SemanticAdv: Generating Adversarial Examples via Attribute-conditional Image Editing

Haonan Qiu \(^1\) Chaowei Xiao \(^2\) Lei Yang \(^3\) Xincheng Yan \(^2\) Honglak Lee \(^2\) Bo Li \(^4\)

\(^1\)The Chinese University of Hong Kong
\(^2\)University of Michigan, Ann Arbor
\(^3\)CUHK - SenseTime Joint Lab, The Chinese University of Hong Kong
\(^4\)University of Illinois Urbana-Champaign

Abstract

Deep neural networks (DNNs) have achieved great success in various applications due to their strong expressive power. However, recent studies have shown that DNNs are vulnerable to adversarial examples which are manipulated instances targeting to mislead DNNs to make incorrect predictions. Currently, most such adversarial examples try to guarantee "subtle perturbation" by limiting its \(L_p\) norm. In this paper, we aim to explore the impact of semantic manipulation on DNNs predictions by manipulating the semantic attributes of images and generate "unrestricted adversarial examples". Such semantic based perturbation is more practical compared with pixel level manipulation. In particular, we propose an algorithm SemanticAdv which leverages disentangled semantic factors to generate adversarial perturbation via altering either single or a combination of semantic attributes. We conduct extensive experiments to show that the semantic based adversarial examples can not only fool different learning tasks such as face verification and landmark detection, but also achieve high attack success rate against real-world black-box services such as Azure face verification service. Such structured adversarial examples with controlled semantic manipulation can shed light on further understanding about vulnerabilities of DNNs as well as potential defensive approaches.

1 Introduction

Deep neural networks (DNNs) have demonstrated great successes in advancing the state-of-the-art performance of discriminative tasks [Krizhevsky et al., 2012, Goodfellow et al., 2016, He et al., 2016, Collobert and Weston, 2008, Deng et al., 2013, Silver et al., 2016]. However, recent research found that DNNs are vulnerable to adversarial examples which are carefully crafted instances aiming to induce arbitrarily prediction errors for learning systems. Such adversarial examples containing small magnitude of perturbation have shed light on understanding and discovering potential vulnerabilities of DNNs [Szegedy et al., 2013, Goodfellow et al., 2014b, Moosavi-Dezfooli et al., 2016, Papernot et al., 2016, Carlini and Wagner, 2017, Xiao et al., 2018b,c,a, 2019]. Most existing work focused on constructing adversarial examples by adding pixel-wise perturbations [Goodfellow et al., 2014b] or spatially transforming the image [Xiao et al., 2018c, Engstrom et al., 2017] (e.g., in-plane rotation or out-of-plane rotation). Generating structured perturbations with semantically meaningful patterns is an important yet under-explored field.

At the same time, deep generative models have demonstrated impressive performance in learning disentangled semantic factors through data generation in an unsupervised [Radford et al., 2015, Karras et al., 2018, Brock et al., 2019] or weakly-supervised manner based on semantic attributes [Yan et al., 2016, Choi et al., 2018]. Empirical findings in [Yan et al., 2016, Zhu et al., 2016a, Radford et al., 2015] demonstrated that a simple linear interpolation on the learned image manifold can produce smooth visual transitions between a pair of input images.

In this paper, we introduce a novel attack SemanticAdv which generates structured perturbations with semantically meaningful patterns. Motivated by the findings mentioned above, we leverage an attribute-conditional image editing model [Choi et al., 2018] to synthesize adversarial examples by interpolating between source and target images in the feature-map. Here, we focus on changing a single attribute dimension to achieve
adversarial goals while keeping the generated adversarial image realistic (e.g., see Figure 1 for details). To validate the effectiveness of the proposed attack method, we consider two tasks, namely, face verification and landmark detection, as face recognition field has been extensively explored and the existing models are shown to be reasonably robust to attacks. We conduct both qualitative and quantitative evaluations on CelebA dataset [Liu et al., 2015]. Please find more visualization results on the anonymous website: https://sites.google.com/view/generate-semantic-adv-example.

The contributions of the proposed SemanticAdv are three-folds. First, we propose a novel attack method capable of generating structured adversarial perturbations guided by semantic attributes. This allows us to analyze the robustness of a recognition system against different types of semantic attacks. Second, the proposed attack exhibits high transferability and leads to 65% black-box attack success rate on a real-world face verification platform. Third, the proposed method is more effective than pixel-wise perturbations in attacking existing defense methods, which could potentially open up new research opportunities and challenges in the long run.

2 Related Work

Semantic image editing. Semantic image synthesis and manipulation is a popular research topic in machine learning, graphics and vision. Thanks to recent advances in deep generative models [Kingma and Welling, 2014, Goodfellow et al., 2014a, Oord et al., 2016] and the empirical analysis of deep classification networks [Krizhevsky et al., 2012, Simonyan and Zisserman, 2014, Szegedy et al., 2015], past few years have witnessed tremendous breakthroughs towards high-fidelity pure image generation [Radford et al., 2015, Karras et al., 2018, Brock et al., 2019], text-to-image generation [Mansimov et al., 2015, Reed et al., 2016, Van den Oord et al., 2016], and image-to-image translation [Isola et al., 2017, Zhu et al., 2017, Liu et al., 2017, Wang et al., 2018b, Hong et al., 2018].

As a compact and interpretable representation describing factors of the physical world, semantic attributes [Farhadi et al., 2009, Kumar et al., 2009, Parikh and Grauman, 2011] has enjoyed good research attention. In particular, Yan et al. [2016] proposed a probabilistic formulation to synthesize diverse and realistic portrait images from visual attributes. Chor et al. [2018] tackled the face image editing problem with a multi-domain image-to-image translation network which treats attributes as additional input condition. Besides attribute-based generation, our work is also relevant to other research work on neural image editing and completion methods [Zhu et al., 2016a, Shu et al., 2017, Li et al., 2017, Sangkloy et al., 2017, Xian et al., 2018]. Empirical findings in Yan et al. [2016], Zhu et al. [2016a] demonstrated that a simple linear interpolation on the natural image manifold can produce smooth visual transitions between a pair of input images.

Adversarial examples. Generating pixel-wise adversarial perturbations has been extensively studied in the past [Szegedy et al., 2013, Goodfellow et al., 2014b, Moosavi-Dezfooli et al., 2016, Papernot et al., 2016, Carlini and Wagner, 2017, Xiao et al., 2018b]. The key aspect of the previous work is that a $L_p$ norm constraint on the pixel-wise perturbations is usually adopted in preserving the perceptual realism of the generated adversarial examples. Recently, Xiao et al. [2018c, Engstrom et al., 2017] proposed to spatially transform the image patches instead of adding pixel-wise perturbations, which opens a new challenge on
defending against such adversarial attacks. While most of existing work is able to generate adversarial examples that are perceptually realistic, the study on generating semantically meaningful perturbations is relatively under-explored. In contrast, our proposed semanticAdv focuses on generating unrestricted perturbation \cite{Brown et al., 2018} (e.g., our perturbation is not bounded by the \(L_p\) norm) with semantically meaningful patterns guided by visual attributes. Regarding the semantic adversarial perturbations, our work is also related to the concurrent work \cite{Bhattad et al., 2019} which applies semantic transformation in the color or texture space. However, we argue that the proposed semanticAdv is able to generate adversarial examples in a more controllable fashion using visual attributes. We further analyze the robustness of the recognition system by generating adversarial examples guided by different visual attributes.

3 Semantic Adversarial Examples

3.1 Problem Definition

Let \(\mathcal{M}\) be a machine learning model trained on a dataset \(D = \{(x, y)\}\) consisting of image-label pairs, where \(x \in \mathbb{R}^{H \times W \times D_I}\) and \(y \in \mathbb{R}^{D_L}\) denote the image and the ground-truth label, respectively. Here, \(H\), \(W\), \(D_I\), and \(D_L\) denote the image height, image width, number of image channels, and label dimensions, respectively. For each image \(x\), our model \(\mathcal{M}\) makes a prediction \(\hat{y} = \mathcal{M}(x) \in \mathbb{R}^{D_L}\). To simplify the notations in our presentation, we assume the machine learning model \(\mathcal{M}\) is oracle such that \(y = \hat{y}\) holds for every image in the dataset. Given a target image-label pair \((x^\text{tgt}, y^\text{tgt})\) and \(y \neq y^\text{tgt}\), a traditional attacker aims to synthesize adversarial examples \(\{x^\text{adv}\}\) by adding pixel-wise perturbations to or spatially transforming the original image \(x\) such that \(\mathcal{M}(x^\text{adv}) = y^\text{tgt}\).

In this work, we introduce the concept of semantic attacker that aims at generating adversarial examples by adding semantically meaningful perturbation with a conditional generative model \(\mathcal{G}\). Compared to traditional attacker that usually produces unstructured pixel-wise perturbations, the proposed method is able to produce structured perturbations with semantic meaning.

Semantic image editing. Let \(c \in \mathbb{R}^{D_C}\) be an attribute representation reflecting the semantic factors (e.g., expression or hair color of a portrait image) of image \(x\), where \(D_C\) indicates the attribute dimension and \(c_i \in \{0, 1\}\) indicates the appearance of \(i\)-th attribute. Here, our goal is to use the conditional generator for semantic image editing. For example, given a portrait image of a girl with black hair and blonde hair as the new attribute, our generator is supposed to synthesize a new image that turns the girl’s hair from black to blonde. More specifically, we denote the augmented (new) attribute as \(c^\text{new} \in \mathbb{R}^{D_C}\) such that the synthesized image is given by \(x^\text{new} = \mathcal{G}(x, c^\text{new})\). In the special case when there is no attribute change \((c = c^\text{new})\), the generator simply reconstructs the input: \(x = \mathcal{G}(x, c)\). Supported by the findings mentioned in \cite{Bengio et al., 2013, Reed et al., 2014}, our synthesized image \(x^\text{new}\) should fall close to the data manifold if we constrain the change of attribute values to be sufficiently small (e.g., we only update one semantic attribute at a time). In addition, we can potentially generate many such images by linearly interpolating between the semantic embeddings of the conditional generator \(\mathcal{G}\) using original image \(x\) and the synthesized image \(x^\text{new}\) with the augmented attribute.

Attribute-space interpolation. We start with a simple solution (detailed in Eq. 1) assuming the adversarial example can be found by directly interpolating in the attribute-space. Let \(c^\text{adv} \in \mathbb{R}^{D_C}\) be the adversarial attribute vector that used as input to the attribute-conditioned generator. This is also supported by the empirical results on attribute-conditioned image progression \cite{Yan et al., 2016, Radford et al., 2015} that a well-trained generative model has the capability to synthesize a sequence of images with smooth attribute transitions.

\[
\begin{align*}
    x^\text{adv} &= \mathcal{G}(x, c^\text{adv}) \\
    c^\text{adv} &= \alpha \cdot c + (1 - \alpha) \cdot c^\text{new}, \quad \text{where} \quad \alpha \in (0, 1)
\end{align*}
\]

Feature-map interpolation. Alternatively, we propose to interpolate using the feature map produced by the generator \(\mathcal{G} = \mathcal{G}_{\text{dec}} \circ \mathcal{G}_{\text{enc}}\). Here, \(\mathcal{G}_{\text{enc}}\) is the encoder module that takes the image as input and outputs the feature map. Similarly, \(\mathcal{G}_{\text{dec}}\) is the decoder module that takes the feature map as input and outputs the synthesized image. Let \(f = \mathcal{G}_{\text{enc}}(x, c) \in \mathbb{R}^{H_F \times W_F \times C_F}\) be the feature map of an intermediate layer in the generator, where \(H_F, W_F\) and \(C_F\) indicate the height, width, and number of channels in the feature map.

\[
\begin{align*}
    f^\text{adv} &= \alpha \odot \mathcal{G}_{\text{enc}}(x, c) + (1 - \alpha) \odot \mathcal{G}_{\text{enc}}(x, c^\text{new}) \\
    x^\text{adv} &= \mathcal{G}_{\text{dec}}(f^\text{adv})
\end{align*}
\]
Compared to attribute-space interpolation which is parameterized by a scalar, we parameterize feature-map interpolation by a tensor $\alpha \in \mathbb{R}^{H_F \times W_F \times C_F}$ ($\alpha_{h,w,k} \in (0, 1)$, where $1 \leq h \leq H_F$, $1 \leq w \leq W_F$, and $1 \leq k \leq C_F$) with the same shape as the feature map. Compared to linear interpolation over attribute-space, such design introduces more flexibility for interpolating between original image and the synthesized image.

3.2 Adversarial Optimization Objectives

As we see in Eq. 3, we find the adversarial image $x^{adv}$ by minimizing the objective $\mathcal{L}(\cdot)$ with respect to the synthesized image $x^*$. Here, each synthesized image $x^*$ is produced by the interpolation using the conditional generator $\mathcal{G}$. In our objective function, the first term is the adversarial metric, the second term is a smoothness constraint, and $\lambda$ is used to control the balance between the two terms. The adversarial metric is minimized once the model $\mathcal{M}$ has been successfully attacked towards the target image-label pair $(x^{tgt}, y^{tgt})$. In identify verification, $y^{tgt}$ is the identity representation of the target image; In landmark detection, $y^{tgt}$ represents certain coordinates.

$$x^{adv} = \arg\min_{x \in \mathcal{X}} \mathcal{L}(x^*), \text{ by Eq.} (1) \text{ and Eq.} (2)$$

$$\mathcal{L}(x^*) = \mathcal{L}_{adv}(x^*; \mathcal{M}, y^{tgt}) + \lambda \cdot \mathcal{L}_{smooth}(x^*) \quad (3)$$

**Identity verification.** In the identity verification task, two images are considered to be the same identity if the corresponding identity embeddings from the verification model $\mathcal{M}$ are reasonably close.

$$\mathcal{L}_{adv}(x^*; \mathcal{M}, y^{tgt}) = \max (\kappa, \Phi_{\mathcal{M}}(x^*, y^{tgt})) \text{, assuming } \mathcal{M}(x^{tgt}) = y^{tgt} \quad (4)$$

As we see in Eq. 4, $\Phi_{\mathcal{M}}^\kappa(\cdot, \cdot)$ measures the distance between two identity embeddings from the model $\mathcal{M}$, where the normalized $L_2$ distance is used in our setting. In addition, we introduce the parameter $\kappa$ representing the constant related to the false positive rate (FPR) threshold computed from the development set.

**Landmark detection.** For structured prediction tasks such as landmark detection, we use Houdini objective proposed in [Cisse et al. 2017] as our adversarial metric. Specifically, we directly attack the target landmark $y^{tgt}$ as the corresponding image is not defined. In addition, $\Phi_{\mathcal{M}}(\cdot, \cdot)$ is a scoring function for each image-label pair and $\gamma$ is the threshold.

$$\mathcal{L}_{adv}(x^*; \mathcal{M}, y^{tgt}) = P_{\gamma \sim \mathcal{N}(0,1)} [\Phi_{\mathcal{M}}(x^*, y) - \Phi_{\mathcal{M}}(x^*, y^{tgt}) < \gamma] \cdot l(y^*, y^{tgt}) \quad (5)$$

where $l(y^*, y^{tgt})$ is task loss decided by the specific adversarial target.

**Interpolation smoothness $\mathcal{L}_{smooth}$.** As the interpolation tensor in the feature-map case has far more parameters compared to the attribute-space case, we propose to enforce a smoothness constraint on the tensor $\alpha$ used in feature-map interpolation. As we see in Eq. 6, the smoothness loss encourages the interpolation tensors to consist of piece-wise constant patches spatially, which has been widely used as a pixel-wise de-noising objective for natural image processing [Mahendran and Vedaldi 2015, Johnson et al. 2016].

$$\mathcal{L}_{smooth} = \sum_{h=1}^{H_F-1} \sum_{w=1}^{W_F} \| \alpha_{h+1,w} - \alpha_{h,w} \|^2_2 + \sum_{h=1}^{H_F} \sum_{w=1}^{W_F-1} \| \alpha_{h,w+1} - \alpha_{h,w} \|^2_2 \quad (6)$$

4 Experiments

In this paper, we focus on analyzing the proposed semanticAdv in attacking state-of-the-art face recognition systems. First, face recognition has been extensively studied for decades and the state-of-the-art recognition systems are assumed to be reasonably robust. Second, face recognition has many real-world applications such as (1) face identification for mobile payment and (2) landmark detection for face editing and stylization. Detailed analysis of semantic adversarial examples on such applications allows better understanding of the vulnerability of the current systems. We believe the proposed semanticAdv is a general semantic attack method that is applicable to many other domains as well.

The experimental section is organized as follows. First, we analyze the quality of generated adversarial examples and compare our method with a pixel-wise optimization based method [Carlini and Wagner 2017] qualitatively. Second, we provide both qualitative and quantitative results by controlling each of the semantic attributes at a time. In terms of attack transferability, we evaluate our proposed semanticAdv on various settings and further demonstrate the effectiveness of our method via black-box attacks against online face verification platforms. Third, we compare our method with the baseline against different defense methods on the face verification task. Finally, we demonstrate that the proposed semanticAdv also applies to the face landmark detection.
4.1 Experimental Setup

Identity verification. We select ResNet50 and ResNet-101 [He et al., 2016] trained on MS-Celeb-1M [Guo et al., 2016] as our face verification models. The models are trained using two different objectives, namely, softmax loss [Sun et al., 2014; Zhang et al., 2018] and cosine loss [Wang et al., 2018a]. For simplicity, we use the notation “R-N-S” to indicate the model with N-layer residual blocks as backbone trained using softmax loss, while “R-N-C” indicates the same backbone trained using cosine loss. For R-101-S model, we decide the parameter κ based on the false positive rate (FPR) for the identity verification task. Three different FPRs have been used: 10^{-4} (with κ = 0.60), 3 \times 10^{-4} (with κ = 1.05), and 10^{-3} (with κ = 1.24). Please check the supplementary materials[A.1] for more details. To distinguish between the FPR we used in generating adversarial examples and the other FPR used in evaluation, we introduce two notations “Generating FPR (G-FPR)” and “Test FPR (T-FPR)”. For the experiment with black-box API attacks, we use the online face verification services provided by Face++ [fac] and AliYun [ali].

Landmark detection. We select Face Alignment Network (FAN) [Bulat and Tzimiropoulos, 2017a] trained on 300W-LP [Zhu et al., 2016b] and fine-tuned on 300-W [Sagonas et al., 2013] for 2D landmark detection. The network is constructed by stacking Hour-Glass network [Newell et al., 2016] with hierarchical block [Bulat and Tzimiropoulos, 2017a]. Given a portrait image as input, FAN outputs 2D heatmaps which can be subsequently leveraged to yield 68 2D landmarks.

Semantic attacks. In our experiments, we randomly sample 1, 280 distinct identities form CelebA [Liu et al., 2015]. To reduce the reconstruction error brought by the generator (e.g., x \neq G(x, c)) in practice, we take one more step to obtain the updated feature map f' = G_{enc}(x', c), where x' = \text{argmin}_x \| G(x', c) - x \| \text{in feature-map interpolation}. In our experiments, we use the last conv layer before upsampling in the generator as our as feature-map f given by the attack effectiveness. We also fix the parameter λ (e.g., balances the adversarial loss and smoothness constraint in Eq. [3]) to be 0.01 for face verification and 0.001 for landmark detection, respectively. We use Adam [Kingma and Ba, 2015] optimizer to produce adversarial examples.

We used the StarGAN [Choi et al., 2018] for attribute-conditional image editing. In particular, we re-trained model on CelebA dataset [Liu et al., 2015] by aligning the face landmarks and then resizing images to resolution 112 \times 112. In addition, we select 17 identity-preserving attributes as our input condition, as such attributes related to facial expression and hair color.

For each distinct identity pair (x, x^{\text{tgt}}), we perform semanticAdv guided by each of the 17 attributes (e.g., we intentionally add or remove one specific attribute while keeping the rest unchanged). In total, for each image x, we generate 17 adversarial images with different augmented attributes. In the experiments, we select a pixel-wise adversarial attack method [Carlini and Wagner, 2017] (referred to CW) as our baseline for comparison. Compared to our proposed method, CW does not require visual attributes as part of the system, as it only generates one adversarial example for each instance. We refer the corresponding attack success rate as instance-wise success rate in which the attack success is calculated for each instance.

4.2 SemanticAdv on Identity Verification

![Figure 2: Qualitative comparisons between our proposed semanticAdv and pixel-wise adversarial examples generated by CW [Carlini and Wagner, 2017]. Along with the adversarial examples, we also provide the corresponding perturbations (residual) on the right. Perturbations generated by our semanticAdv are structured with semantically meaningful patterns.](image)

Overall analysis. We first compare our proposed semanticAdv with the pixel-wise attack method CW both quantitatively and qualitatively. For fair comparisons, we also calculate the instance-wise attack success rate.
For each instance with 17 adversarial images using different augmented attributes, if one of the 17 resulting images can attack successfully, we count the attack of this instance as one success, vice versa. The attack success rate using attribute-space interpolation is much lower than the one using feature-map interpolation. We believe the feature-map interpolation adds more flexibility compared to attribute-space interpolation with more parameters to optimize. In addition, we found that the attack success rate is higher with a larger G-FPR.

Figure 2 shows the generated adversarial images and corresponding perturbations against R-101-S of semanticAdv and CW respectively. The text below each figure is the name of augmented attribute, the sign before the name represents “adding” (in red) or “removing” (in blue) the corresponding attribute from the original image. We see that semanticAdv is able to generate perceptually realistic examples guided by the corresponding attribute. In particular, semanticAdv is able to generate perturbations on the corresponding regions correlated with the augmented attribute, while the perturbations of CW have no specific pattern and are evenly distributed across the image.

**Analysis: controlling single attribute.** One of the key advantages of semanticAdv is that we can generate adversarial perturbations in a more controllable fashion guided by the semantic attributes. This allows to analyze the robustness of a recognition system against different types of semantic attacks. We group the adversarial examples by augmented attributes in various settings. In Figure 2, we present attack success rate against two face verification models, namely, R-101-S and R-101-C, guided by different attributes. We highlight the bar with light blue for G-FPR equals to $10^{-3}$ and blue for G-FPR equals to $10^{-4}$, respectively. As we see in this figure, with a larger G-FPR $10^{-3}$, our semanticAdv can achieve almost 100% attack success rate across different attributes. With a smaller G-FPR $10^{-4}$, we find that semanticAdv guided by some attributes such as Mouth Slightly Open and Arched Eyebrows achieve less than 50% attack success rate, while the other attributes such as Pale Skin and Eyeglasses are relatively less affected. In summary, we found that semanticAdv guided by attributes describing the local shape (e.g., mouth, earrings) achieve relatively lower attack success rate compared to attributes relevant to the color (e.g., hair color) or entire face region (e.g., skin). This suggests that the face verification models used in our experiments are more robustly trained in terms of detecting local shapes compared to colors.

Figure 3 shows the adversarial examples with augmented semantic attributes against R-101-S model. The attribute names are shown in the bottom. The upper images are $G(x, c^{\text{new}})$ generated by StarGAN with augmented attribute $c^{\text{new}}$ where the lower images are the corresponding adversarial images with the same augmented attribute.

**Analysis: semantic attack transferability.** To further understand the property of semanticAdv, we analyze the transferability of semanticAdv on various settings. For each model with different FPRs, we select the successfully attacked adversarial examples from Section 4.1 to construct our evaluation dataset. We evaluate them on different models. Table 1a illustrates the transferability of semanticAdv among different models by using the same FPRs (G-FPR = T-FPR = $10^{-3}$). Table 1b illustrates the result with different FPRs (G-FPR = $10^{-3}$ and T-FPR = $10^{-3}$) for generation and evaluation. As we see in Table 1a, adversarial examples generated against models trained with softmax loss exhibit certain transferability compared to models trained with cosine loss. We conduct the same experiment by generating adversarial examples with CW and found it does not have transferability compared to our semanticAdv (results in supplementary materials B.3).

![Figure 3](image-url) Figure 3: Quantitative analysis on the attack success rate with different single-attribute attacks. In each figure, we show the results correspond to a larger FPR (G-FPR = T-FPR = $10^{-3}$) in skyblue and the results correspond to a smaller FPR (G-FPR = T-FPR = $10^{-4}$) in blue, respectively.
Surprisingly, as we see in Table 1b, the adversarial examples generated against model with smaller G-FPR = $10^{-3}$ exhibit strong attack success rate when evaluating on the model with larger T-FPR = $10^{-3}$. Especially, we found the adversarial examples generated against R-101-S have the best attack performance on other models. These findings motivate the analysis of black-box API attack detailed in the following paragraph.

| $M_{o_{it}}$ / $M_{opt}$ | R-50-S | R-101-S | R-50-C | R-101-C |
|--------------------------|--------|---------|--------|---------|
| R-50-S                   | 1.000  | 0.017   | 0.022  | 0.017   |
| R-101-S                  | 0.165  | 0.029   | 0.032  | 0.047   |
| R-50-C                   | 0.165  | 0.201   | 1.000  | 0.047   |
| R-101-C                  | 0.119  | 0.253   | 0.039  | 1.000   |

Table 1: Transferrability analysis: cell $(i, j)$ shows attack success rate of adversarial examples generated against $j$-th model and evaluate on $i$-th model. Left table: Results generated with G-FPR = $10^{-3}$ and T-FPR = $10^{-3}$; Right table: Results generated with G-FPR = $10^{-4}$ and T-FPR = $10^{-3}$.

**Black-box API attack.** In this experiment, we generate adversarial examples against R-101-S with G-FPR = $10^{-4}$. We evaluate our algorithm on two industry level APIs, namely, Face++ and AliYun face verification platform. To demonstrate the effectiveness of our method, we also generate pixel-wise adversarial examples by using CW method with the same settings. As Table 2 shows, our method achieves much higher attack success rate than CW among both APIs and all FPR thresholds (e.g., our adversarial examples generated with G-FPR < $10^{-4}$ achieves 64.63% attack success rate on Face++ platform with T-FPR = $10^{-3}$).

| API name | Face++ | AliYun |
|----------|--------|--------|
|          | T-FPR = $10^{-3}$ | T-FPR = $10^{-4}$ | T-FPR = $10^{-3}$ | T-FPR = $10^{-4}$ |
| Original | 2.04   | 0.51   | 0.50   | 0.00   |
| Generated $x^{avg}$ | 4.21   | 0.53   | 0.50   | 0.00   |
| CW (G-FPR = $10^{-3}$) | 16.24  | 3.55   | 4.50   | 0.00   |
| SemanticAdv (G-FPR = $10^{-3}$) | 27.32  | 9.79   | 7.50   | 2.00   |
| CW (G-FPR = $10^{-4}$) | 30.61  | 15.82  | 12.50  | 4.50   |
| SemanticAdv (G-FPR = $10^{-4}$) | 57.22  | 38.66  | 29.50  | 17.50  |
| CW (G-FPR < $10^{-4}$) | 41.62  | 24.37  | 19.00  | 12.00  |
| SemanticAdv (G-FPR < $10^{-4}$) | 64.63  | 42.69  | 35.50  | 22.17  |

Table 2: Quantitative analysis on black-box attack. We use ResNet-101 optimized with softmax loss for evaluation and report the attack success rate(%) on two online face verification platforms.

**User study.** To measure the perceptual realism of the adversarial images generated by semanticAdv, we conduct a user study on Amazon Mechanical Turk (AMT). In total, we collect 2,620 annotations from 77 participants. In 39.14 ± 1.96% of trials the adversarial images generated by semanticAdv are selected as realistic images and in 30.27 ± 1.96% of trials, the adversarial images generated by CW are selected as realistic images. It indicates that our semantic adversarial examples are more perceptual realistic than CW.
4.3 SemanticAdv against Defense Methods

We evaluate the strength of the proposed attack by testing against four existing defense methods, namely, Feature squeezing [Xu et al., 2017], Blurring [Li and Li, 2017], JPEG [Dziugaite et al., 2016] and AMI [Tao et al., 2018]. For AMI [Tao et al., 2018], we first extract attribute witnesses with our aligned face images and then leverage them to construct attribute-steered model. We use fc7 of pretrained VGG [Parkhi et al., 2015] as the face representation. AMI yields a consistency score for each face image to indicate whether it is a benign image. The score is measured by the cosine similarity between the representations from original model and attribute-steered model. With 10% false positives on benign inputs, it only achieves 8% detection accuracy for semanticAdv and 12% detection accuracy for CW.

Figure 5 illustrates semanticAdv is more robust against these defense methods comparing with CW. The same G-FPR and T-FPR are used for evaluation. Under the condition that T-FPR is $10^{-3}$, both semanticAdv and CW achieve high attack success rate, while semanticAdv marginally outperforms CW when FPR goes down to $10^{-4}$. While defense methods have proven to be effective against CW attacks on classifiers trained with ImageNet [Krizhevsky et al., 2012], our results indicate that these methods are still vulnerable in face verification system with small T-FPR.

4.4 SemanticAdv on Landmark Detection

We also evaluate the effectiveness of semanticAdv on Face Landmark Detection, which is structure based prediction model. We select two attack targets, “Rotating Eyes” and “Out of Region”. “Rotating Eyes” means we rotate the coordinates of the eyes in the image counter-clockwise by 90°; “Out of Region” means that we set a target bounding box and push all points out of the box. The evaluation metrics and overall attack success rate among different attributes have been shown in the supplementary materials [2.3]. We find that our method can also attack landmark detection successfully. Figure 6 illustrates the adversarial examples on landmark detection.

5 Conclusions

In this paper, we presented a novel attack method called SemanticAdv capable of generating structured adversarial perturbations guided by semantic attributes. Compared to existing methods, our proposed method works in a more controllable fashion, which allows for detailed analysis. Experimental evaluations on face verification and landmark detection demonstrated several nice properties including attack transferability and attack effectiveness against existing defense methods. We believe this work could potentially open up great research opportunities and challenges in the field of adversarial learning in the long run.
Acknowledgement This work was supported in part by the National Science Foundation under Grant CNS-1422211, CNS-1616575, IIS-1617767, DARPA under Grant 00009970, and Google PhD Fellowship to X. Yan.

References
Alibaba Cloud Computing Co. Ltd. https://help.aliyun.com/knowledge_detail/53535.html.
Megvii technology co. ltd. https://console.faceplusplus.com/documents/5679308.
Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai. Better mixing via deep representations. In ICML, 2013.
A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth. Big but imperceptible adversarial perturbations via semantic manipulation. arXiv preprint arXiv:1904.06347, 2019.
A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. In ICLR, 2019.
T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. Christiano, and I. Goodfellow. Unrestricted adversarial examples. arXiv preprint arXiv:1809.08352, 2018.
A. Bulat and G. Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In Proceedings of the IEEE International Conference on Computer Vision, pages 3706–3714, 2017a.
A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In ICCV, 2017b.
N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (S&P). IEEE, 2017.
Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In CVPR, 2018.
M. Cisse, Y. Adi, N. Neverova, and J. Keshet. Houdini: Fooling deep structured prediction models. In NIPS, 2017.
R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160–167. ACM, 2008.
L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. L. Seltzer, G. Zweig, X. He, J. D. Williams, et al. Recent advances in deep learning for speech research at microsoft. In ICASSP, volume 26, page 64, 2013.
G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. A study of the effect of jpg compression on adversarial images. arXiv preprint arXiv:1608.00853, 2016.
L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. arXiv preprint arXiv:1712.02779, 2017.
A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In CVPR. IEEE, 2009.
I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014a.
I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. Deep learning, volume 1. MIT Press, 2016.
I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In ICLR, 2014b.
Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In ECCV. Springer, 2016.
K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
S. Hong, X. Yan, T. S. Huang, and H. Lee. Learning hierarchical semantic image manipulation through structured representations. In NeurIPS, 2018.

P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In CVPR, pages 1125–1134, 2017.

J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV. Springer, 2016.

J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In CVPR, pages 1219–1228, 2018.

T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In ICLR, 2018.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In ICLR, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In ICLR, 2014.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In ICCV. IEEE, 2009.

X. Li and F. Li. Adversarial examples detection in deep networks with convolutional filter statistics. In Proceedings of the IEEE International Conference on Computer Vision, pages 5764–5772, 2017.

Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. In CVPR, 2017.

M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In NIPS, 2017.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In ICCV, 2015.

A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In CVPR, 2015.

E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov. Generating images from captions with attention. In ICLR, 2015.

S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2574–2582, 2016.

A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In European Conference on Computer Vision, pages 483–499. Springer, 2016.

A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In ICML. JMLR, 2017.

A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In ICML, 2016.

N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In Security and Privacy (EuroS&IP), 2016 IEEE European Symposium on, 2016.

D. Parikh and K. Grauman. Relative attributes. In ICCV. IEEE, 2011.

O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In bmvc, volume 1, page 6, 2015.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2015.

S. Reed, K. Sohn, Y. Zhang, and H. Lee. Learning to disentangle factors of variation with manifold interaction. In ICML, 2014.

S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In ICML, 2016.
C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV Workshop*, 2013.

P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *CVPR*, 2017.

Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *CVPR*, 2017.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lancot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

G. Tao, S. Ma, Y. Liu, and X. Zhang. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *NeurIPS*, 2018.

A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *NIPS*, 2016.

H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018a.

T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018b.

W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays. Texturegan: Controlling deep image synthesis with texture patches. In *CVPR*, 2018.

C. Xiao, R. Deng, B. Li, F. Yu, M. Liu, and D. Song. Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In *ECCV*, 2018a.

C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks. In *IJCAI*, 2018b.

C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song. Spatially transformed adversarial examples. In *ICLR*, 2018c.

C. Xiao, D. Yang, B. Li, J. Deng, and M. Liu. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019.

W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *ECCV*. Springer, 2016.

H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

X. Zhang, L. Yang, J. Yan, and D. Lin. Accelerated training for massive classification via dynamic class selection. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *ECCV*. Springer, 2016a.
J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In ICCV, 2017.

X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In CVPR, 2016b.
A Implementation Details

A.1 Experiment Settings of Identity Verification

In order to better understand our identity verification models, Table A shows the benchmarks of identity verification system.

In order to generate adversarial examples by our proposed semanticAdv, we adopt Adam [Kingma and Ba, 2015] as the optimizer. When G-FPR >= $10^{-4}$, we set max iterations to 200 with a start learning rate 0.05. When G-FPR < $10^{-4}$, we set max iterations to 500 with a start learning rate 0.01. For CW, we add a reconstruction loss and set the corresponding loss weight to 5. The max iterations is 1,000 with a start learning rate 0.001. Table B shows the threshold for determining whether two images belong to the same identity.

| $M$ / benchmarks | LFW | AgeDB-30 | CFP-FF | CFP-FP |
|-------------------|-----|----------|--------|--------|
| R-50-S            | 99.27 | 94.15 | 99.26 | 91.49 |
| R-101-S           | 99.42 | 95.93 | 99.57 | 95.07 |
| R-50-C            | 99.38 | 95.08 | 99.24 | 90.24 |
| R-101-C           | 99.67 | 95.58 | 99.57 | 92.71 |

Table A: Identity Verification Benchmarks.

| FPR / $M$ | R-50-S | R-101-S | R-50-C | R-101-C |
|-----------|--------|---------|--------|---------|
| $10^{-3}$  | 1.181  | 1.244   | 1.447  | 1.469   |
| $3 \times 10^{-4}$ | 1.058 | 1.048 | 1.293 | 1.242 |
| $10^{-4}$  | 0.657  | 0.597   | 0.864  | 0.809   |

Table B: Thresholds of Identity Verification.

A.2 Experiment Settings of the Defense Methods.

Feature squeezing is a simple but effective method by reducing color bit depth to remove the adversarial effects. We compress the image represented by 8 bits for each channel to 4 bits for each channel to evaluate the effectiveness. For Blurring, we use a 3x3 Gaussian kernel with standard deviation 1 to smooth the adversarial perturbations. For JPEG, it leverages the compression and decompression to remove the adversarial perturbation. We set compression ratio as 0.75 in our experiment.

A.3 Experiment Settings of Landmark Detection

The max iterations is set to 2,000 with a start learning rate 0.05. The balancing factor $\lambda$ is set to 0.001 and 0.01 for “Rotating Eyes” and “Out of Region” respectively.

Evaluation Metrics. We apply two different metrics for two adversarial attack tasks respectively. For “Rotating Eyes” task, we use the NME (Normalized Mean Error) [Bulat and Tzimiropoulos, 2017b], a standard evaluation metric in landmark detection, as the evaluation metric.

$$\text{NME} = \frac{1}{N} \sum_{k=1}^{N} \frac{||\mathbf{p}_k - \hat{\mathbf{p}}_k||_2}{d},$$

(7)

where $\mathbf{p}$ denotes the ground-truth landmarks, $\hat{\mathbf{p}}$ denotes the predicted landmarks and $d = \sqrt{W_B \times H_B}$ is the square-root of area of ground-truth bounding box, where $W_B$ and $H_B$ represents the width and height of the box.

For “Out of Region” task, we define a metric to evaluate the capability of attack method to fool the landmark detection network to yield predictions outside a pre-defined area. The metric is computed as

$$\text{RATE} = \frac{N_{\text{out}}}{N_{\text{total}}},$$

(8)

where $N_{\text{out}}$ denotes the number of predicted landmarks outside the pre-defined bounding box and $N_{\text{total}}$ denotes the total number of landmarks.
A.4 Experiment Settings of User Study

We conducted a user study on the adversarial images of semanticAdv and CW used in experiment of API-attack and the original images. We present a pair of original image and adversarial image to the participant and ask them to rank the two options. The order of these two images was randomized and the images are appeared for 2 seconds in the middle of the screen during each trial. After disappearing, the participant has unlimited time to select the more realistic image according to their perception. For each participant, one could only conduct at most 50 trials, and each adversarial image was shown to 5 different participants. In total, we collect 2,620 annotations from 77 participants. In 39.14 ± 1.96% of trials the adversarial images generated by semanticAdv are selected as realistic images and in 30.27 ± 1.96% of trials, the adversarial images generated by CW are selected as realistic images. It indicates that our semantic adversarial examples are more perceptual realistic than CW.

B Additional Analysis

B.1 Interpolation Analysis

We analyze attacking performance among different layers. We apply single attribute attack of semanticAdv on the last three layers before the up-sampling. Table C shows the attack success rate on R-101-S. Here, \( f_i \) denotes the \( i \)-th layer after the residual blocks. The result demonstrates that \( f_0 \) achieves the highest success rate, thus it is taken as the default setting in the following experiments.

| Layer (\( f_i \)) | \( f_0 \) | \( f_1 \) | \( f_2 \) | \( f_0 \) | \( f_1 \) | \( f_2 \) |
|-----------------|---------|---------|---------|---------|---------|---------|
| Attack Success Rate | 99.29 | 98.32 | 75.62 | 97.35 | 94.10 | 57.15 |

Table C: Attack success rate by selecting different layer’s feature-map for interpolation on R-101-S(%). \( f_i \) indicates the \( i \)-th layer after the residual blocks.

**Table D: Quantitative result of identity verification (%). It shows accuracy of face verification model and attack success rate of semanticAdv and CW. \( \mathbf{x}' \), \( G(\mathbf{x}', \mathbf{c}) \) and \( G(\mathbf{x}', \mathbf{c}_{\text{new}}) \) are the intermediate results of our method before adversarial perturbation.**
B.2 Additional Analysis: Identity Verification

Table D shows overall performance (accuracy) of face verification model and attack success rate of semanticAdv and CW. We denote $x'$, $G(x', c)$ and $G(x', c_{\text{new}})$ as the synthesized image without optimizing the adversarial objective. To evaluate the performance of semanticAdv under different attributes, we devise three metrics as follows: (1) Best means if there is one attribute among 17 attributes that can be successfully attacked, we count the attack success rate for this person as 1. (2) Average means we calculate the average attack success rate among 17 attributes for a person. (3) Worst means only when all of 17 attributes can be successfully attacked, we count the attack success rate for this person as 1. As CW is the traditional pixel-wise attack method independent of the semantic attributes, we compared it with the Best metric.

The generated adversarial samples also provide a way to diagnose the face verification model. To analyze the model, we show the benchmarks of the face verification’s models in Table A. The results are shown in Table D; we observe that although the face model trained with cosine loss achieves higher face recognition performance, it is more vulnerable to adversarial attack compared with softmax. As cosine loss introduces large-margin constraints during training, it may draw strong assumption from training distribution. When the distribution of face images is slightly changed, the distribution of face feature varies a lot.

B.3 Additional Results: Semantic Attack Transferrability

In Table E, we present the quantitative results of the transferability with G-FPR = $10^{-4}$, T-FPR = $10^{-4}$. We observe that with the strict criterion (Lower T-FPR) of the verification model, the transferability becomes lower cross different models.

Table E: Transferability analysis: cell $(i,j)$ shows attack success rate of adversarial examples generated against $j$-th model and evaluate on $i$-th model. Results generated with G-FPR = $10^{-4}$, T-FPR = $10^{-4}$

| $M_{\text{test}} / M_{\text{opt}}$ | R-50-S | R-101-S | R-50-C | R-101-C |
|-------------------------------|--------|--------|--------|--------|
| R-50-S | 1.000 | 0.005 | 0.000 | 0.000 |
| R-101-S | 0.000 | 1.000 | 0.000 | 0.000 |
| R-50-C | 0.000 | 0.000 | 1.000 | 0.000 |
| R-101-C | 0.000 | 0.000 | 0.000 | 1.000 |

B.4 SemanticAdv on Landmark detection

In Table F, we present the quantitative results of SemanticAdv on Landmark detection systems. We have two adversarial targets “Out of Region” and “Rotating Eyes”. We observe that our method is efficient to perform attacking on Landmark detection models. For some attributes such as “Eyeglasses”, “Pale Skin”, SemanticAdv can achieve really high performance.

Table F: Adversarial attack on landmark detection(%). Lower is better.