Self-Attention Generative Adversarial Networks

Han Zhang, Ian Goodfellow, Dimitris Metaxas and Augustus Odena

NIPS18

Presented by Wenjing Wang

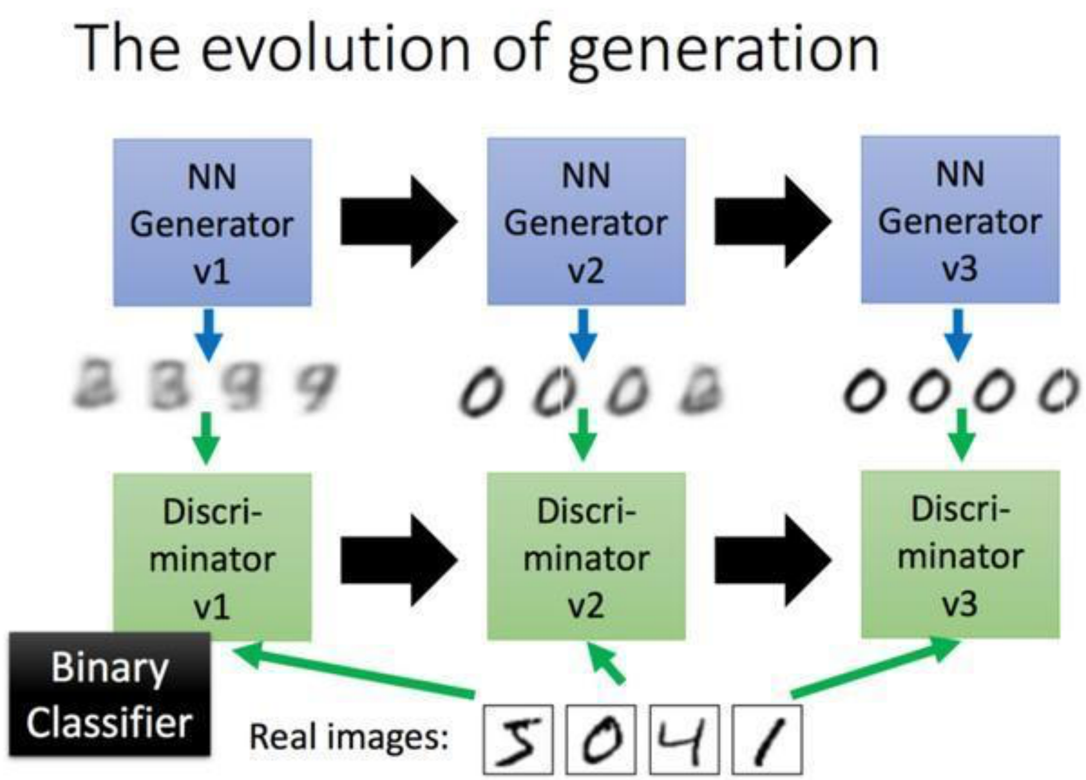
STRUCT Group Seminar

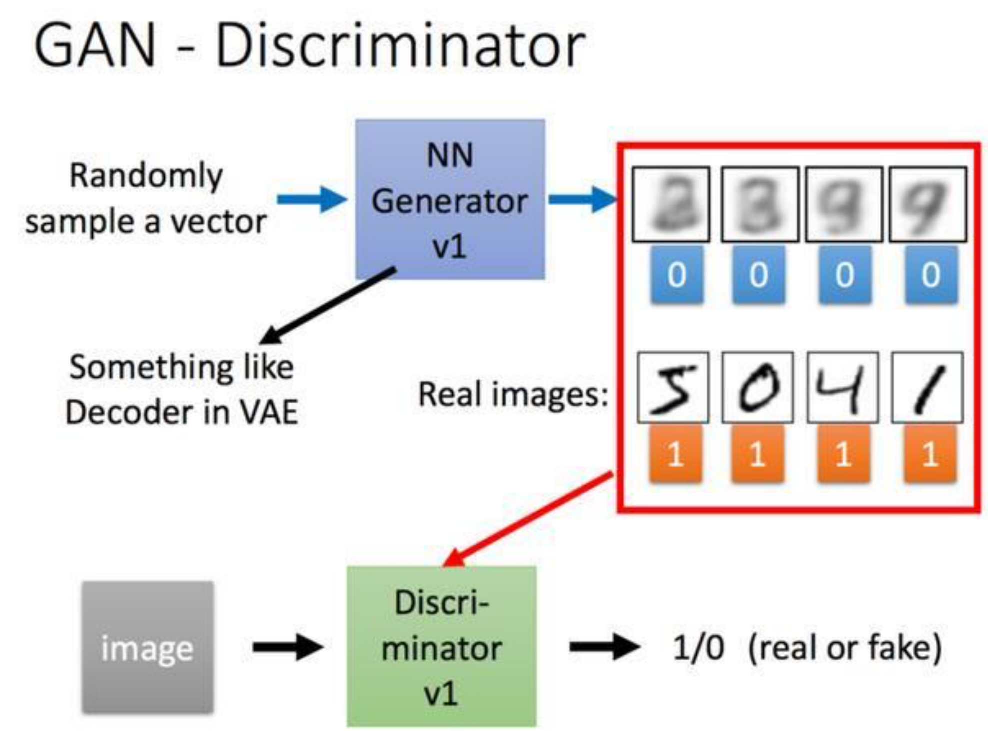
CONTENT

- Authorship
- Background
- Method
- Experiments
- Discussion and Conclusion

BACKGROUND

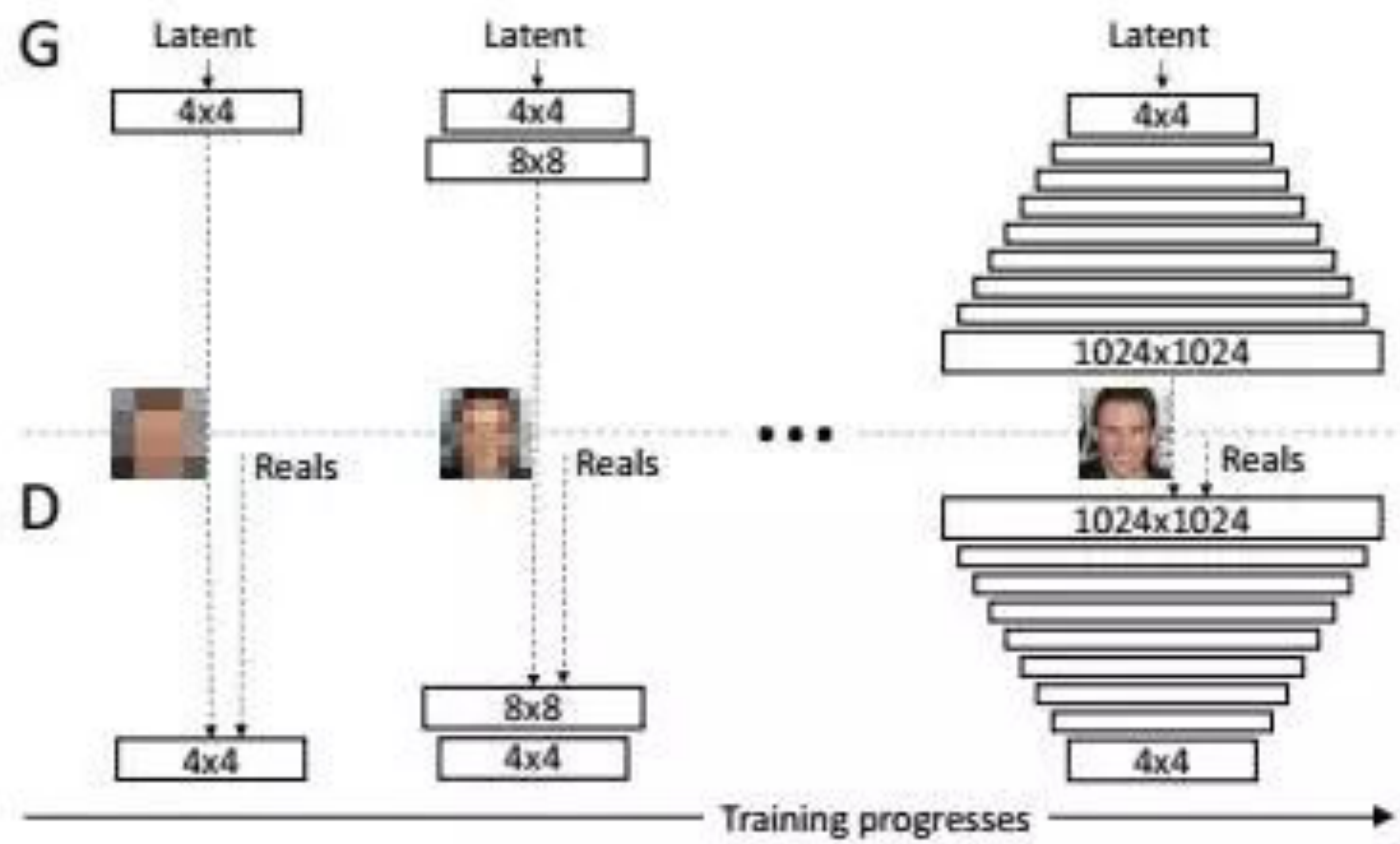
- Generative Adversarial Network (GAN)





BACKGROUND

- To improve GAN:
  - Designing New Network Architectures
    - Example: Progressive Growing (ICLR18)



- Progressive growing of gans for improved quality,stability, and variation

BACKGROUND

➤ To improve GAN:

- Modifying the Learning Objectives and Dynamics
  - Example: EBGAN (ICLR17)

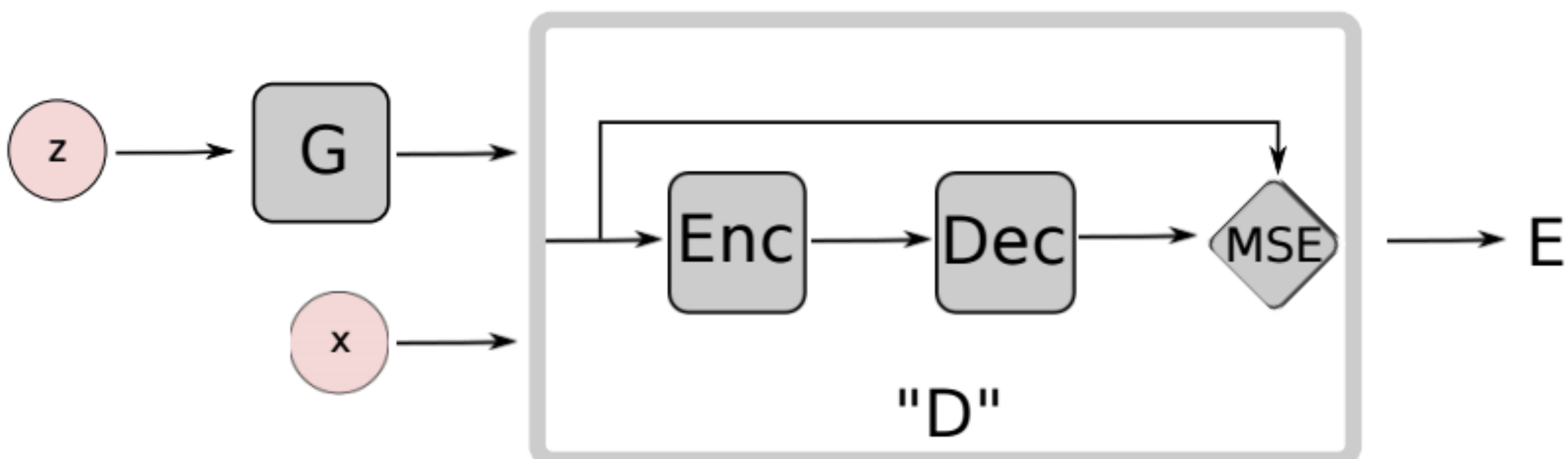


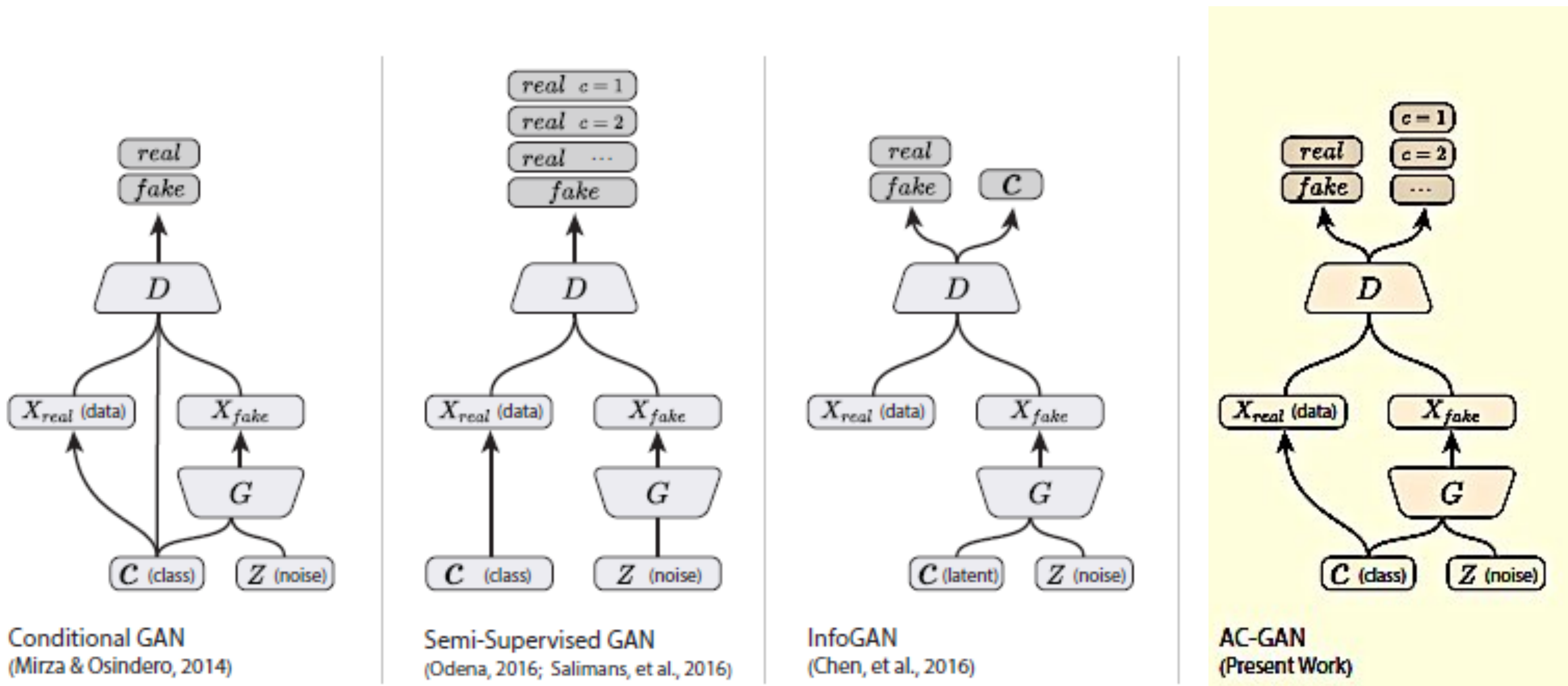
Figure 1: EBGAN architecture.

- Energy-based generative adversarial network

BACKGROUND

➤ To improve GAN:

- Introducing Heuristic Tricks
  - Example: ACGAN (ICLR17)



- Conditional image synthesis with auxiliary classifier gans

BACKGROUND

- To improve GAN:
  - Adding Regularization Methods
    - Example: WGAN-GP (NIPS17)

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

- Improved training of wasserstein gans

CONTENT

- Authorship
- Background
- Method
- Experiments
- Discussion and Conclusion

METHOD

- Self-Attention
- Spectral Norm
- TTUR

METHOD

- Previous Methods:
  - Good at classes with few structural constraints, e.g. ocean, sky
  - Fails to capture geometric or structural patterns (dogs are with realistic fur texture but without clearly defined separate feet)
- Causes:
  - receptive fields grow slowly

    → need deeper architecture → computational efficiency
  - optimization algorithms not good enough

    → cannot discover parameter values that carefully coordinate multiple layers to capture these dependencies

METHOD

- Self-Attention:
  - Well-used in NLP.
  - Image transformer (ICLR18) use it for image generation.
  - Non-local neural networks (CVPR18) proposed a non-local operation for video recognition.
  - Has not yet been explored in the context of GANs.

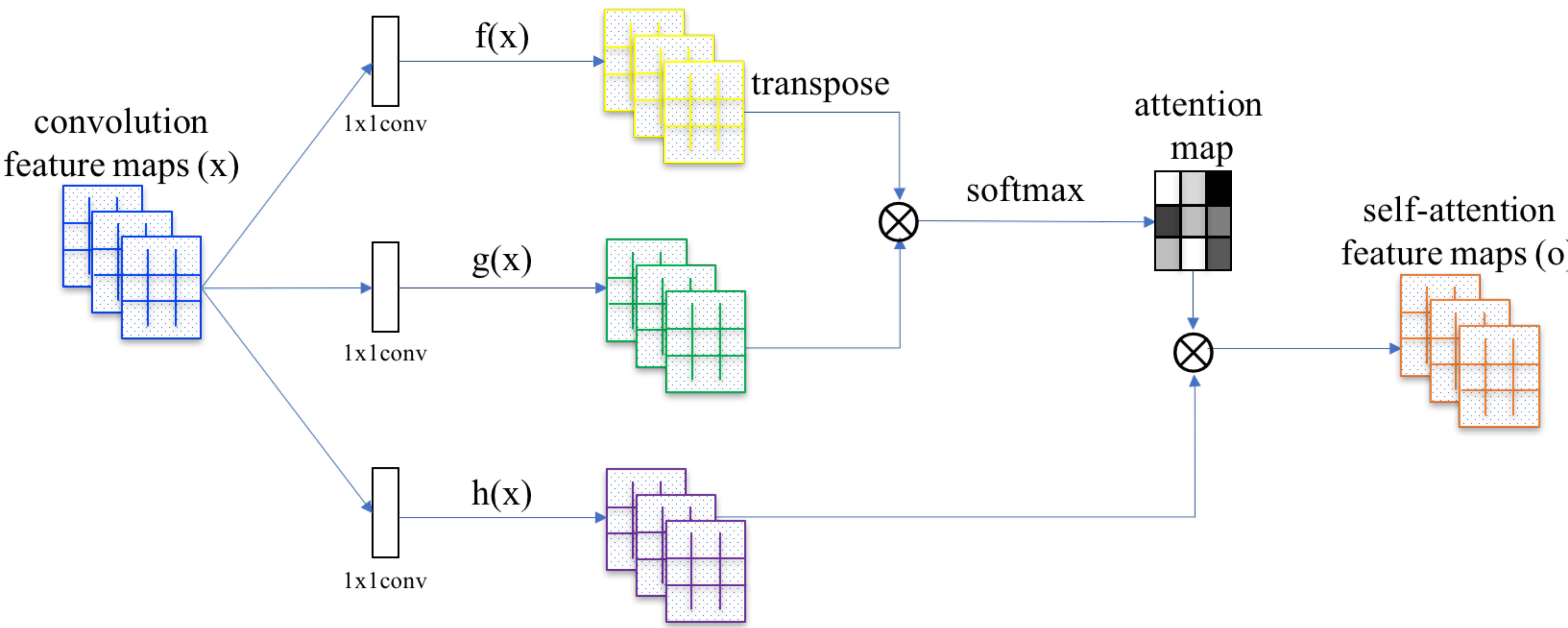
METHOD

- ➤ Self-Attention:
- ➤ Better balance long-range reception and efficiency

- Using features in distant portions of the image rather than local regions of fixed shape
- With only a small computational cost.

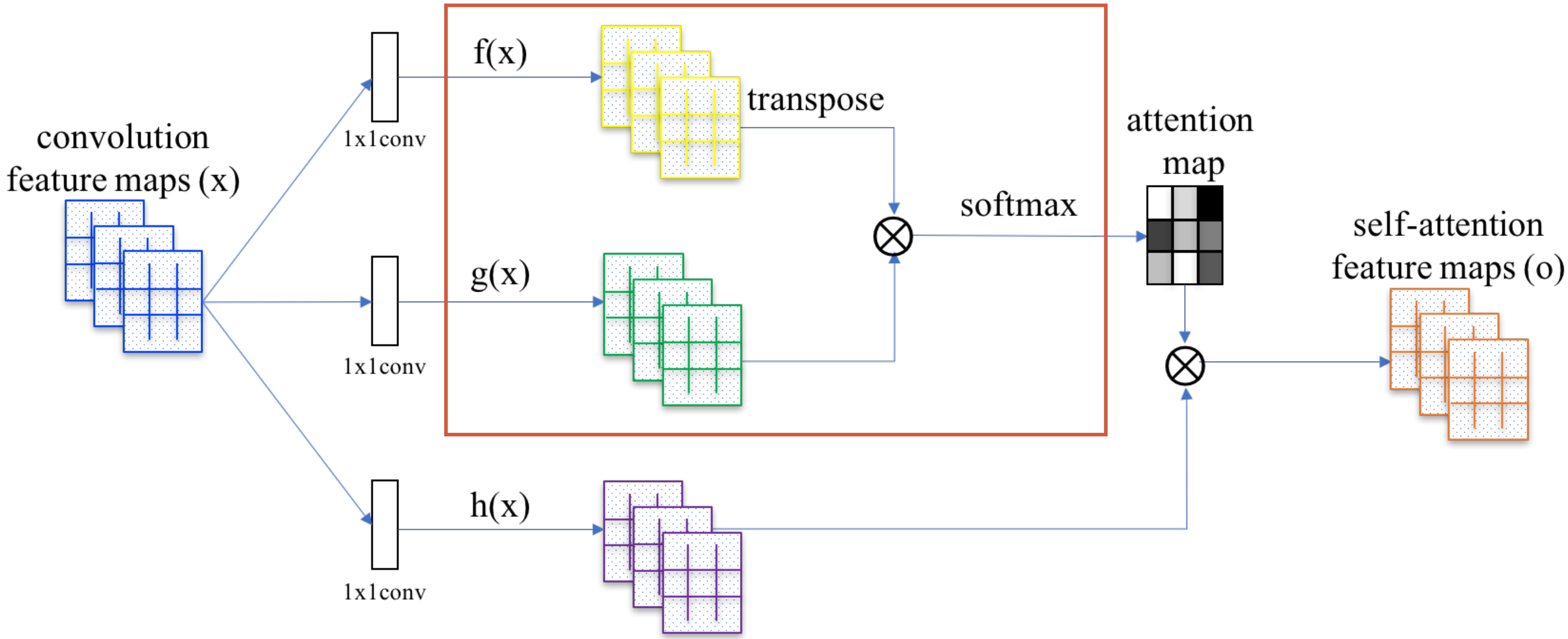
METHOD

➤ Self-Attention:



METHOD

➤ Self-Attention:

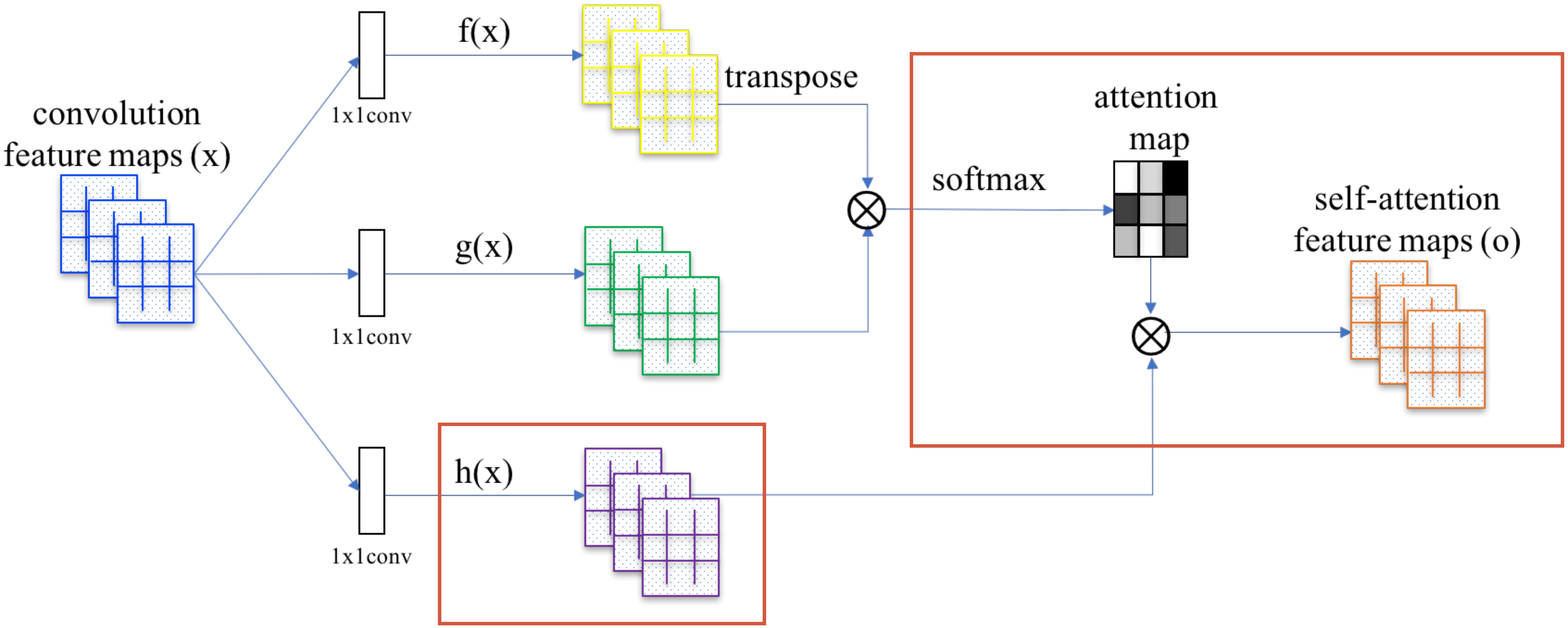


$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W_f x},\ \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{W_g x}\ ,\ \ \boldsymbol{W_g} \in \mathbb{R}^{\bar{C}\times C},\ \boldsymbol{W_f} \in \mathbb{R}^{\bar{C}\times C},\ \ \boldsymbol{x} \in \mathbb{R}^{C\times N}$$

$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N}\exp(s_{ij})}, \text{where } s_{ij} = \boldsymbol{f}(\boldsymbol{x_i})^T\boldsymbol{g}(\boldsymbol{x_j}),$$

METHOD

➤ Self-Attention:



$$\boldsymbol{W_h} \in \mathbb{R}^{C\times C}, \ \boldsymbol{o} = (\boldsymbol{o_1}, \boldsymbol{o_2}, ..., \boldsymbol{o_j}, ..., \boldsymbol{o_N}) \in \mathbb{R}^{C\times N},$$

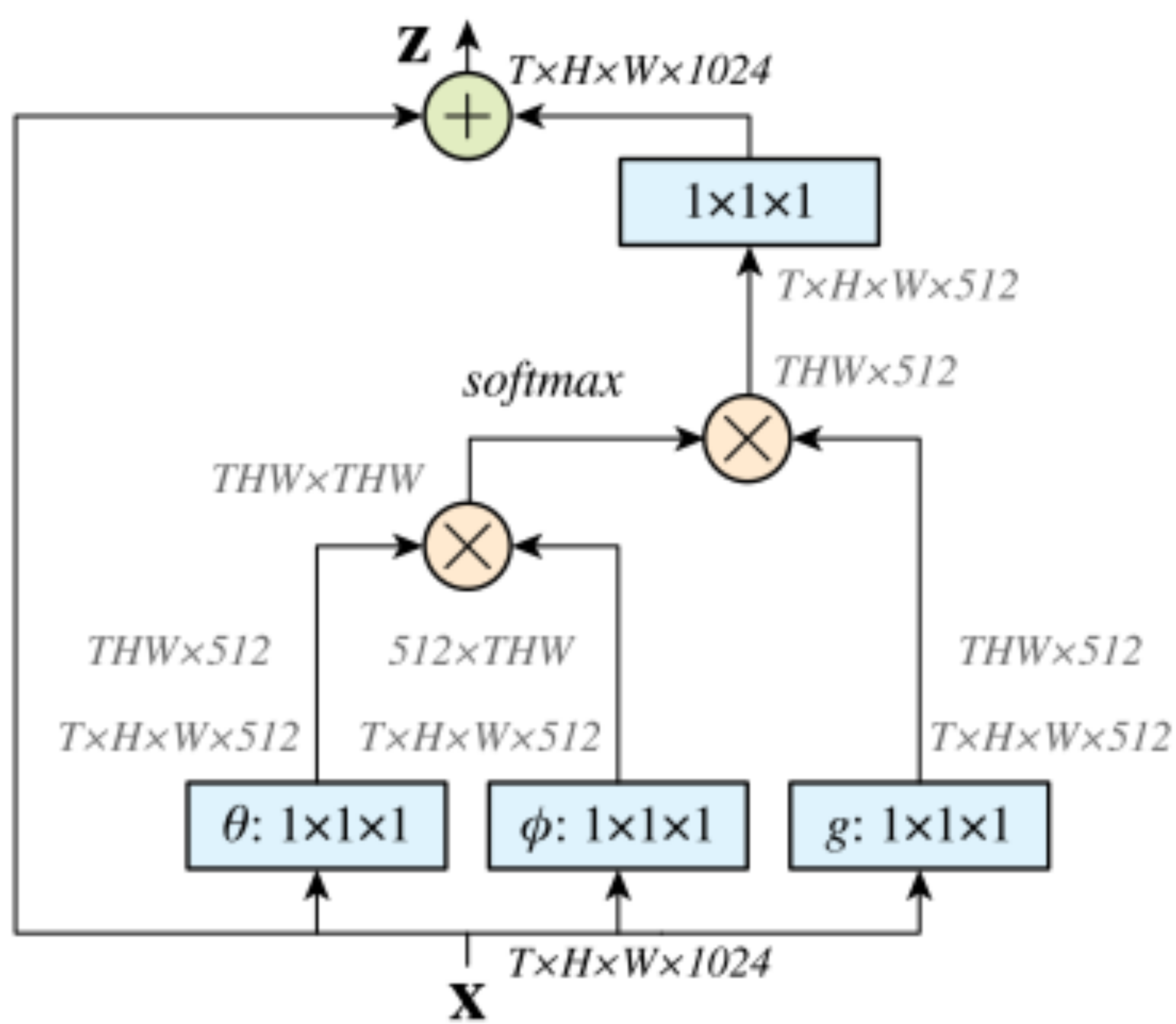
$$\boldsymbol{o_j} = \sum_{i=1}^{N} \beta_{j,i} \boldsymbol{h}(\boldsymbol{x_i}), \text{where } \boldsymbol{h}(\boldsymbol{x_i}) = \boldsymbol{W_h x_i}.$$

METHOD

- Self-Attention:
- Final Output: $\boldsymbol{y_i} = \gamma \boldsymbol{o_i} + \boldsymbol{x_i}$, γ is initialized as 0
  - The network first relies on local neighborhood
  - Then gradually learn to use the non-local evidence

METHOD

- Self-Attention:
  - Non-local neural networks (CVPR18) proposed a non-local operation for video recognition.



METHOD

- Self-Attention
- Spectral Norm
- TTUR

METHOD

- SNGAN(ICLR18) only uses spectral normalization on netD
- This paper use it for both netG and netD
- Benefit: fewer netD updates per netG updates

- Spectral normalization for generative adversarial networks (ICLR18)

METHOD

- Spectral Norm
- limit spectral norm of the weight matrices in the netD

  $\rightarrow$ to constrain the Lipschitz constant of the netD
- By definition, Lipschitz norm $\|g\|_{\mathrm{Lip}} = \sup_{\boldsymbol{h}} \sigma(\nabla g(\boldsymbol{h}))$

$$\sigma(A) := \max_{\boldsymbol{h}:\boldsymbol{h}\neq\boldsymbol{0}} \frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2} = \max_{\|\boldsymbol{h}\|_2\leq 1} \|A\boldsymbol{h}\|_2,$$

which is equivalent to the largest singular value of A

- Spectral normalization for generative adversarial networks (ICLR18)

METHOD

- Spectral Norm
- limit spectral norm of the weight matrices in the netD

  $\rightarrow$ to constrain the Lipschitz constant of the netD

- By definition, Lipschitz norm $\|g\|_{\mathrm{Lip}} = \sup_{\boldsymbol{h}} \sigma(\nabla g(\boldsymbol{h}))$

$$\sigma(A) := \max_{\boldsymbol{h}:\boldsymbol{h}\neq\boldsymbol{0}} \frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2} = \max_{\|\boldsymbol{h}\|_2\leq 1} \|A\boldsymbol{h}\|_2,$$

which is equivalent to the largest singular value of A

- for a linear layer $g(\boldsymbol{h}) = W\boldsymbol{h}$,

$$\|g\|_{\mathrm{Lip}} = \sup_{\boldsymbol{h}} \sigma(\nabla g(\boldsymbol{h})) = \sup_{\boldsymbol{h}} \sigma(W) = \sigma(W).$$

- Spectral normalization for generative adversarial networks (ICLR18)

METHOD

- Spectral Norm
- limit spectral norm of the weight matrices in the netD

  $\rightarrow$ to constrain the Lipschitz constant of the netD

- By definition, Lipschitz norm $\|g\|_{\mathrm{Lip}} = \sup_{\boldsymbol{h}} \sigma(\nabla g(\boldsymbol{h}))$

$$\sigma(A) := \max_{\boldsymbol{h}:\boldsymbol{h}\neq\boldsymbol{0}} \frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2} = \max_{\|\boldsymbol{h}\|_2\leq 1} \|A\boldsymbol{h}\|_2,$$

which is equivalent to the largest singular value of A

- for a linear layer $g(\boldsymbol{h}) = W\boldsymbol{h}$,

norm it to 1

$$\|g\|_{\mathrm{Lip}} = \sup_{\boldsymbol{h}} \sigma(\nabla g(\boldsymbol{h})) = \sup_{\boldsymbol{h}} \sigma(W) = \boxed{\sigma(W).}$$

- Spectral normalization for generative adversarial networks (ICLR18)

METHOD

- Spectral Norm
- Benefit:
  - Fewer netD updates per netG updates
  - Does not require extra hyper-parameter tuning

- Spectral normalization for generative adversarial networks (ICLR18)

METHOD

- Self-Attention
- Spectral Norm
- TTUR

METHOD

- Two-Timescale Update Rule (TTUR)
- Learning rate of netD : netG = 4:1 (0.0004 and 0.0001)
- Benefit: fewer netD updates per netG updates

- Gans trained by a two time-scale update rule converge to a local nash equilibrium (NIPS17)

CONTENT

- Authorship
- Background
- Method
- Experiments
- Discussion and Conclusion

EVALUATION METRICS

- Inception score $\mathrm{IS}(\mathbb{P}_g) = e^{\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_g}[KL(p_{\mathcal{M}}(y|\mathbf{x})||p_{\mathcal{M}}(y))]},$
    - KL divergence between the conditional class distribution and the marginal class distribution
    - Higher the better
    - cannot assess realism of details or intra-class diversity
- FID $\mathrm{FID}(\mathbb{P}_r, \mathbb{P}_g) = \|\mu_r - \mu_g\| + \mathrm{Tr}(\mathbf{C}_r + \mathbf{C}_g - 2(\mathbf{C}_r\mathbf{C}_g)^{1/2}),$
    - Wasserstein-2 distance in the feature of an Inception-v3.
    - Lower the better

NETWORK STRUCTURES

- Resolution: 128×128
- netG:

    Block → Block → Block → SA → Block → SA → Last

    Block: DeConv - Spectral Norm - BN - ReLU

    Last: DeConv - Tanh

- netD:

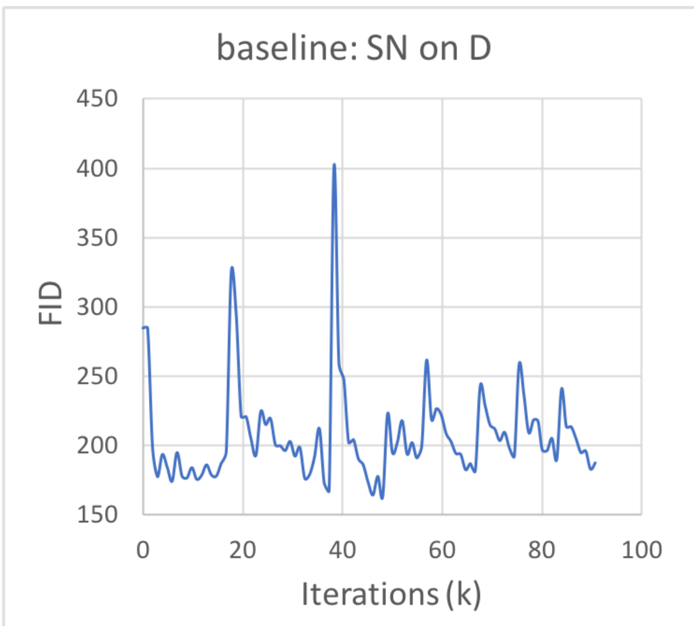
    Block → Block → Block → SA → Block → SA → Last

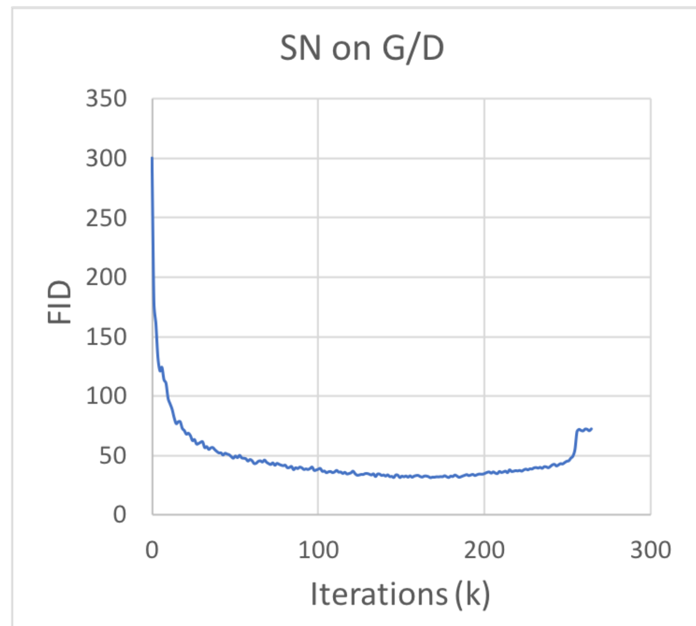
    Block: Conv - Spectral Norm - LeakyReLU

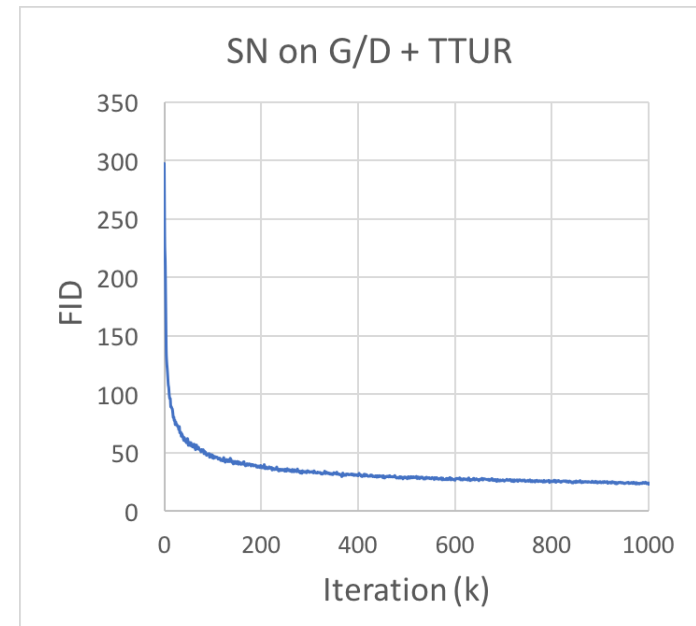
    Last: Conv

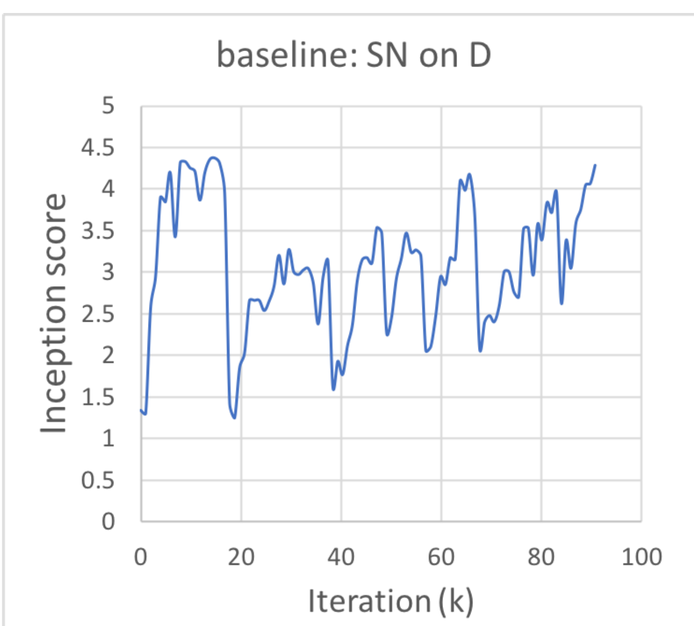
SN AND TTUR

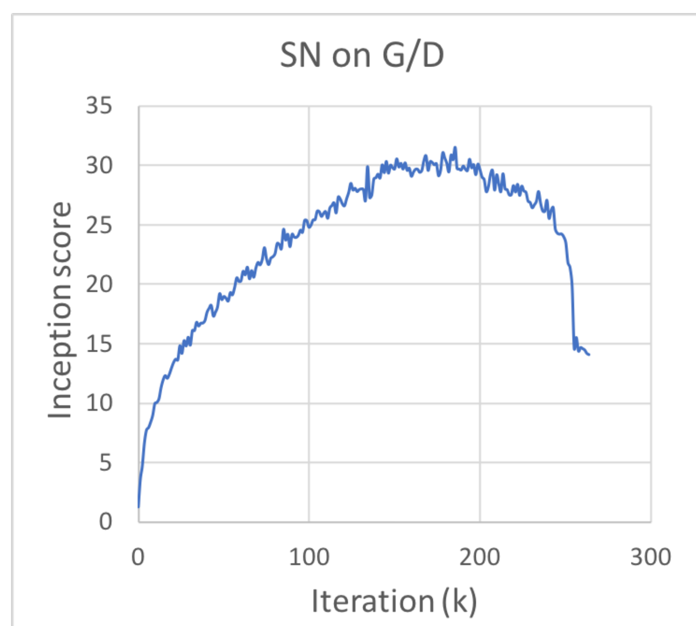
SN on D | *SN on G/D* | *SN on G/D + TTUR*

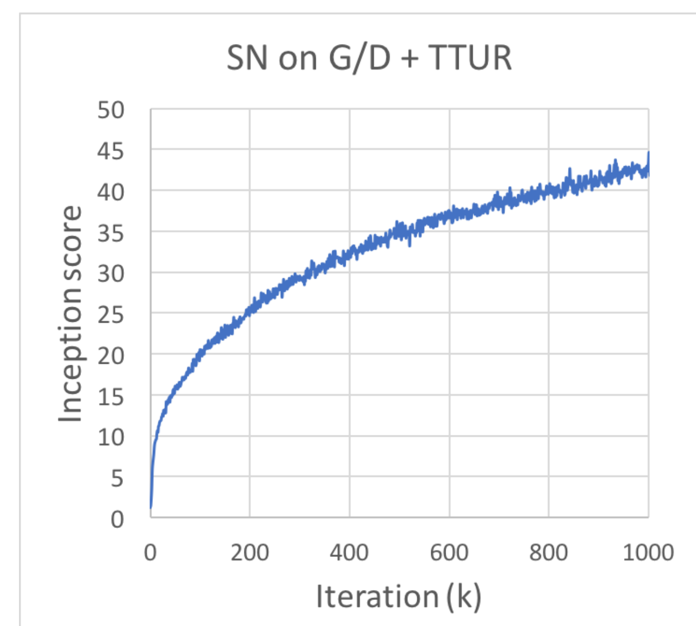












SELF ATTENTION

- self-attention mechanism:
- better at the middle-to-high level feature maps (e.g., feat32 and feat64) than at the low level feature maps (e.g., feat8 and feat16).

Model	no attention	SAGAN				Residual			
		$feat_8$	$feat_{16}$	$feat_{32}$	$feat_{64}$	$feat_8$	$feat_{16}$	$feat_{32}$	$feat_{64}$
FID	22.96	22.98	22.14	**18.28**	18.65	42.13	22.40	27.33	28.82
IS	42.87	43.15	45.94	51.43	**52.52**	23.17	44.49	38.50	38.96

Table 1: Comparison of Self-Attention and Residual block on GANs. These blocks are added into different layers of the network. All models have been trained for one million iterations, and the best Inception scores (IS) and Fréchet Inception distance (FID) are reported.

SELF ATTENTION

- self-attention mechanism:

Figure 5: Visualization of attention maps. These images were generated by SAGAN. We visualize the attention maps of the last generator layer that used attention, since this layer is the closest to the output pixels and is the most straightforward to project into pixel space and interpret. In each cell, the first image shows three representative query locations with color coded dots. The other three images are attention maps for those query locations, with corresponding color coded arrows summarizing the most-attended regions. We observe that the network learns to allocate attention according to

COMPARATIVE RESULTS

Model	Inception Score	FID
AC-GAN [31]	28.5	/
SNGAN-projection [17]	36.8	27.62*
SAGAN	**52.52**	**18.65**

Table 2: Comparison of the proposed SAGAN with state-of-the-art GAN models [19, 17] for class conditional image generation on ImageNet. FID of SNGAN-projection is calculated from officially released weights.

saint bernard

tiger cat

stone wall

CONTENT

- Authorship
- Background
- Method
- Experiments
- Discussion and Conclusion

DISCUSSION AND CONCLUSION

- Self-Attention $\rightarrow$ better global structure, higher score
- Spectral Norm
- TTUR

$\rightarrow$ stability

DISCUSSION AND CONCLUSION

➤ BigGAN uses SAGAN as Baseline:

Batch	Ch.	Param (M)	Shared	Hier.	Ortho.	Itr $\times 10^3$	FID	IS
256	64	81.5	SA-GAN Baseline			1000	18.65	52.52
512	64	81.5	✗	✗	✗	1000	15.30	58.77($\pm$1.18)
1024	64	81.5	✗	✗	✗	1000	14.88	63.03($\pm$1.42)
2048	64	81.5	✗	✗	✗	732	12.39	76.85($\pm$3.83)
2048	96	173.5	✗	✗	✗	295($\pm$18)	9.54($\pm$0.62)	92.98($\pm$4.27)
2048	96	160.6	✓	✗	✗	185($\pm$11)	9.18($\pm$0.13)	94.94($\pm$1.32)
2048	96	158.3	✓	✓	✗	152($\pm$7)	8.73($\pm$0.45)	98.76($\pm$2.84)
2048	96	158.3	✓	✓	✓	165($\pm$13)	8.51($\pm$0.32)	99.31($\pm$2.10)
2048	64	71.3	✓	✓	✓	371($\pm$7)	10.48($\pm$0.10)	86.90($\pm$0.61)