GANimation

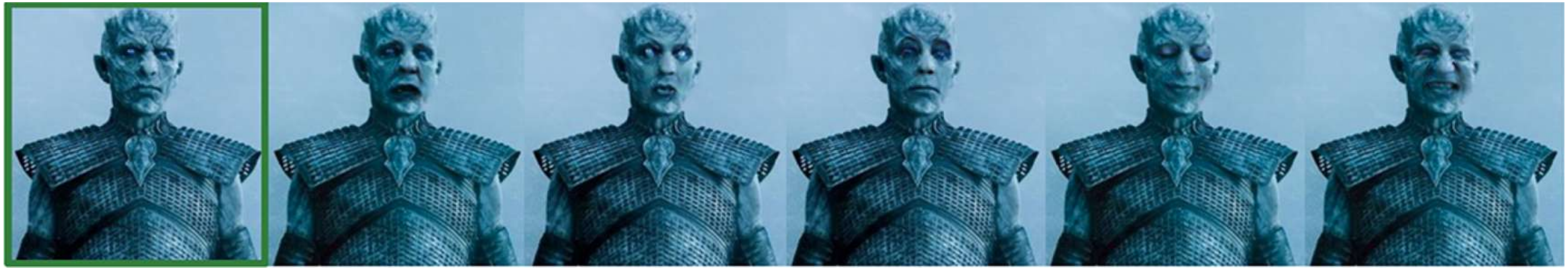
Anatomically-Aware Facial Animation from a Single Image

Presented by Owen Jow

Original authors: Albert Pumarola, Antonio Agudo,
Aleix M. Martinez, Alberto Sanfeliu, Francesc Moreno-Noguer

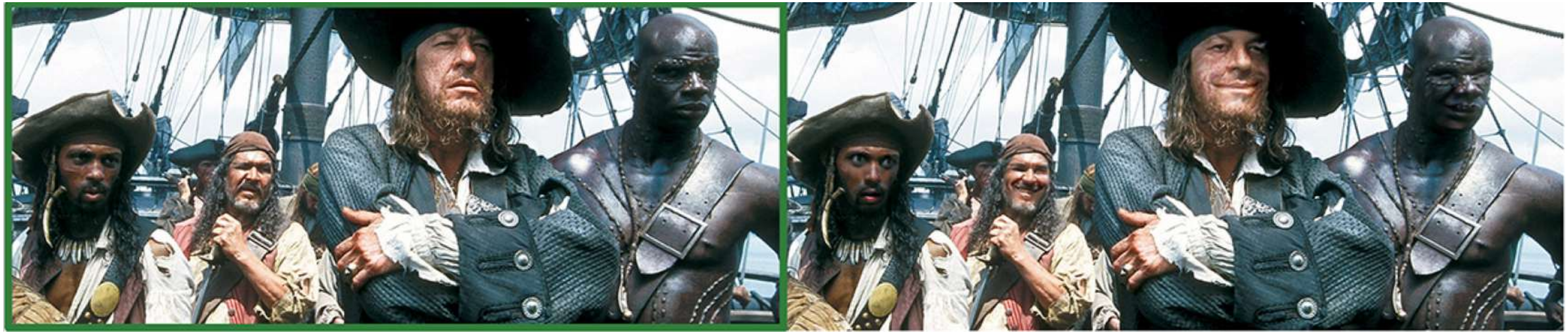
Overview

- Given
  - An input face *(as an image)*
  - An expression *(as a vector of action unit magnitudes)*
- Synthesize the input face as it would appear with the given expression



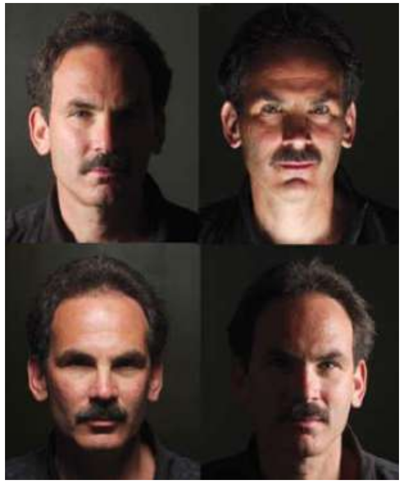
Motivation

- Applications in **image/video editing**
- Previous methods could only modify a **discrete** set of attributes
- Previous outputs were visibly blurrier than corresponding inputs



Challenges

- **Distractors**: occlusion, illumination, variety in facial structure, blurriness, many possible orientations (looking to the right? left?)…
- No datasets containing lots of the **same** faces **“in the wild”** and with many different **continuously-varying** expressions



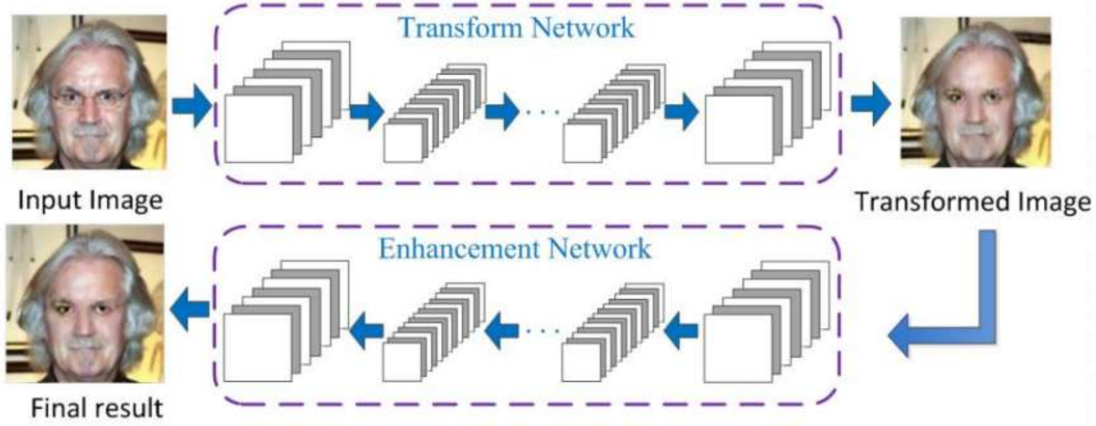
make this man smile, please…

Recent Progress

- **DIAT** (2016) **D**eep **I**dentity-**A**ware **T**ransfer of Facial Attributes
- **IcGAN** (2016) **I**nvertible **C**onditional **GAN**s for Image Editing
- **CycleGAN** (2017) Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks
- **StarGAN** (2018) Unified GANs for Multi-Domain Image-to-Image Translation

DIAT (2016)

Deep Identity-Aware Transfer of Facial Attributes

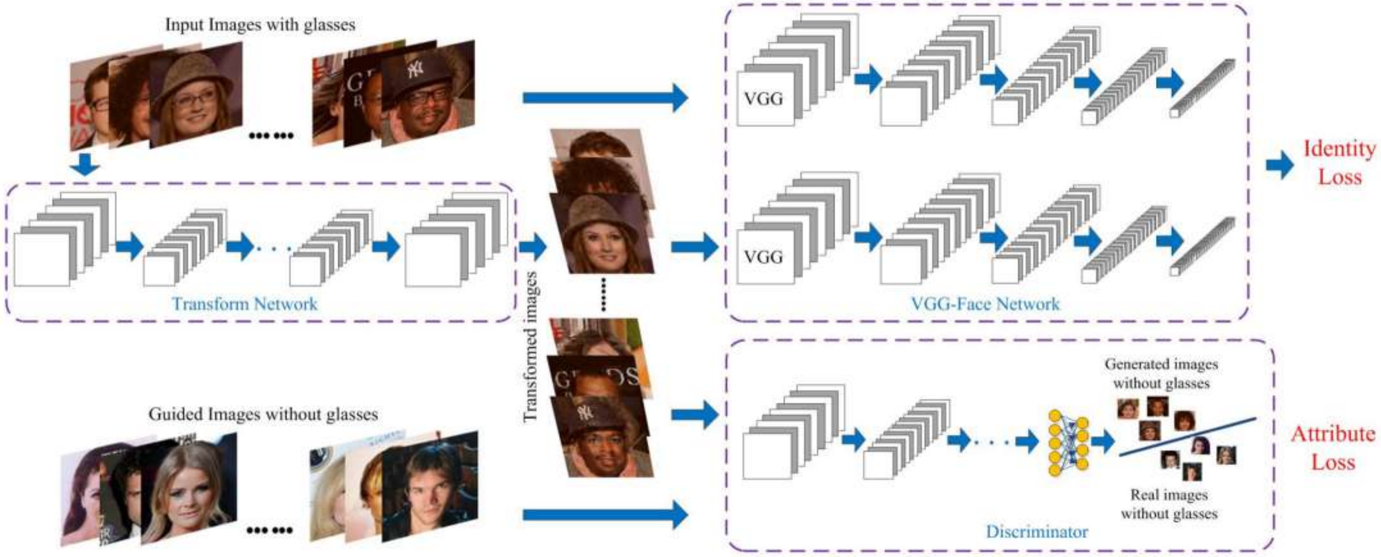


- Transform one image s.t. it exhibits attribute characterized by guided set of images.
- No strong supervision. Instead have guided set of images with desired attribute.
- Not a conditional GAN! Must retrain network (and have a dataset) for every attribute.

DIAT (2016) - Transform Network

- **Transform network (T)** maps input image to output with desired attribute.
- **Discriminator (D)** ensures that output is in desired distribution of images.

- **Identity loss.**
  Run face recognition on original and transformed images. Feature maps at later stages in network should be the same.
- **Attribute loss.**
  Train D to distinguish generated and reference images. Train T to create images that D can’t tell apart from reference images.
- **Perceptual regularization.**
  Encourage spatial smoothness **and** preserve small-scale texture detail.



Input Images with glasses
Transform Network
Transformed images
VGG
VGG
VGG-Face Network
Identity Loss
Guided Images without glasses
Generated images without glasses
Real images without glasses
Discriminator
Attribute Loss

a) Training of transform network

Note: GANimation says DIAT uses a cycle consistency loss for identity preservation; might be typo for CycleGAN.

DIAT (2016) - Perceptual Regularization

Encourage spatial smoothness ***and*** *preserve small-scale texture detail.*

g(x) Reconstruction network

  - **T** (encoder-decoder architecture) trained on **identity loss only**
  - Supposed to output the same thing as the input, but with bottleneck in middle

g(x) - x Artifacts stemming from transform network architecture and identity loss

f(…) Denoising network which is supposed to remove these artifacts

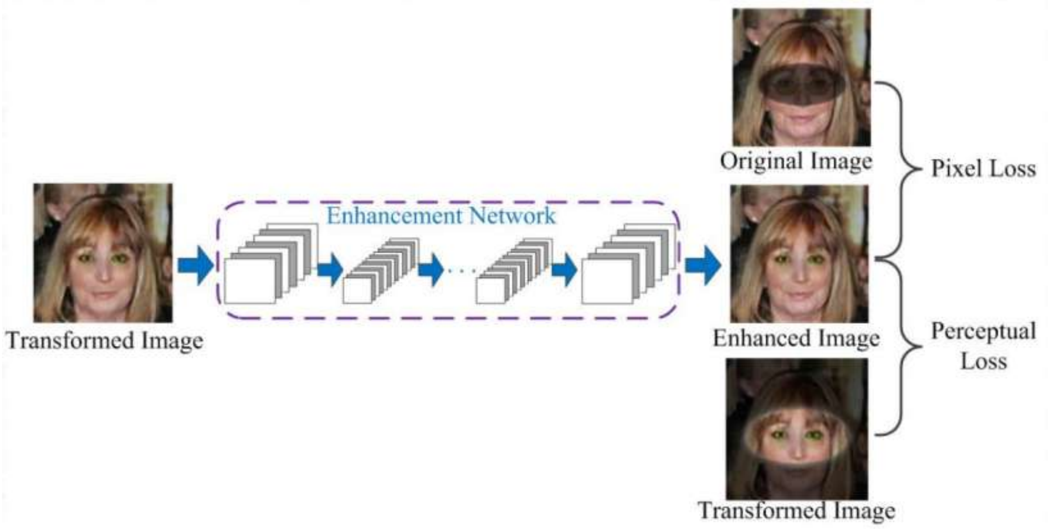
  - Two 3x3 convolutional layers
  - Objective: $\min_{f} \|f(g(\mathbf{x})) - \mathbf{x}\|_F^2 + \|f(\mathbf{x}) - \mathbf{x}\|_F^2$

All leads up to perceptual loss term: $\ell_{smooth}(T(\mathbf{x})) = \|f(T(\mathbf{x})) - T(\mathbf{x})\|_F^2$

DIAT (2016) - Enhancement Network

- **Enhancement network** improves visual quality of transform network's output.
- Obtain **local attribute mask** from convex hull of facial landmarks for attribute.

- **Pixel loss.**
  Outside of attribute mask region, enhanced image should be close to input image in pixel space.
- **Perceptual loss.**
  Within attribute mask region, enhanced image should be close to T image in VGG feature space.
  - "Perceptual" because the loss is in feature space (so over higher-level semantic concepts, not just pixels).

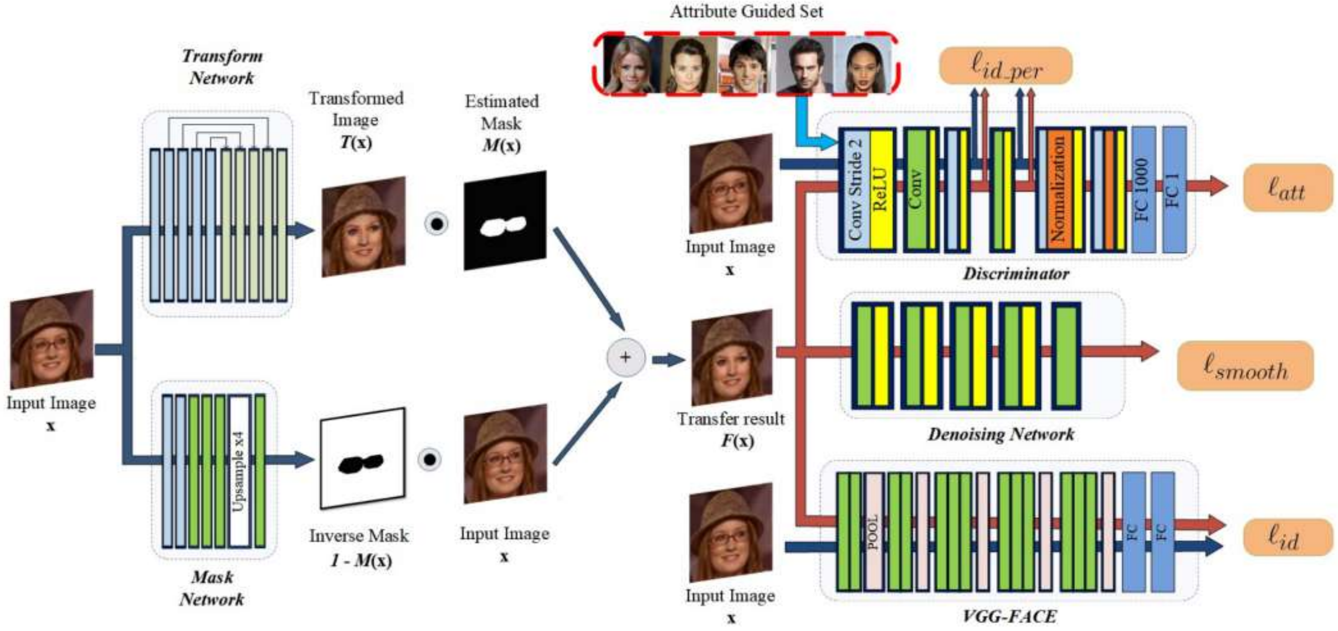


b) Training of enhancement network

DIAT (2018)

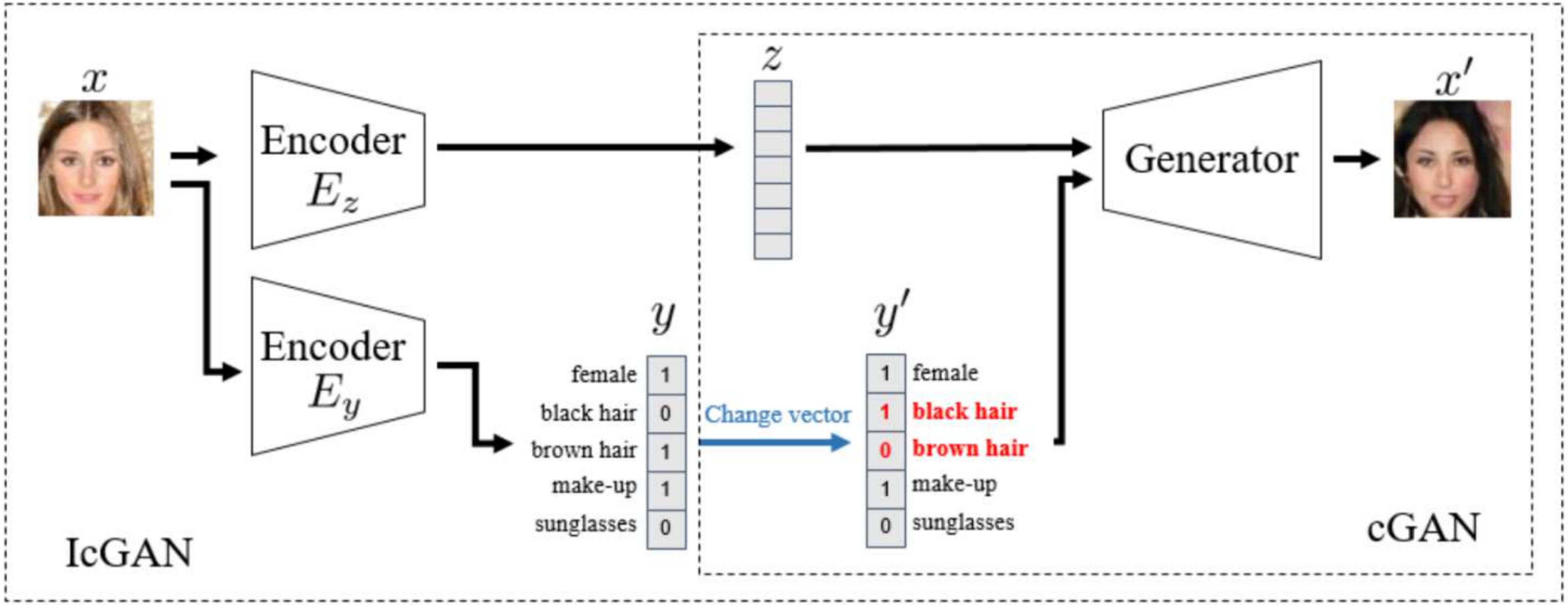
- DIAT authors released a V2 of their paper in December 2018 which seems to have adopted some of the GANimation methods.

- Predict mask and transformed image, then combine results in the same way as GANimation.



IcGAN (2016)

- Encoder + conditional GAN.



IcGAN (2016)

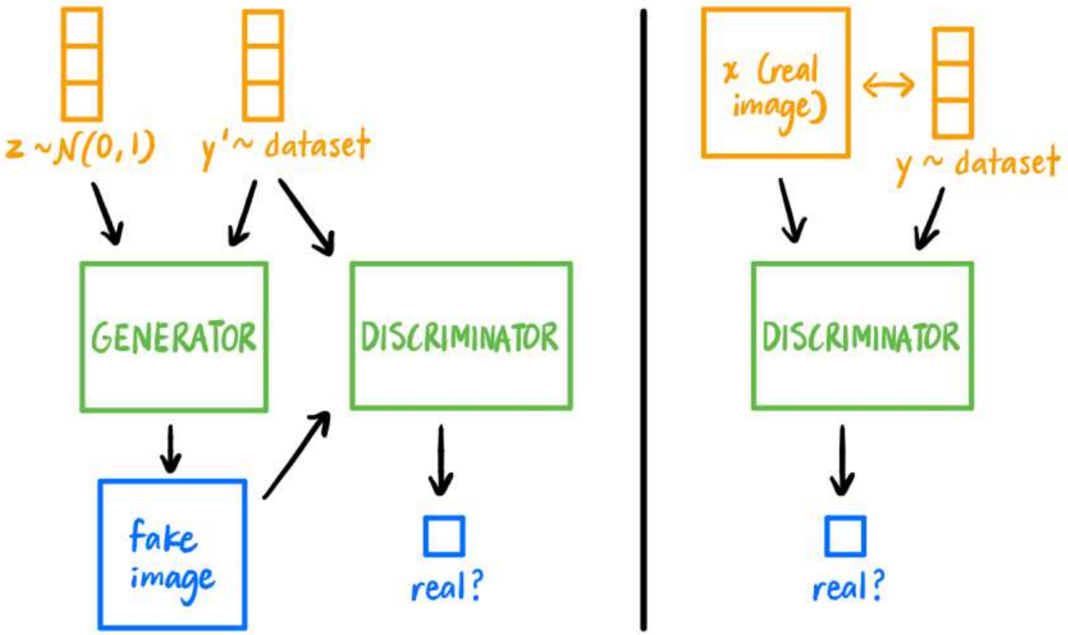
- **z-Encoder**: image → z (*latent vector*)
- **y-Encoder**: image → y (*conditional vector*)
- **Conditional GAN**: z, y → image

- *Invertible*: can go from image to z, y / can go from z, y to image
- Can theoretically condition on arbitrary attributes
  - But the way they do it, training might not work for continuous values
- Dataset: CelebA (images of celebrities with 40-attribute binary labels)

- Train z-encoder to predict z for data generated from z
- Train y-encoder to predict known conditional labels for dataset

IcGAN (2016) - cGAN Training

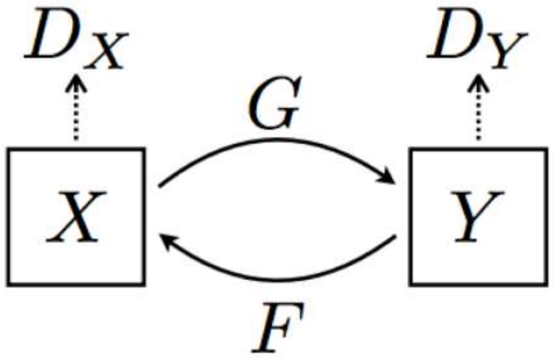
- **Conditional GAN (generator)**: z, y → image
- **Conditional GAN (discriminator)**: image, y → probability ∈ [0, 1] of being real

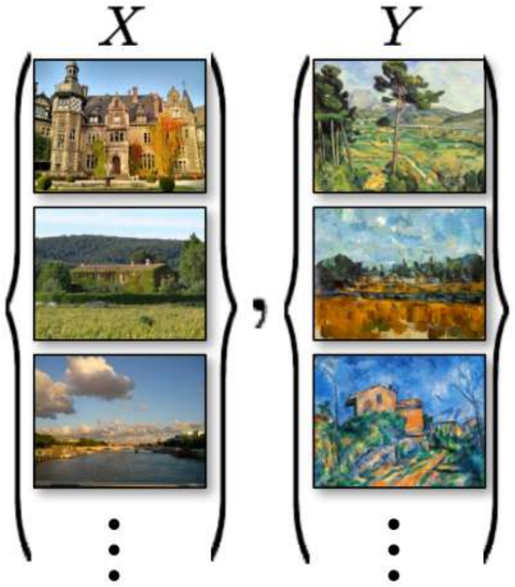


$$\min_{G} \max_{D} \mathbb{E}_{x,y \sim \text{dataset}}\left[\log D(x,y)\right] + \mathbb{E}_{z \sim \mathcal{N}(0,1),\, y' \sim \text{dataset}}\left[\log\left(1 - D(G(z,y'),y')\right)\right]$$

CycleGAN (2017)

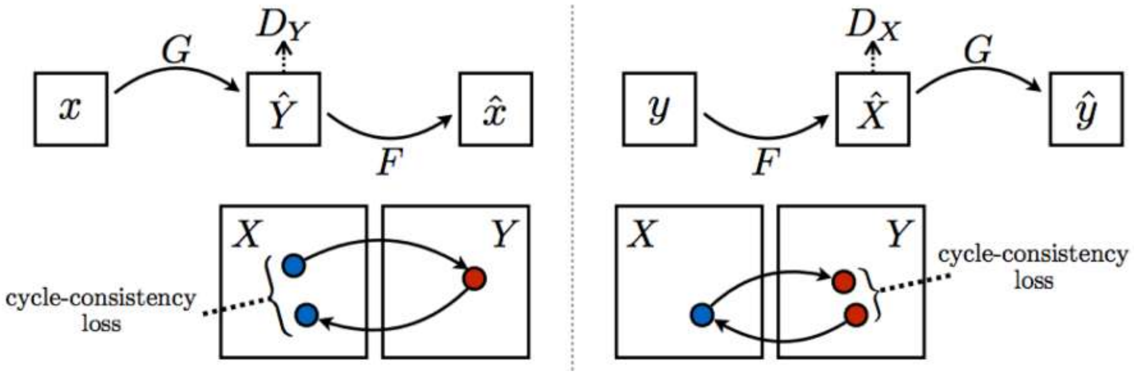
- Learn mappings between two* domains X and Y
- Dataset: a bunch of examples from X, a bunch of examples from Y
- Both mappings performed by vanilla GANs





CycleGAN (2017)

- Train using
  - standard adversarial losses
    - causes mappings to go to correct domain
  - cycle-consistency loss in both directions [make F(G(x)) = x, G(F(y)) = y]
    - causes mappings to preserve distinguishing attributes (“identity”)

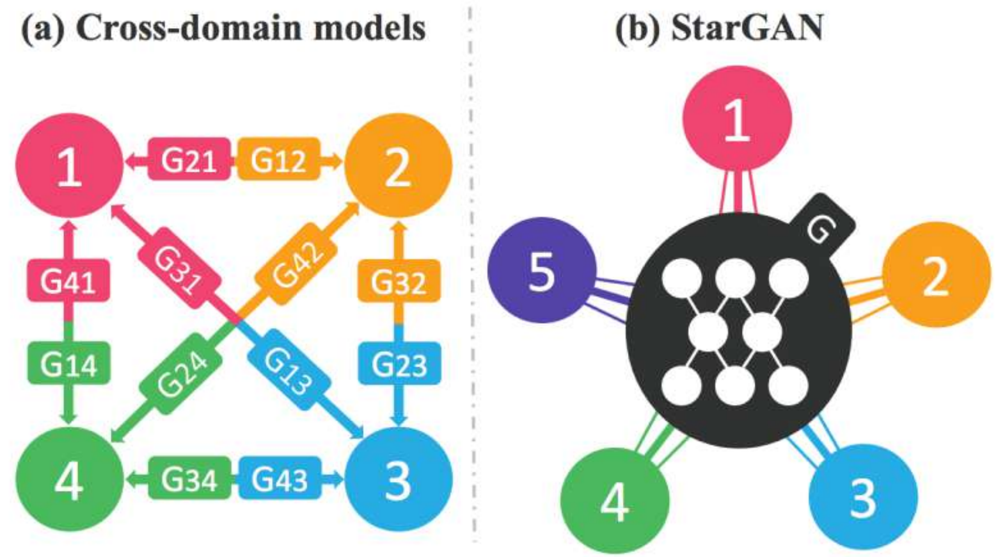


$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

StarGAN (2018)

Unified GANs for Multi-Domain Image-to-Image Translation

- Give model the ability to translate between many different domains
  - Avoid building model for every (X, Y)
  - IcGAN also did this
- Train conditional GAN to translate from one image domain to another
  - Condition on label representing domain (e.g. binary vector)



(a) Cross-domain models

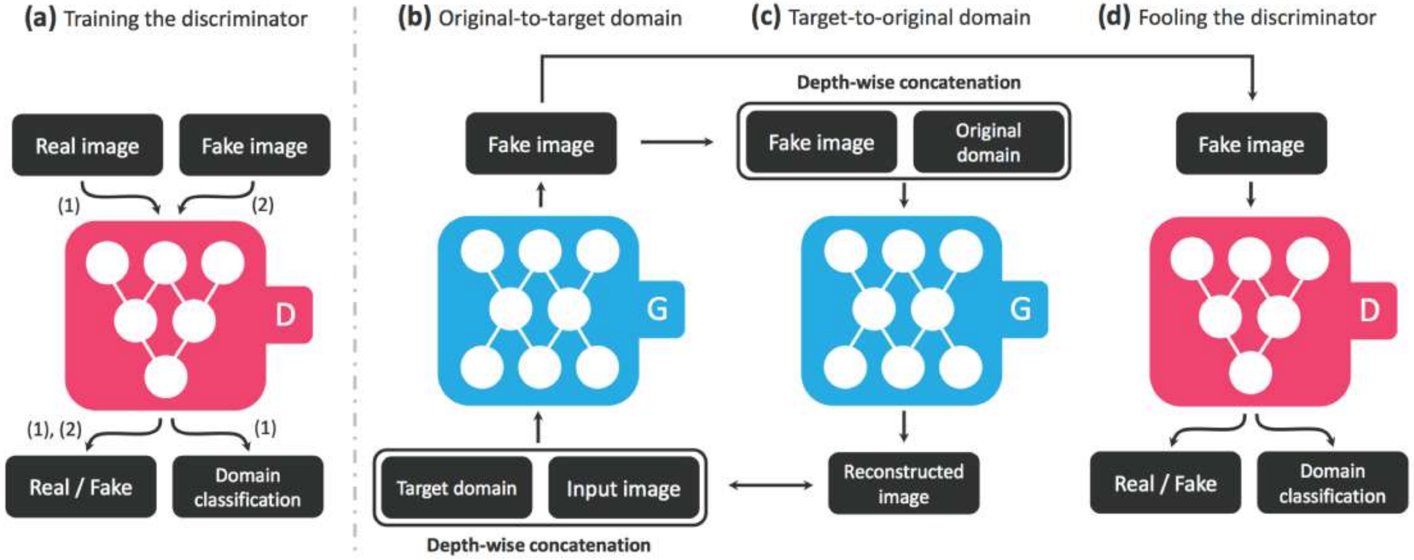
(b) StarGAN

Figure 2. Comparison between cross-domain models and our proposed model, StarGAN. (a) To handle multiple domains, cross-domain models should be built for every pair of image domains. (b) StarGAN is capable of learning mappings among multiple domains using a single generator. The figure represents a star topology connecting multi-domains.

StarGAN (2018)

Unified GANs for Multi-Domain Image-to-Image Translation

- Discriminator says “real/fake” **and** “which domain”
    - Only takes image, not conditional information (unlike IcGAN)



StarGAN (2018)

Unified GANs for Multi-Domain Image-to-Image Translation

- G(x, c) → transformed image
- D(x) → “real” probability, predicted class

Adversarial loss
Make sure output images look like real images

$$\mathcal{L}_{adv} = \mathbb{E}_x \left[\log D_{src}(x)\right] + \mathbb{E}_{x,c}[\log(1 - D_{src}(G(x,c)))]$$

Cycle consistency loss
Make sure identity is preserved (for generator)

$$\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'}[||x - G(G(x,c),c')||_1]$$

Domain classification loss
Make sure outputs’ domains are properly identified
Make sure outputs are put into the proper domains

$$\mathcal{L}^r_{cls} = \mathbb{E}_{x,c'}[-\log D_{cls}(c'|x)]$$

$$\mathcal{L}^f_{cls} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))]$$

StarGAN (2018)

Unified GANs for Multi-Domain Image-to-Image Translation

- In practice, use Wasserstein GAN objective with gradient penalty

$$\begin{aligned}\mathcal{L}_{adv} = &\, \mathbb{E}_x\left[\log D_{src}(x)\right] \; + \\ &\, \mathbb{E}_{x,c}[\log\left(1 - D_{src}(G(x,c))\right)]\end{aligned} \quad \longrightarrow \quad \begin{aligned}\mathcal{L}_{adv} = &\, \mathbb{E}_x[D_{src}(x)] - \mathbb{E}_{x,c}[D_{src}(G(x,c))] \\ &\, - \lambda_{gp}\, \mathbb{E}_{\hat{x}}[(||\nabla_{\hat{x}} D_{src}(\hat{x})||_2 - 1)^2]\end{aligned}$$

- Use PatchGAN architecture for discriminator network
    - Classifies whether image patches are real or fake (then averages results)
    - Implement as regular CNN, receptive fields of output neurons are patches

- Also use instance normalization
    - Normalize each batch independently

Wasserstein GAN (WGAN)

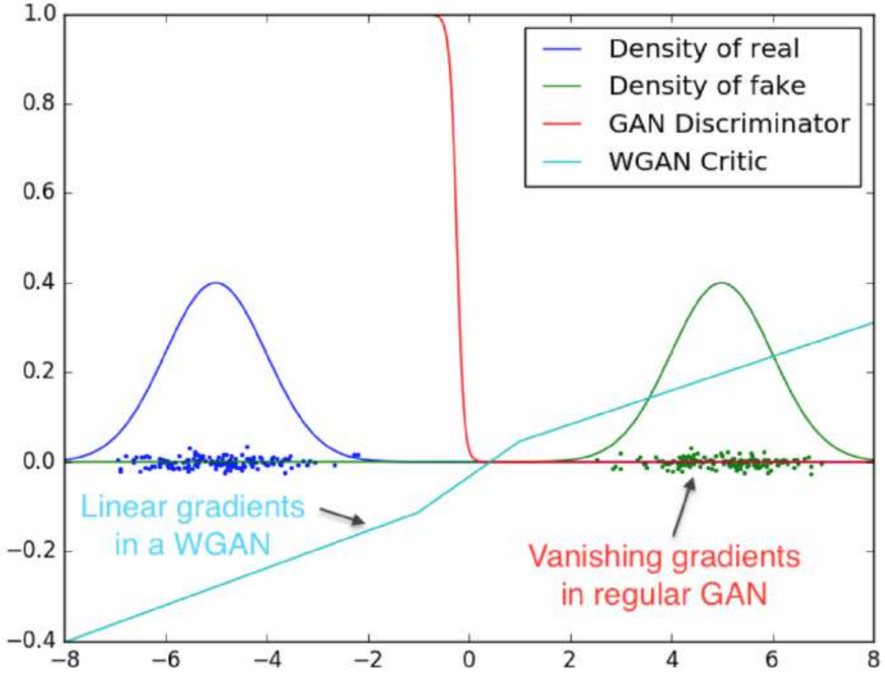
- A GAN which minimizes [an approximation of] the Wasserstein (Earth Mover's) distance between two probability distributions

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \big[\, \|x - y\| \,\big]$$

- In the case of a GAN, $\mathbb{P}_\mathrm{r}$ is the real distribution and $\mathbb{P}_\mathrm{g}$ is the generated distribution; we want to make these two match

Wasserstein Distance

- Minimum work to move and transform one probability distribution into another probability distribution (problem of **optimal transport**)



Advantages of WGAN

- GAN training is hard:
  - Training slow and unstable (e.g. vanishing gradients if G or D unbalanced)
  - Prone to mode collapse (G producing outputs that all look the same, D fooled)
  - Loss not interpretable
- WGAN training is better, more theoretically justified:
  - Wasserstein distance is smooth and meaningful (more so than KL or Jensen-Shannon divergence) even when probability distributions don’t overlap
  - Empirically seems to avoid problem of mode collapse
  - Loss now has a connection to quality of generator

Kantorovich-Rubinstein Duality

- How to compute Wasserstein distance? Infimum is hard to deal with
- Kantorovich-Rubinstein duality tells us that the Wasserstein distance

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

is equivalent to

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

(note the supremum over 1-Lipschitz functions)

Kantorovich-Rubinstein Duality

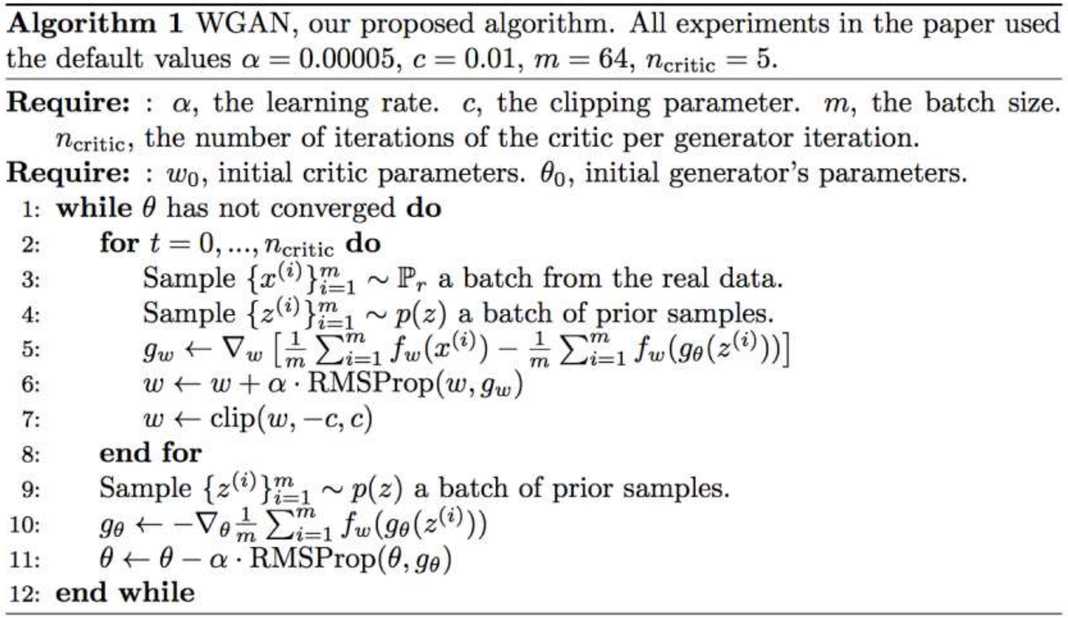
- Less hard to deal with supremum
- Approximate

$$\sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

as

$$\max_{f\in\mathcal{F}} \mathbb{E}_{x\sim\mathbb{P}_r}\left[f(x)\right] - \mathbb{E}_{x\sim\mathbb{P}_\theta}\left[f(x)\right]$$

for $\boldsymbol{\mathcal{F}}$ the set of 1-Lipschitz functions



Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

```
Require: : $\alpha$, the learning rate. $c$, the clipping parameter. $m$, the batch size.
    $n_{\text{critic}}$, the number of iterations of the critic per generator iteration.
Require: : $w_0$, initial critic parameters. $\theta_0$, initial generator's parameters.
while $\theta$ has not converged do
    for $t = 0, ..., n_{\text{critic}}$ do
        Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
        Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
        $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
        $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
        $w \leftarrow \text{clip}(w, -c, c)$
    end for
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
    $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while
```

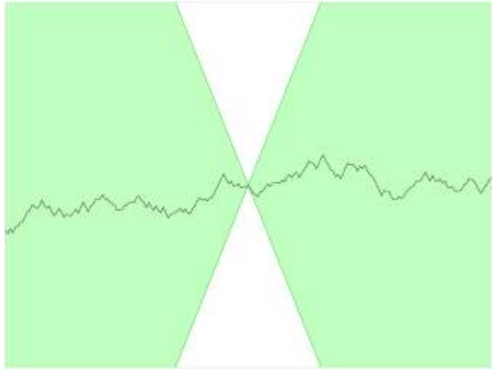
Note: in our case, f is the critic (discriminator).

Lipschitz Continuity

- For f(x) to be 1-Lipschitz continuous, we must have

$$|f(x_1) - f(x_2)| \leq |x_1 - x_2| \;\; \rightarrow \;\; \frac{|f(x_1) - f(x_2)|}{|x_1 - x_2|} \leq 1$$

- LHS is finite difference approximation of derivative
- *Another perspective*: gradients must have norm ≤ 1 everywhere



Enforcing 1-Lipschitz Continuity

$$\min_G \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \big[D(\boldsymbol{x}) \big] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \big[D(\tilde{\boldsymbol{x}})) \big]$$

WGAN objective as written by Gulrajani et al.

- Need our critic D to be 1-Lipschitz continuous
- To enforce, use gradient penalty

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \left[D(\tilde{\boldsymbol{x}}) \right] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \left[D(\boldsymbol{x}) \right]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[(\| \nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}}) \|_2 - 1)^2 \right]}_{\text{Our gradient penalty}}$$

Wasserstein GAN Recap

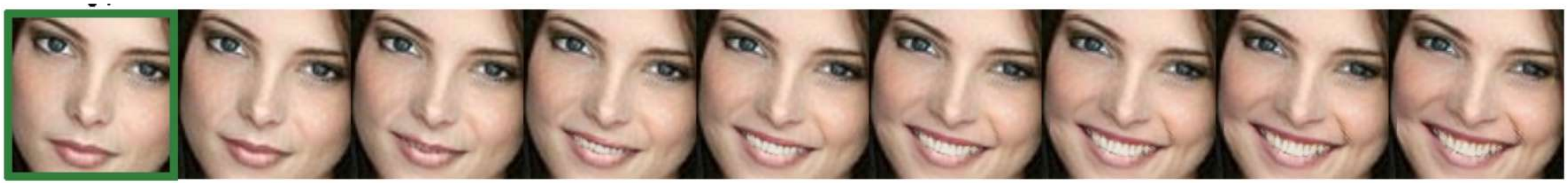
- Minimize Wasserstein distance (instead of, say, Jensen-Shannon divergence) between distributions
- Must limit gradients somehow, e.g. with gradient penalty
    - (limit the space of possible discriminators)
- A lot of theory explaining why WGANs help resolve training issues

Summary (2016-2018)

- Everyone uses GANs, nobody has supervised input-output pairs
  - “Meta-supervision”: supervise using constraints on outputs
- No one uses continuous values for domains
  - Generators are not fully expressive
- DIAT and CycleGAN* not conditional, just between two domains

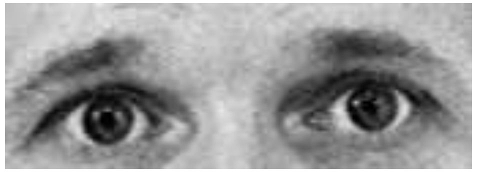
The Main Event: GANimation

- Why am I talking about all this?
  - For context/comparison
  - Also, GANimation reuses a lot of these ideas!

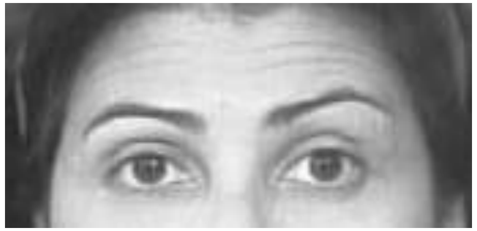


Facial Action Coding System

- **Action unit**: corresponds to a muscle movement
  - **Magnitude of AU activation**: *how much* the muscle is contracted
- Continuous representation
- ~30 AUs related to facial expressions



AU1: inner brow raiser



AU2: outer brow raiser

GANimation (2018)

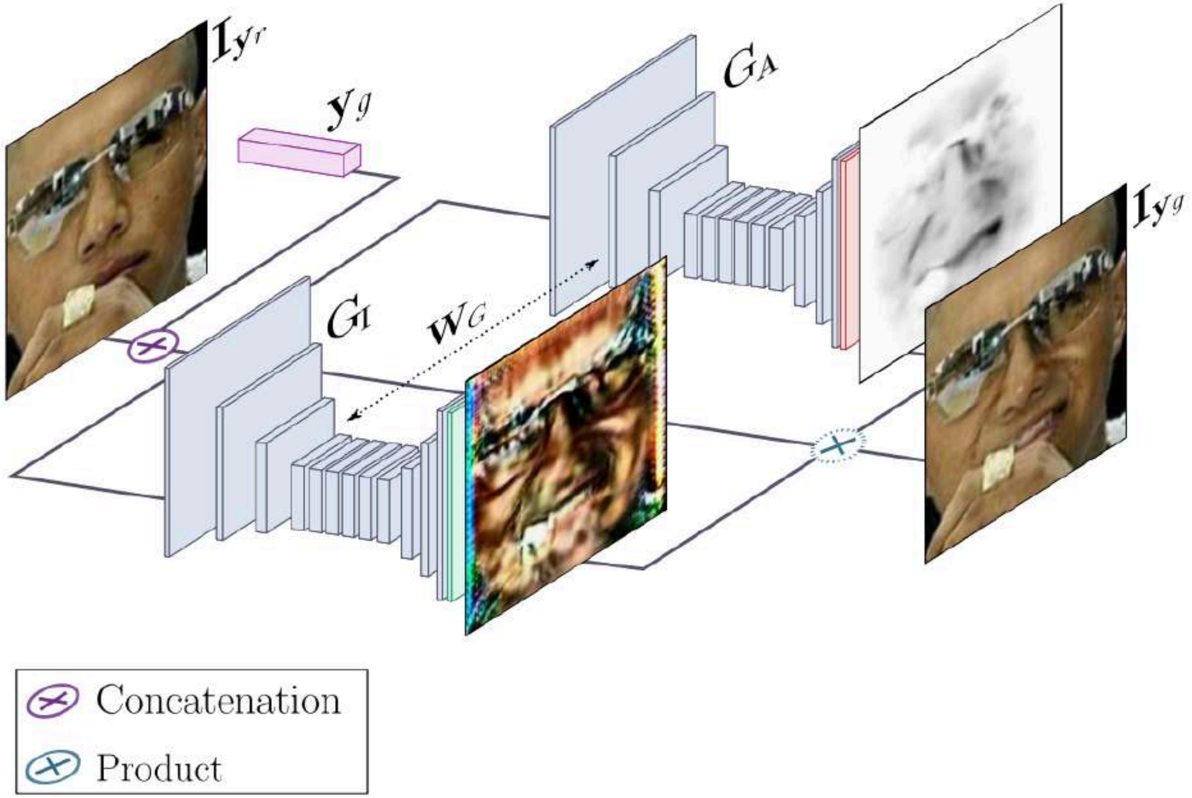
- Conditional Wasserstein GAN
  - Generator
    - **In:** face, vector of 17 action unit activations
    - **Out:** face with action units activated as desired / attention mask
  - Discriminator
    - **In:** face
    - **Out:** real or fake? / predicted vector of action unit activations
- Attention mask prediction
- Four loss components (train without supervision input/output data)

GANimation (2018)

- Basically StarGAN, with
  - continuous conditional vector
  - attention mask prediction (importance of each pixel for desired expression)

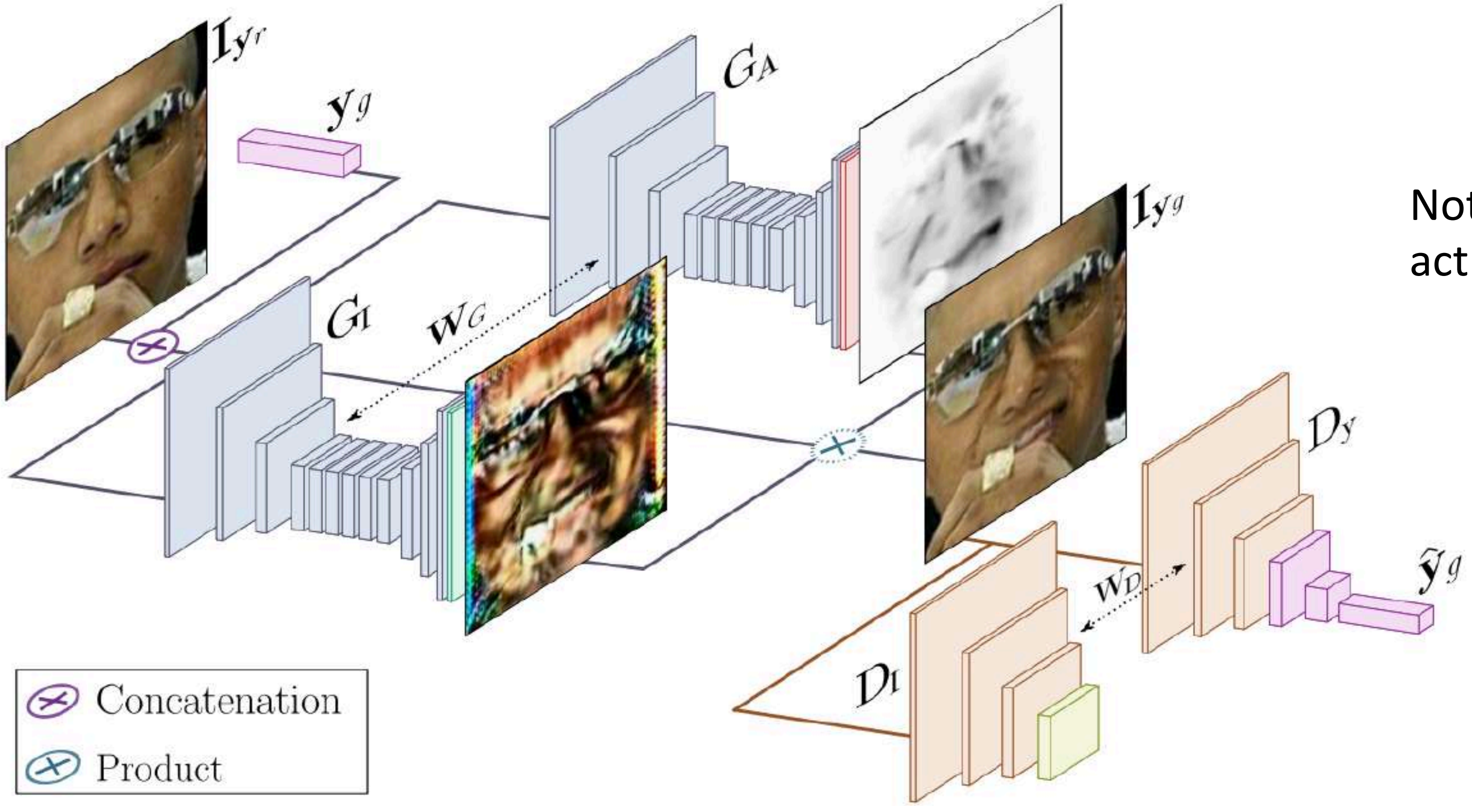
Input data:	$\{\mathbf{I}^{m}_{\mathbf{y}_r}, \mathbf{y}^{m}_{r}, \mathbf{y}^{m}_{g}\}^{M}_{m=1}$	real image / AUs for real image / AUs for generated image
Learn this:	$\mathcal{M} : (\mathbf{I}_{\mathbf{y}_r}, \mathbf{y}_g) \rightarrow \mathbf{I}_{\mathbf{y}_g}$	real image / AUs for generated image → generated image

Inference Architecture



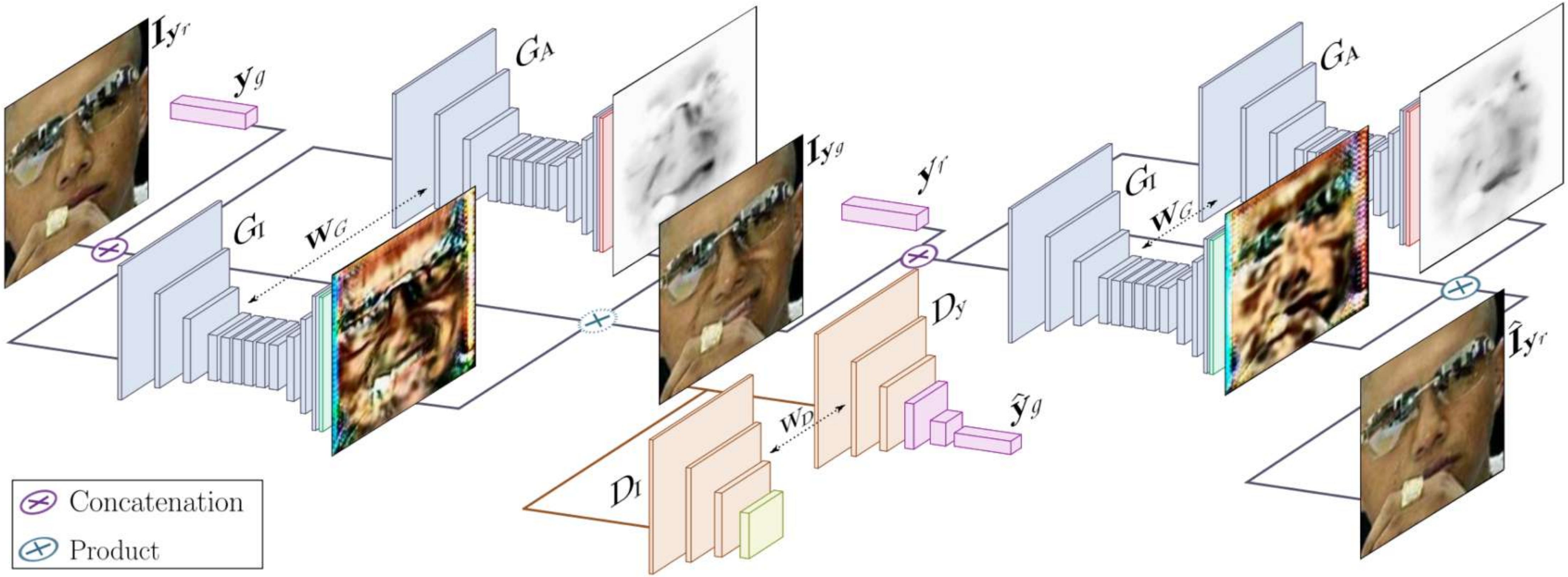
Note: just two output heads, not actually two whole separate networks

Inference/Training Architecture



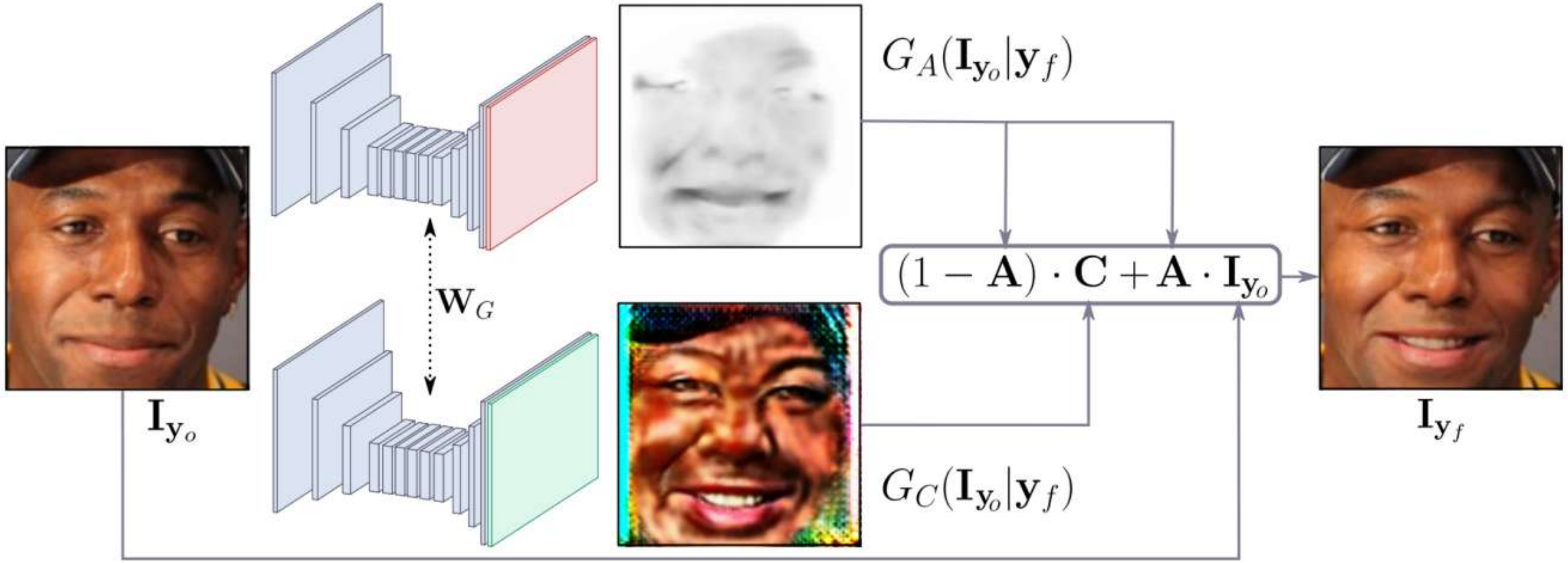
Note: just two output heads, not actually two whole separate networks

Inference/Training Architecture



Attention

- Makes it easier to produce sharp, realistic images
  - Don’t have to deal with irrelevant stuff, just copy over (/blend)!
- Adds robustness to background and illumination


$G_A(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f)$
$\mathbf{W}_G$
$(1-\mathbf{A})\cdot\mathbf{C}+\mathbf{A}\cdot\mathbf{I}_{\mathbf{y}_o}$
$\mathbf{I}_{\mathbf{y}_o}$
$\mathbf{I}_{\mathbf{y}_f}$
$G_C(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f)$

Some Motivation for Attention

- Difficult for network to generate backgrounds (and hair e.g.), which might vary drastically for every image it gets as input.
  - But we're not even trying to change the backgrounds
  - So why bother reconstructing them? (same argument for anything irrelevant)

Image source: Which Face Is Real

Dataset

- EmotioNet with AUs annotated using method from Baltrusaitis et al. (Cross-dataset learning and person-specific normalisation for automatic action unit detection)
  - i.e. pseudo-ground truth labels
- One million faces in the wild
- Only use 200,000 images from dataset (faster training)

Query by Action Units	Number of images	Retrieved images
AU 4	281,732	. . .
AU 6	267,660	. . .

More Architectural and Training Details

- Enforces Lipschitz constraint with gradient penalty
- Uses instance norm instead of batch norm in the generator
- Four loss components

GANimation	**StarGAN analogue**
• image adversarial loss	“adversarial loss”
• attention loss	
• conditional expression loss	“domain classification loss”
• identity loss	“cycle consistency loss”

Realistic-looking images
Meaningful attention masks
Correct facial expressions
Preservation of identity

Term #1: Image Adversarial Loss

(G minimizes the terms it affects, D maximizes the terms it affects)

Minimize ~Wasserstein distance between $G(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f)$ *and* $\mathbf{I}_{\mathbf{y}_o}$ *distributions*

Penalize D's gradient norm as it deviates from 1

$$\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}[D_{\mathrm{I}}(G(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f))] - \mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}[D_{\mathrm{I}}(\mathbf{I}_{\mathbf{y}_o})] + \lambda_{\mathrm{gp}}\mathbb{E}_{\widetilde{I}\sim\mathbb{P}_{\widetilde{I}}}\left[(\|\nabla_{\widetilde{I}}D_{\mathrm{I}}(\widetilde{I})\|_2 - 1)^2\right]$$

- **Photorealism**: make *generated images* look like *images in training set*
- "o" is "original", "f" is "final"
- $\tilde{I} \backsim \mathbb{P}_{\tilde{I}}$ is a random interpolation between $G(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f)$ and $\mathbf{I}_{\mathbf{y}_o}$

Realistic-looking images
Meaningful attention masks
Correct facial expressions
Preservation of identity

Term #2: Attention Loss

(G minimizes the terms it affects, D maximizes the terms it affects)

$$\lambda_{\mathrm{TV}}\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\sum_{i,j}^{H,W}\left[(\mathbf{A}_{i+1,j}-\mathbf{A}_{i,j})^2+(\mathbf{A}_{i,j+1}-\mathbf{A}_{i,j})^2\right]\right]+\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\|\mathbf{A}\|_2\right]$$

- Enforce attention mask to be
  - smooth (total variation regularization)
  - utilized (L2 regularization) – don’t go to 1!

Realistic-looking images
Meaningful attention masks
Correct facial expressions
Preservation of identity

Term #3: Conditional Expression Loss

(G minimizes the terms it affects, D maximizes the terms it affects)

Generator should produce output with y_f expression

Discriminator should identify labeled expressions

$$\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o} \sim \mathbb{P}_o} \left[\| D_{\mathrm{y}}(G(\mathbf{I}_{\mathbf{y}_o} | \mathbf{y}_f))] - \mathbf{y}_f \|_2^2 \right] + \mathbb{E}_{\mathbf{I}_{\mathbf{y}_o} \sim \mathbb{P}_o} \left[\| D_{\mathrm{y}}(\mathbf{I}_{\mathbf{y}_o}) - \mathbf{y}_o \|_2^2 \right]$$

- Make sure the final expression matches the desired expression
  - Discriminator should say that the final expression is as desired
- Discriminator should be good at determining expressions
  - Make sure it can identify original expression

Term #4: Identity Loss

Realistic-looking images
Meaningful attention masks
Correct facial expressions
Preservation of identity

(G minimizes the terms it affects, D maximizes the terms it affects)

$$\mathcal{L}_{\mathrm{idt}}(G, \mathbf{I}_{\mathbf{y}_o}, \mathbf{y}_o, \mathbf{y}_f) = \mathbb{E}_{\mathbf{I}_{\mathbf{y}_o} \sim \mathbb{P}_o}\left[\|G(G(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f)|\mathbf{y}_o) - \mathbf{I}_{\mathbf{y}_o}\|_1\right]$$

- *Cycle consistency loss*: make sure that identity is preserved by transforming the generated image to have the same attribute(s) as the original, and checking that the result is the same as the original
    - When you change an attribute, only that attribute should change
    - The identity of the subject should remain the same

Realistic-looking images
Meaningful attention masks
Correct facial expressions
Preservation of identity

Term #4: Identity Loss

(G minimizes the terms it affects, D maximizes the terms it affects)

$$\mathcal{L}_{\text{idt}}(G, \mathbf{I}_{\mathbf{y}_o}, \mathbf{y}_o, \mathbf{y}_f) = \mathbb{E}_{\mathbf{I}_{\mathbf{y}_o} \sim \mathbb{P}_o} \left[\| G(G(\mathbf{I}_{\mathbf{y}_o} | \mathbf{y}_f) | \mathbf{y}_o) - \mathbf{I}_{\mathbf{y}_o} \|_1 \right]$$

- **Why L1 loss?** Enforces correctness for low frequencies.
  - L1/L2 losses both tend to generate blurry results for image generation.
    - Perhaps related to multimodality problem (optimize for blurry middle value?).
  - Nevertheless, L1 is good at capturing low frequency information.
- Seems like perceptual loss might be better (see e.g. “Loss Functions for Image Restoration with Neural Networks” by Zhao et al.), but authors find no improvement.

Term #4: Identity Loss

Realistic-looking images
Meaningful attention masks
Correct facial expressions
Preservation of identity

(G minimizes the terms it affects, D maximizes the terms it affects)

$$\mathcal{L}_{\text{idt}}(G, \mathbf{I}_{\mathbf{y}_o}, \mathbf{y}_o, \mathbf{y}_f) = \mathbb{E}_{\mathbf{I}_{\mathbf{y}_o} \sim \mathbb{P}_o} [\|G(G(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f)|\mathbf{y}_o) - \mathbf{I}_{\mathbf{y}_o}\|_1]$$

- *Other side of coin*: PatchGAN discriminator focuses on local patches and thus enforces correctness for high frequencies.
  - (Intuition: high frequencies are more apparent at the scale of patches.)

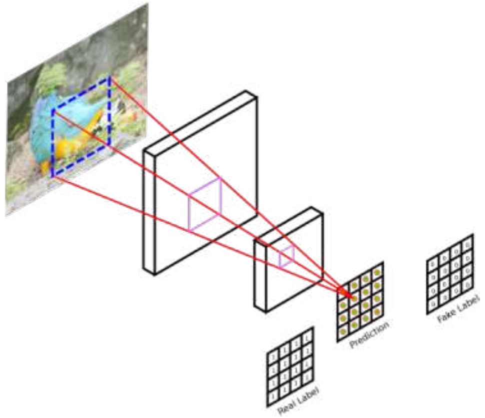


Image source: GroundAI

Some Strengths and Weaknesses

- Strengths:
  - Can handle attributes in continuous spaces
  - Uses attention to improve output resolution and handle changing backgrounds
    - Just ignores anything irrelevant to face transformation
  - Can use example image as proxy for AUs (like in previous methods)
    - Just feed to discriminator, get correct AUs, then use AUs for generation
    - **Or** use detector (e.g. the one used to annotate EmotioNet)
      - *Side note*: I wonder how good the discriminator is at identifying AUs

- Weaknesses:
  - Need faces to be cropped; must use auxiliary face detector for general images
  - If support for continuous conditional vectors is desired, training is somewhat dependent on having continuous annotations which might be hard to acquire

Experimental Evaluation

Single Action Unit Activations

Discrete Emotions Editing (Comparison)

Support for Images in the Wild

Multiple Action Unit Activations

Variability of Facial Expressions

Some Limitations and Failure Cases

Single Action Unit Activations

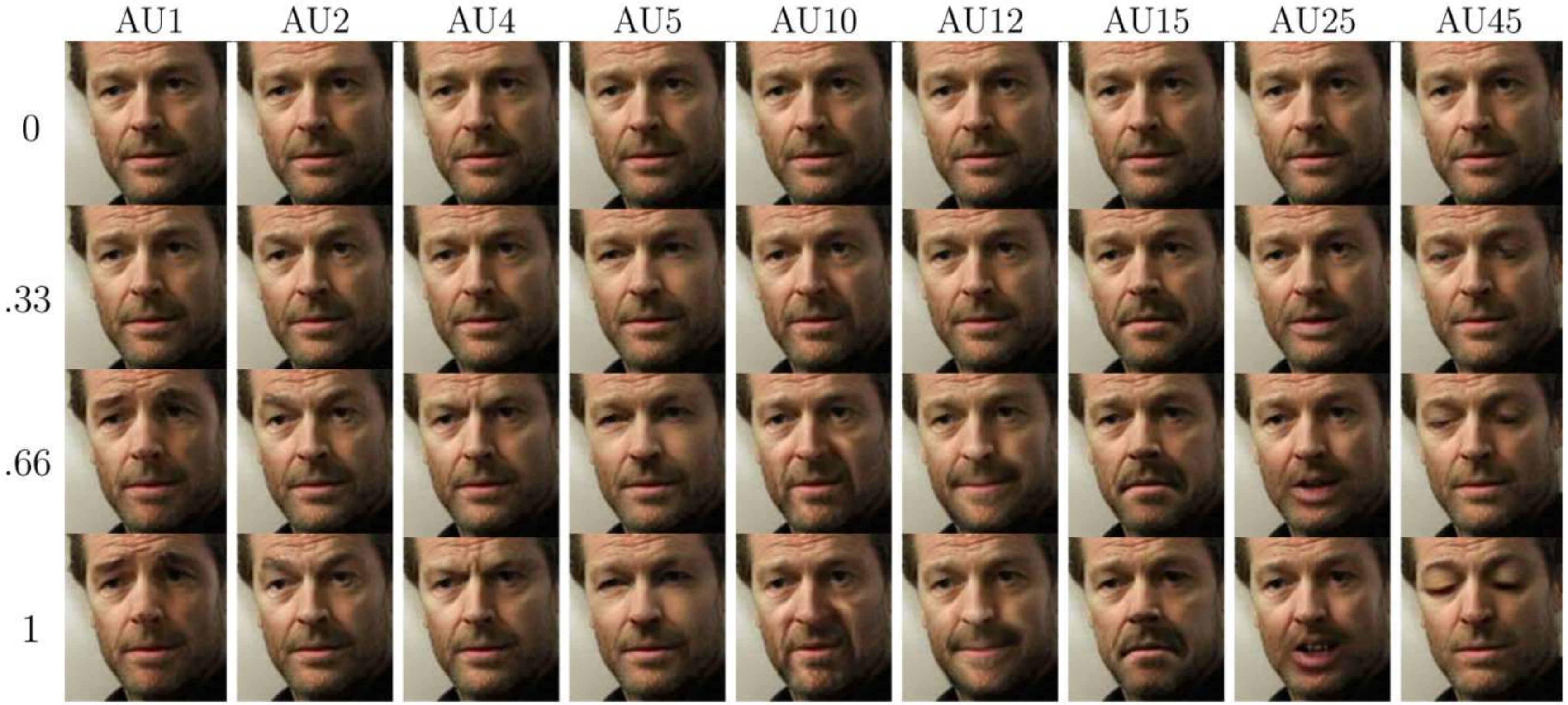


Fig. 4. Single AUs edition. Specific AUs are activated at increasing levels of intensity (from 0.33 to 1). The first row corresponds to a zero intensity application of the AU which correctly produces the original image in all cases.

Learns identity transformation for 0 AU activation

Single Action Unit Activations

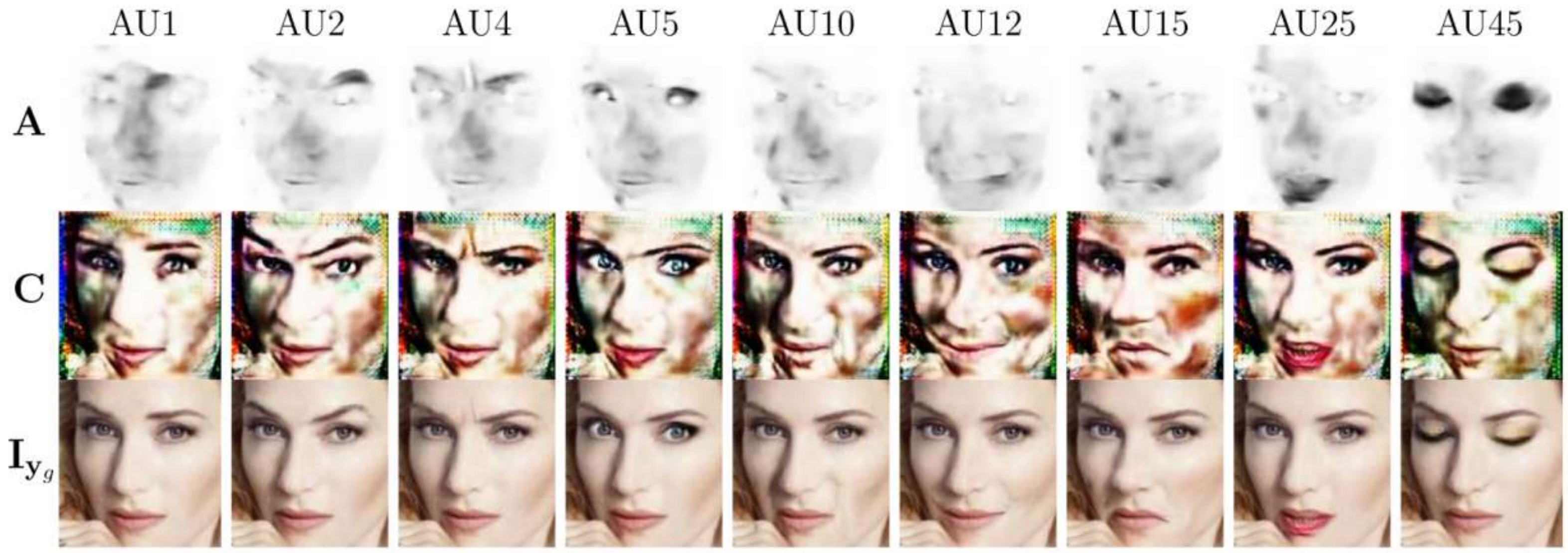


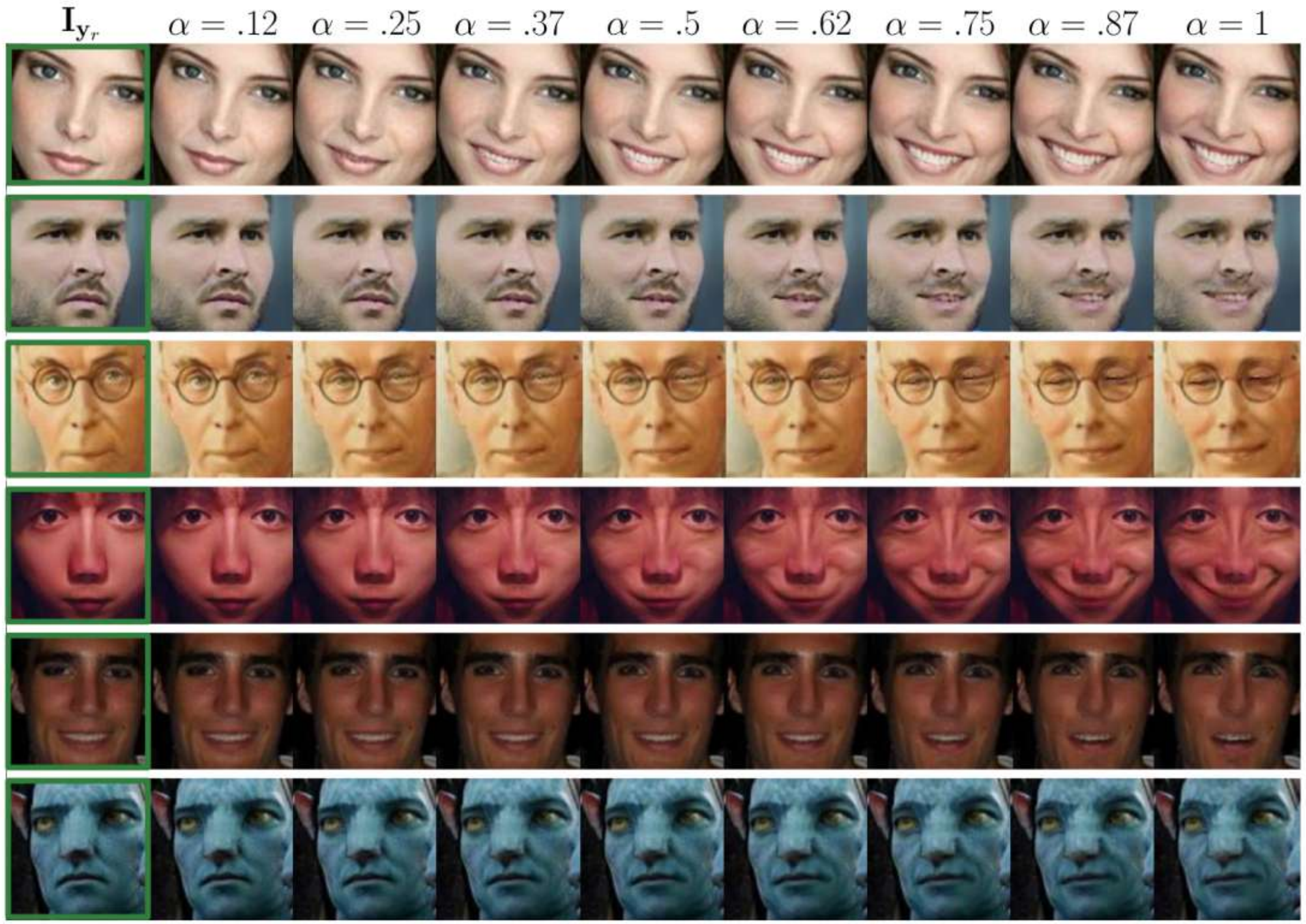
Fig. 5. Attention Model. Details of the intermediate attention mask **A** (first row) and the color mask **C** (second row). The bottom row images are the synthesized expressions. Darker regions of the attention mask **A** show those areas of the image more relevant for each specific AU. Brighter areas are retained from the original image.

Attention mask learns to focus on AUs

Color mask has noise, but doesn't matter

Result is noise-free, most pixels just copied!

Multiple Action Unit Activations

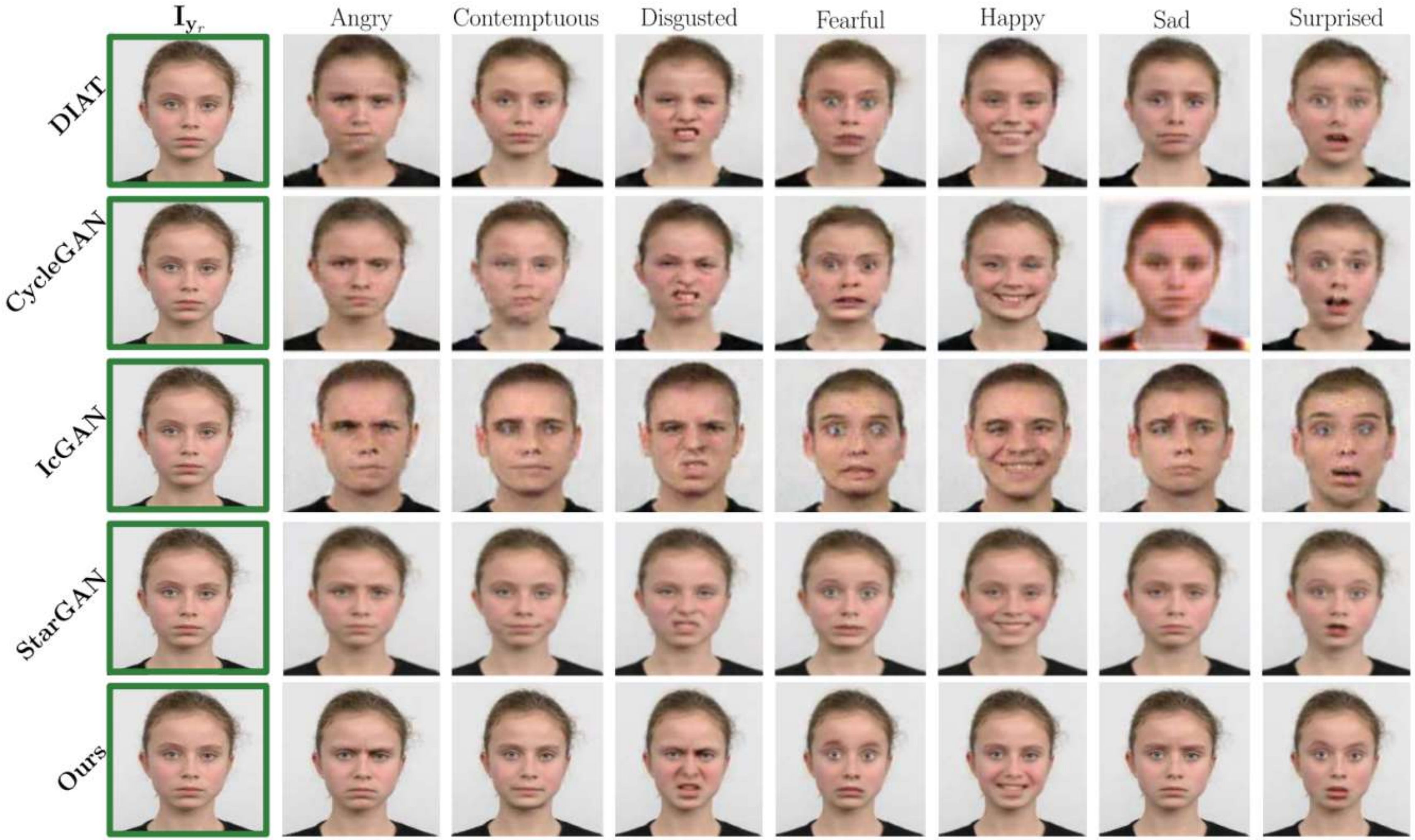


Interpolate between expressions

$$\alpha \mathbf{y}_g + (1 - \alpha)\mathbf{y}_r$$

← Robust to lighting conditions and non-real world data

Comparison with Previous Methods



GANimation produces higher-resolution results!

Many Anatomically Coherent Expressions!

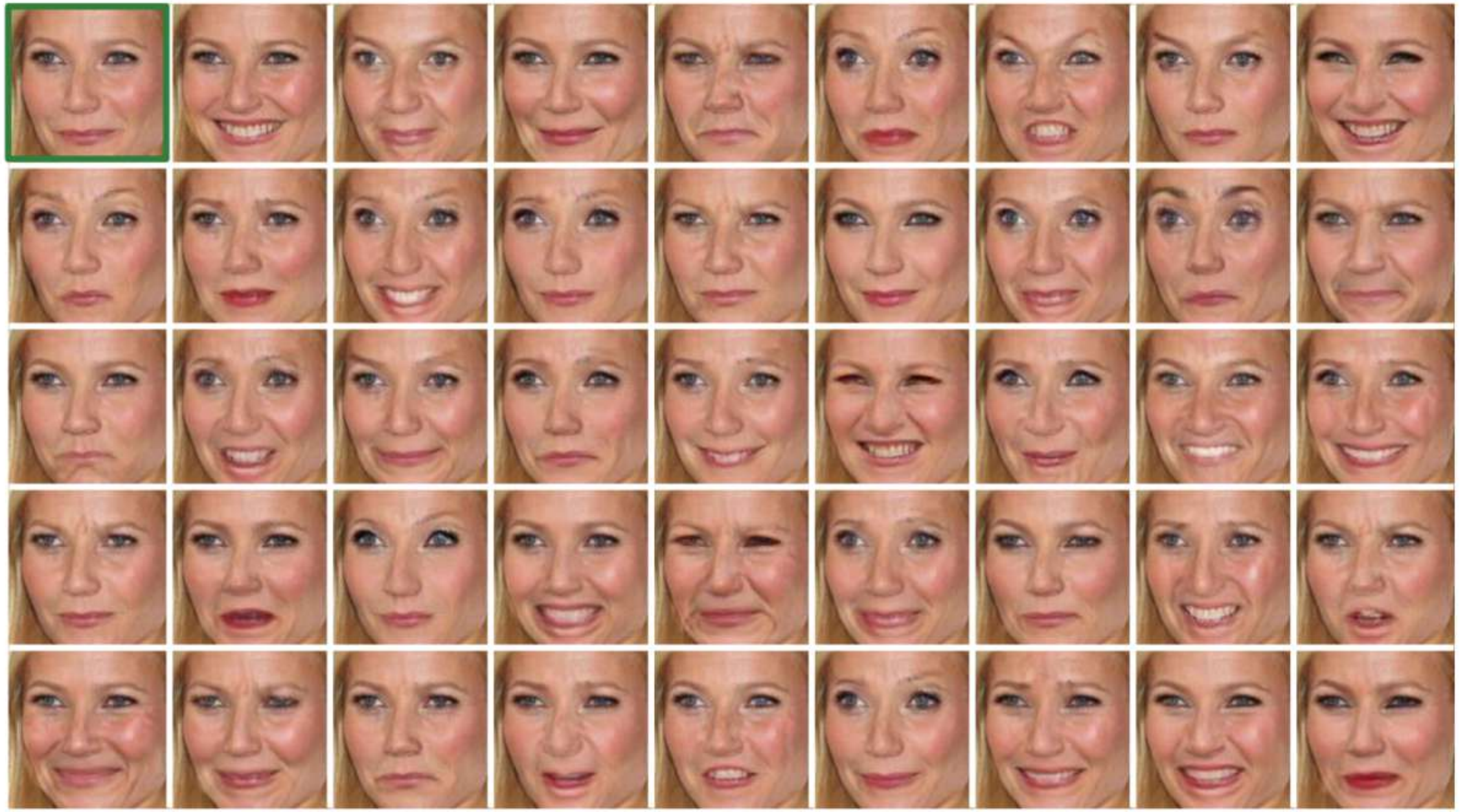
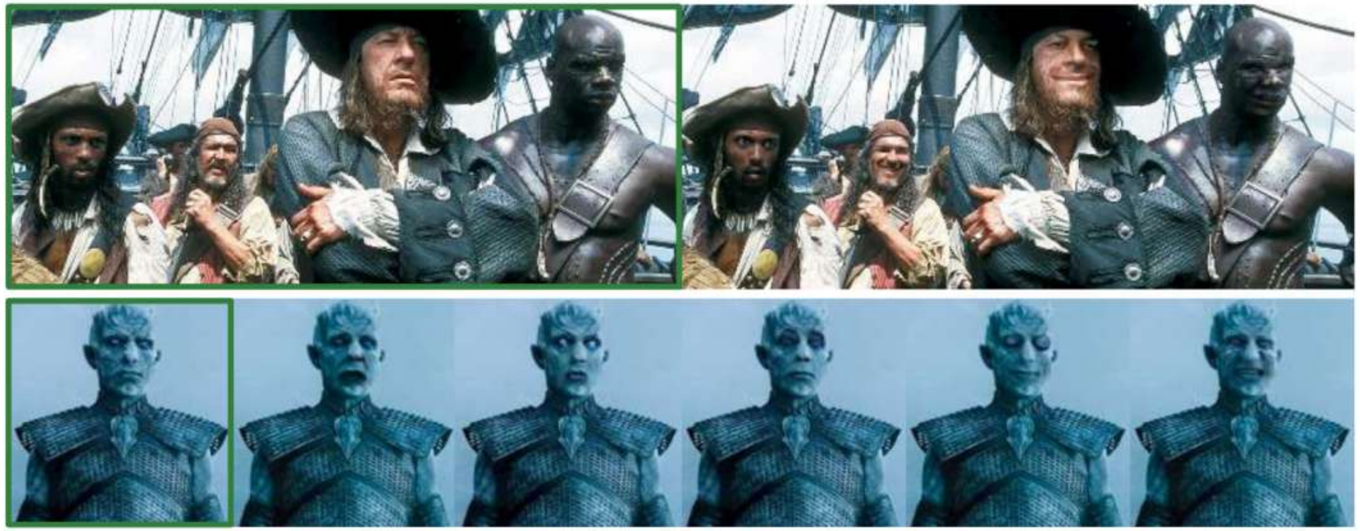


Fig. 7. Sampling the face expression distribution space. As a result of applying our AU-parametrization through the vector $\mathbf{y}_g$, we can synthesize, from the same source image $\mathbf{I}_{\mathbf{y}_r}$, a large variety of photo-realistic images.

Some generated facial expressions which are based on only 14 AUs

“In the Wild” Images

- (Localize and crop single faces using detector)
    - GANimation only operates on a single cropped face at once
- Thanks to the attention mask, can handle images in the wild
- No comparison with previous methods on this front?



- Different illumination over different parts of faces
- Variety of resolutions for faces
- Bluish skin texture

Limitations and Failure Cases

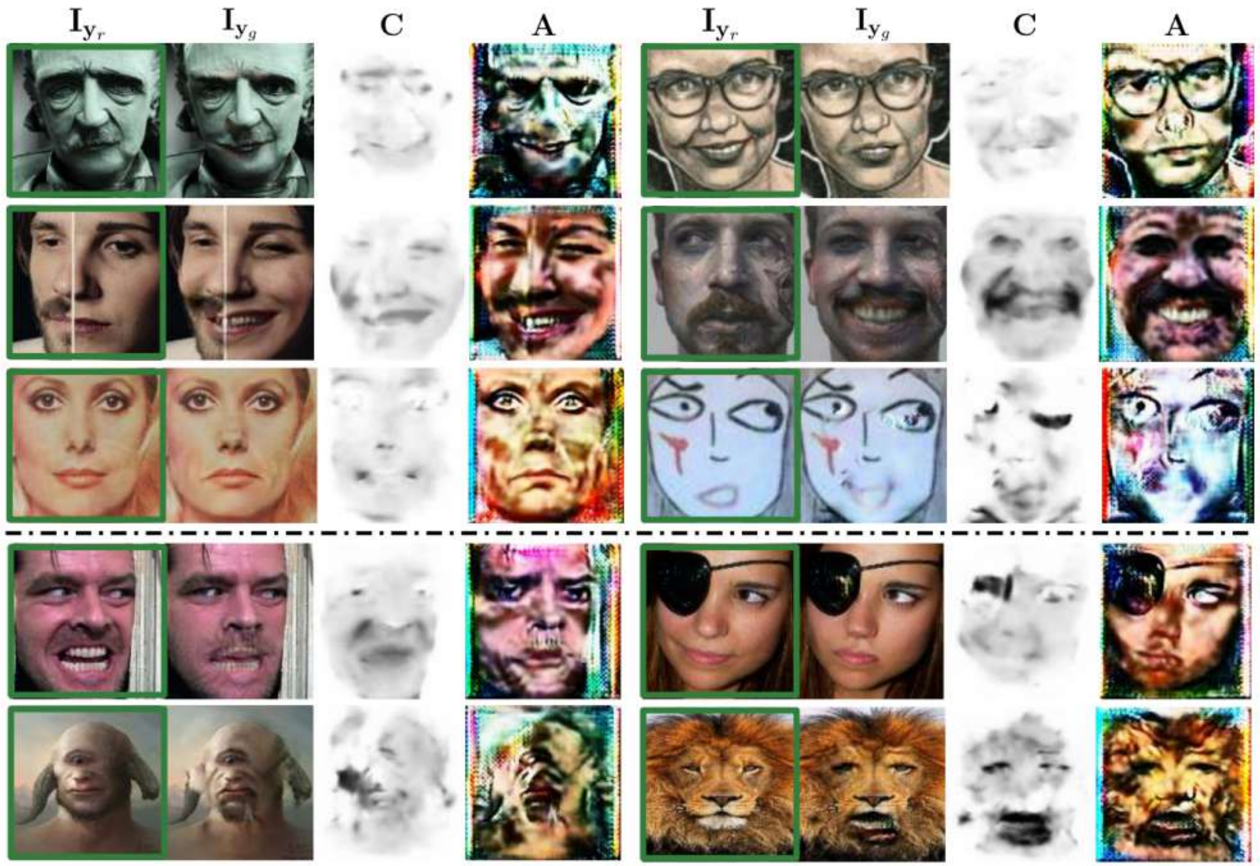

$\mathbf{I}_{\mathbf{y}_r}$
$\mathbf{I}_{\mathbf{y}_g}$
C
A
$\mathbf{I}_{\mathbf{y}_r}$
$\mathbf{I}_{\mathbf{y}_g}$
C
A

Fig. 9. Success and Failure Cases. In all cases, we represent the source image $\mathbf{I}_{\mathbf{y}_r}$, the target one $\mathbf{I}_{\mathbf{y}_g}$, and the color and attention masks $\mathbf{C}$ and $\mathbf{A}$, respectively. **Top:** Some success cases in extreme situations. **Bottom:** Several failure cases.

Note: pretty sure C and A are switched

Left: human-like sculptures
Right: drawing (attention ignores glasses)

Left: robustness to texture across face
Right: robustness to non-real textures

Left: non-standard illumination/colors
Right: face sketch (surprisingly good result)

Left: extreme expression → **A mask** error
Right: eye patch creates attention artifacts

Left: non-human data (cyclops)
Right: get some human face features

In Summary

- Facial deformation in continuous space
  - Unlocks smooth transitions (animations) and video-editing potential
- Attention mask helps ignore distracting/irrelevant features
  - Increases overall sharpness, allows method to work on images in the wild
- Only requires data annotated with AUs (no strong supervision)
- Qualitative evaluation shows impressive, high-resolution results
- *Remaining problems*: occlusion/clutter, multiple faces at once…