

Отчет по дз-2 DGM (StarGan + улучш)

Kirill Gelvan

Мой топ багов этой домашки:

- Сразу несколько цветов волос генерил
- Делал степ оптимайзера по переменной "шаг", а не "эпоха"
- Перевел модель в eval навсегда

Общее

- Железка: в основном 2080 TI
- BS=32, картинки 128x128 (для ориг статьи BS=16)
- Использовал 5 атрибутов как в статье: 3 цвета волос, Young, Male.

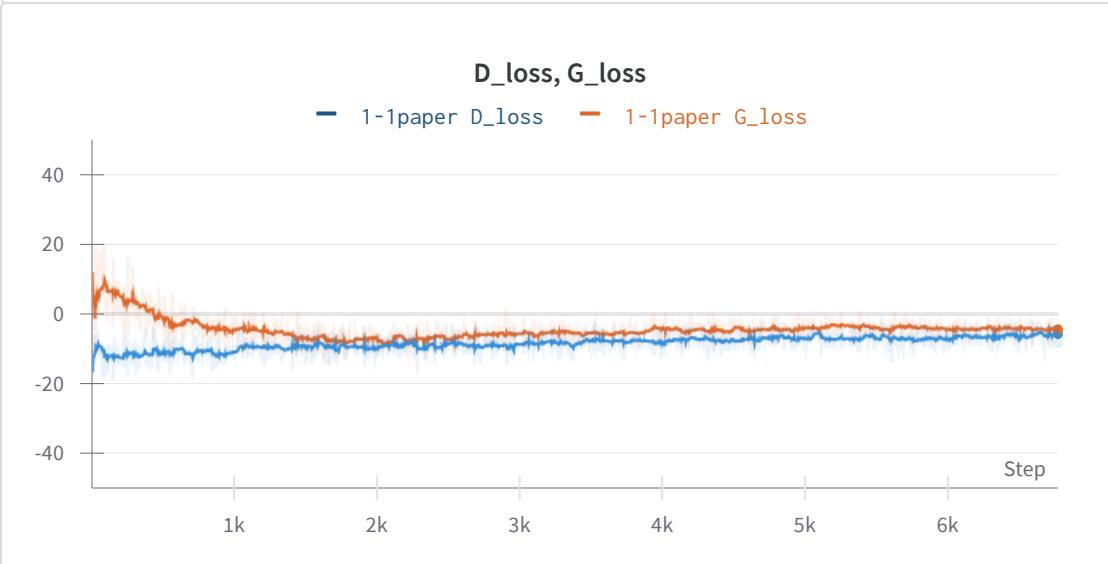
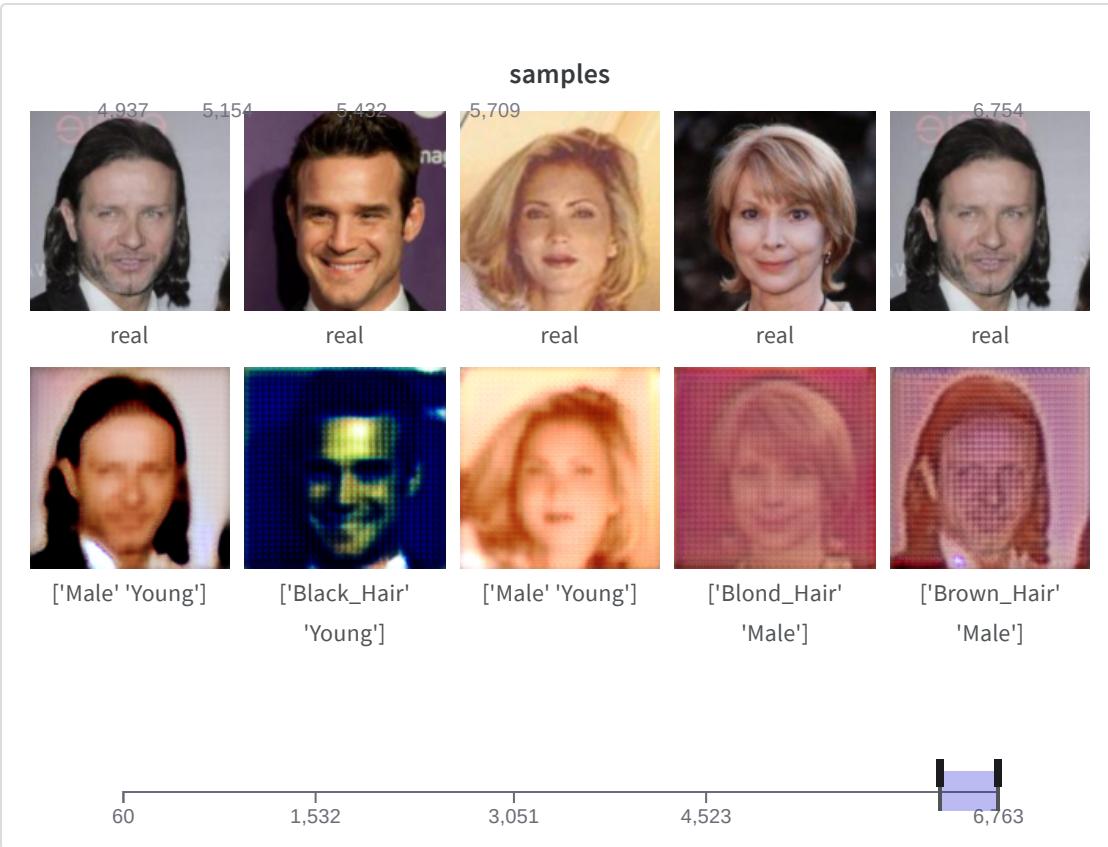
1) Оригинальная статья

Короче я делал все ровно как в статье. Взял 5 атрибутов: 3 цвета волос, Young, Male. Внизу сгенерированных картинок подпись. Если ее нет, это значит Female + Old. Если нет Male, то там Female и тд.

Единственное отличие это то, что я рандомно генерировал метки, а не шаффлил их.

Для реализации статьи я подчерпнул из оригинального кода следующие моменты:

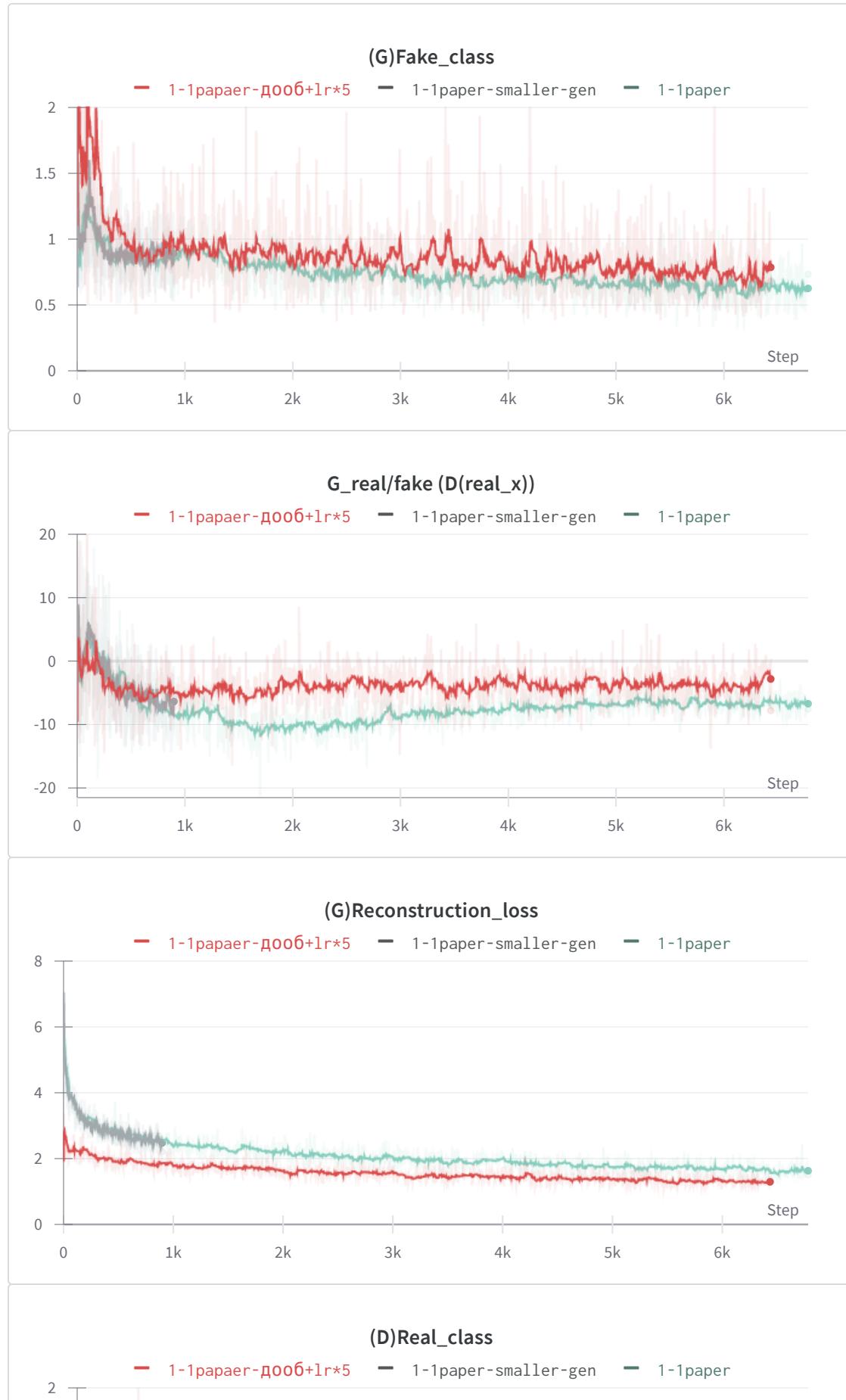
- 0) Порядок обучения генератора и дискриминатора
- 1) bias=False почти во всех свертках
- 2) Res block оказался из CycleGAN, а именно: Conv, IN, ReLU, Conv, IN (хотя из статьи могло показаться Conv, IN, ReLU)

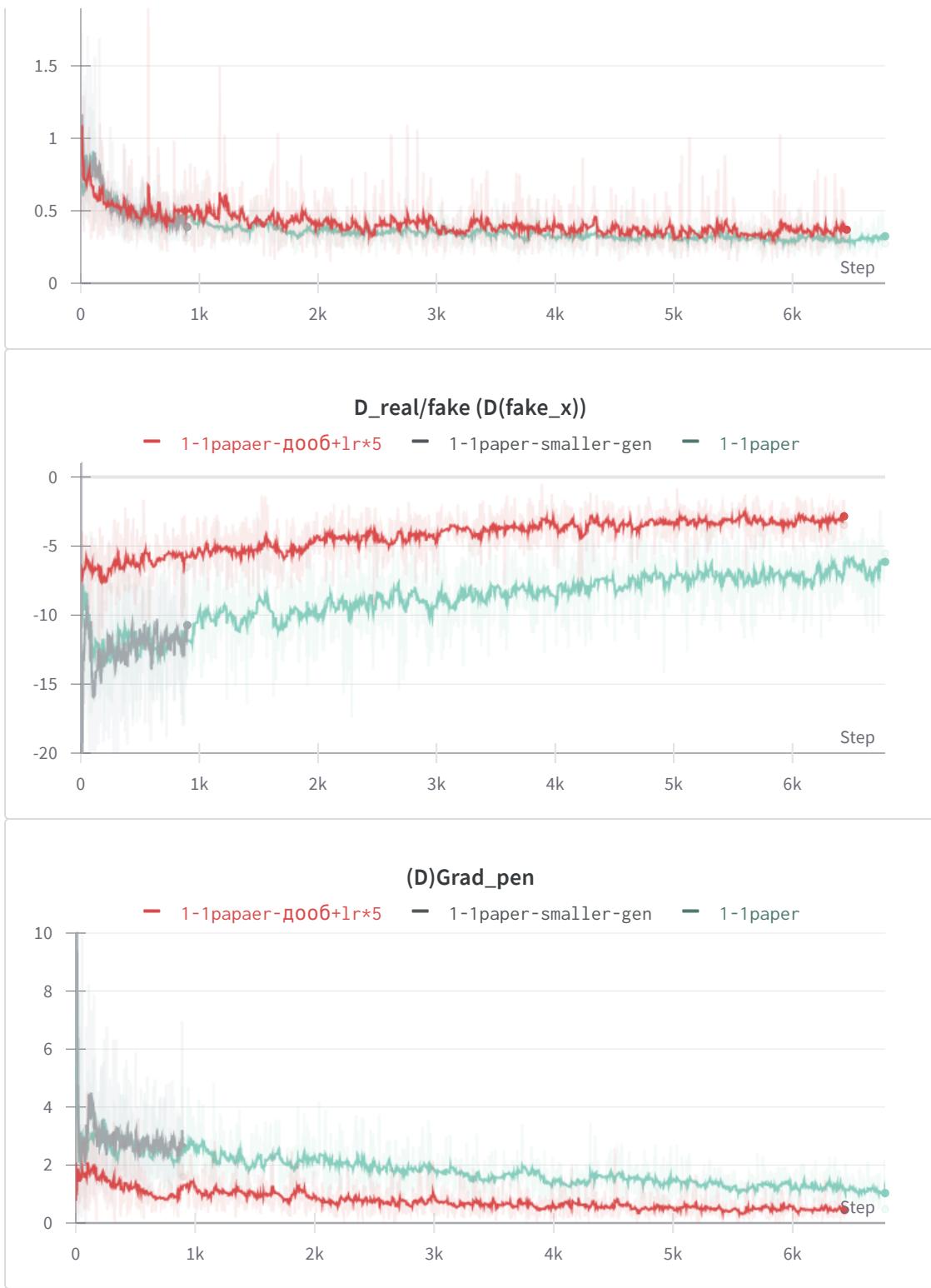


Как можно заметить картинки генерятся очень плохие. Сетка учились пару эпох. Как потом выяснилось, картинки настолько плохие потому, что из-за IN надо генерировать картинки в режиме `model.train()`. Также у сетки все еще был большой Reconstruction loss, то есть картинки в целом были не похожи.

Я пробовал учить Res Block как в статье написано, крутить бачсайз, учить дольше и многое другое. Возможно, что если учить сетку

реально 20 эпох как говорят авторы, она будет генерировать лучше картинки:





В общем тут все метрики менялись очень-очень медленно.

2) Про FID

Далее я реализовал подсчет FID - взял из прошлой домашки код.

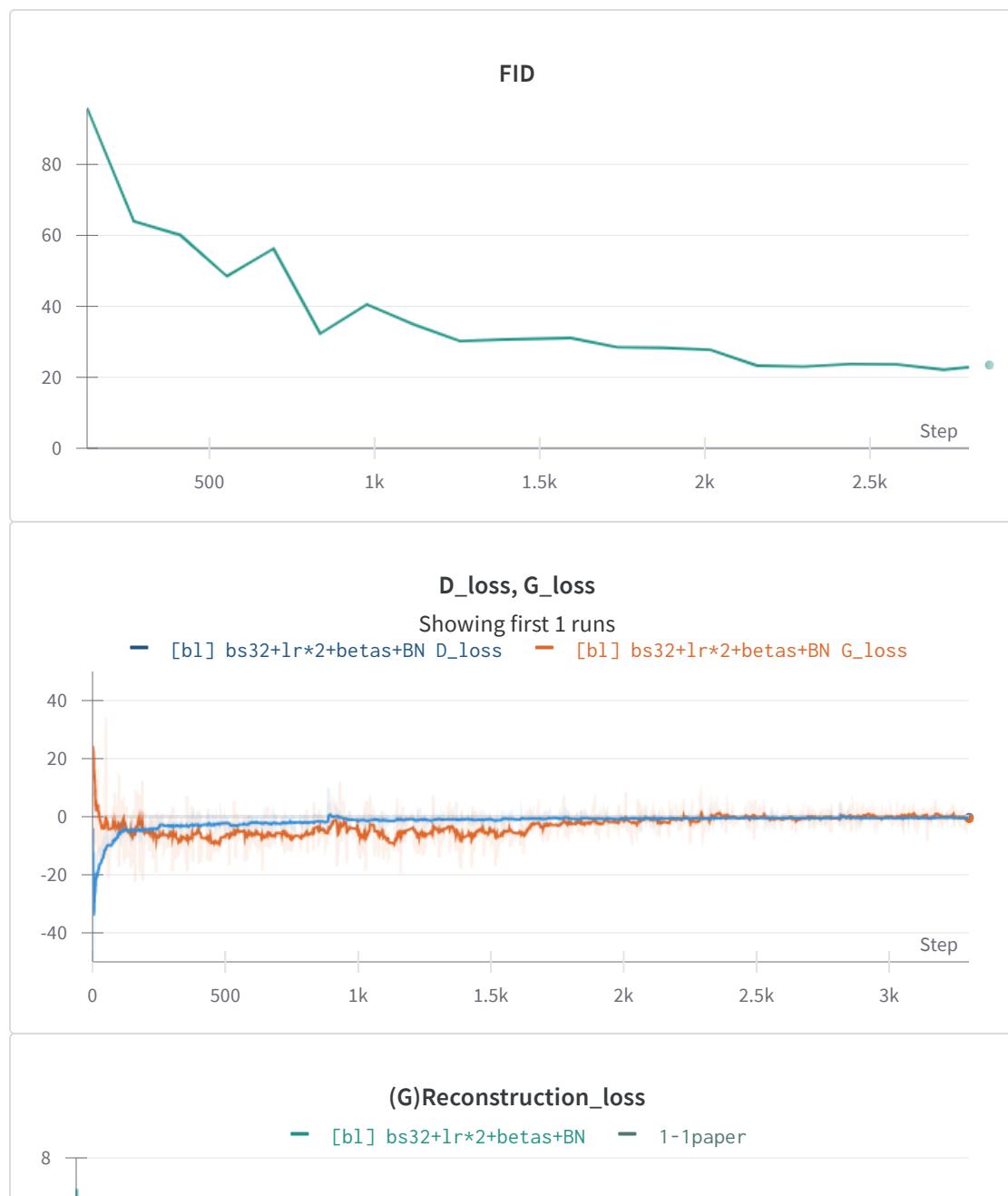
Реализацию подгрузки InceptionV3 (с весами из PyTorch) я стырил с

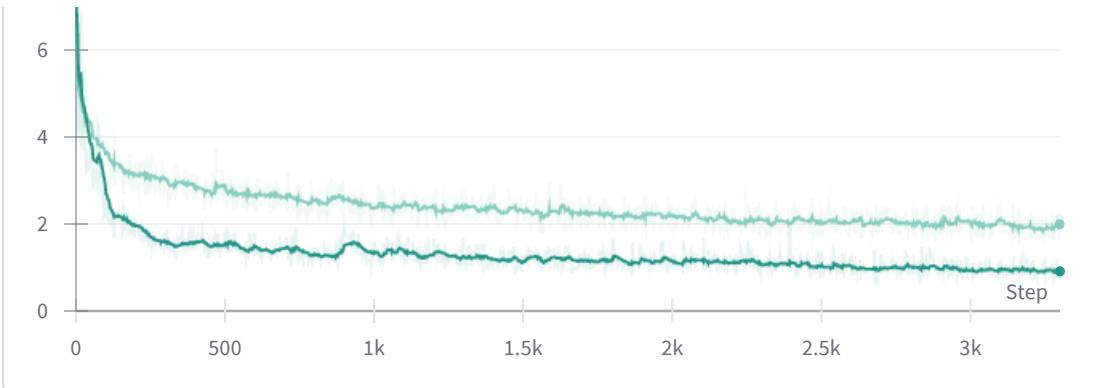
гитхаба (https://github.com/mseitzer/pytorch-fid/blob/master/src/pytorch_fid/inception.py)

3) Далее об экспериментах

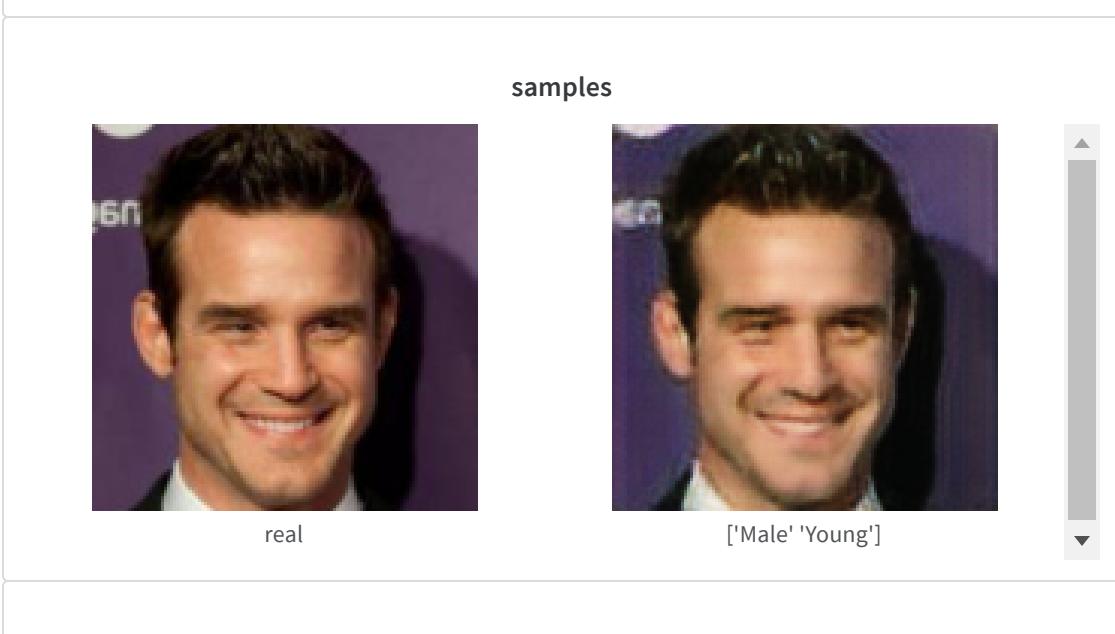
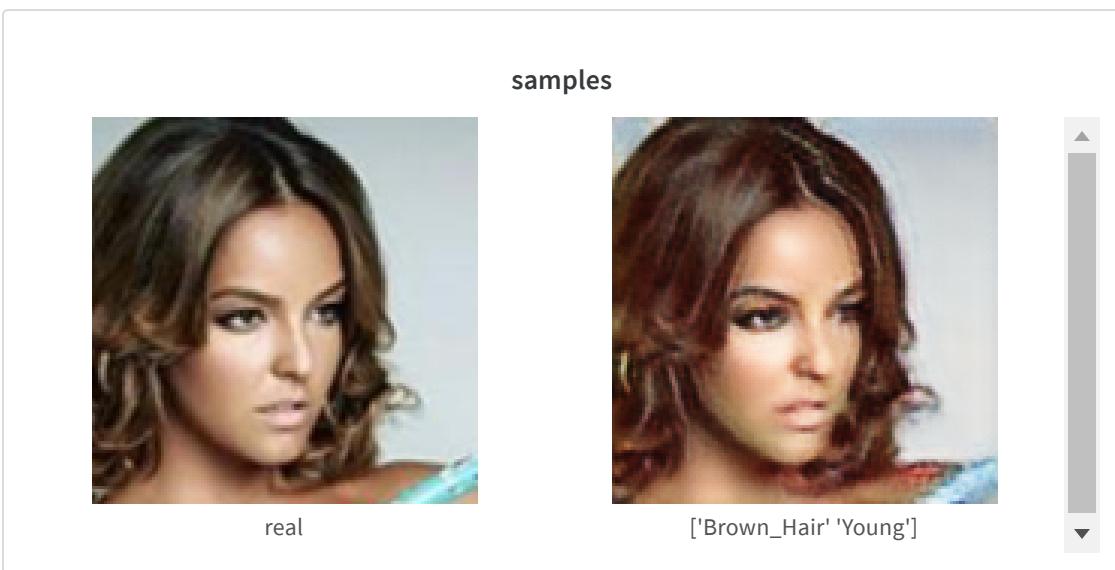
Я считаю, что у меня получилось предложить улучшения, которые привели к лучшему качеству и более быстрому обучению. Далее я их опишу

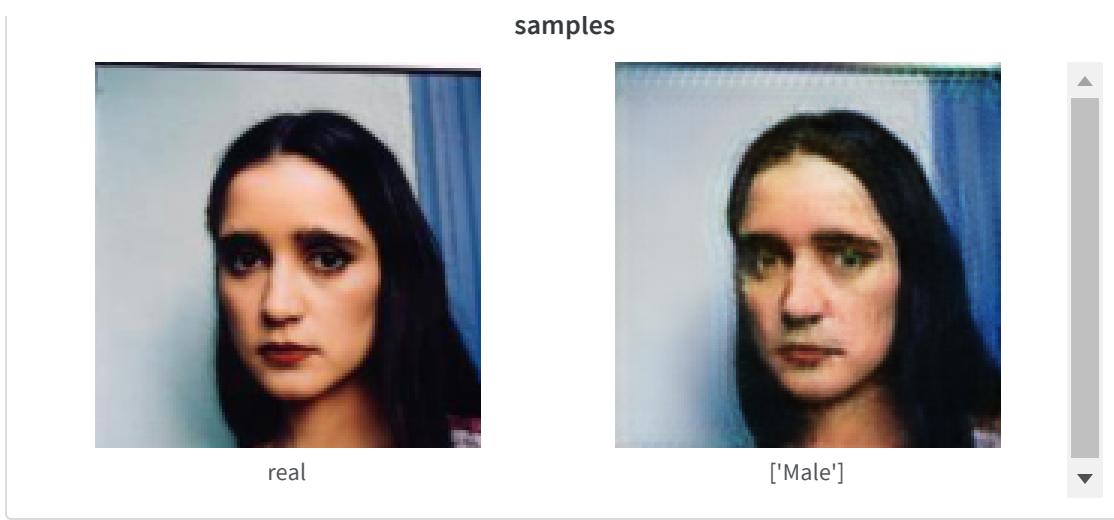
1) Заменяем IN на BN. Поскольку в оригинальной статье нормы только в генераторе, то мы можем их заменить (в критике использовать BN как мы знаем нельзя). Также я вместил на карту BS=32, взял LR в 2 раза больше и все это назвал базовый ран:





На графиках выше видно, что здесь Reconstruction Loss сразу ушел ниже, что дало более красивые картинки. Также видно, что значения общих лоссов оба около 0, а для ориг статьи были около -5. Я еще привел FID (но только на новом бейзлайне). Он порядка 20-21. Качественные примеры:

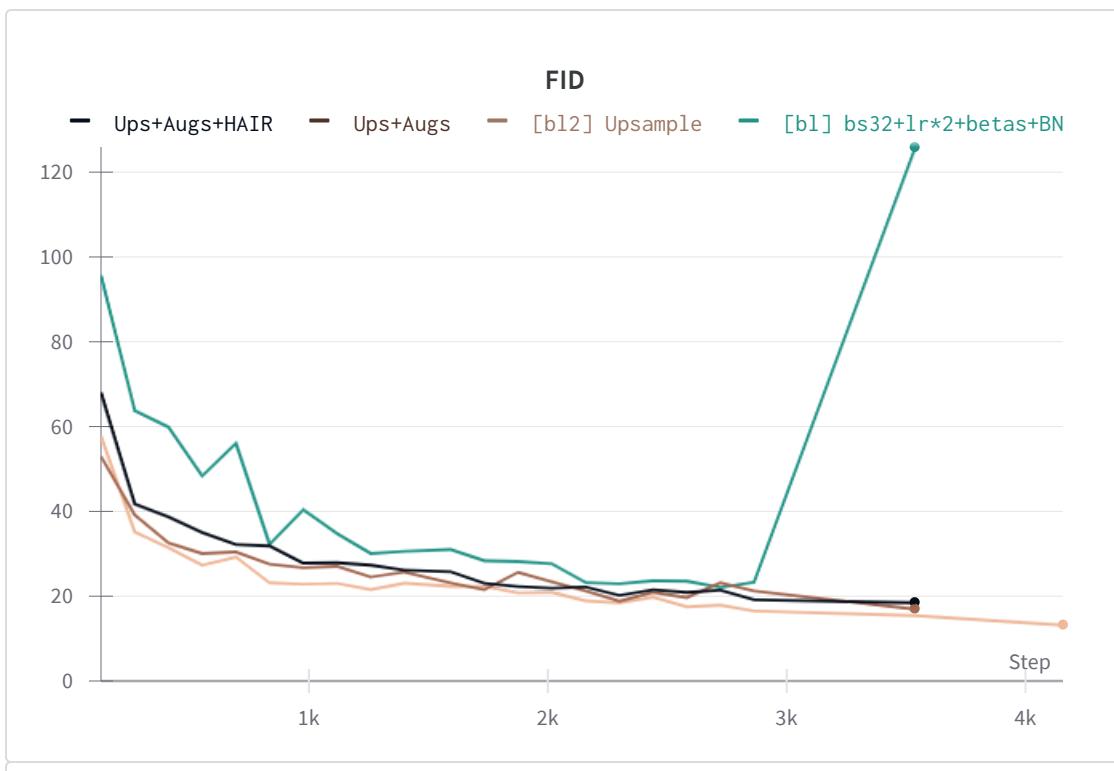




Далее посмотрим на следующие изменения:

- 1) Замена ConvTranspose на Upsample + Transpose
- 2) 1) + Augmentations (много)
- 3) 1) + Augmentations (мало)

Рисую ниже FID и картинки (по этим картинкам мало что можно понять, но в целом посмотреть на случайные сэмплы из каждой модели)



samples

2,872
Ups+Augs+HAIR

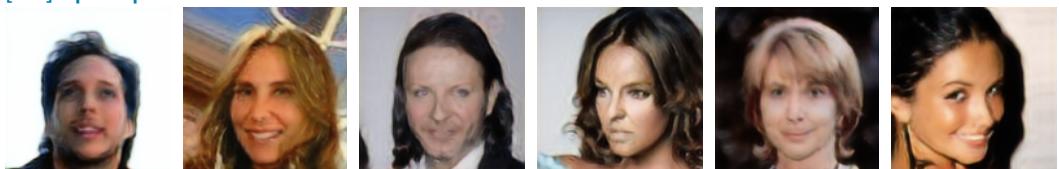


3,228

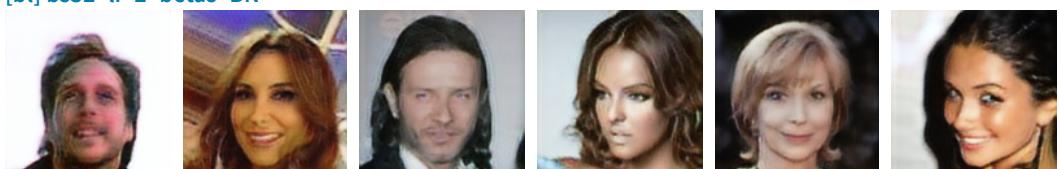
Ups+Augs



[bl2] Upsample



[bl] bs32+lr*2+betas+BN



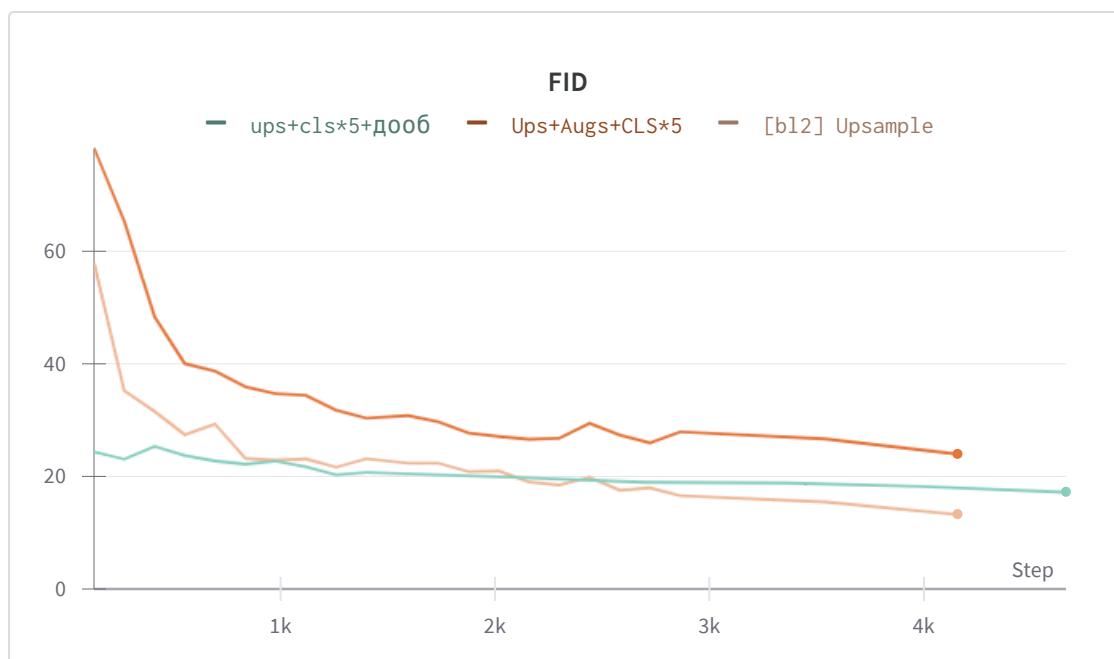
Видно по графику, что модели с аугментациями проигрывают. Стоит отметить, что я FID считаю тоже на аугментированных картинках. Возможно, если подольше поучить получатся лучше картинки для моделей с аугментациям.

И тут мне рассказали, что в коде авторы усредняли CLS loss по другому. То есть он усреднялся только по батчу, а по кол-ву атрибутов не усреднялся. По итогу это эквивалентно тому, что я всегда брал 0.2 CLS лоссов. Я помножил эти лосы на 5 и обучил несколько моделей по-быстрому т.к. скоро дедлайн. И как можно было догадаться модели стали лучше менять атрибуты (более явно) и хуже рисовать картинки (более размыто). Вот наглядные примеры (на примере видно, как в первой строке у мужчины очень белые волосы, чего мы раньше не видели в примерах; однако картинки сильно более размыты - чего и следовало ожидать).

Далее я покажу результаты для 3 моделей:

Каждая из них как я выяснил в экспериментах выше имеет оригинальную архитектуру StarGan, но имеет BN вместо IN, BS=32, LR=0.0002, Upsample+Conv вместо ConvTranspose.

- 1) Baseline (описана выше)
- 2) Baseline & CLS_LOSS * 5
- 3) Baseline & CLS_LOSS * 5 & Augmentations



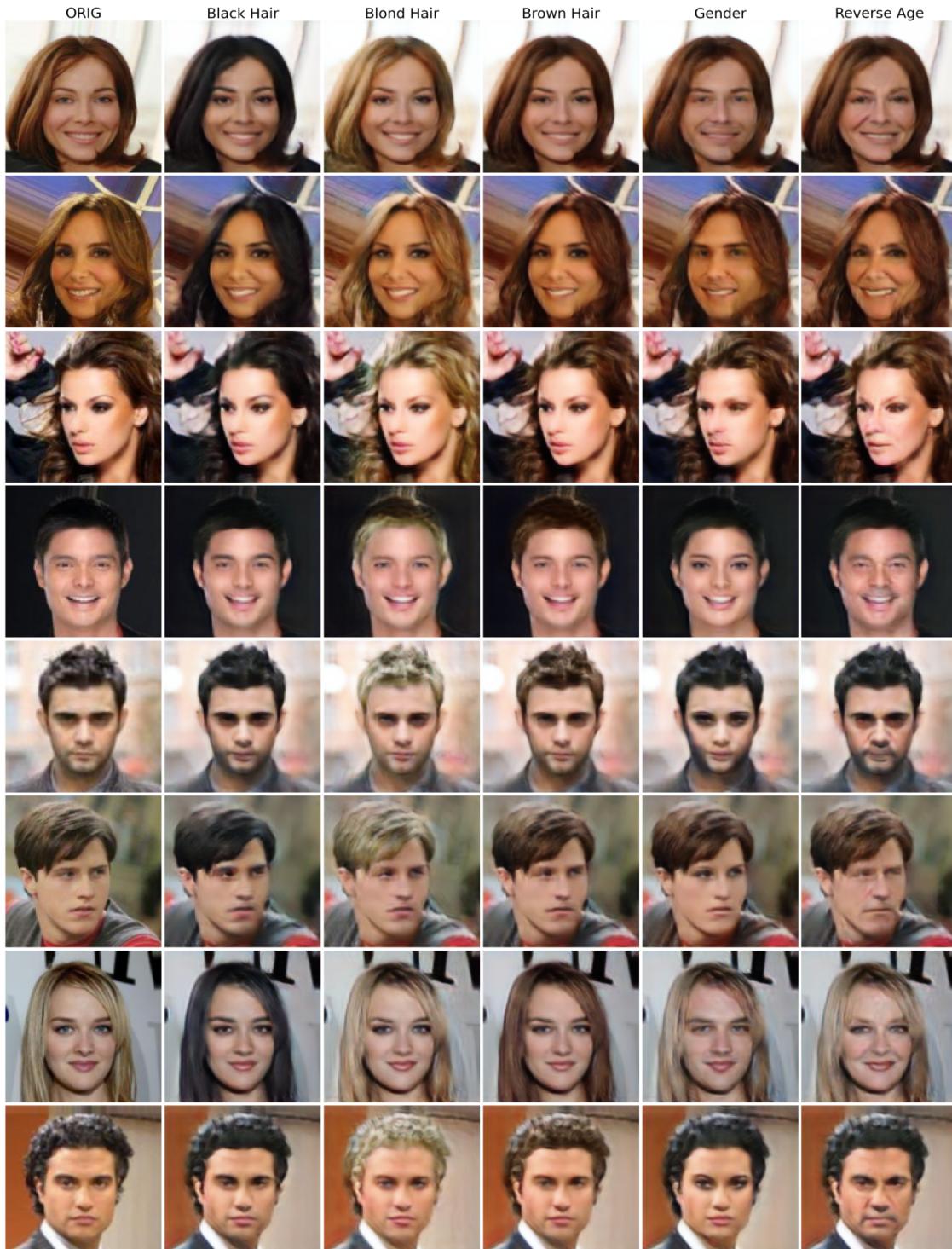


На картинках выше отчетливо видно, что нижний ряд (без перевеса CLS лосса) картинки более четкие, но изменения почти незаметны. На верхнем ряду (с перевесом) картинки менее четкие, но изменения более качественные и лучше видны. Это подтверждается и метрикой FID: у нижнего рана она меньше (т.к. не учитывает сами изменения).

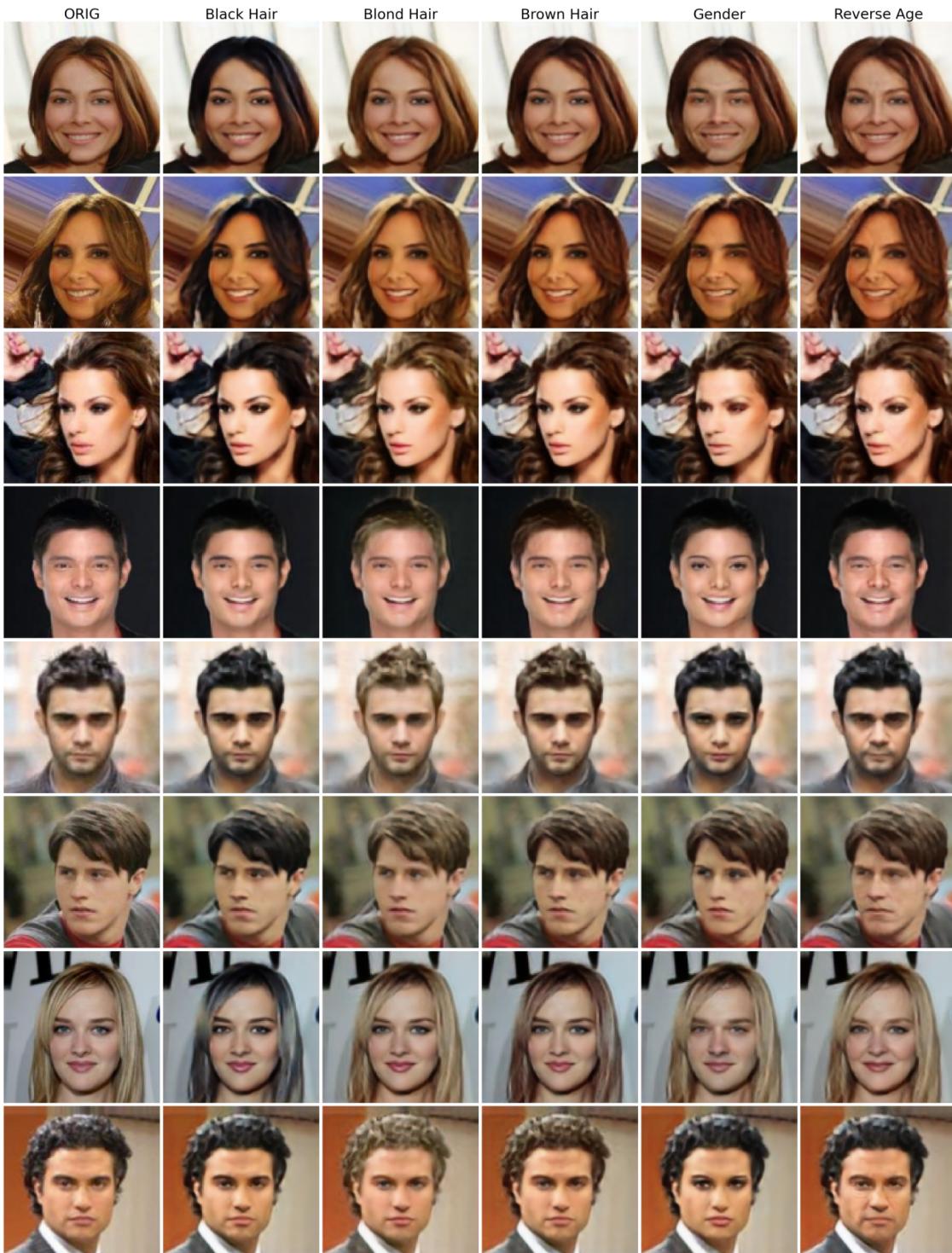
Во втором же ряду картинки очень сильно размыты из-за аугментаций. Вероятно, надо сильно дольше учить такую модель.

Финальные cherry pick'и

Из моделей без аугов я начеррипикал ниже. 1 это модель с CLS_LOSS * 5, а вторая без перевзвеса. Опять же, разница видна - вторая модель модее консервативна и получает лучше качество и FID, но хуже делает преобразования. Я бы выбрал первую (но надо доучить)



CLS LOSS * 5



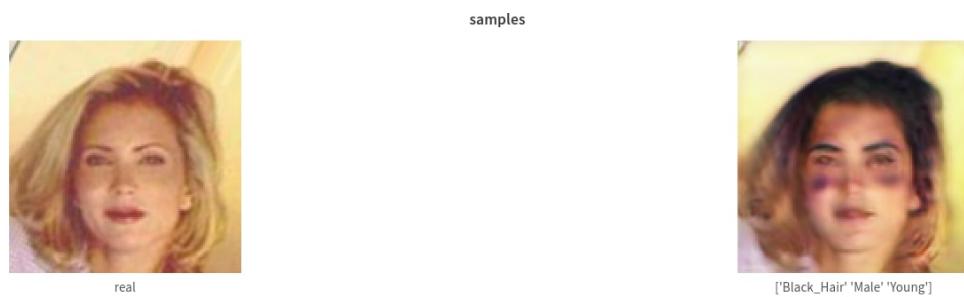
BASE (CLS_LOSS * 1)

Итог: Получилась работа по отлавливанию багов и непонятных описаний авторов в статье. Однако попробовал несколько изменений типа Upsample, BN еще пробовал учить чисто архитектуру авторов и разные BS, LR, добавлять аугментации. И получились неплохие картинки!

Настоящий итог: куча багов и рейджа; интересные вещи типа IN в критике и спектральной нормализации попробовать не успел =(

ненавижу DL

Бонус!!!! Хорошие/смешные примеры!



Брови перекрасились в черный. Считаю прикольно



У этого парня будто косметика!



Created with ❤️ on Weights & Biases.

<https://wandb.ai/kirili4ik/dgm-ht2/reports/-2-DGM-StarGan--Vmlldzo1MjE1NTk>