

Style GAN 2

Бобков Денис 192





Style GAN 2

Что предложено нового

- Изменена нормализация в генераторе
- Добавлена PPL регуляризация
- Изменена система увеличения изображений в процессе обучения
- Рассмотрено влияние разрешения на конечную картинку
- Предложен новый способ отображения на латентное пространство

Проблема пузыря



Проблема пузыря



Гипотеза – проблема в нормализации! Генератор хитрит.

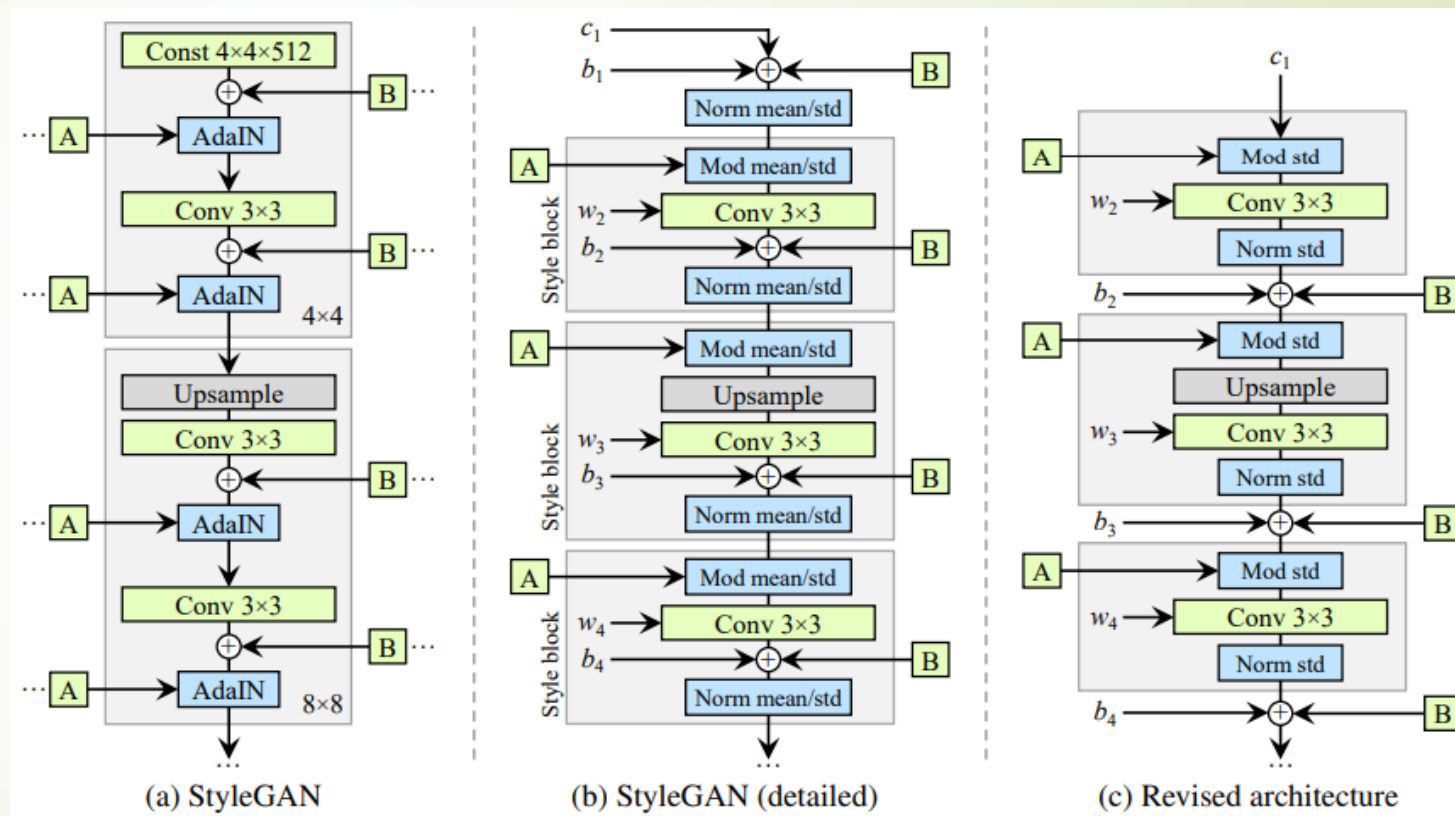


Style GAN 2

Изменения в архитектуре генератора

В прошлой архитектуре (b) смещение и зашумление данных происходит внутри блока стиля, что довольно слабо влияет на воздействие модуляции, т.к. после зашумления идёт нормализация. Предлагается вынести эти операции за блок стиля.

После этих изменений также было обнаружено, что в нормализации и модуляции нет необходимости вычитании сдвига.

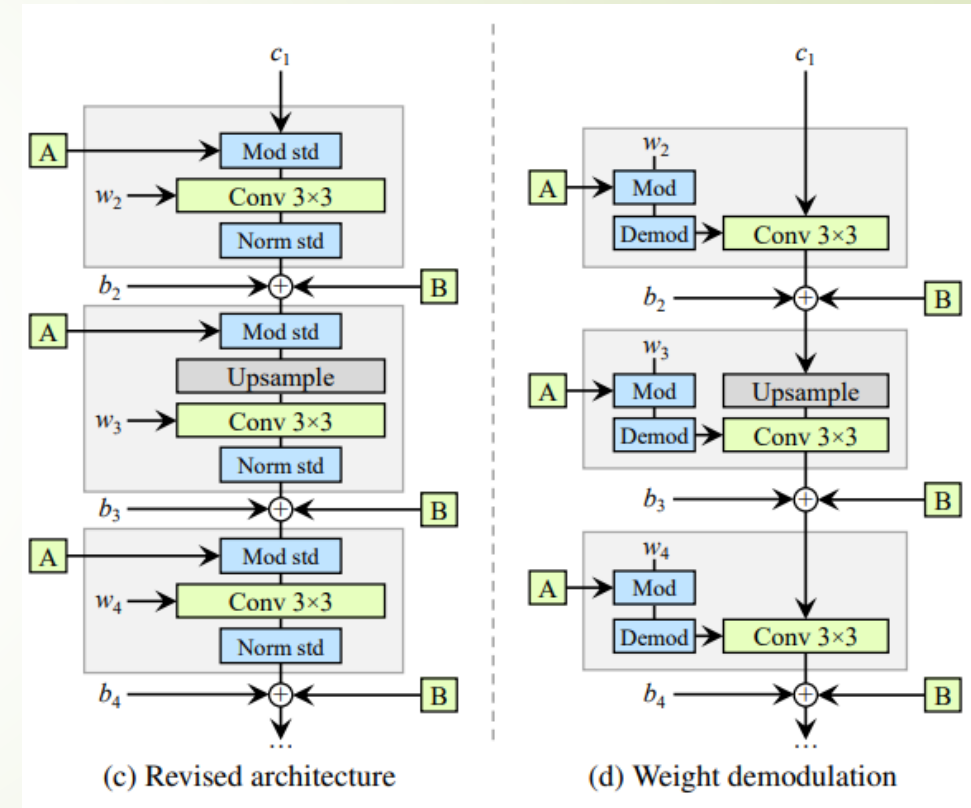


Style GAN 2

Изменения в архитектуре генератора

Так как теперь мы работаем только с домножением на значения, то если предположить, что на вход в блок стиля приходят н.сл.вел. с единичной дисперсией и нулевым матожиданием, то операции модуляции и нормализации (демодуляции) можно занести внутрь свёртки (d), что продемонстрировано в формулах ниже.

$$w'_{ijk} = s_i \cdot w_{ijk}$$



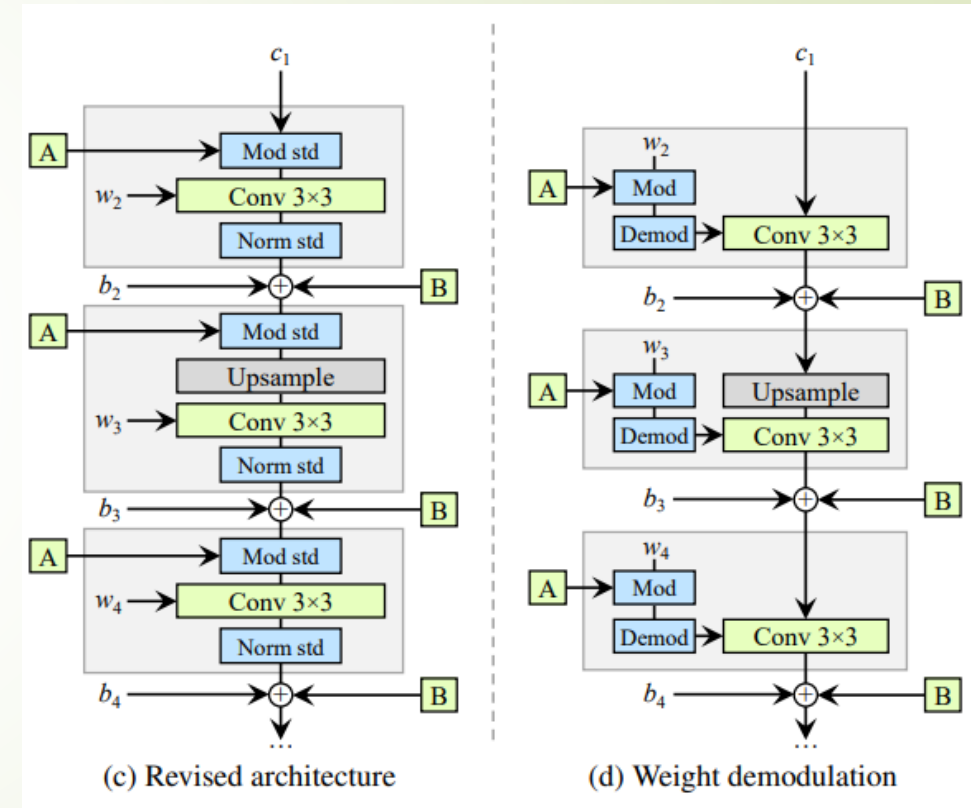
i – номер входного канала
 j – номер выходного канала
 K – номер пикселя

Style GAN 2

Изменения в архитектуре генератора

Так как теперь мы работаем только с домножением на значение, то если предположить, что на вход в блок стиля приходят н.сл.вел. с единичной дисперсией, то операции модуляции и нормализации (демодуляции) можно занести внутрь свёртки (d), что продемонстрировано в формулах ниже.

$$w'_{ijk} = s_i \cdot w_{ijk} \quad \sigma_j = \sqrt{\sum_{i,k} w'_{ijk}{}^2},$$



i – номер входного канала
 j – номер выходного канала
 K – номер пикселя

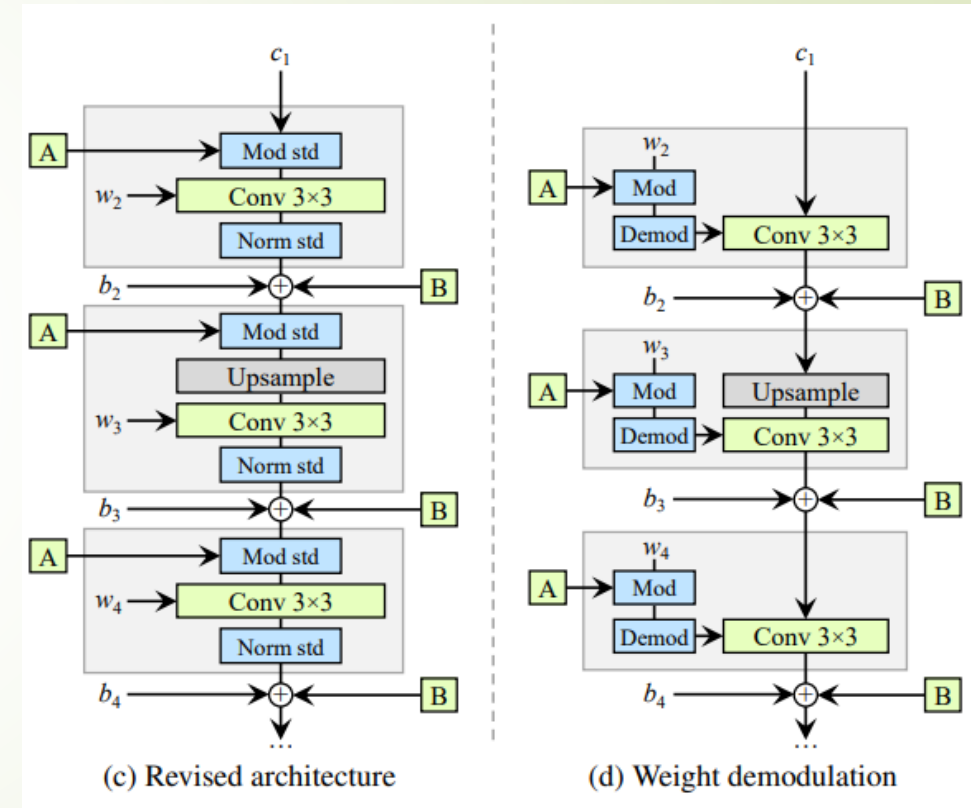
Style GAN 2

Изменения в архитектуре генератора

Так как теперь мы работаем только с домножением на значение, то если предположить, что на вход в блок стиля приходят н.сл.вел. с единичной дисперсией, то операции модуляции и нормализации (демодуляции) можно занести внутрь свёртки (d), что продемонстрировано в формулах ниже.

$$w'_{ijk} = s_i \cdot w_{ijk} \quad \sigma_j = \sqrt{\sum_{i,k} w'^2_{ijk}},$$

$$w''_{ijk} = w'_{ijk} / \sqrt{\sum_{i,k} w'^2_{ijk} + \varepsilon}$$



i – номер входного канала
 j – номер выходного канала
 K – номер пикселя

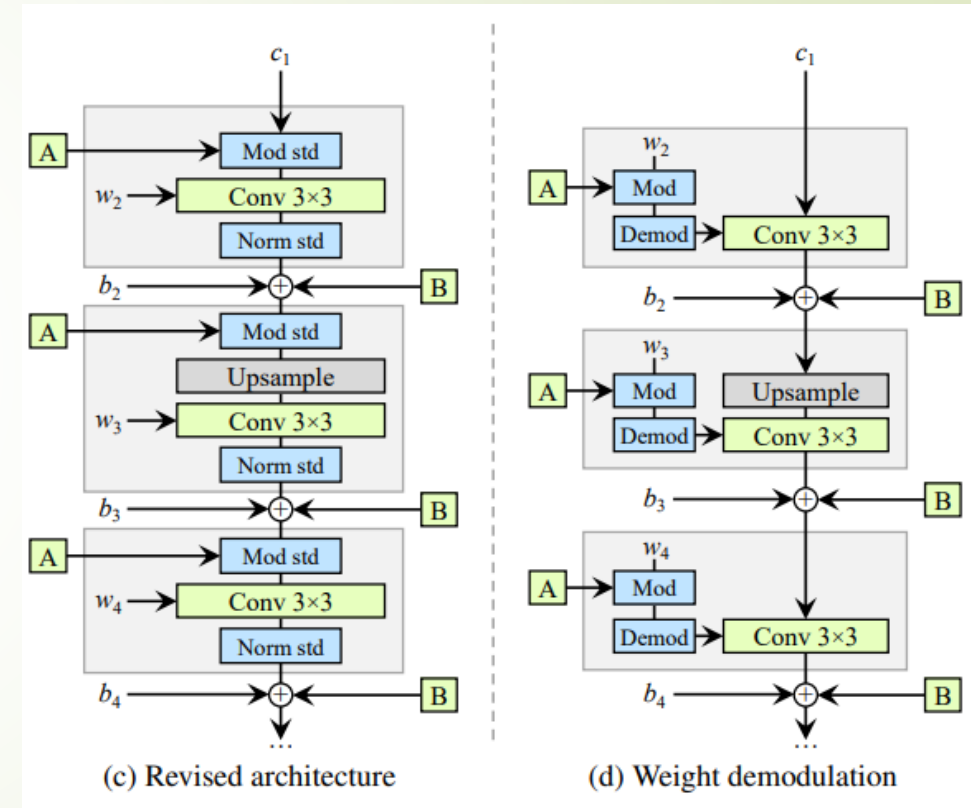
Style GAN 2

Изменения в архитектуре генератора

Теперь будем нормировать только веса свёртки (раньше нормировали ещё и входные данные).

$$w'_{ijk} = s_i \cdot w_{ijk} \quad \sigma_j = \sqrt{\sum_{i,k} w'_{ijk}{}^2},$$

$$w''_{ijk} = w'_{ijk} / \sqrt{\sum_{i,k} w'_{ijk}{}^2 + \varepsilon}$$



i – номер входного канала
 j – номер выходного канала
 K – номер пикселя

Style GAN 2

Path length regularization - идейно

На рисунке отчётливо видно, что изображения, полученные с высоким perceptual path length (PPL) сильно хуже тех, где PPL маленький (остальные метрики примерно одинаковы). Хотим добавить в нашу модель регуляризацию на основе PPL.

Также было выявлено, что регуляризацию не обязательно делать в каждом батче, а достаточно один раз в несколько батчей.



(a) Low PPL scores



(b) High PPL scores

Идея PPL (метрики) в том, что если мы возьмём достаточно близкие векторы w и w' из латентного пространства, то картинки, сгенерированные из них будут также должны быть очень похожи. Именно эту похожесть картинок и сравнивают.

Style GAN 2

Path length regularization

Если $g(w) : W \rightarrow Y$ – отображение генератора, а $J_w = \partial g(w)/\partial w$ – якобиан этого отображения, то предложенная регуляризация будет:

$$\mathbb{E}_{\mathbf{w}, \mathbf{y} \sim \mathcal{N}(0, \mathbf{I})} (\|\mathbf{J}_{\mathbf{w}}^T \mathbf{y}\|_2 - a)^2$$

Где \mathbf{y} – случайное изображение с нормально распределённой интенсивностью пикселей, а a – константа, меняющаяся в процессе решения как экспоненциальное скользящее среднее длин $\|\mathbf{J}_{\mathbf{w}}^T \mathbf{y}\|_2$

Однако полностью считать якобиан очень затратно, поэтому на практике будет использоваться формула $\mathbf{J}_{\mathbf{w}}^T \mathbf{y} = \nabla_{\mathbf{w}}(g(\mathbf{w}) \cdot \mathbf{y})$.

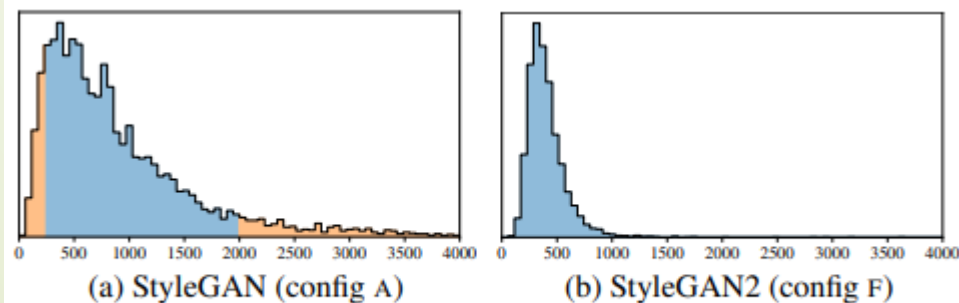


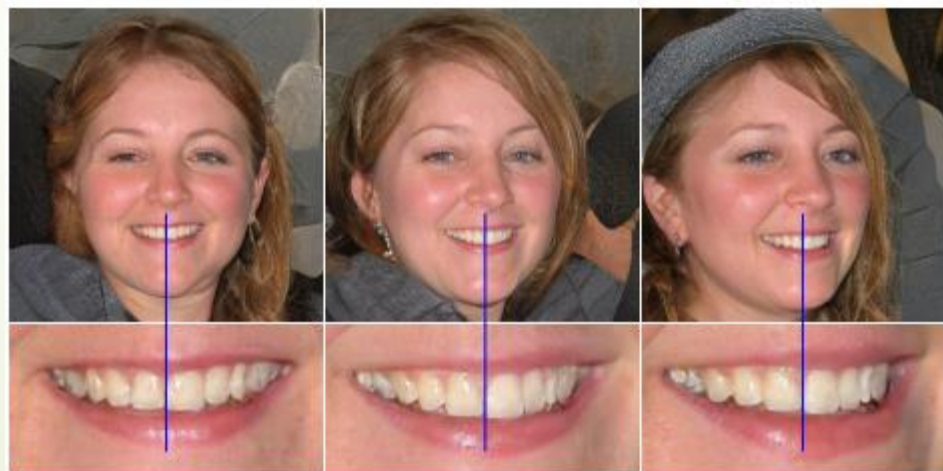
Figure 5. (a) Distribution of PPL scores of individual images

Style GAN 2

Alternative progressive growing - интуиция

Существует проблема, что некоторые признаки (глаза, зубы и т.п.) при изменении, к примеру, позы остаются неподвижными (рис.), чего быть не должно. Это происходит из-за того, что во время обучения есть момент, когда определённое разрешение считается последним для модели, и в этот момент модель сильно приучается к деталям на низких разрешениях.

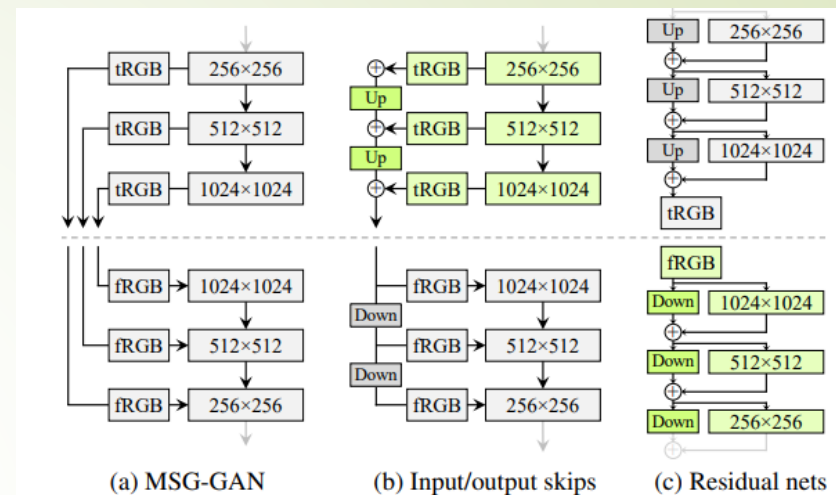
Чтобы этого избежать, предложено отойти от модели progressive growing.



Style GAN 2

Alternative progressive growing

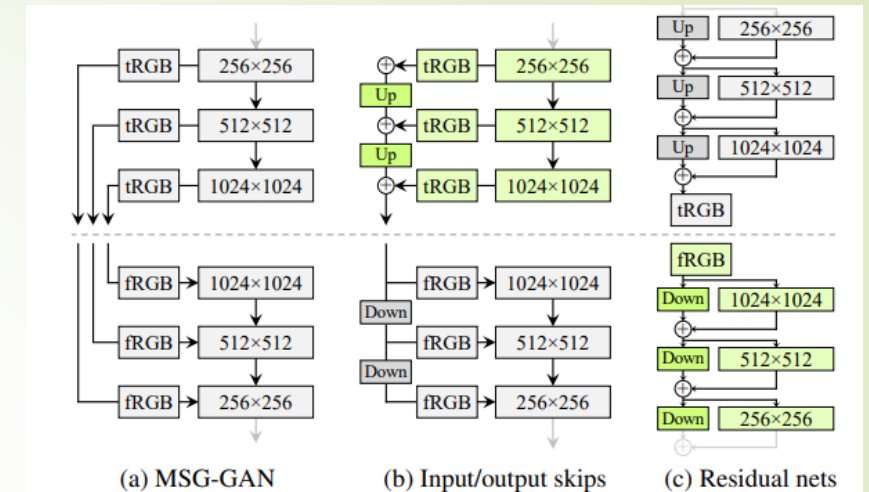
- За основу новой upsample системы взят MSG-GAN использующий несколько skip connections (a).
- В (b) эта архитектура упрощена тем, что каждый раз данные RGB апсемплятся и прибавляются к последующим.
- В (c) были добавлены residual skip connections.



Style GAN 2

Alternative progressive growing

- За основу новой upsample системы взят MSG-GAN использующий несколько skip connections.
- В (b) эта архитектура упрощена тем, что каждый раз данные RGB апсемплятся и прибавляются к последующим.
- В (c) были добавлены residual skip connections.



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

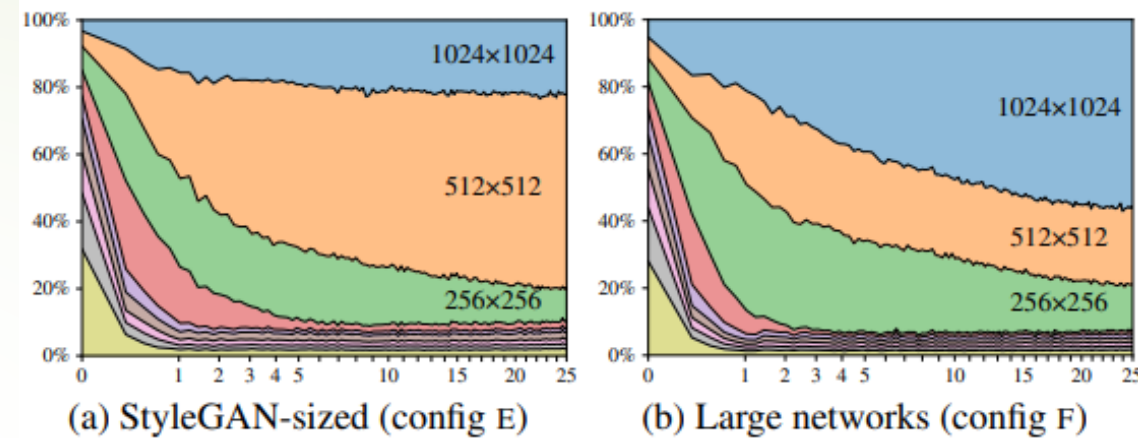
Лучший результат показала модель, у которой добавлены output skips в генераторе и residual skips в дискриминаторе.

Style GAN 2

Resolution usage

Поскольку новая модель суммирует значения RGB с разных разрешений, мы можем оценить вклад каждого из разрешений в итог.

Обнаружилось, что самое высокое разрешение имеет слишком малый импакт (a), из-за чего картинки получаются худшего качества.



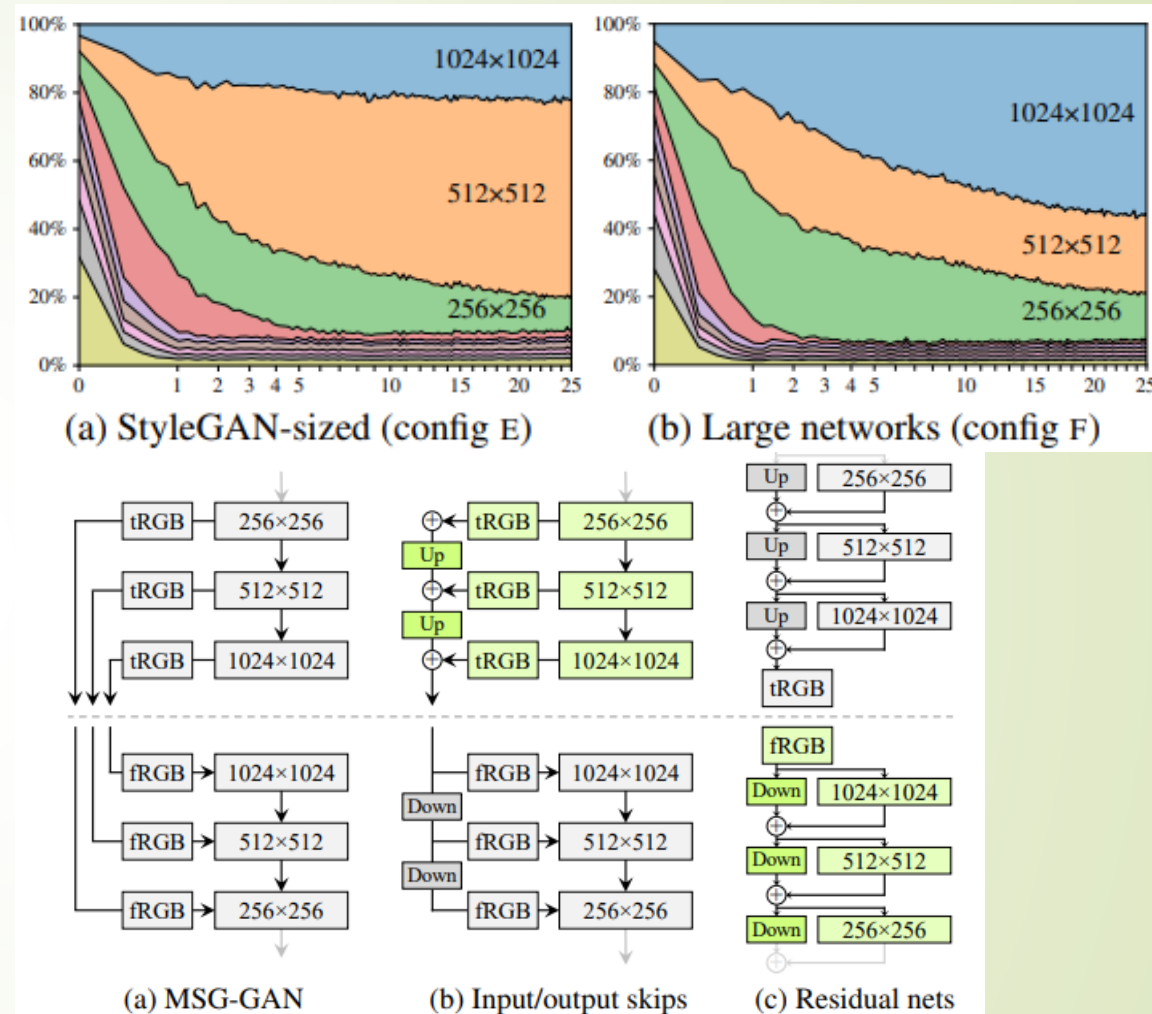
Style GAN 2

Resolution usage

Поскольку новая модель суммирует значения RGB с разных разрешений, мы можем оценить вклад каждого из разрешений в итог.

Обнаружилось, что самое высокое разрешение имеет слишком малый импакт (a), из-за чего картинки получаются худшего качества.

Это исправили удвоив количество признаков для каждого пикселя в разрешениях 64 - 1024 (b).



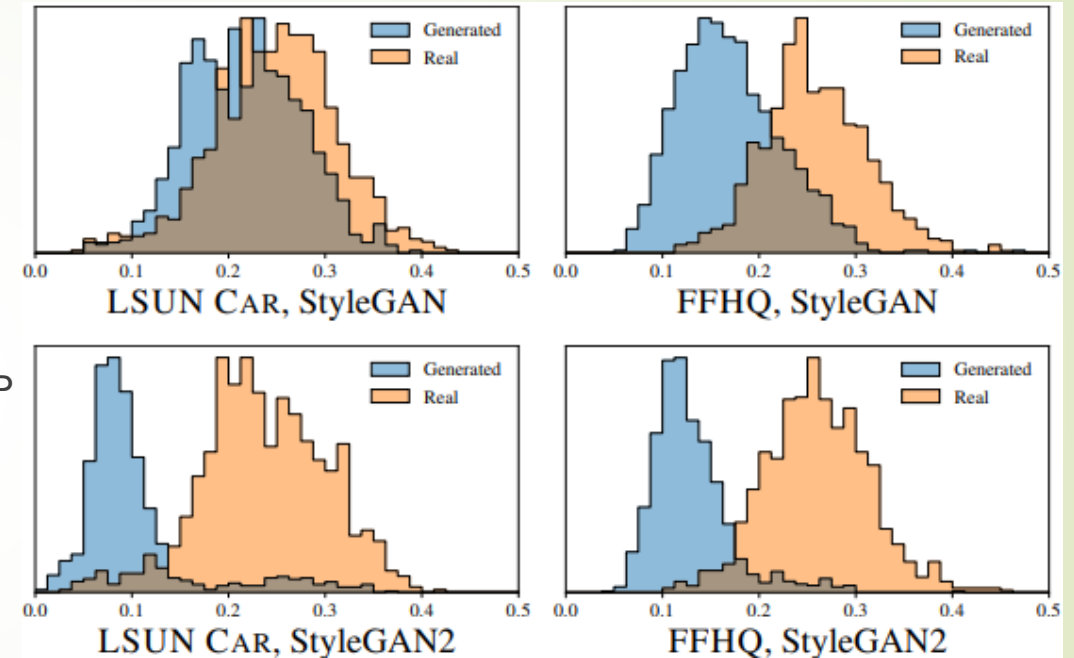
Style GAN 2

Projection of images to latent space

Помимо отображения генератором латентного пространства в картинки хочется также решать обратную задачу – по картинке создаваться для неё вектор стилей.

Предложенный в статье метод имеет 2 основных отличия от предыдущих. Во-первых, к скрытому коду добавляется шум, чтобы более полно охватить латентное пространство. Во-вторых, обучается не только латентный код, но также и уровень шумов генератора.

При этом StyleGAN 2 позволяет гораздо лучше определять, настоящее ли изображение или сгенерированное нейросетью. Распределения изображений по метрике LIPS изображены на рисунке.

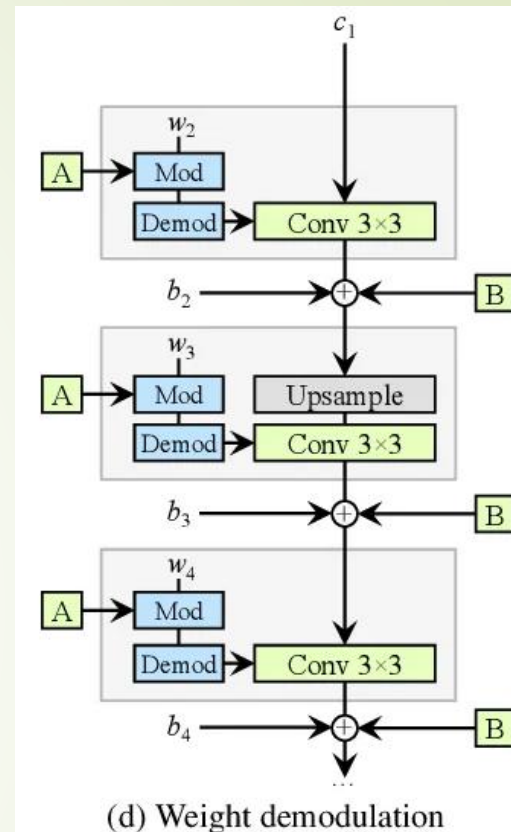


Style GAN 2

Как получить вектор стилей по картинке?

Предполагаем, что у нас есть обученная Style GAN архитектура и изображение, у которого хотим получить вектор стилей. Замораживаем все веса модели, делаем обучаемым вектор w (из латентного пространства) и дисперсию в зашумлениях (B на схеме).

На каждом шаге обучения генерируем из w изображение, получаем с него эмбединги и сравниваем их с эмбедингами исходного изображения. На основании этого делаем backprop.



Style GAN 2

ИТОГИ

- За бейзлан взят Style GAN, в (b) добавлена новая архитектура генератора с демодуляцией, в (c) регуляризация работает только раз в несколько батчей, в (d) регуляризация изменена на предложенную PLR, в (e) вместо progressive growing используются skip connections, в (f) у последнего разрешения удвоено количество признаков для каждого пикселя.
- При добавлении PLR (d) произошёл обмен PPL и FID.
- При добавлении дополнительных признаков (f) произошёл сильный скачок для FID и Recall
- При сравнении отдельных категорий LSUN Style-GAN 2 показывает себя сильно лучше Style GAN

Configuration	FFHQ, 1024×1024				LSUN Car, 512×384			
	FID ↓	Path length ↓	Precision ↑	Recall ↑	FID ↓	Path length ↓	Precision ↑	Recall ↑
A Baseline StyleGAN [24]	4.40	212.1	0.721	0.399	3.27	1484.5	0.701	0.435
B + Weight demodulation	4.39	175.4	0.702	0.425	3.04	862.4	0.685	0.488
C + Lazy regularization	4.38	158.0	0.719	0.427	2.83	981.6	0.688	0.493
D + Path length regularization	4.34	122.5	0.715	0.418	3.43	651.2	0.697	0.452
E + No growing, new G & D arch.	3.31	124.5	0.705	0.449	3.19	471.2	0.690	0.454
F + Large networks (StyleGAN2)	2.84	145.0	0.689	0.492	2.32	415.5	0.678	0.514
Config A with large networks	3.98	199.2	0.716	0.422	—	—	—	—

Dataset	Resolution	StyleGAN (A)		StyleGAN2 (F)	
		FID	PPL	FID	PPL
LSUN CAR	512×384	3.27	1485	2.32	416
LSUN CAT	256×256	8.53	924	6.93	439
LSUN CHURCH	256×256	4.21	742	3.86	342
LSUN HORSE	256×256	3.83	1405	3.43	338



А что дальше?

- Style GAN 2 ADA – аугментации, позволяющие улучшить качество и увеличить скорость обучения
- Style GAN 3 – совсем новая статья, решающая проблему алиасинга, картинки ещё лучше, движение в сторону генерации гиф\видео



Литература

- Style GAN: <https://arxiv.org/pdf/1812.04948.pdf>
- Style GAN 2: <https://arxiv.org/pdf/1912.04958.pdf>
- Style GAN 2 ADA: <https://arxiv.org/pdf/2006.06676.pdf>
- Style GAN 3: <https://arxiv.org/pdf/2106.12423.pdf>