,0,0)

ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

- If player B optimizes all the way, its optimal strategy is always paper (0, 1, 0)
- Now player A should play only scissors (0, 0, 1)
- Now player B should only play rock (1, 0, 0)
- Now player A should only play paper (0, 1, 0)

ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

- If player B optimizes all the way, its optimal strategy is always paper.
- Now player A should play only scissors
- Now player B should only play rock
- Now player A should only play paper
- …….

TRAINING ISSUES IN GAS

- Oscillations
- Mode Collapse : Generates a small subspace but does not cover the entire distribution (https://www.youtube.com/watch?v=ktxhiKhWoEE)

CONTENTS

- GANs Recap
- Understanding Training Issue in GANs
- GAN TRAINING AND STABILIZATION
- Wasserstein GANs
- GANstory - GAN Architectures

IMPROVED TECHNIQUES FOR TRAINING GANS (2016)

A collection of interesting techniques and experiments

- Feature Matching
- Minibatch Discrimination
- Historical Averaging
- One-sided Label Smoothing
- Virtual Batch Normalization

FEATURE MATCHING

Carnegie Mellon University

Statistics of generated images should match statistics of real images

- Discriminator produces multidimensional output, a "statistic" of the data
- Generator trained to minimize L_2 between real and generated data
- Discriminator trained to maximize L_2 between real and generated data

$$\|\mathbb{E}_X D(X) - \mathbb{E}_Z D(G(Z))\|_2^2$$

MINIBATCH DISCRIMINATION

Discriminator can look at multiple inputs at once and decide if those inputs come from the real or generated distribution

- GANs frequently collapse to a single point
- Discriminator needs to differentiate between two distributions
- Easier task if looking at multiple samples

HISTORICAL AVERAGING

Dampen oscillations by encouraging updates to converge to a mean

- GANs frequently create a cycle or experience oscillations
- Add a term to reduce oscillations that encourages the current parameters to be near a moving average of the parameters

$$\left\|\theta - \frac{1}{t}\sum_{i}^{t}\theta_i\right\|_2^2$$

ONE-SIDED LABEL SMOOTHING

Don't over-penalize generated images

- Label smoothing is a common and easy technique that improves performance across many domains
  - Sigmoid tries hard to saturate to 0 or 1 but can never quite reach that goal
  - Provide targets that are ϵ or $1 - \epsilon$ so the sigmoid doesn't saturate and overtrain
- Experimentally, smooth the real targets but do not smooth the generated targets when training the discriminator

VIRTUAL BATCH NORMALIZATION

Use batch normalization to accelerate convergence

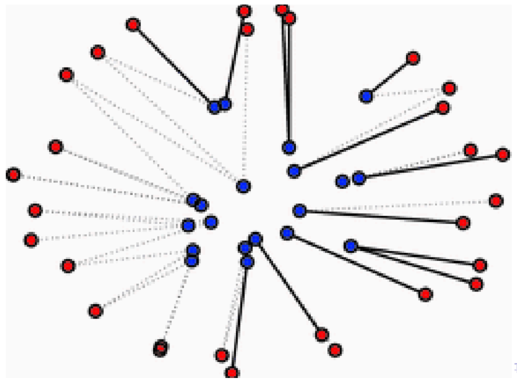
- Batch normalization accelerates convergence
- However, hard to apply in an adversarial setting
- Collect statistics on a fixed batch of real data and use to normalize other data

CONTENTS

Carnegie Mellon University

WASSERSTEIN DISTANCE

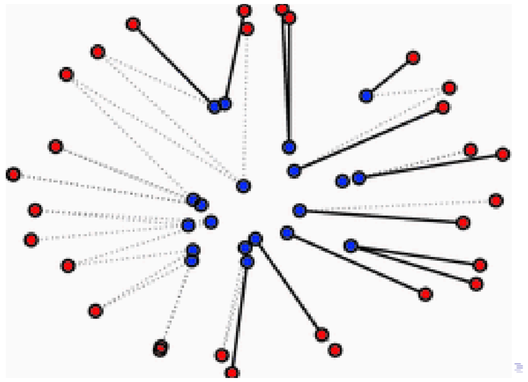
- The distance between probability distributions
- Intuitively, each distribution is viewed as a unit amount of earth (soil)
- The total Σ mass × mean distance required to transform one distribution to another
- Also called earth mover's distance



Red points, Blue points represent two different distributions.

WASSERSTEIN DISTANCE

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \big[\, \|x - y\| \, \big]$$



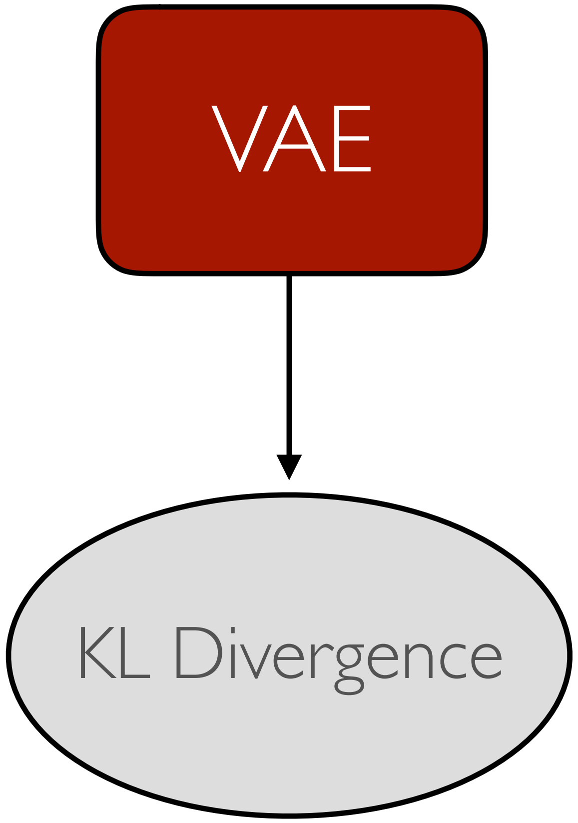
Red points, Blue points represent two different distributions.

THE GAME OF DISTANCE MEASURES

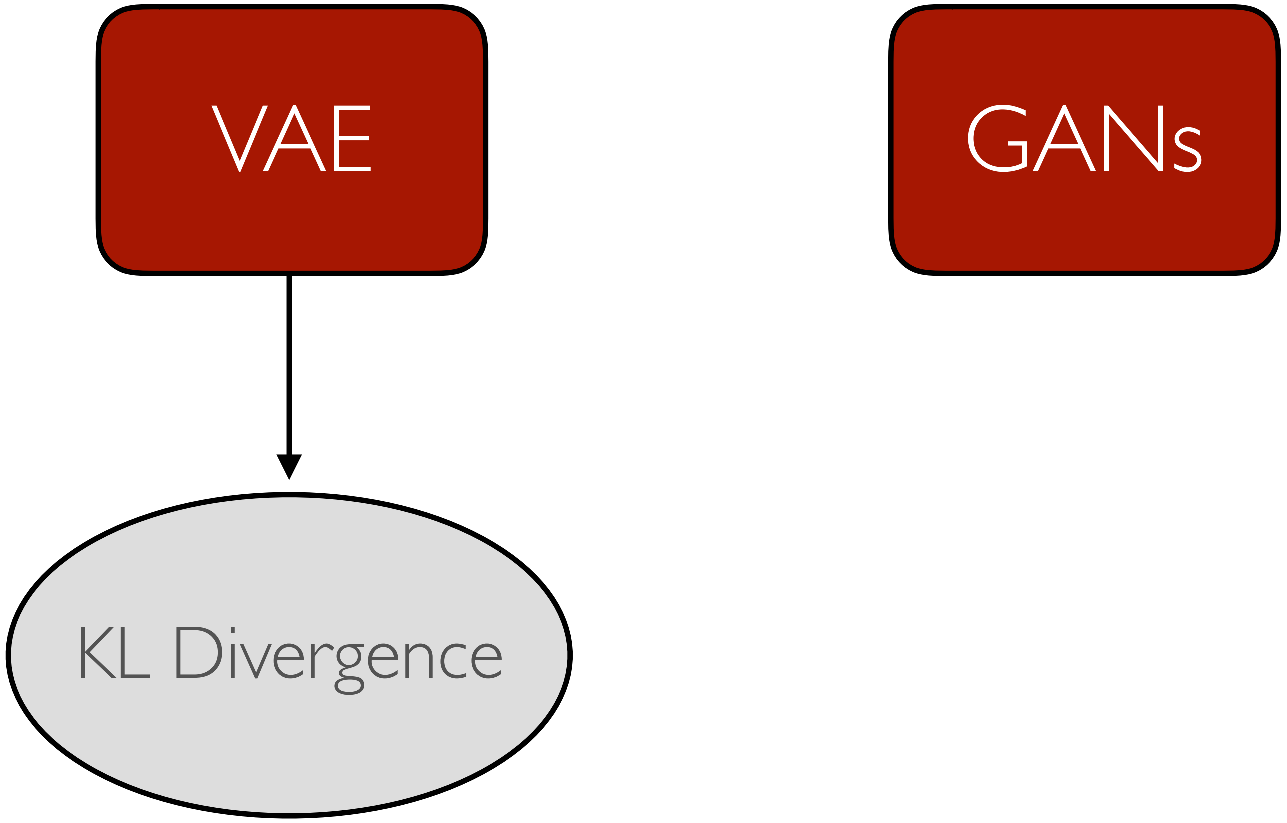
THE GAME OF DISTANCE MEASURES

VAE

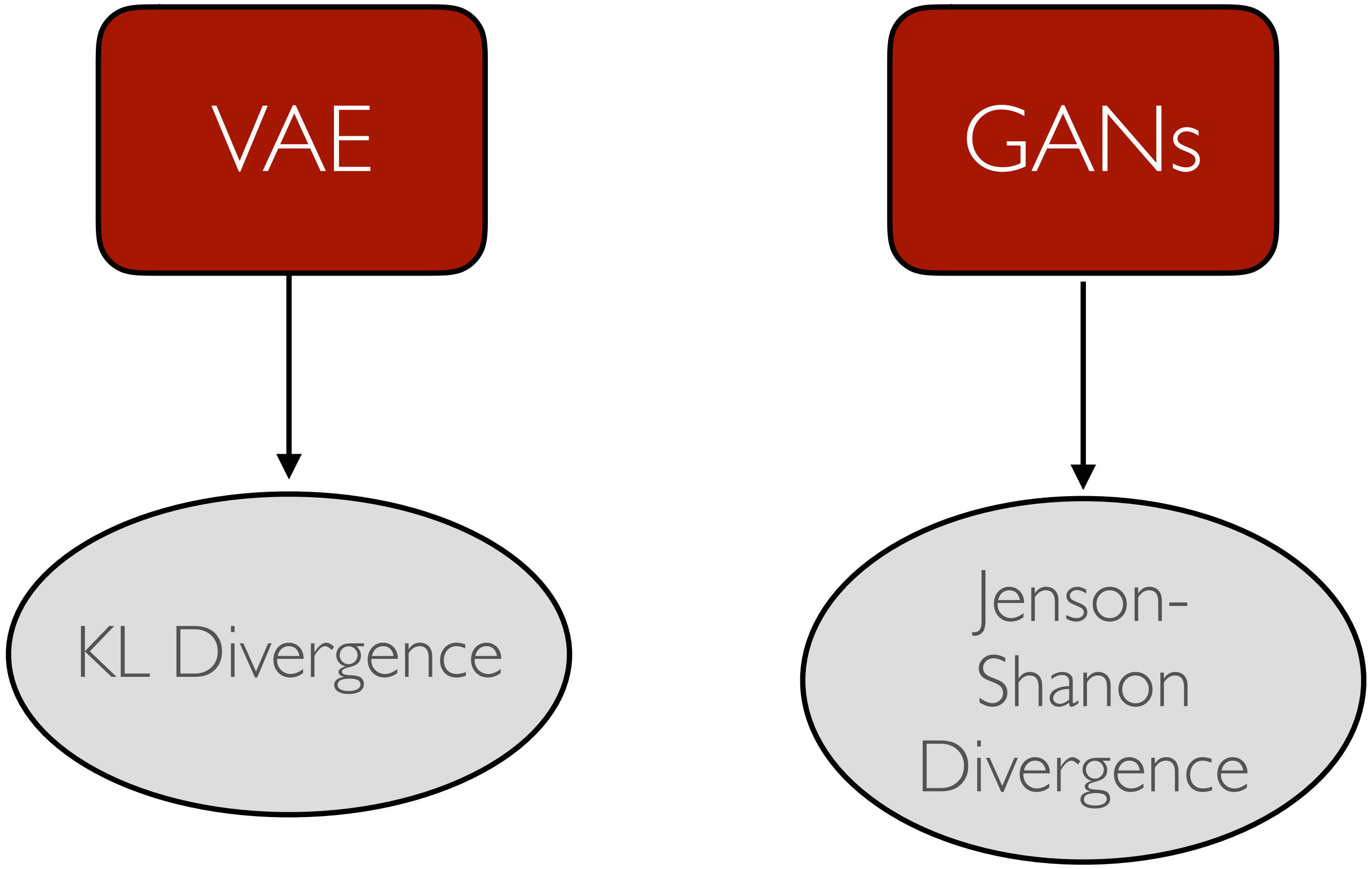
THE GAME OF DISTANCE MEASURES



THE GAME OF DISTANCE MEASURES

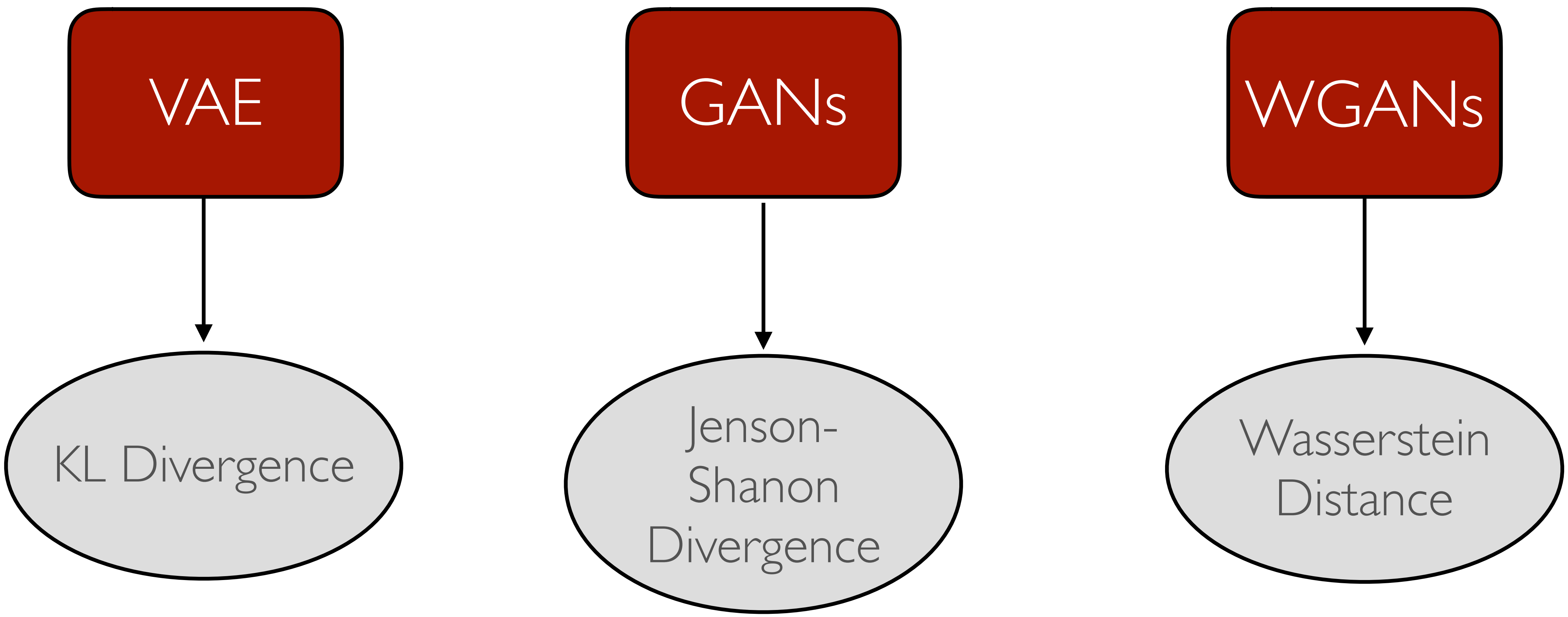


THE GAME OF DISTANCE MEASURES



Carnegie Mellon University

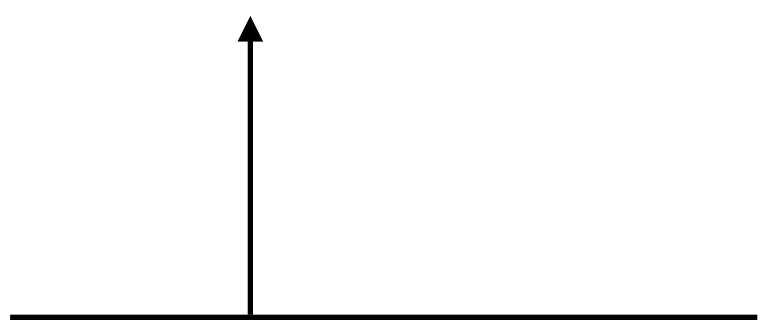
THE GAME OF DISTANCE MEASURES



KL-DIVERGENCE

$$KL(p\|q) = \int_x p(x)\log\frac{p(x)}{q(x)}$$

P(X)



X = 0

Q(X)

X = 1

Let θ be the distance between the two peaks of the distribution

If $\theta \neq 0$, $KL(P||Q) = 1\ \log(1/0) = \infty$

If $\theta = 0$, $KL(P||Q) = 1\ \log(1/1) = 0$

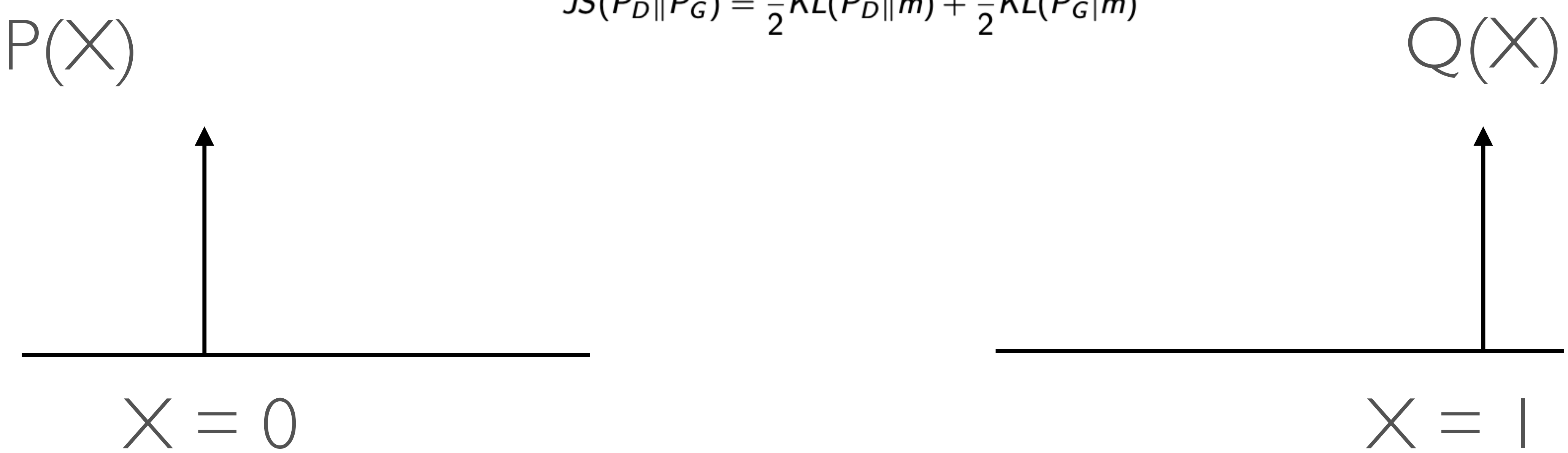
Not differentiable w.r.t θ

Carnegie Mellon University

JENSON-SHANON DIVERGENCE

$$m(X) = \frac{P_D + P_G}{2}$$

$$JS(P_D \| P_G) = \frac{1}{2} KL(P_D \| m) + \frac{1}{2} KL(P_G | m)$$



Let θ be the distance between the two peaks of the distribution

If θ ≠ 0, JSD(P||Q) = 0.5 * (1 log(1 / 0 . 5) + 1 log(1 / 0 . 5)) = log4

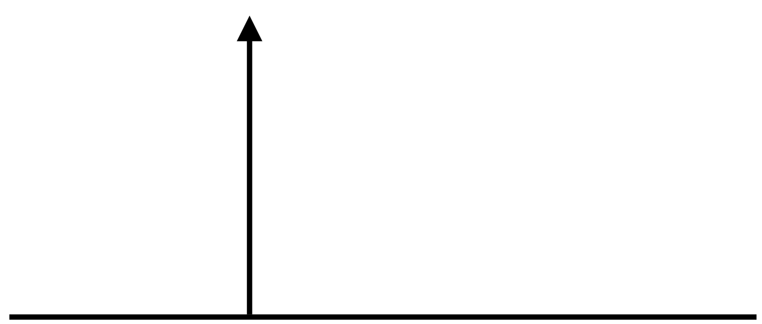
If θ = 0, JSD(P||Q) = 0.5 * (1 log(1 / 1) + 1 log(1 / 1)) = 0

Not differentiable w.r.t θ

WASSERSTEIN DISTANCE

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

P(X)



X = 0

Q(X)

X = 1

W(P,Q) = | θ |

Differentiable w.r.t θ !!

Carnegie Mellon University

JSD VS WASSERSTEIN (EM)

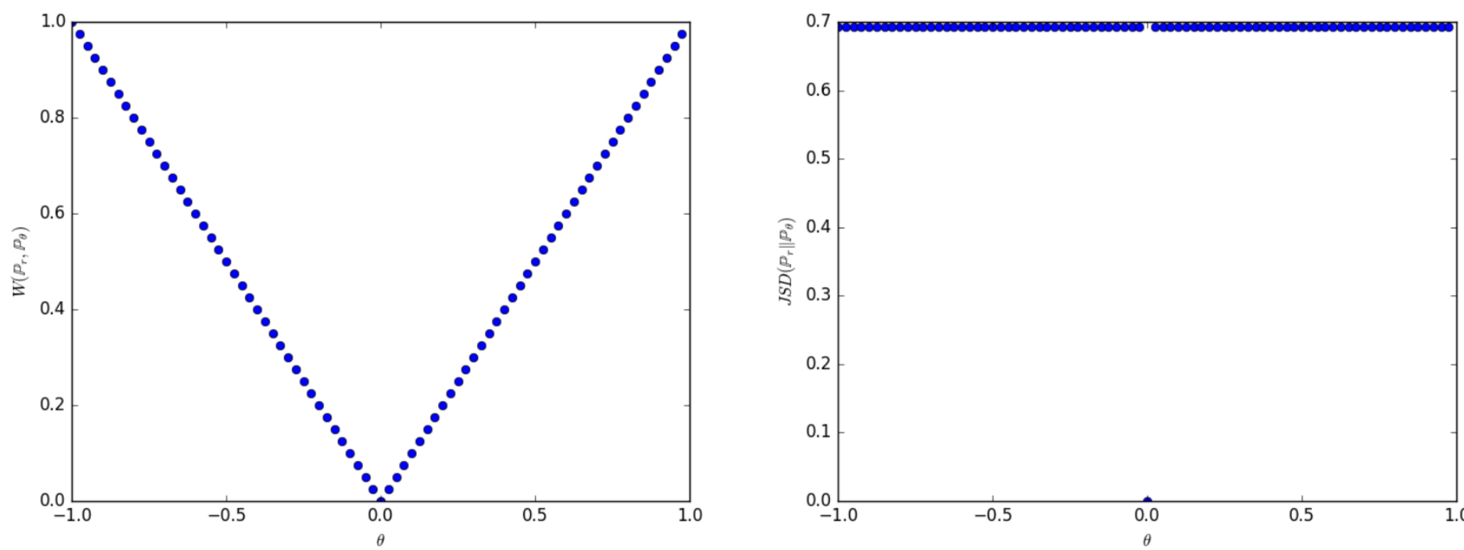


Figure 1: These plots show $\rho(\mathbb{P}_\theta, \mathbb{P}_0)$ as a function of θ when ρ is the EM distance (left plot) or the JS divergence (right plot). The EM plot is continuous and provides a usable gradient everywhere. The JS plot is not continuous and does not provide a usable gradient.

Carnegie Mellon University

WASSERSTEIN (EM) VS JSD

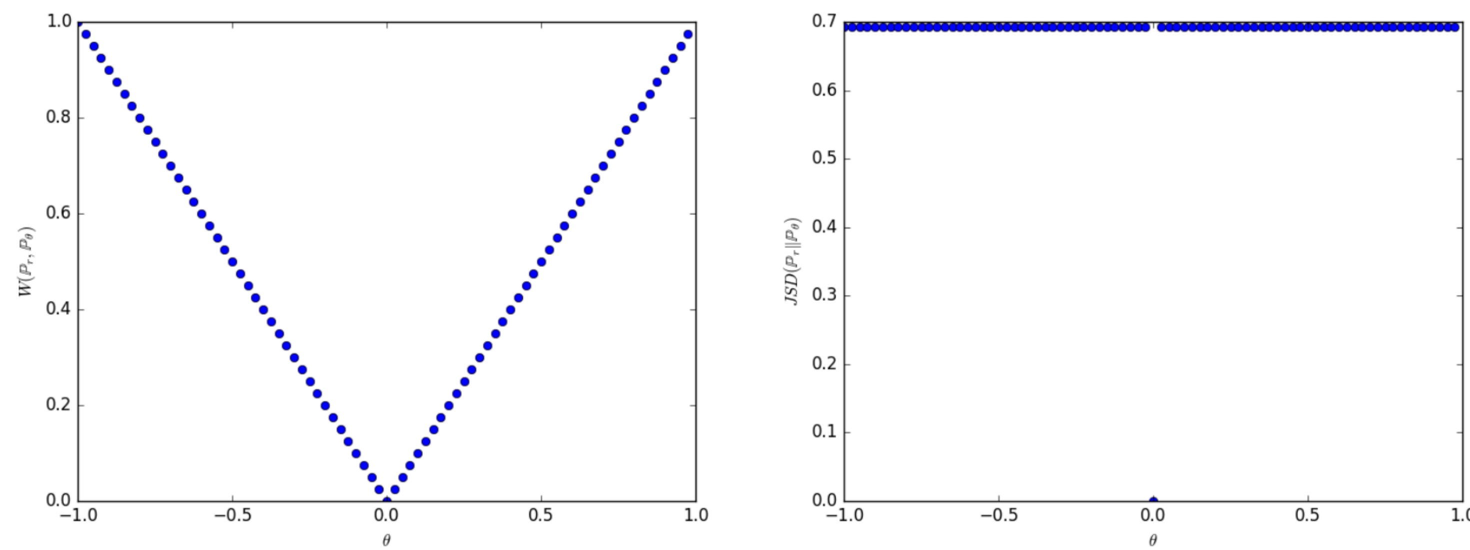


Figure 1: These plots show $\rho(\mathbb{P}_\theta, \mathbb{P}_0)$ as a function of θ when ρ is the EM distance (left plot) or the JS divergence (right plot). The EM plot is continuous and provides a usable gradient everywhere. The JS plot is not continuous and does not provide a usable gradient.

- Distance value is not constant for non-overlapping distributions
- Differentiable w.r.t θ

WGAN

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \big[D(\boldsymbol{x}) \big] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \big[D(\tilde{\boldsymbol{x}})) \big]$$

Kantorovich-Rubinstein duality

WGAN

Carnegie Mellon University

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \left[D(\boldsymbol{x}) \right] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \left[D(\tilde{\boldsymbol{x}})) \right]$$

Kantorovich-Rubinstein duality

D should be a 1-Lipschitz function

WGAN

Carnegie Mellon University

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \big[D(\boldsymbol{x}) \big] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \big[D(\tilde{\boldsymbol{x}})) \big]$$

Kantorovich-Rubinstein duality

D should be a 1-Lipschitz function

Weight clipping:

- Restrict weights between [-c, c]

WGAN-GP

$$L = \underbrace{\underset{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g}{\mathbb{E}}\left[D(\tilde{\boldsymbol{x}})\right] - \underset{\boldsymbol{x} \sim \mathbb{P}_r}{\mathbb{E}}\left[D(\boldsymbol{x})\right]}_{\text{Original critic loss}} + \underbrace{\lambda \underset{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}}{\mathbb{E}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}.$$

A function is 1-Lipschitz if its gradients are at most 1 everywhere.

WGAN-GP

$$L = \underbrace{\underset{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g}{\mathbb{E}} [D(\tilde{\boldsymbol{x}})] - \underset{\boldsymbol{x} \sim \mathbb{P}_r}{\mathbb{E}} [D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \underset{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}}{\mathbb{E}} \left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2 \right]}_{\text{Our gradient penalty}} .$$

A function is 1-Lipschitz if its gradients are at most 1 everywhere.

Gradient penalty introduces a softer constraint on gradients

CONTENTS

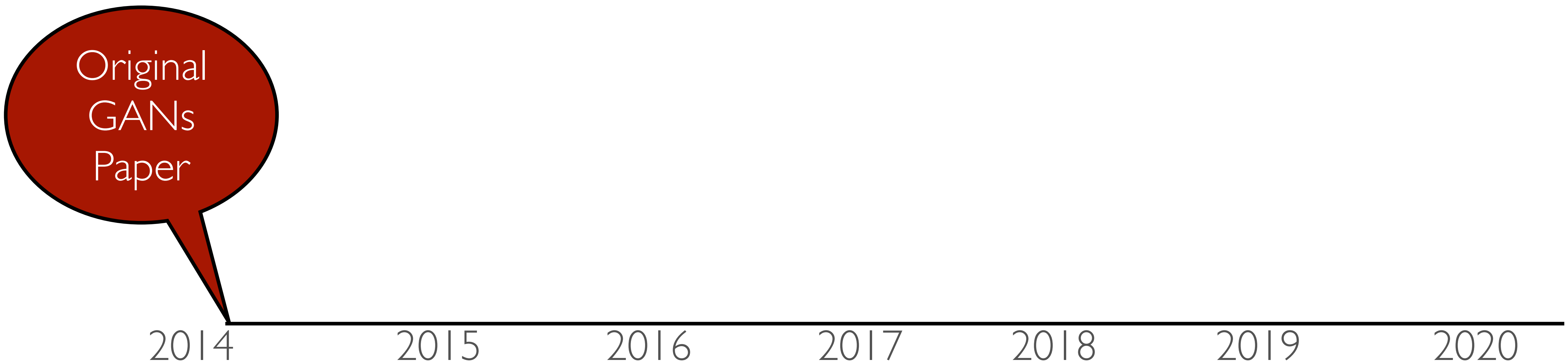
- GANs Recap
- Understanding Training Issue in GANs
- GAN Training and Stabilization
- Wasserstein GANs
- GANSTORY - GAN ARCHITECTURES

GANS PROGRESSION

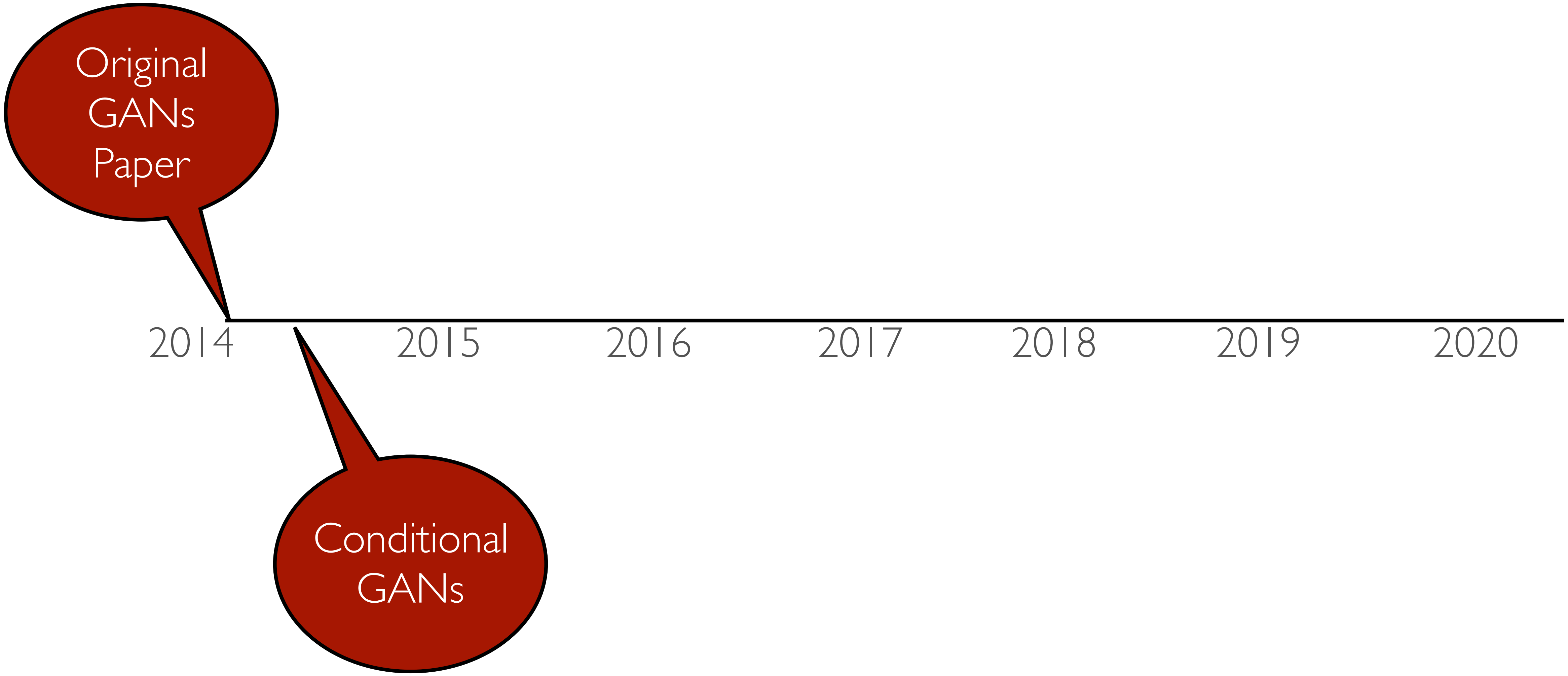
- Better quality
- High Resolution

https://twitter.com/goodfellow_ian/status/1084973596236144640?lang=en

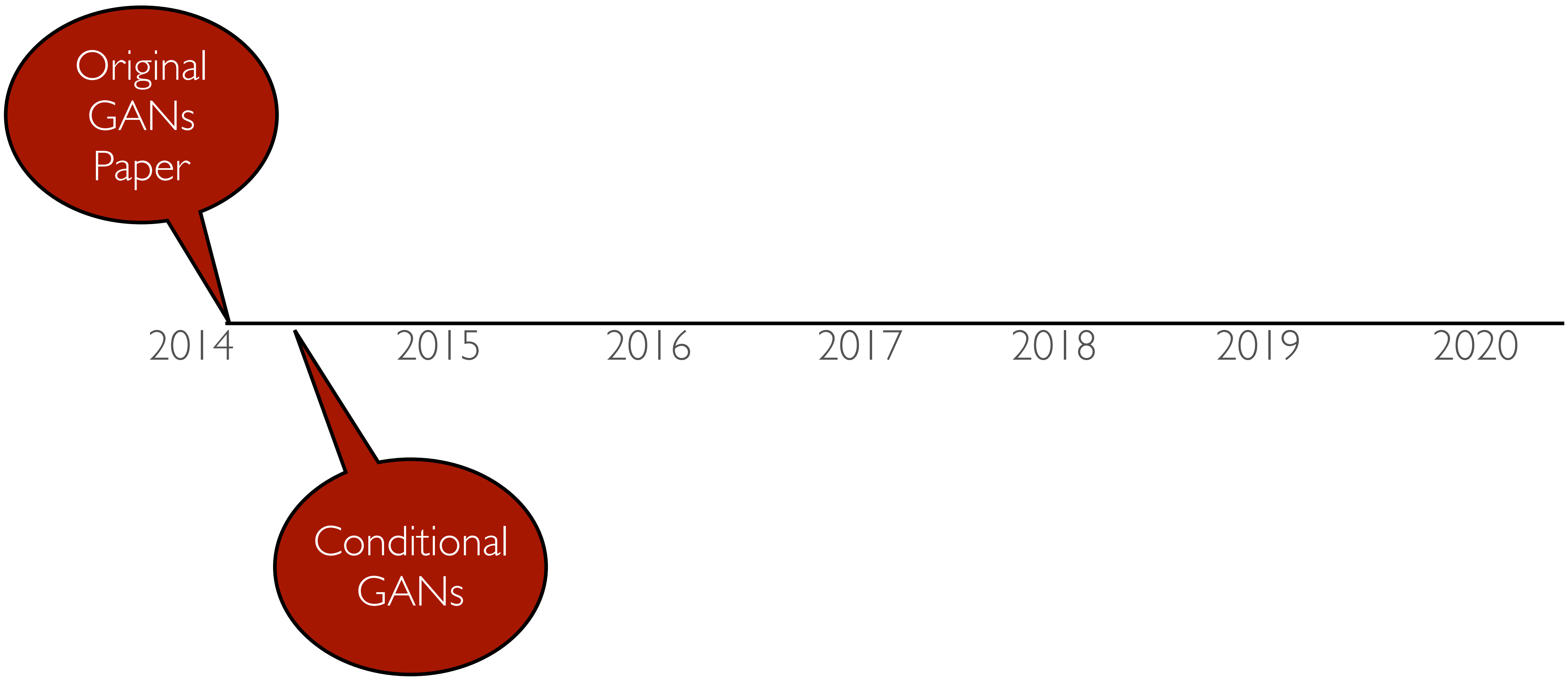
GANS PROGRESSION



GANS PROGRESSION

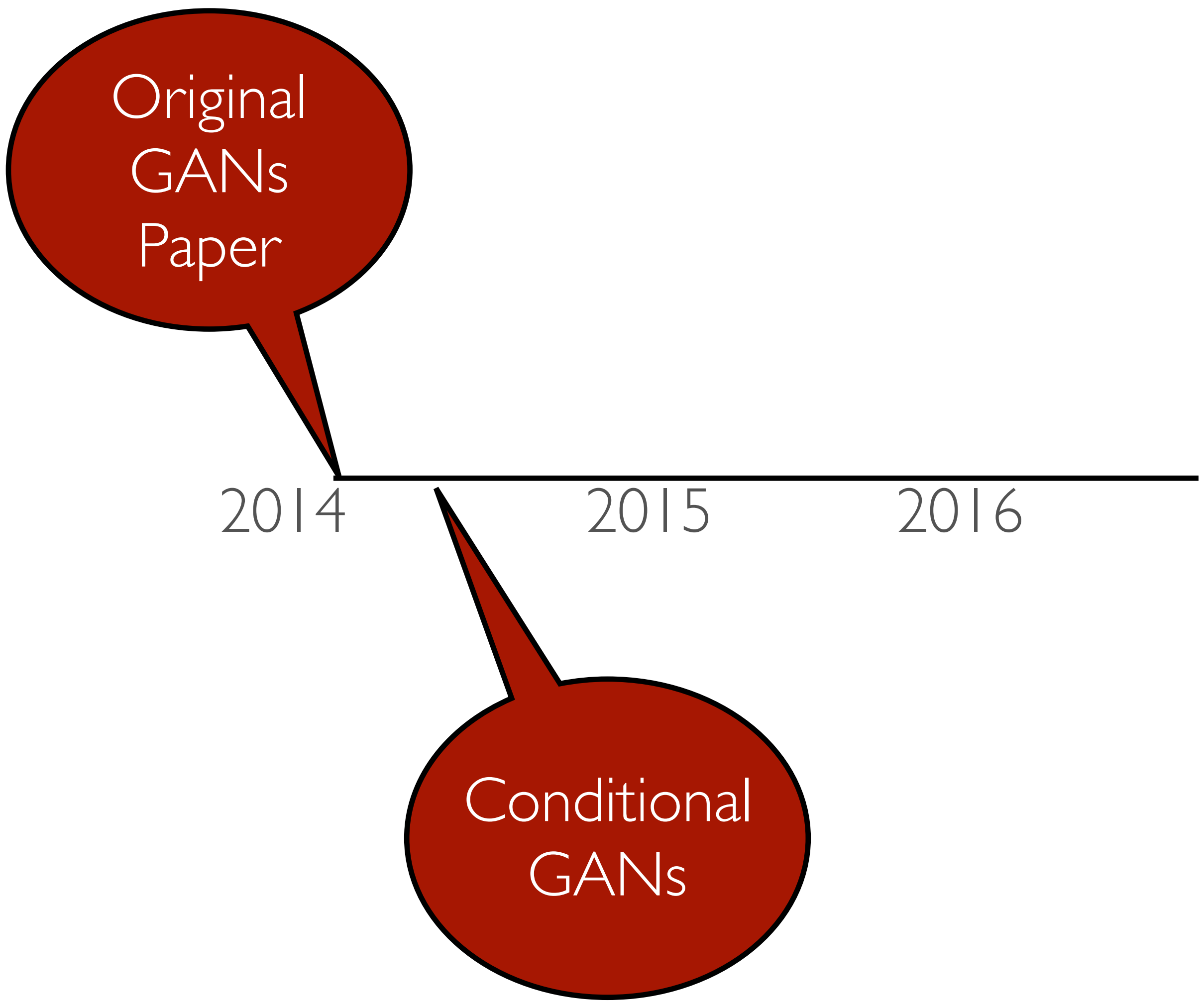


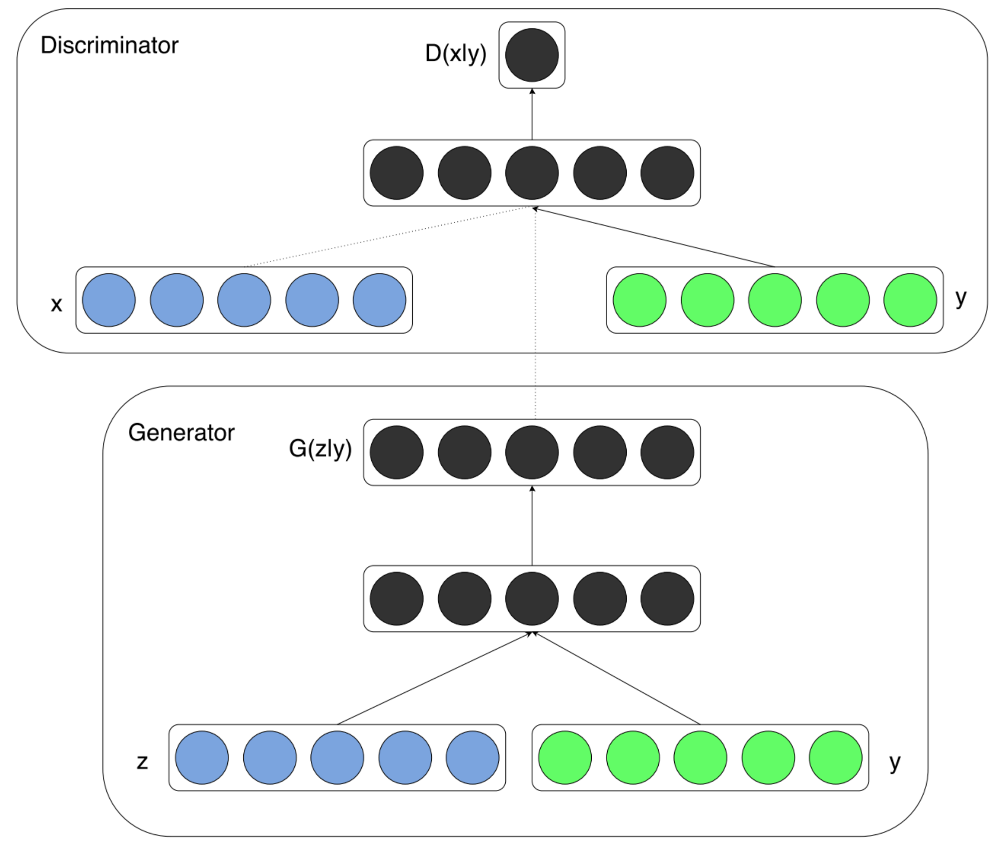
GANS PROGRESSION



$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))].$$

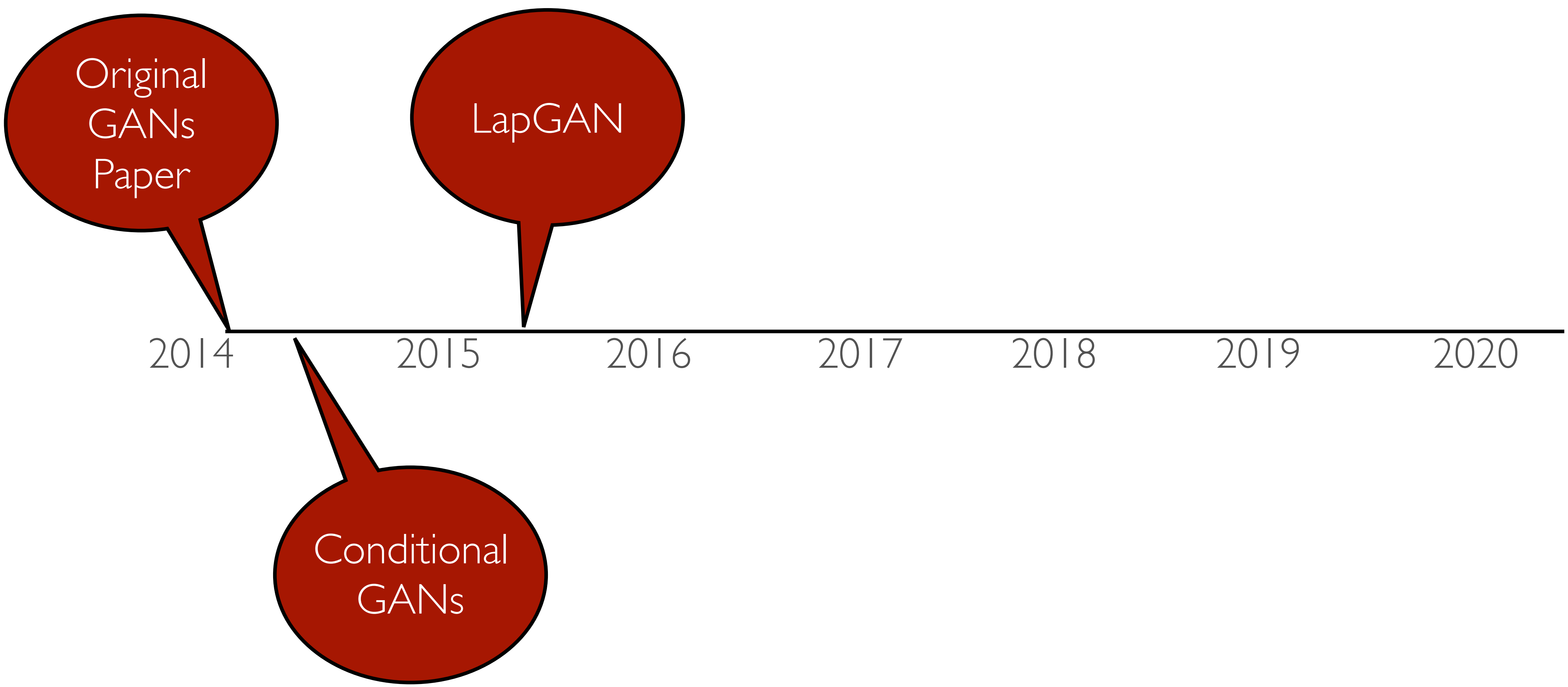
GANS PROGRESSION



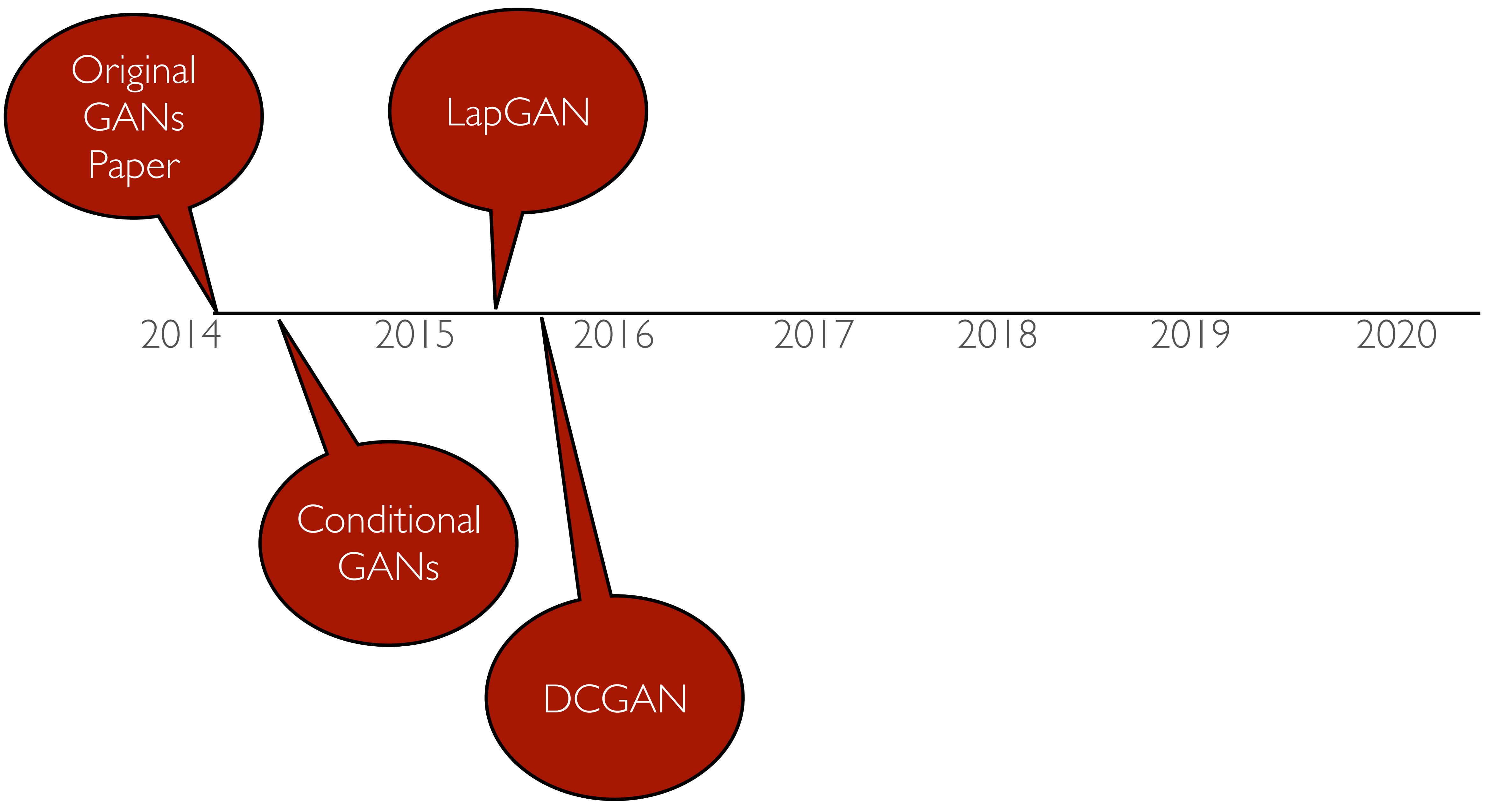


$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_{z}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))].$$

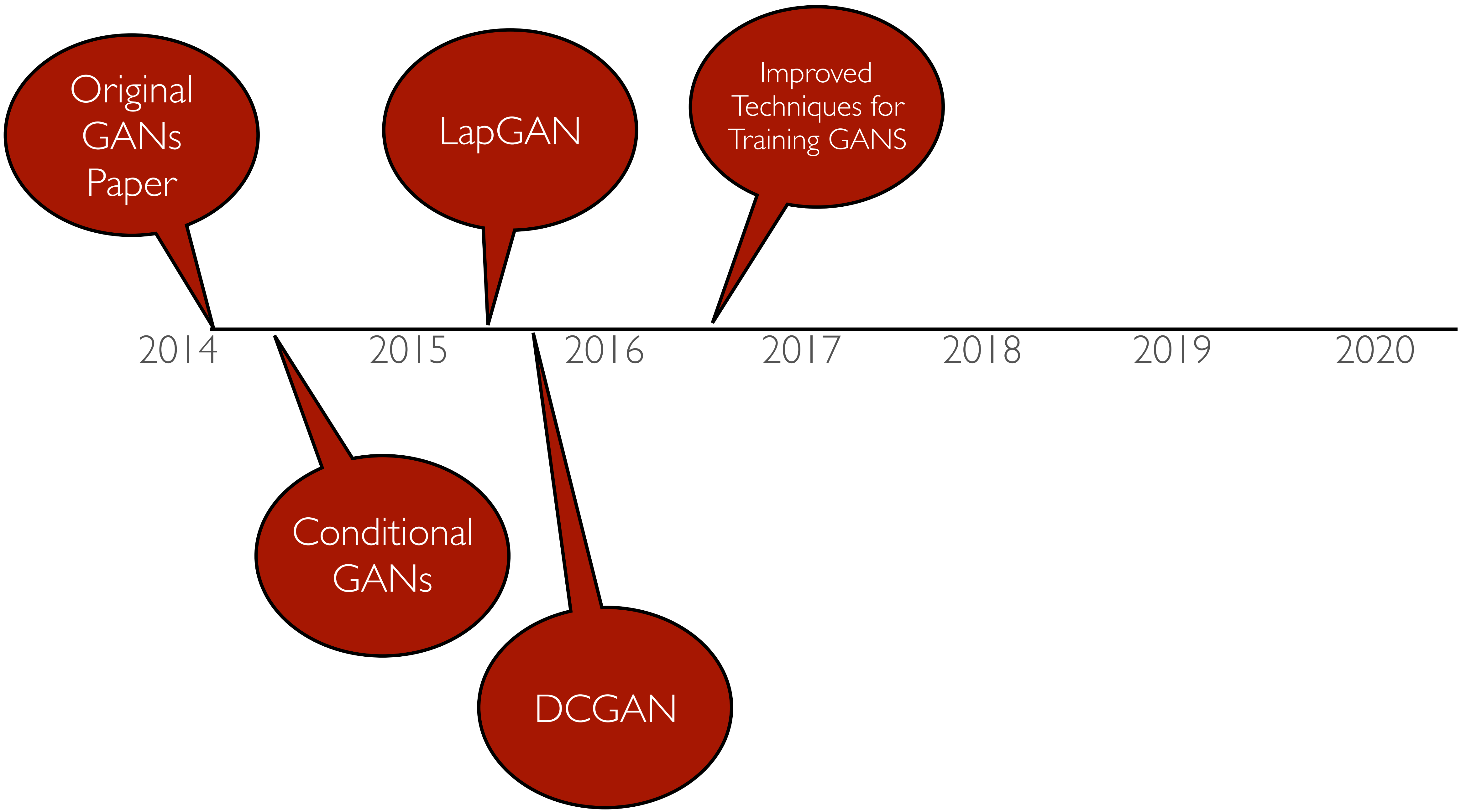
GANS PROGRESSION



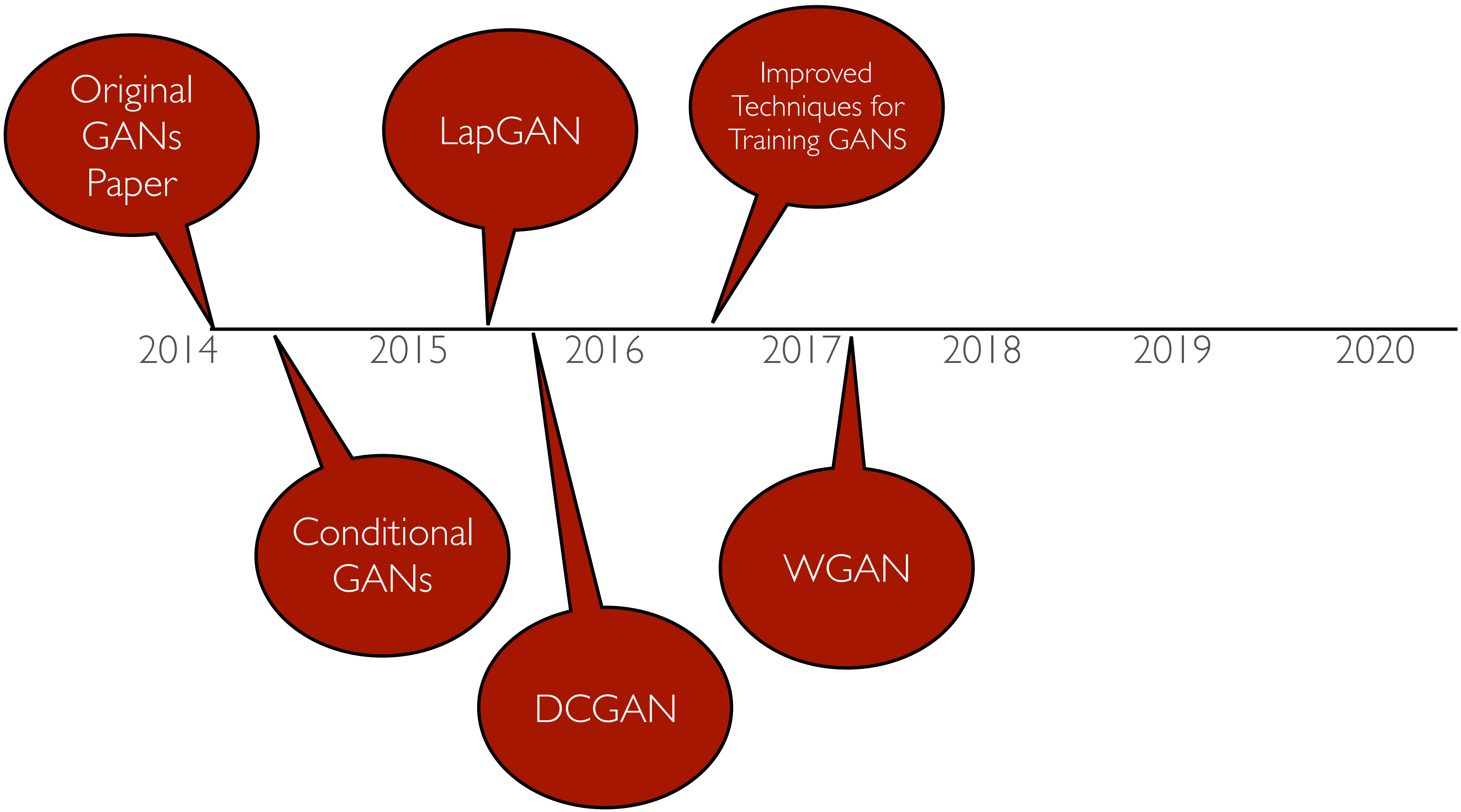
GANS PROGRESSION



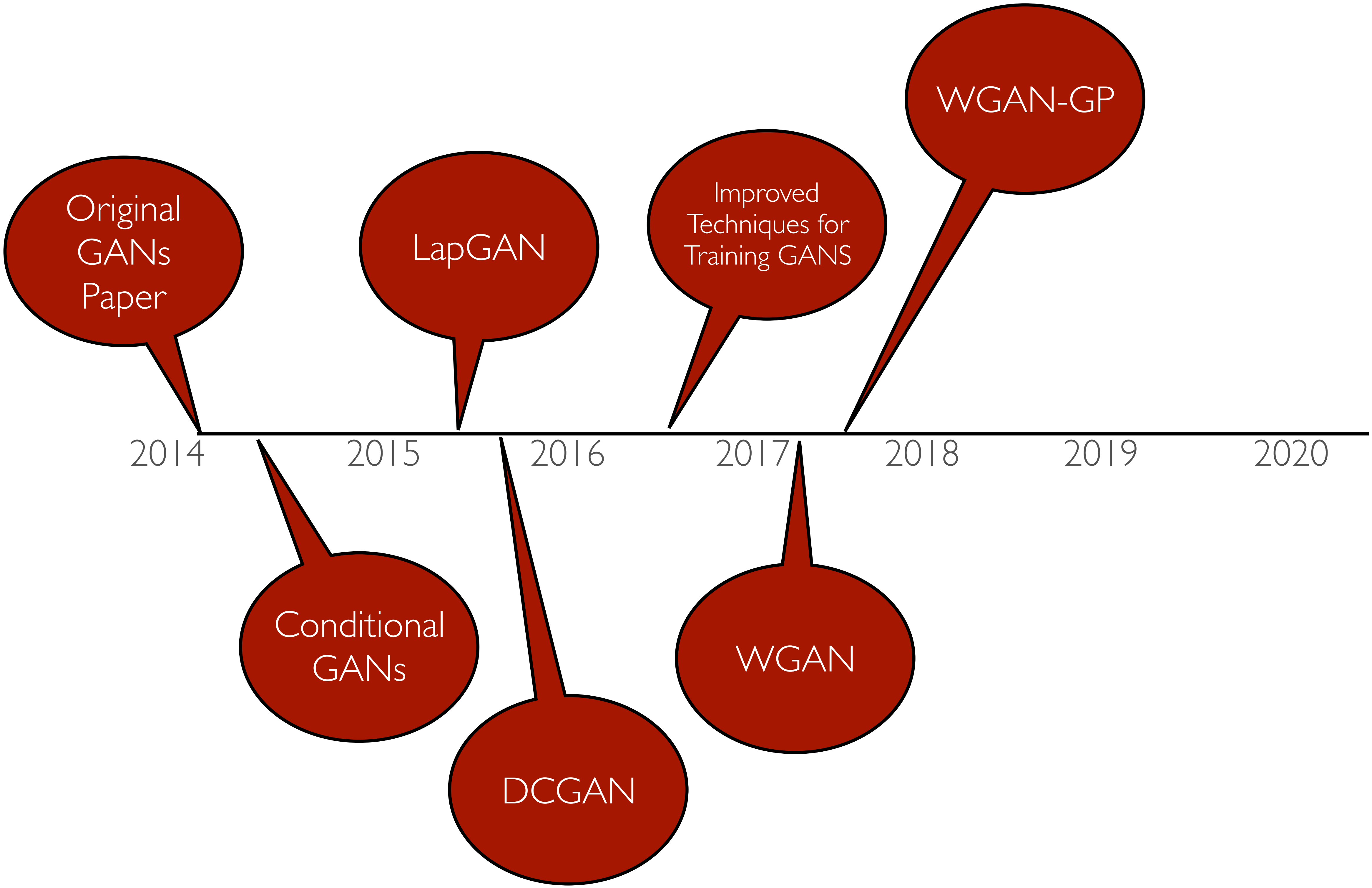
GANS PROGRESSION



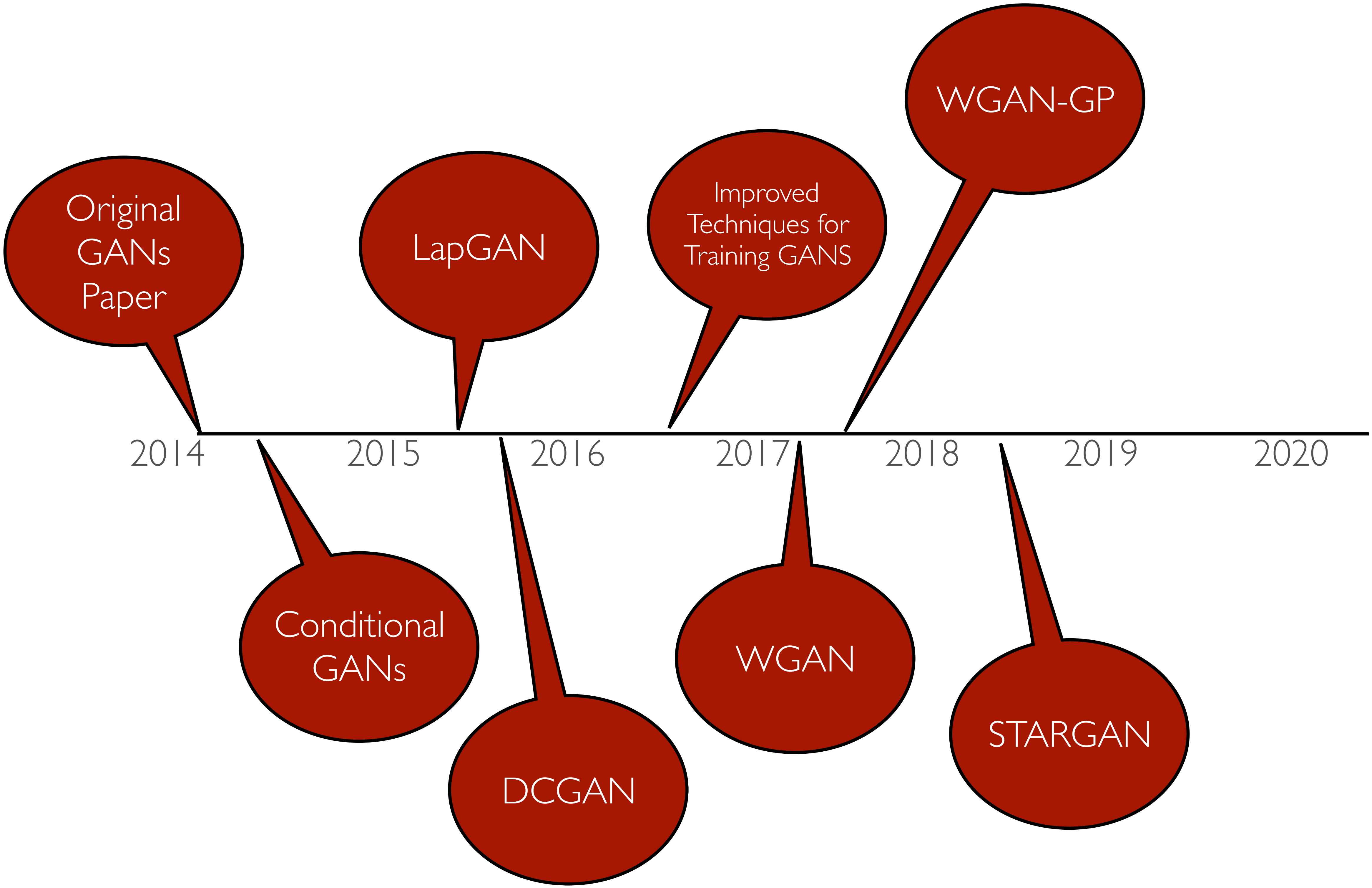
GANS PROGRESSION



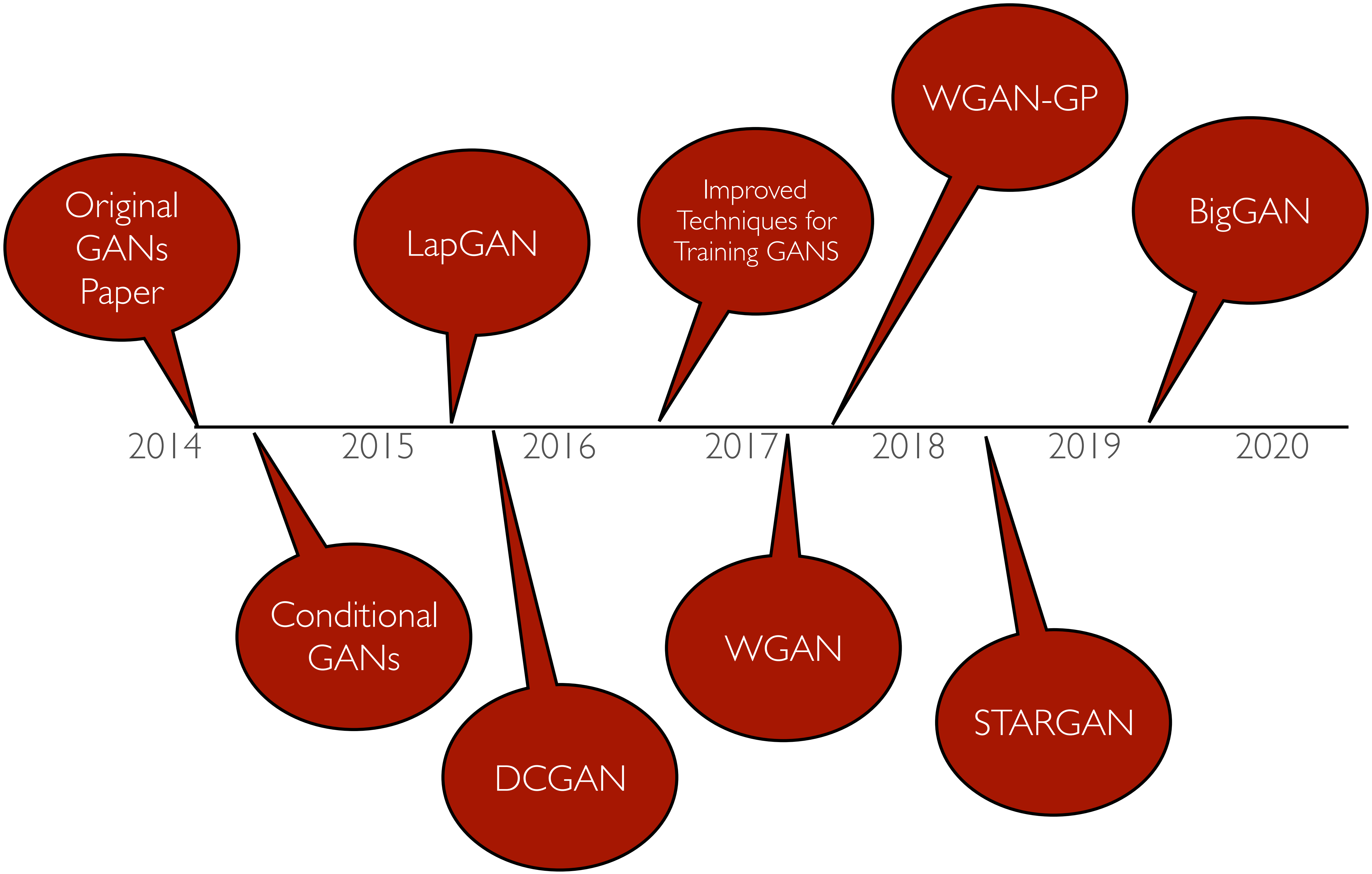
GANS PROGRESSION



GANS PROGRESSION



GANS PROGRESSION



Carnegie Mellon University

QUESTIONS?