Can Perceptual Guidance Lead to Semantically Explainable Adversarial Perturbations?

Charantej Reddy Pochimireddy, Aditya T. Siripuram, and Sumohana S. Channappayya, Senior Member, IEEE

Abstract—It is well known that carefully crafted imperceptible perturbations can cause state-of-the-art deep learning classification models to misclassify. Understanding and analyzing these adversarial perturbations play a crucial role in the design of robust convolutional neural networks. However, their mechanics are not well understood. In this article, we investigate the following question: Is there any relationship between the imperceptibility and semantic significance of adversarial perturbations? In other words, can adversarial perturbations change only the regions of the image that are important for classification, while still being imperceptible? To address this, we consider generating adversarial perturbations by explicitly incorporating a “perceptual quality ball” constraint in our formulation. We pose this adversarial example generation problem as a tractable convex optimization problem, with constraints taken from a mathematically amenable variant of the popular SSIM index. We also provide a fast approximation algorithm to solve the proposed convex optimization problem, bringing our technique in line with other \( l_p \) based techniques like PGD, both in form and computational complexity. We use the MobileNetV2 network trained on the ImageNet dataset for our experiments. By comparing the SSIM maps generated by our method with class activation maps, we observe that the perceptually guided perturbations generated by the proposed technique introduce changes specifically in the regions that contribute to classification decisions i.e., the proposed method induces changes that are semantic.

Index Terms—SSIM, adversarial perturbations, explainability.

I. INTRODUCTION

Analyzing the robustness of deep learning models to adversarial inputs has received significant attention over the past few years. A majority of these works can be classified into four broad areas: 1) generating adversarial examples/crafting adversarial perturbations, 2) defending against such ‘attacks’ [1], [2], 3) evaluation and certification [3], [4], and 4) interpretability of adversarial examples/perturbations [5]. However, we still lack a clear understanding of the key underlying factors for the existence of this phenomenon. Because of this, it is challenging to design defense mechanisms that work for all kinds of adversarial attacks/perturbations. The adversarial example generation problem is highly non-convex, and heuristic or other approximation techniques are required to solve it.

In their seminal work, Szegedy et al. introduced adversarial examples in [6] that are crafted by solving an optimization. Later, faster adversarial perturbation methods with closed-form solution (a direct expression to generate perturbations) were proposed by using the norm bounded constraints; examples include FGSM \((l_{\infty} \text{ bounded perturbations})\) [7], FGM \((l_2 \text{ bounded perturbations})\) [8], and PGD \((l_{\infty} \text{ and } l_2 \text{ bounded perturbations})\). Several attack and defense algorithms have since been proposed, and many of these methods are summarized in [1], [2]. Most existing defences are not completely robust: adversarial attacks/perturbations can be specifically engineered to target the said defense technique. This has led to a self-sustaining cycle - with attacks leading to defenses, and defenses paving the way for new attacks. Despite the many results in this area, a complete understanding of the adversarial perturbation landscape is still lacking. Some proposed hypotheses for explaining adversarial perturbations include high dimensional spaces [7], data in-completion, model capacity [1], and the presence of highly predictive but non-robust features (spurious correlations) [5]. The authors in [9] proposed that the distribution shift between training and test data set, combined with the high dimensional continuous data space, are the key reasons for adversarial examples.

Along these lines, our interest is not to create another attack but rather to understand the adversarial landscape, primarily with respect to perceptual similarity. Our investigation aims to systematically explore the semantic significance of the regions affected by adversarial perturbations. In particular, we try to answer the following question: Is there any relationship between the imperceptibility and semantic significance of adversarial perturbations? We answer this question partially by proposing a technique to generate imperceptible adversarial perturbations that correlate highly with discriminative image regions.

In most of the adversarial example attacks, \( l_p \) distance metrics are used for crafting the adversarial perturbations. However, it is well-known that \( l_p \) distance metrics are not good at measuring perceptual similarity between two images [10], and that these metrics are neither necessary nor sufficient for perceptual similarity [11]. For example, an uniform shift in the image results in a high \( l_p \) distance, but perceptually, the uniformly shifted image remains close to the original image. For the perturbations to be imperceptible, the original and modified image must be similar in some metric that respects structural (perceptual) similarity. Thus to answer the question above, we use the Structural
Similarity (SSIM) index [10] to generate the adversarial perturbations instead of \( l_p \) distance metrics. The SSIM index is a popular measure to evaluate the perceptual similarity/quality of images. It is a full-reference image quality metric that measures the quality of a distorted image with respect to the ground-truth pristine image. Compared to traditional \( l_p \) distance metrics, the SSIM index is known to correlate better with the human perception of image quality/distortion. The following are the main contributions of this work:

1) In order to address the question posed earlier, we investigate a perceptually guided modification to the existing \( l_p \) norm based attack frameworks. By leveraging the useful mathematical properties of the SSIM index, we construct a novel convex formulation (which we call perceptually guided adversarial perturbation (PGAP)) to generate the adversarial perturbations. We also provide a closed-form approximation (called Faster PGAP (FPGAP)) to the proposed convex problem.

2) We then try to systematically investigate (both qualitatively and quantitatively) adversarial examples and their relationship to semantically significant regions of the image. We do this by comparing the perturbations generated using the SSIM index with class activation maps generated using GradCAM++ [12]. We quantify this comparison using IOU-based metrics and precision; and observe that compared to other norm-bounded approaches, our method gives about a factor of 2-3 higher precision scores (Fig. 5).

3) We thus provide evidence for the connection between imperceptibility and semantic explainability of adversarial perturbations, partially answering the question posed in the introduction. Adversarial perturbations generated by incorporating an SSIM ball constraint (instead of the \( l_p \) ball constraint as in other works) seem to be changing only the regions of the image significant for classification, at least to a degree much higher than other norm-bounded attack methods.

We find it very intriguing that a perceptually-aware formulation makes the adversarial example generation semantically-aware.

II. RELATED WORK

The SSIM index has been considered previously in the adversarial perturbation setting, and we briefly summarize two relevant methods next. In [13], the authors proposed stronger attacks by combining different types of attacks (adding adversarial noise, rotating, translating, or performing spatial transformations on images) and used the SSIM index to quantify the strength of the adversarial attack. In [14], the authors introduced a new measure called Perceptual Adversarial Similarity Score (PASS) using the SSIM index to quantify the adversarial examples and use the PASS in the process of generating adversarial examples.

Our contribution differs significantly from both of these methods, which we outline below. By imposing constraints from a mathematically amenable variant of the SSIM index [15] (and exploiting useful properties like quasi convexity), and by taking a suitable approximation of the loss function, we pose the adversarial example generation problem as a quadratically constrained quadratic program (QCQP). We also provide a closed-form solution to the approximation of the QCQP that allows for faster implementation. In fact, none of the existing adversarial example techniques that use image quality metrics provide a closed-form solution to (1). We not only give a convex formulation (9), but also provide a closed-form approximation (10). Thus our method is in line with other norm bounded techniques (PGD, FGSM, etc.), which employ a closed-form solution to generate adversarial perturbations. The perturbations we generate are not necessarily additive, nor are they obtained by a parameterized rotation or spatial transformation; we propose a model-free technique to generate adversarial examples that are structurally and thereby perceptually similar to the original image.

While this manuscript was in development, we also became aware of another line of work [16] that uses SSIM to generate adversarial examples. However, our work differs in the following key aspects: First, our goal, as opposed to the work in [16], is to investigate the question posed in the introduction, and not simply to generate another attack. As such, we have a detailed and systematic analysis of the regions of perturbations as compared to the regions identified by GradCAM++, which are not present in [16]. Secondly, we provide a tractable convex formulation of the adversarial example generation problem. We use the analysis from [15] primarily to achieve this goal. Thirdly, in addition to the convex formulation, we also provide a fast approximation to the convex formulation (FPGAP), which brings our technique (computationally and structurally) in line with existing norm-bounded methods. This is in contrast to the gradient-descent-based Lagrangian optimization used in [16] (as a consequence, our method also does not need any additional hyper parameters). This fast approximation also allows us to perform experiments on datasets like ImageNet, which would not be possible with the initial formulation. Indeed such experiments are not done in [16]. Finally, we also provide a systematic analysis of the comparative impact of the hyper-parameters involved in our algorithm (\( \epsilon_1 \) and \( \epsilon_2 \) in (8)).

In [17], the authors use LPIPS distance to design attack/defence mechanism. The LPIPS metric uses features from trained network to evaluate the quality, which is not required in our framework. In [18], the authors considers feature distance at different layers outputs as perceptual measure. We note that neither of these are closed-form solutions. Our work differs from recent contributions in that we consider a classical image quality metric in our formulation, provide a closed-form solution to the adversarial example generation problem and thoroughly analyze the semantic significance of the resulting adversarial perturbations by comparing with class activation maps.

A. Organization

The paper is organized as follows: in Section III we briefly discuss the standard norm bounded adversarial example generation methods and formulate an optimization problem (2). In Section
III-A, we explain the SSIM index along with its mathematically amenable variant and discuss some of its properties which will be used to generate adversarial examples using SSIM constraints. In Section IV, we use ideas from III-A to modify (2) to formulate an optimization problem (9) to generate adversarial examples using SSIM index. We also propose an approximate solution (10) to this optimization problem which enables the faster generation of adversarial examples. In Section V, we analyze the norm bounded methods along with the proposed method qualitatively and quantitatively and present some additional results on the perceptual quality of adversarial examples generated by different methods. We conclude the paper with some closing remarks in Section VI.

III. PROBLEM SETUP

Suppose $\mathcal{D}$ represents the data set with entries $(x_i, y_i)$, where $x_i$ represents the $i$th data point and $y_i$ represents the corresponding label. Let $f$ denote the machine learning model and its prediction $\hat{y}$ (i.e., $y_i = f(x_i)$). In the supervised learning framework, we try to minimize the loss function $\mathcal{L}$ with respect to given data set $\mathcal{D}$ and find the best suited model parameters $w$.

$$\min_w \sum_i \mathcal{L}(w, x_i, y_i)$$

An adversarial example $x_{adv}$ is similar (in some metric) to a data point $x$ in the data set $\mathcal{D}$, such that the machine learning models misclassifies the input (i.e., $f(x_{adv}) \neq f(x)$). Finding an adversarial example at a given data point $x$ with label $y$ can be formulated as an optimization problem based on loss function. In this formulation, we find $x_{adv}$ close to a given data point $x$, which maximizes the loss function

$$\arg\max_{x_{adv}} \mathcal{L}(w, x_{adv}, y) \text{ s.t. } d(x, x_{adv}) \leq \epsilon. \quad (1)$$

Here $\epsilon$ is the allowed level of perturbation. We refer to $x_{adv}$ as adversarial example and $x - x_{adv}$ as adversarial perturbation.

Note that the feasibility region of the optimization problem above (1) varies based on the value of $\epsilon$. The formulation in (1) is similar to the inner maximization problem from [19]. The distance metric $d(x, x_{adv})$ is typically an $l_p$ distance

$$d(x, x_{adv}) = ||x - x_{adv}||_p \text{ where } ||a||_p = \left( \sum_i |a_i|^p \right)^{\frac{1}{p}}.$$

Typically, a linear approximation of the loss function around the data point $x$ is used:

$$\mathcal{L}(w, x_{adv}, y) \approx \mathcal{L}(w, x, y) + (x_{adv} - x)^T \nabla_x \mathcal{L}(w, x, y).$$

So to find $x_{adv}$ we maximize the second term above. Note that since

$$\arg\max_{x_{adv}} \mathcal{L}(w, x_{adv}, y) \approx \arg\max_{x_{adv}} (x_{adv} - x)^T \nabla_x \mathcal{L}(w, x, y),$$

we can just maximize the inner product between $x_{adv}$ and $\nabla_x \mathcal{L}(w, x, y)$. Also note that the gradients $\nabla_x \mathcal{L}(w, x, y)$ can be readily extracted from the model.

Important distance metrics used in the literature are $l_\infty$, which measures the maximum absolute change in the pixel values [7], [20], [21]; $l_2$ which measures the Euclidean distance of change in the pixel values [21], [8]; $l_1$ measures the total absolute change in the pixel values [22], and $l_0$ measures the number of pixels that differ [21], [23], [24].

As discussed earlier, $l_p$ distance metrics are not good at measuring perceptual similarity between two images. If we consider $d(x, x_{adv}) = \sqrt{1 - \text{SSIM}(x, x_{adv})}$, then the optimization problem for generating adversarial example becomes:

$$\arg\max_{x_{adv}} (x_{adv})^T \nabla_x \mathcal{L}(w, x, y) \text{ s.t. } \text{SSIM}(x, x_{adv}) \geq 1 - \epsilon^2 \quad (2)$$

However, the metric $d(x, x_{adv}) = \sqrt{1 - \text{SSIM}}$ defined above is non-convex, and so the above problem can become intractable. So we use a variant of the SSIM index [15] to generate imperceptible adversarial perturbations. We review these ideas next. Readers familiar with the SSIM index can skip to Section IV.

A. Structural Similarity (SSIM) Index

The SSIM index between the two image patches $X$ and $Y$ is computed using a combination of three distortion measurement components: luminance ($l$), contrast ($c$), and structure/correlation ($s$), that are defined as follows.

$$l = \frac{2\mu_X \mu_Y + c_1}{\mu_X^2 + \mu_Y^2 + c_1}, \quad c = \frac{s_X s_Y + c_2}{s_X^2 + s_Y^2 + c_2}, \quad s = \frac{c s_X s_Y + c_3}{c s_X s_Y + c_3}$$

where $\mu_X, \mu_Y$ represent the mean of $X$ and $Y$ respectively, $s_X^2, s_Y^2$ represent the variances of $X$ and $Y$ respectively, and $s_{XY}$ represents the co-variance between the $X$ and $Y$. Here, $c_1, c_2$ and $c_3$ are small numerical constants that ensure numerical stability when the denominators are close to zero. We can also say that these constants aim to characterize the saturation effects of the visual system at low luminance and contrast regions. The first two terms $l$ and $c$ measure nonstructural distortion, while the last term $s$ measures structural distortion (or absence of correlation) between the two images. The structural similarity or SSIM between the images $X$ and $Y$ is defined as the product of the luminance, contrast and structure terms defined above, i.e., $\text{SSIM}(X, Y) = l.c.s.$

An SSIM quality map is constructed by computing the SSIM index between pairs of corresponding local patches in the two images, and the overall SSIM index is computed by averaging the patch level values in the SSIM map.

While the SSIM index is indeed a better method compared to MSE for measuring perceptual similarity between two images, it does not satisfy the triangle inequality and thus is not a distance metric, limiting its use in convex problem formulations. However, the SSIM index can be converted to a normalized root mean square error (NRMSE) measure, which is a valid distance metric [15]. The square of such a metric is not convex but is locally convex, and quasi-convex [15], thereby making the SSIM index a feasible target for optimization. We use these insights for our problem formulation. Next, we briefly review these ideas, develop the notation.
In the standard form of the SSIM index, we set the numerical constants $c_3 = c_2/2$, resulting in the SSIM having only two terms

$$SSIM(X, Y) = S_1(X, Y) \cdot S_2(X, Y),$$

where $S_1 = l$ and

$$S_2 = c.s = \frac{2s_{X,Y} + c_2}{\left(s_X^2 + s_Y^2 + c_2\right)}.$$

It can be seen that $\sqrt{1 - SSIM}$ is not a metric but $\sqrt{1 - S_1}$ and $\sqrt{1 - S_2}$ are normalized metrics [15]. Now we set $d_1 = \sqrt{1 - S_1}$ and $d_2 = \sqrt{1 - S_2}$, and the vector $d = [d_1, d_2]$. It can be seen that $d$ is a vector of normalized metrics obtained from the root mean square error [15].

The SSIM index can be approximated with the vector of metrics $d(X, Y)$ as

$$||d(X, Y)||_2 = \sqrt{(d_1)^2 + (d_2)^2} = \sqrt{2 - S_1 - S_2}.$$ (4)

We note that

$$\sqrt{1 - SSIM} = \sqrt{1 - S_1} \cdot S_2 = \sqrt{d_1^2 + d_2^2 - d_1^2 d_2^2}. $$ (5)

We observe that $||d(X, Y)||_2$ serves as a lower order approximation of $\sqrt{1 - SSIM}$. We can also write

$$S_1 = 1 - \text{NMSE}(\mu_X, \mu_Y, c_1)$$

$$S_2 = 1 - \text{NMSE}(X - \mu_X, Y - \mu_Y, c_2),$$

where NMSE is the normalized mean squared error given by

$$\text{NMSE}(X, Y, c) = \frac{||X - Y||^2}{||X||^2 + ||Y||^2 + c}.$$ (7)

We use these ideas to modify problem (2).

IV. ADVERSARIAL EXAMPLE GENERATION USING SSIM

The structural similarity index can be written as a product of two terms i.e., $SSIM = S_1 \cdot S_2$ where $S_1$ captures the luminance similarity, and $S_2$ captures the structure and contrast similarity. From [15] (summarized in Section III-A above) we know that $S_1$ and $S_2$ have appealing convexity properties. Going back to the problem formulation of (2), we see that the condition on $SSIM(x, x_{adv})$ in (2) can be replaced by conditions on $S_1(x, x_{adv})$ and $S_2(x, x_{adv})$, as discussed in (3). By choosing $\epsilon_1$ and $\epsilon_2$ suitably, we may rewrite the optimization problem from (2) as

$$\arg\max_{x_{adv}} (x_{adv})^T \nabla_x L(w, x, y)$$

s.t. $S_1(x, x_{adv}) \geq 1 - \epsilon_1^2$ \hspace{1cm} (8)

$s_2(x, x_{adv}) \geq 1 - \epsilon_2^2$ \hspace{1cm} (8)

We analyse the constraints of (8) in more detail in Appendix A.

The first constraint $S_1(x, x_{adv}) \geq 1 - \epsilon_1^2$ is a linear constraint (this forces $x_{adv}$ to lie in an intersection of two half-spaces). The second constraint in (8) $S_2(x, x_{adv}) \geq 1 - \epsilon_2^2$ is a quadratic constraint (this forces $x_{adv}$ to be in a high dimensional sphere).

Based on the constraints (12), (13), and the objective, the optimization problem in (8) is convex; in particular it is a Quadratically Constrained Quadratic Program (QCQP).

Conceptually, Constraint 1 corresponds to non-structural perceptual features (luminance), and Constraint 2 corresponds to structural/perceptual features in the image. The parameters $\epsilon_1$ and $\epsilon_2$ fix the allowed tolerances in these features and decide the feasibility region of the optimization problem. To understand the relative impact of $\epsilon_1$ and $\epsilon_2$, we take an image with label $y = 4$ from the MNIST digits test data set and generate adversarial examples for different values of $\epsilon_1$ and $\epsilon_2$. For this exercise, we use a CNN model, which has around 99% accuracy on the test data. In Fig. 1, the image on the left shows the model prediction of optimization problem output and its SSIM index with respect to original image, as a function of $\epsilon_1$ and $\epsilon_2$. On the right, the corresponding output images are shown. From Fig. 1 one important observation that can be made is that the impact of $\epsilon_2$ (Constraint 2) is more on the solution (model prediction and perceptual quality of optimization output) compared to the $\epsilon_1$ (Constraint 1). This observation can also be validated by calculating the corresponding optimal dual variables. For the above example in Fig. 2, the dual variable corresponding to Constraint 2 is around 145 times larger compared to the dual variable of Constraint 1. Since $\epsilon_2$ has significantly higher impact compared to $\epsilon_1$, the algorithm can be simplified by taking $\epsilon_1 = 0$, effectively removing the Constraint 1 above. This modification results in solving the following optimization problem:

$$\arg\max_{x_{adv}} (x_{adv})^T \nabla_x L(w, x, y)$$

s.t. $\mu_{x_{adv}} = \mu_x$ \hspace{1cm} (9)

$$S_2(x, x_{adv}) \geq 1 - \epsilon_2^2$$

A. Proposed Method – Perceptually Guided Adversarial Perturbation (PGAP)

In practice, solving (9) may not lead to an adversarial example. This is because the formulation of (8) assumed a linear approximation to the loss function, which may be inaccurate. Along the lines of [19], we propose an iterative technique that repeatedly solves (9). We first fix an $\epsilon_2$ and solve (9). Note that at this point the obtained solution $x_{adv}$ may not be adversarial (i.e., we may not have $f(x_{adv}) \neq f(x)$). We recalculate the gradients at the obtained $x_{adv}$ and solve (9) using the updated gradient. This process is repeated until an adversarial example is found (see Algorithm 1 for a summary). The iterative approach helps in overcoming the limitations imposed by the linear approximation to the loss function.

One drawback of the proposed method (Algorithm 1) is that the QCQP is slow to solve on large datasets (for e.g., ImageNet). Hence next, we present a faster algorithm that uses an efficient approximation of the solution to (9).

B. Approximate Solution – Faster PGAP

We formulate an equivalent optimization problem from (9) by relaxing Constraint 1 and substituting it in Constraint 2, and converting it into a minimization problem. The solution to this
Fig. 1. Image on the left: x-axis and y-axis correspond to $\epsilon_1$ and $\epsilon_2$ respectively, at each highlighted point on the grid has label with first co-ordinate as model prediction and second co-ordinate as ssim value. Image on the right: corresponding images with $\epsilon_1$ and $\epsilon_2$ mentioned on top of them.

Fig. 2. Image on the left: original test image with label $y = 4$ and model prediction $\hat{y} = 4$. Image on the right: adversarial example generated by solving (8) ($\epsilon_1 = 0.05$ and $\epsilon_2 = 0.11$) with perceptual quality 0.95 (SSIM).

**Algorithm 1:** Adversarial example generation.

```plaintext
PGAP(x, model, label, $\epsilon_2$, iterNum)

i = 0

While $i \leq$ iterNum do

x_adv ← argmax$_{x_{adv}}$ $(x_{adv})^T \nabla_x L(w, x, y)$

s.t. $\mu_{x_{adv}} = \mu_x, S_2(x_{adv}, x) \geq 1 - \epsilon_2^2$

y_adv ← model.predict(x_adv)

If label $\neq$ y_adv

return x_adv

else

x = x_adv

return x_adv
```

The problem is given by:

$$x_{adv} = 1(\mu_x) + k_{22} + (\sqrt{k_{21}}) \left( \frac{\nabla_x L(w, x, y)}{||\nabla_x L(w, x, y)||} \right), \quad (10)$$

where $k_{21}, k_{22}$ are defined in (13) and 1 in $1(\mu_x)$ is all ones of size $x_{adv}$, thus providing a closed form solution to (14). We refer the reader to Appendix B for the intermediate steps.

**Algorithm 2:** Faster adversarial example generation.

```plaintext
FPGAP(x, model, label, $\epsilon_2$, iterNum)

i = 0

While $i \leq$ iterNum do

x_adv ← $1(\mu_x) + k_{22} + (\sqrt{k_{21}}) \left( \frac{\nabla_x L(w, x, y)}{||\nabla_x L(w, x, y)||} \right)$

y_adv ← model.predict(x_adv)

If label $\neq$ y_adv

return x_adv

else

x = x_adv

return x_adv
```

Algorithm 2 presents the steps of FPGAP (a fast approximate variant of PGAP) by incorporating the closed-form solution above in the iterations.

**C. Some Remarks**

We would like to point out that the structure of the presented approximate (10) solution is very similar to adversarial examples generated by FGM [8] attack, which generates $l_2$ norm bounded additive perturbations. The expression for adversarial examples given by FGM is

$$x_{adv} = x + \epsilon \left( \frac{\nabla_x L(w, x, y)}{||\nabla_x L(w, x, y)||} \right). \quad (11)$$

Both the formulations (10) and (11) use normalized gradient $\nabla_x L(w, x, y)/||\nabla_x L(w, x, y)||$. However the significant difference in the performance of the proposed attack is due to the presence of the terms $k_{21}$ and $k_{22}$ which are derived from the structure term in the SSIM index. It is very intriguing that such a simple change results in a huge improvement in the precision scores (Section V-B) and fooling rate (Section V-C).

An alternate approach to find adversarial examples with high SSIM could be to subtract a differentiable version of SSIM from the loss function, similar to other such approaches in...
Note that the existing norm bounded approaches impose a norm ball constraint on the perturbation. This allows for the intermediate formulation (as in (1)) to be interpreted as a dual norm, hence enabling a closed-form solution. This would not be possible if the norm is added directly to the loss function instead. Our formulation follows a similar philosophy, where we impose a perceptual similarity constraint instead of subtracting the SSIM from the loss function. Specifically, the mathematically amenable variant of SSIM allows us to derive a closed-form solution in this perceptually guided setting as well. As in the $l_p$ norm case, adding the SSIM index directly to the loss term does not admit a closed-form solution. In addition, the resulting non-convex formulation may also lead to increased complexity and convergence-related issues.

V. RESULTS AND DISCUSSION

As discussed earlier, our key goal is to analyze the image regions which the adversarial perturbations affect the most. In particular, our interest is more on understanding the impact of adversarial perturbations on the structurally (perceptually) important regions of the image. We first start with a qualitative analysis with some illustrative examples and then quantify our observations using some well-known metrics.

1) Model used: We use the MobileNetV2 [26] architecture with pre-trained weights, which has 72% top-1 accuracy on the ImageNet validation set [27] for quantitative and qualitative analysis. All the algorithms (PGAP, FPGAP, and the competing methods) are applied to this model to generate adversarial perturbations, one for each image in the ImageNet validation set.

2) Methods being compared: We compare our methods (PGAP, FPGAP) with other popular norm-bounded techniques—FGSM ($\infty$ bounded perturbations) [7], FGM ($l_2$ bounded perturbations) [8], and PGD ($l_\infty$ and $l_2$ bounded perturbations) [19].

3) Techniques used: For each of the methods (PGAP, FPGAP, etc.), we generate the SSIM maps between the adversarial examples and original images. As discussed in Section III-A, SSIM maps [10] provides a visualization of the distortions in the test image with respect to the reference image. Lighter regions in these maps correspond to lower distortion, while darker regions correspond to higher distortion levels.

In order to obtain the semantically important regions of an image, we use Class Activation Maps. In particular, we use GradCAM++ [12]. The heat maps generated by GradCAM++ provide spatial regions that are most important for classification.

4) Hyper-parameter tuning: Considering that each of the involved algorithms has different hyperparameters, we propose the following setup to compare them on an even ground. Consider the following Fooling Rate (FR) score:

$$\frac{1}{N} \sum_{i=1}^{N} 1 \left( \text{pre-attack label}(i) \neq \text{post-attack label}(i) \right),$$

where $N$ is the number of images in the dataset. For a fair comparison, we select the parameters for different methods such that the Fooling Rate is $\approx 1$.

A. Qualitative Analysis

We first provide some illustrative examples that compare the SSIM maps generated by all the methods with GradCAM++ heat maps. For instance, consider Fig. 3, we see that locations of the distortions in the proposed methods (PGAP, FPGAP) agree very well with the GradCAM++ maps. In particular, the number of distortions generated outside the GradCAM++ regions of interest is very few. This, in turn provides evidence that our method does indeed perturb the regions in the image important for classification decisions. We can also observe (from the much lighter maps corresponding to the proposed methods) that the proposed methods introduce far lesser perceptual distortions (or are more localized) compared to other methods.

Similar observation can be made from Fig. 4 that illustrates performance on the MNIST and CIFAR-10 datasets. On MNIST data, the proposed method introduces perturbations in and around the digit present in the image, whereas the perturbations added by the other methods are distributed across the image. On the natural images in the CIFAR-10 dataset, our method is able to carefully identify regions important for classification decisions and introduces perturbations only in such regions.

B. Quantitative Analysis

In this section, we systematically quantify the observations made in the previous section. For this, we consider the following maps

- **Noise map corresponding the adversarial examples**: For each image $i$ in the dataset, we generate a binary image $N_i$ given by

$$N_i = \{(1 - \text{SSIM map}(i)) > \text{noise threshold}\},$$

we evaluate each of the algorithms being compared at various values of noise threshold.

- **Noise map corresponding to GradCAM++**: For each image $i$ in the dataset, we generate the binary image $G_i$ given by

$$G_i = \{\text{GradCAM++ output}(i) > 0.75\}.$$ 

The binary image $N_i$ gives the regions affected by the adversarial perturbations, and the binary image $G_i$ gives the regions that are important for classification. We would like to note here that the GradCAM++ scale is normalized to lie between 0 and 1, with regions contributing to the classification decision taking on higher values. We empirically fixed the threshold on GradCAM++ to 0.75 to extract these important regions for classification (for example, the regions that are highlighted red in GradCAM++ in Fig. 3).

By considering the entries in $G_i$ to be the ground truth, we can use traditional metrics such as precision and IOU (intersection over union). We first compute the True Positives ($TP_i$), this is the number of positive pixels in $N_i$ that are also positive in $G_i$, and the number of false positives ($FP_i$), which is the number of pixels positive in $N_i$ but not in $G_i$. Similarly, we compute the
true negatives ($\text{TN}_i$) and the false negatives ($\text{FN}_i$).

$$\text{TP}_i = \text{sum}(N_i \land G_i) \quad \text{FP}_i = \text{sum}(N_i \land \neg G_i)$$

$$\text{FN}_i = \text{sum}(\neg N_i \land G_i) \quad \text{TN}_i = \text{sum}(\neg N_i \land \neg G_i).$$

Based on these, we evaluate the algorithms on the following metrics:

1) **Precision**: the ratio of number of pixels perturbed in the important regions to the total number of perturbed pixels.

$$\text{Precision} = \frac{1}{N} \sum_i \frac{\text{TP}_i}{(\text{TP}_i + \text{FP}_i)},$$

where $N$ is the total number of images in the dataset. High values of this metric implies that the perturbations are precise in targeting the regions that are sensitive for classification.

2) **Number of pixels perturbed**: This is the fraction of the number of positives in $N_i$, averaged over the dataset.

3) **Recall** ($= \sum_i \frac{\text{TP}_i}{N(\text{TP}_i + \text{FN}_i)}$) gives the ratio of number of pixels perturbed in the important regions to the total number of pixels in the important regions. Note that this metric may not be relevant enough for our setup: as an example, consider a trivial case where the perturbation perturbs all the pixels; in this case, the recall will be high, even though the perturbation is completely agnostic to the structure of the image. Nevertheless, we include this metric for completion.

4) **Intersection over union (IOU)** Similar to other such metrics for image segmentation we can use

$$\text{IOU} = \frac{1}{N} \sum_i \frac{\text{TP}_i}{\text{FP}_i + \text{TP}_i + \text{FN}_i} = \frac{1}{N} \sum_i \text{IOU}(N_i, G_i).$$

--

**Fig. 3.** SSIM maps comparison of adversarial examples generated. (a): Adversarial perturbations with different methods and SSIM index value (rounded off to two decimal places), (b), (c) and (d): SSIM maps of RGB channels respectively, (e): GradCAM++ output of original image.
This is also the ratio of the intersection of the regions of \( N_i \) and \( G_i \) to their union. This metric has similar issues to recall, discussed above. To overcome this, we define Intersection over union outside (IOU outside). Consider the following:

\[
\text{IOU outside} = \frac{1}{N} \sum_i \frac{\text{FP}_i}{(\text{TP}_i + \text{FP}_i + \text{TN}_i)}
\]

\[
= \frac{1}{N} \sum_i \text{IOU}(N_i, \neg G_i).
\]

This gives us the amount of perturbation the adversarial example generation method introduces in the regions that are not important for classification (i.e. outside \( G_i \)) averaged on entire data set. Ideally, if a perturbation only targets the important regions (\( G_i \)), the value of IOU outside should be very small.

For each of the algorithms being compared, these scores are calculated for many different values of noise threshold and the resulting plots are shown in Fig. 5. For reference, Table I lists the scores at a particular noise threshold. These experiments clearly validate the qualitative observations made earlier in Section V-A. Our method (FPGAP) beats other existing norm bounded methods by a factor of \( \approx 2 \) on the precision scores. The number of pixels perturbed by our method is also lower compared with the other methods. For FPGAP, the recall scores are low; this observation is in line with our earlier observations that FPGAP introduces a smaller number of perturbations (as compared to the FGSM, which introduces perturbations all over the image).

### Additional Results

In addition to the quantitative and qualitative analysis presented earlier, in this section, we analyze the perceptual quality of the adversarial examples generated by different methods. For each of the algorithms considered, we compute the structural similarity (SSIM) and PSNR between the adversarial examples and original images, averaged over the dataset. We plot the obtained Average SSIM and Average PSNR values vs FR (as defined in Section V-A) in Fig. 6.

From Fig. 6, it is clear that the approximation method FPGAP (Algorithm 2) does a fairly good job in approximating PGAP (Algorithm 1) on MNIST and CIFAR-10 datasets. Note that on
the ImageNet dataset, we only report the results of FPGAP and ignore PGAP since it is computationally expensive to solve the QCQP iteratively.

From Fig. 6, we can see that the adversarial perturbations generated by the proposed method lead to a much higher FR (than norm bounded approaches) in the high-SSIM regime; thus, our technique is able to generate adversarial examples with minimal impact on the image quality across all the three datasets. For illustration, we consider a parameter configuration that leads to roughly the same Average SSIM for all the techniques (see Table II). It is evident that the proposed method clearly achieves high FR within the given perceptual quality range. We also corroborate this using FR comparison at the same Average PSNR (see Table III).

D. Remarks

For some data points, we observed that the QCQP was not solved accurately (the cvxpy logs showed a much higher duality gap), resulting in anomalous adversarial examples with a much lower SSIM. This, in turn, underestimates the true average SSIM of PGAP, leading to higher fooling rates at lower (estimated)
SSIM. This underestimation can be observed in Fig. 6 (first column), where we see that the FR achieved by FPGAP is higher than the FR achieved by PGAP for the same average SSIM. This shortcoming is overcome by our Faster PGAP (FPGAP) solution.

As mentioned earlier, we find it very intriguing that incorporating an SSIM constraint (which enforces perceptual similarity) also makes the perturbations semantically meaningful. One potential hypothesis is that the constraints derived from the SSIM index restrict the space of perturbations, and together with the guidance provided by the gradients calculated from the network, enable us to find the semantically important regions while inducing minimal perturbations in those regions.

VI. CONCLUSION

In this work, we investigated adversarial examples and their relationship to semantically significant regions of the image. Also, as a byproduct of trying to answer this question, we proposed a perceptually guided technique with a fast closed form approximation to generate adversarial examples that are structurally similar to the original image. In fact, none of the existing adversarial example techniques that use image quality metrics have a closed-form solution. This is in stark contrast to other norm bounded techniques (PGD, FGSM, etc.), which employ a closed-form solution to generate adversarial perturbations. Our formulation also does not assume any model on the adversarial perturbations. We analyzed the adversarial perturbations generated by the proposed technique on images from the ImageNet validation set using SSIM maps and GradCAM++. By comparing the precision and IOU scores, we observed that, unlike standard $l_p$ norm based frameworks, the perceptually guided technique is semantically-aware, i.e., it specifically targets the regions of the image that are important for classification. We find it very intriguing that a perceptually-aware formulation makes the adversarial example generation semantically-aware.

APPENDIX

A. Constraints Analysis

We analyse the constraints of (8) in more detail in this section.

1) Constraint 1: The first constraint $S_1(x, x_{adv}) \geq 1 - \epsilon_1^2$ is a linear constraint (this forces $x_{adv}$ to lie in an intersection of two half-spaces). We can see this using (6) and (7):

$$||\mu_{x_{adv}} - \eta_1 \mu_x||^2 \leq ||\mu_x||^2_2(\eta_1)^2 + c_1(\epsilon_1)^2, \eta_1,$$

or equivalently

$$N(-\sqrt{k_{11}} + k_{12}) \leq \left(\sqrt{k_{11}} + k_{12}\right) N(12),$$

where

$$k_{11} = (\eta_1)^2||\mu_x||^2_2 + c_1(\epsilon_1^2), \eta_1,$$

$$k_{12} = \eta_1 \mu_x, \text{ and } \eta_1 = 1/(1 - \epsilon_1^2).$$

2) Constraint 2: The second constraint in (8) $S_2(x, x_{adv}) \geq 1 - \epsilon_2^2$ is a quadratic constraint (this forces $x_{adv}$ to be in a high dimensional sphere). We can see this using (6):

$$||x_{adv} - 1 - \mu_{x_{adv}}||^2_2 - \eta_2(x - 1 - \mu_x)||^2 \leq ||x - 1 - \mu_x||^2_2(\eta_2)^2 + c_2(\epsilon_2^2),$$

or equivalently

$$||x_{adv} - 1 - \mu_{x_{adv}}||^2_2 - k_{22}|| \leq \sqrt{k_{21}}; \quad 13$$

where

$$k_{21} = ||x - 1 - \mu_x||^2_2(\eta_2)^2 + c_2(\eta_2^2\epsilon_2^2),$$

$$k_{22} = \eta_2(x - 1 - \mu_x) \text{ and } \eta_2 = 1/(1 - \epsilon_2^2).$$

Based on the constraints (12), (13), and the objective, the optimization problem in (8) is convex; in particular it is a Quadratically Constrained Quadratic Program (QCQP).

B. Problem Approximation

We formulate an equivalent optimization problem (14) from (9), by relaxing the first constraint and substituting it in the second. While this solution is an approximation to PGAP on account of relaxing one of the constraints, it has the advantage of having a closed form solution.
Consider the following approximation to (9).
\[
\arg\min_{x_{adv}} \quad - (x_{adv})^T \nabla_x L(w, x, y)
\text{s.t.} \quad \| (x_{adv} - 1(\mu_x)) - k_{22} \|^2 \leq k_{21}
\] (14)

We first define the Lagrangian [28] of the above problem:
\[
L = - (x_{adv})^T \nabla_x L(w, x, y)
+ \lambda \left( \| (x_{adv} - 1(\mu_x)) - k_{22} \|^2 - k_{21} \right)
\] (15)

We first take gradient of Lagrangian \( L \) and equate it to zero:
\[
\nabla_{x_{adv}} L = 0.
\]
This leads us to
\[
\nabla_x L(w, x, y) = 2\lambda ((x_{adv} - 1(\mu_x)) - k_{22}).
\] (16)

Since the objective in (14) is linear, from geometrical intuition the maximum/minimum occurs only on the boundary of the sphere (specified by the constraint):
\[
\| (x_{adv} - 1(\mu_x)) - k_{22} \|^2 = k_{21},
\]
which can be rewritten using (16) as
\[
\nabla_x L(w, x, y) = 2\lambda (x_{adv} - 1(\mu_x) - k_{22}).
\] (17)

Using the value of \( \lambda \) from (17) in (16) we get:
\[
x_{adv} = 1(\mu_x) + k_{22} + (\sqrt{k_{21}} \left( \nabla_x L(w, x, y) \right)/\| \nabla_x L(w, x, y) \|),
\]
where \( k_{21}, k_{22} \) are defined in (13) and \( 1(\mu_x) \) is all ones of size \( x_{adv} \); thus providing a closed form solution to (14).

REFERENCES

[1] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[2] S. Qiu, Q. Liu, S. Zhou, and C. Wu, “Review of artificial intelligence adversarial attack and defense technologies,” Appl. Sci., vol. 9, no. 5, 2019, Art. no. 909.

[3] E. Wong and J. Z. Kolter, “Provable defenses against adversarial examples via the convex outer adversarial polytope,” 2017, arXiv:1711.00851.

[4] A. Raghunathan, J. Steinhardt, and P. S. Liang, “Semidefinite relaxations for certifying robustness to adversarial examples,” in Proc. 42nd Int. Conf. Adv. Neural Inf. Process. Syst., 2018, pp. 10877–10887.

[5] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, “Adversarial examples are not bugs, they are features,” in Proc. 33rd Int. Conf. Neural Inf. Process. Syst. 2019, pp. 125–136.

[6] C. Szegedy et al., “Intriguing properties of neural networks,” 2013, arXiv:1312.6119.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, arXiv:1412.6572.