

StyleGAN2

Караваев Георгий, ПМИ 212

Введение

2

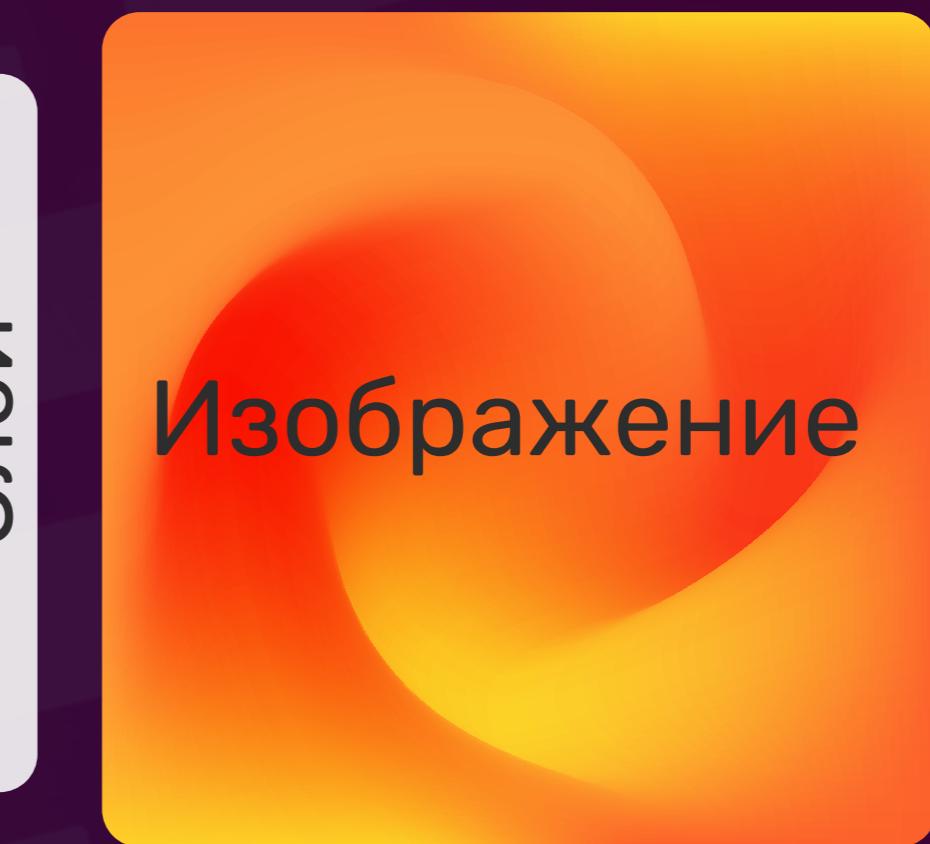
Используем глубинное обучение для генерации изображений.

Случайный вектор

Слой

Слой

Слой



Изображение

Введение

3

Для обучения нейросети нам нужна функция потерь.

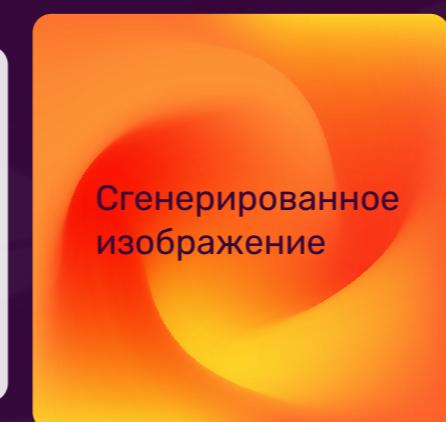
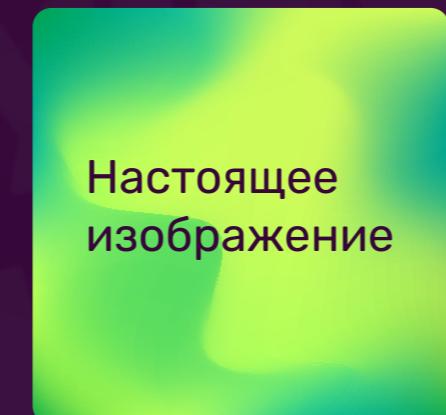
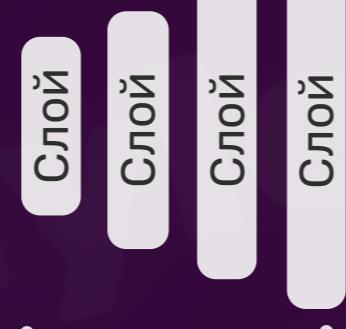
В генеративно-состязательных сетях в роли функции потерь, определяющей качество полученных изображений, используется вторая нейросеть. Её цель – отличить сгенерированные изображения от настоящих.

Введение

4

Между генератором и дискриминатором ведётся игра с нулевой суммой. Именно поэтому сеть называется соревновательной.

Случайный вектор



StyleGAN

5

У генеративно-состязательных сетей, сделанных до StyleGAN, была одна общая **проблема**: они действовали по принципу чёрного ящика. Основной целью StyleGAN было создание более интерпретируемой архитектуры, предоставляющей возможность предсказуемым способом влиять на результат генерации.

StyleGAN

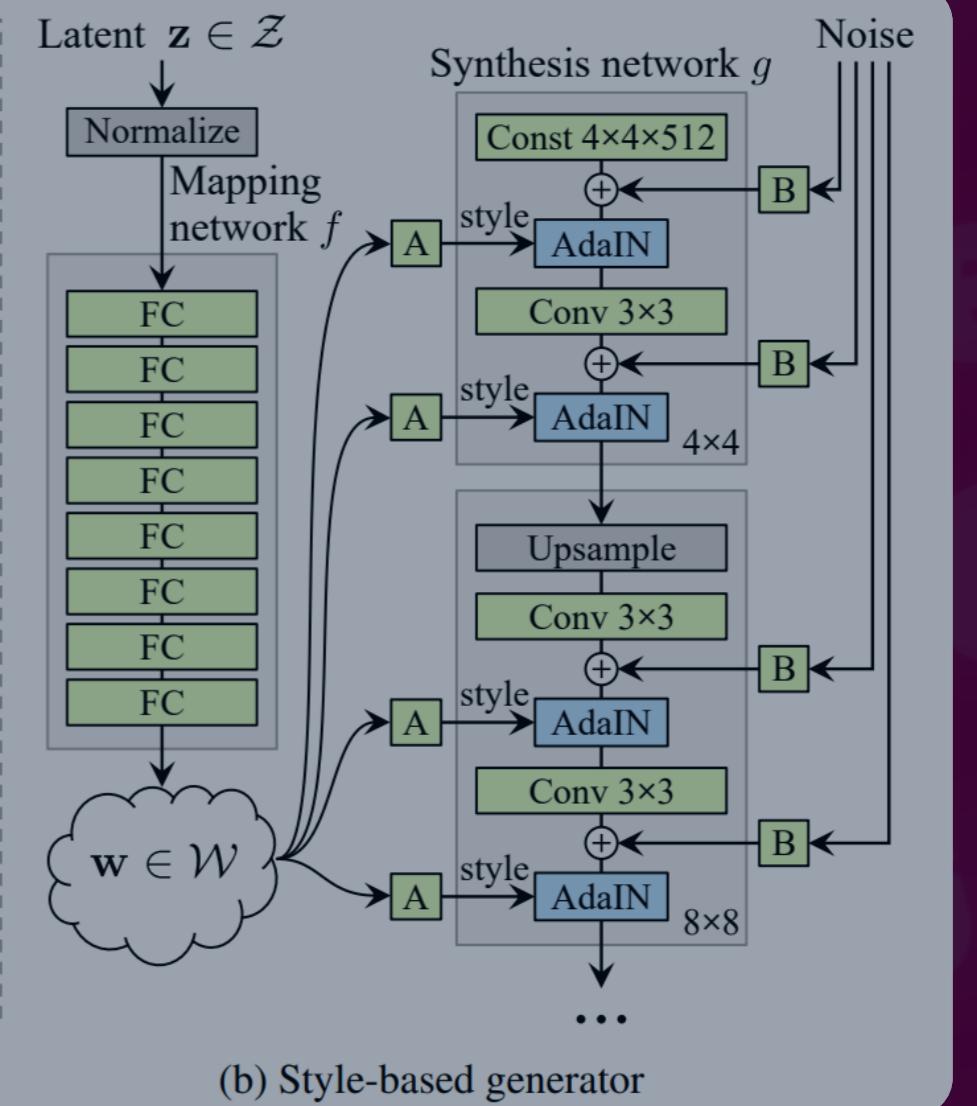
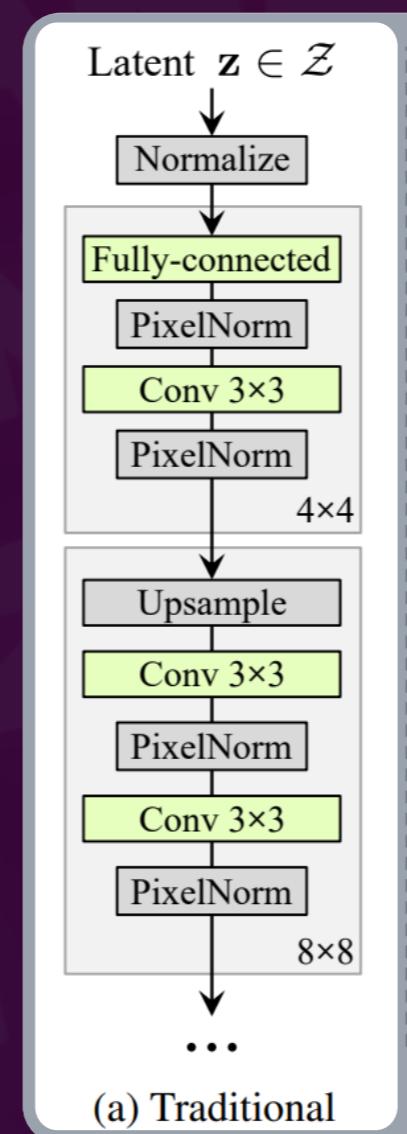
6

В результате оказалось, что архитектура StyleGAN не только отлично разделяет различные характеристики (стили в оригинальной статье), но и значительно улучшает качество генерации.

StyleGAN

7

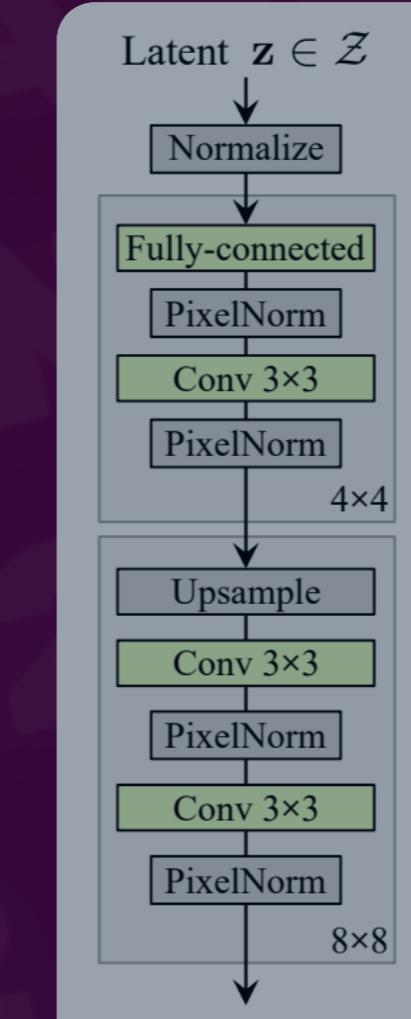
Традиционный подход к GAN предполагает трансформацию случайного вектора скрытых переменных с помощью ряда разнообразных слоёв.



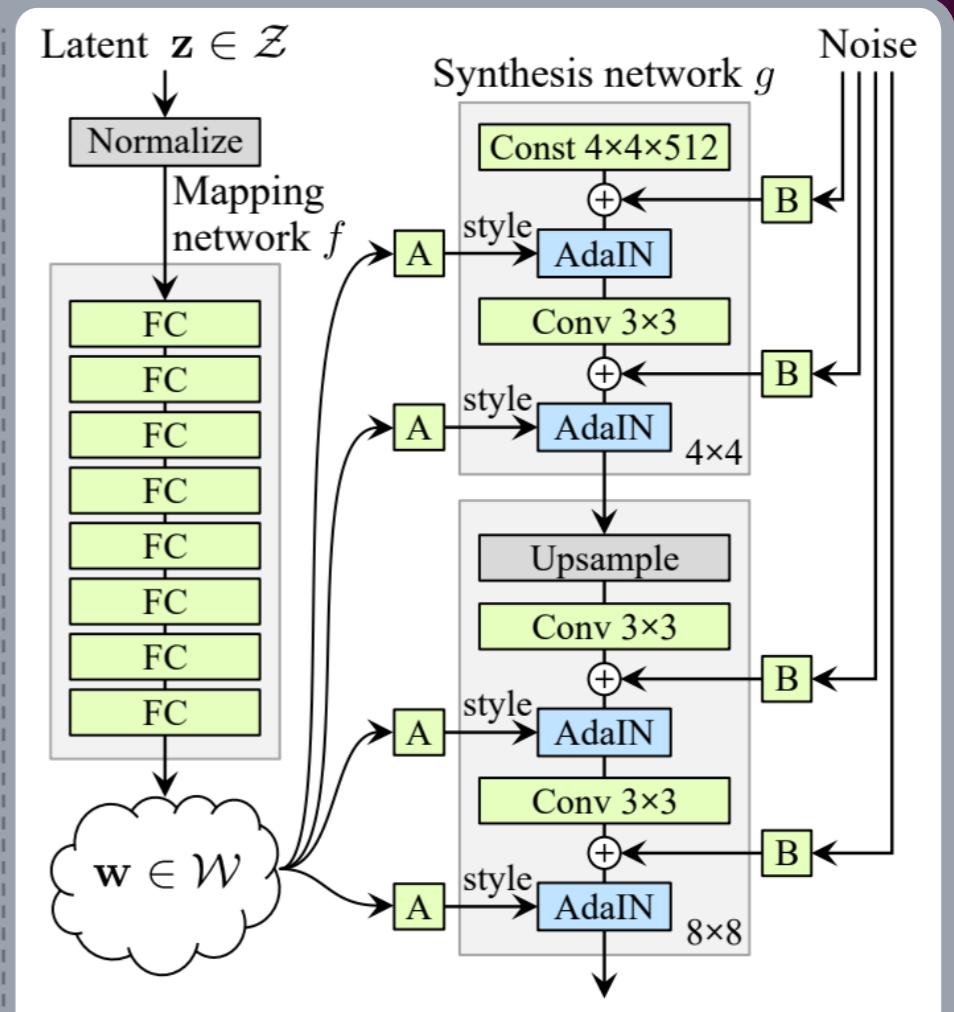
StyleGAN

8

StyleGAN делает вход константным, а случайность применяет к набору стилей. Также на каждом слое добавляется шум, позволяющий разнообразить работу нейросети.



(a) Traditional

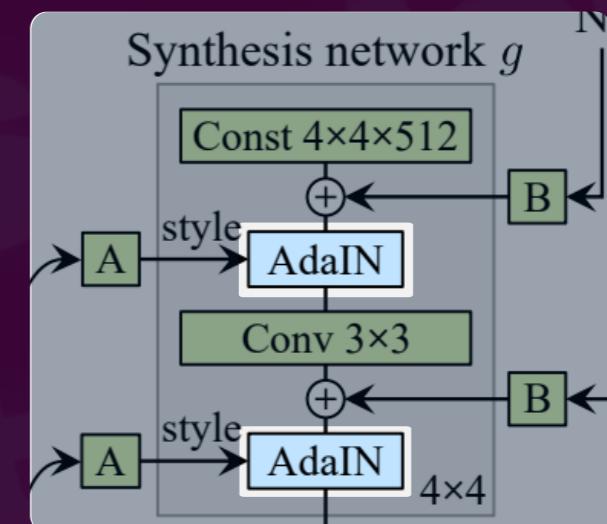
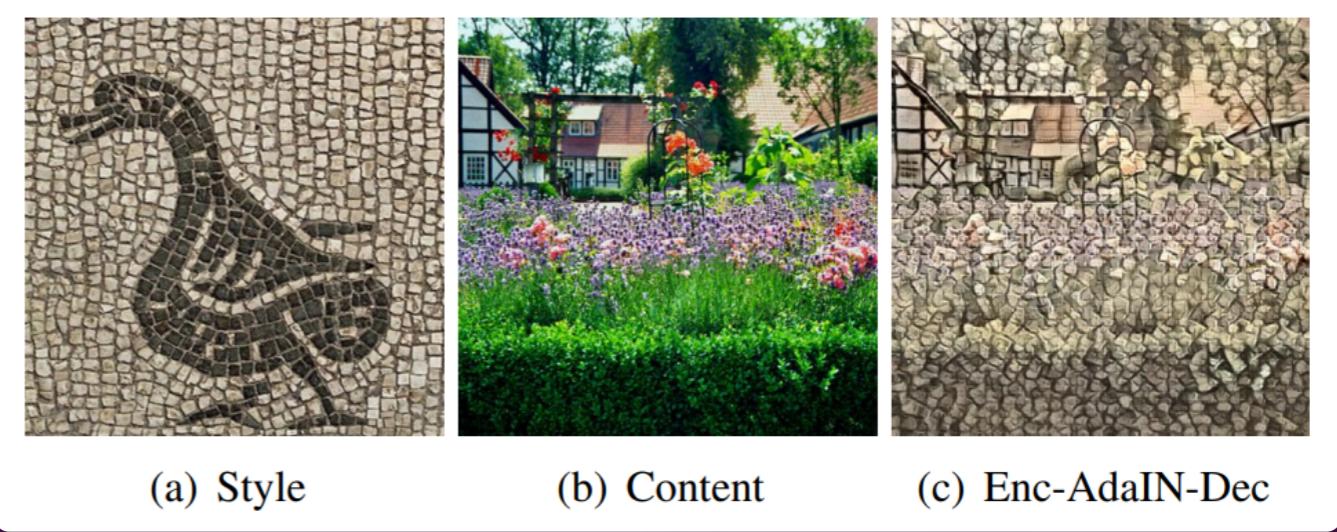


(b) Style-based generator

AdalIN

9

AdalIN – слой, отвечающий за перенос стиля.
Он нормирует среднее и выборочную дисперсию текущего вектора на соответствующие показатели вектора, задающего стиль.



Качество

10

Method	CelebA-HQ	FFHQ
A Baseline Progressive GAN [30]	7.79	8.04
B + Tuning (incl. bilinear up/down)	6.11	5.25
C + Add mapping and styles	5.34	4.85
D + Remove traditional input	5.07	4.88
E + Add noise inputs	5.06	4.42
F + Mixing regularization	5.17	4.40

Table 1. Fréchet inception distance (FID) for various generator designs (lower is better). In this paper we calculate the FIDs using 50,000 images drawn randomly from the training set, and report the lowest distance encountered over the course of training.

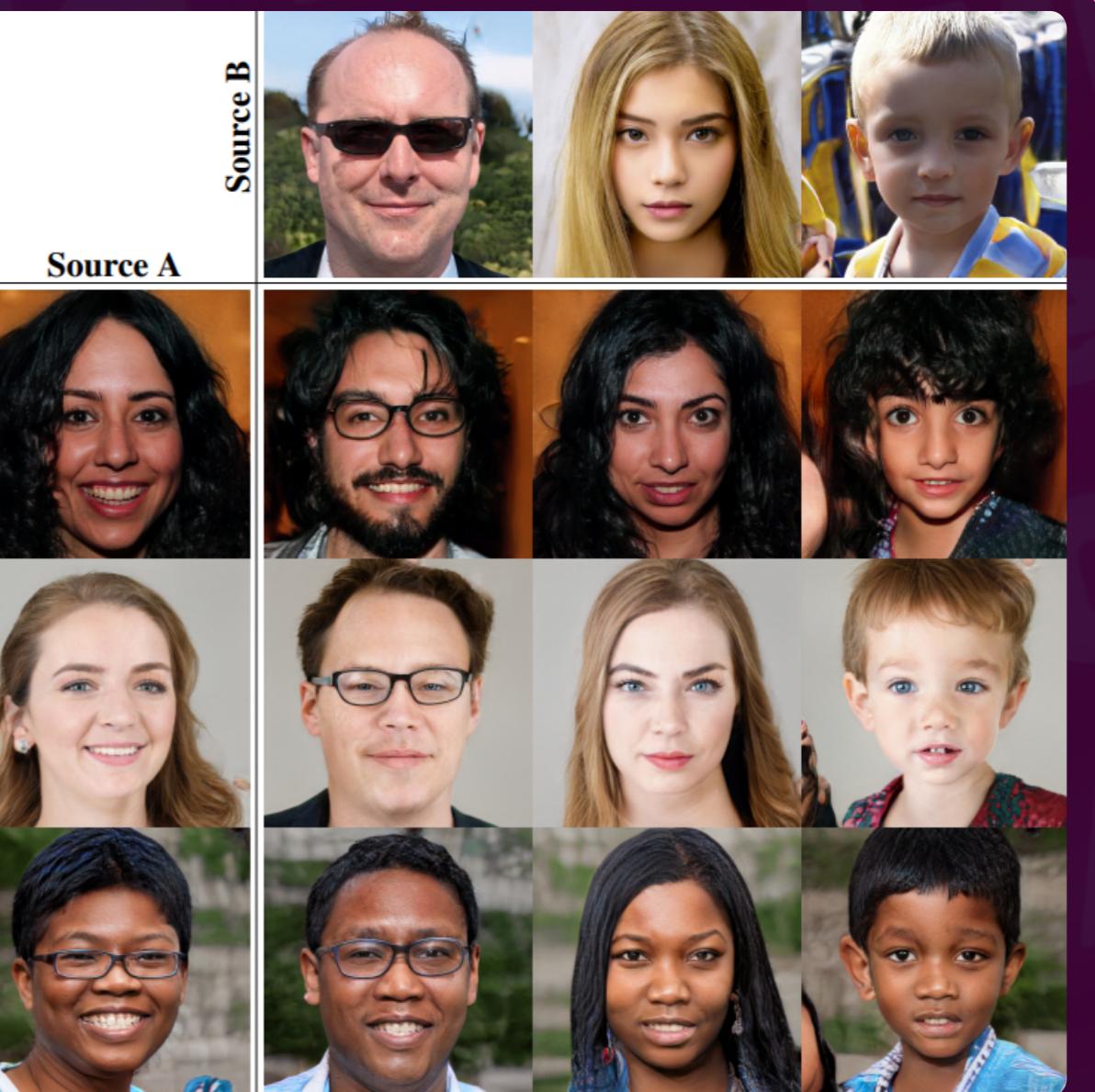


Figure 2. Uncurated set of images produced by our style-based generator (config F) with the FFHQ dataset. Here we used a variation of the truncation trick [42, 5, 34] with $\psi = 0.7$ for resolutions $4^2 - 32^2$. Please see the accompanying video for more results.

Смешение стилей

11

Если взять два сгенерированных изображения и скомбинировать их стили, получится изображение, наследующее признаки данных двух. Можно выбирать, откуда какой признак брать.



Шум и детали

12

Добавление шума в каждом слое помогает реализовать генерацию таких тонких деталей, как волосы, детали фона, поры. Чем раньше добавляется шум, тем на более крупные детали он влияет. Без шума изображения получаются размытыми.

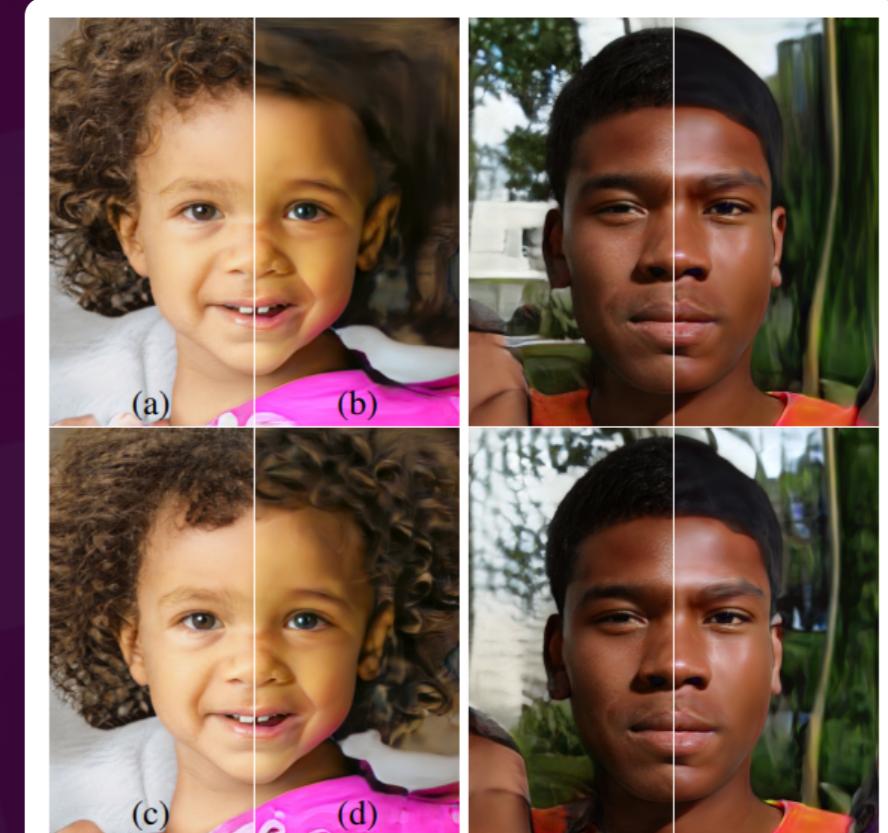


Figure 5. Effect of noise inputs at different layers of our generator. (a) Noise is applied to all layers. (b) No noise. (c) Noise in fine layers only ($64^2 - 1024^2$). (d) Noise in coarse layers only ($4^2 - 32^2$). We can see that the artificial omission of noise leads to featureless “painterly” look. Coarse noise causes large-scale curling of hair and appearance of larger background features, while the fine noise brings out the finer curls of hair, finer background detail, and skin pores.

Несчетаемые стили

13

В случае, если в пространстве стилей некоторые комбинации отсутствуют, функция, переводящая скрытые переменные в стили, искажает пространство так, чтобы оно реализовывало только представленные комбинации.

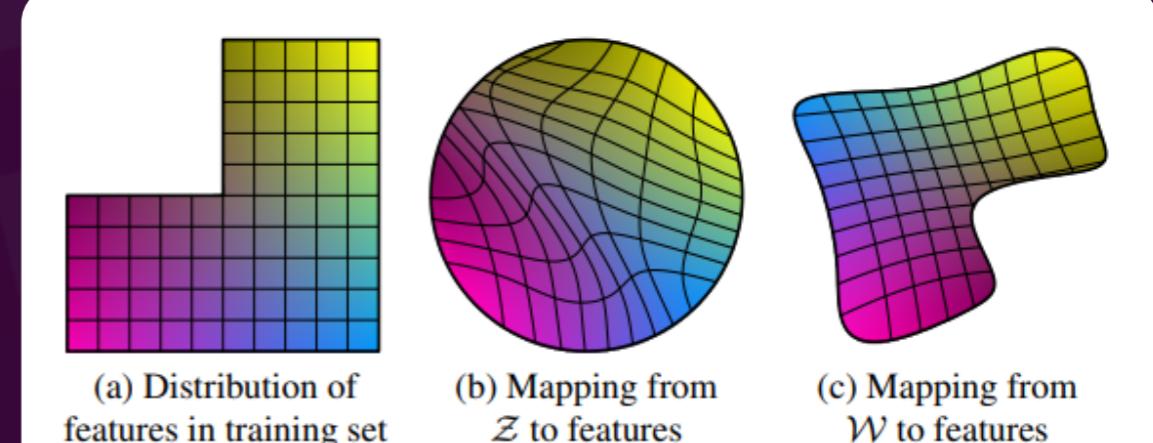


Figure 6. Illustrative example with two factors of variation (image features, e.g., masculinity and hair length). (a) An example training set where some combination (e.g., long haired males) is missing. (b) This forces the mapping from \mathcal{Z} to image features to become curved so that the forbidden combination disappears in \mathcal{Z} to prevent the sampling of invalid combinations. (c) The learned mapping from \mathcal{Z} to \mathcal{W} is able to “undo” much of the warping.

Разделимость

14

Авторы StyleGAN исследовали, насколько сильно отличаются изображения с похожими векторами в пространстве скрытых переменных и в пространстве стилей. Выяснилось, что более глубокие сети, показывающие лучшее качество, генерируют более близкие изображения при схожих векторах стилей, однако более далёкие при схожих векторах скрытых переменных. Это подтвердило гипотезу о запутанности пространства скрытых переменных традиционных GAN и о хорошей линейной разделимости пространства стилей. Добавление промежуточного линейно разделимого пространства улучшило даже обычные GAN.

StyleGAN2

15

Основная цель, которую преследовали при создании StyleGAN2 – борьба с артефактами StyleGAN. Избавление от этих артефактов повлекло не только улучшение качества генерации, но и лучшую обратимость сети: это позволило находить для изображений их прообразы в пространстве стилей.

Артефакты-капли

16

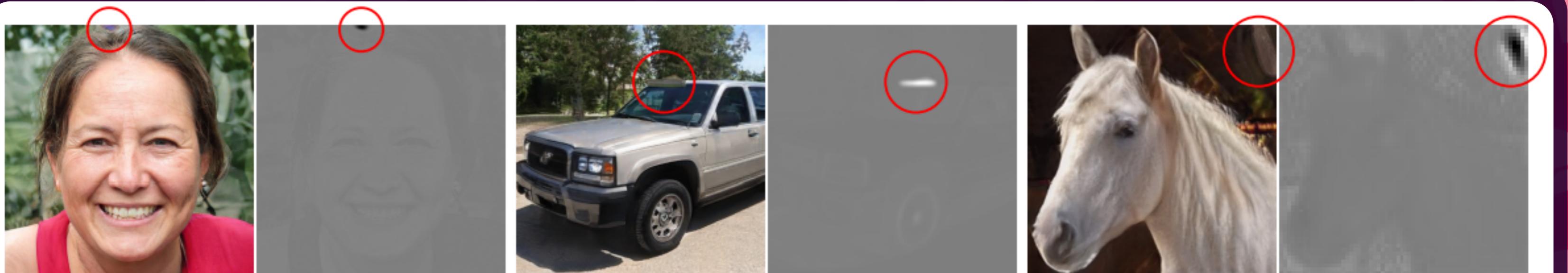


Figure 1. Instance normalization causes water droplet -like artifacts in StyleGAN images. These are not always obvious in the generated images, but if we look at the activations inside the generator network, the problem is always there, in all feature maps starting from the 64x64 resolution. It is a systemic problem that plagues all StyleGAN images.

Авторы статьи предполагают, что причина каплевидных артефактов кроется в слое AdalN.

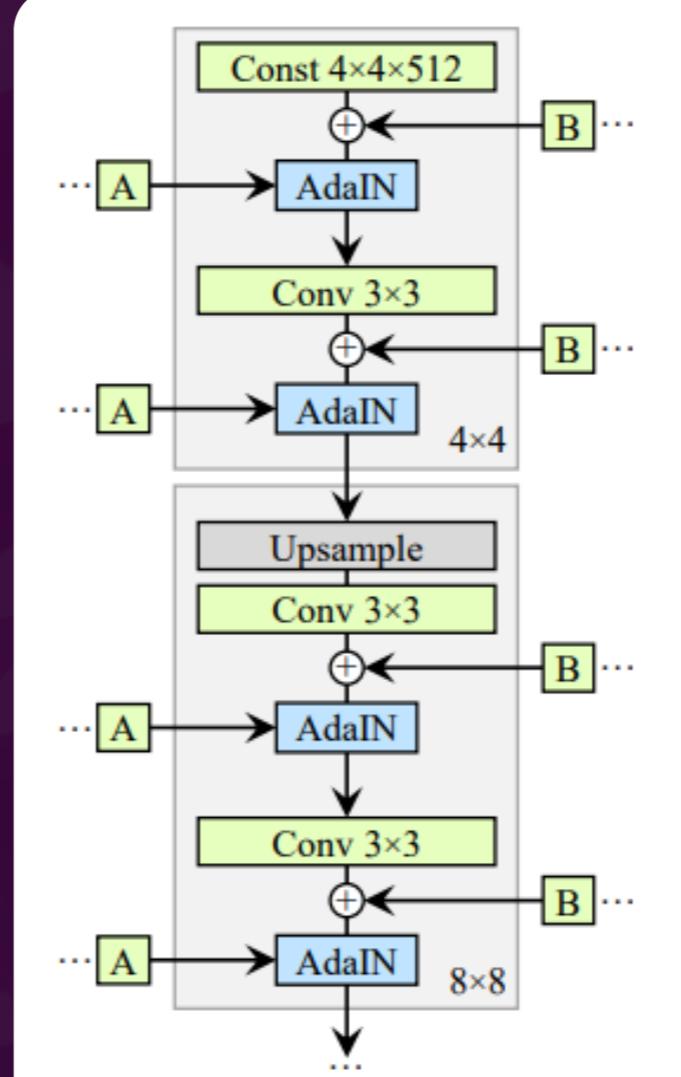
Артефакты-капли

17

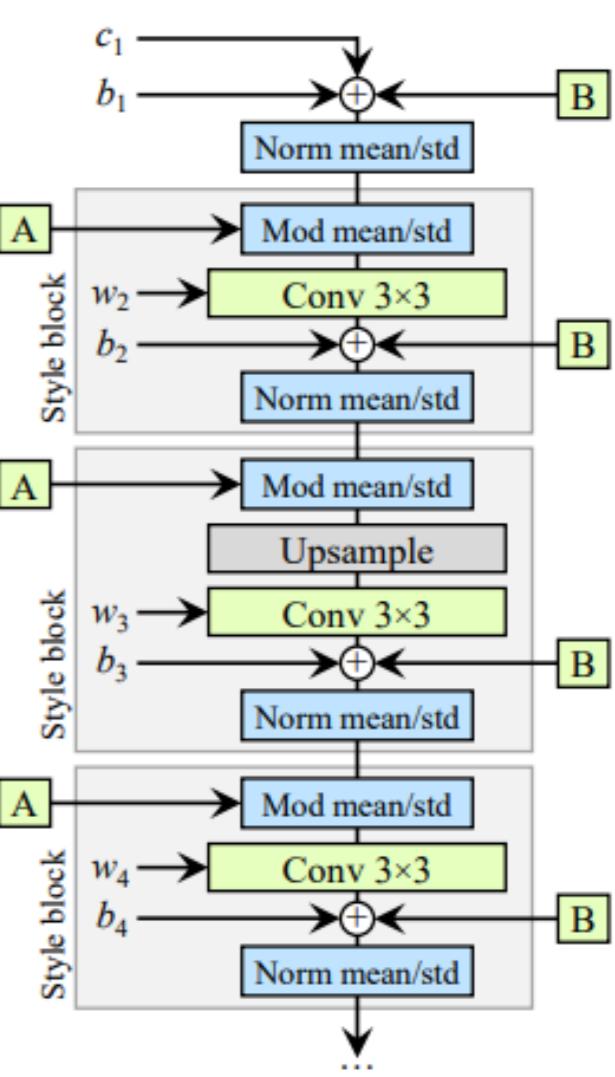
Причина появления капель, по мнению авторов, заключается в том, что генератор «проносит» через нормализацию сильный сигнал, чтобы переиспользовать его на следующих слоях. В частности, при отсутствии нормализации таких артефактов нет, а изображения StyleGAN без этих артефактов, как правило, очень плохие.

Новая архитектура

Проследим за изменениями в архитектуре, помогающими устранить капли. Для начала разделяем AdaIN слой на нормализацию и модуляцию (сеть не меняется).



(a) StyleGAN

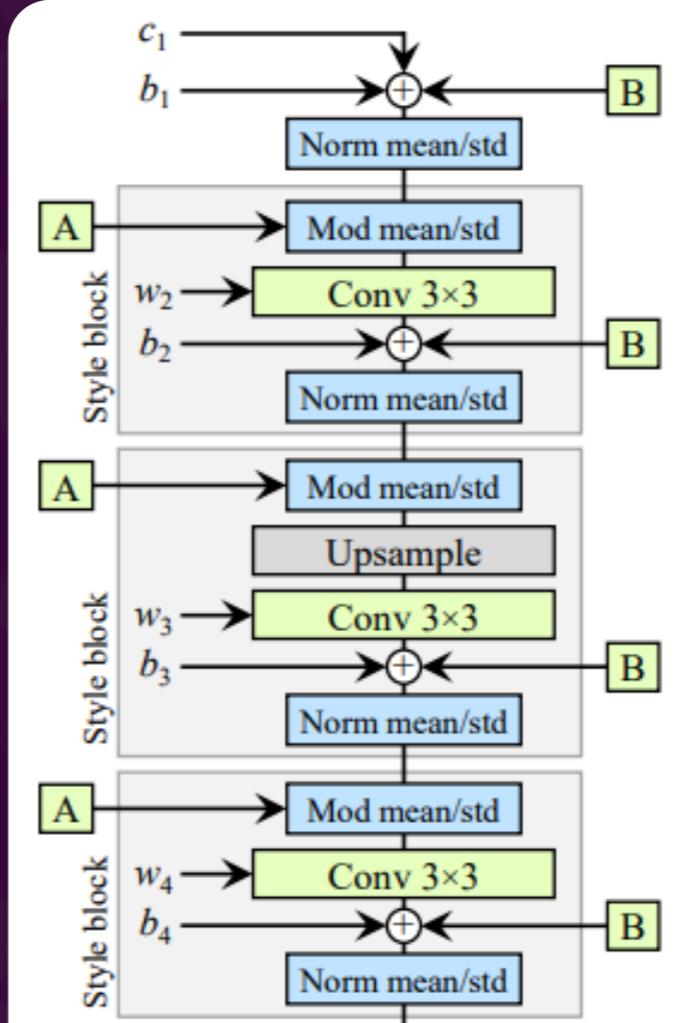


(b) StyleGAN (detailed)

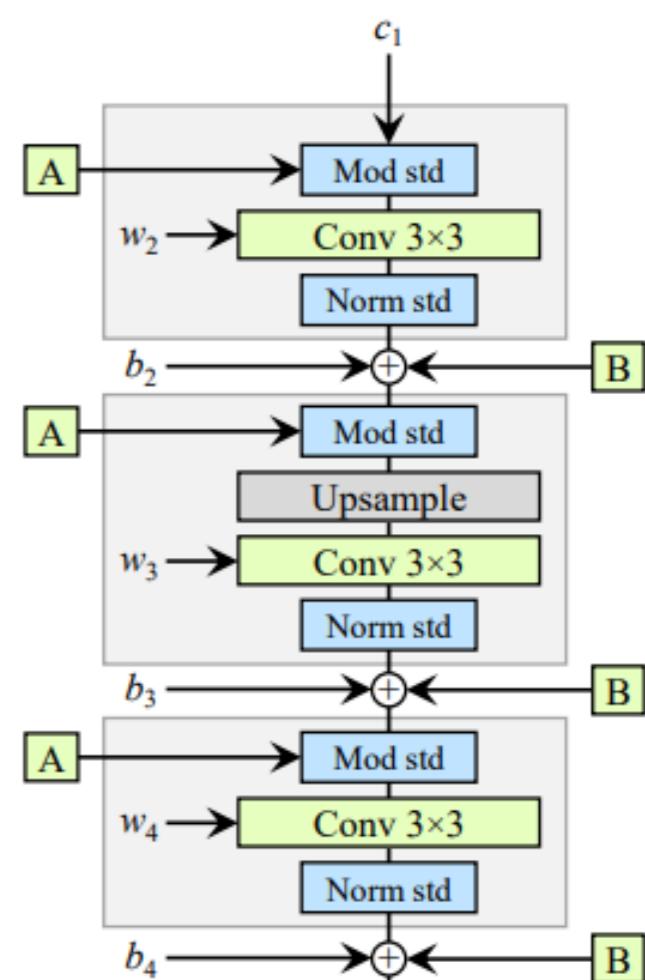
Новая архитектура

19

Теперь переместим
добавление шума: будем
добавлять его до модуляции,
а не после.



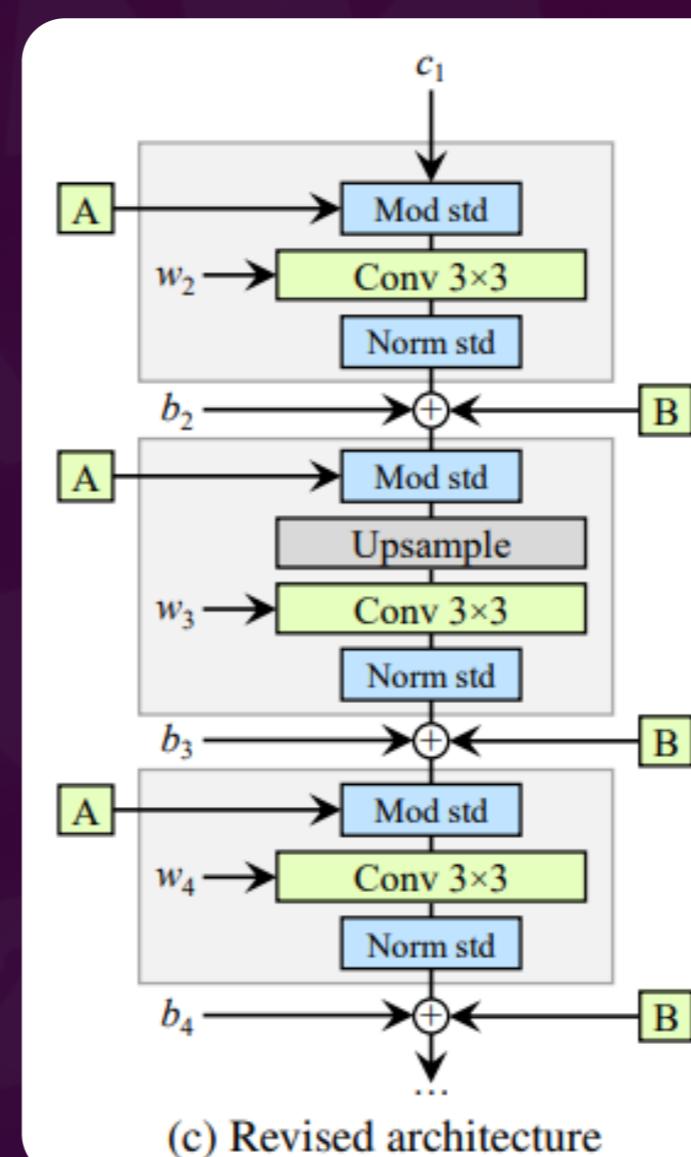
(b) StyleGAN (detailed)



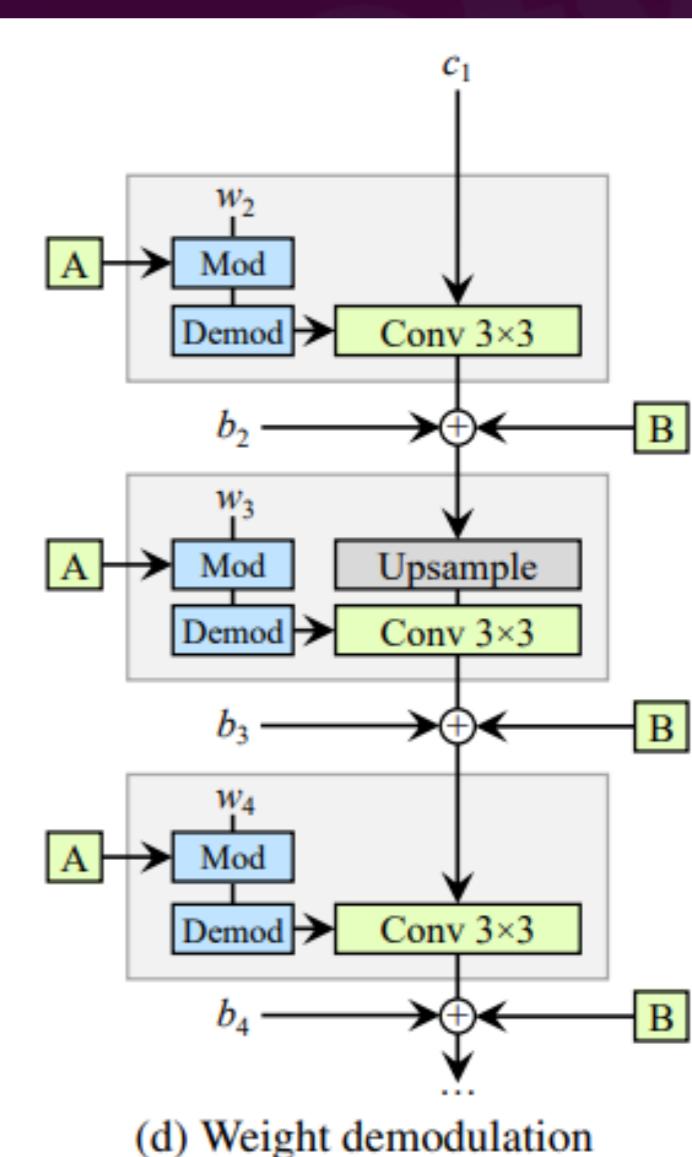
(c) Revised architecture

Новая архитектура

Заменим нормирование на демодуляцию: отмену эффекта модуляции с учётом действия свёртки. Такое изменение препятствует сохранению резких вспышек-капель.



(c) Revised architecture

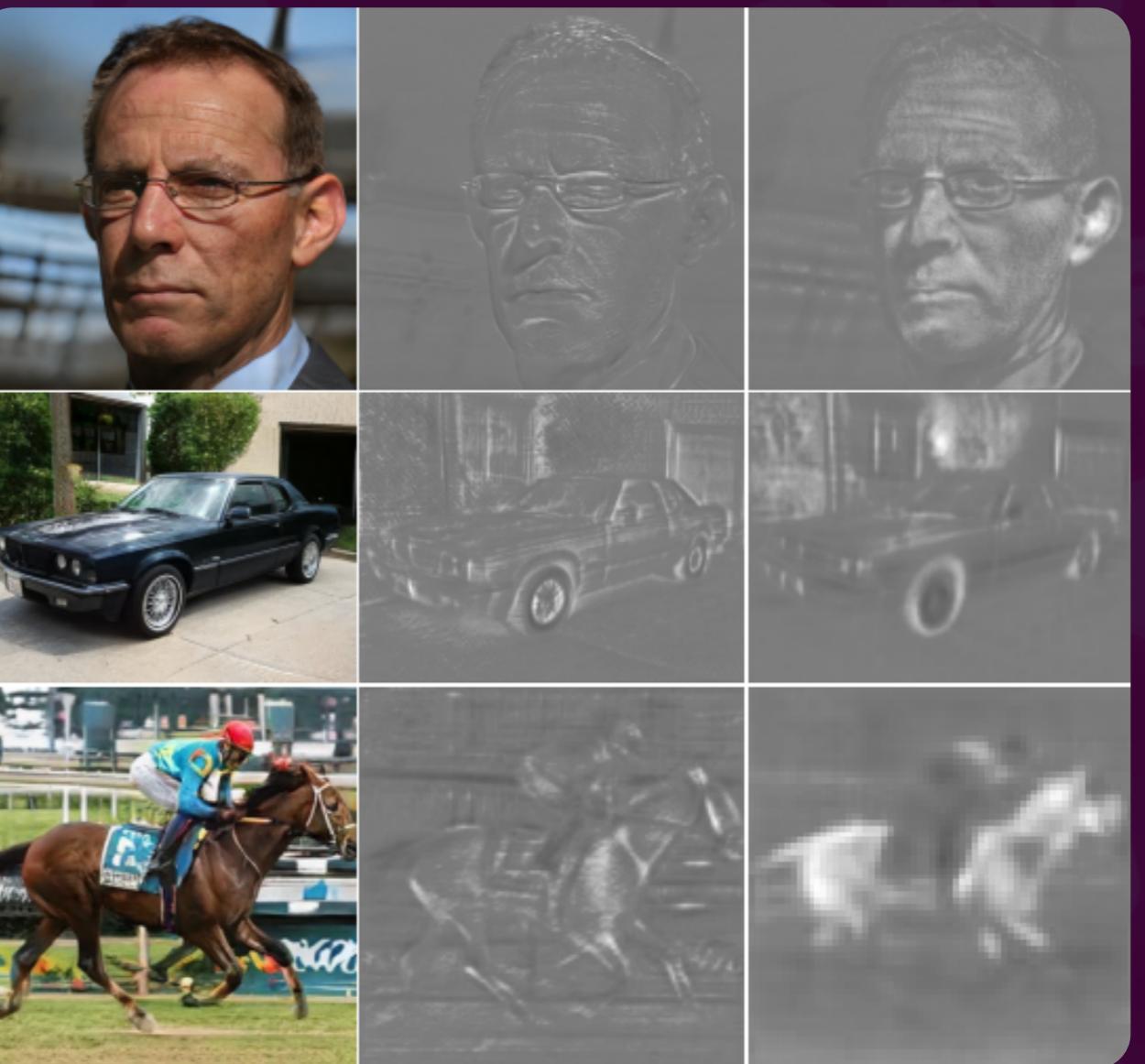


(d) Weight demodulation

Новая архитектура

21

С новой архитектурой
артефакты-капли исчезли!



PPL-регуляризация

22

Используемые для обучения метрики, такие как мера Фреше, порой пропускают картинки с серьёзными артефактами. Бороться с этим помогает регуляризация с помощью метрики PPL (perceptual path length). Чем ниже метрика PPL, тем более гладкое преобразование из скрытых переменных в изображения.

PPL-регуляризация

23

Авторы выдвинули следующую гипотезу: сеть наказывается за плохие изображения, что при ограничении на гладкость вынуждает её растягивать области хороших изображений и сужать участки, превращающиеся в некачественные картинки. На практике добавление PPL улучшает качество генерации.

PPL-регуляризация

24

PPL регуляризация довольно дорогостоящая, поэтому используется только на 1 минибатче из 16. Этот приём авторы назвали ленивой регуляризацией, и её оказывается достаточно для значительного улучшения качества.

Прогрессивный рост

25

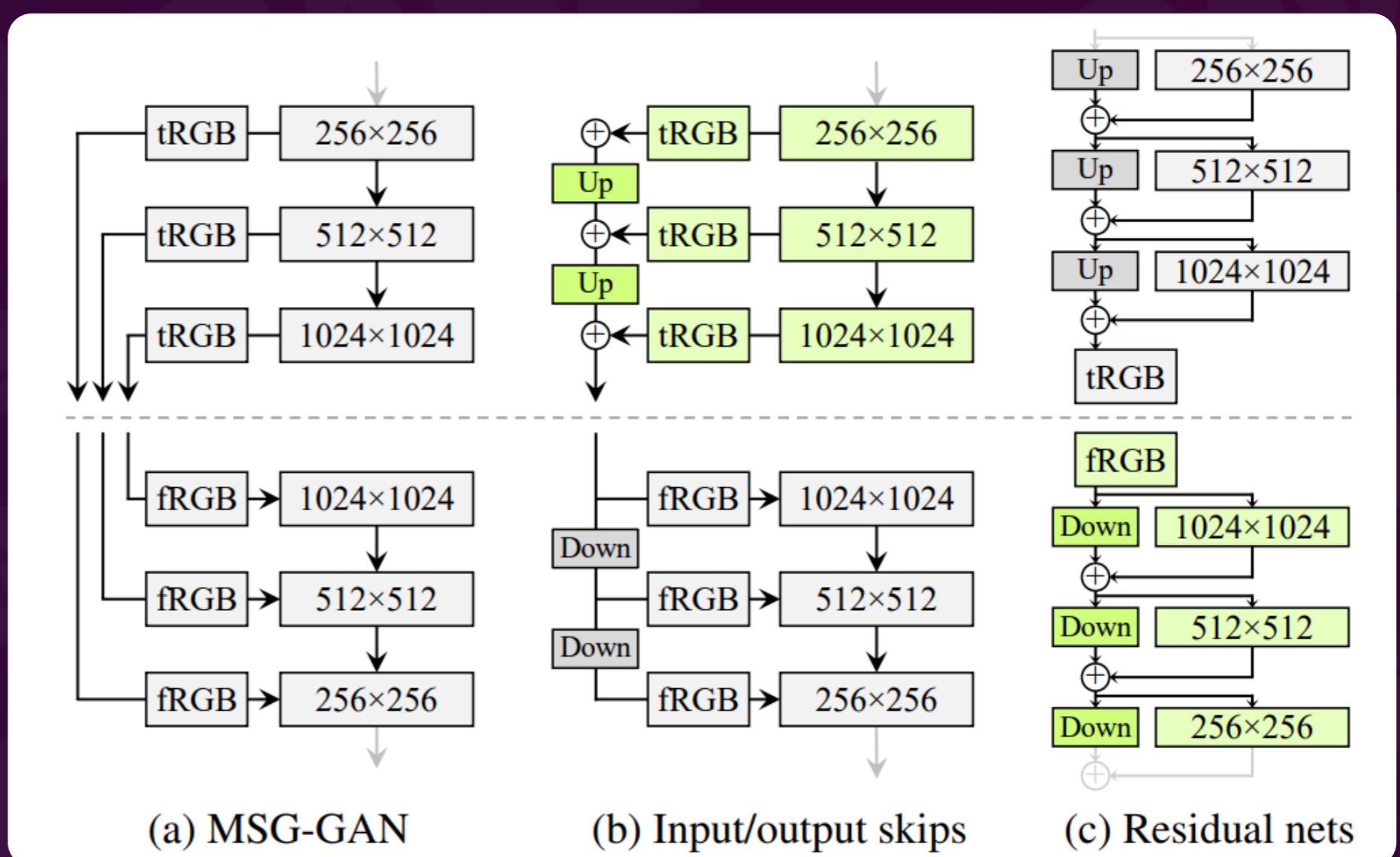
Прогрессивное
увеличение размера
изображений улучшает
качество, но порождает
специфические
артефакты, привязанные
к положению на фото.



Skip connections

26

Пробуем разные
skip connections,
выбираем лучший
вариант.



Skip connections

27

Пробуем разные
skip connections,
выбираем лучший
вариант.

FFHQ	D original		D input skips		D residual	
	FID	PPL	FID	PPL	FID	PPL
G original	4.32	265	4.18	235	3.58	269
G output skips	4.33	169	3.77	127	3.31	125
G residual	4.35	203	3.96	229	3.79	243

LSUN Car	D original		D input skips		D residual	
	FID	PPL	FID	PPL	FID	PPL
G original	3.75	905	3.23	758	3.25	802
G output skips	3.77	544	3.86	316	3.19	471
G residual	3.93	981	3.40	667	2.66	645

Table 2. Comparison of generator and discriminator architectures without progressive growing. The combination of generator with output skips and residual discriminator corresponds to configuration E in the main result table.

Прогрессивный рост

28

Если посмотреть на вклад слоёв каждого размера в сумму, видно, что потенциал высоких размеров не использован. Авторы обучили сеть больших размеров и значительно улучшили результаты генерации более крупных изображений.

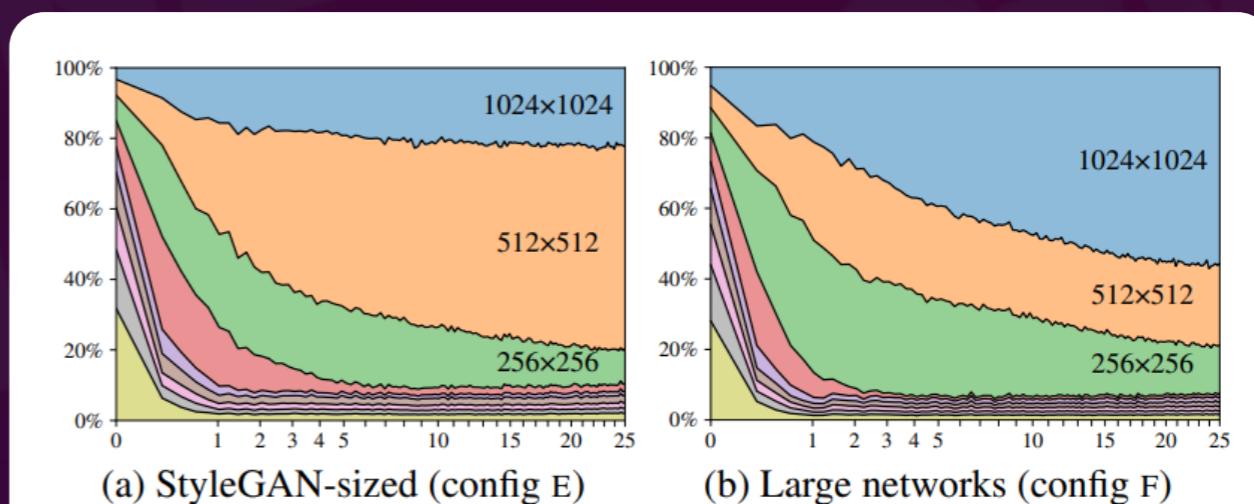


Figure 8. Contribution of each resolution to the output of the generator as a function of training time. The vertical axis shows a breakdown of the relative standard deviations of different resolutions, and the horizontal axis corresponds to training progress, measured in millions of training images shown to the discriminator. We can see that in the beginning the network focuses on low-resolution images and progressively shifts its focus on larger resolutions as training progresses. In (a) the generator basically outputs a 512^2 image with some minor sharpening for 1024^2 , while in (b) the larger network focuses more on the high-resolution details.

Восстановление

29

Попробуем восстановить по изображению его прообраз в пространстве стилей. Для этого сеть фиксируется, а затем поочерёдно подбирается оптимальный стиль на каждом слое. Авторы также добавляют регуляризованный шум меньшей силы, чем при генерации, для более тщательного исследования пространства.

Восстановление

30



StyleGAN — generated images



StyleGAN2 — generated images



StyleGAN2 — real images

Figure 9. Example images and their projected and re-synthesized counterparts. For each configuration, top row shows the target images and bottom row shows the synthesis of the corresponding projected latent vector and noise inputs. With the baseline StyleGAN, projection often finds a reasonably close match for generated images, but especially the backgrounds differ from the originals. The images generated using StyleGAN2 can be projected almost perfectly back into generator inputs, while projected real images (from the training set) show clear differences to the originals, as expected. All tests were done using the same projection method and hyperparameters.

Восстановление

31

По диаграммам видно, что изображения, сгенерированные StyleGAN2 восстанавливаются ей гораздо лучше, чем StyleGAN восстанавливает свои!

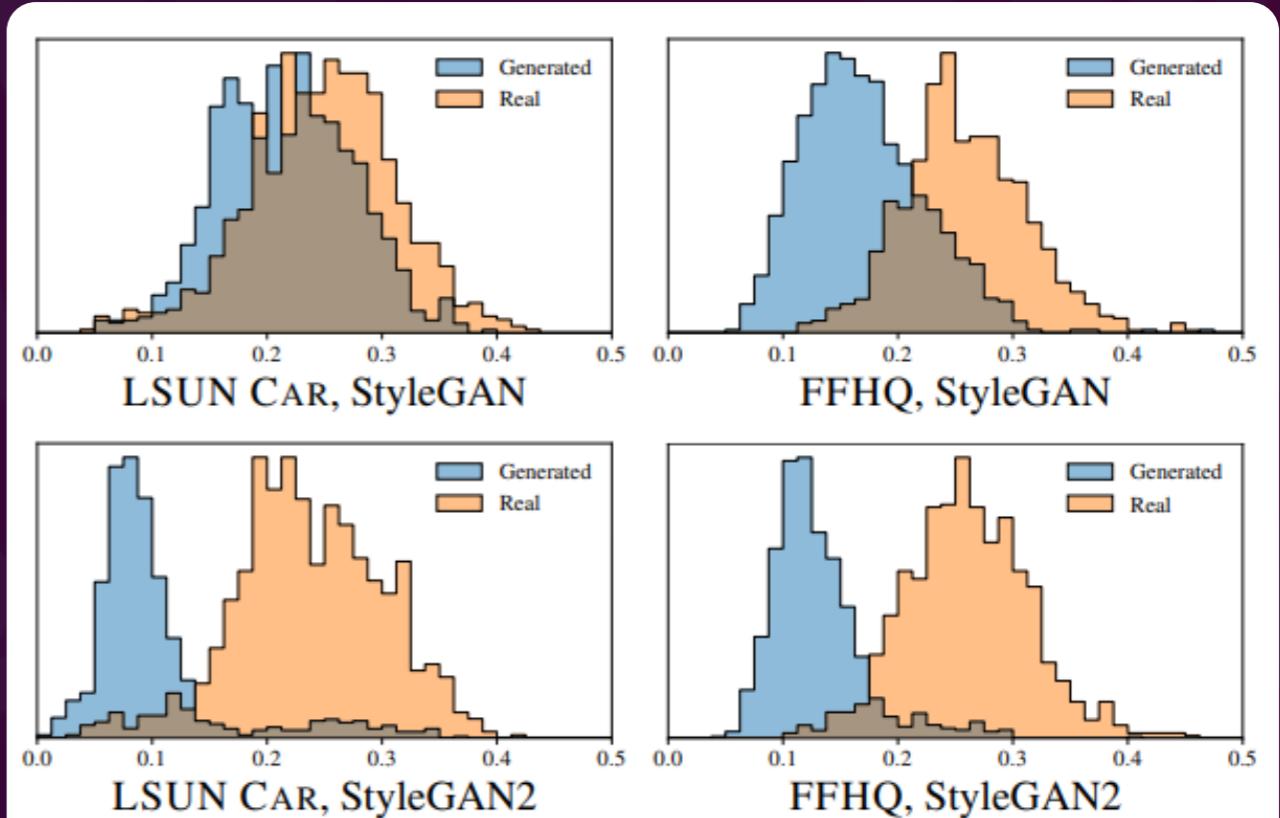


Figure 10. LPIPS distance histograms between original and projected images for generated (blue) and real images (orange). Despite the higher image quality of our improved generator, it is much easier to project the generated images into its latent space \mathcal{W} . The same projection method was used in all cases.

Результаты StyleGAN2

32

Вторая версия оказалась не только качественней, но и на 40% быстрее первой. На всё исследование ушёл 51 год single-GPU вычислений и 132 мегаватт-часа электроэнергии, что примерно равно электроэнергии, потребляемой 20 жителями России за год (данные EIA за 2021 год).

Выводы

33

Генеративно-состязательные сети противопоставляют генератор изображений дискриминатору для генерации изображений, похожих на предоставленные. StyleGAN радикально изменяет архитектуру решения, сохраняя исходную идею, и добивается куда лучшего качества.

Выводы

34

StyleGAN2 избавляется от артефактов предыдущей версии, параллельно упрощая и ускоряя обучение за счёт перестановки операций и замены нормировки на демодуляцию.

Источники

35

MIT Introduction to Deep Learning 6.S191: Lecture 4

A Style-Based Generator Architecture for Generative Adversarial Networks (Karras et al.)

Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization (Huang et al.)

Analyzing and Improving the Image Quality of StyleGAN (Karras et al.)