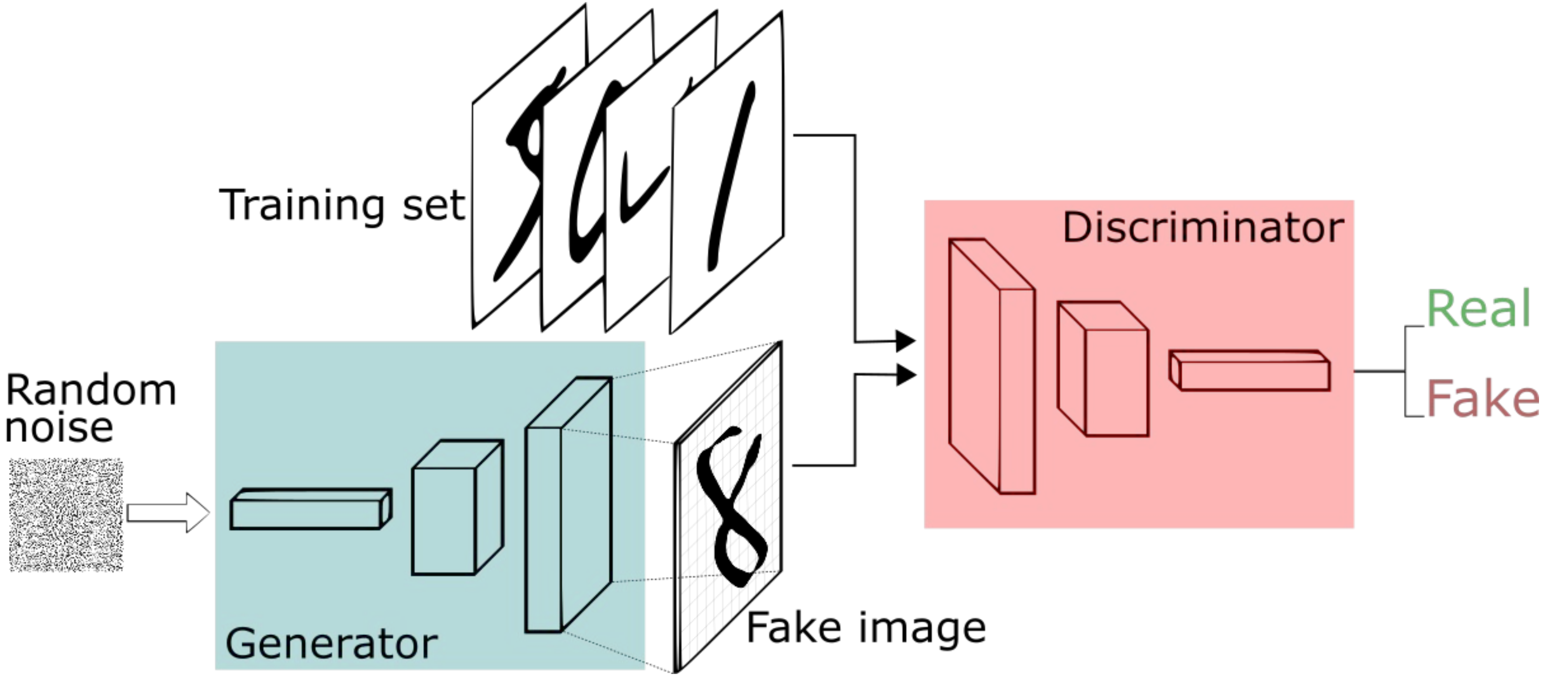
Deep Fusion-GAN for Text-to-Image Synthesis

Aditya Shah
MS CS

Contents

- ❏ Context
- ❏ Challenges with previous researches
- ❏ Algorithm
- ❏ Experiments and Results
- ❏ Ablation Study
- ❏ Strengths and Weakness
- ❏ Future work

Generative Adversarial Networks



Ref: https://sthalles.github.io/intro-to-gans/

Related works in text-to-image generation

- Stacked - GAN: Uses a series of G-D networks to generate images of different scale

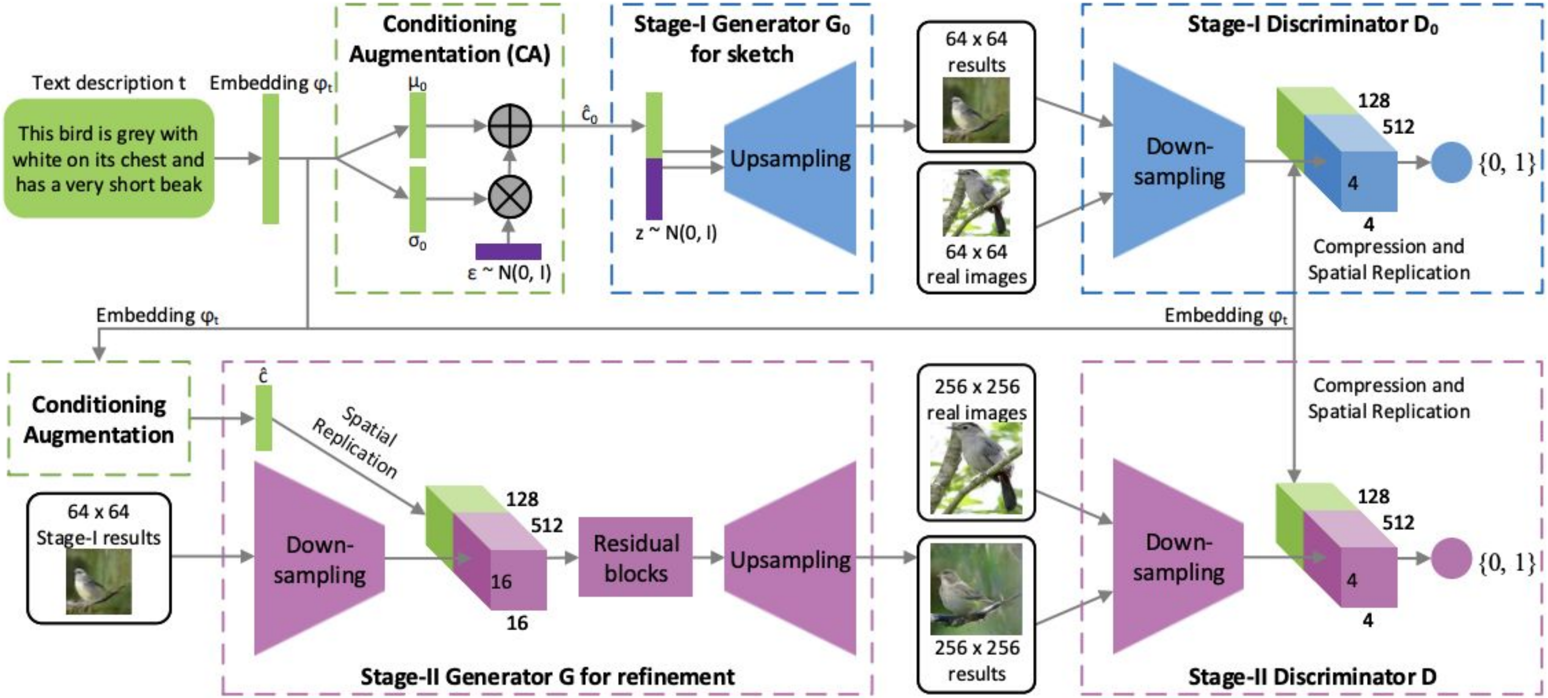
Related works in text-to-image generation

- Stacked - GAN: Uses a series of G-D networks to generate images of different scale
- AttnGAN - Uses cross-modal attention mechanism

Related works in text-to-image generation

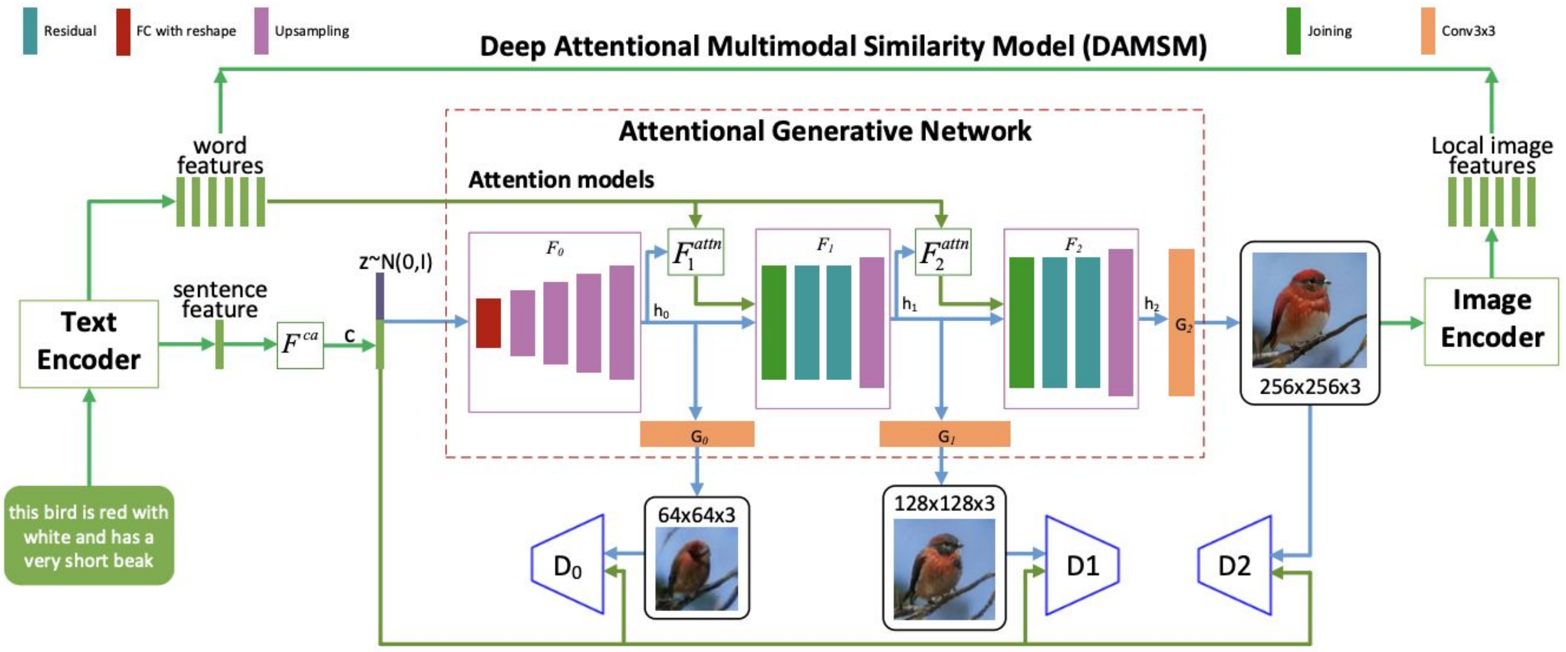
- ❏ Stacked - GAN: Uses a series of G-D networks to generate images of different scale
- ❏ AttnGAN - Uses cross-modal attention mechanism
- ❏ SD-GAN: Uses siamese structure to distill the semantic commons from texts

Stacked GAN



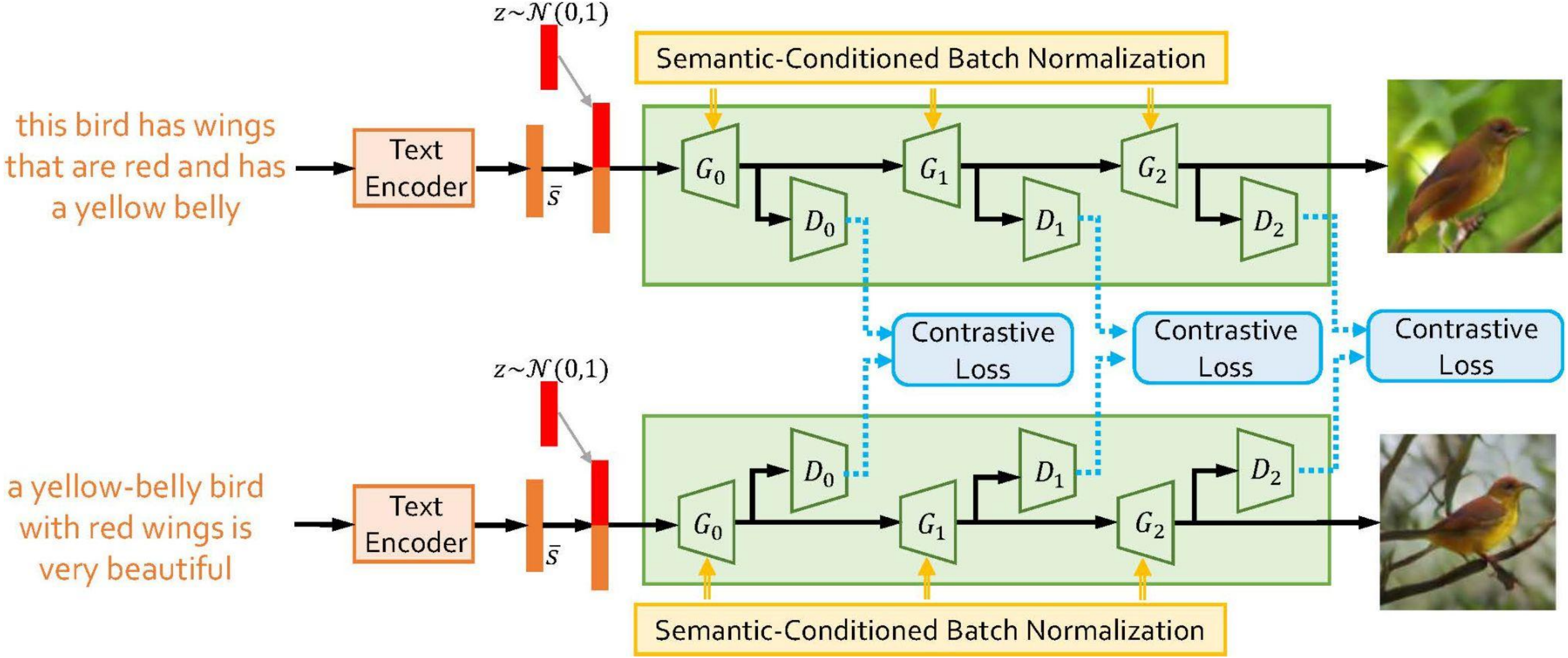
Source: https://arxiv.org/pdf/1612.03242.pdf

AttnGAN (Cross-Modal Attn Mechanism)



Source: https://arxiv.org/pdf/1711.10485.pdf

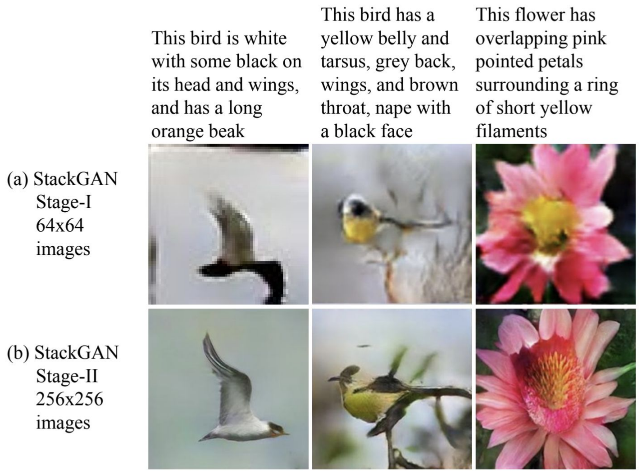
SD-GAN (Siamese structure for contrastive loss)



Source: https://gjyin91.github.io/projects/sdgan.html

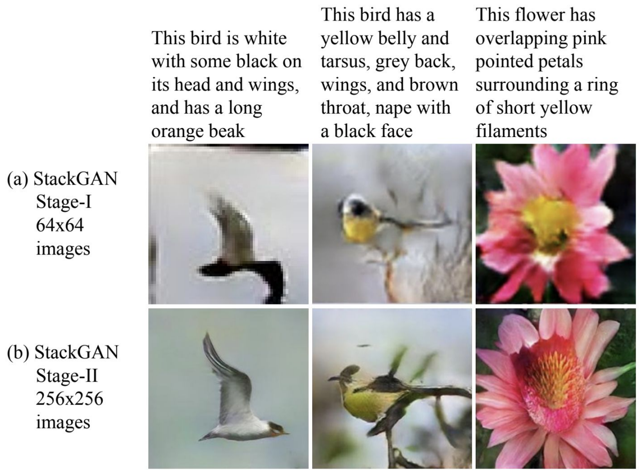
Challenges with previous work

- Use of multiple G-D networks to generate images of different scale



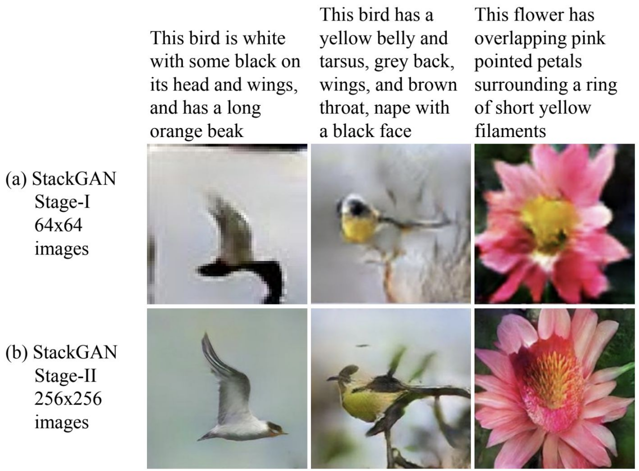
Challenges with previous work

- ❑ Use of multiple G-D networks to generate images of different scale
    - ❑ Costly to generate images this way



Challenges with previous work

- Use of multiple G-D networks to generate images of different scale
  - Costly to generate images this way
  - Images generated by later stage generators heavily depend on the initial G-D networks



Challenges with previous work

- Concatenation: Simple concatenation of text and image features - inefficient

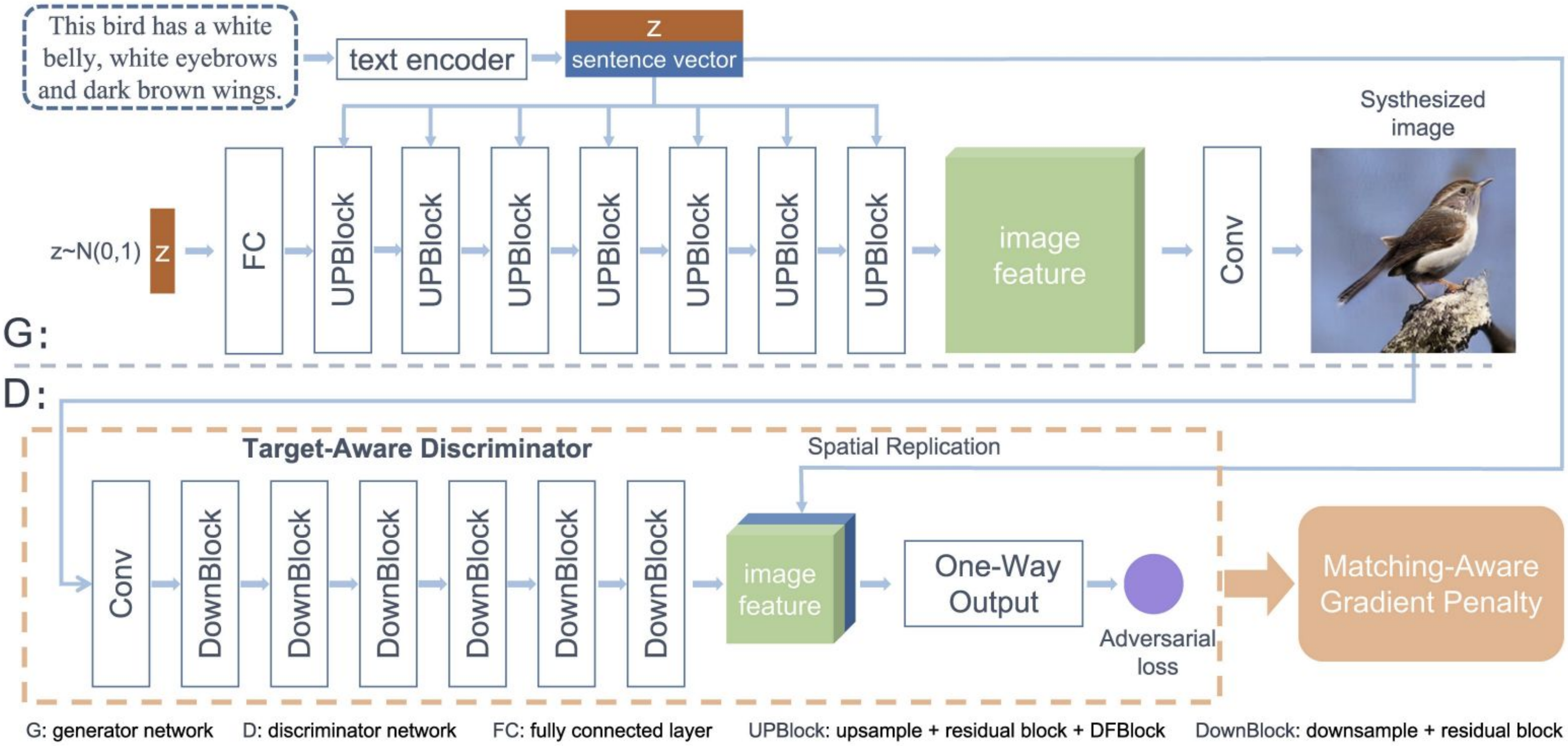
Challenges with previous work

- Concatenation: Simple concatenation of text and image features - inefficient
- Cross modal attention: As image size grows, the computation cost grows too.

Challenges with previous work

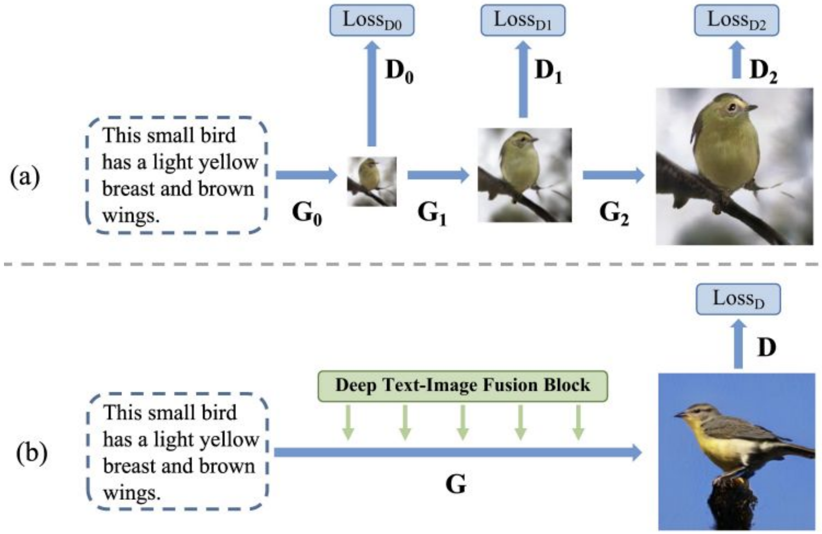
- ❑ Concatenation: Simple concatenation of text and image features - inefficient
- ❑ Cross modal attention: As image size grows, the computation cost grows too.
- ❑ Tries to find relation between each pixel and textual information.

Deep Fusion GAN



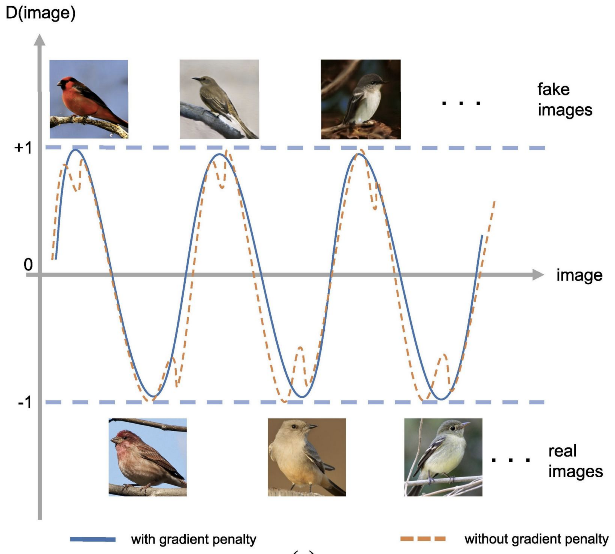
Simplified Text-to-Image backbone

- Instead of stacking, it uses a single Generator - Discriminator network
- Uses *hinge loss* to stabilize training process



Matching aware zero centered Gradient Penalty

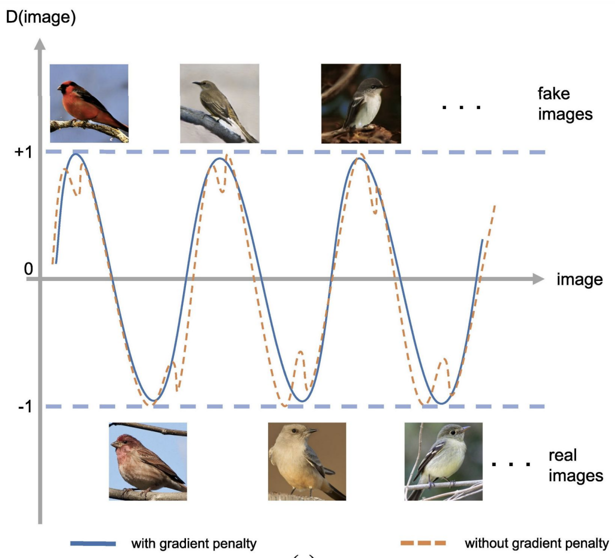
- ❏ Pushes the real data points towards minimum of loss curve



(a)

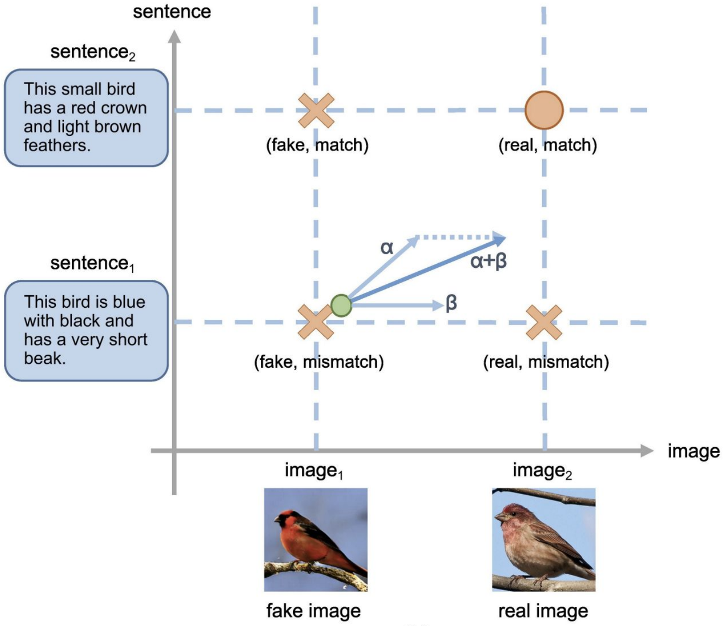
Matching aware zero centered Gradient Penalty

- ❑ Pushes the real data points towards minimum of loss curve
- ❑ Smoothens the surface for real data points - better convergence



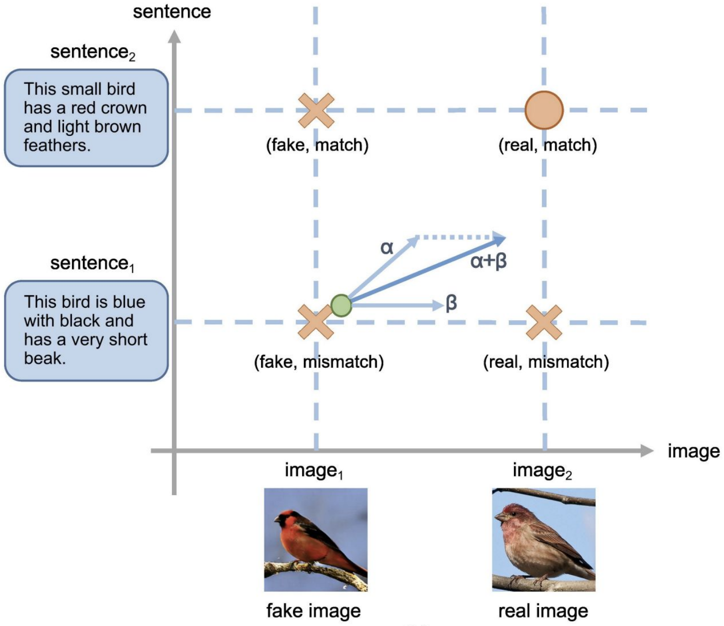
(a)

- Push the real image-text pair to the minimum of the loss function



(b)

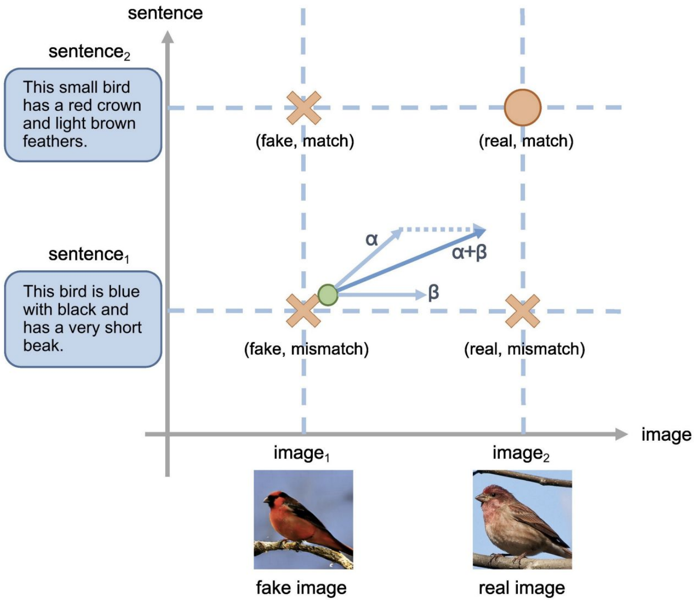
- Push the real image-text pair to the minimum of the loss function
- Enables the generator to synthesize more realistic images



(b)

Use of one way Discriminator

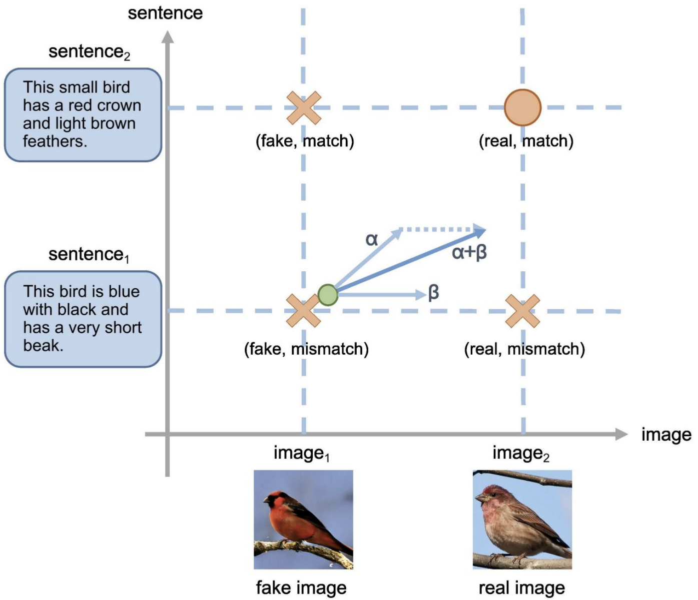
α and β collectively don't point towards the real and matching images



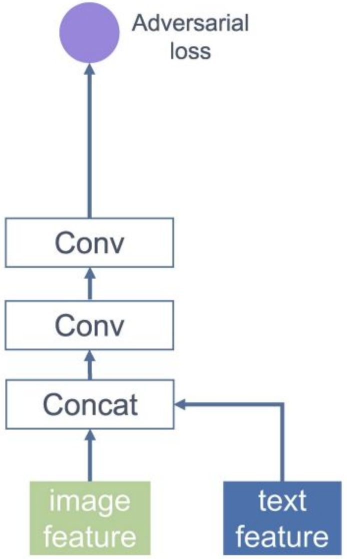
(b)

Use of one way Discriminator

α and β collectively don't point towards the real and matching images



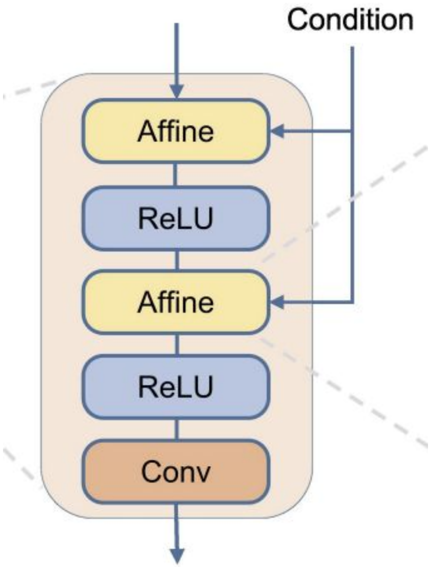
(b)



(b) One-Way Output

Deep - Fusion Block

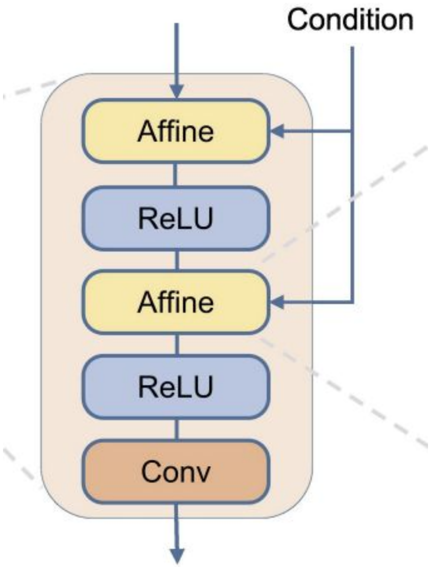
- In Conditional Batch Norm, affine parameters are found using additional network



(b) DFBlock

Deep - Fusion Block

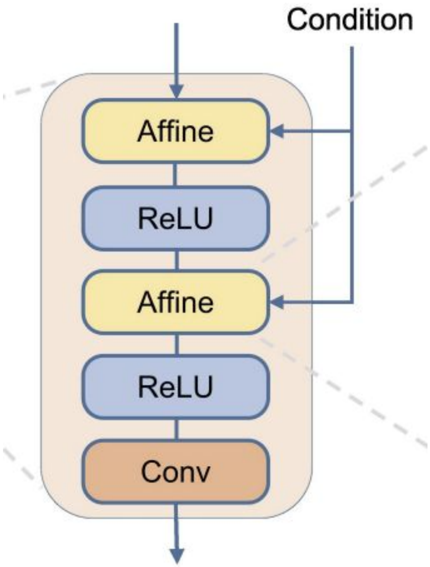
- In Conditional Batch Norm, affine parameters are found using additional network
- In DF Block, normalization of feature maps is skipped rather Affine transformations are used



(b) DFBlock

Deep - Fusion Block

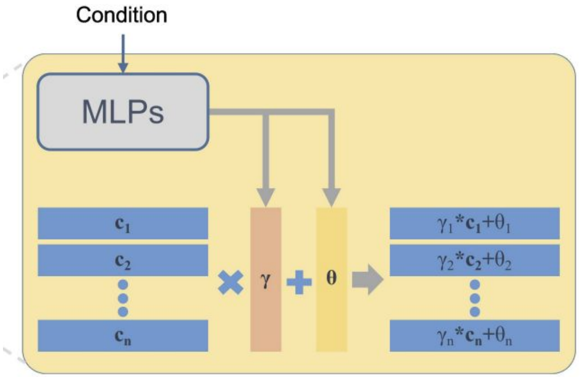
- In Conditional Batch Norm, affine parameters are found using additional network
- In DF Block, normalization of feature maps is skipped rather Affine transformations are used
- Affine + ReLU blocks are stacked together to form DF Block
- Helps to introduce Non linearity



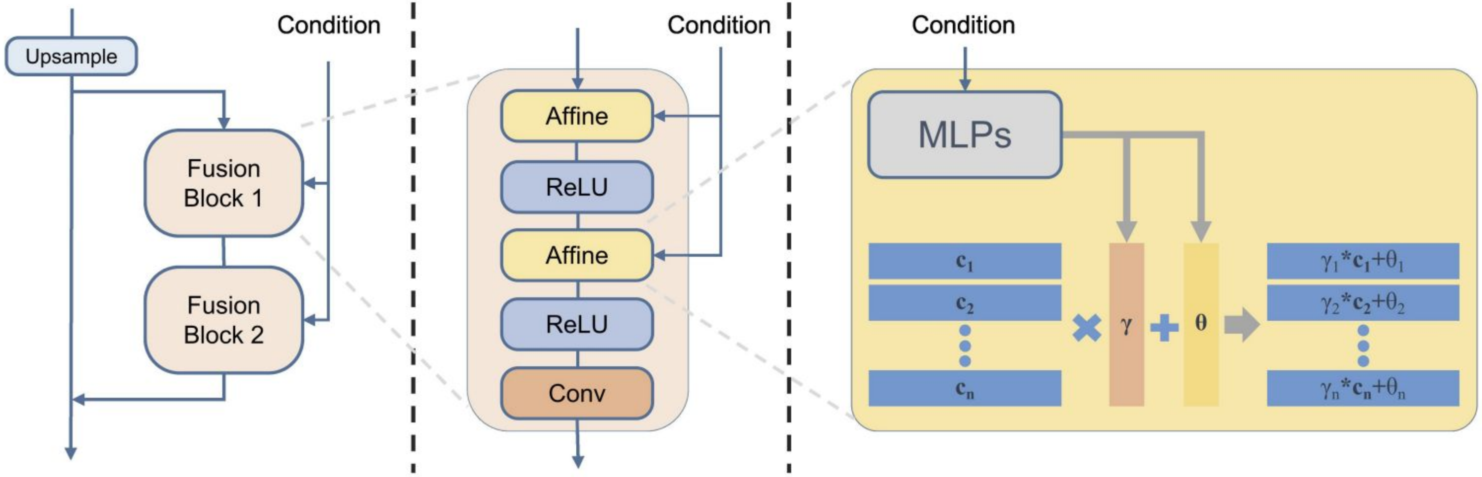
(b) DFBlock

Affine transformation

- Affine transformation
- Condition: Sentence vector passed through MLP
- All channels $c_1 \ldots c_2$ are multiplied by γ and added by θ



Zoomed out view



(a) UPBlock (b) DFBlock (c) Affine

Experiments and Results

COCO

- Contains 80k images for training and 40k images for testing
- Each image has 5 language descriptions
- Multiple objects in single image
- Evaluation metric used:
    - Frechet Inception distance

Experiments and Results

COCO

- Contains 80k images for training and 40k images for testing
- Each image has 5 language descriptions
- Multiple objects in single image
- Evaluation metric used:
    - Frechet Inception distance

CUB - 200

- Contains 12k images belonging to 200 bird species
- Each bird image has 10 language descriptions
- 150 bird species with 9k images as training set and 50 species with 3k images as the test set.
- Evaluation metric used:
    - Inception score
    - Frechet Inception distance

Experiments and Results

- Optimizer used: Adam
- Learning rate:
  - Generator: 0.0001
  - Discriminator: 0.0004
- Epochs:
  - CUB-200 : 600
  - COCO: 120

Experiments and Results

- CUB
  - DF GAN performs outperforms previous methods in IS metric

Table 1. The results of IS, FID and NoP compared with the state-of-the-art methods on the test set of CUB and COCO.

Model	CUB		COCO	
	IS ↑	FID ↓	FID ↓	NoP ↓
StackGAN [56]	3.70	-	-	-
StackGAN++ [57]	3.84	-	-	
AttnGAN [50]	4.36	23.98	35.49	230M
MirrorGAN [33]	4.56	18.34	34.71	-
SD-GAN [51]	4.67	-	-	-
DM-GAN [60]	4.75	16.09	32.64	46M
CPGAN [22]	-	-	55.80	318M
XMC-GAN [55]	-	-	9.30	166M
DAE-GAN [39]	4.42	15.19	28.12	98M
TIME [26]	4.91	14.30	31.14	120M
DF-GAN (Ours)	5.10	14.81	19.32	19M

Experiments and Results

- CUB
  - DF GAN performs outperforms previous methods in IS metric
- COCO
  - DF-GAN performs decent enough in FID score
  - Uses significantly least parameters

Table 1. The results of IS, FID and NoP compared with the state-of-the-art methods on the test set of CUB and COCO.

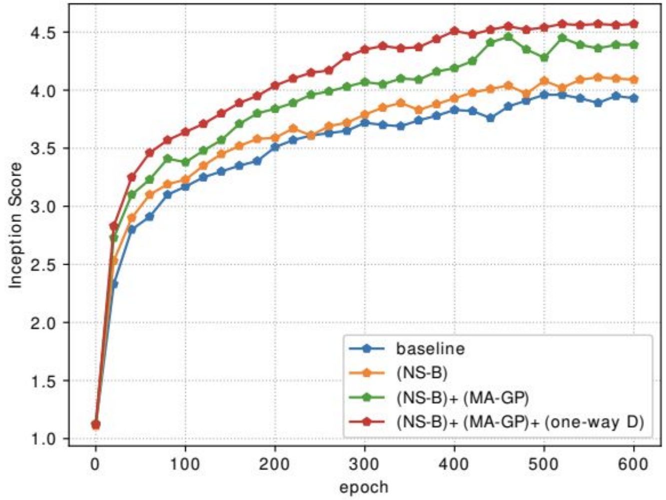
Model	CUB		COCO	
	IS ↑	FID ↓	FID ↓	NoP ↓
StackGAN [56]	3.70	-	-	-
StackGAN++ [57]	3.84	-	-	
AttnGAN [50]	4.36	23.98	35.49	230M
MirrorGAN [33]	4.56	18.34	34.71	-
SD-GAN [51]	4.67	-	-	-
DM-GAN [60]	4.75	16.09	32.64	46M
CPGAN [22]	-	-	55.80	318M
XMC-GAN [55]	-	-	9.30	166M
DAE-GAN [39]	4.42	15.19	28.12	98M
TIME [26]	4.91	14.30	31.14	120M
DF-GAN (Ours)	5.10	14.81	19.32	19M

Qualitative Results

Qualitative Results

Ablation studies

- Baseline: Stacked text-to-image GAN which employs two way discriminator
- One-Stage text-to-image Backbone (OSB)
- Matching-Aware Gradient Penalty (MA-GP)

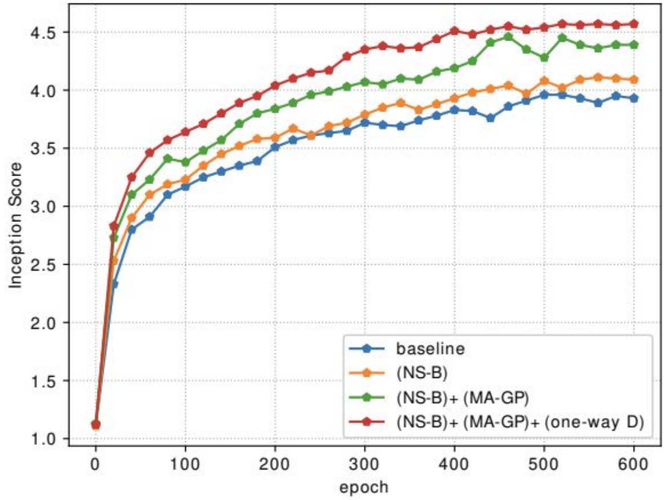


Ablation studies

- Baseline: Stacked text-to-image GAN which employs two way discriminator
- One-Stage text-to-image Backbone (OSB)
- Matching-Aware Gradient Penalty (MA-GP)

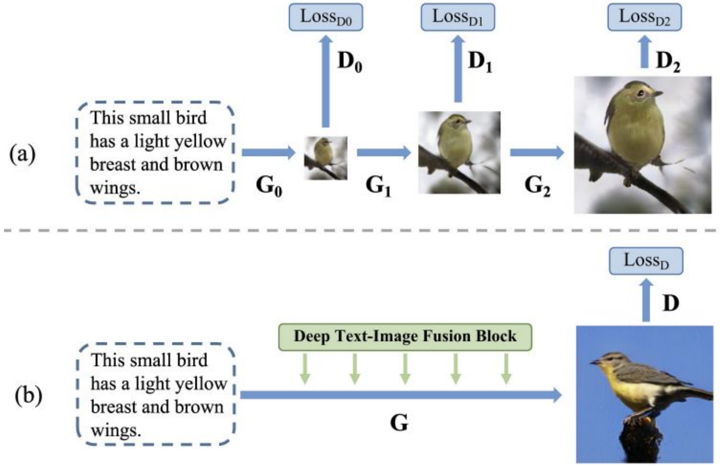
Table 2. The performance of different components of our model on the test set of CUB.

Architecture	IS ↑	FID ↓	SC ↑
Baseline	3.96	51.34	-
OS-B	4.11	43.45	1.46
OS-B w/ DAMSM	4.28	36.72	1.79
OS-B w/ MA-GP	4.46	32.52	3.55
OS-B w/ MA-GP w/ OW-O	**4.57**	**23.16**	**4.61**



Strengths

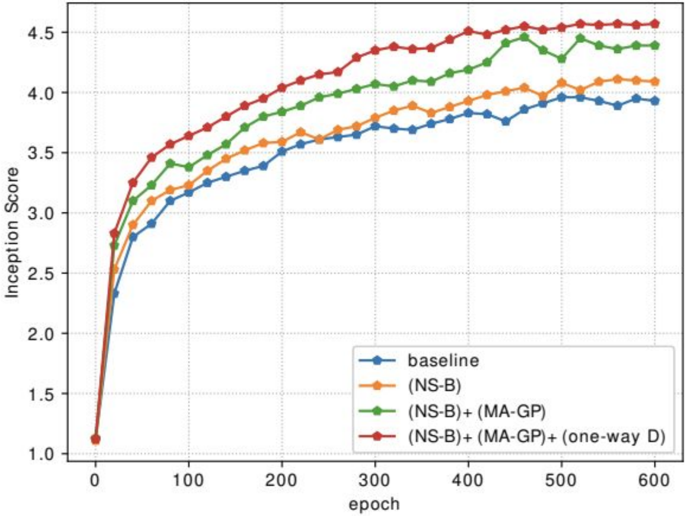
- Uses single G-D network - final image generation does not depend on initial images - prevents the generated image from getting trapped within previous context.



Strengths

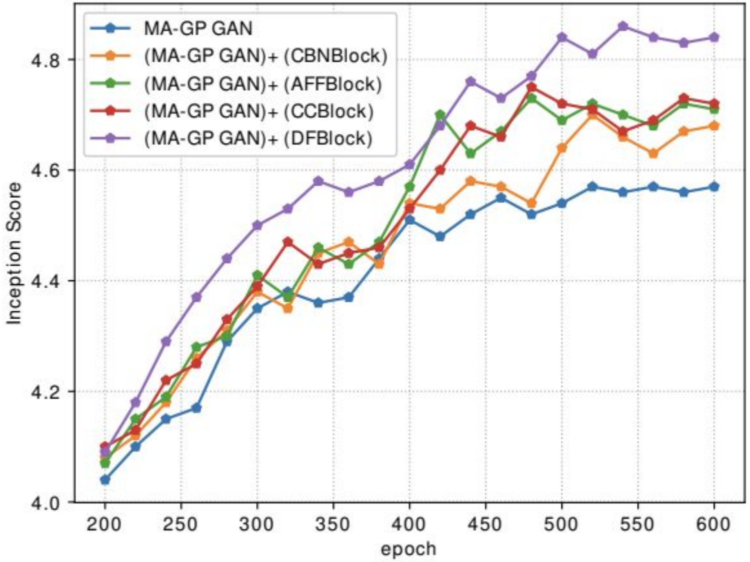
- Uses single G-D network - final image generation does not depend on initial images - prevents the generated image from getting trapped within previous context.

- Adding MA-GP and OB-B improves the performance consistently over the epochs - supports the hypothesis made in the paper



Strengths

- Uses single G-D network, so the final image generation does not depend on initial images. This prevents the generated image from not getting trapped within previous context.
- Adding MA-GP and OB-B improves the performance consistently over the epochs - supports the hypothesis made in the paper
- Normalization is computationally expensive. This paper proves that even slightly removing normalization increases performance.
- DFBlock consistently outperforms other modules like Concat, CBN, AFFBLK, etc throughout the epochs



Weakness

- The approach is trained on limited specific dataset i.e COCO and bird species. Difficult to have conclusive evidence on robustness of the model.

Weakness

- The approach is trained on limited specific dataset i.e COCO and bird species. Difficult to have conclusive evidence on robustness of the model.
- Inconsistencies in results. TIME has better FID score for CUB dataset. XMC-GAN has better FID score for COCO

Table 1. The results of IS, FID and NoP compared with the state-of-the-art methods on the test set of CUB and COCO.

Model	CUB		COCO	
	IS ↑	FID ↓	FID ↓	NoP ↓
StackGAN [56]	3.70	-	-	-
StackGAN++ [57]	3.84	-	-	
AttnGAN [50]	4.36	23.98	35.49	230M
MirrorGAN [33]	4.56	18.34	34.71	-
SD-GAN [51]	4.67	-	-	-
DM-GAN [60]	4.75	16.09	32.64	46M
CPGAN [22]	-	-	55.80	318M
XMC-GAN [55]	-	-	$\boxed{9.30}$	166M
DAE-GAN [39]	4.42	15.19	28.12	98M
TIME [26]	4.91	$\boxed{14.30}$	31.14	120M
DF-GAN (Ours)	5.10	14.81	19.32	19M

Weakness

- The approach is trained on limited specific dataset i.e COCO and bird species. Difficult to have conclusive evidence on robustness of the model.
- Inconsistencies in results. TIME has better FID score for CUB dataset. XMC-GAN has better FID score for COCO
- Can be difficult to interpret and identify how the model generates specific outputs and the edge cases where it fails.

Table 1. The results of IS, FID and NoP compared with the state-of-the-art methods on the test set of CUB and COCO.

Model	CUB		COCO	
	IS ↑	FID ↓	FID ↓	NoP ↓
StackGAN [56]	3.70	-	-	-
StackGAN++ [57]	3.84	-	-	
AttnGAN [50]	4.36	23.98	35.49	230M
MirrorGAN [33]	4.56	18.34	34.71	-
SD-GAN [51]	4.67	-	-	-
DM-GAN [60]	4.75	16.09	32.64	46M
CPGAN [22]	-	-	55.80	318M
XMC-GAN [55]	-	-	9.30	166M
DAE-GAN [39]	4.42	15.19	28.12	98M
TIME [26]	4.91	14.30	31.14	120M
DF-GAN (Ours)	5.10	14.81	19.32	19M

Future Work

- Evaluating the method for different domain specific text-to-image datasets.

- DF-GAN currently uses significantly lower parameters (19M) compared to other state-of-the-art methods. (> 100 M)

  Can the model further improve performance by simply scaling up the architecture ?

Discussion / Questions ?

Feel free to connect on LinkedIn!