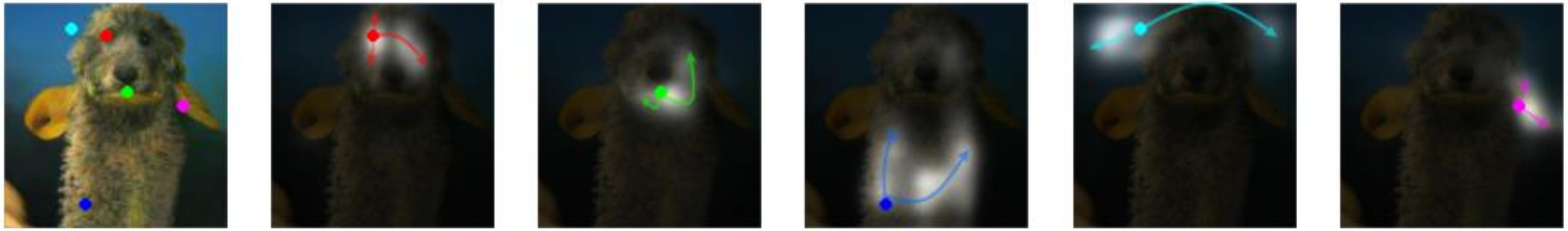
Self-Attention Generative Adversarial Networks

Han Zhang* Rutgers University
Ian Goodfellow Google Brain
Dimitris Metaxas Rutgers University
Augustus Odena Google Brain



Contents

1. Motivation
2. Self-Attention GAN mechanism
3. Tricks to stabilize GAN training
4. Experiments
5. Conclusion

Motivation

1. GANs for natural image generation performs:
   a. Good in texture-featured classes (ocean, sky, landscape)
   b. Bad in structural-patterned classes (animals)
2. Convolution’s receptive field (kernel size) is often small
   a. Long range dependencies can only be possessed very late
   b. Large kernels are computationally expensive
3. **Target: To make GANs pay more attention to long range structural information (the relationship between far away image regions)**

Self-Attention Mechanism

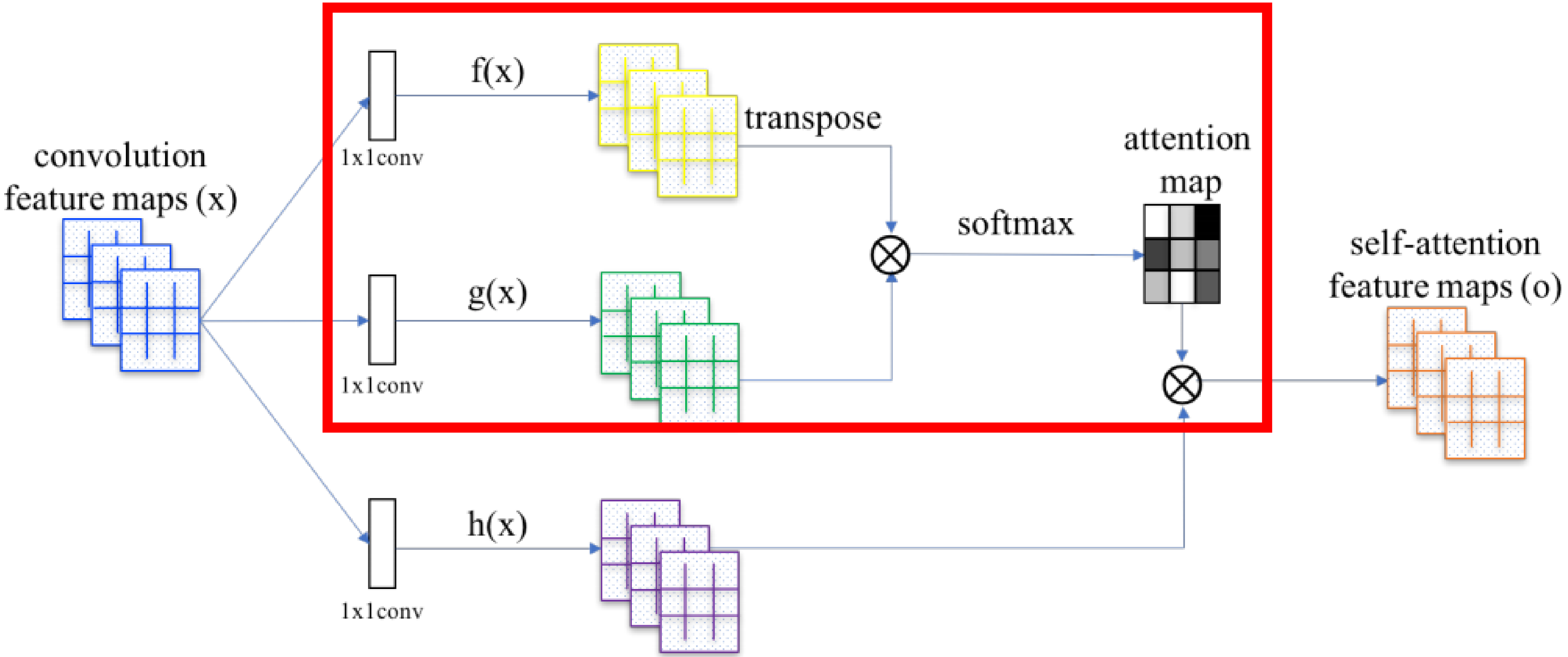
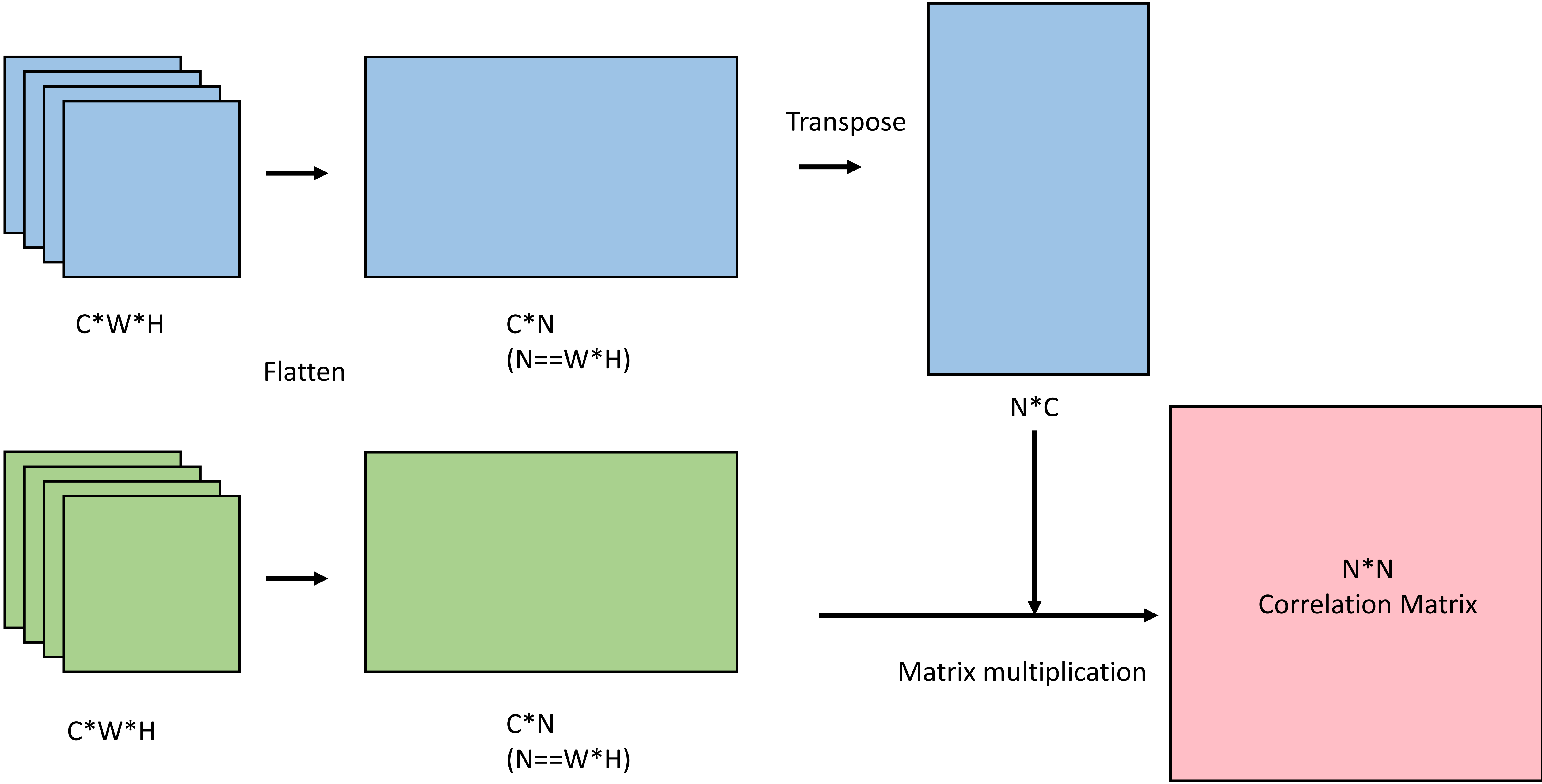


Figure 2: The proposed self-attention mechanism. The ⊗ denotes matrix multiplication. The softmax operation is performed on each row.


C*W*H
Flatten
C*N
(N==W*H)
Transpose
N*C
C*W*H
C*N
(N==W*H)
N*N
Correlation Matrix
Matrix multiplication

Self-Attention Mechanism

1. To find a way efficiently utilize global information

The image features from the previous hidden layer $\boldsymbol{x} \in \mathbb{R}^{C \times N}$ are first transformed into two feature spaces $\boldsymbol{f}, \boldsymbol{g}$ to calculate the attention, where $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W_f x}$, $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{W_g x}$

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

2. W are learnable weight matrices, implemented by 1*1 convolution

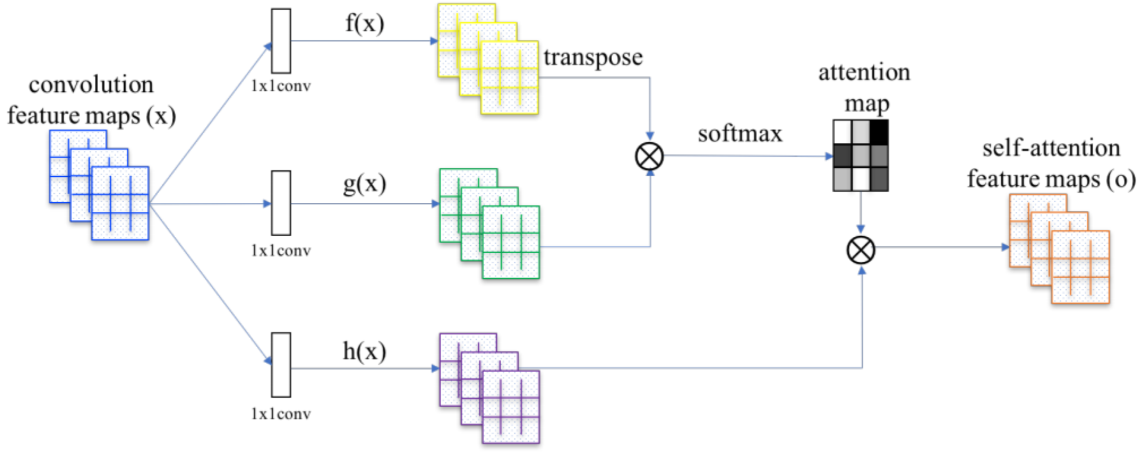


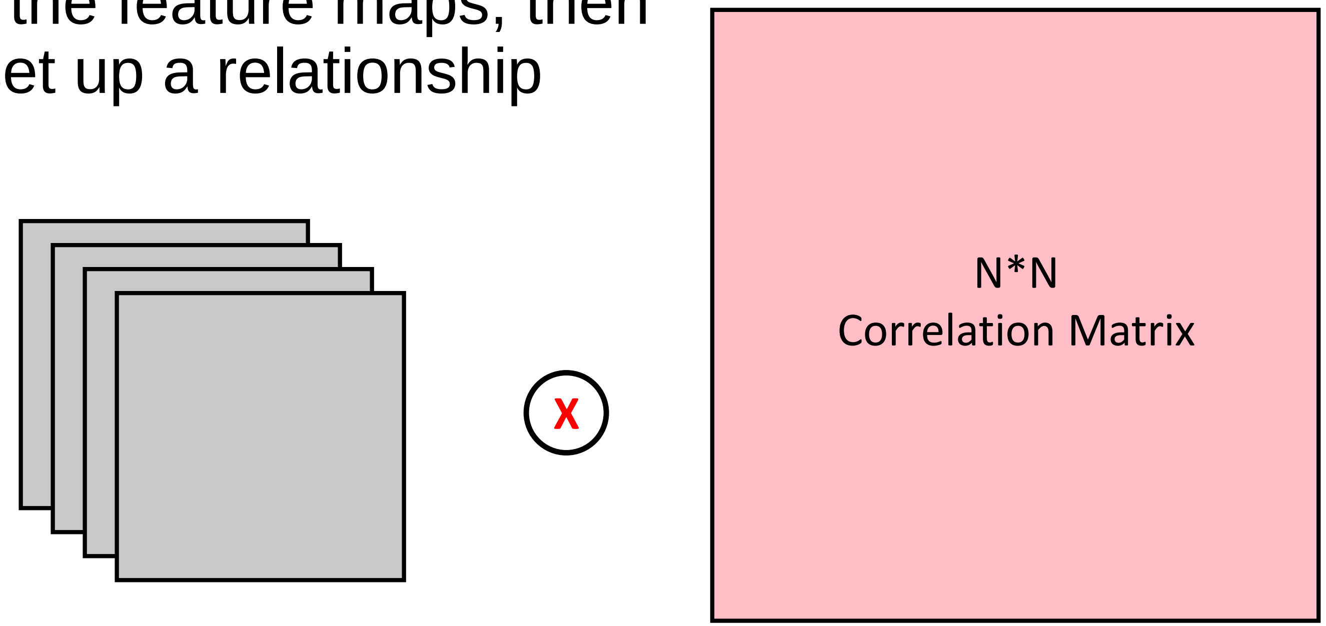
Figure 2: The proposed self-attention mechanism. The $\otimes$ denotes matrix multiplication. The softmax operation is performed on each row.

Self-Attention Mechanism

1. N rows times N columns, SoftMax for **each row**
2. Each row is actually N=W*H, represents one attention
3. Apply the whole Attention Map to the feature maps, then every pair of pixel locations can set up a relationship

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

$$\boldsymbol{y_i} = \gamma \boldsymbol{o_i} + \boldsymbol{x_i},$$



h(xi), C*W*H

Attention Map!

```python
def attention(self, x, ch, sn=False, scope='attention', reuse=False):
    with tf.variable_scope(scope, reuse=reuse):
        f = conv(x, ch // 8, kernel=1, stride=1, sn=sn, scope='f_conv') # [bs, h, w, c']
        g = conv(x, ch // 8, kernel=1, stride=1, sn=sn, scope='g_conv') # [bs, h, w, c']
        h = conv(x, ch, kernel=1, stride=1, sn=sn, scope='h_conv') # [bs, h, w, c]

        # N = h * w
        s = tf.matmul(hw_flatten(g), hw_flatten(f), transpose_b=True) # # [bs, N, N]

        beta = tf.nn.softmax(s, axis=-1)  # attention map

        o = tf.matmul(beta, hw_flatten(h)) # [bs, N, C]
        gamma = tf.get_variable("gamma", [1], initializer=tf.constant_initializer(0.0))

        o = tf.reshape(o, shape=x.shape) # [bs, h, w, C]
        x = gamma * o + x

    return x
```

Self-Attention Mechanism

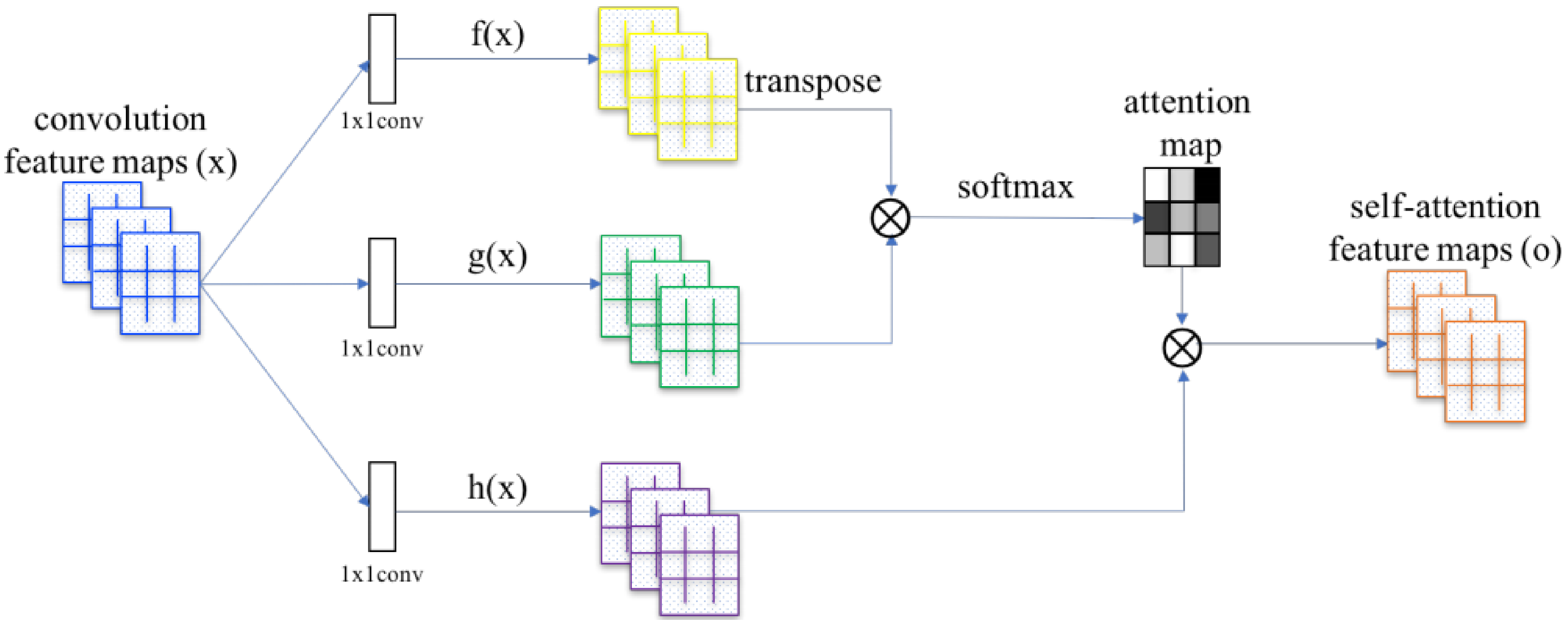


Figure 2: The proposed self-attention mechanism. The ⊗ denotes matrix multiplication. The softmax operation is performed on each row.

Other tricks for stabilizing

1. Spectral normalization for both generator and discriminator
2. Imbalanced learning rate for generator and discriminator

SPECTRAL NORMALIZATION FOR GENERATIVE ADVERSARIAL NETWORKS

Takeru Miyato[1], Toshiki Kataoka[1], Masanori Koyama[2], Yuichi Yoshida[3]
{miyato, kataoka}@preferred.jp
koyama.masanori@gmail.com
yyoshida@nii.ac.jp
[1]Preferred Networks, Inc. [2]Ritsumeikan University [3]National Institute of Informatics

GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium

Martin Heusel **Hubert Ramsauer** **Thomas Unterthiner** **Bernhard Nessler**

Evaluation Metric

1. **Inception Score (IS):** computes the KL divergence between the conditional class distribution and the marginal class distribution
2. **Fréchet Inception distance (FID) :** Wasserstein-2 distance between the generated images and the real images in the feature space of an Inception-v3 network

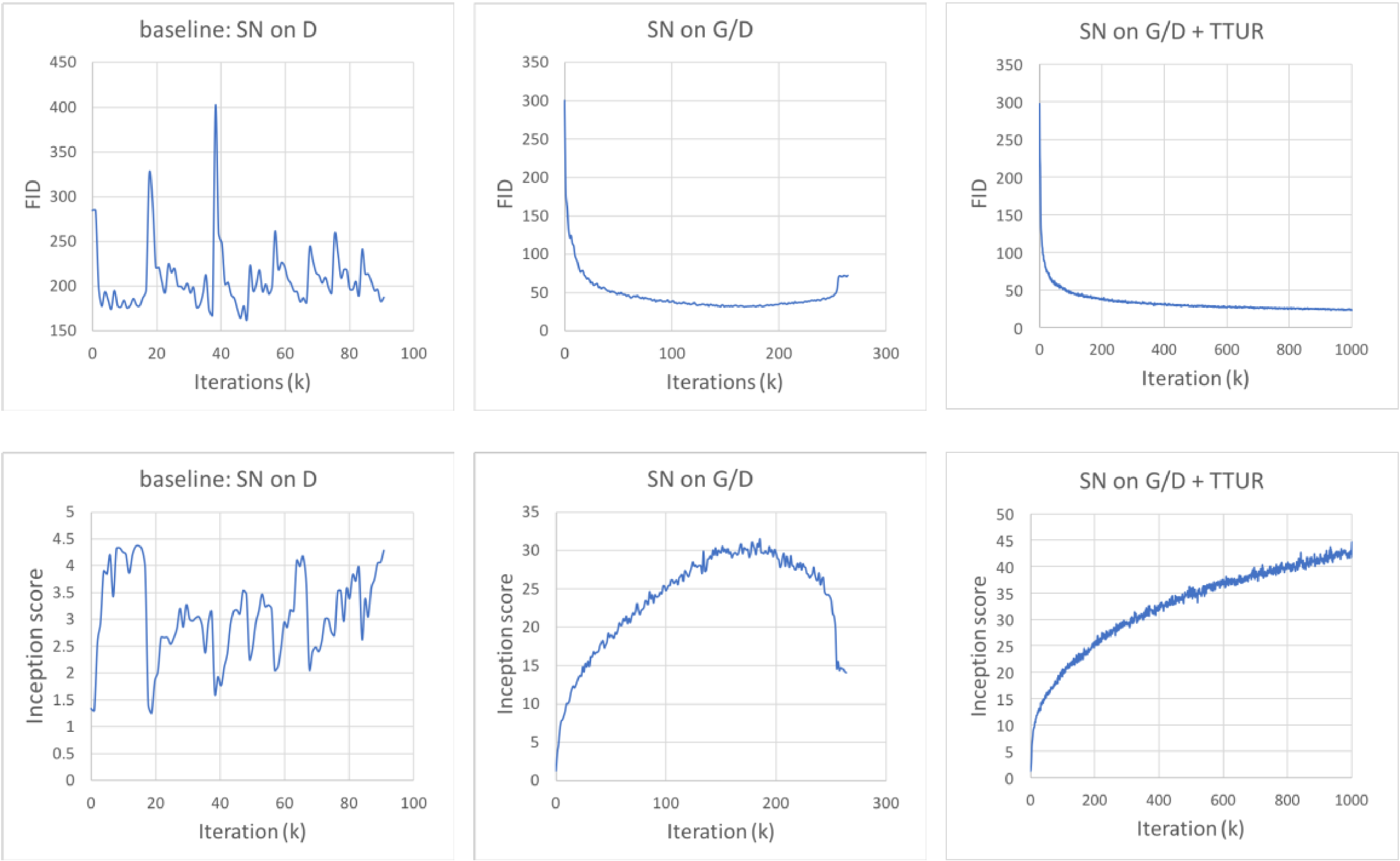
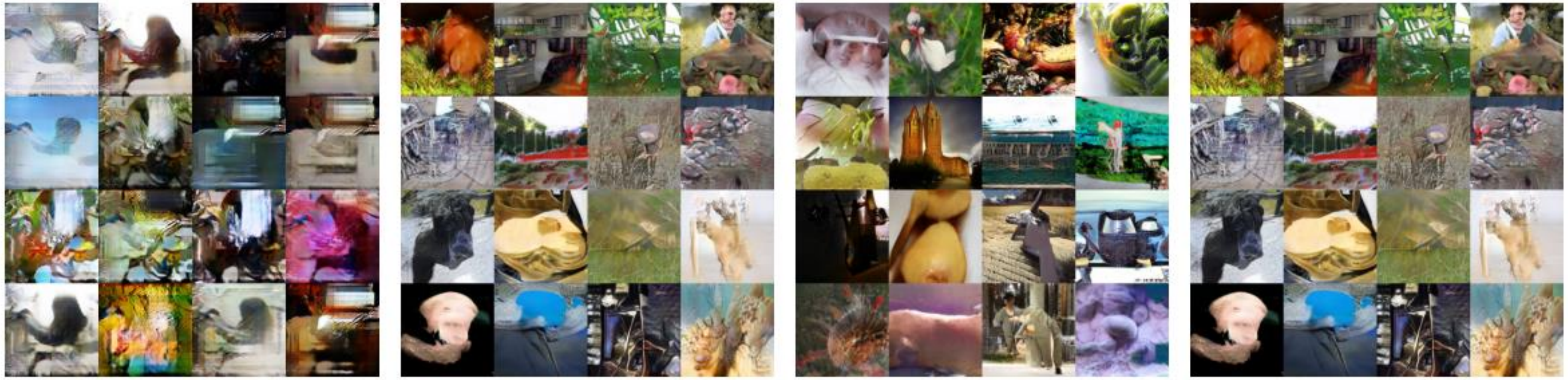


Figure 3: Training curves for the baseline model and our models with the proposed stabilization techniques, "SN on G/D" and two-timescale learning rates (TTUR). All models are trained with 1:1 balanced updates for G and D.

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

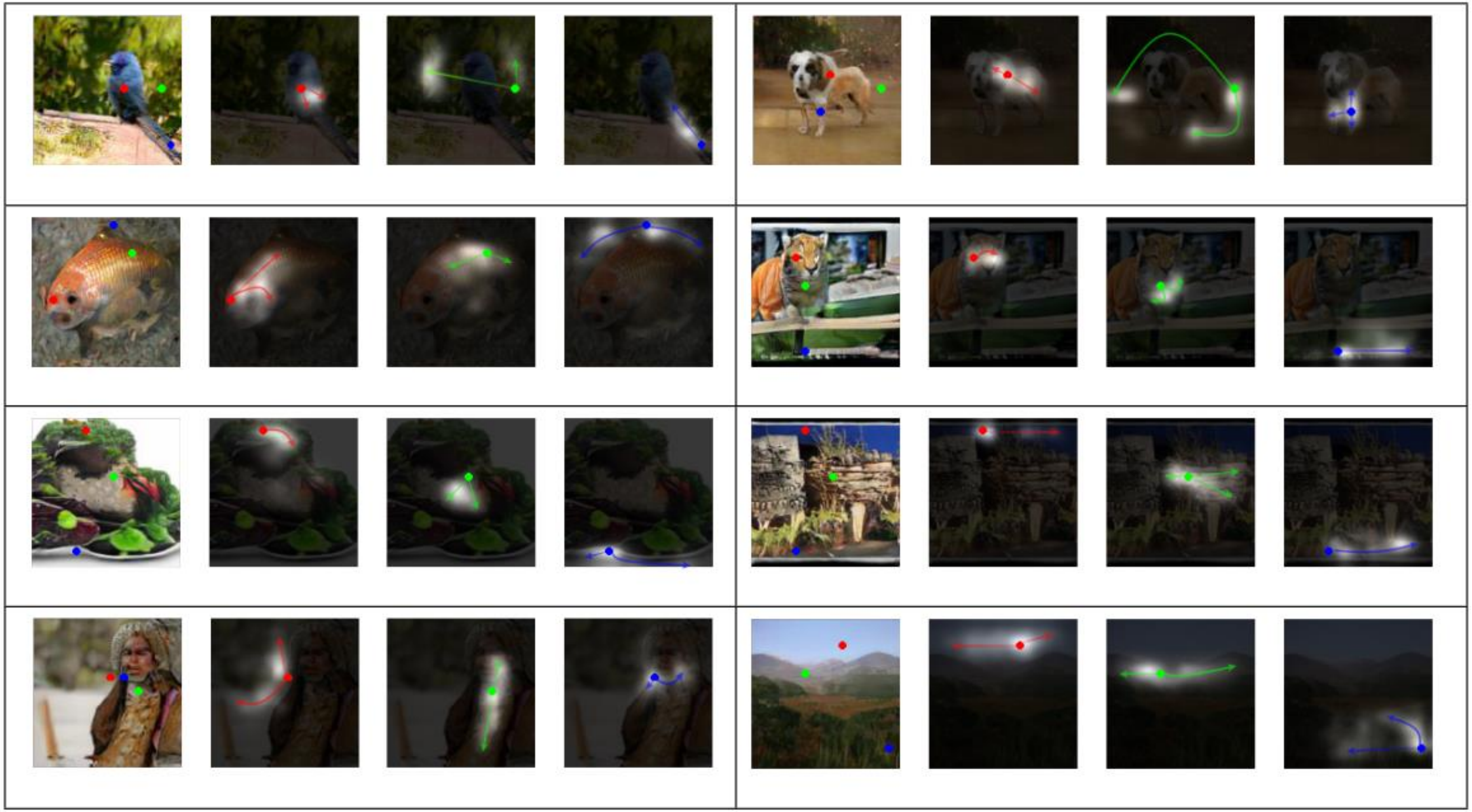
Table 1: Comparison of Self-Attention and Residual block on GANs. These blocks are added into different layers of the network. All models have been trained for one million iterations, and the best Inception scores (IS) and Fréchet Inception distance (FID) are reported.



Baseline: SN on D (10k, FID=181.84) | SN on G/D (10k, FID=93.52) | SN on G/D (160k, FID=33.39) | SN on G/D (260k, FID=72.41)

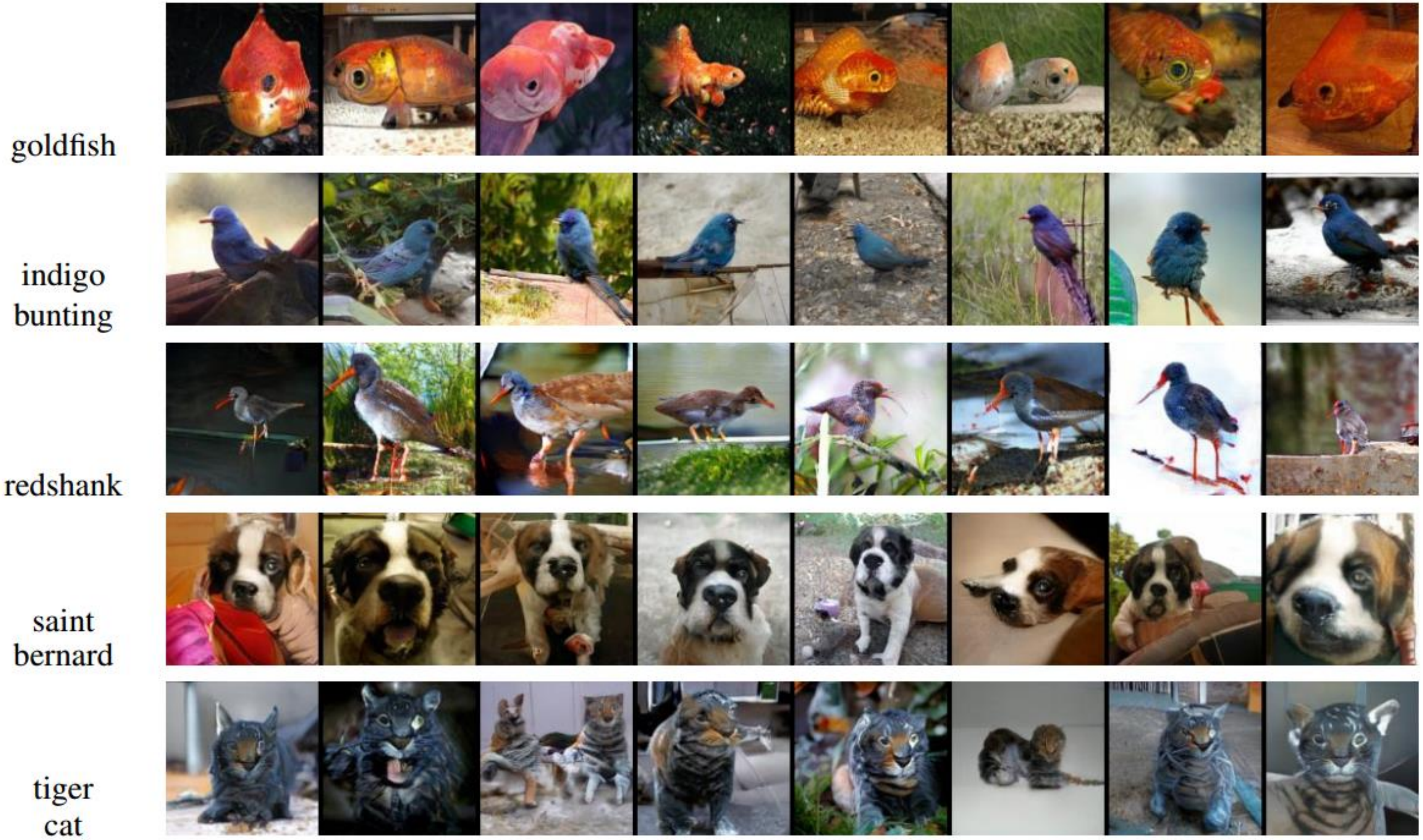
SN on G/D+TTUR (10k, FID=99.04) | SN on G/D+TTUR (160k, FID=40.96) | SN on G/D+TTUR (260k, FID=34.62) | SN on G/D+TTUR (1M, FID=22.96)

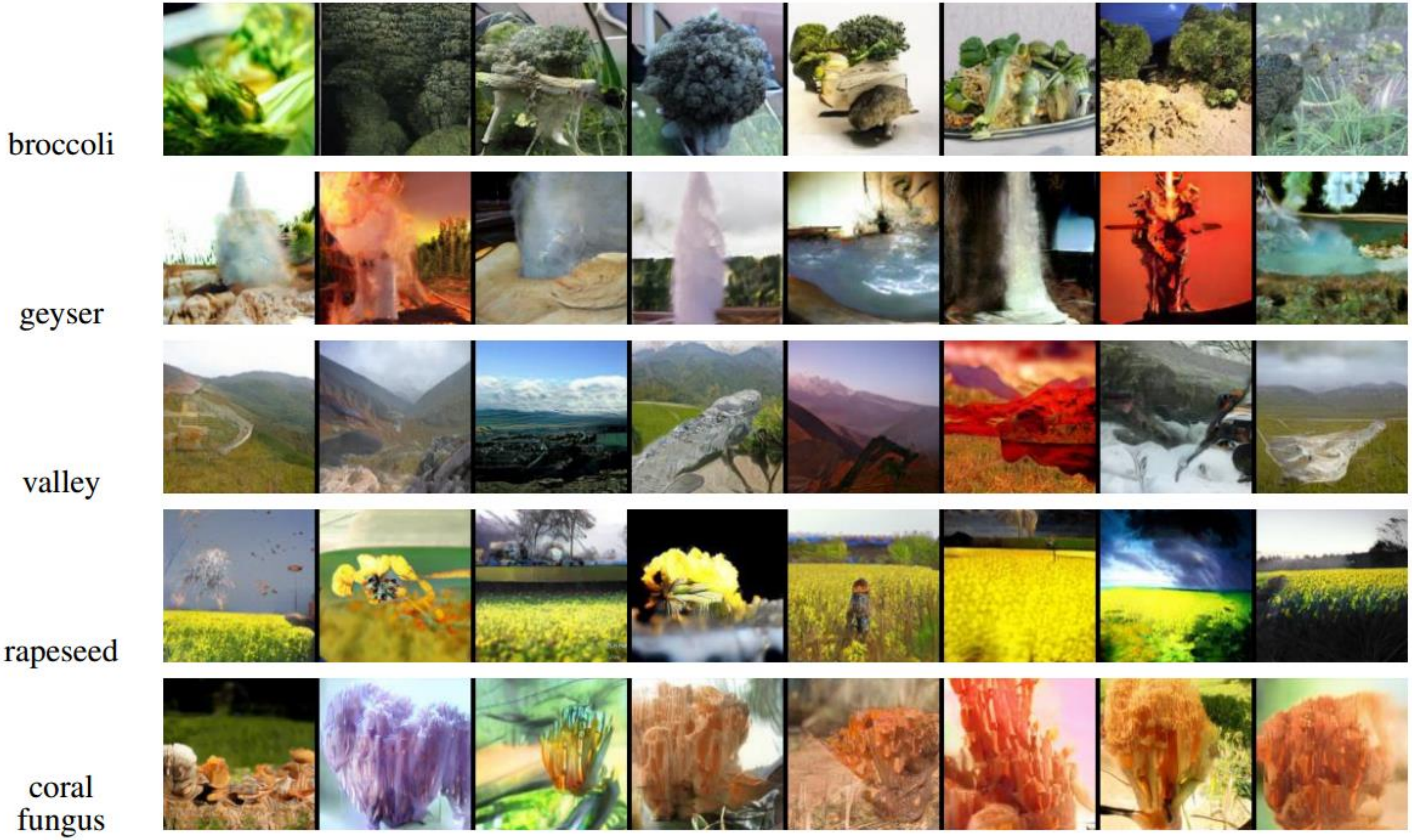
Figure 4: 128×128 examples randomly generated by the baseline model and our models "SN on G/D" and "SN on G/D+TTUR".



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.


goldfish
indigo bunting
redshank
saint bernard
tiger cat


broccoli
geyser
valley
rapeseed
coral
fungus

Conclusion

1. The self-attention module is effective in modeling long-range dependencies.
2. Spectral normalization applied to the generator stabilizes GAN training and that TTUR speeds up training of regularized discriminators.