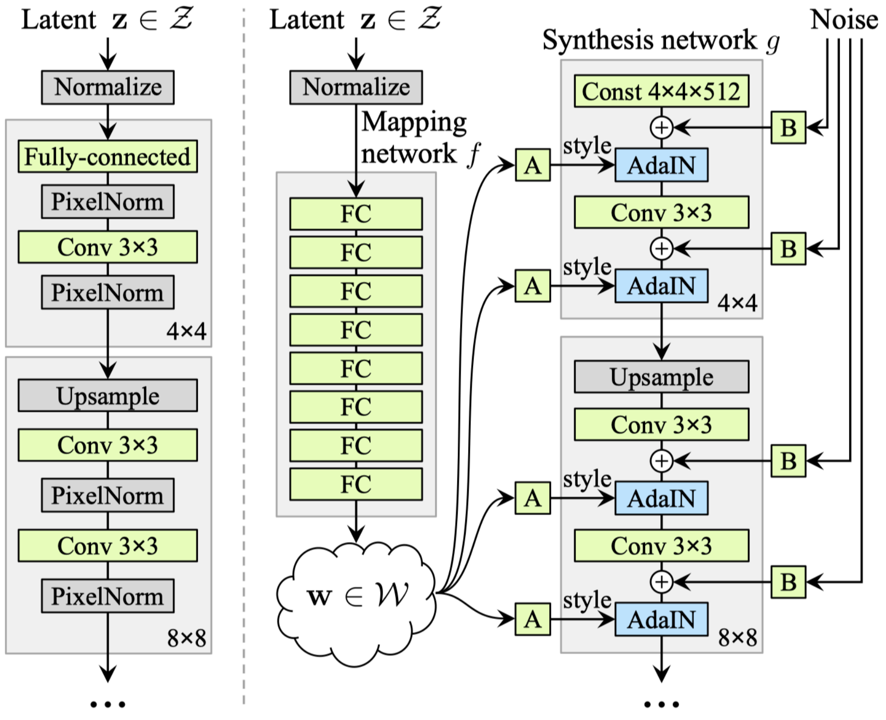
Analyzing and Improving the Image Quality of StyleGAN

Karras et al, 2020

NVIDIA



(a) Traditional (b) Style-based generator

StyleGAN "v1"

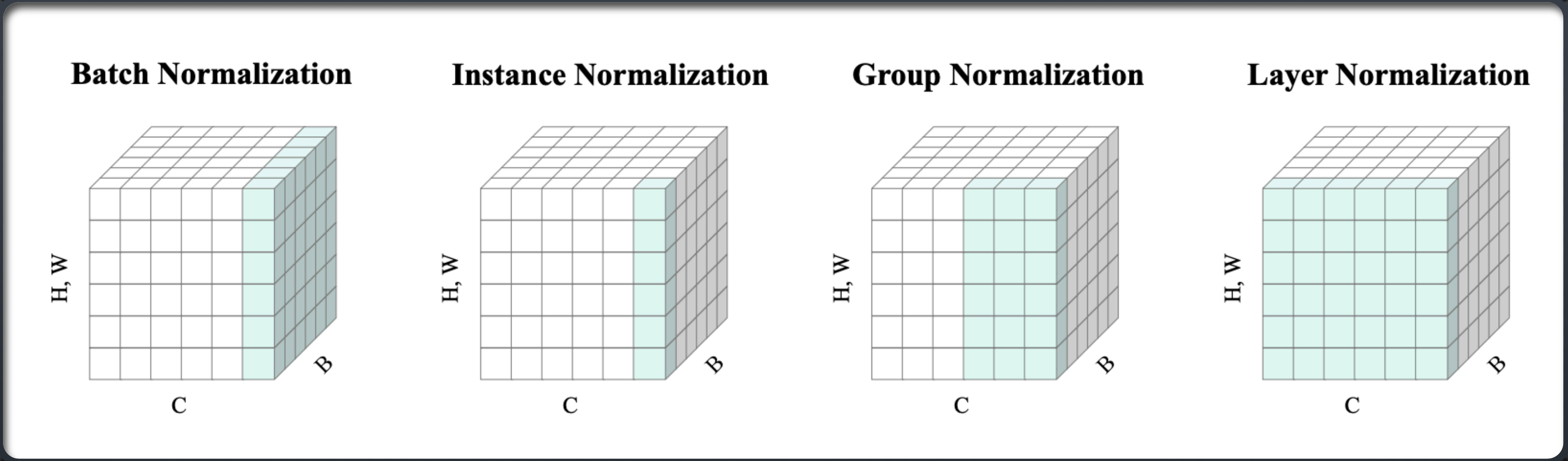
- Modular architecture that injects noise vector at each stage of generation
- Can generate high-quality images
- Feature disentanglement enables controllable generation

Context

- Empirically motivated project
- "Bag-of-tricks"
- Today's presentation focuses on Architectural Changes

“Droplet” artifacts in the generated images

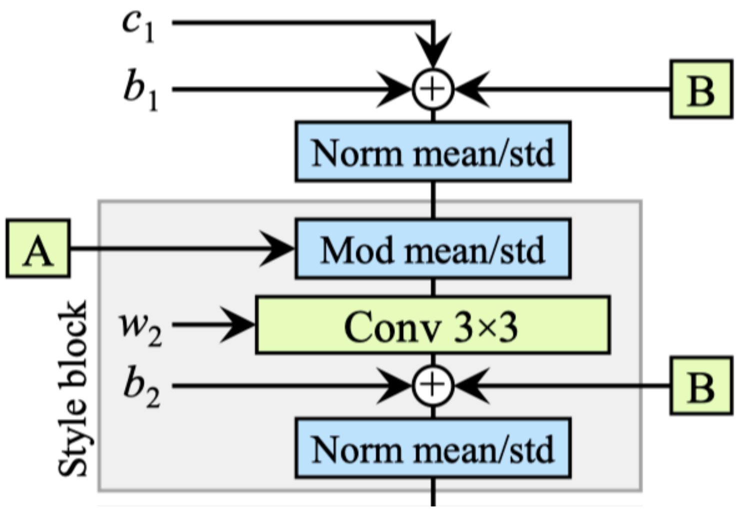
StyleGAN v2 proposes architectural changes to remove such artifacts



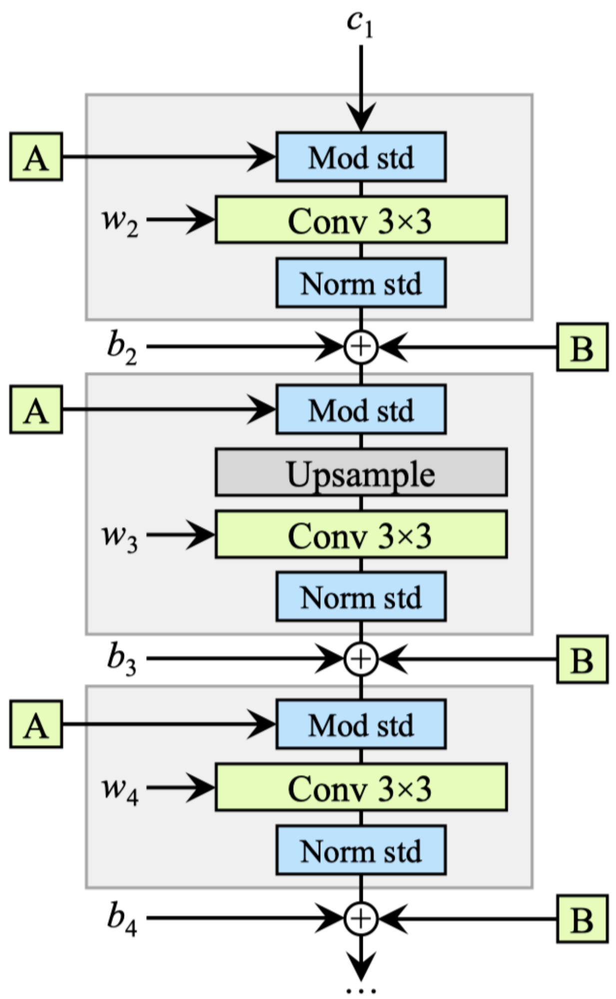
Visualizing instance normalization

(https://proceedings.neurips.cc/paper/2019/hash/6d0f846348a856321729a2f36734d1a7-Abstract.html)

$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}$$



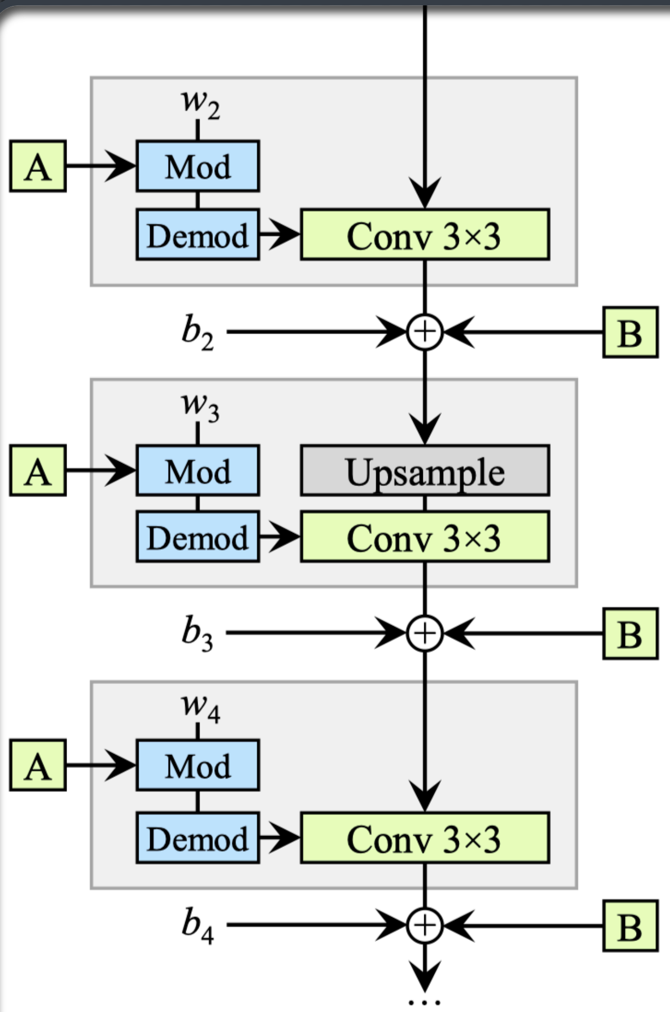
Diving into AdaIN: normalization removes magnitude information



(c) Revised architecture

CHANGING GENERATOR

- Move noise vectors after normalization
- Only normalize and modulate over stdev
- Remove bias, noise, normalization on the input

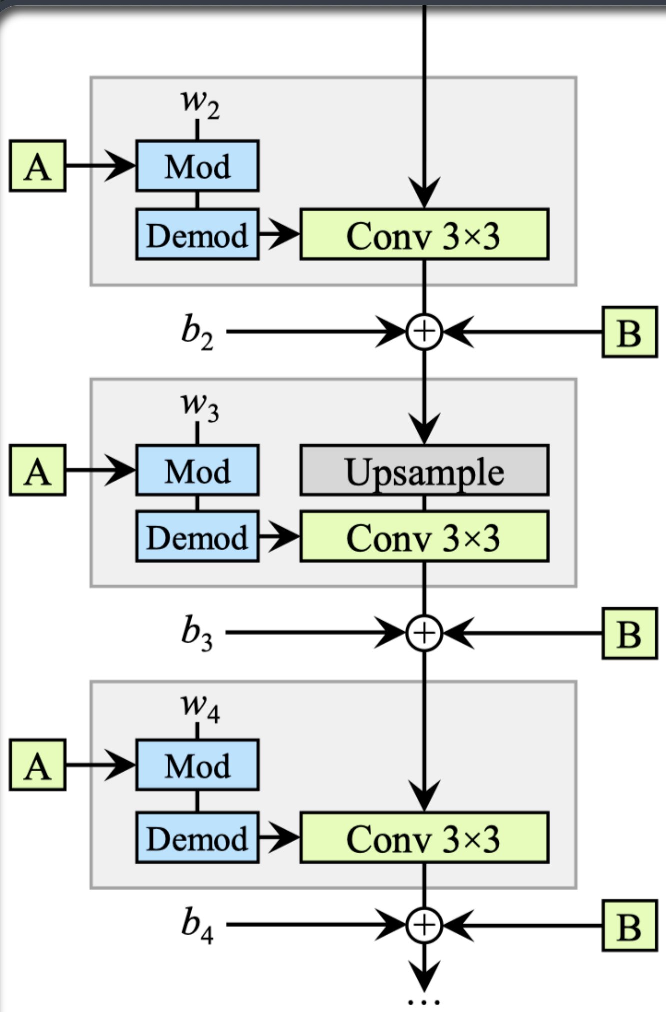


(d) Weight demodulation

Rewiring instance modulation

- SCALING CAN BE REWRITE INTO CONVOLUTION WEIGHTS

$$w'_{ijk} = s_i \cdot w_{ijk}, \tag{1}$$

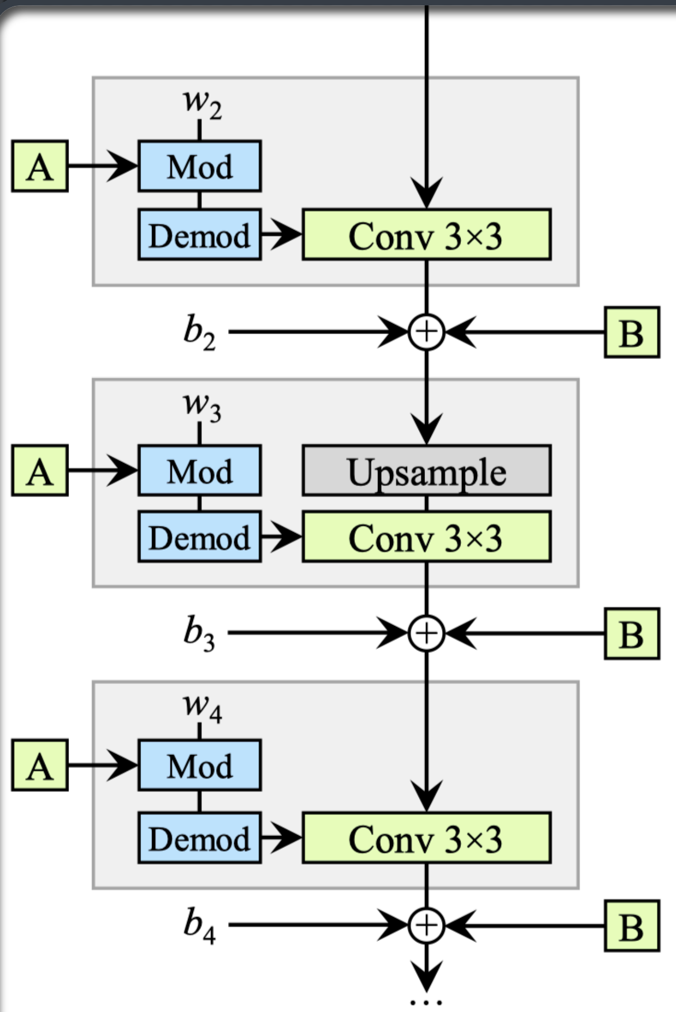


(d) Weight demodulation

Data-independent Normalization

- Assuming activations are i.i.d zero-mean unit-variance, the expected per-feature map stdev can be written as

$$\sigma_j = \sqrt{\sum_{i,k} {w'_{ijk}}^2}, \tag{2}$$



(d) Weight demodulation

Input-independent Normalization

- Assuming activations are i.i.d zero-mean unit-variance, the expected per-feature map stdev can be written as

$$\sigma_j = \sqrt{\sum_{i,k} {w'_{ijk}}^2}, \quad (2)$$

$$w''_{ijk} = w'_{ijk} \Big/ \sqrt{\sum_{i,k} {w'_{ijk}}^2 + \epsilon}, \quad (3)$$

WEIGHT DEMODULATION REMOVES DROPLET ARTIFACTS

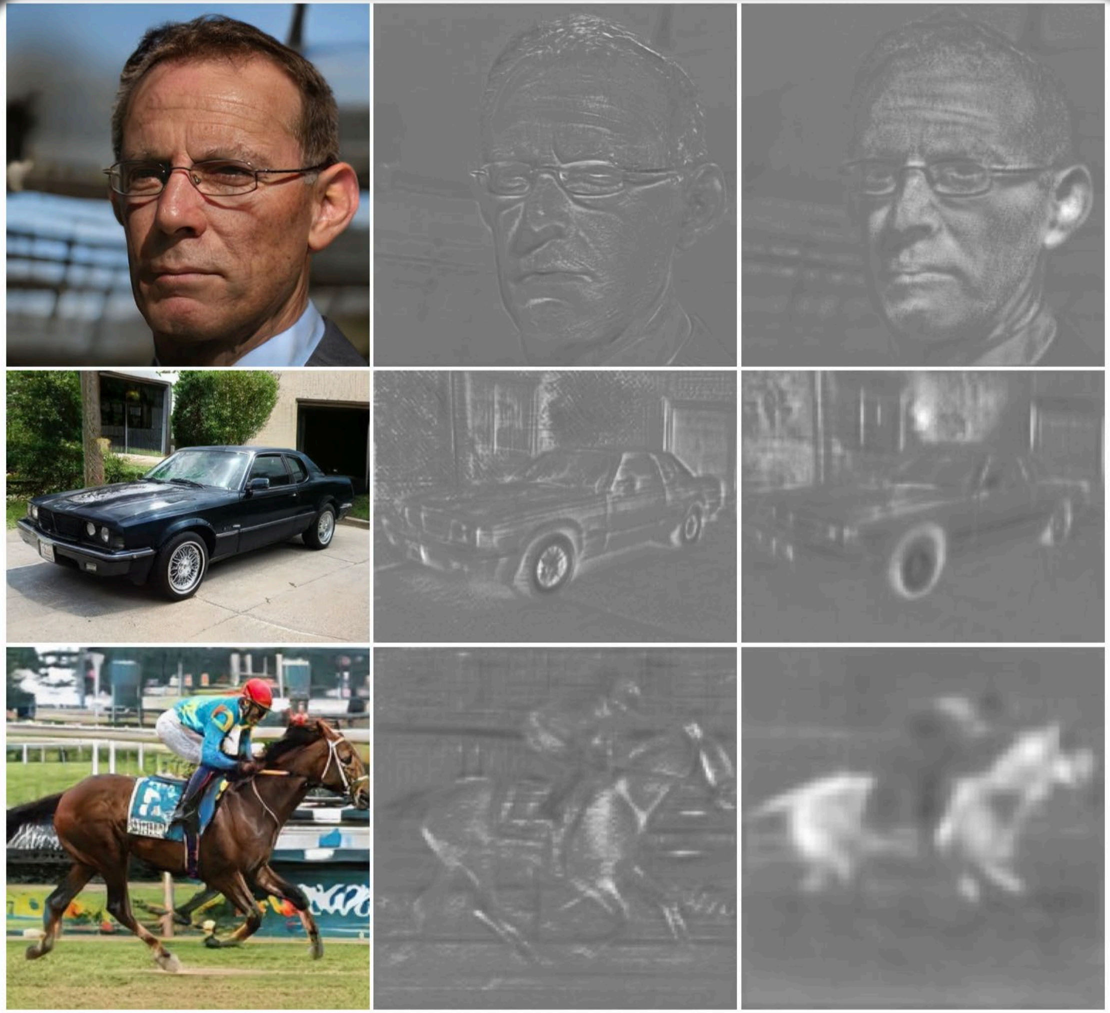


Figure 3. Replacing normalization with demodulation removes the characteristic artifacts from images and activations.

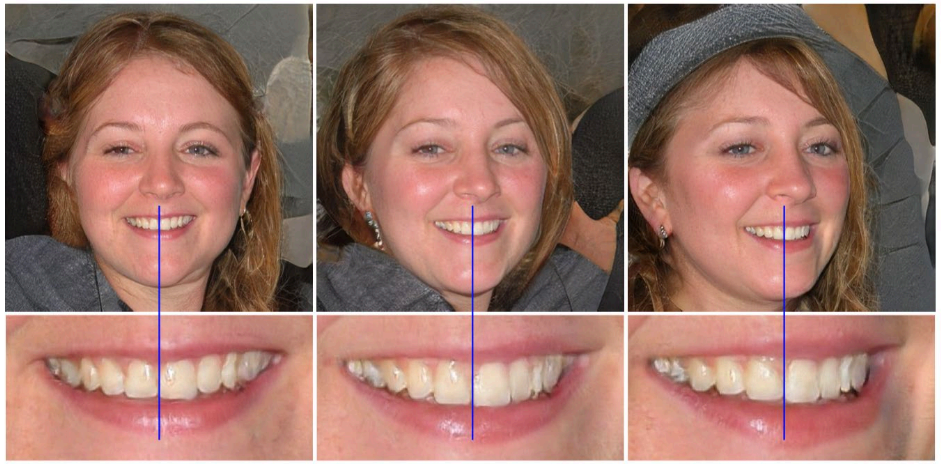
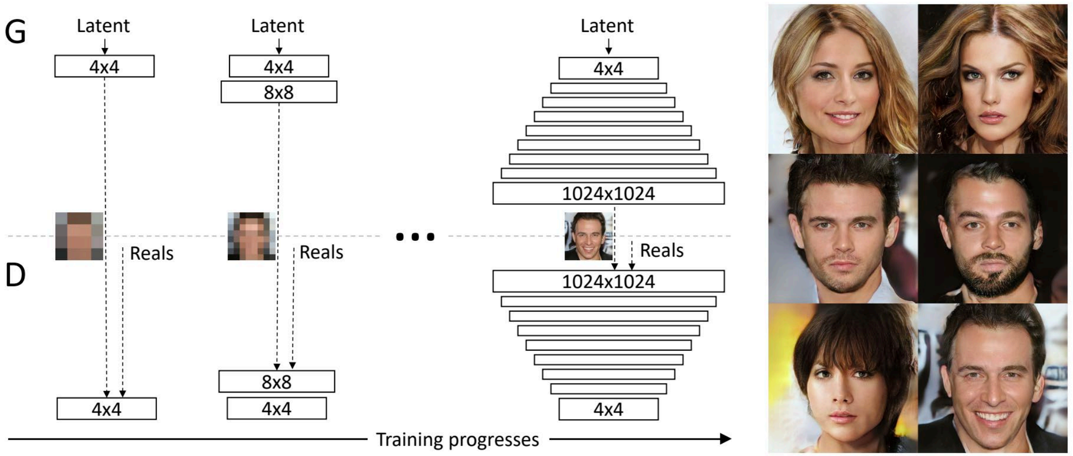
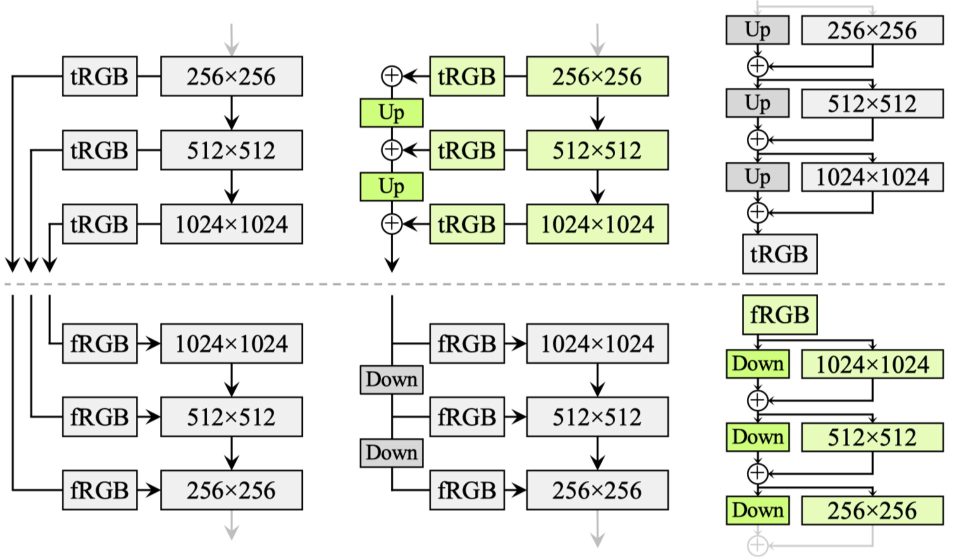


Figure 6. Progressive growing leads to "phase" artifacts. In this example the teeth do not follow the pose but stay aligned to the camera, as indicated by the blue line.



PROGRESSIVE GROWING CAUSES "PHASE" ARTIFACTS



(a) MSG-GAN (b) Input/output skips (c) Residual nets

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.

SKIP-GENERATOR AND RESIDUAL DISCRIMINATOR IMPROVE RESULTS

MORE IN PAPER & APPENDIX

- Generator Smoothness -> Image quality -> better regularization
- Effect of scaling up
- A new latent code projection method