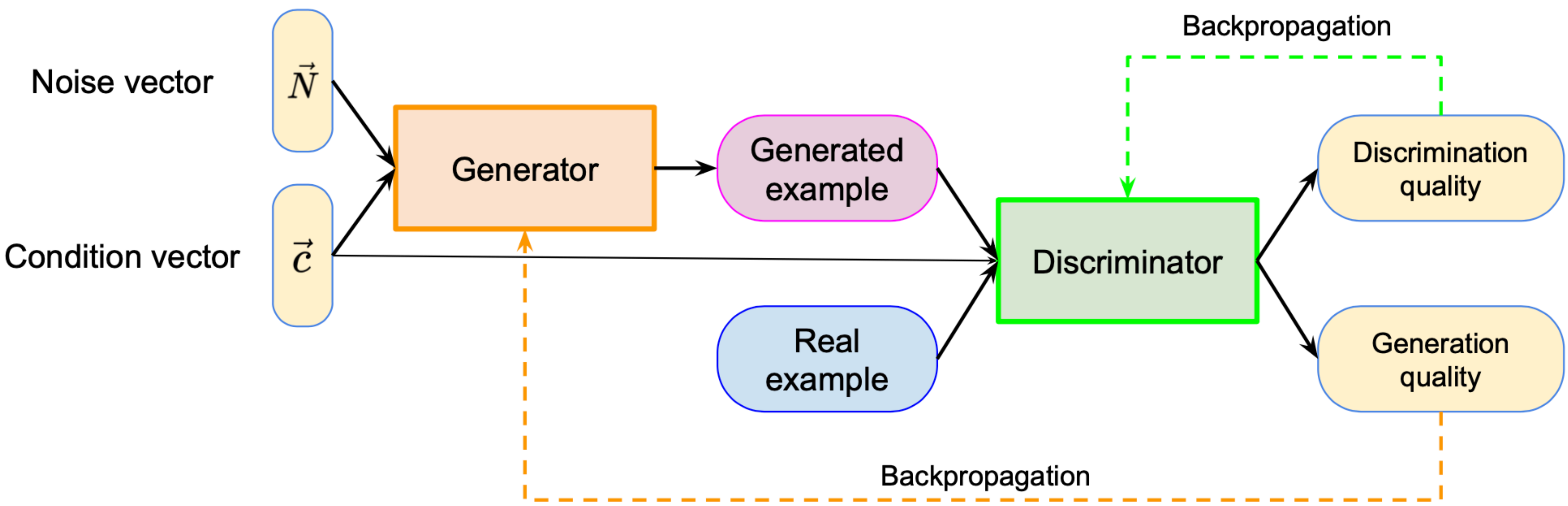
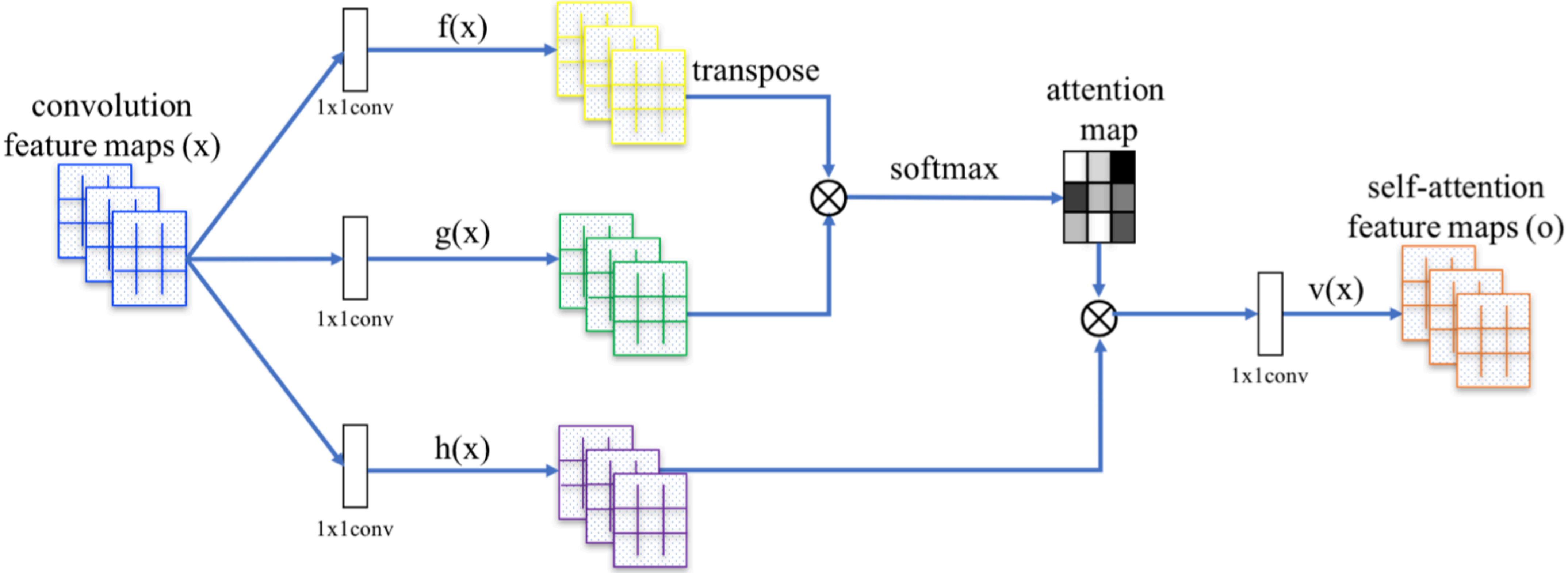
Large Scale GAN Training for High Fidelity Natural Image Synthesis

Джаин Никита БПМИ172

Conditional GAN



Self-Attention GAN



$$output = y * o + x$$

Large Scale GAN

Batch	Ch.	Param (M)	Shared	Skip-z	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)

- Увеличенный размер батча
- Число параметров в каждом слое увеличено на 50%
- Shared conditional BatchNorm
- Truncation trick
- Orthogonal regularization

Truncation trick

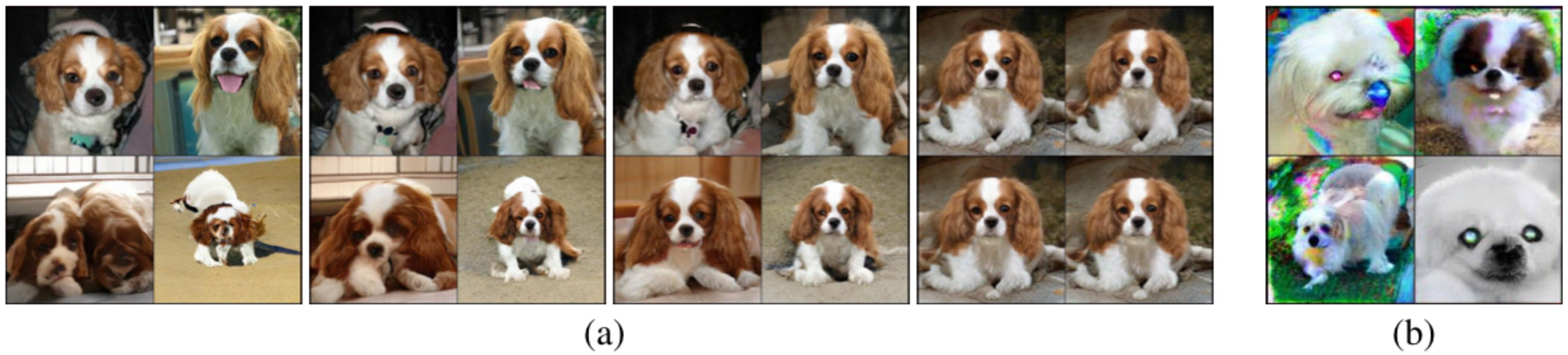
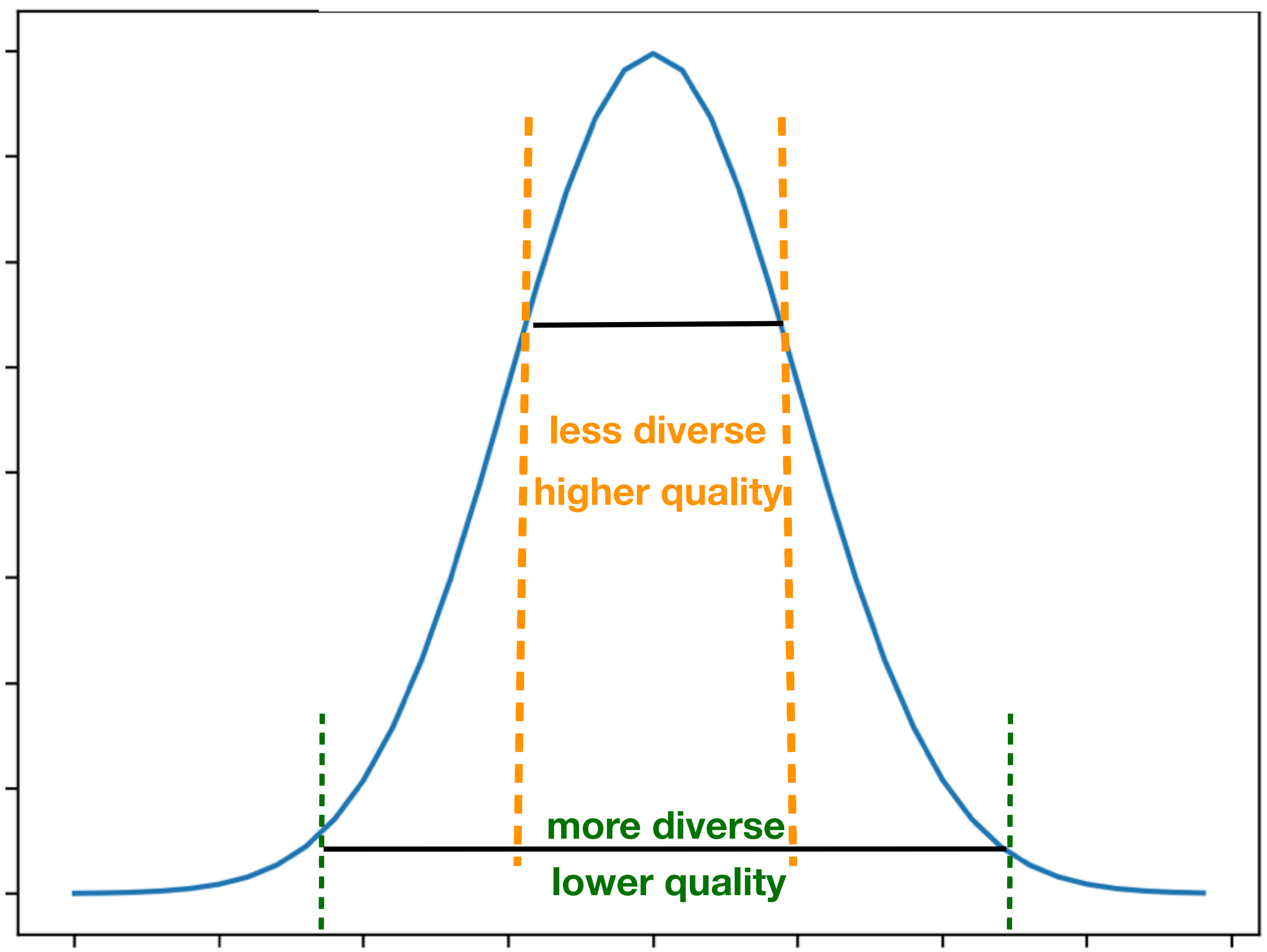


Figure 2: (a) The effects of increasing truncation. From left to right, threshold=2, 1.5, 1, 0.5, 0.04. (b) Saturation artifacts from applying truncation to a poorly conditioned model.



Truncation trick

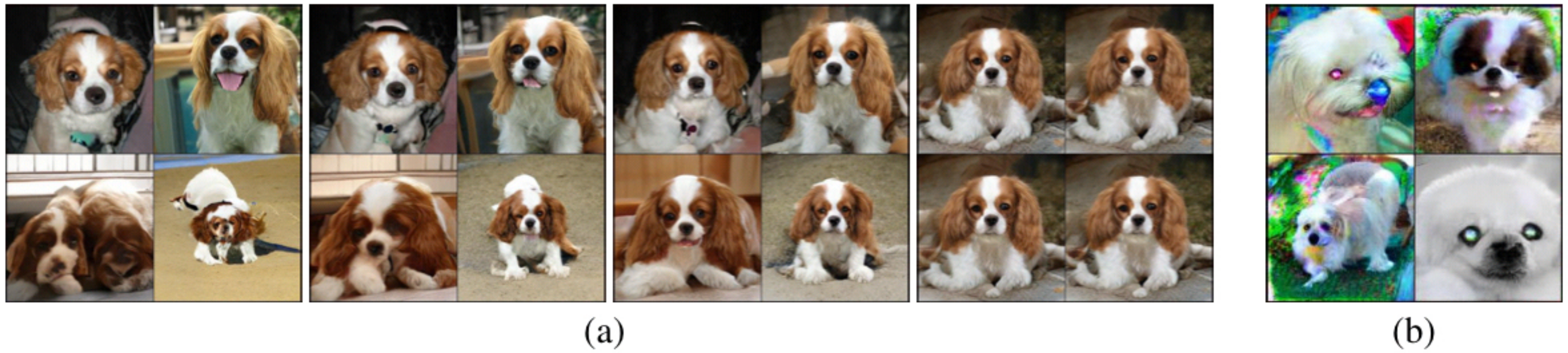
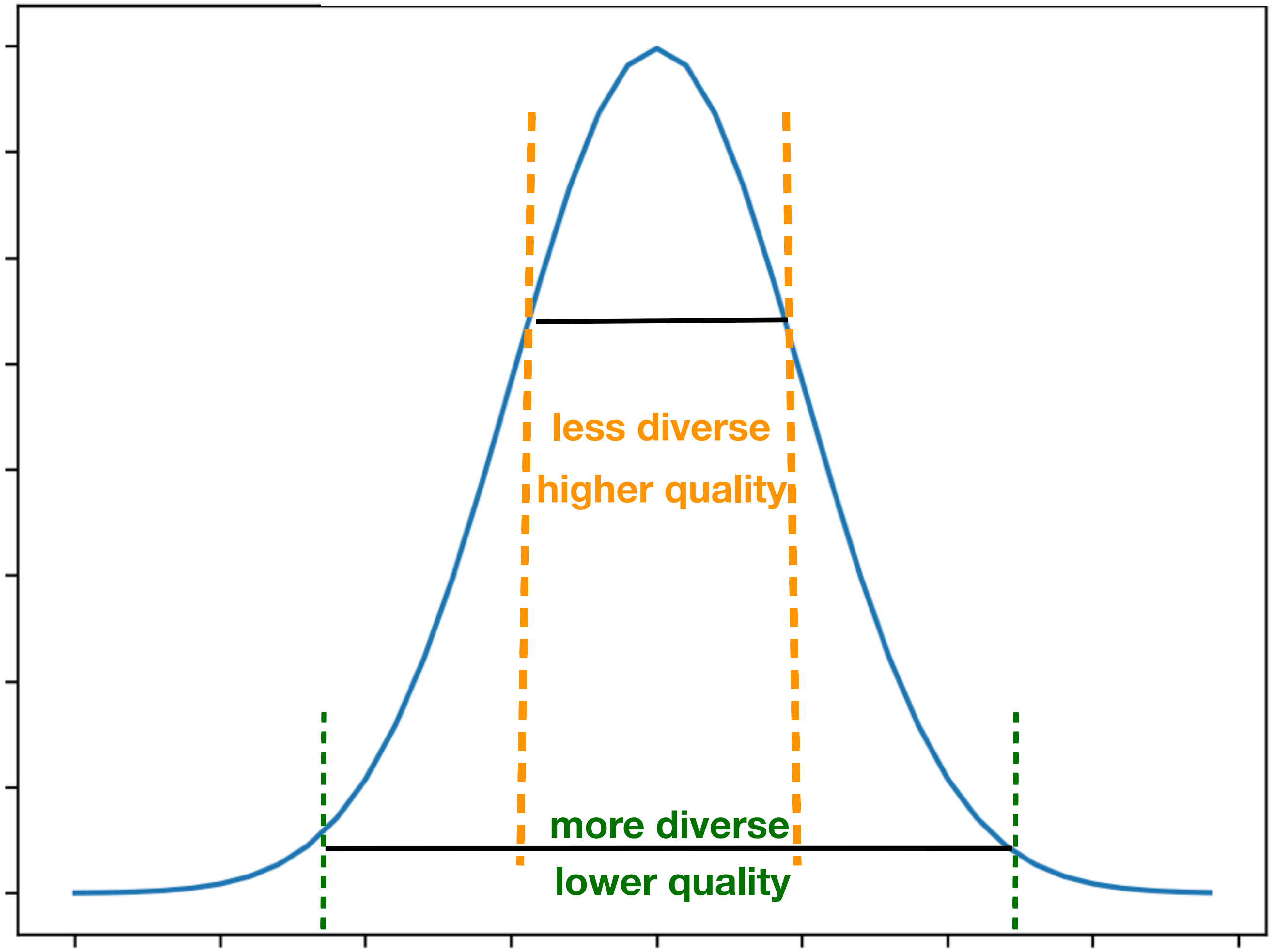


Figure 2: (a) The effects of increasing truncation. From left to right, threshold=2, 1.5, 1, 0.5, 0.04. (b) Saturation artifacts from applying truncation to a poorly conditioned model.



Однако некоторые модели довольно плохо работают с truncated normal.

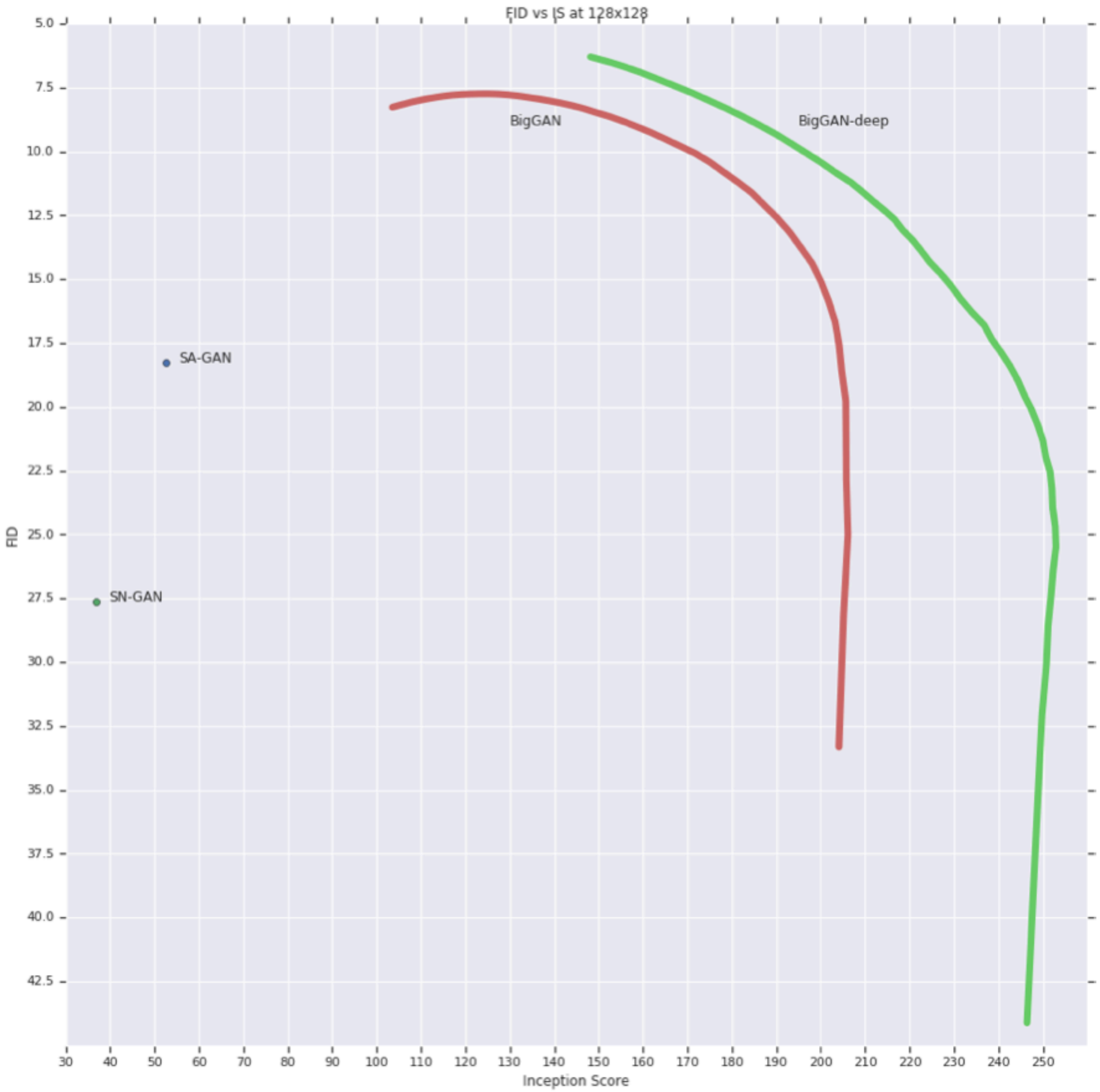
В таких случаях можно использовать ортогональную регуляризацию:

$$R_\beta = \beta \| W^T W - I \|_F^2$$

Или менее строгий вариант:

$$R_\beta = \beta \| W^T W \odot (1 - I) \|_F^2$$

Truncation trick

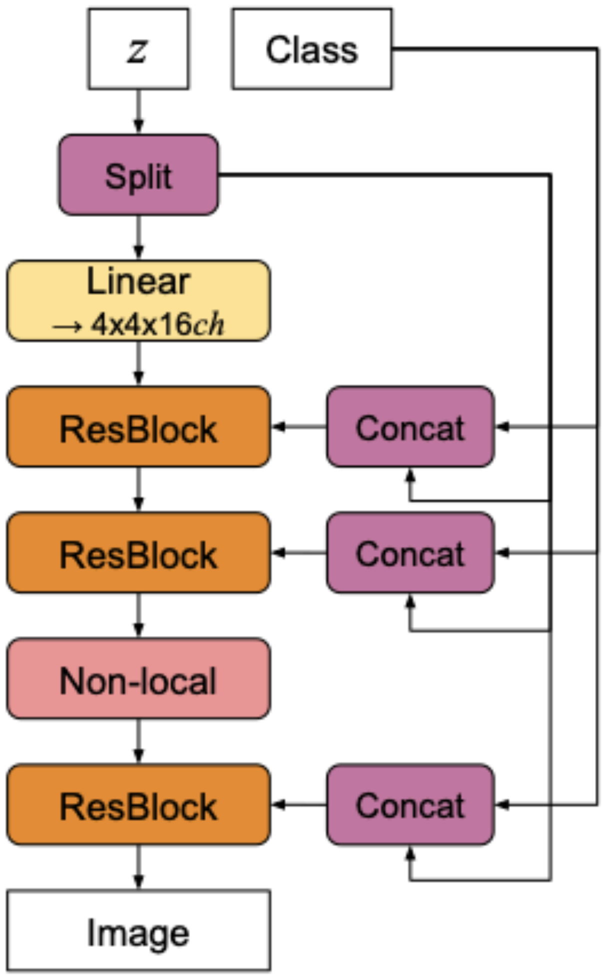


Shared Conditional Batchnorm

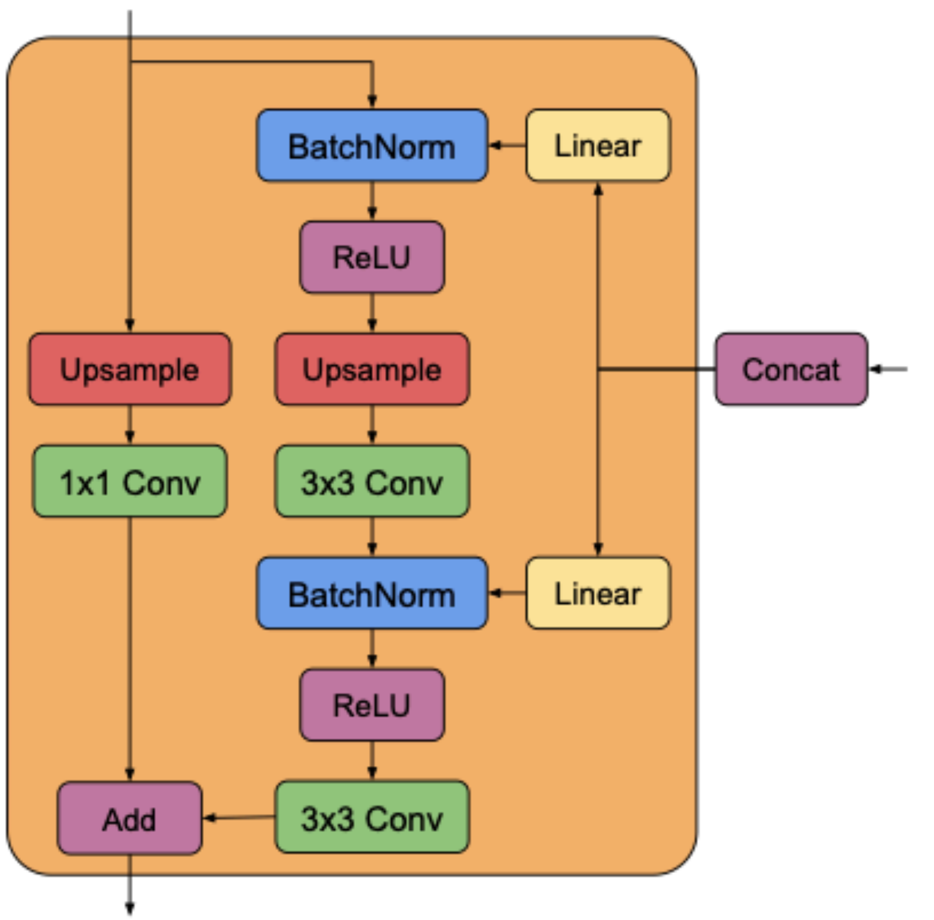
$$output = \gamma * \hat{x} + \beta$$

$$\gamma_i = Linear(concat(y, z_i))$$

$$\beta_i = Linear(concat(y, z_i))$$



(a)



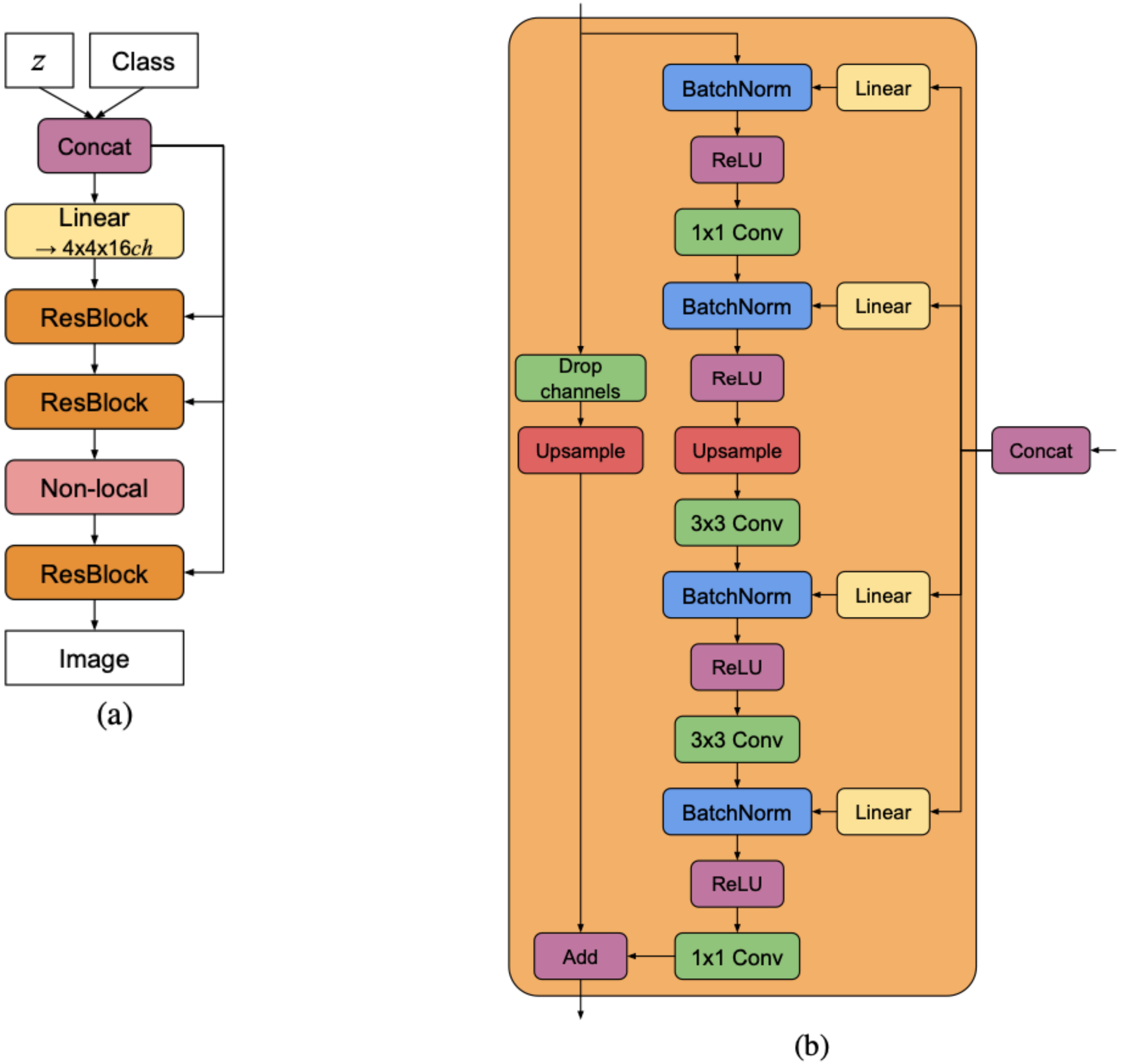
(b)

Shared Conditional Batchnorm

$$output = \gamma * \hat{x} + \beta$$

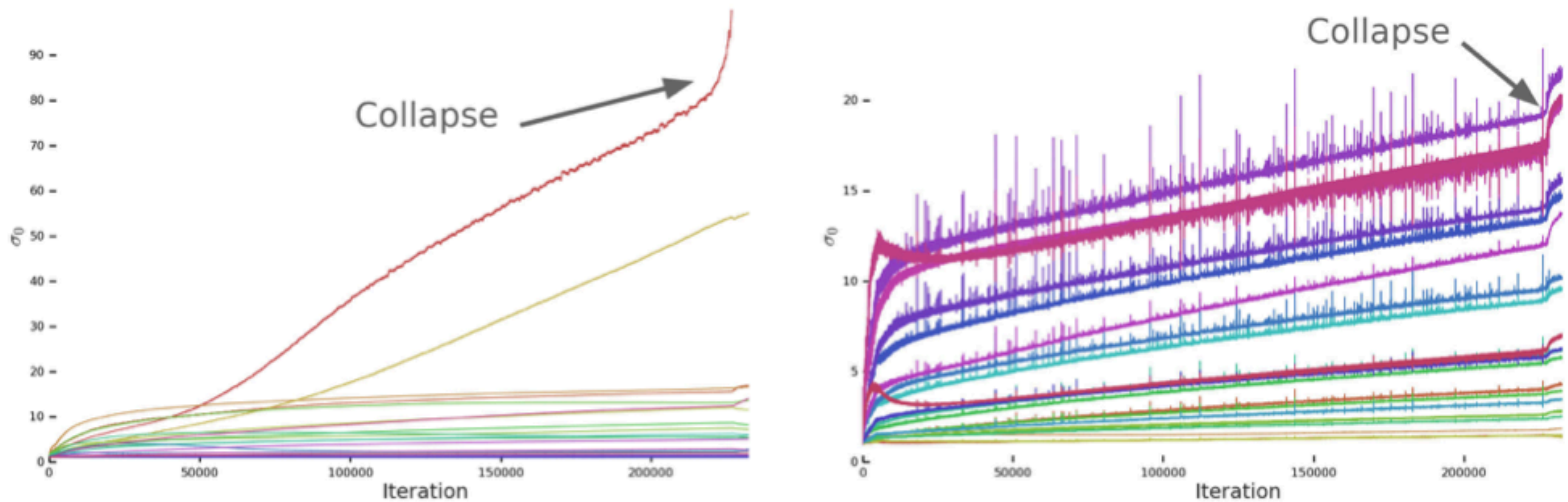
$$\gamma_i = Linear(concat(y, z_i))$$

$$\beta_i = Linear(concat(y, z_i))$$



- Требует меньше памяти
- Увеличивает скорость обучения на 37%
- Уменьшает вычислительную стоимость

Instability: Generator



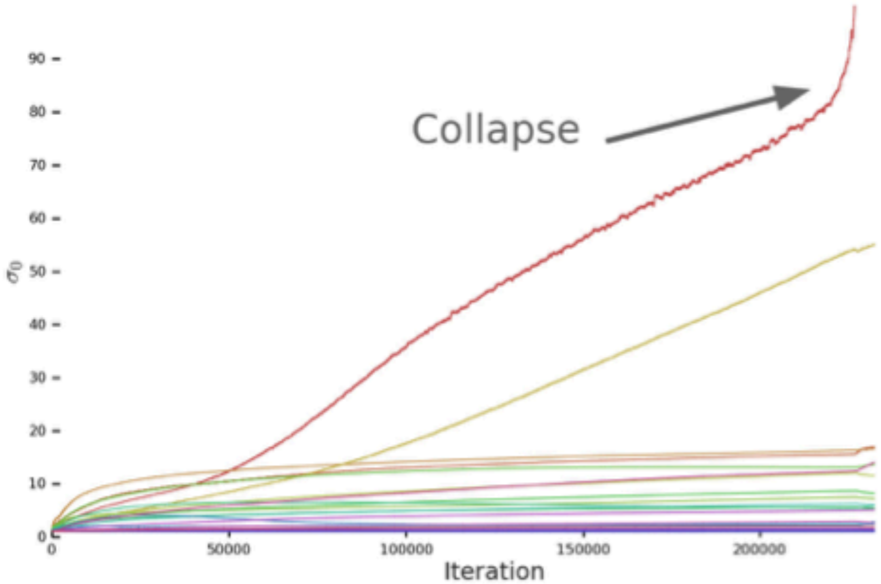
(a) **G** (b) **D**

Можно попробовать предотвратить взрыв с помощью регуляризации:

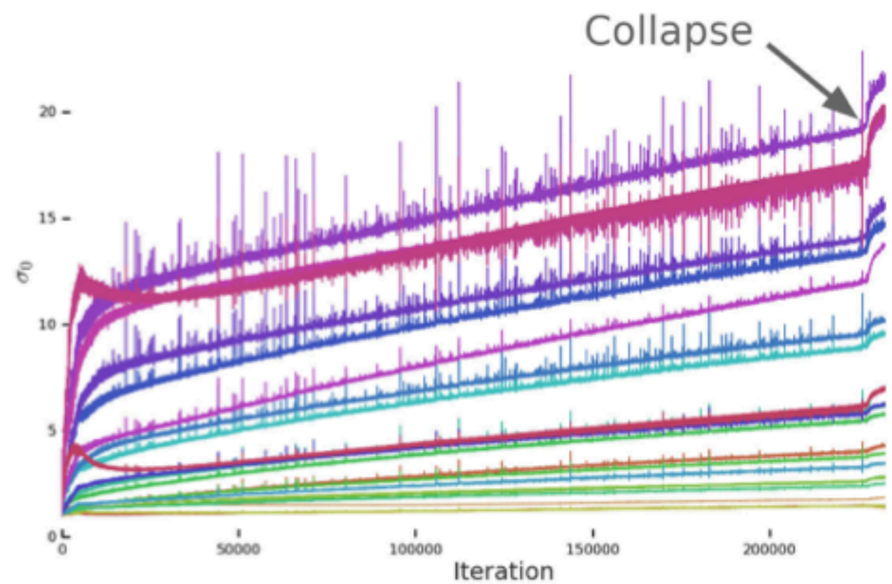
$$W = W - max(0, \sigma_0 - \sigma_{clamp}) v_0 u_0^T$$

Такая регуляризация в некоторых случаях может немного улучшить стабильность, однако полностью решить проблему не помогает.

Instability: Discriminator



(a) **G**



(b) **D**

В случае дискриминатора также можно попробовать избавиться от зашумленности и скачков, введя дополнительно градиентный штраф:

$$R_1 = \frac{\gamma}{2}\mathbb{E}_{pD(x)}[\|\nabla D(x)\|_F^2]$$

Такая регуляризация действительно помогает достигнуть стабильности, но ценой потери качества генерации.

Generated examples

Figure 5: Samples generated by our BigGAN model at 256×256 resolution.

Figure 6: Samples generated by our BigGAN model at 512×512 resolution.