Self-Attention Generative Adversarial Networks

$Han\ Zhang, et\ al.$, arXiv

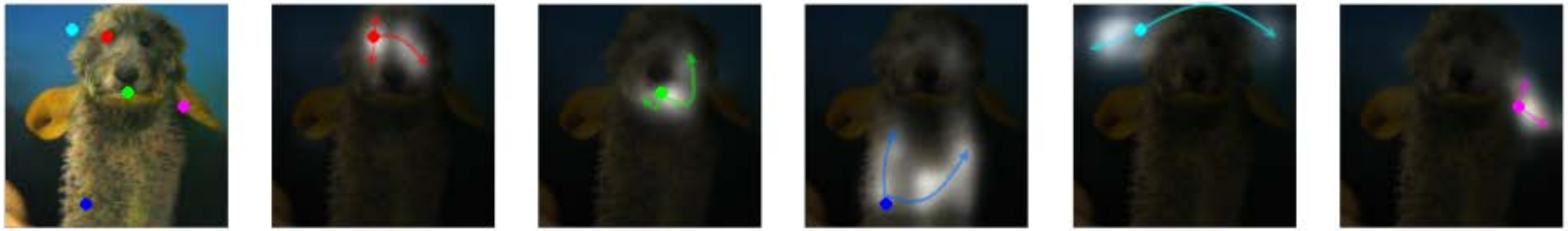
2019/02/19, KangYeol Kim

Motivation

- Previous GAN models bear the limitation that they **fail to capture geometric or structural patterns** that occur consistently in some classes (for example, dogs are often drawn with realistic fur texture but without clearly defined separate feet).

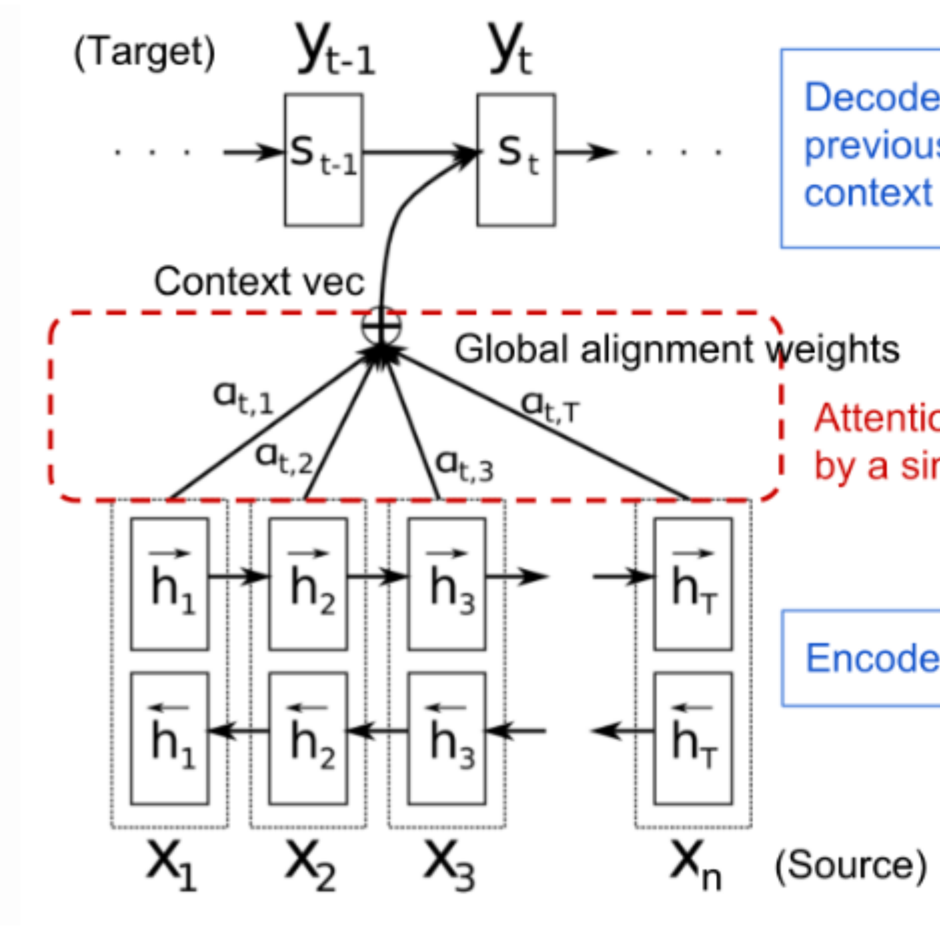
- One possible explanation for this is it mainly due to limited range of receptive field, thus model **cannot learn about long-term dependencies** (i.e. relationship between distant regions)

- The self-attention module calculates response at a position **as a weighted sum of the features at all positions**, where the weights – or attention vectors – are calculated with only a small computational cost.



The most-attended regions depending on each different colored query

What is Attention in NMT ?



Decoder: RNN with input from previous state + dynamic context vector.

$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \boldsymbol{h}_i$$

Attention layer: parameterized by a simple feed-forward network

$$\alpha_{t,i} = \text{align}(y_t, x_i) = \frac{\exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_i))}{\sum_{i'=1}^{n} \exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_{i'}))}$$

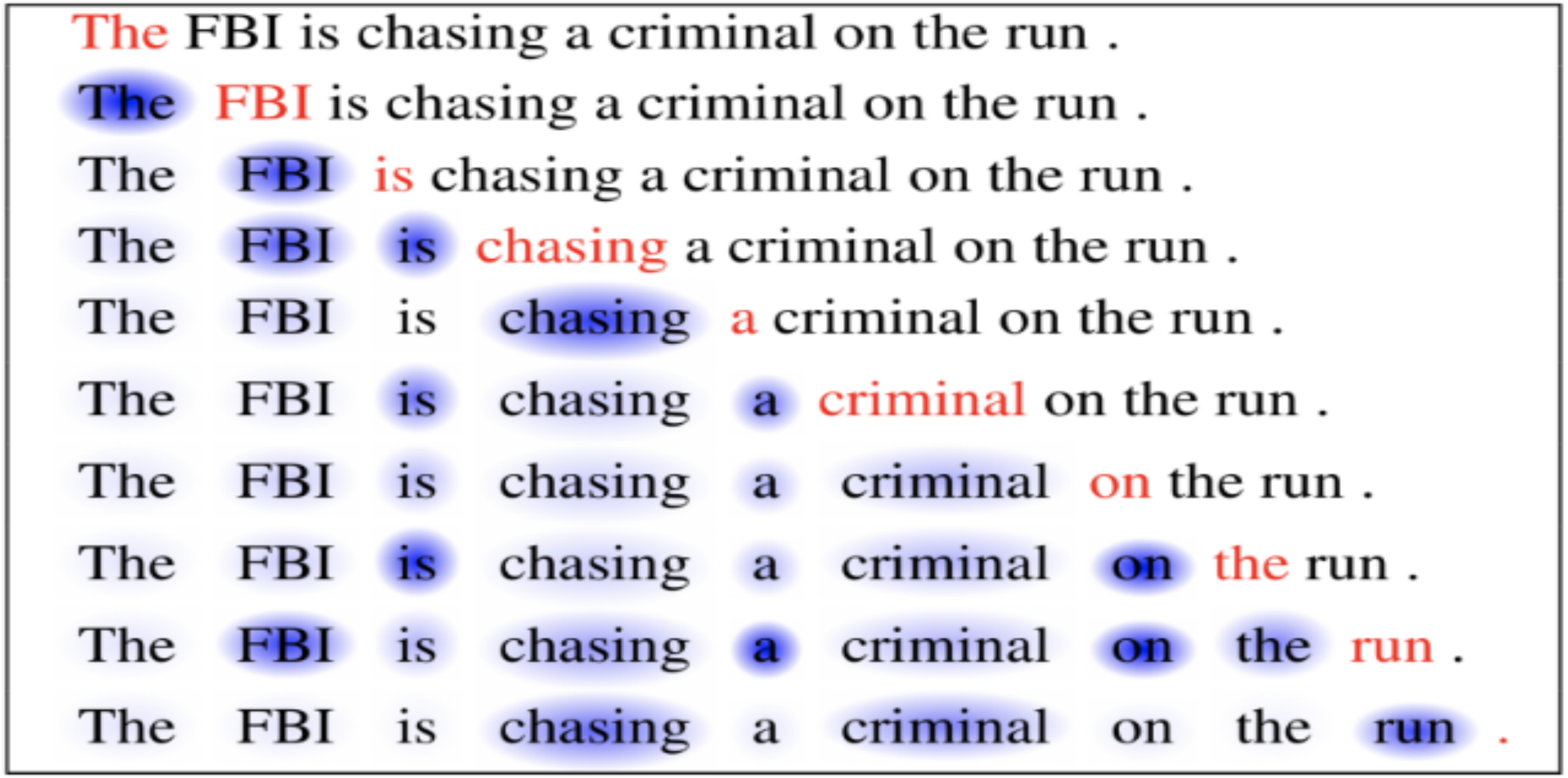
Encoder: bidirectional RNN

$$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$$

SO SO MANY score functions!!!

What is Self-Attention in NMT ?

- Machine can learn **where to attend** by the relationship with other words in the sentence.
- We can represent same words differently based on the contexts.(Same in images)

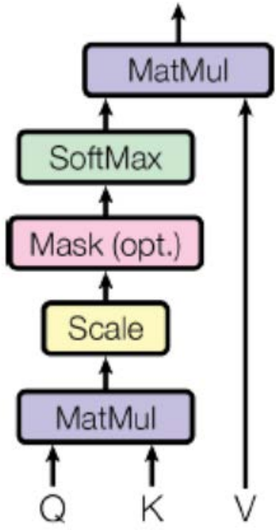


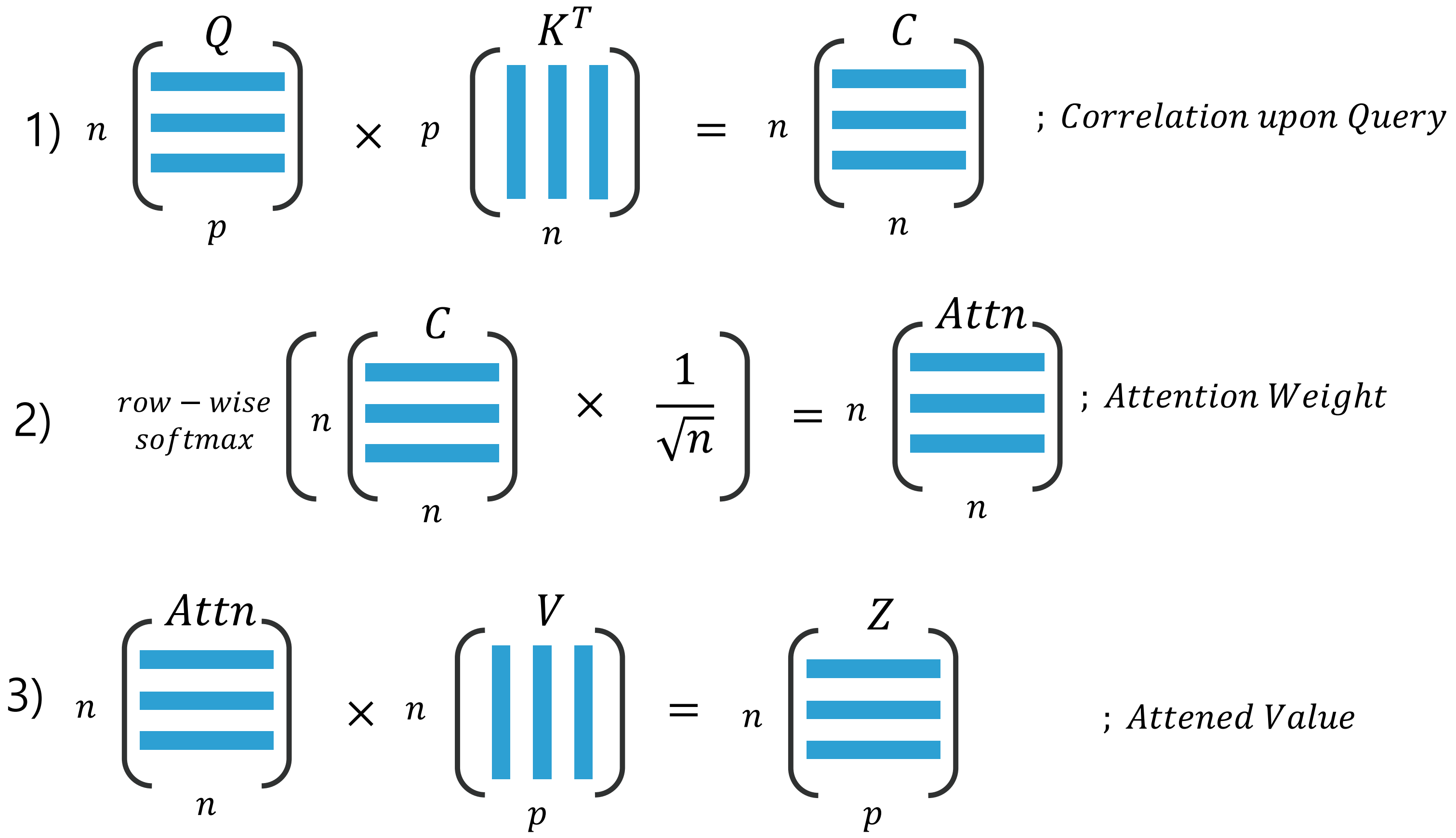
Query, Key, Value Computation

Assume,

- Encoded representation of the input - **(K,V) (n x p)**
- Target Query – **Q (n x p)**
- n = # of words in sequence / p = embedded dimension

$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{n}}\right)\mathbf{V}$$





Self-Attention Module

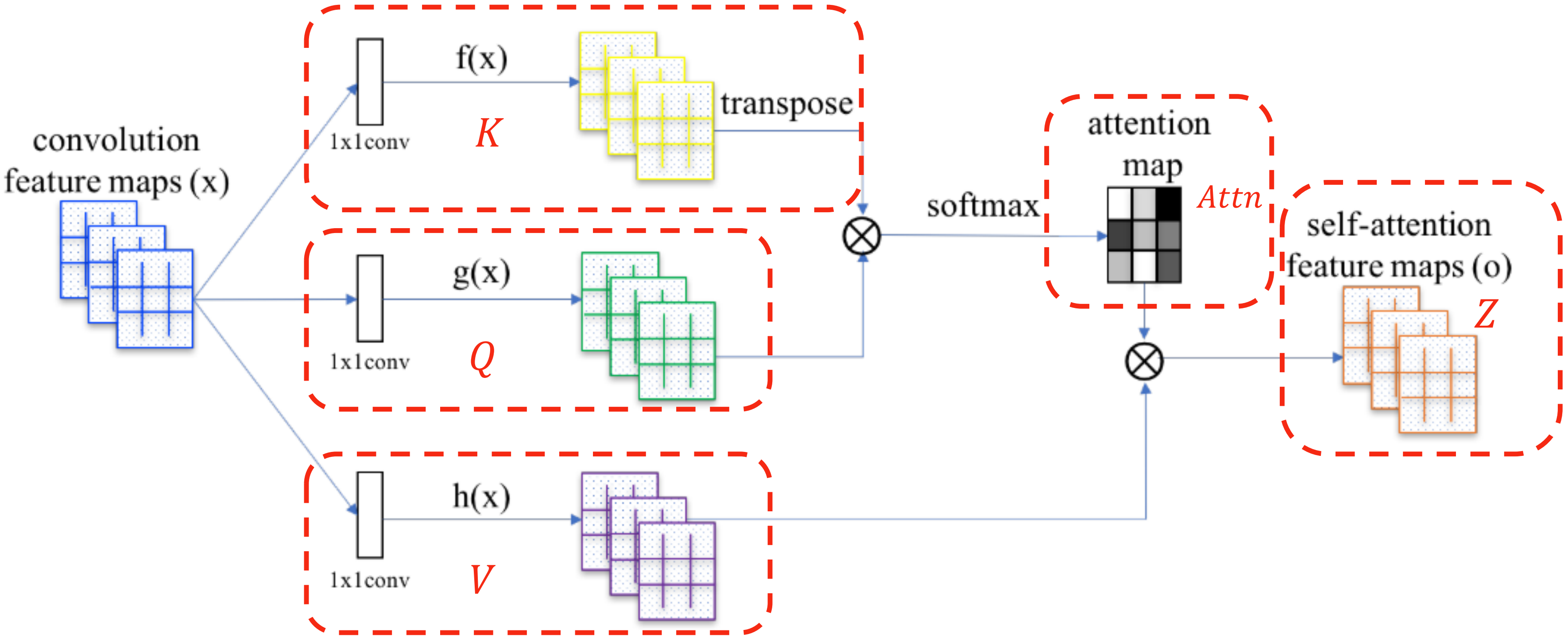
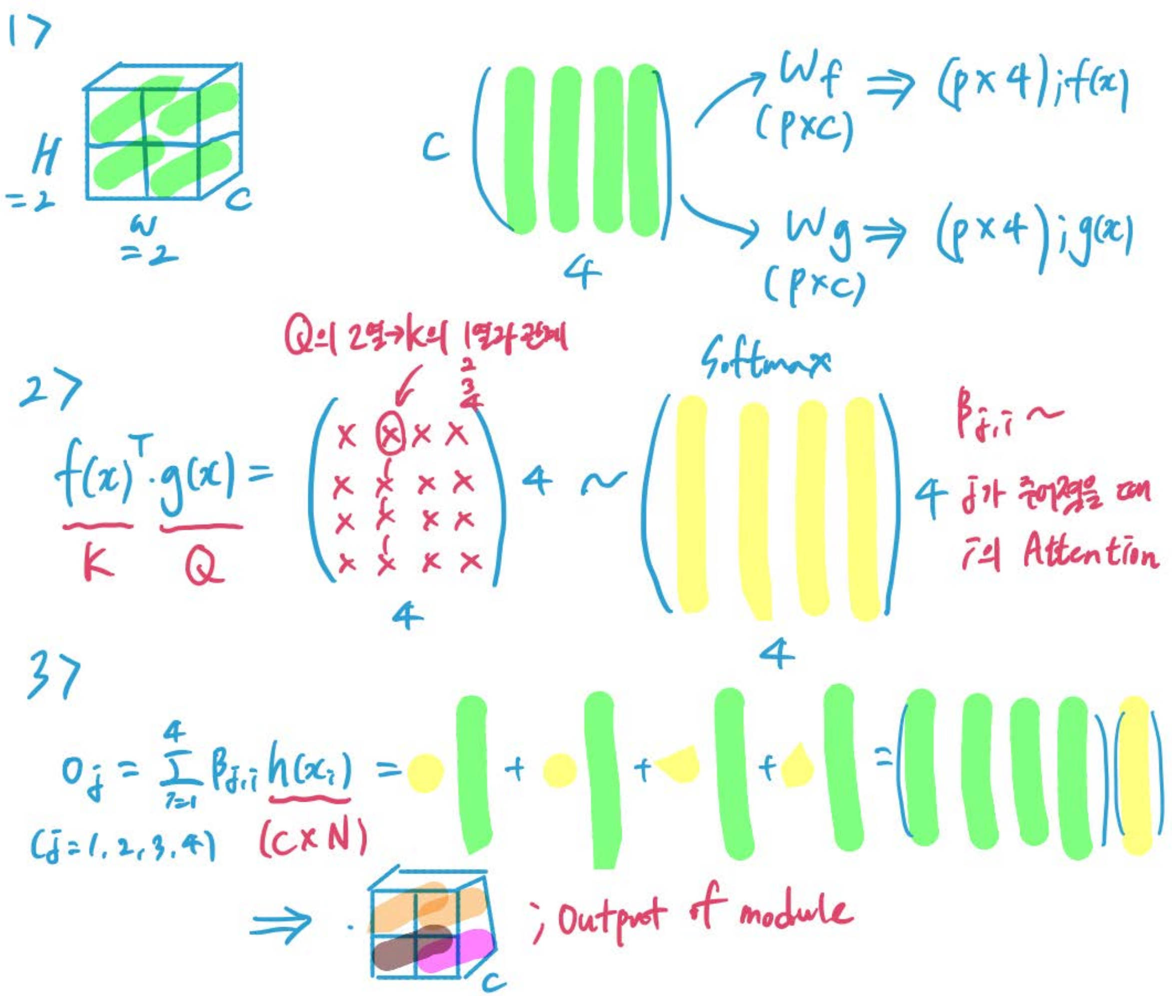


Figure 2: The proposed self-attention mechanism. The ⊗ denotes matrix multiplication. The softmax operation is performed on each row.

Self-Attention Module



(1)

$$\boldsymbol{x} \in \mathbb{R}^{C \times N}$$

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W_f}\boldsymbol{x},\ \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{W_g}\boldsymbol{x}$$

(2)

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

(3)

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

1. 행렬 A의 $spectral\ norm$ = 행렬 A의 가장 큰 $singular\ value$

2. 행렬의 $singular\ value$는 어떤 벡터 a 를 곱했을 때, 이리저리 늘어나는 방향의 늘어나는 정도를 의미함, 따라서 대충 A의 $spectral\ norm$를 제약하는 것은 곱해지는 벡터가 삐죽 삐죽하게 나오는 것이 아니라 smooth하게 나오는 것을 의미함

3. 모델 g에 대해서 $Lipschitz\ norm, ||g||_{Lip}$는 g 의 gradient(∇g)의 $spectral\ norm$ 이다. 그러면 g 의 $Lipschitz\ norm$ 를 제한하는 것은 gradient의 $spectral\ norm$ 를 제약해서 gradient space가 smooth하게 될 수 있도록 해준다.

4. 계산을 해보면 $||g||_{Lip} \leq \prod sn(W^l), W = weight\ matrix$ (참고로 $relu$같은 activation function은 일정 상수로 $Lipschitz\ norm$ 이 계산됨)

5. 따라서 $W_{SN} \leftarrow W/sn(W)$로 normalization 시켜서 모델의 $Lipschitz\ norm$ 를 제약한다.

Stabilization of GAN training

Why Smoothing?

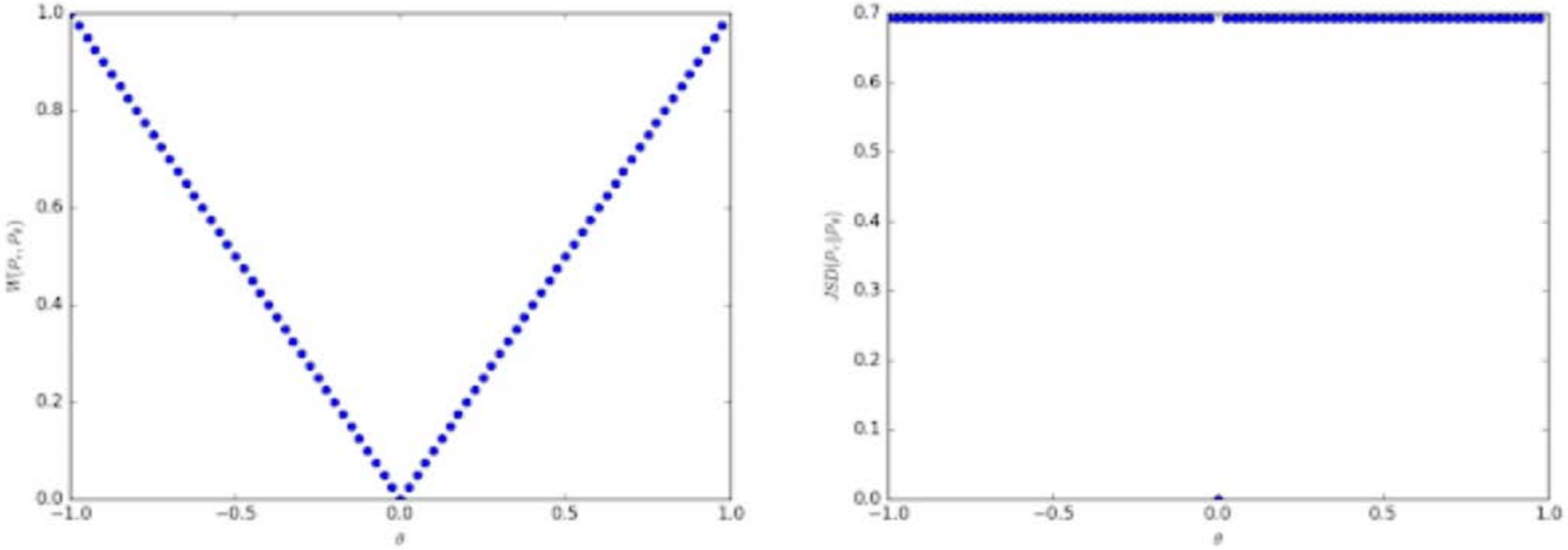


Figure 1: These plots show $\rho(\mathbb{P}_\theta, \mathbb{P}_0)$ as a function of θ when ρ is the EM distance (left plot) or the JS divergence (right plot). The EM plot is continuous and provides a usable gradient everywhere. The JS plot is not continuous and does not provide a usable gradient.

Parameter space after gradient space smoothed

Parameter space before gradient space smoothed

Stabilization Result

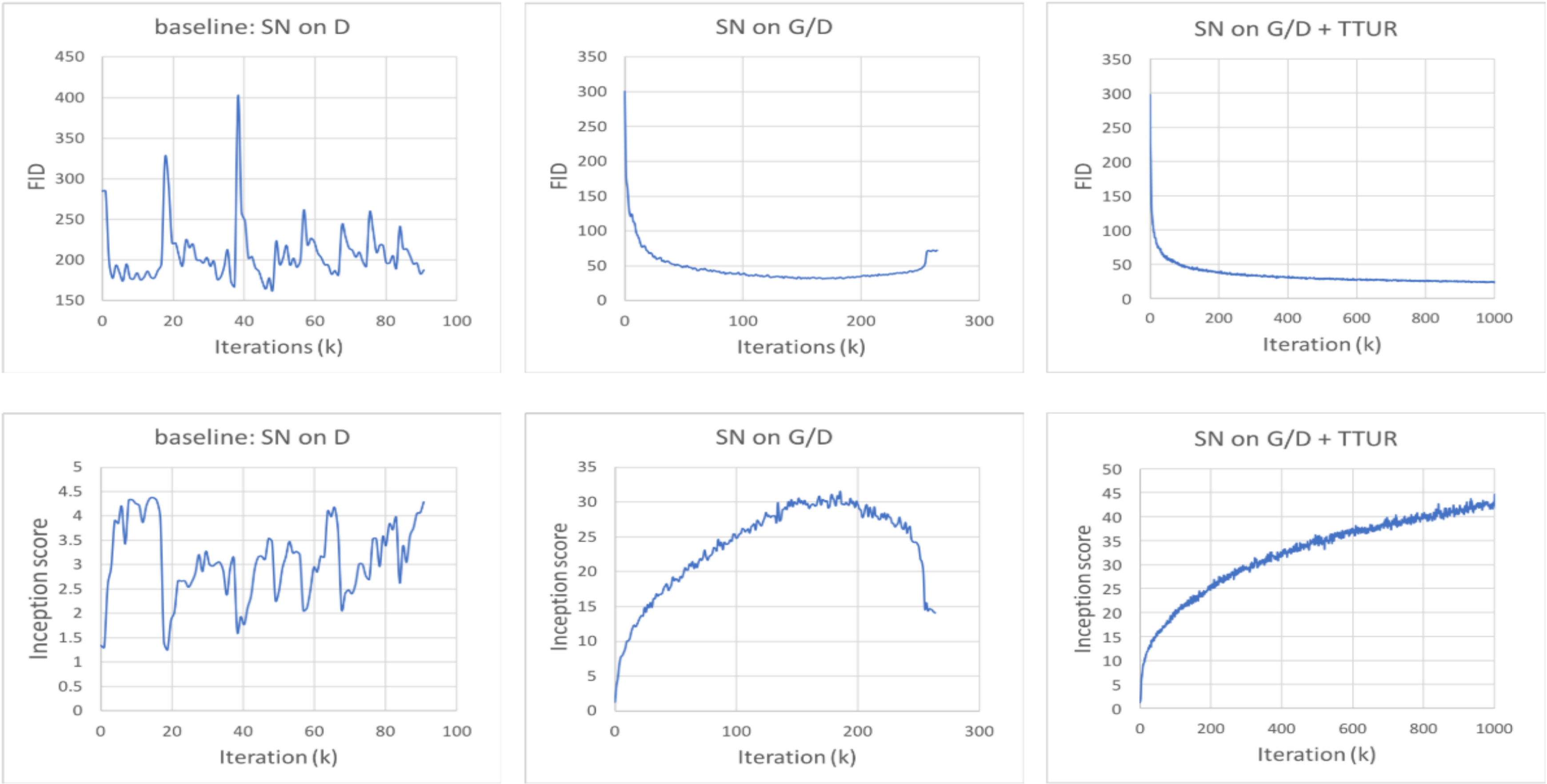


Figure 3: Training curves for the baseline model and our models with the proposed stabilization techniques, "SN on G/D" and two-timescale learning rates (TTUR). All models are trained with 1:1 balanced updates for G and D.

Generation Result (1)

Figure 4: 128×128 examples randomly generated by the baseline model and our models "SN on G/D" and "SN on G/D+TTUR".

Generation Result (2)

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.

Thanks a lot !!
Any Questions?