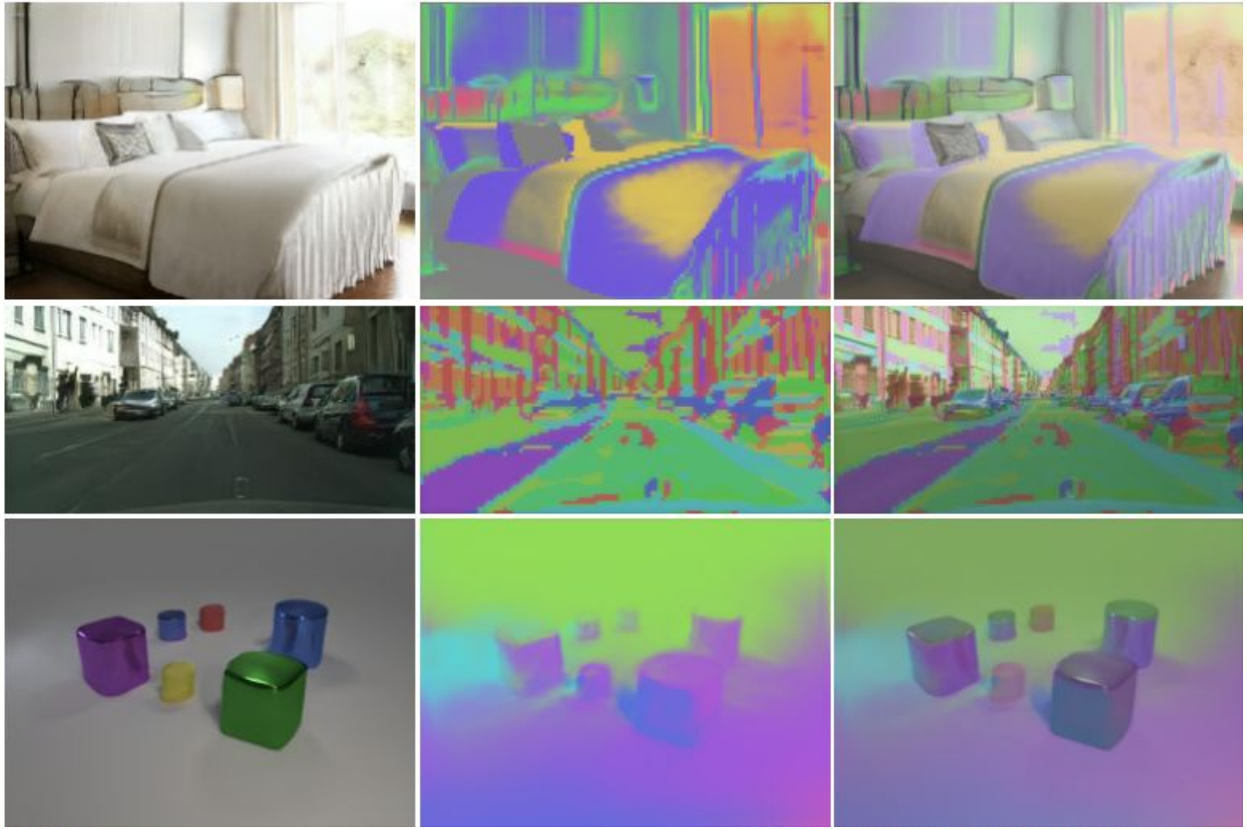
Generative Adversarial Transformer

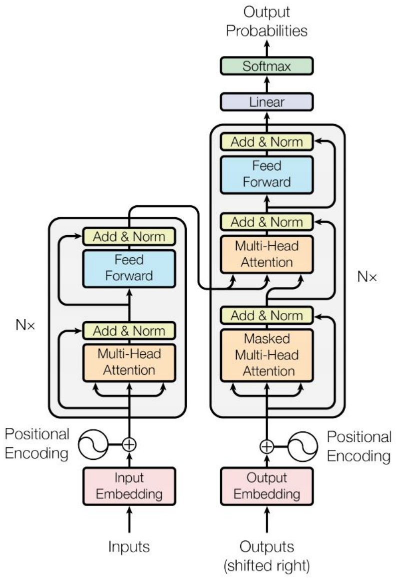
Drew A. Hudson and C. Lawrence Zitnick

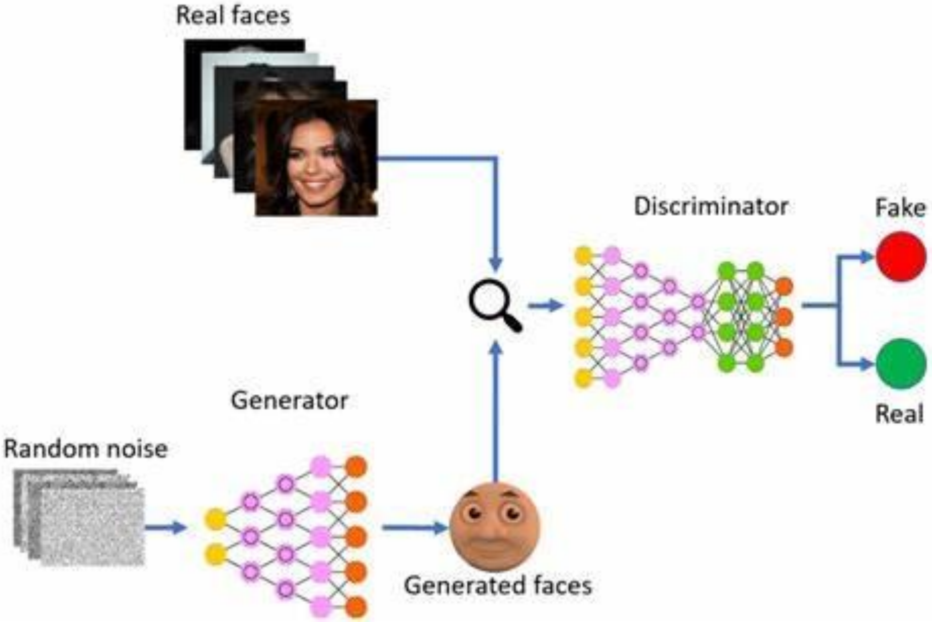
Presented by: Glanda Darie-Teofil

Research Question

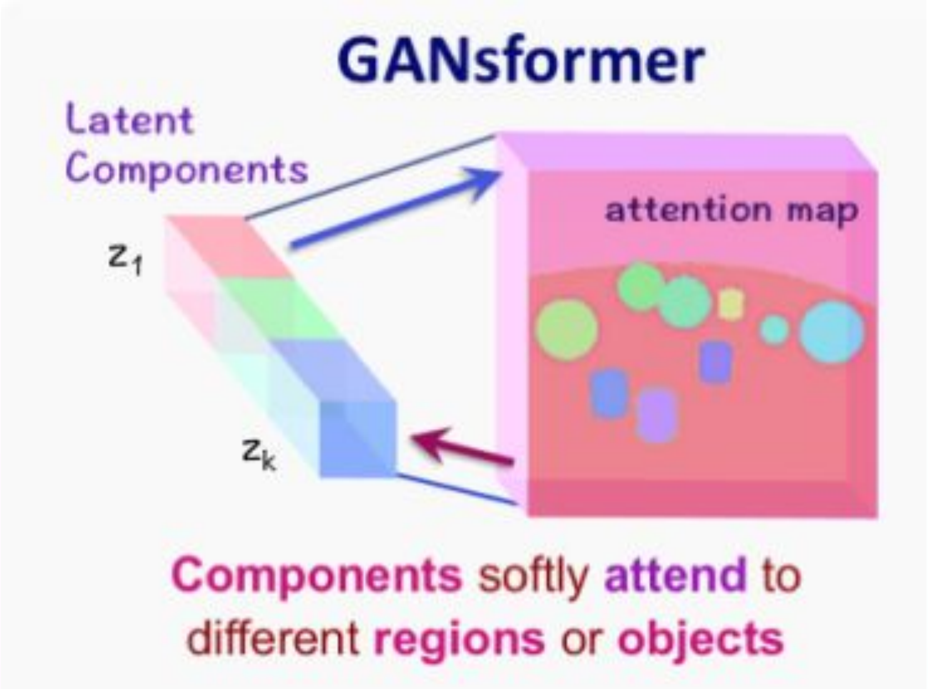


Literature Overview





GANformer Architecture



Generator

Discriminator

Bipartite Transformer

Bipartite Attention

X - *image features*   Y - *latent variables*

Form of attentions: *1. Simplex*

2. Duplex

NOTE: similarity score is computed using the dot product between image features and the latent variables.

Simplex Attention

$X^{n\times d}$ → n — number of features; d — number of channels

$Y^{m\times d}$ → m — number of latents; d — number of channels

Query: $Q_i = W_q \cdot x_i$

Key: $K_i = W_k \cdot y_i$

Value: $V_i = W_v \cdot y_i$

$$Attention(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$

$$a(X,Y) = Attention(q(X), k(Y), v(Y))$$

$$u^s(X,Y) = \gamma\left(a(X,Y)\right) \odot \omega(X) + \beta\left(a(X,Y)\right)$$

Duplex Attention

$X^{n\times d}$ → n — number of features; d — number of channels

$Y = (K^{m\times d}, V^{m\times d})$ → $\mathbf{V}$ — latent variables

$$K = a(Y, X)$$

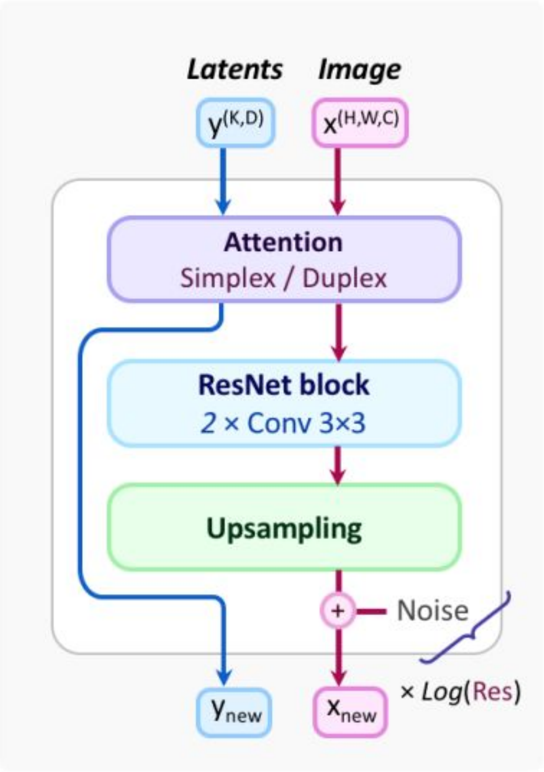
Query: $Q_i = W_q \cdot x_i$

$$Attention(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$

$$u^d(X,Y) = \gamma(A(Q,K,V)) \odot \omega(X) + \beta(A(Q,K,V))$$

Transformer Model Structure

- Does not use classical embedding as vanilla Transformer
- Uses the sinusoidal positional encoding
- Kernel size of 3 for the ResNet block after each self-attention layer.
- Leaky ReLU activation after each ResNet block
- Upsample or downsample the image for the generator



Computational Efficiency

When computing the similarity score:

$$\cancel{\mathcal{O}(n^2)} \longrightarrow \mathcal{O}(mn)$$

$m \longrightarrow$ in the range of 8–32

Experiments and Results

Model	CLEVR FID ↓	IS ↑	Precision ↑	Recall ↑	LSUN-Bedrooms FID ↓	IS ↑	Precision ↑	Recall ↑
GAN	25.02	2.17	21.77	16.76	12.16	2.66	52.17	13.63
k-GAN	28.29	2.21	22.93	18.43	69.90	2.41	28.71	3.45
SAGAN	26.04	2.17	30.09	15.16	14.06	2.70	54.82	7.26
StyleGAN2	16.05	2.15	28.41	23.22	11.53	**2.79**	51.69	19.42
GANformer$_s$	10.26	**2.46**	38.47	37.76	8.56	2.69	55.52	22.89
GANformer$_d$	**9.17**	2.36	**47.55**	**66.63**	**6.51**	2.67	**57.41**	**29.71**

Model	FFHQ FID ↓	IS ↑	Precision ↑	Recall ↑	Cityscapes FID ↓	IS ↑	Precision ↑	Recall ↑
GAN	13.18	4.30	67.15	17.64	11.57	1.63	61.09	15.30
k-GAN	61.14	4.00	50.51	0.49	51.08	1.66	18.80	1.73
SAGAN	16.21	4.26	64.84	12.26	12.81	1.68	43.48	7.97
StyleGAN2	9.24	4.33	68.61	**25.45**	8.35	**1.70**	59.35	27.82
GANformer$_s$	8.12	**4.46**	**68.94**	10.14	14.23	1.67	**64.12**	2.03
GANformer$_d$	**7.42**	4.41	68.77	5.76	**5.76**	1.69	48.06	**33.65**

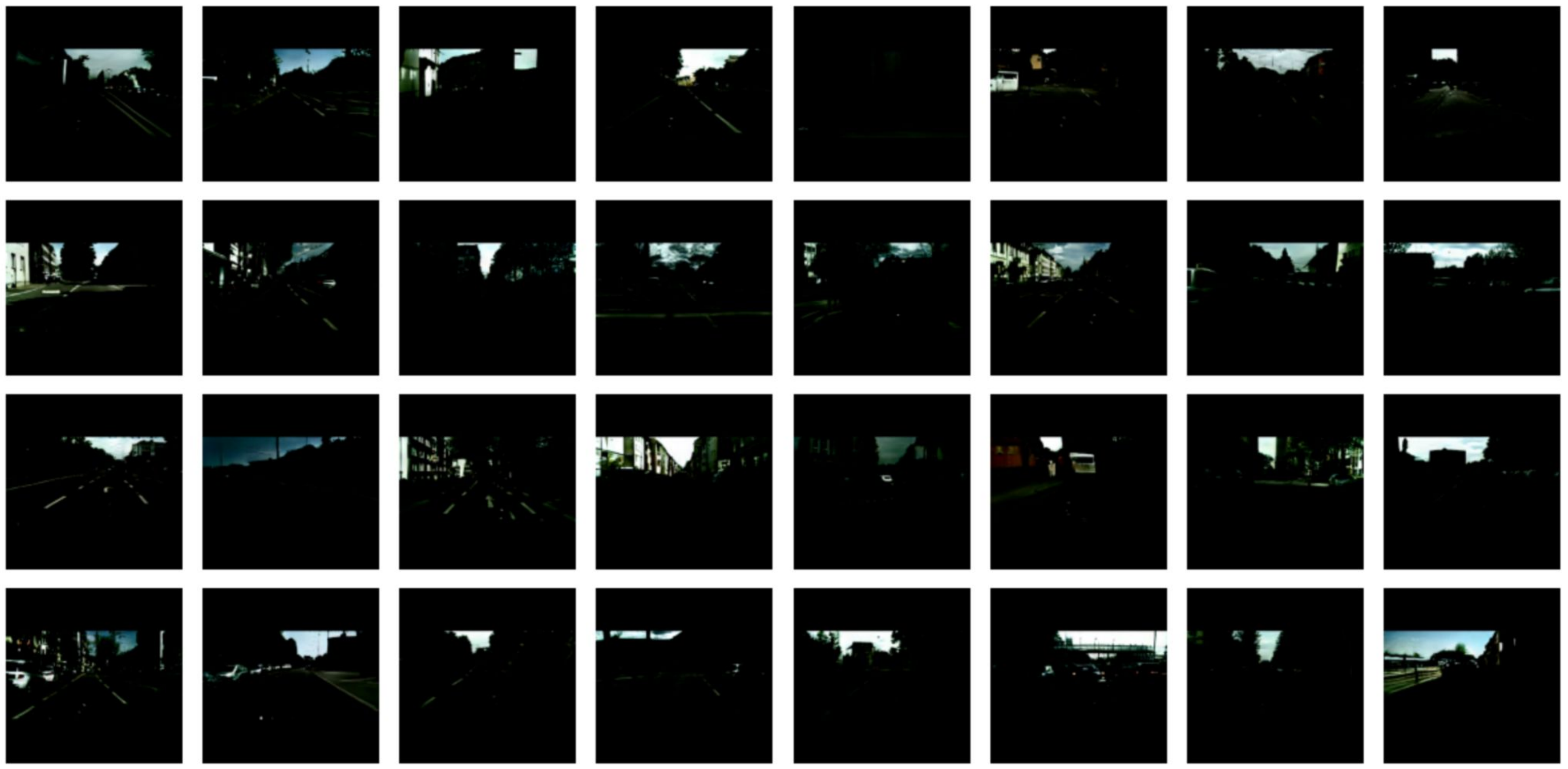
My Experiments

LSUN-Bedrooms

My Experiments

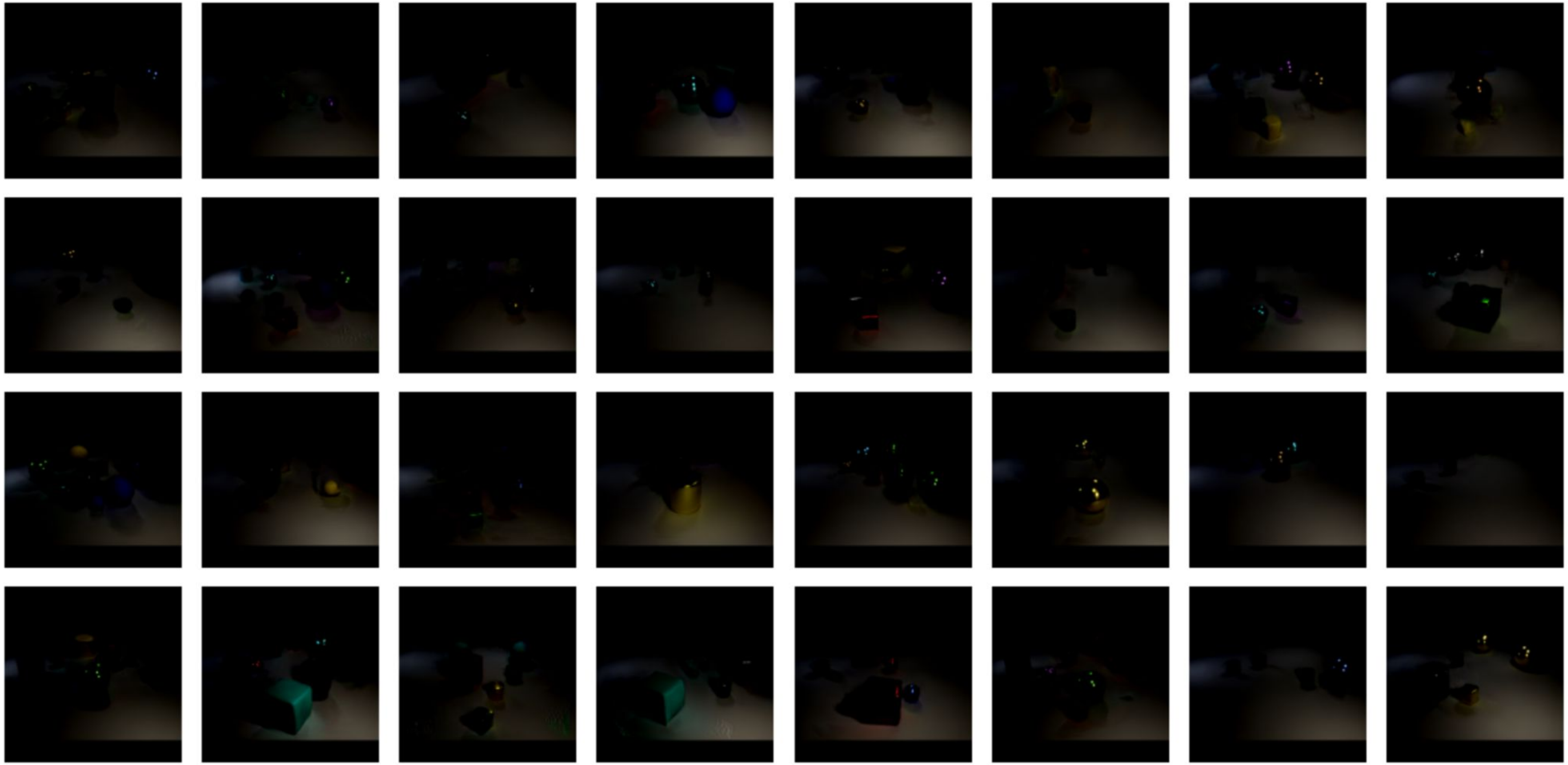
FFHQ

My Experiments



Cityscapes

My Experiments



CLEVR

Conclusions

- The authors introduced the GANformer, an efficient bipartite transformer that combines top-down and bottom-up interactions, and explored it for the task of generative modeling.
- Fits well within the general philosophy that aims to incorporate stronger biases into the Neural Networks, to encourage desirable properties such as transparency, data-efficiency and co.
- While GANformer's primary focus is generative modeling, its potential extends well beyond. It is equally suited for tasks across both Natural Language Processing (NLP) and Computer Vision (CV), offering adaptability and powerful performance.
- Achieves state-of-the-art performance in the context of image generation and manipulation, particularly in the task of generating images with high compositionality and layout diversity.

Thank you!