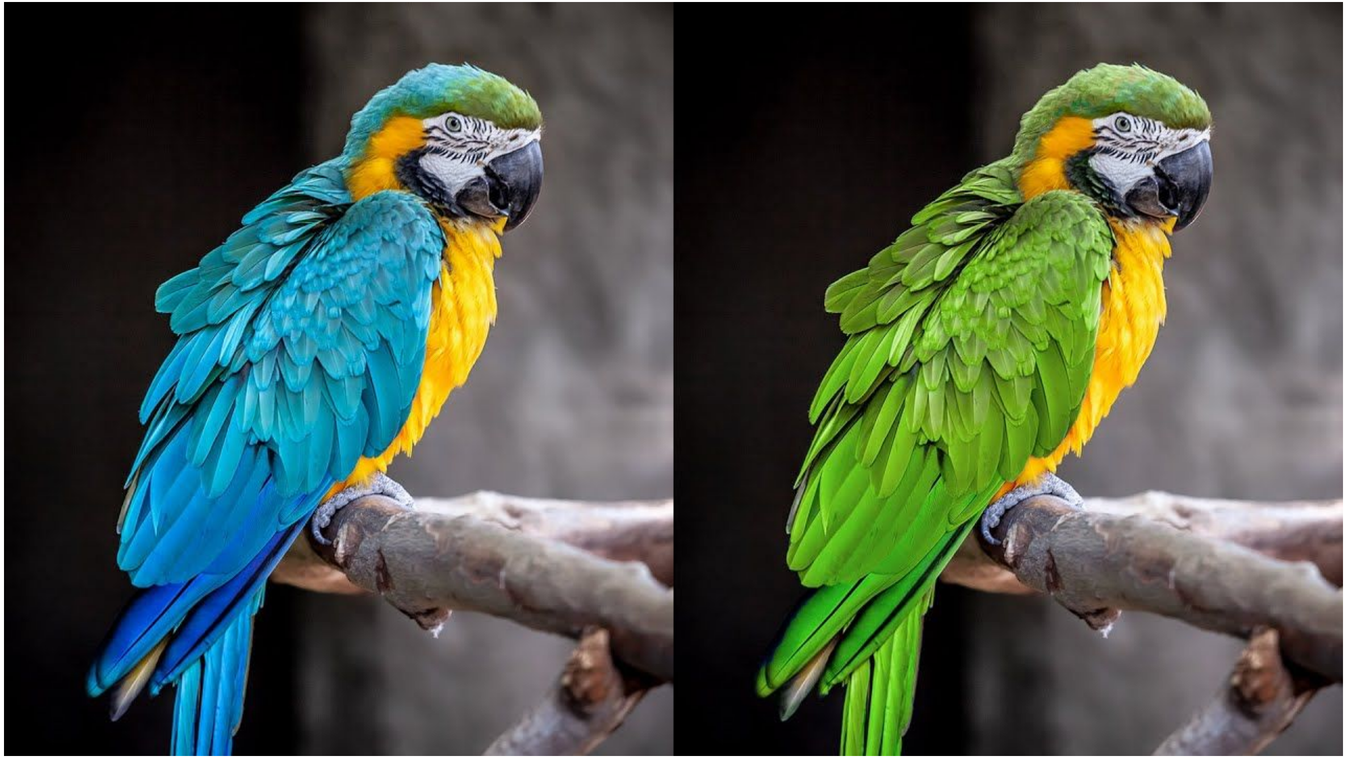
Bird Transformers: Modifying Birds Using Natural Language

Emma Qin, Will Schwarzer, Kyra Wilson, Orlando Zuniga

“Make this bird green!”



Specialized Skills

Computers

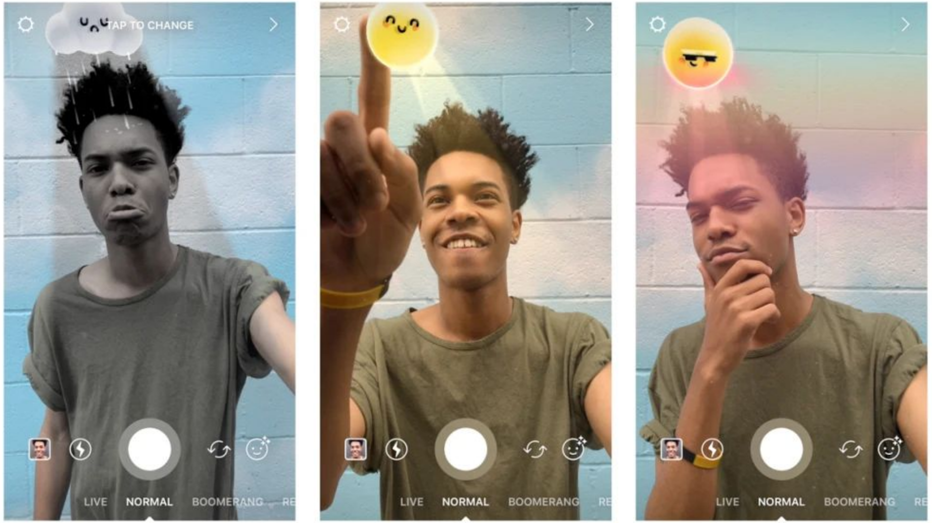
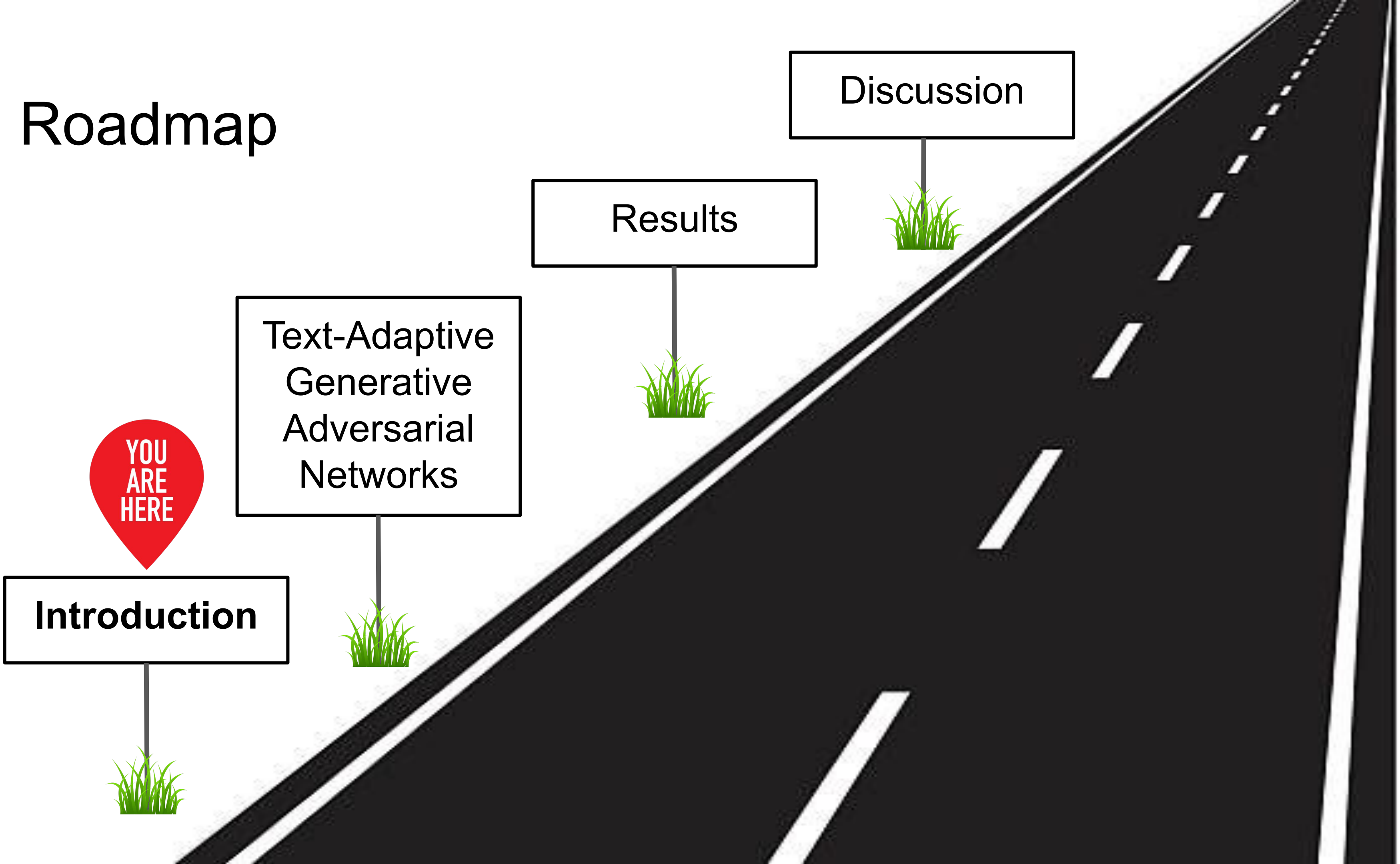


Image Modification with Natural Language

Roadmap



Introduction

Text-Adaptive Generative Adversarial Networks

Results

Discussion

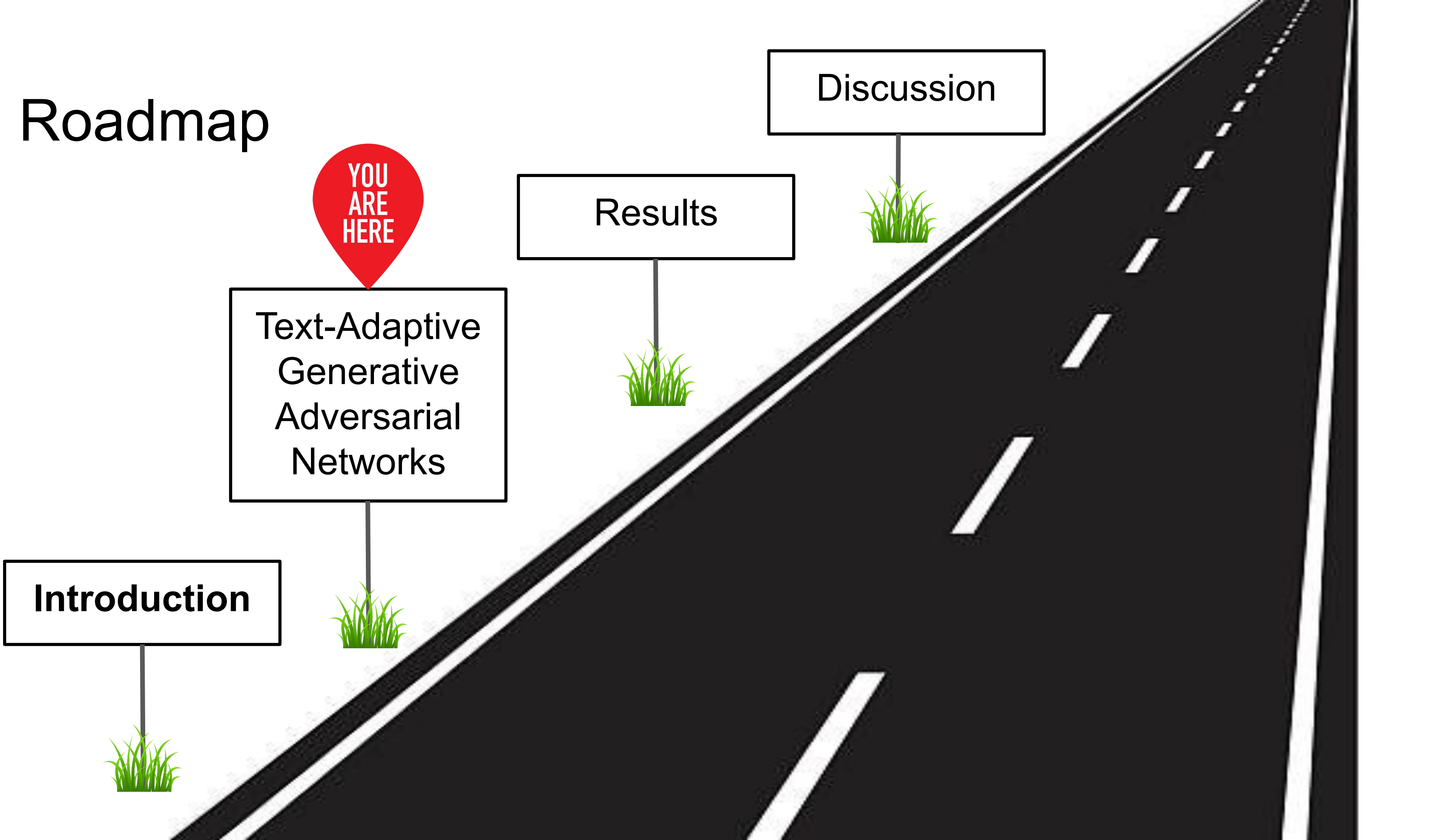
Generative Adversarial Networks (GANs)

00:00
13
14
GENERATOR
DISCRIMINATOR

Generator

Discriminator

Roadmap



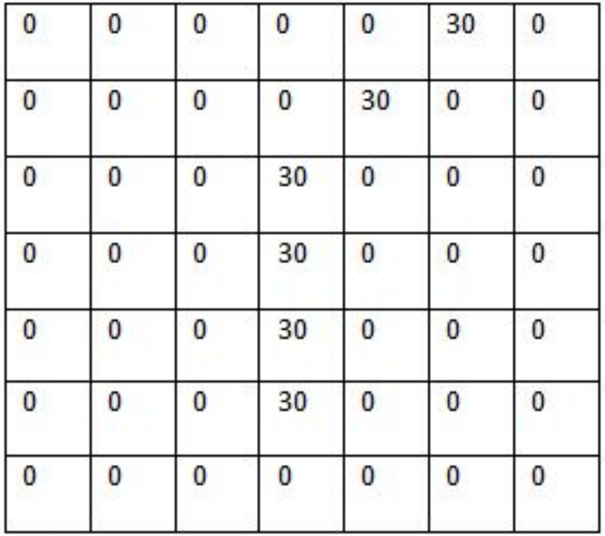
Introduction

Text-Adaptive Generative Adversarial Networks

Results

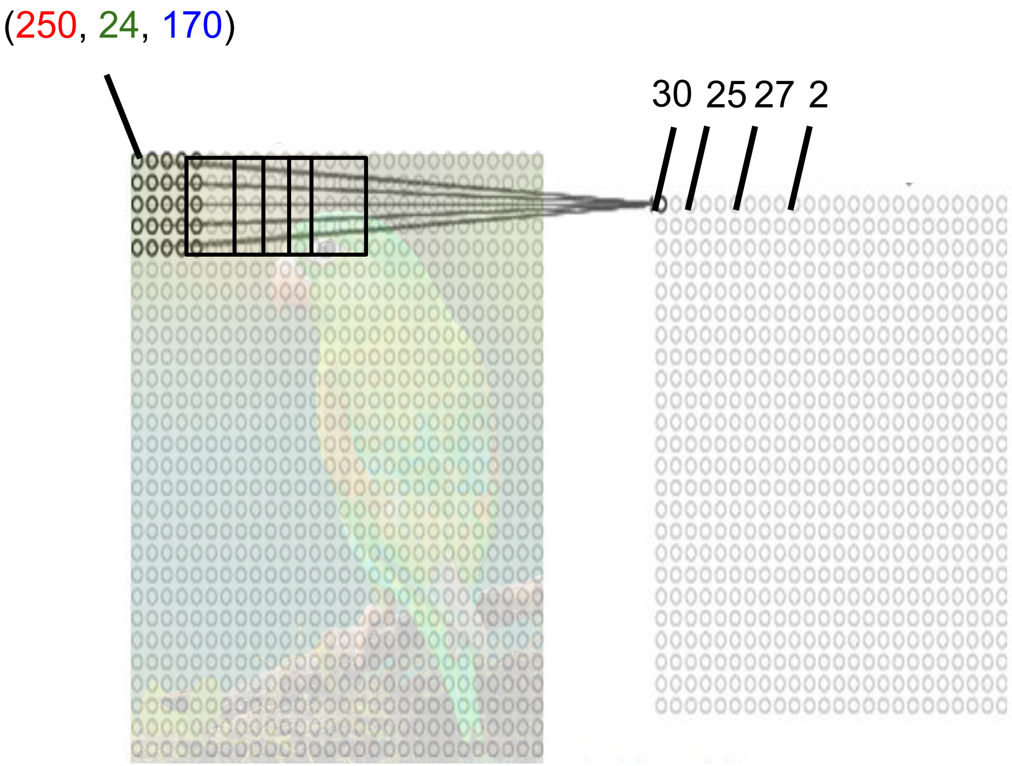
Discussion

Image Encoding

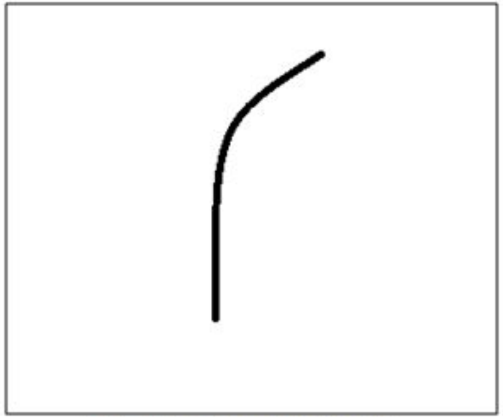


0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

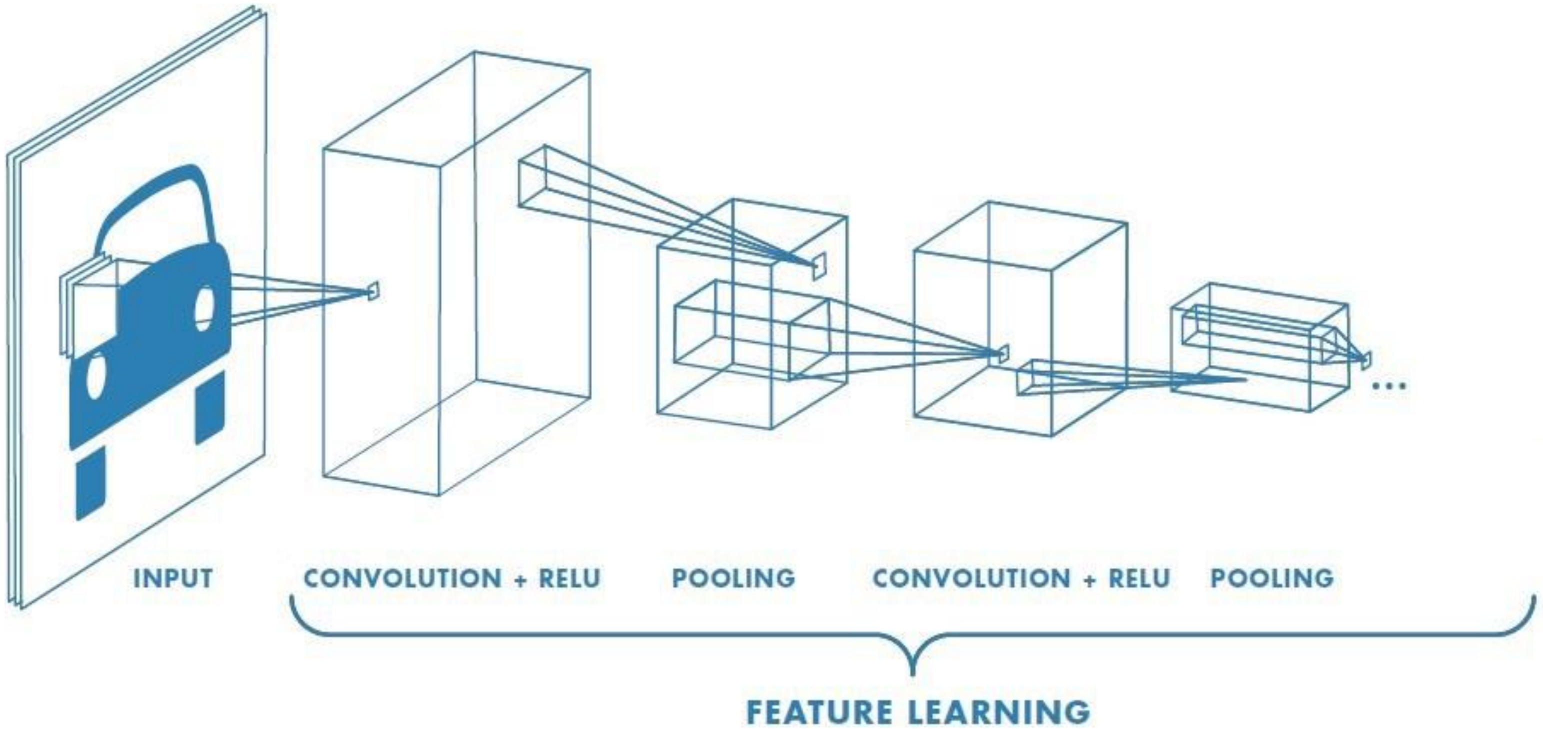
Image Encoding



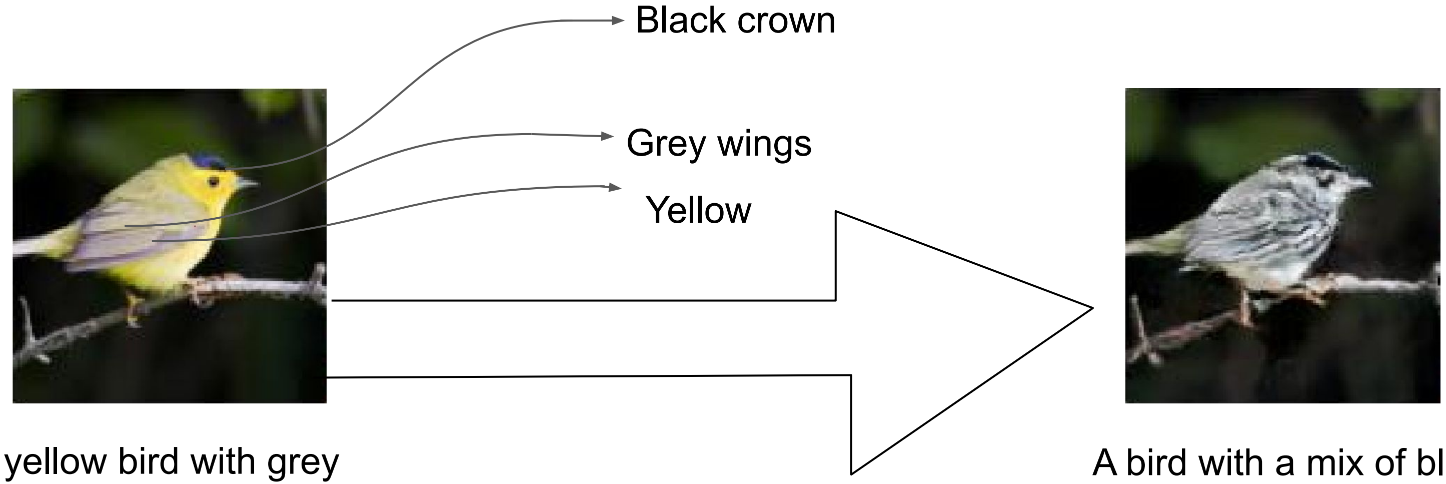
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0



Convolutional Neural Network



Intuition: modifying image with text



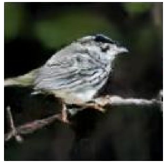
Black crown

Grey wings

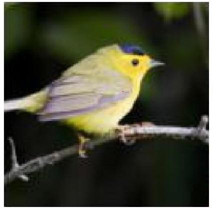
Yellow

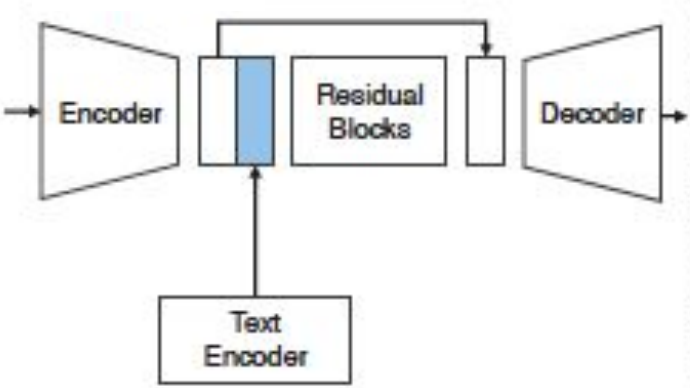
A yellow bird with grey wings and a black crown

A bird with a mix of black and white



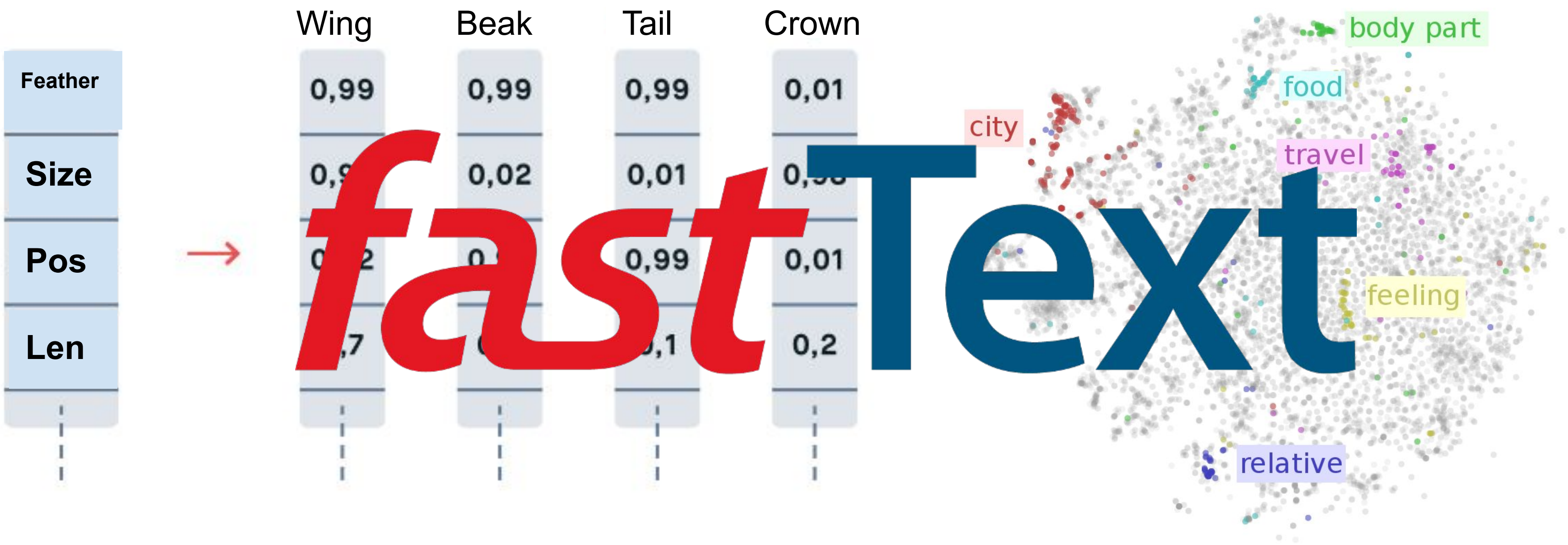
Generator



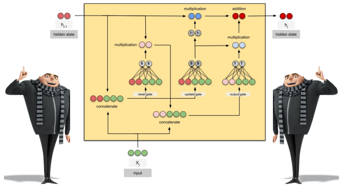


The bird is a
mix of black
and white..

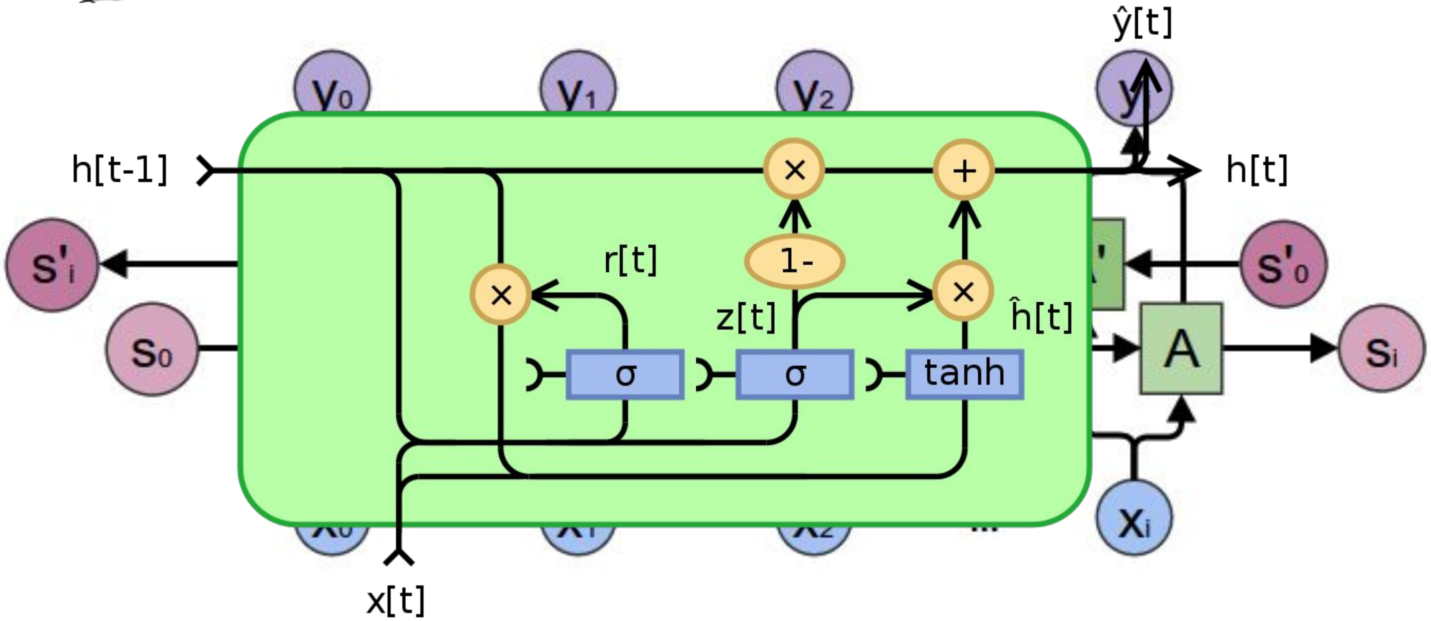
Text Encoding



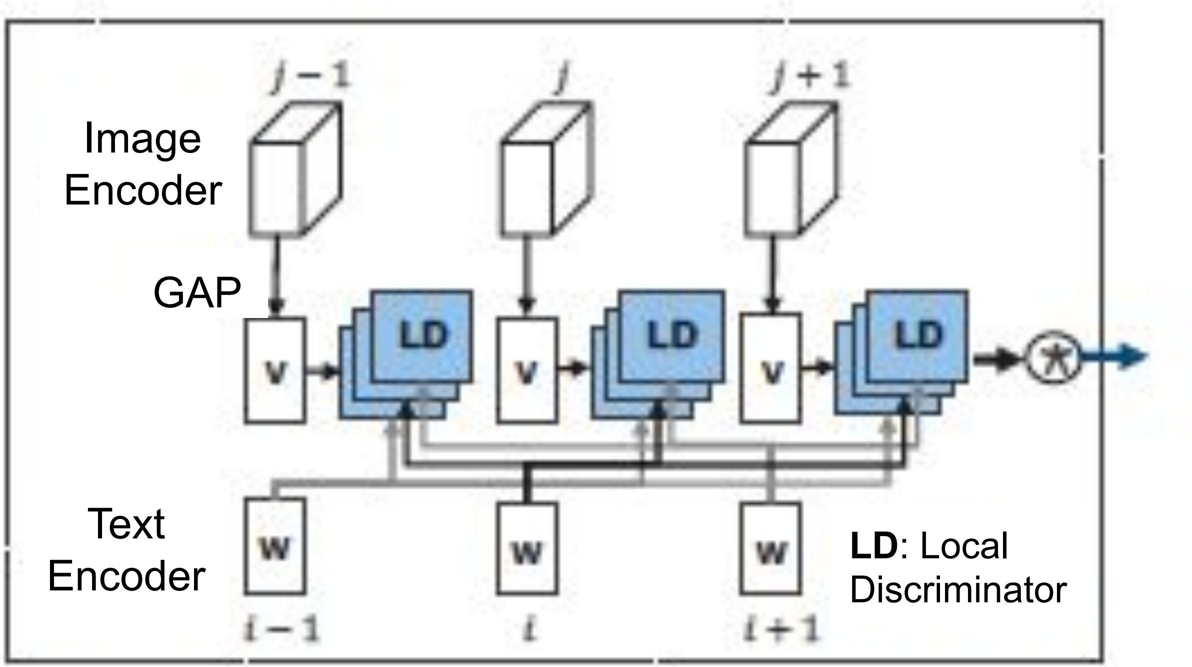
	Wing	Beak	Tail	Crown
Feather	0,99	0,99	0,99	0,01
Size	[illegible]	0,02	0,01	[illegible]
Pos	[illegible]	[illegible]	0,99	0,01
Len	[illegible]	[illegible]	[illegible]	0,2



Bidirectional GRU



Discriminator

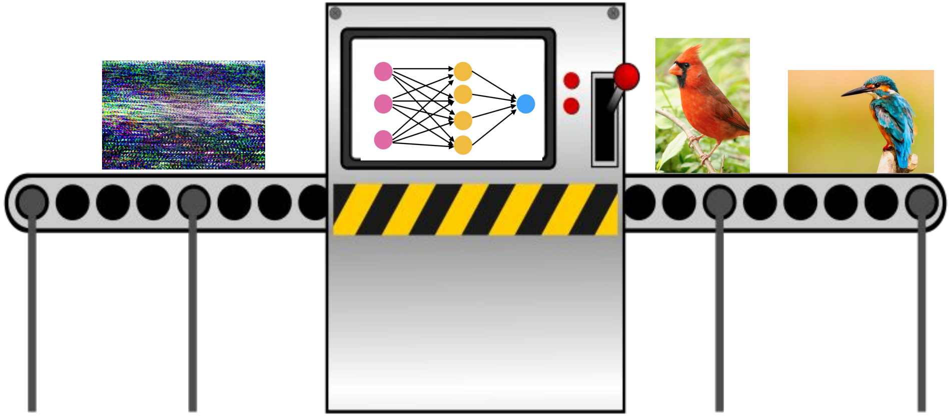


Loss

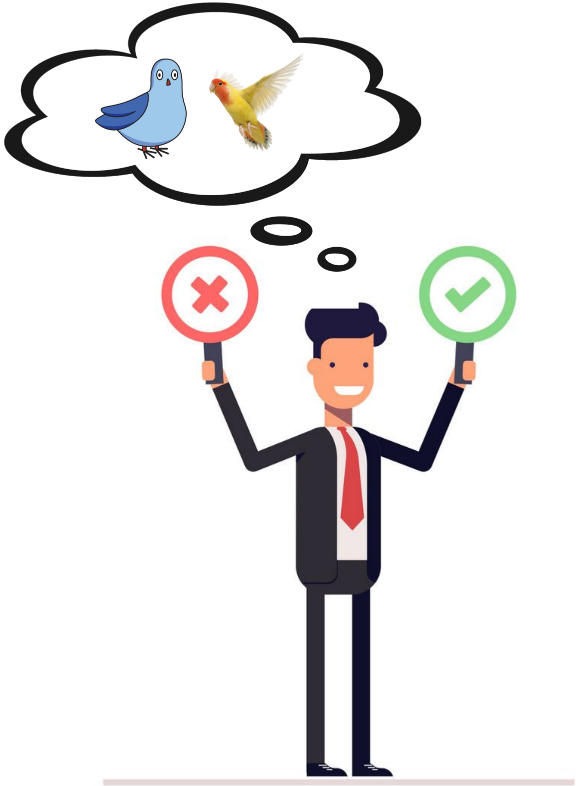
- Reconstruction loss
- Unconditional loss
- Conditional loss

Recap

What’s a GAN?



Generator



Discriminator

Recap

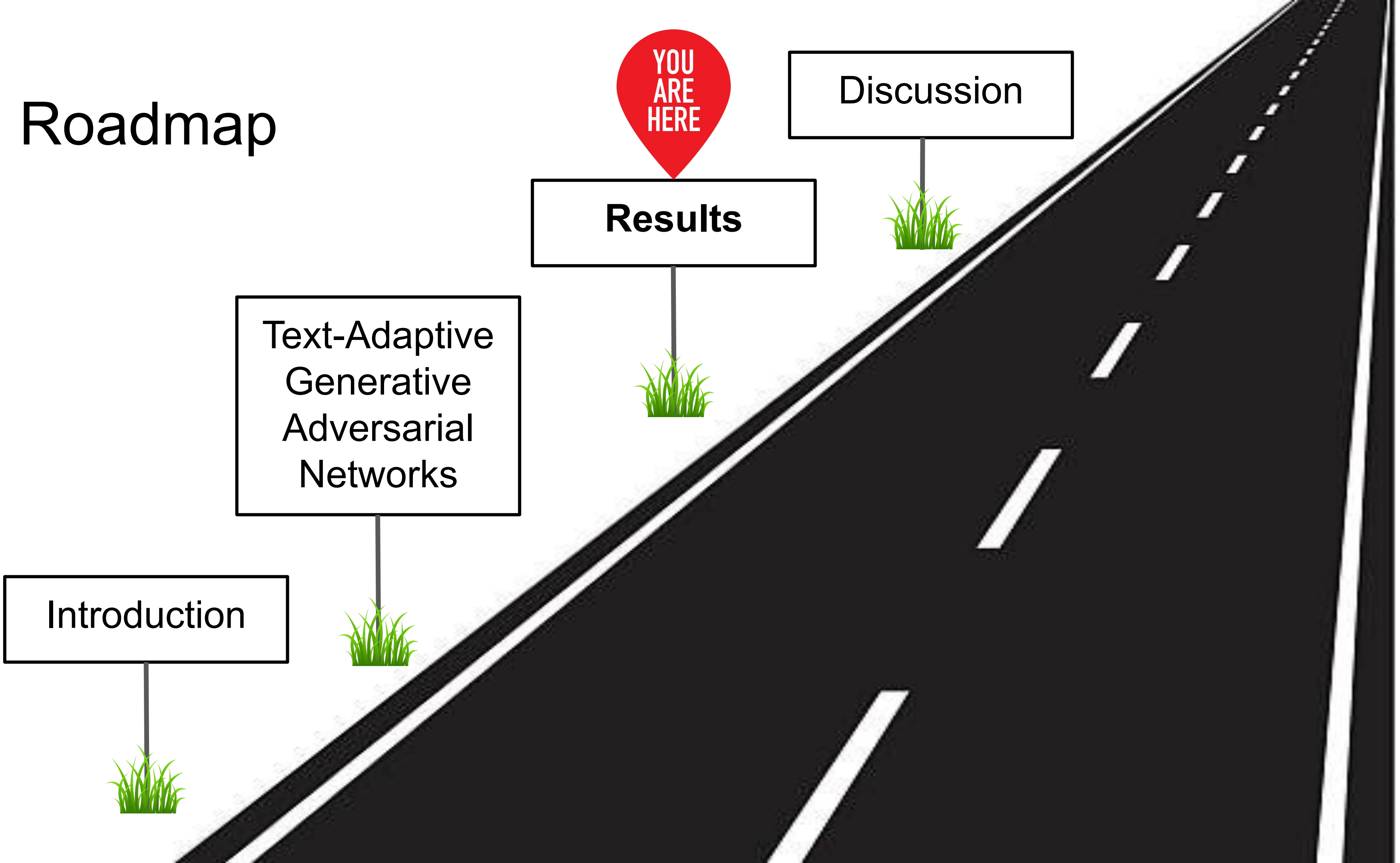
The problem?

- We want to manipulate images based on natural language descriptions

How does TAGAN differ?

- **Key difference:** the text-adaptive discriminator creates word-level local discriminators.

Roadmap



Introduction

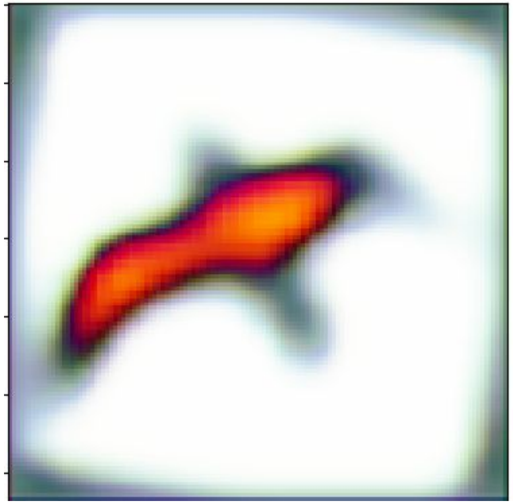
Text-Adaptive Generative Adversarial Networks

YOU ARE HERE

Results

Discussion

Results (Generator)

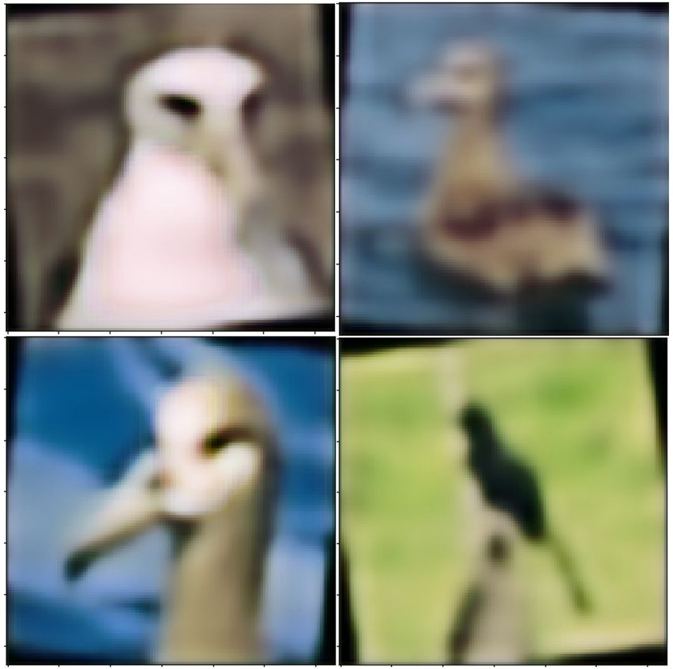


5 epochs

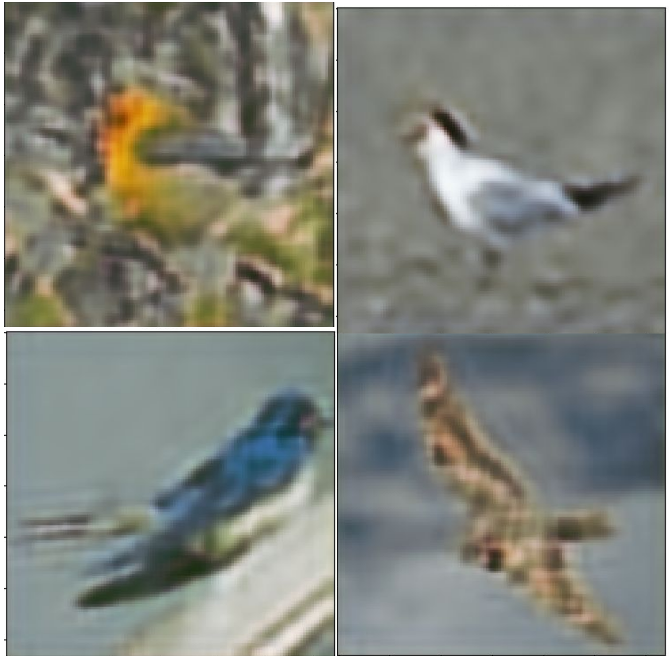
20 epochs

Results (Discriminator pt1.)

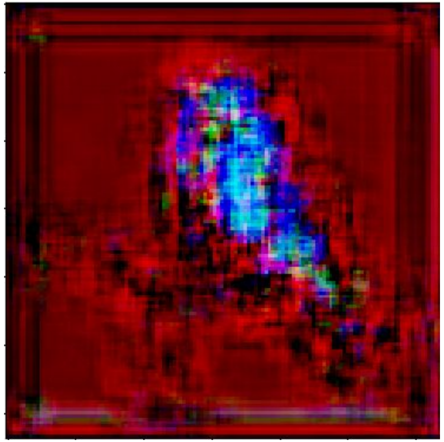
Generator with no text



Network with small dataset



Epoch 1

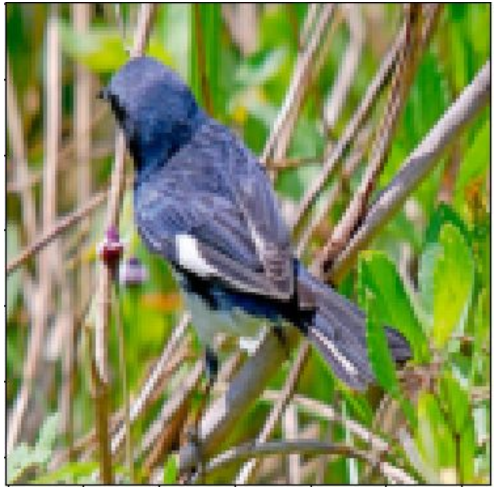


Epoch 6

Epoch 10

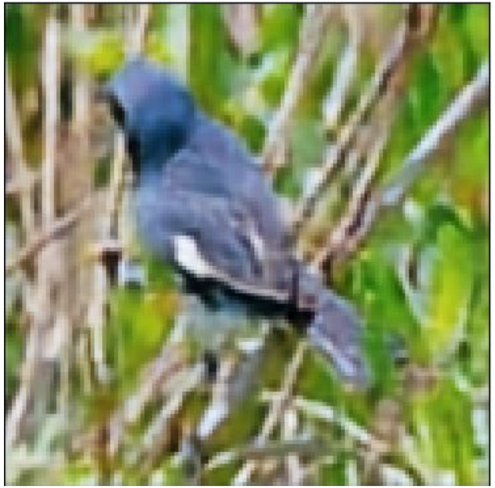
Epoch 20

30 Epochs



Real Image

Caption: this particular bird has a **belly that is white and has black spots**

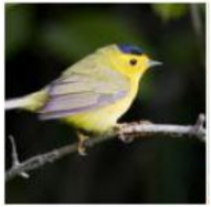


Fake Image

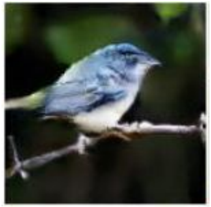
TAGAN Qualitative Results

TAGAN Results:

Original

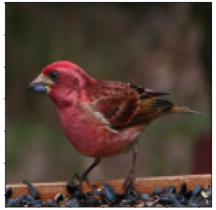


The bird has **wings that are blue** and has a **white belly.**

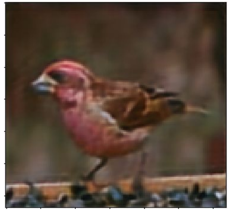


Our Results:

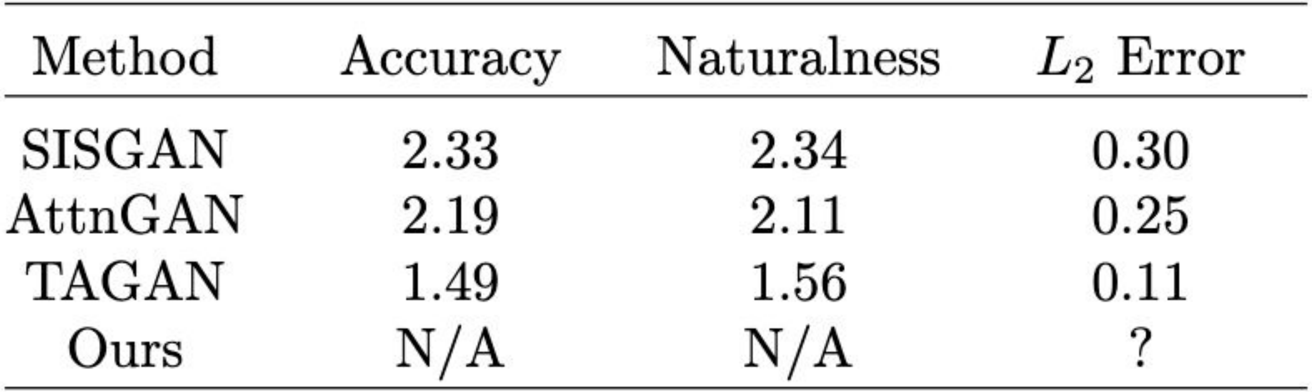
Original



The small bird is **yellow and brown speckled** with a **pointed beak**

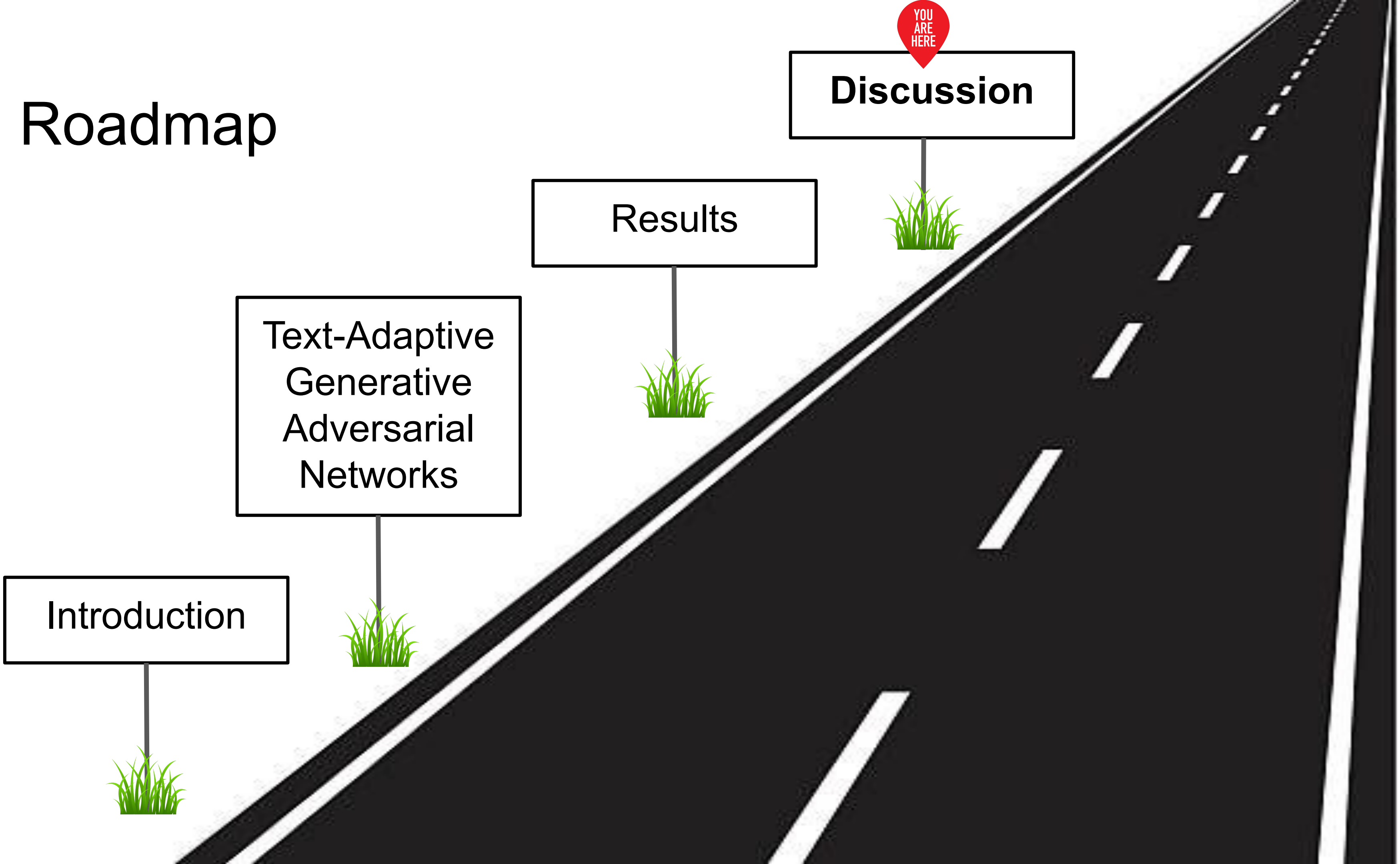


TAGAN Quantitative Results



Method	Accuracy	Naturalness	L_2 Error
SISGAN	2.33	2.34	0.30
AttnGAN	2.19	2.11	0.25
TAGAN	1.49	1.56	0.11
Ours	N/A	N/A	?

Roadmap



Introduction

Text-Adaptive Generative Adversarial Networks

Results

Discussion

Discussion

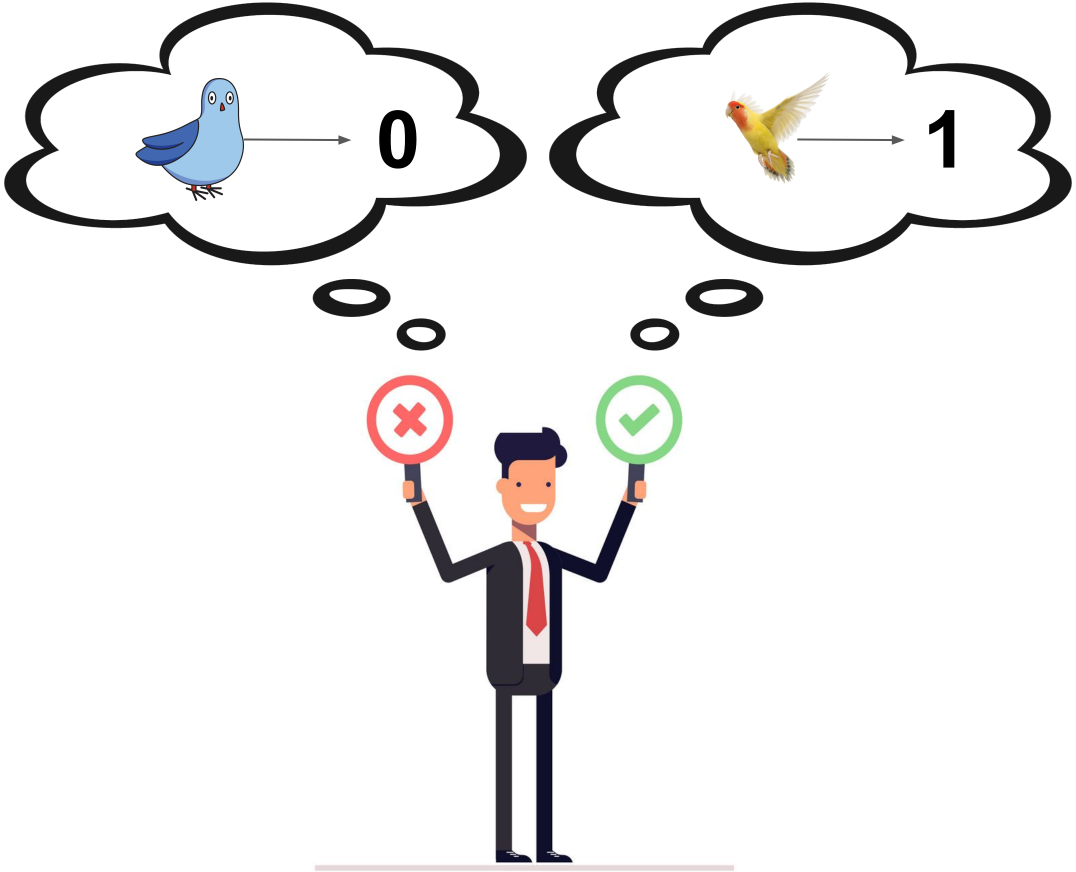
- Why are our results different?
- Can we make them not different?
- If we made them not different, what would we then make different?

Discussion

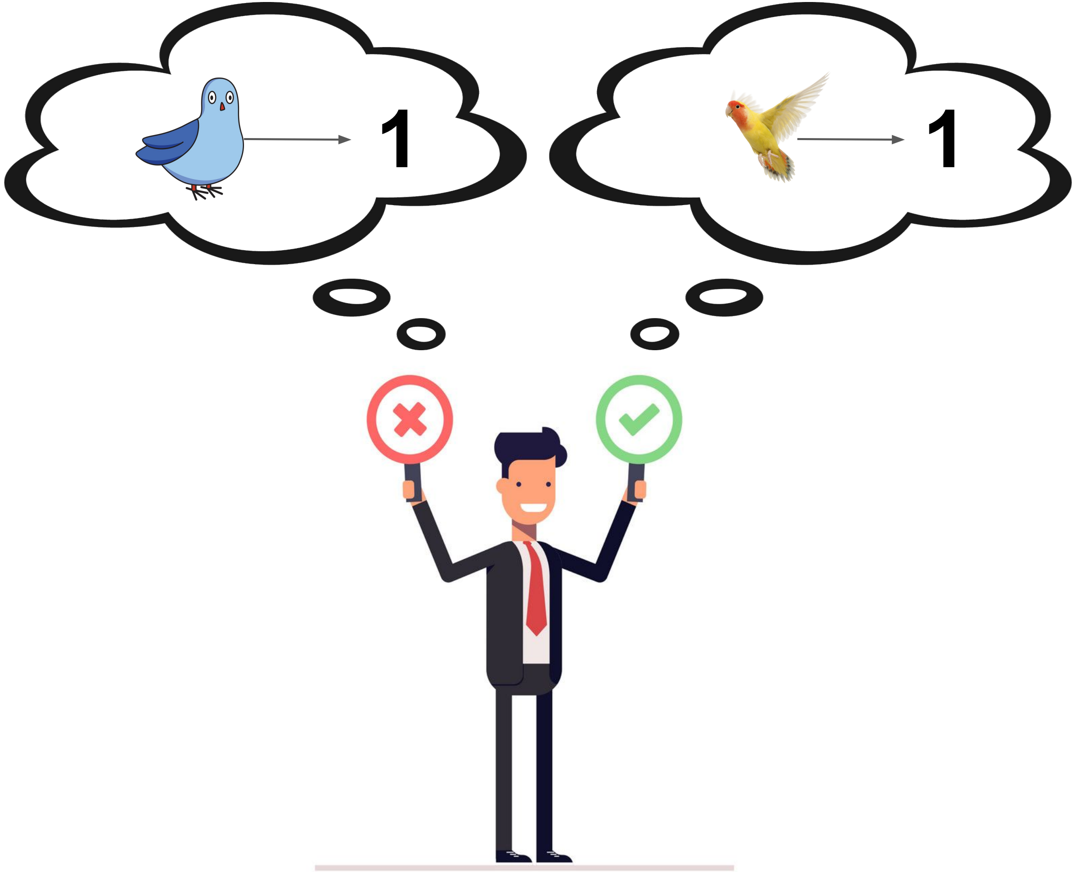
- **Why are our results different?**
- Can we make them not different?
- If we made them not different, what would we then make different?

Q: Why are our results different?

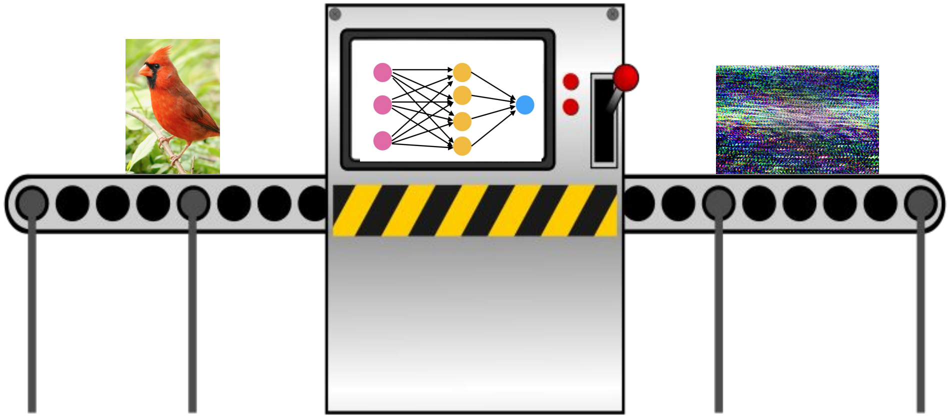
A: Unstable training



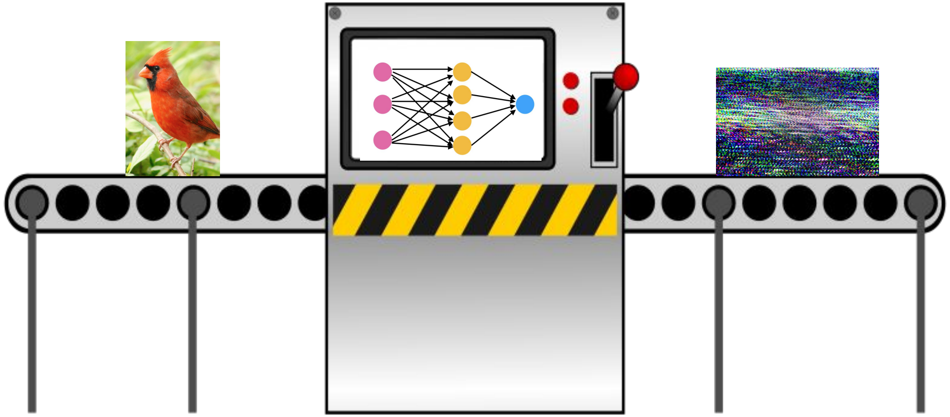
Discriminator

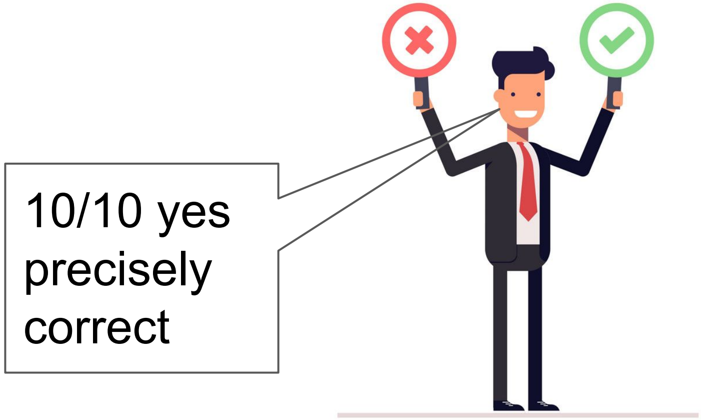


Discriminator



“this particular bird has a belly that
is white and has black spots”

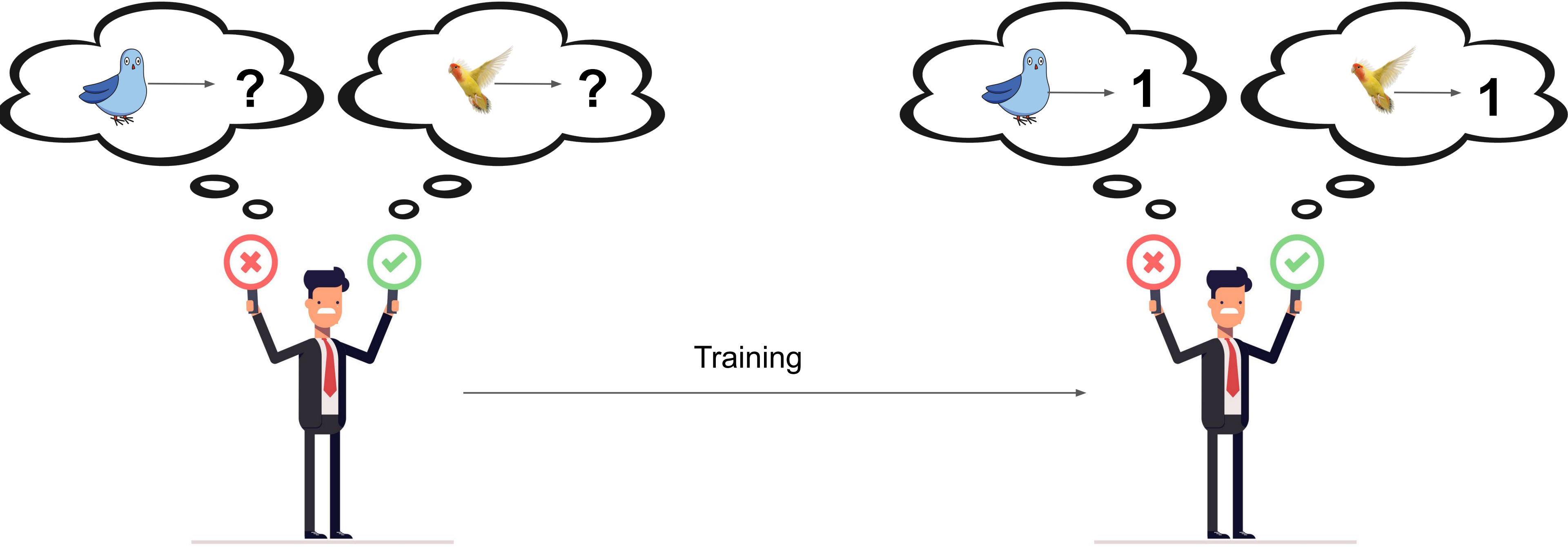




Discussion

- Why are our results different? *A: Unstable training*
- **Can we make them not different?**
- If we made them not different, what would we then make different?

Could it be our problem?



?

?

Training

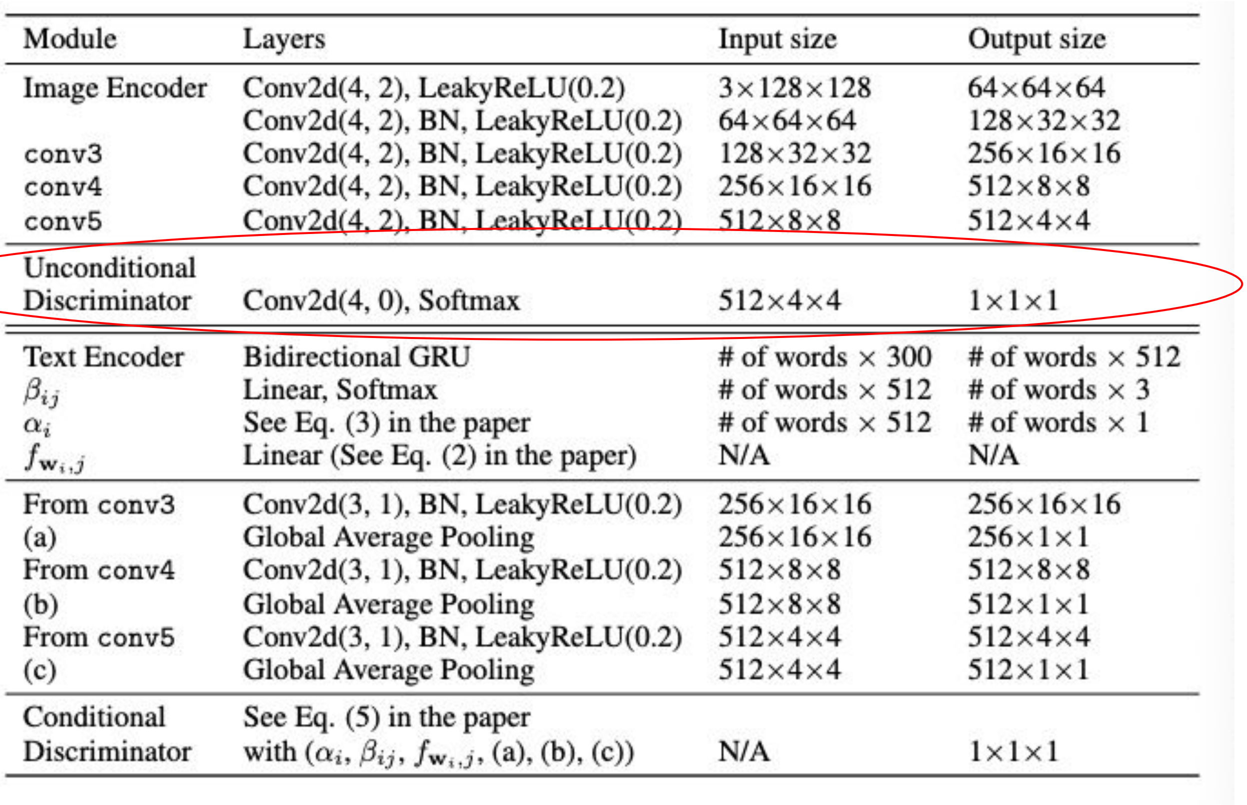
1

1

(Buggy)
Discriminator

(Buggy)
Discriminator

Could it be the paper’s problem?



Module	Layers	Input size	Output size
Image Encoder	Conv2d(4, 2), LeakyReLU(0.2)	3×128×128	64×64×64
	Conv2d(4, 2), BN, LeakyReLU(0.2)	64×64×64	128×32×32
conv3	Conv2d(4, 2), BN, LeakyReLU(0.2)	128×32×32	256×16×16
conv4	Conv2d(4, 2), BN, LeakyReLU(0.2)	256×16×16	512×8×8
conv5	Conv2d(4, 2), BN, LeakyReLU(0.2)	512×8×8	512×4×4
Unconditional Discriminator	Conv2d(4, 0), Softmax	512×4×4	1×1×1
Text Encoder	Bidirectional GRU	# of words × 300	# of words × 512
β_{ij}	Linear, Softmax	# of words × 512	# of words × 3
α_i	See Eq. (3) in the paper	# of words × 512	# of words × 1
$f_{\mathbf{w}_i,j}$	Linear (See Eq. (2) in the paper)	N/A	N/A
From conv3	Conv2d(3, 1), BN, LeakyReLU(0.2)	256×16×16	256×16×16
(a)	Global Average Pooling	256×16×16	256×1×1
From conv4	Conv2d(3, 1), BN, LeakyReLU(0.2)	512×8×8	512×8×8
(b)	Global Average Pooling	512×8×8	512×1×1
From conv5	Conv2d(3, 1), BN, LeakyReLU(0.2)	512×4×4	512×4×4
(c)	Global Average Pooling	512×4×4	512×1×1
Conditional Discriminator	See Eq. (5) in the paper with (α_i, β_{ij}, $f_{\mathbf{w}_i,j}$, (a), (b), (c))	N/A	1×1×1

Could it be the model’s problem?

Not only are GANs famous for training instability, the authors themselves mention having such problems:

- “Note that we do not penalize generated outputs using the conditional discriminator in Eq. (6) due to **instability of training**.”
- “We set λ_1 and λ_2 to 10 and 2 respectively considering both the visual quality and the **training stability**.”

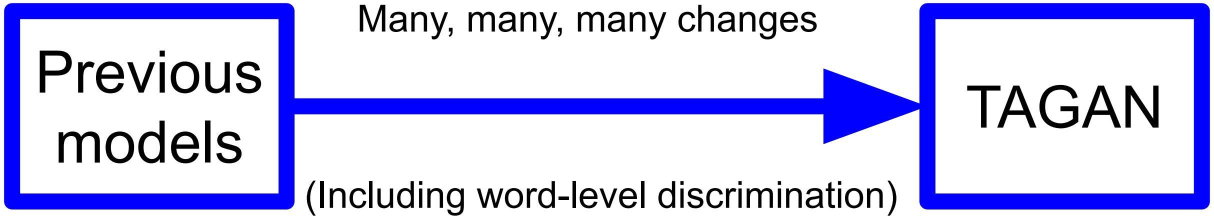
Q: But then how did they transform birds???

(Possible) A: Trying over and over and over and over and over and over and over and over and over and over and over again

Discussion

- Why are our results different? *A: Unstable training*
- Can we make them not different? *A: It depends, but probably*
- **If we made them not different, what would we then make different?**

Putting the science back in CS: the value of controls

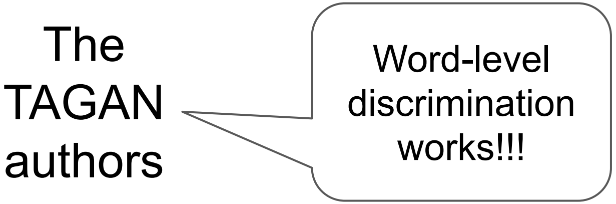


Previous models

Many, many, many changes

(Including word-level discrimination)

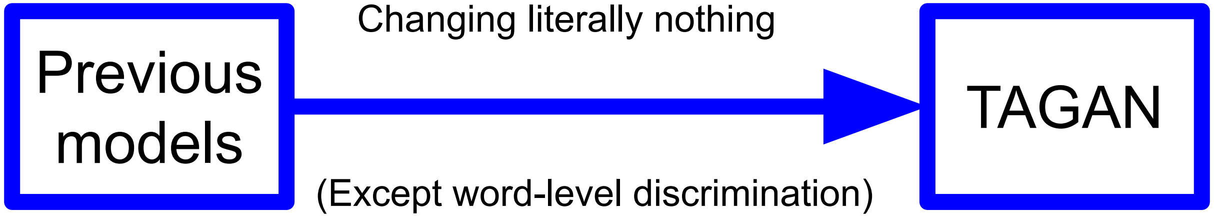
TAGAN



The TAGAN authors

Word-level discrimination works!!!

Putting the science back in CS: the value of controls



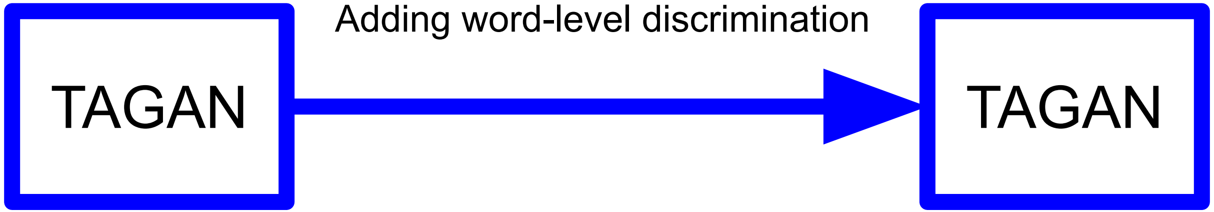
Previous models

Changing literally nothing

(Except word-level discrimination)

TAGAN

Putting the science back in CS: the value of controls



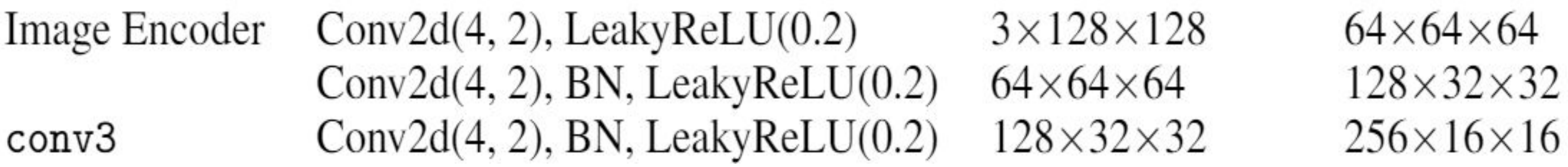
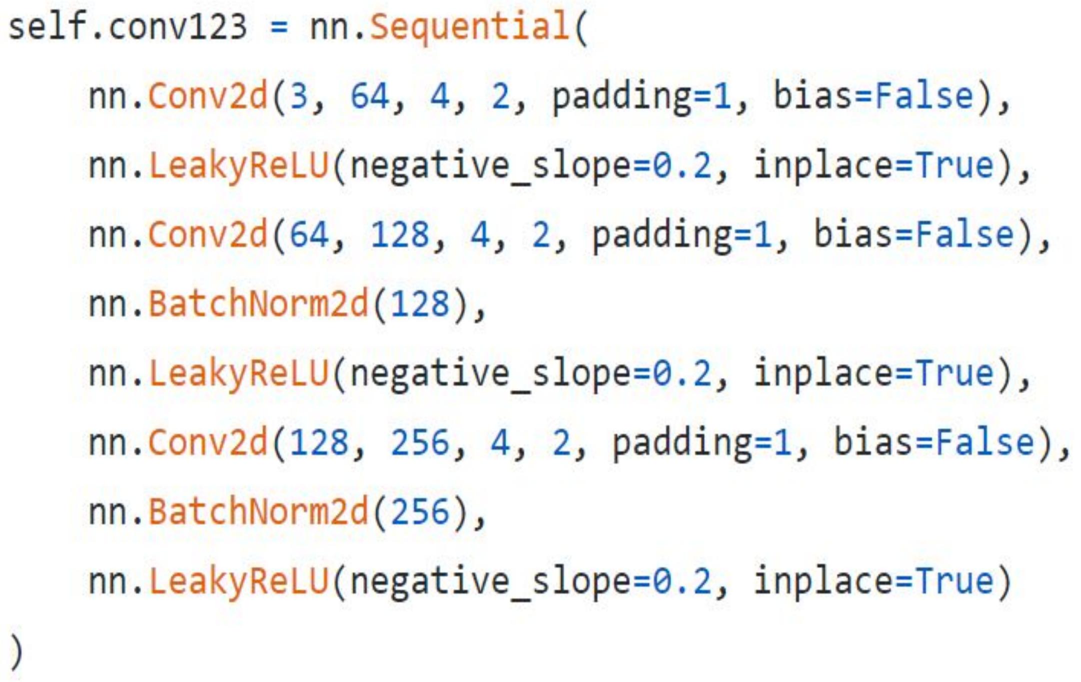
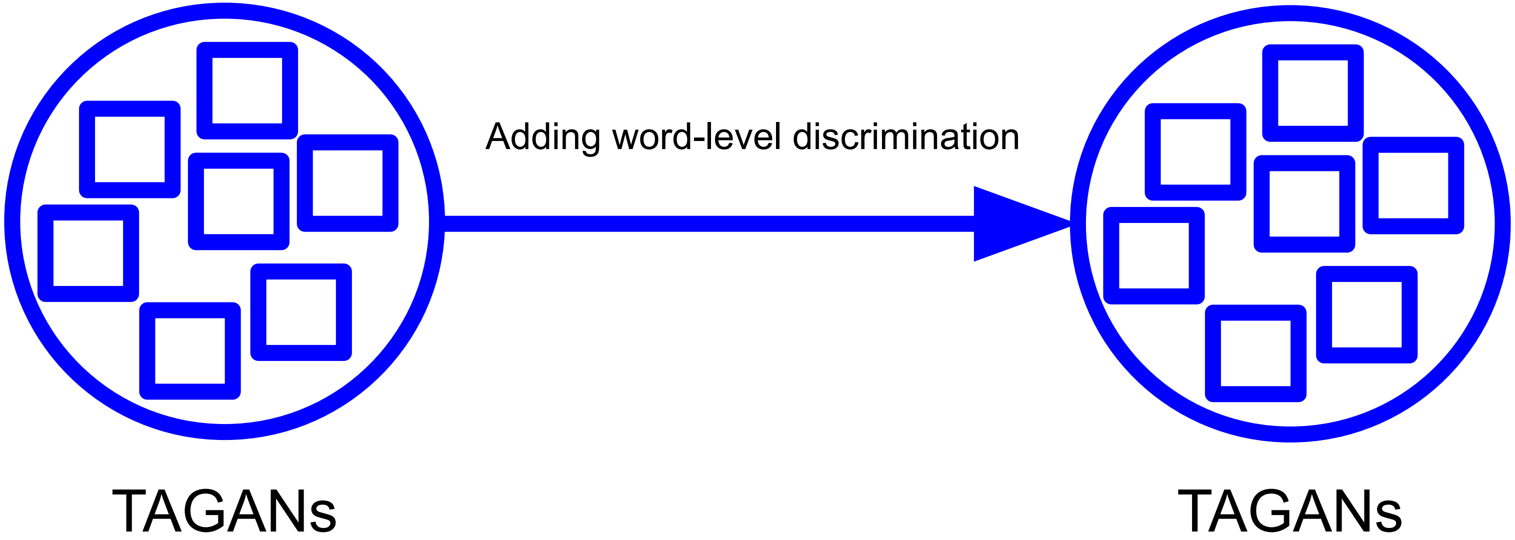


Image Encoder	Conv2d(4, 2), LeakyReLU(0.2)	3×128×128	64×64×64
	Conv2d(4, 2), BN, LeakyReLU(0.2)	64×64×64	128×32×32
conv3	Conv2d(4, 2), BN, LeakyReLU(0.2)	128×32×32	256×16×16



```
self.conv123 = nn.Sequential(
    nn.Conv2d(3, 64, 4, 2, padding=1, bias=False),
    nn.LeakyReLU(negative_slope=0.2, inplace=True),
    nn.Conv2d(64, 128, 4, 2, padding=1, bias=False),
    nn.BatchNorm2d(128),
    nn.LeakyReLU(negative_slope=0.2, inplace=True),
    nn.Conv2d(128, 256, 4, 2, padding=1, bias=False),
    nn.BatchNorm2d(256),
    nn.LeakyReLU(negative_slope=0.2, inplace=True)
)
```

Putting the science back in CS: DIY replication



Discussion

- Why are our results different? *A: Unstable training*
- Can we make them not different? *A: It depends, but probably*
- If we made them not different, what would we then make different? *A: Science*

Acknowledgements

Seonghyeon Nam, Yunji Kim, and Seon Joo Kim

Thank you!

Questions?