

The Creation and Detection of Deepfakes: A Survey

Yisroel Mirsky

*Georgia Institute of Technology
Ben-Gurion University of the Negev
yisroel@post.bgu.ac.il*

Wenke Lee

*Georgia Institute of Technology
wenke@cc.gatech.edu*

Abstract—Generative deep learning algorithms have progressed to a point where it is difficult to tell the difference between what is real and what is fake. In 2018, it was discovered how easy it is to use this technology for unethical and malicious applications, such as the spread of misinformation, impersonation of political leaders, and the defamation of innocent individuals. Since then, these ‘deepfakes’ have advanced significantly.

In this paper, we explore the creation and detection of deepfakes and provide an in-depth view how these architectures work. The purpose of this SoK is to provide the reader with a deeper understanding of (1) how deepfakes are created and detected, (2) the current trends and advancements in this domain, (3) the shortcomings of the current defense solutions, and (4) the areas which require further research and attention.

Index Terms—Deepfakes, Deep fakes, reenactment, replacement, face swap

I. INTRODUCTION

A deepfake is content, generated by an artificial intelligence, that is authentic in the eyes of a human being. The word *deepfake* is a combination of the words ‘*deep learning*’ and ‘*fake*’ and primarily relates to content generated by an artificial neural network, a branch of machine learning.

The most common form of deepfakes involve the generation and manipulation of human imagery. This technology has creative and productive applications. For example, realistic video dubbing of foreign films¹, education through the reanimation of historical figures [1], and virtually trying on clothes while shopping.² There are also numerous online communities devoted to creating deepfake memes for entertainment³, such as music videos portraying the face of actor Nicolas Cage.

However, despite the positive applications of deepfakes, the technology is infamous for its unethical and malicious aspects. At the end of 2017, a Reddit user by the name of ‘deepfakes’ was using deep learning to swap faces of celebrities into pornographic videos, and was posting them online⁴. The discovery caused a media frenzy and a large number of new deepfake videos began to emerge thereafter. In 2018, BuzzFeed released a deepfake video of former president Barak Obama giving a talk on the subject. The video was made using the Reddit user’s software (FakeApp), and raised concerns over identity theft, impersonation, and the spread of misinformation on social media.

Following these events, the subject of deepfakes gained traction in the academic community, and the technology has been rapidly advancing over the last few years. Since 2017,

Samples created by machines to fool...

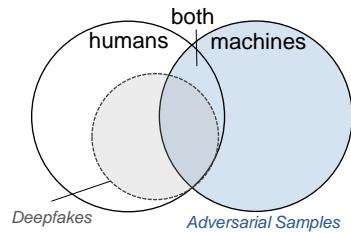


Fig. 1: The difference between *adversarial machine learning* and *deepfakes*.

the number of papers published on the subject rose from 3 to over 150 (2018-19).

To understand where the threats are moving and how to mitigate them, we need a clear view of the technology’s, challenges, limitations, capabilities, and trajectory. Unfortunately, to the best of our knowledge, there are no other works which present the techniques, advancements, and challenges, in a technical and encompassing way. Therefore, the goals of this paper are (1) to provide the reader with an understanding of how modern deepfakes are created and detected, (2) to inform the reader of the recent advances, trends, and challenges in deepfake research, (3) to serve as a guide to the design of deepfake architectures, and (4) to identify the current status of the attacker-defender game, the attacker’s next move, and future work that may help give the defender a leading edge.

We achieve these goals through an overview of human visual deepfakes (Section II), followed by a technical background which identifies technology’s basic building blocks, limitations, and challenges (Section III). We then provide a chronological and systematic review for each category of deepfake, and provide the networks’ schematics to give the reader a deeper understanding of the various approaches (Sections IV and V). Finally, after reviewing the countermeasures (Section VI), we discuss their weaknesses, suggest alternative research, consider the adversary’s next steps, and raise awareness to the spread of deepfakes to other domains (Section VII).

Scope. In this SoK we will focus on deepfakes pertaining to the human face and body. We will not be discussing the synthesis of new faces or the enhancement of facial features because they do not have a clear attack goal associated with them. In Section VII-C we will discuss deepfakes with a much broader scope, note the future trends, and exemplify how deepfakes have spread to other domains and media such as forensics, finance, and healthcare. We note to the reader that deepfakes should not be confused with adversarial machine learning, which is the subject of fooling machine learning algorithms with maliciously crafted inputs (Fig. 1). The difference being that for deepfakes, the objective of the generated content is to

¹<https://variety.com/2019/biz/news/ai-dubbing-david-beckham-multilingual-1203309213/>

²<https://www.forbes.com/sites/forbestechcouncil/2019/05/21/gans-and-deepfakes-could-revolutionize-the-fashion-industry/>

³<https://www.reddit.com/r/SFWdeepfakes/>

⁴https://www.vice.com/en_us/article/gydym/gal-gadot-fake-ai-porn

fool a human and not a machine.

II. OVERVIEW & ATTACK MODELS

We identify four categories of deepfakes relating to human visuals: reenactment, replacement, editing, and synthesis. Although human image editing and synthesis are active research topics, reenactment and replacement deepfakes are the greatest concern [2]–[4] because they give an attacker control over one’s identity. Fig. 2 illustrates some examples of these four categories and their sub-types in the context of faces. Throughout this paper we denote s and t as the source and the target identities. We also denote x_s and x_t as images of these identities and x_g as the deepfake generated from s and t .

A. Reenactment

A reenactment deepfake is where x_s is used to drive the expression, mouth, gaze, pose, or body of x_t :

Expression reenactment is where x_s drives the expression of x_t . It is the most common form of reenactment since these technologies often drive target’s mouth and pose as well, providing a wide range of flexibility. Benign uses are found in the movie and video game industry where the performances of actors are tweaked in post, and in educational media where historical figures are reenacted.

Mouth reenactment, also known as ‘dubbing’, is where the mouth of x_t is driven by that of x_s , or an audio input a_s containing speech. Benign uses of the technology includes realistic voice dubbing into another language and editing.

Gaze reenactment is where direction of x_t ’s eyes, and the position of the eyelids, are driven by those of x_s . This is used to improve photographs or to automatically maintain eye contact during video interviews [5].

Pose reenactment is where the head position of x_t is driven by x_s . This technology has primarily been used for face frontalization of individuals in security footage, and as a means for improving facial recognition software [6].

Body reenactment, a.k.a. pose transfer and human pose synthesis, is similar to the facial reenactments listed above except that’s its the pose of x_t ’s body being driven.

The Attack Model. Reenactment deep fakes give attackers the ability to impersonate an identity, controlling what he or she says or does. This enables an attacker to perform acts of defamation, cause discredibility, spread misinformation, and tamper with evidence. For example, an attacker can impersonate t to gain trust of a colleague, friend, or family member as a means to gain access to money, network infrastructure, or some other asset. An attacker can also generate embarrassing content of t for blackmailing purposes or generate content to affect the public’s opinion of an individual or political leader. Finally, the technology can be used to tamper surveillance footage or some other archival imagery in an attempt to plant false evidence in a trial.

B. Replacement

A replacement deepfake is where the content of x_t is replaced with that of x_s , preserving the identity of s .

Transfer is where the content of x_t is replaced with that of x_s . A common type of transfer is facial transfer, used in the fashion industry to visualize an individual in different outfits.

Swap is where the content transferred to x_t from x_s is driven by x_t . The most popular type of swap replacement is ‘face swap’, often used to generate memes or satirical content

by swapping the identity of an actor with that of a famous individual. Another benign use for face swapping includes the anonymization of one’s identity in public content in-place of blurring or pixelation.

The Attack Model. Replacement deepfakes are well-known for their harmful applications. For example, revenge porn is where an attacker swaps a victim’s face onto the body of a porn actress to humiliate, defame, and blackmail the victim. Face replacement can also be used as a short-cut to fully reenacting t by transferring t ’s face onto the body of a look-alike. This approach has been used as a tool for disseminating political opinions in the past [7].

C. Enhancement

An enchantment deepfake is where the attributes of x_t are added, altered, or removed. Some examples include the changing a target’s clothes, facial hair, age, weight, beauty, and ethnicity. Apps such as FaceApp enable users to alter their appearance for entertainment and easy editing of multimedia. The same process can be used by an attacker to build a false persona for misleading others. For example, a sick leader can be made to look healthy [8], and child or sex predators can change their age and gender to build dynamic profiles online. A known unethical use of enhancement deepfakes is the removal of a victim’s clothes for humiliation or entertainment [9].

D. Synthesis

Synthesis is where the deepfake x_g is created with no target as a basis. Human face and body synthesis techniques such as [10] (used in Fig. 2) can create royalty free stock footage or generate characters for movies and games. However, similar to enhancement deepfakes, it can also be used to create fake personas online.

III. TECHNICAL BACKGROUND

Although there are a wide variety of neural networks, most deepfakes are created using variations or combinations of generative networks and encoder decoder networks. In this section we provide a brief introduction to these networks, how they are trained, and the notations which we will be using throughout the paper.

A. Neural Networks

Neural networks are non-linear models for predicting or generating content based on an input. They are made up of layers of neurons, where each layer is connected sequentially via synapses. The synapses have associated weights which collectively define the concepts learned by the model. To execute a network on an n -dimensional input x , a process known as *forward-propagation* is performed where x propagated through each layer and an activation function is used to summarize a neuron’s output (e.g., the Sigmoid or ReLU function).

To summarize this process, we consider M a black box and denote its execution as $M(x) = y$. To train M in a supervised setting, a dataset of paired samples with the form (x_i, y_i) is obtained and an objective loss function \mathcal{L} is defined. The loss function is used to generate a signal at the output of M which is *back-propagated* through M to find the errors of each weight. An optimization algorithm, such as gradient descent (GD), is then used to update the weights for a number of epochs. The function \mathcal{L} is often a measure of error between the input x and predicted output y' . As a result the network learns the function

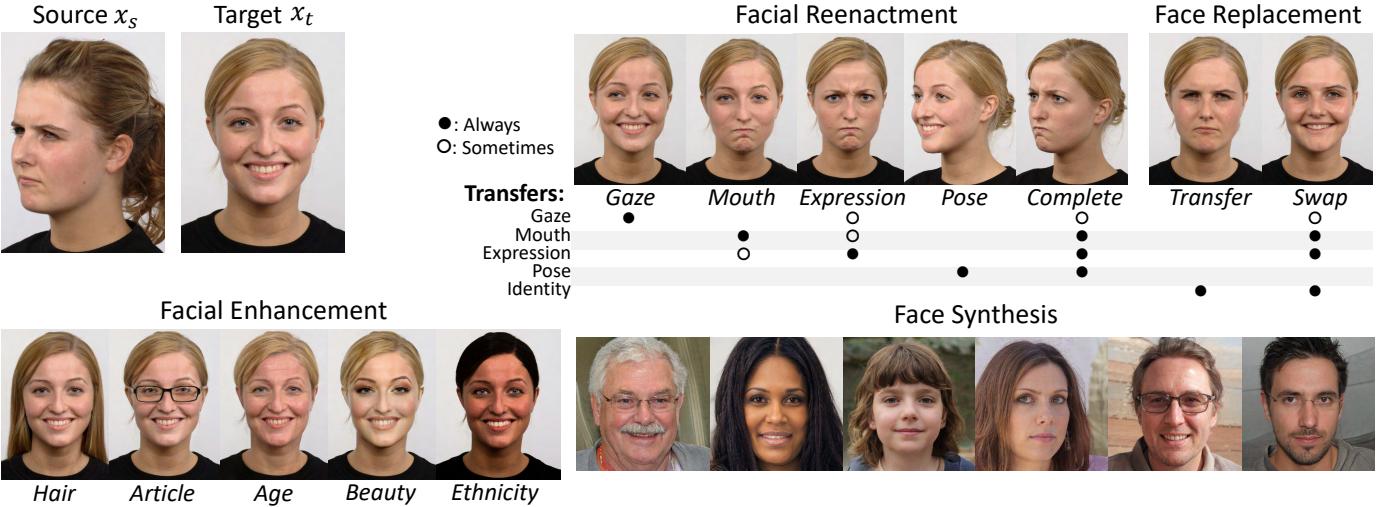


Fig. 2: Examples and illustrations of reenactment, replacement, editing, and synthesis deepfakes of the human face.

$M(x_i) \approx y_i$ and can be used to make predictions on unseen data. Some deepfake networks use a technique called one-shot or few-shot learning which enables a pre-trained network to adapt to a new dataset X' similar to X on which it was trained.

B. Loss Functions

Loss functions must be differentiable in order to use an optimization algorithm, such as GD, to update the weights accordingly. There are various types of loss functions which can be applied in different ways depending on the learning objective. For example, when training a M as an n -class classifier, the output of M would be the probability vector $y \in \mathbb{R}^n$. To train M , we perform *forward-propagation* to obtain $y' = M(x)$, compute the cross-entropy loss (\mathcal{L}_{CE}) by comparing y' to the ground truth label y , and then perform *back-propagation* and to update the weights with the training signal. The loss \mathcal{L}_{CE} over the entire training set X is calculated as

$$\mathcal{L}_{CE} = - \sum_{i=1}^{|X|} \sum_{c=1}^n y_i[c] \log(y'_i[c]) \quad (1)$$

where $y'[c]$ is the predicted probability of x_i belonging to the c -th class.

Other popular loss functions used in deepfake networks include the L1 and L2 norms $\mathcal{L}_1 = |x - x_g|^1$ and $\mathcal{L}_2 = |x - x_g|^2$. However, L1 and L2 require paired images (e.g., of s and t with same expression) and perform poorly when there are large offsets between the images. Another approach to compare two unaligned images is to pass them through another network and measure the difference between the layer's activations (feature maps). This loss is called perceptual loss (\mathcal{L}_{perc}) and, for deepfakes, is often performed using a face recognition network such as VGGFace. Similarly there is a feature matching loss (\mathcal{L}_{FM}) which uses the last output of a network, and content loss ($\mathcal{L}_{Content}$) which passes only x_g and measures the difference between the activations of the network's layers.

C. Generative Neural Networks (for deepfakes)

Deep fakes are often created using combinations or variations of six different networks, five of which are illustrated in Fig. 3.

Encoder-Decoder Networks (ED). An ED consists of at least two networks, an encoder En and decoder De . The ED

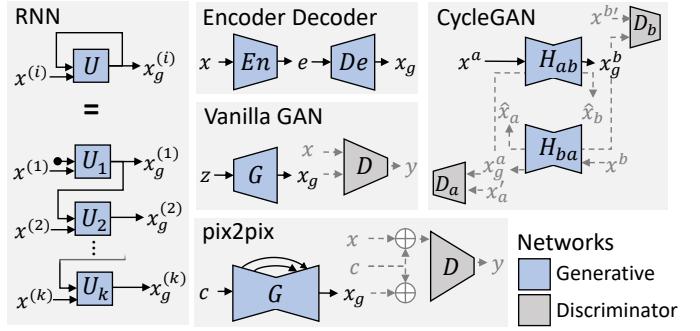


Fig. 3: Five basic neural network architectures used to create deepfakes. The lines indicate dataflows used during deployment (black) and training (grey).

has narrower layers towards its center so that when it's trained as $De(En(x)) = x_g$, the network is forced to summarize the observed concepts. The summary of x , given its distribution X , is $En(x) = e$, often referred to as an encoding or embedding and $E = En(X)$ is referred to as the 'latent space'. Deepfake technologies often use multiple encoders or decoders and manipulate the encodings to influence the output x_g . If an encoder and decoder are symmetrical, and the network is trained with the objective $De(En(x)) = x$, then the network is called an autoencoder and the output is the reconstruction of x denoted \hat{x} . Another special kind of ED is the variational autoencoder (VAE) where the encoder learns the posterior distribution of the decoder given X . VAEs are better at generating content than autoencoders because the concepts in the latent space are disentangled, and thus encodings respond better to interpolation and modification.

Convolutional Neural Network (CNN). In contrast to a fully connected (dense) network, a CNN learns pattern hierarchies in the data and is therefore much more efficient at handling imagery. A convolutional layer in a CNN learns filters which are shifted over the input forming an abstract feature map as the output. Pooling layers are used to reduce the dimensionality as the network gets deeper and up-sampling layers are used to increase it. With convolutional, pooling, and upsampling layers, it is possible to build an ED CNNs for imagery.

Generative Adversarial Networks (GAN) The GAN was

first proposed in 2014 by Goodfellow et al. in [11]. A GANs consist of two neural networks which work against each other: the generator G and the discriminator D . G creates fake samples x_g with the aim of fooling D , and D learns to differentiate between real samples ($x \in X$) and fake samples ($x_g = G(z)$ where $z \sim N$). Concretely, there is an adversarial loss used to train D and G respectively:

$$\mathcal{L}_{adv}(D) = \max \log D(x) + \log(1 - D(G(z))) \quad (2)$$

$$\mathcal{L}_{adv}(G) = \min \log(1D(G(z))) \quad (3)$$

This zero-sum game leads to G learning how to generate samples that are indistinguishable from the original distribution. After training, D is discarded and G is used to generate content. When applied to imagery, this approach produces photo realistic images.

Image-to-Image Translation (pix2pix). Numerous of variations and improvements on GANs have been proposed over the years. One popular version is the pix2pix framework which enables translations from one image domain to another [12]. In pix2pix, G tries to generate the image x_g given a visual context x_c as an input, and D discriminates between (x, x_c) and (x_g, x_c) . Moreover, G is a an ED CNN with skip connections from En to De (called a U-Net) which enables G to produce high fidelity imagery by bypassing the compression layers when needed. Later, pix2pixHD was proposed [13] for generating high resolution imagery with better fidelity.

CycleGAN. An improvement of pix2pix which enables image translation through unpaired training [14]. The network forms a cycle consisting of two GANs used to convert images from one domain to another, and then back again to ensure consistency with a cycle consistency loss (\mathcal{L}_{cyc}).

Recurrent Neural Networks (RNN) An RNN is type of neural network that can handle sequential and variable length data. The network remembers its internal state after processing $x^{(i-1)}$ and can use it to process $x^{(i)}$ and so on. In deepfake creation, RNNs are often used to handle audio and sometimes video. More advanced versions of RNNs include long short-term memory (LSTM) and gate recurrent units (GRU).

D. Feature Representations

Most deep fake architectures use some form of intermediate representation to capture and sometimes manipulate s and t 's facial structure, pose, and expression. One way is to use the facial action coding system (FACS) and measure each of the face's taxonomized action units (AU). Another way is to use monocular reconstruction to obtain a 3D morphable model (3DMM) of the head from a 2D image, where the pose and expression are parameterized by a set of vectors and matrices. Then use the parameters or a 3D rendering of the head itself. Some use a UV map of the head or body to give the network a better understanding of the shape's orientation. Another approach is to use image segmentation to help the network separate the different concepts (face, hair, etc). The most common representation is landmarks (a.k.a. key-points) which are a set of defined positions on the face or body which can be efficiently tracked using open source CV libraries. The landmarks are often presented to the networks as a 2D image with Gaussian points at each landmark. Some works separate the landmarks by channel to make it easier for the network to identify and associate them. Similarly, facial boundaries and body skeletons can also be used. For audio (speech), the most common approach is to measure the Mel-Cepstral Coefficients (MCC) of an audio segment (the dominant voice frequencies).

E. Deepfake Creation Basics

To generate x_g , reenactment and face swap networks follow some variation of this process: Pass x through a pipeline that (1) detects and crops the face, (2) extracts intermediate representations, (3) generates a new face based on some driving signal (e.g., another face), and then (4) blend the generated face into the target frame.

In general there are six approaches to driving an image: (1) The direct approach of letting network work directly on the image and perform the mapping itself, (2) using an ED to disentangle the identity from the expression, and then modify/swap the encodings, (3) add an additional encoding (e.g., expression) before passing it to the decoder, (4) convert the intermediate face/body representation to the desired identity/expression before generation, (5) drive the generation with optical flow field derived from subsequent driving frames in a video, and (6) create composite of the original content (hair, scene, etc) with a combination of the 3D rendering, warped image, or generated content, and pass the composite through another network (such as pix2pix) to refine the realism.

F. Generalization

A deepfake network may be trained or designed to work with only a specific set of target and source identities. An identity agnostic model is sometimes hard to achieve due to correlations learned by the model between s and t during training. We identify three primary relationships in this regard: **one-to-one** $x_g = M_t(E_s(x_s))$, **many-to-one** $x_g = M_t(E(x_s))$, and **many-to-many** $x_g = M(E_1(x_s), E_2(x_t))$, where E is some model or process for representing or extracting features from x .

G. Challenges

The following are some challenges in creating realistic deepfakes:

Generalization. Generative networks are data driven, and therefore reflect the training data in their outputs. This means that high quality images of a specific identity requires a large number of samples of that identity. Moreover, access to a large dataset of the driver is typically much easier to obtain than the victim. As a result, over the last few years, researchers have worked hard to minimize the amount of training data required, and to enable the execution of a trained model on new target and source identities (unseen during training).

Paired Training. One way to train a neural network is to present the desired output to the model for each given input. This process of *data pairing* is a laborious and sometimes impractical when training on multiple identities and actions. To avoid this issue, many deepfake networks either (1) train in a self-supervised manner by using frames selected from the same video of t , (2) use unpaired networks such as Cycle-GAN, or (3) utilize the encodings of an ED network.

Identity Leakage. Sometimes the identity of the driver (e.g., s in reenactment) is partially transferred to x_g . This occurs when training on a single input identity, or when the network is trained on many identities but data pairing is done with the same identity. Some solutions proposed by researchers include attention mechanisms, few-shot learning, disentanglement, boundary conversions, and AdaIN or skip connections to carry the relevant information to the generator.

Occlusions. Occlusions are where part of x_s or x_t is obstructed with a hand, hair, glasses, or any other item. Another

type of obstruction is the eyes and mouth region that may be hidden or dynamically changing. As a result, artifacts appear such as cropped imagery or inconsistent facial features. To mitigate this, works such as [15]–[17] perform segmentation and in-painting on the obstructed areas.

Temporal Coherence. Deepfake videos often produce more obvious artifacts such as flickering and jitter [18]. This is because most deepfake networks process each frame individually with no context of the preceding frames. To mitigate this, some researchers either provide this context to G and D , implement temporal coherence losses, use RNNs, or perform a combination thereof.

IV. REENACTMENT

In this section we present a chronological review of deep learning based reenactment, organized according to their class of identity generalization. Table I provides a summary and systematization of all the works mentioned in this section.

A. Expression Reenactment

Expression reenactment turns an identity into a puppet, giving attackers the most flexibility to achieve their desired impact. Before we review the subject, we note that expression reenactment has been around long before deepfakes were popularized. In 2003, researchers morphed models of 3D scanned heads [52]. In 2005, it was shown how this can be done without a 3D model [53]. Later, between 2015 and 2018, Thies et al. demonstrated how 3D parametric models can be used to achieve high quality and real-time results with depth sensing and ordinary cameras ([54] and [55], [56]).

Regardless, today deep learning approaches are recognized as the simplest way to generate believable content. To help the reader understand the networks and follow the text, we provide network schematics and their loss functions in Fig. 4.

1) **One-to-One (Identity to Identity):** In 2017, the authors of [57] proposed using a CycleGAN for facial reenactment, without the need for data pairing. The two domains where video frames of s and t . However, to avoid artifacts in x_g , the authors note that both domains must share a similar distributions (e.g., poses and expressions). In 2018, Bansal et al. proposed a generic translation network based on CycleGAN called Recycle-GAN [58]. Their framework improves temporal coherence and mitigates artifacts by including next-frame predictor networks for each domain. For facial reenactment, the authors train their network to translate the facial landmarks of x_s into portraits of x_t .

2) **Many-to-One (Multiple Identities to a Single Identity):** In 2017, the authors of [59] proposed CVAE-GAN, a conditional VAE-GAN where the generator is conditioned on an attribute vector or class label. However, reenactment with CVAE-GAN requires manual attribute morphing by interpolating the latent variables (e.g., between target poses). Later, in 2018, a large number of source-identity agnostic models were published, each proposing a different method to decoupling s from t :

Facial Boundary Conversion. One approach was to first convert the structure of source's facial boundaries to that of the target's before passing them through the generator [19]. Their framework 'ReenactGAN' the authors use a CycleGAN to transform the boundary b_s to the target's face shape as b_t before generating x_g with a pix2pix-like generator.

Temporal GANs. To improve the temporal coherence of deepfake videos, the authors of [20] proposed MoCoGAN: a temporal GAN which generates videos while disentangling the motion and content (objects) in the process. Each frame is generated using a target expression label z_c , and a motion embedding $z_M^{(i)}$ for the i -th frame, obtained from a noise seeded RNN. MoCoGAN uses two discriminators, one for realism (per frame) and one for temporal coherence (on the last T frames). In [21], the authors proposed a framework called Vid2Vid, which is similar to pix2pix but for videos. Vid2Vid considers the temporal aspect by generating each frame based on the last L source and generated frames. The model also considers optical flow to perform next-frame occlusion prediction (due to moving objects). Similar to pix2pixHD, a progressive training strategy is to generate high resolution imagery. In their evaluations, the authors demonstrate facial reenactment using the source's facial boundaries. In [22], the authors use a GAN to achieve complete facial reenactment (gaze, blinking, pose, mouth, etc.), training on only one minute of video. Their approach was to extract the source and target's 3D facial models from 2D images using monocular reconstruction, and then for each frame, (1) transfer the facial pose and expression of the source's 3D model to the target's, and (2) produce x_g with a modified pix2pix framework, using the last 11 frames of rendered heads, UV maps, and gaze masks as the input.

3) **Many-to-Many (Multiple IDs to Multiple IDs):** The first attempts at identity agnostic models were made in 2017, where the authors of [60] used a conditional GAN (CGAN) for the task. Their approach was to (1) extract the inner-face regions as (x_t, x_s) , and then (2) pass them an ED to produce x_g subjected to \mathcal{L}_1 and \mathcal{L}_{adv} losses. The challenge of using a CGAN was that the training data had to be paired (images of different identities with the same expression). Going one step further, in [61] the authors reenacted full portraits at low resolutions. Their approach was to decoupling the identities was to use a conditional adversarial autoencoder to disentangle the identity from the expression in the latent space. However, their approach is limited to driving x_t with discreet AU expression labels (fixed expressions) that capture x_s .

Later, in 2018, the authors of [23] proposed GATH which can drive x_t using continuous action units (AU) as an input, extracted from x_s . Their generator is ED network trained on the loss signals from using three other networks: (1) a discriminator, (2) an identity classifier, and (3) a pretrained AU estimator. The classifier shares the same hidden weights as the discriminator to disentangle the identity from the expressions.

Self-Attention Modeling. Similar to [23], another work called GANimation [16] reenacts faces through AU value inputs estimated from x_s . Their architecture uses an AU based generator that uses a self attention model to handle occlusions, and mitigate other artifacts. Furthermore, another network penalizes G with an expression prediction loss, and shares its weights with the discriminator to encourage realistic expressions. Similar to CycleGAN, GANimation uses a cycle consistency loss which eliminates the need for image pairing.

Instead of relying on AU estimations, the authors of [24] propose GANnotation which generates x_g based on (x_t, l_s) , where l_s is the facial landmarks of x_s . GANnotation uses the same self attention model as GANimation, but proposes a novel "triple consistency loss" to minimize artifacts in x_g . The loss teaches the network how to deal with intermediate

TABLE I: Summary of Deep Learning Reenactment Models (Body and Face)

		Reenactment	Retraining for new...	Model	Representation	Model Training	Model Execution	Model Output			
		Mouth Expr. Pose Gaze Body	Source (s)	Target (t)	Identity Agnostic						
					Encoders Decoders Discriminators Other Netw.	AU/AM 3DMM/Rendering UV Mapping Segmentation Landmark / Skeleton Boundary / Action Labelling of: ID No Pairing Pairing within Same Video Pairing ID to Same ID Pairing ID to Diff. Actions Requires Video		Source (x _s ...)	Target (x _t ...)	Image/Frame Video	Resolution
One-to-One	[57] 2017	FT-GAN	• • • o	>20 min. video >20 min. video	2 2 2 0			portrait	portrait	•	128x128
	[58] 2018	Recycle-GAN	• • •	5-10 min. video 5-10 min. video	4 4 2 0			portrait	-	•	512x512
	[87] 2018	DeepFaceLab	• • •	1-3 hr. video 1-3 hr. video	1 2 1 1			portrait video	-	•	512x512
	[39] 2019	Liu et al. 2019	• • •	1-3 hr. video 1-3 hr. video	4 4 2 1 •			upperbody video	-	•	>256x256
Many-to-One	[40] 2017	Syth. Obama	•	None	17 hr. video	0 0 0 1		audio	portait video	•	2048x1024
	[66] 2017	ObamaNet	•	None	17 hr. video	1 1 1 1		text	-	•	256x256
	[22] 2018	Deep Video Portr.	• • •	None	1-3 min. video	1 1 1 0	•	portait video	neural texture	•	1024x1024
	[19] 2018	ReenactGAN	• • •	None	30 min. video	N N N 1		portrait	-	•	256x256
	[21] 2018	Vid2vid	• • o or •	None	3-8 min. video	3 3 2 1		portait video	-	•	2048x1024
	[20] 2018	MocoGAN	• • or •	None	1 min. video	2 1 2 N		expression label	identity label	•	64x64
	[42] 2018	SD-CGAN	• o o	None	2 hr. video	0 1 1 1		audio	-	•	128x128
	[76] 2019	GRN	•	None	3-10 images	o 3 1 0 2		gaze	3-10 eye images	•	64x128
	[41] 2019	TETH	•	None	1 hr. video	1 1 2 0	•	text	portait video	•	512x512
	[43] 2019	N.V. Puppetry	•	None	2-3 min. video	3 2 2 4	•	portait video	portait video	•	512x512
	[49] 2019	NRR-HAV	•	None	8 min. video	1 1 1 0		body image	background	•	512x512
	[50] 2019	Deep Video P.C.	• • •	None	2 min. video	0 1 2 2		body image	-	•	256x256
	[48] 2019	Everybody D. N.	• • •	None	20 min. video	0 2 4 2		body image	-	•	256x256
	[77] 2019	D. D. Generation	• • •	None	3 min. video	2 2 2 2		body video	-	• o	512x512
	[37] 2019	N. Talking Heads	• • •	None	1-3 portraits	o 1 2 1 1		portrait/landmarks	1-3 portraits	•	256x256
	[64] 2019	Few-shot Vid2Vid	• • o or •	None	1-10 portraits	o 3 3 2 4		portrait/body video	1-10 portr./bodies	•	2048x1024
Many-to-Many	[69] 2015	Shimba et al.	• o	None	None	• 0 0 0 1 •		audio	face database	•	*
	[75] 2016	DeepWarp	•	None	None	• 0 0 0 2		gaze	eye image	•	>40x50
	[59] 2017	VAE-GAN	• • •	None	None	• 1 1 1 1		latent variables	portrait	•	>128x128
	[60] 2017	RDFT	• • •	None	None	• 1 1 1 0	•	portrait	portrait	•	256x256
	[61] 2017	FE-CDAE	•	None	None	• 1 1 2 0 •		portrait	AU label	•	32x32
	[27] 2018	paGAN	•	None	None	• 1 1 1 1	•	portrait	portrait - neutral	•	512x512
	[28] 2018	X2Face	• • •	None	None	• 2 2 0 1		portrait	1-3 portraits	•	256x256
	[24] 2018	GANnotation	• • •	None	None	• 1 1 1 3		portrait/landmarks	portrait	•	128x128
	[23] 2018	GATH	• • •	None	None	• 1 1 2 0 •		portrait/AUs	portrait	•	100x100
	[25] 2018	FaceID-GAN	• • •	None	None	• 1 1 2 1	•	portrait	portait	•	128x128
	[26] 2018	FaceFeat-GAN	• • •	None	None	• 1 1 3 4		latent variables	portrait	•	128x128
	[73] 2018	CAPG-GAN	•	None	None	• 1 1 2 1		portrait	portrait	•	128x128
	[6] 2018	DR-GAN	•	None	None	• 1 1 1 0		pose	1+ portraits	•	96x96
	[78] 2018	Deformable GAN	• •	None	None	• 1 1 1 0		body image/landm.	body image	•	256x256
	[82] 2018	SHUP	• • •	None	None	• 3 3 1 1		body image	body image/pose	•	256x256
	[83] 2018	DPIG	• • •	None	None	• 4 2 1 0		body image	body image	•	128x64
	[80] 2018	Dense Pose Tr.	• • •	None	None	• 25 25 1 2	•	body image	body image	•	256x256
	[70] 2018	Song et al.	• o	None	None	• 2 1 3 0		audio	portrait	•	128x128
	[33] 2019	wg-GAN	•	None	None	• 2 2 3 0		portrait	portrait	•	256x256
	[15] 2019	FSGAN	• • •	None	None	• 1 1 1 1	•	portrait/landmarks	portrait	•	256x256
	[16] 2019	GANimation	• • •	None	None	• 2 2 1 1 •		portrait/AUs	portrait	•	128x128
	[31] 2019	ICFace	• • •	None	None	• 2 2 1 2 •		portrait/AUs	portrait	•	128x128
	[29] 2019	FaceSwapNet	• • •	None	None	• 4 2 1 0		portrait/landmarks	portrait/landmarks	•	256x256
	[36] 2019	Monkey-Net	• • •	or •	None	• 3 3 1 0		portrait/body	portrait/body	•	64x64
	[17] 2019	First-Order-Model	• • •	or •	None	• 3 3 1 1		portrait/body	portrait/body	•	256x256
	[34] 2019	M&T GAN	• • •	None	None	• 2 1 2 1		expression label	portrait	•	64x64
	[32] 2019	AF-VAE	• • •	None	None	• 2 1 0 1		portrait/boundaries	portrait	•	256x256
	[30] 2019	Fu et al. 2019	• • •	None	None	• 3 2 3 4		portrait/label	portrait	•	1024x1024
	[63] 2019	FusionNet	• • •	None	None	• 1 2 3 3		portrait/landmarks	portrait	•	256x256
	[74] 2019	AD-GAN	• • •	None	None	• 2 2 2 1	•	pose	portrait	•	128x128
	[18] 2019	Speech D. Anm. 1	2 o o o	None	None	• 5 1 2 3		audio	portrait	•	96x128
	[47] 2019	Speech D. Anm. 2	2 o o o	None	None	• 5 1 3 3		audio	portrait	•	96x128
	[72] 2019	Speech D. Anm. 3	2 o o o	None	None	• 5 1 3 3		audio	portrait	•	96x128
	[45] 2019	DAVS	• •	None	None	• 3 1 1 4		audio/portrait video	portrait	o	256x256
	[44] 2019	ATVGnet	• •	None	None	• 1 0 1 5		audio	portait video	•	128x128
	[46] 2019	Speech2Vid	• •	None	None	• 3 1 0 2		audio	portait video	•	109x109
	[51] 2019	DwNet	• • •	None	None	• 2 1 1 3	•	body video	body image	•	256x256
	[84] 2019	C-DGPose	• • •	None	None	• 2 1 1 0		body image	body/pose image	•	64x64
	[79] 2019	PPAT-PIG	• • •	None	None	• 2 1 2 1		body image	body/pose image	•	256x256
	[35] 2020	ImaGINator	• • •	None	None	• 1 1 2 0		expression label	portrait	•	64x64
	[38] 2020	MarioNETte	• • •	None	None	• 2 2 1 3		portrait	1-8 portraits	•	256x256

poses/expressions not found in the training set. Given l_s , l_t and l_z sampled randomly from the same video, the loss is computed as $\mathcal{L}_{trip} = \|G(x_t, l_s) - G(G(x_t, l_z), l_s)\|^2$

3D Parametric Approaches. Concurrent to the work of [22], other works also leveraged 3D parametric facial models to prevent identity leakage in the generation process. In [25], the authors propose FaceID-GAN which can reenacts t at oblique poses and high resolution. Their ED generator is trained in tandem with a 3DMM face model predictor, where the model parameters of x_t are used to transform x_s before being joined with the encoder's embedding. Furthermore, to prevent identity leakage from x_s to x_g , FaceID-GAN incorporates an identification classifier within the adversarial game. The classifier

has $2N$ outputs where the first N outputs (corresponding to training set identities) are activated if the input is real and the rest are activated if it's fake. Later, the authors of [25] proposed FaceFeat-GAN which improves the diversity of the faces while preserving the identity [26]. The approach is to use a set of GANs to learn facial feature distributions as encodings, and then use these generators to create new content with a decoder. However, the featurer GAN's input seed z be selected empirically to fit x_s . In [27], the authors present paGAN, a method for complete facial reenactment on a 3D avatar, using a single image of the target as input. An expression neutral image of x_t is used to generate a 3D model which is then

driven by x_s . The driven model is used to create inputs for a U-Net generator: the rendered head, its UV map, its depth map, a masked image of x_t for texture, and a 2D mask indicating the gaze of x_s .

Using Multi-Modal Sources. In [28] the authors propose X2Face which can reenact x_t with x_s or some other modality such as audio or a pose vector. X2Face uses two ED networks: an embedding network and a driving network. First the embedding network encodes 1-3 examples of the target’s face to v_t : the optical flow field required to transform x_t to a neutral pose and expression. Next, x_t is interpolated according to m_t producing x_t' . Finally, the driving network maps x_s to the vector map v_s , crafted to interpolate x_t' to x_g , having the pose and expression of x_s .

In 2019, nearly all works pursued identity agnostic models:

Facial Landmark & Boundary Conversion. In [29], the authors propose FaceSwapNet which tries to mitigate the issue of identity leakage from facial landmarks. First two encoders and a decoder are used to transfer the expression in landmark l_s to the face structure of l_t , denoted l_g . Then a generator network is used to convert x_t to x_g where l_g is injected into the network with AdaIn layers like a Style-GAN. The authors found that it is crucial to use triplet perceptual loss with an external VGG network. In [15], the authors present FSGAN: a face swapping and facial reenactment model which can handle occlusions. For reenactment a pix2pixHD generator receives x_t and the source’s 3D facial landmarks l_s , represented as a 256x256x70 image (one channel for each of the 70 landmarks). The output is x_g and its segmentation map m_g with three channels (background, face, and hair). The generator is trained recurrently where each output is passed back as input for several iterations while l_s is interpolated incrementally from l_s to l_t . To improve results further, delaunay Triangulation and barycentric coordinate interpolation are used to generate content similar to the target’s pose. In their evaluations they achieve real time reenactment at 30fps. In [30], the authors propose a method for high resolution reenactment and at oblique angles. A set of networks encode the source’s pose, expression, and the target’s facial boundary for a decoder that generates the reenacted boundary b_g . Finally, an ED network generates x_g using an encoding of x_t ’s texture in its embedding. A multi-scale loss is used to improve quality and the authors utilize a small labeled dataset by training their model in a semi-supervised way.

Latent Space Manipulation. In [31], the authors present a model called ICFace where the expression, pose, mouth, eye, and eyebrows of x_t can be driven independently. Their architecture is similar to a CycleGAN in that one generator translates x_t into a neutral expression domain as x_t^η and another generator translates x_t^η into an expression domain as x_g . Both generators are conditioned on the target AU. In [32] the authors propose an Additive Focal Variational Auto-encoder (AF-VAE) for high quality reenactment. This is accomplished by separating a C-VAE’s latent code into an appearance encoding e_a and identity-agnostic expression coding e_x . To capture a wide variety of factors in e_a (e.g., age, illumination, complexion, ...), the authors use an additive memory module during training which conditions the latent variables on a Gaussian mixture model, fitted to clustered set of facial boundaries. Subpixel convolutions were used in the decoder to mitigate artifacts and improve fidelity.

Warp-based Approaches. In the past, facial reenactment

was done by warping the image x_t to the landmarks l_s [62]. In [33], the authors propose wgGAN which uses the same approach but creates high-fidelity facial expressions by refining the image through a series of GANs: one for refining the warped face and another for in-painting the occlusions (eyes and mouth). A challenge with wgGAN is that the warping process is sensitive to head motion (change in pose). In [63], the authors propose a system to control both expression and gaze: a decoder generates x_g with an encoding of x_t as the input and a segmentation map of x_s as reenactment guidance via SPADE residual blocks. The authors blend x_g with a warped version, guided by the segmentation, to mitigate artifacts in the background.

Motion-Content Disentanglement. In [34] the authors propose a GAN to reenact neutral expression faces with smooth animations. The authors describe the animations as temporal curves in 2D space, summarized as points on a spherical manifold by calculating their square-root velocity function (SRVF). A WGAN is used to complete this distribution given target expression labels, and a pix2pix GAN is used to convert the sequences of reconstructed landmarks into a video frames of the target. [35] In contrast to MoCoGAN [20], the authors of [35] propose ImaGINator: a conditional GAN which fuses both motion and content, but with transposed 3D convolutions to capture the distinct spatio-temporal relationships. The GAN also uses a temporal discriminator, and to increase diversity, the authors train the temporal discriminator with some videos using the wrong label. A disadvantage of [34] and [35] is that they are label driven and produce videos with a set number of frames. In [36], the authors propose Monkey-Net: a self supervised network for driving an image with an arbitrary video sequence. Similar to MoCoGAN [20], the authors decouple the source’s content and motion. First a series of networks produce a motion heat map (optical flow) using the source and target’s keypoints, and then an ED generator produces x_g using x_s and the optical flow (in its embedding). Later in [17], the authors extend Monkey-Net by improving the object appearance when large pose transformations occur. They accomplish this by (1) modeling motion around the keypoints using affine transformations, (2) updating the key-point loss function accordingly, and (3) having the motion generator predict an occlusion mask on the preceding frame for in-painting inference.

4) **Few-Shot Learning:** Towards the end of 2019 and into the beginning of 2020, researchers began looking to minimizing the amount of training data via one-shot and few-shot learning. In [37], the authors propose a few-shot model which works well at oblique angles. To accomplish this, the authors perform meta-transfer learning, where the network is first trained on many different identities and then fine-tuned on the target’s identity. First, an identity encoding of x_t is obtained by averaging k sets of (x_t, l_t) . Then a pix2pix GAN is used to generate x_g using l_s as an input, and the identity encoding via AdaIN layers. Unfortunately, the authors note that their method is sensitive to identity leakage. In [64] the authors of Vid2Vid (Section IV-A2) extend their work with few-shot learning. They use a network weight generation module which utilizes an attention mechanism. The module learns to extract appearance patterns from a few sample x_t which are injected into the video synthesis layers. In [38], the authors propose MarioNETe which alleviates identity leakage when the pose of x_s is different than x_t . In contrast to other works which encode the identity separately or use of AdaIN

layers, the authors use an image attention block and target feature alignment. This enables the model to better handle the differences between face structures. Finally, the identity is also preserved using a novel landmark transformer inspired by [65].

B. Mouth Reenactment (Dubbing)

In contrast to expression reenactment, mouth reenactment (a.k.a., video or image dubbing) is concerned with driving a target’s mouth with a segment of audio. Fig. 6 presents the relevant schematics for this section.

1) *Many-to-One (Multiple Identities to a Single Identity):*

Obama Puppetry. In 2017, the authors of [40] created a realistic reenactment of former president Obama. This was accomplished by (1) using a time delayed RNN over MFCC audio segments to generate a sequence of mouth landmarks (shapes), (2) generating the mouth textures (nose and mouth) by applying a weighted median to images with similar mouth shapes via PCA-space similarity, (3) refining the teeth by transferring the high frequency details other frames in the target video, and (4) by using dynamic programming to re-time the target video to match the source audio and blend the texture in. Later that year, the authors of [66] presented ObamaNet: a network that reenacts an individual’s mouth and voice using text as input. The process is to (1) convert the source text to audio using Char2Wav [67], (2) generate a sequence of mouth-keypoints using a time-delayed LSTM on the audio, and (3) use a U-Net CNN to perform in-painting on a composite of the target video frame with a masked mouth and overlayed keypoints. Later in 2018, Jalalifar et al. [42] proposed a network that synthesizes the entire head portrait of Obama, and therefore does not require pose re-timing like [40], [66]. First, a bidirectional LSTM converts MFCC audio segments into sequence of mouth landmarks, and then a pix2pix like network generates frames using the landmarks and a noise signal. After training, the pix2pix network is fine-tuned using a single video of the target to ensure consistent textures.

3D Parametric Approaches. Later on in 2019, the authors of [41] proposed a method for editing a transcript of a talking heads which, in turn, modifies the target’s mouth and speech accordingly. The approach is to (1) align phonemes to a_s , (2) fit a 3D parametric head model to each frame of X_t like [22], (3) blend matching phonemes to create any new audio content, (4) animate the head model with the respective frames used during the blending process, and (5) generate X_g with a CGAN RNN using composites as inputs (rendered mouths placed over the original frame). The authors of [43] had a different approach: (1) animate a the reconstructed 3D head with the predicted blend shape parameters from a_s using a DeepSpeech model for feature extraction, (2) use Deferred Neural Rendering [68] to generate the mouth region, and then (3) use a network to blend the mouth into the original frame. The authors found that their approach only requires 2-3 minutes of video of the target to produce very realistic results.

2) *Many-to-Many (Multiple IDs to Multiple IDs):* One of the first works to perform identity agnostic video dubbing was [69]. There the authors used an LSTM to map MFCC audio segments to the face shape. The face shapes were represented as the coefficients of an active appearance model (AAM), which were then used to retrieve the correct face shape of the target.

Improvements in Lip-sync. Noting a human’s sensitivity to temporal coherence, the authors of [70] use a GAN with

three discriminators: on the frames, video, and lip-sync. Frames are generated by (1) encoding each MFCC audio segment $a_s^{(i)}$ and x_t with separate encoders, (2) passing the encodings through an RNN, and (3) decoding the outputs as $x_g^{(i)}$ using a decoder. In [71] the authors try to improve lip sync with textual context. A time-delayed LSTM is used to predict mouth landmarks given MFCC segments and the spoken text using a text-to-speech model. The target frames are then converted into sketches using an edge filter and the predicted mouth shapes are composited into them. Finally, a pix2pix like GAN with self-attention is used to generate the frames with both video and image conditional discriminators. The authors of [44] state that direct models are problematic since the model learns irrelevant correlations between the audiovisual signal and the speech content. To avoid this issue, an LSTM audio-to-landmark network and a landmark-to-identity CNN-RNN are used in sequence, where the facial landmarks are compressed with PCA and the attention mechanism from [16] is used. To improve synchronization, the authors propose a regression based discriminator which considers both sequence and content information.

EDs for Preventing Identity Leakage. The authors in [45] mitigate identity leakage by disentangling the speech and identity latent spaces using adversarial classifiers. Since their speech encoder is trained to project audio and video into the same latent space, the authors show how x_g can be driven using x_s or a_s . In [46], the authors propose Speech2Vid which also uses separate encoders for audio and identity. However, to capture the identity better, the identity encoder En_I uses a concatenation of five images of the target, and there are skip connections from the En_I to the decoder. To blend the mouth in better, a third ‘context’ encoder is used to encourage in-painting. Finally, a VDSR CNN is applied to x_g to sharpen the image. In [18], the authors propose a dubbing network which also generates facial expressions and blinking. The approach is to generate frames with a stride transposed CNN decoder on GRU-generated noise, in addition to the audio and identity encodings. Their video discriminator uses two RNNs for both the audio and video. Later in [47], the same authors improve the temporal coherence by splitting the video discriminator into two: (1) for temporal realism in mouth to audio synchronization, and (2) for temporal realism in overall facial expressions. Then in [72], the authors tune their approach further by fusing the encodings (audio, identity, and noise) with a polynomial fusion layer as opposed to simply concatenating the encodings together. Doing so makes the network less sensitive to large facial motions compared to [47] and [46].

C. Pose Reenactment

Most deep learning works in this domain focus on the problem of face frontalization. However, there are some works which focus on facial pose reenactment.

In [73] the authors use a U-Net to convert (x_t, l_t, l_s) into x_g using a GAN with two discriminators: one conditioned with the neutral pose image, and the other conditioned with the landmarks. In [6], the authors propose DR-GAN for pose-invariant face recognition. To adjust the pose of x_t , the authors use an ED GAN which encodes x_t as e_t , and then decodes (e_t, p_s, z) as x_g , where p_s is the sources pose vector and z is a noise vector. The authors show that the quality of x_g can be improved by averaging multiple examples of the identity encoding before passing it through the decoder. In [74], the authors suggest using two GANs: The first frontalizes the face

and produces a UV map, and second rotates the face, given the target angle as an injected embedding.

D. Gaze Reenactment

There are only a few deep learning works which have focused on gaze reenactment. In [75] the authors convert a cropped eye x_t , its landmarks, and the source angle, to a flow (vector) field using a 2-scale CNN. x_g is then generated by applying a flow field to x_t to warping it to the source angle. The authors then correct the illumination of x_g with a second CNN. In [76], the authors propose the Gaze Redireciton Network (GRN). The target’s cropped eye, head pose, and source angle are encoded separately and then passed through an ED network to generate an optical flow field. The field is used to warp x_t into x_g . To overcome the lack of training data and the challenge of data pairing, the authors (1) pre-train their network on 3D synthesized examples, (2) further tune their network on real images, and then (3) fine tune their network on 3-10 examples of the target.

E. Body Reenactment

Several facial reenactment papers from Section IV-A discuss body reenactment too. For example, Vid2Vid [21], [64], MocoGAN [20], and others [17], [36]. In this section, we focus on methods which specifically target body reenactment.

1) **One-to-One (Identity to Identity):** In the work [39], the authors perform facial reenactment with the upper-body as well (arms and hands). The approach is to (1) use a pix2pixHD GAN to convert the source’s facial boundaries to the targets, (2) and then paste them onto a captured pose skeleton of the source, and (3) use a pix2pixHD GAN to generate x_g from the composite.

2) **Many-to-One (Multiple Identities to a Single Identity):** **Dance Reenactment.** In [48] the authors make people dance using a target specific pix2pixHD GAN with a custom loss function. The generator receives an image of the captured pose skeleton and the discriminator receives the current and last image conditioned on their poses. The quality of face is then improved with a residual predicted by an additional pix2pixHD GAN, given the face region of the pose. A many-to-one relationship is achieved by normalizing the input pose to that of the target’s. The authors of [49] then tried to overcome artifacts which occur in [48] such stretched limbs due to incorrectly detected pose skeletons. They used photogrammetry software on hundreds of images of the target, and then reenacted the 3D rendering of the target’s body. The rendering, partitioned depth map, and background are then passed to a pix2pix model for image generation, using an attention loss. The authors of [50] try to generalize to unseen poses by training a network on many identities other than s and t . This was accomplished by first training the GAN on paired data (the same identity doing different poses) and then later adding another discriminator that evaluates the temporal coherence given (1) $x_g^{(i)}$ driven by another video, and (2) the optical flow predicted version. In [77] the authors were able to reenact a target with only a 3 minutes of video. First the target’s body was segmented by limb and the pieces are oriented to the source’s pose. Then a pix2pixHD GAN uses this composition and the last k frames’ poses to generate the body. Finally, another pix2pixHD GAN is used to blend the body into the background.

3) **Many-to-Many (Multiple IDs to Multiple IDs): Pose Alignment.** In [78] the authors try to resolve the issue of misalignment when using pix2pix like architectures. They propose ‘deformable skip connections’ which help orient the shuttled feature maps according to the source pose. The authors also propose a novel nearest neighbor loss instead of using L1 or L2 losses. To modify unseen identities at test time, an encoding of x_t is passed to the decoder’s inner layers. The authors of [79] propose using a GAN that uses novel Pose-Attentional Transfer blocks (PATB) inside the generator. The architecture passes x_t and the poses p_s concatenated with p_t through separate encoders which are passed though a series of PATBs before being decoded. The PATBs progressively transfer regional information of the poses to regions of the image to ultimately create a body that has better shape and appearance consistency.

Pose Warping. In [80] the authors use a pre-trained DensePose network [81] to refine a predicted pose with a warped and in-painted DensePose UV spatial map of the target. Since the spatial map covers all surfaces of the body, the generated image has improved texture consistency. In [51], the authors propose DwNet which helps alleviate misalignment between x_s and x_t via warping. The authors use a ‘warp module’ in an ED network to encode $x_t^{(i-1)}$ warped to $p_s^{(i)}$, where p is a UV body map of a pose obtained a DensePose network.

Background Foreground Compositing. In [82], the authors break the process down into three stages, trained end-to-end: (1) use a U-Net to segment x_t ’s body parts and then orient them according to the source pose p_s , (2) use a second U-Net to generate the body x_g from the composite, and (3) use a third U-Net to perform in-painting on the background and paste x_g into it. The authors of [83] use a ED GAN network to disentangle the foreground appearance (body), background appearance, and pose to gain control each of these aspects. This is accomplished by segmenting each of these aspects before passing them through encoders. In [84] the authors use a CVAE-GAN, conditioned on heatmaps of the detected pose and skeleton. As a result, the authors were able to change the pose and appearance of bodies individually.

4) **Few-Shot Learning:** In [85], the authors demonstrate the few-shot learning technique of [86] on a pix2pixHD network and the network of [82]. Using just a few sample images, they were able to transfer the resemblance of a target to new videos in the wild.

V. REPLACEMENT

The network schematics and summary of works for this Section can be found in Fig. 7 and Table II respectively.

A. Swap

At first, face swapping was a manual process accomplished using tools such as Photoshop. More automated systems first appeared between 2004-08 in [98] and [99]. Later, fully automated methods were proposed in [100], [101] and [102] using methods such as warping and reconstructed 3D morphable face models.

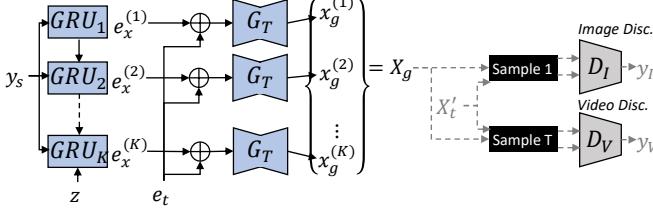
1) **One-to-One (Identity to Identity):**

Online Communities. After the Reddit user ‘deepfakes’ was exposed in the media, researchers and online communities began finding improved ways to perform face swapping with deep neural networks. The original deepfake network, published by the Reddit user, is an ED network. The architecture consists of one encoder En and two decoders De_s and De_t .

x_s, x_t, x_g : The source, target, and generated images (e.g., portraits)
 y : A label (e.g., fake vs real, one-hot encoding, ...)
 x' : Another sample from the same distribution, \hat{x} : reconstructed
 m : Binary mask, s : Segmentation map, l : Landmark or Keypoint, z : Noise
 \oplus : Concatenate, \ominus : Subtract, \odot : Multiply, \circledast : Add, \odot : Paste content
 \cdot_a : Crop out region a from image where $a \in \{f\text{face}, e\text{eye}, m\text{mouth}\}$
 \circ_a : Create mask using region a of the image where $a \in \{f\text{face}, e\text{eye}, m\text{mouth}\}$
 $x^{(a)}$: Image x cropped to the region of $a \in \{f\text{face}, e\text{eye}, m\text{mouth}\}$
 \circledcirc : Spatial replication of a vector (channel-wise or dim-wise)
 $\times \odot$: Scale image down by factor of X
 $LE, BE, AE, 3DE$: Landmark, Boundary, Action Unit (AU),
and 3DMM facial model Extractors (open source CV library)
 $LT, 3DT$: Landmark and 3D model transformers, from s to t
 ME : MFCC audio feature extractor

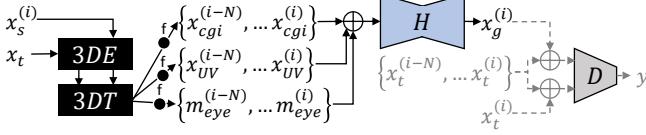
[20] MocoGAN: $\mathcal{L}_{adv}, \mathcal{L}_*$

y_s : source expression label, e_t : one-hot encoding of target identity,
 e_x : temporal expression embedding, GRU: Gated Recurrent Unit of an RNN



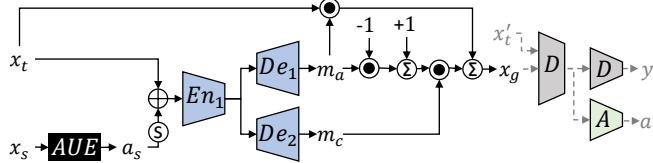
[22] Deep Video Portrait: $\mathcal{L}_{adv}, \mathcal{L}_1$

m_{eye} : mask of eye region (gaze), x_{UV} : UV correspondence map, x_{cgi} : 3D rendered image of x

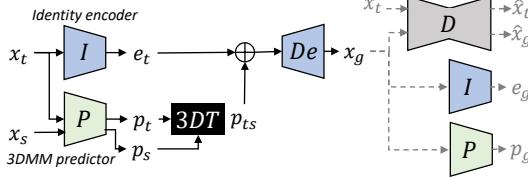


[16] GANimation: $\mathcal{L}_{adv}, \mathcal{L}_{con}, \mathcal{L}_{att}, \mathcal{L}_{tv}, \mathcal{L}_*$

m_a : attention mask, m_c : color mask

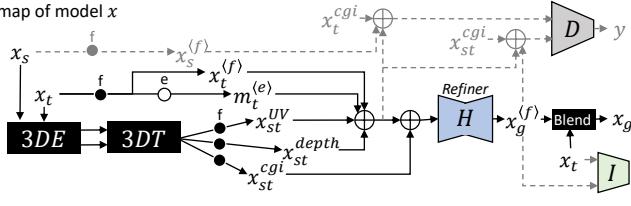


[25] FaceID-GAN: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{CE}, \mathcal{L}_{cos}, \mathcal{L}_*$



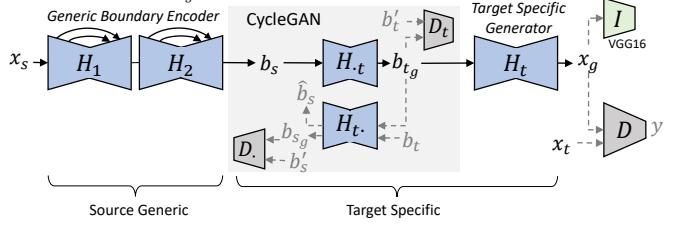
[27] paGAN: $\mathcal{L}_{adv}, \mathcal{L}_1$

x^{UV} : UV correspondence map, x^{cgi} : 3D rendered image of x , x^{depth} : image of depth map of model x



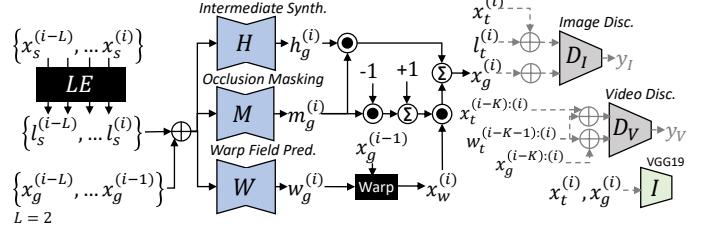
[19] Reenact GAN: $\mathcal{L}_{tv}, \mathcal{L}_2, \mathcal{L}_{cons}, \mathcal{L}_{perc}, \mathcal{L}_{adv}$

b : facial boundaries, b_{x_g} : a boundary translated to domain x

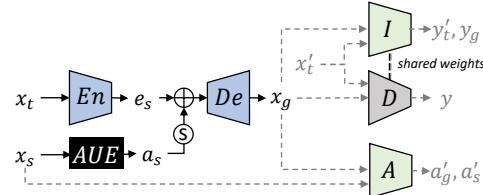


[21] Vid2Vid: $\mathcal{L}_{adv}, \mathcal{L}_{FM}, \mathcal{L}_*$

T : frames in the video clip, L, K : system parameters

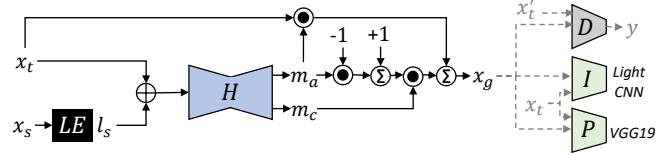


[23] GATH: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{CE}, \mathcal{L}_{tv}, \mathcal{L}_*$



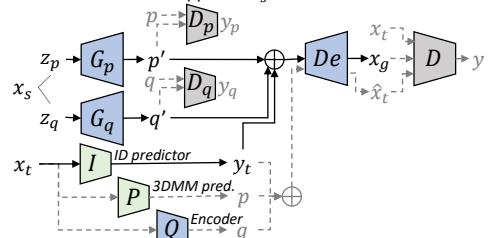
[24] GANotation: $\mathcal{L}_{adv}, \mathcal{L}_{adv}, \mathcal{L}_2, \mathcal{L}_{con}, \mathcal{L}_{FM}, \mathcal{L}_{per}, \mathcal{L}_{hinge}, \mathcal{L}_{tv}$

m_a : attention mask, m_c : color map, training: s and t have same ID



[26] FaceFeat-GAN: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_{CE}, \mathcal{L}_*$

z : Sample of random noise later mapped to x_s



[28] X2Face: $\mathcal{L}_1, \mathcal{L}_{perc}$

v : vector map of pixel deltas (changes), x^η : a face with a neutral expression/pose,
 a : some other modality (e.g., audio)

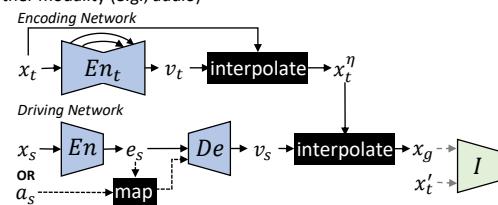
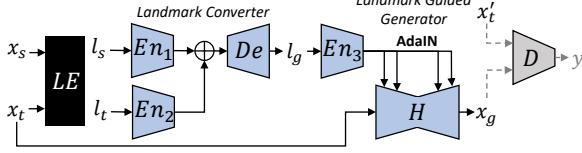


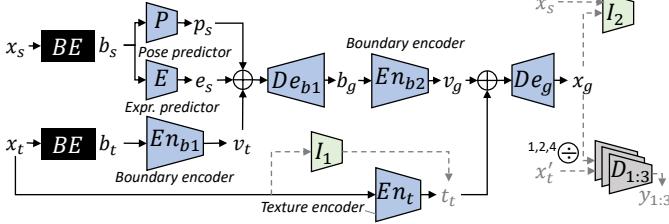
Fig. 4: Architectural schematics of **reenactment networks**. Black lines indicate prediction flows used during deployment, dashed gray lines indicate dataflows performed during training. Zoom in for more detail.

Losses: \mathcal{L}_1 : L1, \mathcal{L}_2 : L2, \mathcal{L}_{CE} : Cross Entropy, \mathcal{L}_{adv} : Adversarial, \mathcal{L}_{MS} : Multi Scale, \mathcal{L}_{FM} : Feature Matching, \mathcal{L}_{per} : Perceptual, \mathcal{L}_{cyc} : Cycle Consistency, \mathcal{L}_{atten} : Attention, \mathcal{L}_{hinge} : Hinge, \mathcal{L}_{trip} : Triplet, \mathcal{L}_{tv} : Total Variance, \mathcal{L}_{KL} : KL Divergence, \mathcal{L}_* : Other

[29] FaceSwapNet: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_{perc}, \mathcal{L}_{trip}, \mathcal{L}_*$

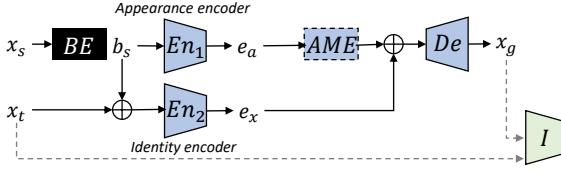


[30] Fu et al. 2019: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{MS}, \mathcal{L}_{FM}, \mathcal{L}_{perc}, \mathcal{L}_*$



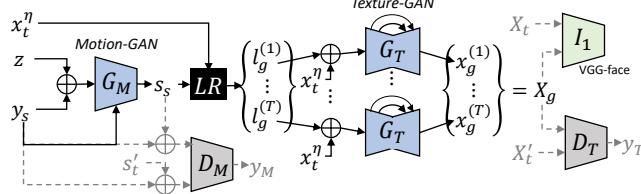
[32] AF-VAE: $\mathcal{L}_{FM}, \mathcal{L}_{KL}, \mathcal{L}_*$

AME: Additive Memory Encoder – models e_a as a Gaussian mixture of clustered facial boundaries.



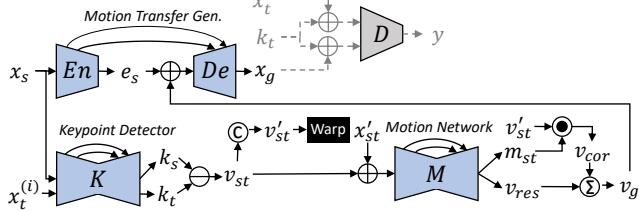
[34] Motion&Texture-GAN: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{perc}, \mathcal{L}_*$

x^η : cropped neutral expression face, y_s : face expression label of source, s : an SRVF point on a spherical manifold, LR : landmark reconstruction from s , l : facial landmarks



[36] Monkey-NET: $\mathcal{L}_{adv}, \mathcal{L}_{FM}$

k : 2D matrix of keypoints, v : vector field, v_{res}, v_{cor} : residual and coarse motion fields, m : estimated motion mask



[38] MarioNETte: $\mathcal{L}_2, \mathcal{L}_{adv}, \mathcal{L}_{FM}, \mathcal{L}_{perc}, \mathcal{L}_{hinge}$

l_{st} : t 's landmarks with s 's expression, v : feature maps, w : warped feature maps
IAB: Image Attention Block

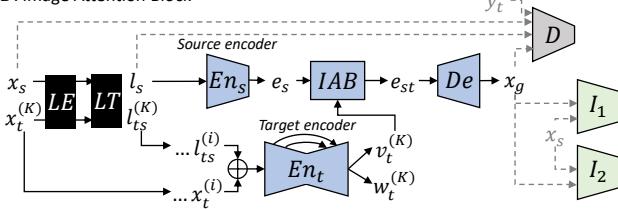
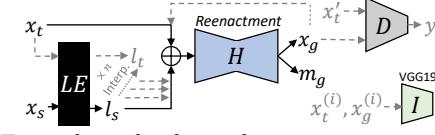


Fig. 5: Architectural schematics of **reenactment networks**. Black lines indicate prediction flows used during deployment, dashed gray lines indicate dataflows performed during training. Zoom in for more detail.

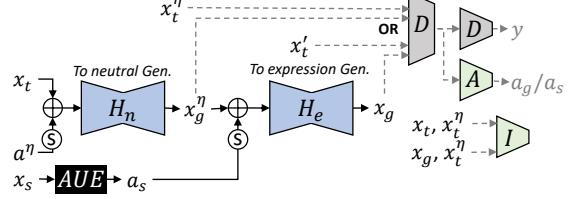
[15] FSGAN: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{con}, \mathcal{L}_{FM}, \mathcal{L}_{perc}$

m : segmentation mask (face, hair, other), l : 3D facial landmarks, H^n : n passes through H while interpolating l_s to l_t



[31] ICFace: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{perc}, \mathcal{L}_*$

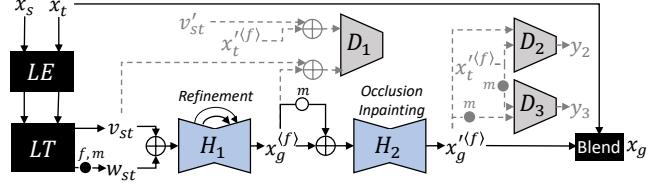
a: Action Units (AU), η : neutral expression



[33] wg-GAN: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_2$

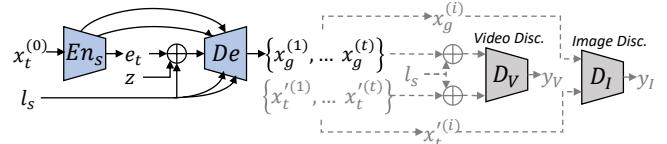
v_{st} : vector map of the warp from x_t to x_s , w_{st} : x_t warped according to v_{st}

Training: for each $x_t^{(i)}, x_s = x_t^{(i-10)}$ taken from the same video clip

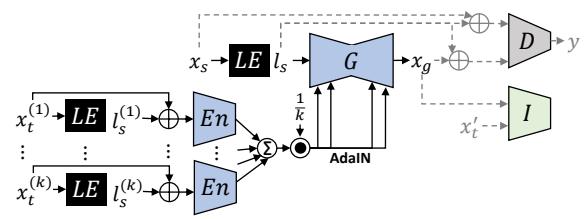


[35] ImaGINator: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_*$

l : One-hot label encoding of expression, z : Random value $z \sim N(0,1)$



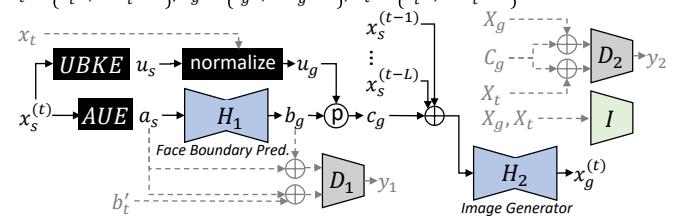
[37] Neural Talking Heads: $\mathcal{L}_{adv}, \mathcal{L}_{FM}, \mathcal{L}_{perc}, \mathcal{L}_{hinge}$



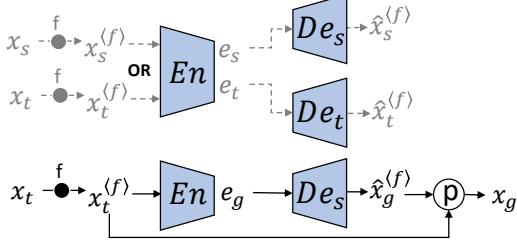
[39] Liu et al. 2019: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{FM}, \mathcal{L}_{perc}, \mathcal{L}_*$

UBKE: Upper-body Key point Extractor

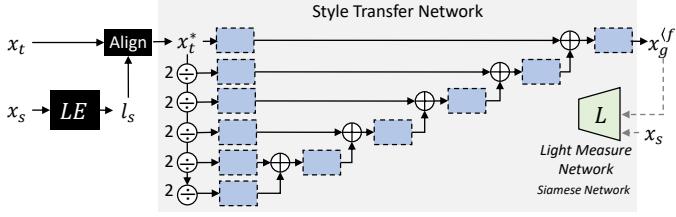
$X_t = \{x_t^{(t)}, \dots, x_t^{(t-L)}\}$, $C_g = \{c_g^{(t)}, \dots, c_g^{(t-L)}\}$, $X_t = \{x_t^{(t)}, \dots, x_t^{(t-L)}\}$



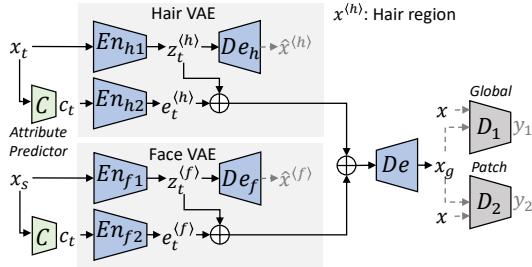
[87]–[89] Reddit ‘deepfakes’ variants: $\mathcal{L}_1, \mathcal{L}_2$



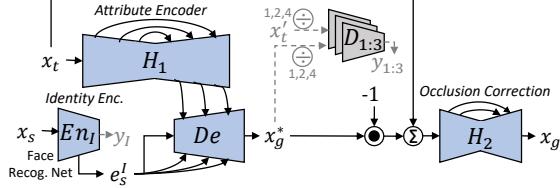
[90] Fast Face Swap: $\mathcal{L}_{FM}, \mathcal{L}_{perc}, \mathcal{L}_{tv}, \mathcal{L}_*$



[92] RSGAN: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{CE}, \mathcal{L}_{KL}$

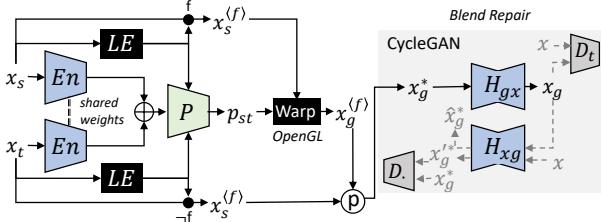


[94] FaceShifter: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_{cos}, \mathcal{L}_{perc}$



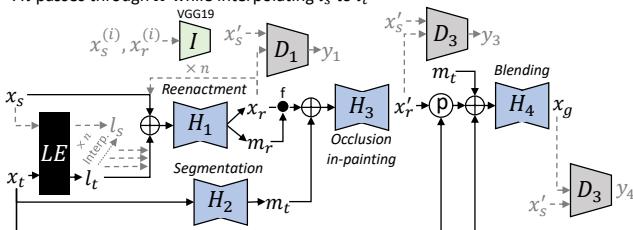
[96] Depth Nets: $\mathcal{L}_{adv}, \mathcal{L}_{cyc}, \mathcal{L}_*$

p_{st} : affine transformation parameters and depth measures

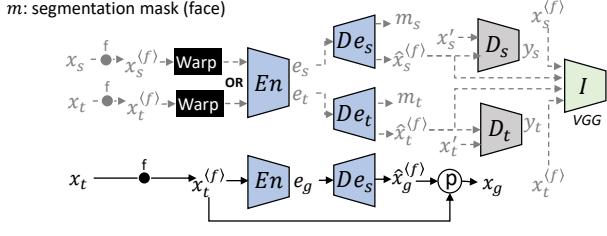


[15] FSGAN: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{FM}, \mathcal{L}_{perc}, \mathcal{L}_*$

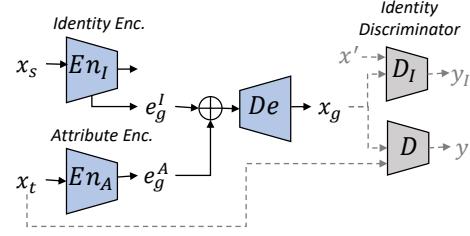
m : segmentation mask (face, hair, other), l : 3D facial landmarks, H^n : n passes through H while interpolating l_s to l_t



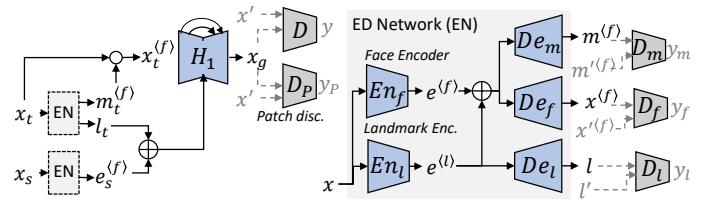
[89] Face Swap GAN: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{perc}, \mathcal{L}_{tv}, \mathcal{L}_*$



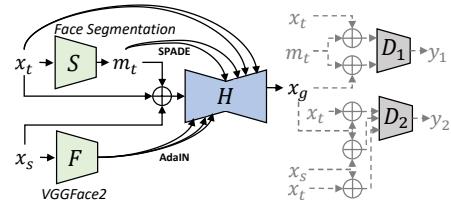
[91] OSIP-FS: $\mathcal{L}_{adv}, \mathcal{L}_2, \mathcal{L}_{CE}, \mathcal{L}_{FM}, \mathcal{L}_{KL}$



[93] FSNet: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{CE}, \mathcal{L}_{trip}, \mathcal{L}_{KL}, \mathcal{L}_*$



[95] Few-Shot Face Translation: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_*$



[97] IHPT: $\mathcal{L}_{adv}, \mathcal{L}_1, \mathcal{L}_{CE}, \mathcal{L}_{MS}, \mathcal{L}_*$

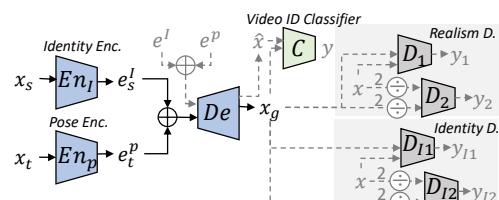


Fig. 7: Architectural schematics of the **replacement networks**. Black lines indicate prediction flows used during deployment, dashed gray lines indicate dataflows performed during training. Zoom in for more detail.

TABLE II: Summary of Deep Learning Replacement Models

	Replacement	Retraining for new...		Model	Repr.	Model Training	Model Execution	Model Output	
	Transfer Swap	Source (s)	Target (t)	Identity Agnostic			Source (x_s ...)	Target (x_t ...)	Resolution
One-to-One	[88] 2017 Deepfakes for All	• 2-5k portraits/frames	2-5k portraits/frames	1 2 0-1 0	•	•	-	portrait	256x256
	[89] 2018 FaceSwap-GAN	• 2-5k portraits/frames	2-5k portraits/frames	1 2 2 1	•	•	-	portrait	256x256
	[87] 2018 DeepFaceLab	• 2-5k portraits/frames	2-5k portraits/frames	1 2 0-1 0	•	•	-	portrait	256x256
One-to-Many	[90] 2017 Fast Face Swap	• 60 portraits	None	0 0 0 2	• • •	•	portrait	portrait	256x256
Many-to-Many	[92] 2018 RSGAN	• None	None	4 3 2 1	•	•	portrait	portrait	128x128
	[93] 2018 FSNet	• None	None	3 4 5 0	• • •	•	portrait	portrait	128x128
	[91] 2018 OSIP-FS	• None	None	2 1 2 0	•	•	portrait	portrait	128x128
	[96] 2018 DepthNets	• None	None	• 3 2 2 1	•	•	portrait	portrait	80x80
	[15] 2018 FSGAN	• None	None	• 4 4 3 1	• •	•	portrait	portrait	256x256
	[103] 2018 IO-FR	• None	None	• 1 1 1 1	•	•	portrait	portrait	256x256
	[95] 2019 FS Face Trans.	• None	None	• 1 1 2 2	•	•	portraits	portrait	128x128
	[97] 2019 IHPT	• None	None	• 2 1 2 0	•	•	cropped object	cropped object	128x128
	[94] 2019 FaceShifter	• None	None	• 3 3 3 0	•	•	portrait	portrait	256x256

The components are trained concurrently as two autoencoders: $De_s(En(x_s)) = \hat{x}_s$ and $Det(En(x_t)) = \hat{x}_t$, where x is a cropped face image. As a result, En learns to map s and t to a shared latent space, such that

$$De_s(En(x_t)) = x_g \quad (4)$$

Currently, there are a number of open source face swapping tools on GitHub based on the original network. One of the most popular is DeepFaceLab [87]. Their current version offers a wide variety of model configurations, including adversarial training, residual blocks, a style transfer loss, and masked loss to improve the quality of the face and eyes. To help the network map the target's identity into arbitrary face shapes, the training set is augmented with random face warps.

Another tool called FaceSwap-GAN [89] follows a similar architecture, but uses a denoising autoencoder with a self-attention mechanisms, and offers cycle-consistency loss. The decoders in FaceSwap-GAN also generate segmentation masks which help the model handle occlusions and is used to blend x_g back into the target frame. Finally, [88] is another open source tool that provides a GUI. Their software comes with 10 popular implementations, including that of [87], and multiple variations of the original Reddit user's code.

2) One-to-Many (Single Identity to Multiple Identities):

In [90], the authors use a modified style transfer with CNN, where the content is x_t and the style is the identity of x_s . The process is (1) align x_t to a reference x_s , (2) transfer the identity of s to the image using a multi scale CNN, trained with style loss on images of s , and (3) align the output to x_t and blend the face back in with a segmentation mask.

3) Many-to-Many (Multiple IDs to Multiple IDs):

One of the first identity agnostic methods was [60], mentioned in Section IV-A3. However, to train the CGAN, one needs a dataset of paired faces with different identities having the same expression.

Disentanglement with EDs. In [91] the authors us an ED to disentangle the identity from the attributes (pose, hair, background, and lighting) during the training process. The identity encodings are the last pooling layer of a face classifier,

and the attribute encoder is trained using a weighted L2 loss and a KL divergence loss to mitigate identity leakage. The authors also show that they can adjust attributes, expression, and pose via interpolation of the encodings. In [103] the authors propose a system for identity obfuscation which swaps x_t 's face with a similar or different identity. An ED is used to predict the 3D head parameters with are then either modified or replaced with the source's. Finally a GAN is used to in-paint the face of x_t given the head model parameters.

Disentanglement with VAEs. In [92], the authors propose RSGAN: a VAE-GAN consisting of two VAEs and a decoder. One VAE encodes the hair region and the other encodes the face region, where both are conditioned on a predicted attribute vector c describing x . Since VAEs are used, the facial attributes can be edited through c . In [93] the authors propose FSNet. A VAE ED network is run on x_s and then x_t producing encodings for the face of x_s and the landmarks of x_t . To perform a face swap, a generator receives the masked portrait of x_t and performs in-painting on the masked face. The generator uses the landmark encodings in its embedding layer. During training, randomly generated faces are used with triplet loss on the encodings to preserve identities.

Face Occlusions. FSGAN [15], mentioned Section IV-A3, is also capable of face swapping and can handle occlusions. After the face reenactment generator produces x_r , a second network predicts the target's segmentation mask m_t . Then $(x_r^{(f)}, m_t)$ is passed to a third network that performs inpainting for occlusion correction. Finally a fourth network blends the corrected face into x_t while considering ethnicity and lighting. In [94], the authors propose FaceShifter which uses a series of novel Adaptive Attentional Denormalization layers (AAD), that receives x_t feature maps, as a generator.

4) Few-Shot Learning: The same author of FaceSwap-GAN [89] also hosts few-shot approach online dubbed "One Model to Swap Them All" [95]. In this version the generator receives $(x_s^{(f)}, x_t^{(f)}, m_t)$ where its encoder is conditioned on VGGFace2 features of x_t using FC-AdaIN layers, and its decoder is conditioned on x_t and the face structure m_t via

layer concatenations and SPADE-ResBlocks respectively. Two discriminators are used: one on image quality given the face segmentation and the other on the identities.

B. Transfer

Although face transfers precede face swaps, today there are very few works that use deep learning for this task. However, we note that face a transfer is equivalent to performing *self-reenactment* on a face swapped portrait. Therefore, high quality face a transfers can be achieved by combining a method from Section IV-A and Section V-A.

In 2018, the authors of [96] proposed DepthNets: an unsupervised network for capturing facial landmarks and translating the pose from one identity to another. The authors use a Siamese network to predicts a transformation matrix that maps the x_s 's 3D facial landmarks to the corresponding 2D landmarks of x_t . A 3D renderer (OpenGL) is then used to warp $x_s^{(f)}$ to the source pose l_t , and the composition is refined using a CycleGAN. Since warping is involved, the approach is sensitive to occlusions.

Later in 2019, the authors of [97] proposed a self-supervised network which can change the identity of an object within an image. Their ED disentangles the identity from an objects pose using a novel disentanglement loss. Furthermore to handle misaligned poses, an L1 loss is computed using a pixel mapped version of x_g to x_s (using the weights of the identity encoder).

VI. COUNTERMEASURES

A summary and systematization of the deepfake detection methods can be found in Table III.

A. Detection.

The subject of image forgery detection is a well researched subject [104]. In our review of detection methods, we will focus on works which specifically deal with detecting deepfakes of humans.

Artifact-based Detection. Deepfakes often generate artifacts which may be subtle to humans, but can be easily detected using machine learning. In 2014, researchers had this hypothesis and monitored physiological signals, such as heart rate, to detect computer generated faces [105]. Regarding deepfakes, [106] monitored irregular eye blinking patterns and [107] monitored blood volume patterns (pulse) under the skin.

Inconsistencies are also a revealing factor. In [108] and [109], the authors noticed that video dubbing attacks can be detected by correlating the speech to landmarks around the mouth. In [110] it was shown that similar artifacts appear when predicting the facial landmarks. With large amounts of data on the target, mannerisms and other behaviors can be monitored for anomalies. For example, in [111] the authors protect world leaders from a wide variety of deepfake attacks by modeling their recorded stock footage.

Some artifacts appear where the generated content was blended back into the frame. The authors of [112]–[116] use edge detectors, quality measures, and frequency analysis to detect artifacts in the pasted content and borders. In, [117] the authors detect deepfakes by decomposing them into to their sources while identifying the content's boundary. Some works identify and visualize the tampered regions by either predicting masks learned from a ground truth, or by mapping the neural activations to the raw image [118]–[121].

The content of a fake face can be anomalous in context to the rest of the frame. For example, residuals from face warping processes [122]–[124], lighting [125], and varying fidelity [126] indicate the presence of generated content. In [127] and [128], the authors found that GANs leave unique fingerprints and show how it is possible to classify the generator given the content, even in the presence of compression and noise. In [129] the authors analyze a camera's unique sensor noise (PRNU) to detect pasted content.

As noted in Section IV-A, realistic temporal coherence is challenging to generate, and some authors capitalize on the resulting artifacts to detect the fake content. For example, [130] uses an RNN to detect artifacts such as flickers and jitter, and [131] uses an LSTM on the face region only. In [132] the optical flow between frames is analyzed, and in [48] a classifier is trained on the two frames directly.

Undirected Approaches. Instead of focusing on a specific artifact, some authors train deep neural networks as generic classifiers, and let the network decide which features to analyze [133]–[137]. In [138]–[140], it was shown that deep neural networks tend to perform better than traditional image forensic tools on compressed imagery. Alternatively, to overcome noise and other distortions, the authors of [141] measure the neural activation (coverage) of a face recognition network to obtain a stronger signal from than just using the raw pixels. In [91] the authors detect deepfakes by measuring an input's embedding distance to real samples using an ED's latent space. To improve performance of their models, some add noise to their training data [142] and others try to develop models which generalize better to unseen attacks/generators through disentanglement [143] and semi-supervised learning [144].

B. Prevention & Mitigation

To prevent deepfakes, some have suggested that data provenance of multimedia should be tracked through distributed ledgers and blockchain networks [145]. In [146] the authors suggest that the content should be ranked by participants and AI. In contrast, [147] proposes that the content should be authenticated and managed as a global file system over Etherium smart contracts. To combat deepfakes, the authors of [148] show how adversarial machine learning can be used to disrupt and corrupt deepfake networks. The authors perform adversarial machine learning to add crafted noise perturbations to x , which prevents deepfake technologies from locating a proper face in x .

VII. DISCUSSION

Over the last few years there has been a shift towards identity agnostic models and high resolution deepfakes. Some notable advancements include (1) unpaired self-supervised training techniques to reduce the amount of initial training data, (2) one/few-shot learning which enables identity theft with a single profile picture, (3) improvements of face quality and identity through AdaIN layers, disentanglement, and pix2pixHD network components, (4) fluid and realistic videos through temporal discriminators and optical flow prediction, and (5) the mitigation of boundary artifacts by using secondary networks to blend composites into seamless imagery (e.g., [13], [41], [43]).

Another large advancement in this domain was the use of perceptual loss on a pre-trained VGG Face recognition network. The approach boosts the facial quality significantly, and as a result, has been adopted in popular online deepfake tools [88], [89]. Another advancement being adopted is the use

TABLE III: Summary of Deepfake Detection Models

	Type	Modality	Content	Method	Eval. Dataset	Performance*
	Reenactment	Replacement	Image	Model	Indicates Affected Area	
	Replacement	Image	Video		Input Resolution	
	Replacement	Image	Audio			
	Feature	Body Part	Face			
	Body Part	Face	Image			
Classic ML	[113] 2017	•	•	SVM-RBF	250x250	• 92.9
	[112] 2017	•	•	SVM	*	18.2
	[110] 2018	•	•	SVM	*	0.97
	[126] 2018	•	•	SVM	128x128	3.33
	[116] 2019	•	•	SVM, LogR., Kmeans	1024x1024	100
	[114] 2019	•	•	SVM	*	13.33
	[111] 2019	•	•	SVM	*	0.98
Deep Learning	[115] 2018	•	•	CNN	256x256	• 99.4
	[106] 2018	•	•	LSTM-CNN	224x224	• 0.99
	[140] 2018	•	•	Capsule-CNN	128x128	• 99.3
	[91] 2018	•	•	ED-GAN	128x128	• 92
	[134] 2018	•	•	CNN	1024x1024	• 0.81
	[130] 2018	•	•	CNN-LSTM	299x299	• 97.1
	[138] 2018	•	•	CNN	256x256	• 94.4
	[143] 2018	•	•	CNN AE	256x256	• 90.5
	[133] 2018	•	•	CNN	256x256	• 0.99
	[131] 2019	•	•	CNN-LSTM	224x224	• 96.9
	[118] 2019	•	•	CNN-DE	• 256x256	• 92.8
	[136] 2019	•	•	CNN	-	• 8.18
	[119] 2019	•	•	CNN AE GAN	• 256x256	• 98.5
	[120] 2019	•	•	CNN+Attention	• 299x299	• 99.2
	[123] 2019	•	•	CNN	128x128	3.11 0.99
	[124] 2019	•	•	CNN	*	0.99 0.99
	[137] 2019	•	•	CNN+HMN	224x224	• 99.4
	[121] 2019	•	•	FCN	256x256	• 98.1
	[142] 2019	•	•	CNN	128x128	• 94.7
	[144] 2019	•	•	CNN	224x224	• 86.4
	[135] 2019	•	•	CNN	1024x1024	• 94
	[107] 2019	•	•	CNN	128x128	• 96
	[122] 2019	•	•	CNN	• 224x224	• 93.2
	[132] 2019	•	•	CNN	224x224	• 81.6
	[108] 2019	•	•	LSTM	*	• 22
	[109] 2019	•	•	LSTM-DNN	*	• 16.4
	[48] 2019	•	•	CNN	256x256	• 97
	[127] 2019	•	•	CNN	128x128	• 99.6 0.53
	[141] 2019	•	•	SVM+VGGnet	224x224	85
	[117] 2019	•	•	CNN	• 64x64	• 99.2
Statistics & Steganalysis	[129] 2018	•	•	PRNU	1280x720	• <i>TPR=1 @FPR= 0.03</i>
	[125] 2019	•	•	Statistics	• -	• 0.90.3
	[128] 2019	•	•	PRNU	*	• 90.3

*We display only the best reported performance of each paper to capture the ‘best-case’ scenario.

of a network pipeline. Instead of enforcing a set of global losses on a single network, a pipeline of networks is used where each network is tasked with a different responsibility (conversion, generation, occlusions, blending, etc.) This give more control over the final output and has been able to mitigate most of the challenges mention in Section III-G.

Aside from quality, there are a few limitations with the current deepfake technologies. First, for reenactment, content is always driven and generated with a frontal pose. This limits the reenactment to a very static performance. Today, this is avoided by face swapping the identity onto a lookalike’s body, but a good match is not always possible and this approach has limited flexibility. Second, reenactments and replacements depend on the driver’s performance to deliver the identity’s personality. We believe that next generation deepfakes will utilize videos of the target to stylize the generated content with the expected expressions and mannerisms. This will enable a much more automatic process of creating believable deepfakes. Finally, a new trend is real-time deepfakes. Works such as [15], [46] have achieved real-time deepfakes at 30fps. Although real-

time deepfakes are an enabler for phishing attacks, the realism is not quite there yet. Regardless, deepfakes are already very convincing [139] and are improving at a rapid rate. Therefore, it is important that we focus on effective countermeasures.

B. The Deepfake Arms Race

Like any battle in cyber security, there is an arms race between the attacker and defender. In our SoK, we observed that the majority deepfake detection algorithms assume a static game with the adversary: They are either focused on identifying a specific artifact, or do not generalize well to new distributions and unseen attacks [143]. Moreover, based on the recent benchmark of [124], we observe that the performance of state-of-the-art detectors are decreasing rapidly as the quality of the deepfakes improve. Concretely, the three most recent benchmark datasets (DFD by Google [149], DFDC by Facebook [150], and Celeb-DF by [124]) were released within one month of each other at the end of 2019. However, the deepfake detectors only achieved an AUC of 0.86, 0.76, and 0.66 on each

of them respectively. Even a false alarm rate of 0.001 is far too low considering the millions of images published online daily.

Evading Artifact-based Detectors. To evade an artifact-based detector, the adversary only needs to mitigate a single flaw to evade detection. For example, G can generate the biological signals monitored by [106], [107] by adding a discriminator which monitors these signals. To avoid anomalies in extensive the neuron activation [141], the adversary can add a loss which minimizes neuron coverage. Methods which detect abnormal poses and mannerisms [111] can be evaded by reenacting the entire head and by learning the mannerisms from the same databases. Models which identify blurred content [115] are affected by noise and sharpening GANs [42], [151], and models which search for the boundary where the face was blended in [112]–[117] do not work on deepfakes passed through refiner networks or those which output full frames (e.g., [22], [27], [77] and [49], [51]). Finally, solutions which search for forensic evidence [127]–[129] can be evaded (or at least raise the false alarm rate) by passing x_g through filters, or by performing physical replication or compression.

Evading Deep Learning Classifiers. There are a number of detection methods which apply deep learning directly to the task of deepfake detection (e.g., [133]–[137]). However, an adversary can use adversarial machine learning to evade detection by adding small perturbations to x_g . Advances in adversarial machine learning has shown that these attacks transfer across multiple models regardless of the training data used [152]. Although more research is required to demonstrate this concept, it is entirely feasible and should be addressed in future work.

Moving Forward. Nevertheless, deepfakes are still imperfect, and these methods offer a modest defense for the time being. Furthermore, these works play an important role in understanding the current limitations of deepfakes, and raise the difficulty threshold for malicious users. At some point, it may become too time-consuming and resource-intensive a common attacker to create a good-enough fake to evade detection. However, we argue that solely relying on the development of content-based countermeasures is not sustainable and may lead to a reactive arms-race. Therefore, we advocate for more out-of-band approaches for detecting a preventing deepfakes. For example, the establishment of content provenance and authenticity frameworks for online videos [145]–[147], and proactive defenses such as the use of adversarial machine learning to protect content from tampering [148].

C. Deepfakes in other Domains

In this SoK, we put a focus on human reenactment and replacement attacks; the type of deepfakes which has made the largest impact so far [2], [4]. However, deepfakes extend beyond human visuals, and have spread many other domains. In healthcare, the authors of [153] showed how deepfakes can be used to inject or remove medical evidence in CT and MRI scan for insurance fraud, disruption, and physical harm. In [154] it was shown how one’s voice can be cloned with only five seconds of audio, and in Sept. 2019 a CEO was scammed out of \$250K via a voice clone deepfake [155]. The authors of [156] have shown how deep learning can generate realistic human fingerprints that can unlock multiple users’ devices. In [157] it was shown how deepfakes can be applied to financial records to evade the detection of auditors. Finally, it has been shown how deepfakes of news articles can be generated which has generated some concern [158].

These examples demonstrate that deepfakes are not just attack tools for misinformation, defamation, and propaganda, but also sabotage, fraud, scams, obstruction of justice, and potentially many more.

D. What’s on the Horizon

We believe that in the coming years, we will see more deepfakes being weaponized for monetization. The technology has proven itself in humiliation, misinformation, and defamtion attacks. Moreover, the tools are becoming more practical [88] and efficient [154]. Therefore, it seems natural that malicious users will find ways to use the technology for a profit. As a result, we expect to see an increase in deepfake phishing attacks and scams targeting both companies and individuals.

As the technology matures, real-time deepfakes will become increasingly realistic. Therefore, we can expect that the technology will be used by hacking groups to perform reconnaissance as part of an APT, and by state actors to perform espionage and sabotage by reenacting of officials or family members.

To keep ahead of the game, we must be proactive and consider the adversary’s next step, not just the weaknesses of the current attacks. We suggest that more work be done on evaluating the theoretical limits of these attacks. For example, by finding a bound on a model’s delay can help detect real-time attacks such as [154], and determining the limits of GANs like [159] can help us devise the appropriate strategies. As mentioned earlier, we recommend further research on solutions which do not require analyzing the content itself. Moreover, we believe it would be beneficial for future works to explore the weaknesses and limitations of current deepfakes detectors. By identifying and understanding these vulnerabilities, researchers will be able to develop stronger countermeasures.

VIII. CONCLUSION

Not all deepfakes are malicious. However, because the technology makes it so easy to create believable media, malicious users are exploiting it to perform attacks. These attacks are targeting individuals and causing psychological, political, monetary, and physical harm. As time goes on, we expect to see these malicious deepfakes spread to many other modalities and industries.

In this SoK we focused on reenactment and replacement deepfakes of humans. We provided a deep review of how these technologies work, the differences between their architectures, and what is being done to detect them. We hope this information will be helpful to the community in understanding and preventing malicious deepfakes.

REFERENCES

- [1] Dami Lee. Deepfake salvador dal takes selfies with museum visitors - the verge. <https://bit.ly/3cEim4m>, 5 2019.
- [2] Holly Kathleen Hall. Deepfake videos: When seeing isn’t believing. *Cath. UJL & Tech.*, 27:51, 2018.
- [3] Robert Chesney and Danielle Keats Citron. Deep fakes: a looming challenge for privacy, democracy, and national security. 2018.
- [4] Arije Antinori. Terrorism and deepfake: from hybrid warfare to post-truth warfare in a hybrid world. In *ECAIR 2019 European Conference on the Impact of Artificial Intelligence and Robotics*, page 23. Academic Conferences and publishing limited, 2019.
- [5] Daniil Kononenko et al. Photorealistic monocular gaze redirection using machine learning. *IEEE transactions on pattern analysis and machine intelligence*, 40(11):2696–2710, 2017.

- [6] Luan Tran, Xi Yin, and Xiaoming Liu. Representation learning by rotating your faces. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):3007–3021, 2018.
- [7] Oscar Schwartz. You thought fake news was bad? – the guardian. <https://www.theguardian.com/technology/2018/nov/12/deep-fakes-fake-news-truth>, 5 2018. (Accessed on 03/02/2020).
- [8] Karen Hao. The biggest threat of deepfakes isn't the deepfakes themselves - miit tech review. <https://www.technologyreview.com/s/614526/the-biggest-threat-of-deepfakes-isnt-the-deepfakes-themselves/>, 2019.
- [9] Sigal Samuel. A guy made a deepfake app to turn photos of women into nudes. it didnt go well. <https://www.vox.com/2019/6/27/18761639/ai-deepfake-deepnude-app-nude-women-porn>, 5 2019.
- [10] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*, 2019.
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [13] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.
- [14] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- [15] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7184–7193, 2019.
- [16] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: One-shot anatomically consistent facial animation. *International Journal of Computer Vision*, pages 1–16, 2019.
- [17] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Advances in Neural Information Processing Systems 32*, pages 7135–7145. Curran Associates, Inc., 2019.
- [18] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. End-to-end speech-driven realistic facial animation with temporal gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 37–40, 2019.
- [19] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. Reenactgan: Learning to reenact faces via boundary transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 603–619, 2018.
- [20] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.
- [21] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [22] Hyeongwoo Kim, Pablo Carrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Perez, Christian Richardt, Michael Zollhofer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018.
- [23] Hai X Pham, Yuting Wang, and Vladimir Pavlovic. Generative adversarial talking head: Bringing portraits to life with a weakly supervised neural network. *arXiv preprint arXiv:1803.07716*, 2018.
- [24] Enrique Sanchez and Michel Valstar. Triple consistency loss for pairing distributions in gan-based face synthesis. *arXiv preprint arXiv:1811.03492*, 2018.
- [25] Yujun Shen, Ping Luo, Junjie Yan, Xiaogang Wang, and Xiaou Tang. Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2018.
- [26] Yujun Shen, Bolei Zhou, Ping Luo, and Xiaou Tang. Facefeat-gan: a two-stage approach for identity-preserving face synthesis. *arXiv preprint arXiv:1812.01288*, 2018.
- [27] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, Hao Li, Richard Roberts, et al. Pagan: real-time avatars using dynamic textures. *ACM Trans. Graph.*, 37(6):258–1, 2018.
- [28] Olivia Wiles, A Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–686, 2018.
- [29] Jiangning Zhang, Xianfang Zeng, Yusu Pan, Yong Liu, Yu Ding, and Changjie Fan. Faceswapnet: Landmark guided many-to-many face reenactment. *arXiv preprint arXiv:1905.11805*, 2019.
- [30] Chaoyou Fu, Yibo Hu, Xiang Wu, Guoli Wang, Qian Zhang, and Ran He. High fidelity face manipulation with extreme pose and expression. *arXiv preprint arXiv:1903.12003*, 2019.
- [31] Soumya Tripathy, Juho Kannala, and Esa Rahtu. Icface: Interpretable and controllable face reenactment using gans. *arXiv preprint arXiv:1904.01909*, 2019.
- [32] Shengju Qian et al. Make a face: Towards arbitrary high fidelity face manipulation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.
- [33] Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. Warp-guided gans for single-photo facial animation. *ACM Transactions on Graphics (TOG)*, 37(6):231, 2019.
- [34] Naima Otterdout, Mohamed Daoudi, Anis Kacem, Lahoucine Ballihi, and Stefano Berretti. Dynamic facial expression generation on hilbert hypersphere with conditional wasserstein generative adversarial nets. *arXiv preprint arXiv:1907.10087*, 2019.
- [35] Yaohui Wang, Piotr Bilinski, Francois Fleuret, and Antitz Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. 2020.
- [36] Aliaksandr Siarohin, Stephane Lathuiliere, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019.
- [37] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lemitsky. Few-shot adversarial learning of realistic neural talking head models. *arXiv preprint arXiv:1905.08233*, 2019.
- [38] Sungjoon Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. Marionette: Few-shot face reenactment preserving identity of unseen targets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.
- [39] Zhaoxiang Liu, Huan Hu, Zipeng Wang, Kai Wang, Jinqiang Bai, and Shiguo Lian. Video synthesis of human upper body with realistic face. *arXiv preprint arXiv:1908.06607*, 2019.
- [40] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017.
- [41] Ohad Fried, Ayush Tewari, Michael Zollhofer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. Text-based editing of talking-head video. *arXiv preprint arXiv:1906.01524*, 2019.
- [42] Seyed Ali Jalalifar, Hosein Hasani, and Hamid Aghajani. Speech-driven facial reenactment using conditional generative adversarial networks. *arXiv preprint arXiv:1803.07461*, 2018.
- [43] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Niessner. Neural voice puppetry: Audio-driven facial reenactment. *arXiv preprint arXiv:1912.05566*, 2019.
- [44] Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7832–7841, 2019.
- [45] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9299–9306, 2019.
- [46] Amir Jamaludin, Joon Son Chung, and Andrew Zisserman. You said that?: Synthesising talking faces from audio. *International Journal of Computer Vision*, pages 1–13, 2019.
- [47] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with gans. *arXiv preprint arXiv:1906.06337*, 2019.
- [48] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5933–5942, 2019.
- [49] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural rendering and reenactment of human actor videos. *ACM Transactions on Graphics (TOG)*, 38(5):139, 2019.
- [50] Kfir Aberman, Mingyi Shi, Jing Liao, D Lischinski, Baoquan Chen, and Daniel Cohen-Or. Deep video-based performance cloning. In *Computer Graphics Forum*. Wiley Online Library, 2019.

- [51] Polina Zablotskaia, Aliaksandr Siarohin, Bo Zhao, and Leonid Sigal. Dwnet: Dense warp-based network for pose-guided human video generation. *arXiv preprint arXiv:1910.09139*, 2019.
- [52] Volker Blanz, Curzio Bassi, Tomaso Poggio, and Thomas Vetter. Reanimating faces in images and video. In *Computer graphics forum*, volume 22, pages 641–650. Wiley Online Library, 2003.
- [53] Yao-Jen Chang and Tony Ezzat. Transferable videorealistic speech animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 143–151. ACM, 2005.
- [54] Justus Thies, Michael Zollhofer, Matthias Niessner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. Real-time expression transfer for facial reenactment. *ACM Trans. Graph.*, 34(6):183–1, 2015.
- [55] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Niessner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.
- [56] Justus Thies, Michael Zollhofer, Christian Theobalt, Marc Stamminger, and Matthias Niessner. Headon: Real-time reenactment of human portrait videos. *ACM Transactions on Graphics (TOG)*, 37(4):164, 2018.
- [57] Runze Xu, Zhiming Zhou, Weinan Zhang, and Yong Yu. Face transfer with generative adversarial network. *arXiv preprint:1710.06090*, 2017.
- [58] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [59] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2745–2754, 2017.
- [60] Kyle Olszewski, Zimo Li, Chao Yang, Yi Zhou, Ronald Yu, Zeng Huang, Sitaq Xiang, Shunsuke Saito, Pushmeet Kohli, and Hao Li. Realistic dynamic facial textures from a single image using gans. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5429–5438, 2017.
- [61] Yuqian Zhou and Bertram Emil Shi. Photorealistic facial expression synthesis by the conditional difference adversarial autoencoder. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 370–376. IEEE, 2017.
- [62] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F. Cohen. Bringing portraits to life. *ACM Transactions on Graphics (Proceeding of SIGGRAPH Asia 2017)*, 36(6):196, 2017.
- [63] Yunxuan Zhang, Siwei Zhang, Yue He, Cheng Li, Chen Change Loy, and Ziwei Liu. One-shot face reenactment. *arXiv preprint arXiv:1908.03251*, 2019.
- [64] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [65] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.
- [66] Rithesh Kumar, Jose Sotelo, Kundan Kumar, Alexandre de Brebisson, and Yoshua Bengio. Obamanet: Photo-realistic lip-sync from text. *arXiv preprint arXiv:1801.01442*, 2017.
- [67] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. Char2wav: End-to-end speech synthesis. *Openreview.net*.
- [68] Justus Thies, Michael Zollhofer, and Matthias Niessner. Deferred neural rendering: Image synthesis using neural textures. *arXiv preprint arXiv:1904.12356*, 2019.
- [69] Taiki Shimba, Ryuhei Sakurai, Hirotake Yamazoe, and Joo-Ho Lee. Talking heads synthesis from audio with deep neural networks. In *2015 IEEE/SICE International Symposium on System Integration (SII)*, pages 100–105. IEEE, 2015.
- [70] Yang Song, Jingwen Zhu, Xiaolong Wang, and Hairong Qi. Talking face generation by conditional recurrent adversarial network. *arXiv preprint arXiv:1804.04786*, 2018.
- [71] Lingyun Yu, Jun Yu, and Qiang Ling. Mining audio, text and visual information for talking face generation. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 787–795. IEEE, 2019.
- [72] Triantafyllos Kefalas, Konstantinos Vougioukas, Yannis Panagakis, Stavros Petridis, Jean Kossaifi, and Maja Pantic. Speech-driven facial animation using polynomial fusion of features. *arXiv preprint arXiv:1912.05833*, 2019.
- [73] Yibo Hu, Xiang Wu, Bing Yu, Ran He, and Zhenan Sun. Pose-guided photorealistic face rotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8398–8406, 2018.
- [74] Jie Cao, Yibo Hu, Bing Yu, Ran He, and Zhenan Sun. 3d aided duet gans for multi-view face image synthesis. *IEEE Transactions on Information Forensics and Security*, 14(8):2028–2042, 2019.
- [75] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *European conference on computer vision*. Springer, 2016.
- [76] Yu Yu, Gang Liu, and Jean-Marc Odobez. Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11937–11946, 2019.
- [77] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L Berg. Dance dance generation: Motion transfer for internet videos. *arXiv preprint arXiv:1904.00129*, 2019.
- [78] Aliaksandr Siarohin, Enver Sangineto, Stephane Lathuiliere, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018.
- [79] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2347–2356, 2019.
- [80] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *Proceedings of the European conference on computer vision (ECCV)*, pages 123–138, 2018.
- [81] Riza Alp Guler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [82] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8340–8348, 2018.
- [83] Liqian Ma et al. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2018.
- [84] Rodrigo De Bem, Arnab Ghosh, Adnane Boukhayma, Thalaiyasingam Ajanthan, N Siddharth, and Philip Torr. A conditional deep generative model of people in natural images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019.
- [85] Jessica Lee, Deva Ramanan, and Rohit Girdhar. Metapix: Few-shot video retargeting. *arXiv preprint arXiv:1910.04742*, 2019.
- [86] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR.org, 2017.
- [87] iperov. Deepfacelab: Deepfacelab is a tool that utilizes machine learning to replace faces in videos. <https://github.com/iperov/DeepFaceLab>, 2019. (Accessed on 12/31/2019).
- [88] deepfakes/faceswap: Deepfakes software for all. <https://github.com/deepfakes/faceswap>, 2017. (Accessed on 01/27/2020).
- [89] shaoanlu. faceswap-gan: A denoising autoencoder + adversarial losses and attention mechanisms for face swapping. <https://github.com/shaoanlu/faceswap-GAN>, 2018. (Accessed on 12/17/2019).
- [90] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [91] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [92] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. Rsgan: face swapping and editing using face and hair representation in latent spaces. *arXiv preprint arXiv:1804.03447*, 2018.
- [93] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. Fsnet: An identity-aware generative model for image-based face swapping. In *Asian Conference on Computer Vision*, pages 117–132. Springer, 2018.
- [94] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Faceshifter: Towards high fidelity and occlusion aware face swapping. *arXiv preprint arXiv:1912.13457*, 2019.
- [95] Shaoanlu. fewshot-face-translation-gan: Generative adversarial networks integrating modules from funit and spade for face-swapping. <https://github.com/shaoanlu/fewshot-face-translation-GAN>, 2019.
- [96] Joel Ruben Antony Moniz, Christopher Beckham, Simon Rajotte, Sina Honari, and Chris Pal. Unsupervised depth estimation, 3d face rotation and replacement. In *Advances in Neural Information Processing Systems*, pages 9736–9746, 2018.
- [97] Fanyi Xiao, Haotian Liu, and Yong Jae Lee. Identity from here, pose from there: Self-supervised disentanglement and generation of objects using unlabeled videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7013–7022, 2019.

- [98] Volker Blanz, Kristina Scherbaum, Thomas Vetter, and Hans-Peter Seidel. Exchanging faces in images. In *Computer Graphics Forum*, volume 23, pages 669–676. Wiley Online Library, 2004.
- [99] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K Nayar. Face swapping: automatically replacing faces in photographs. In *ACM Transactions on Graphics (TOG)*, volume 27, page 39. ACM, 2008.
- [100] Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. Video face replacement. In *ACM Transactions on Graphics (TOG)*. ACM, 2011.
- [101] Ira Kemelmacher-Shlizerman. Transfiguring portraits. *ACM Transactions on Graphics (TOG)*, 35(4):94, 2016.
- [102] Yuval Nirkin, Iacopo Masi, Anh Tran Tuan, Tal Hassner, and Gerard Medioni. On face segmentation, face swapping, and face perception. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 98–105. IEEE, 2018.
- [103] Qianru Sun, Ayush Tewari, Weipeng Xu, Mario Fritz, Christian Theobalt, and Bernt Schiele. A hybrid model for identity obfuscation by face replacement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 553–569, 2018.
- [104] Lilei Zheng, Ying Zhang, and Vrizlynn LL Thing. A survey on image tampering and its detection in real-world photos. *Journal of Visual Communication and Image Representation*, 58:380–399, 2019.
- [105] Valentina Conotter, Ecaterina Bodnari, Giulia Boato, and Hany Farid. Physiologically-based detection of computer generated faces in video. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 248–252. IEEE, 2014.
- [106] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.
- [107] Umar Aybars Ciftci and Ilke Demir. Fakecatcher: Detection of synthetic portrait videos using biological signals. *arXiv preprint arXiv:1901.02212*, 2019.
- [108] Pavel Korshunov and Sebastien Marcel. Speaker inconsistency detection in tampered video. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2375–2379. IEEE, 2018.
- [109] Pavel Korshunov et al. Tampered speaker inconsistency detection with phonetically aware audio-visual features. In *International Conference on Machine Learning*, number CONF, 2019.
- [110] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019.
- [111] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–45, 2019.
- [112] Akshay Agarwal, Richa Singh, Mayank Vatsa, and Afzel Noore. Swapped! digital face presentation attack detection via weighted local magnitude pattern. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 659–665. IEEE, 2017.
- [113] Ying Zhang, Lilei Zheng, and Vrizlynn LL Thing. Automated face swapping and its detection. In *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, pages 15–19. IEEE, 2017.
- [114] Zahid Akhtar and Dipankar Dasgupta. A comparative evaluation of local feature descriptors for deepfakes detection.
- [115] Huaxiao Mo, Bolin Chen, and Weiqi Luo. Fake faces identification via convolutional neural network. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2018.
- [116] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, and Janis Keuper. Unmasking deepfakes with simple features. *arXiv preprint arXiv:1911.00686*, 2019.
- [117] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. *arXiv preprint arXiv:1912.13458*, 2019.
- [118] Huy H Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. *arXiv preprint arXiv:1906.06876*, 2019.
- [119] Mengnan Du, Shiva Pentylala, Yuening Li, and Xia Hu. Towards generalizable forgery detection with locality-aware autoencoder. *arXiv preprint arXiv:1909.05999*, 2019.
- [120] Joel Stehouwer, Hao Dang, Feng Liu, Xiaoming Liu, and Anil Jain. On the detection of digital face manipulation. *arXiv preprint arXiv:1910.01717*, 2019.
- [121] Jia Li, Tong Shen, Wei Zhang, Hui Ren, Dan Zeng, and Tao Mei. Zooming into face forensics: A pixel-level analysis. *arXiv preprint arXiv:1912.05790*, 2019.
- [122] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.
- [123] Yuezun Li and Siwei Lyu. Dsp-fwa: Dual spatial pyramid for exposing face warp artifacts in deepfake videos. <https://github.com/danmohaha/DSP-FWA>, 2019. (Accessed on 12/18/2019).
- [124] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A new dataset for deepfake forensics. *arXiv preprint:1909.12962*, 2019.
- [125] Jeremy Straub. Using subject face brightness assessment to detect deep fakes(conference presentation). In *Real-Time Image Processing and Deep Learning 2019*, volume 10996, page 109960H. International Society for Optics and Photonics, 2019.
- [126] Pavel Korshunov and Sebastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.
- [127] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.
- [128] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 506–511. IEEE, 2019.
- [129] Marissa Koopman, Andrea Macarulla Rodriguez, and Zeno Geradts. Detection of deepfake video manipulation. In *Conference: IMVIP*, 2018.
- [130] David Guera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018.
- [131] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent-convolution approach to deepfake detection-state-of-art results on faceforensics++. *arXiv preprint arXiv:1905.00582*, 2019.
- [132] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [133] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018.
- [134] Nhu-Tai Do, In-Seop Na, and Soo-Hyung Kim. Forensics face detection from gans using convolutional neural network, 2018.
- [135] Shahroz Tariq, Sangyup Lee, Hoyoung Kim, Youjin Shin, and Simon S Woo. Detecting both machine and human created fake face images in the wild. In *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security*, pages 81–87. ACM, 2018.
- [136] Xinyi Ding, Zohreh Raziei, Eric C Larson, Eli V Olinick, Paul Krueger, and Michael Hahsler. Swapped face detection using deep learning and subjective assessment. *arXiv preprint arXiv:1909.04217*, 2019.
- [137] Tharindu Fernando, Clinton Fookes, Simon Denman, and Sridha Sridharan. Exploiting human social cognition for the detection of fake and fraudulent faces via memory networks. *arXiv preprint arXiv:1911.07844*, 2019.
- [138] Francesco Marra, Diego Gragnaniello, Davide Cozzolino, and Luisa Verdoliva. Detection of gan-generated fake images over social networks. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 384–389. IEEE, 2018.
- [139] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. Faceforensics++: Learning to detect manipulated facial images. *arXiv preprint:1901.08971*, 2019.
- [140] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2307–2311. IEEE, 2019.
- [141] Run Wang, Lei Ma, Felix Juefei-Xu, Xiaofei Xie, Jian Wang, and Yang Liu. Fakespotter: A simple baseline for spotting ai-synthesized fake faces. *arXiv preprint arXiv:1909.06122*, 2019.
- [142] Xinheng Xuan, Bo Peng, Wei Wang, and Jing Dong. On the generalization of gan image forensics. In *Chinese Conference on Biometric Recognition*, pages 134–141. Springer, 2019.
- [143] Davide Cozzolino, Justus Thies, Andreas Rossler, Christian Riess, Matthias Niessner, and Luisa Verdoliva. Forensicstransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018.
- [144] Xiaoguang Tu, Hengsheng Zhang, Mei Xie, Yao Luo, Yuefei Zhang, and Zheng Ma. Deep transfer across domains for face anti-spoofing. *arXiv preprint arXiv:1901.05633*, 2019.

- [145] Paula Fraga-Lamas and Tiago M Fernandez-Carames. Leveraging distributed ledger technologies and blockchain to combat fake news. *arXiv preprint arXiv:1904.05386*, 2019.
- [146] Chi-Ying Chen et al. A trusting news ecosystem against fake news from humanity and technology perspectives. In *2019 19th International Conference on Computational Science and Its Applications (ICCSA)*, pages 132–137. IEEE, 2019.
- [147] Haya R Hasan and Khaled Salah. Combating deepfake videos using blockchain and smart contracts. *IEEE Access*, 7:41596–41606, 2019.
- [148] Yuezun Li, Xin Yang, Baoyuan Wu, and Siwei Lyu. Hiding faces in plain sight: Disrupting ai face synthesis with adversarial perturbations. *arXiv preprint arXiv:1906.09288*, 2019.
- [149] Andrew Gully Nick Dufour. Dfd. <https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html>, 9 2019.
- [150] Brian Dolhansky, Russ Howes, Ben Pflaum, Nicole Baram, and Cristian Canton Ferrer. The deepfake detection challenge (dfdc) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019.
- [151] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.
- [152] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.
- [153] Yisroel Mirsky, Tom Mahler, Ilan Shelef, and Yuval Elovici. Ct-gan: Malicious tampering of 3d medical imagery using deep learning. In *USENIX Security Symposium 2019*, 2019.
- [154] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in neural information processing systems*, pages 4480–4490, 2018.
- [155] Jesse Demiani. A voice deepfake was used to scam a ceo out of \$243,000 - forbes. <https://bit.ly/38sXb1I>, 9 2019.
- [156] Philip Bontrager, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross. Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2018.
- [157] Marco Schreyer, Timur Sattarov, Bernd Reimer, and Damian Borth. Adversarial learning of deepfakes in accounting. *arXiv preprint arXiv:1910.03810*, 2019.
- [158] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadji, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32*, pages 9054–9065. Curran Associates, Inc., 2019.
- [159] Sakshi Agarwal and Lav R Varshney. Limits of deepfake detection: A robust estimation viewpoint. *arXiv preprint arXiv:1905.03493*, 2019.
- [160] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*, 2018.
- [161] Ali Khodabakhsh, Raghavendra Ramachandra, Kiran Raja, Pankaj Wasnik, and Christoph Busch. Fake face detection methods: Can they be generalized? In *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6. IEEE, 2018.