Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Rethinking Adversarial Examples

Master's Thesis

Yahya Jabary

yjabary@ethz.ch

Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Prof. Dr. Roger Wattenhofer
Prof. Dr. Schahram Dustdar

December 3, 2024

Acknowledgements

This thesis comes from working on a problem that truly matters to me, in an environment where curiosity and passion were shared, and I felt a sense of belonging. The topic of adversarial examples reflects my journey well, highlighting how subtle differences in perspective can lead to vastly different interpretations and outcomes.

I'm deeply grateful to those who supported me along the way. My parents, Shima and Florian, and my family, for their unwavering support, even when I took risks and turned down financial opportunities to pursue my passion. My partner, Laura, whose love and encouragement crossed the Atlantic and got me through many long nights.

I owe much to those who made this work possible. Prof. Wattenhofer, for trusting me with this project and guiding me with wisdom and humor. Andreas Plesner, who was just as much of a mentor as a collaborator, for his dedication to our vision. Turlan Kuzhagaliyev and Alireza Furutanpey, for their camaraderie.

Thanks also to those whose paths have diverged from mine but whose impact remains with me: Prof. Schahram Dustdar, who enabled my studies abroad, and Prof. Ali Mashtizadeh, who introduced me to operating systems research.

I hope to continue this journey with the same spirit that brought me here.

Abstract

...

Keywords: Robustness, Alignment, Interpretability, Algorithmic Models

Contents

Chapter 1

Introduction

We have two goals in writing this document. One: fulfilling the requirements for a master's degree by presenting and extending our original research [1] in thesis form. Two: offering a fresh and cohesive perspective on the rapidly evolving and, in our view, really exciting field of adversarial machine learning to a broader audience, with fewer technical prerequisites. We hope it will be valuable to those interested.

1.1 Definition

Adversarial examples are closely related to the concept of perturbation methods.

The origin of perturbations can be traced back to the early days of computational geometry by Seidel et al. in 1998 [2]. Perturbation techniques in computational geometry address a fundamental challenge: handling "degeneracies" in geometric algorithms. These are special cases that occur when geometric primitives align in ways that break the general position assumptions the algorithms rely on.

> **Example:** Perturbation scheme for a linear classifier.
>
> Consider a simple case of determining whether a point lies above or below a line [3]. While this classification appears straightforward, numerical issues arise when the point lies exactly on the line. Such degeneracies can cascade into algorithm failures or inconsistent results. The elegant solution is to imagine slightly moving (perturbing) the geometric objects to eliminate these special cases. Formally, we can express symbolic perturbation as $p_\varepsilon(x) = x + \varepsilon \cdot \delta(x)$ where x is the original input, ε is an infinitesimally small positive number the exact value of which is unimportant, and $\delta(x)$ is the perturbation function to break degeneracies.
>
> A perturbation scheme should be (1) consistent, meaning that the same

input always produces the same perturbed output (2) infinitesimal, such that perturbations are small enough not to affect non-degenerate cases and (3) effective, in breaking all possible degeneracies.

One powerful perturbation approach is Simulation of Simplicity (SoS) [4, 5, 6, 7, 8, 9]. SoS systematically perturbs input coordinates using powers of a symbolic infinitesimal. For a point $p_i = (x_i, y_i)$, the perturbed coordinates become:

$$(\tilde{x_i}, \tilde{y_i}) = (x_i + \varepsilon^{2i}, y_i + \varepsilon^{2i+1}) = p_i + \varepsilon^{2i} \cdot (1, \varepsilon)$$

This scheme ensures that no two perturbed points share any coordinate, effectively eliminating collinearity and other degeneracies.

The beauty of perturbation methods lies in their ability to handle degeneracies without explicitly detecting them, making geometric algorithms both simpler and more robust.

Adversarial examples on the other hand, first introduced by Szegedy et al. in 2014 [10], follow the same principles as perturbation methods, but with the opposite objective. Instead of seeking to eliminate degeneracies (brittleness in the decision boundary), they exploit them to cause targeted misclassifications. Intuitively they can be understood as seeking the closest point in the input space that lies on the "wrong side" of a decision boundary relative to the original input. This shift, applied to the original input, creates an adversarial example.

Example: Fast Gradient Sign Method (FGSM)

FGSM is one of the earliest and most widely recognized adversarial attack techniques, introduced by Goodfellow et al. [11] in the context of visual recognition tasks. Given an input image x, FGSM generates an adversarial example x' by perturbing the input in the direction of the gradient of the loss function with respect to the input.

The perturbation is controlled by a parameter $\varepsilon > 0$ [a], which determines the magnitude of the change based on the direction of change for each pixel or feature in the input x. The model's loss function denoted by J, θ represents the model's parameters, and y is the true target label.

It works by calculating the gradient of the loss function with respect to the input, $\nabla_x J(\theta, x, y)$, and then adjusting the input in the direction of this gradient. The sign of the gradient, $\text{sign}(\nabla_x J(\theta, x, y))$, is used to ensure that the perturbation is small, while the ℓ_∞-norm

constraint ensures that the change to the input remains "imperceptible" to human observers [11, 12]. More on the concept of imperceptibility later.

The process for generating an adversarial example with FGSM can be expressed as:

$$x' = x + \underbrace{\varepsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))}_{\text{Perturbation}}$$

In the untargeted version, the perturbation is designed to increase the loss for the correct class. In the targeted version the perturbation is designed to minimize the loss with respect to the adversary's chosen target class, making the model predict it deliberately.

[a] Commonly $\varepsilon = 8/255$ for 8-bit images, so it stays within the precision constraints of the pixel values.

While initially discovered in computer vision applications, the attack can be crafted for any domain or data type, even graphs [13]. Natural language processing models can be attacked by circumventing the discrete nature of text data [14, 15, 16]. One particularly entertaining example is the subversion of the conference paper-reviewer assignment model by Eisenhofer et al. [17], where authors preselect reviewers to gain a competitive advantage. Speech recognition systems are vulnerable to audio-based attacks, where crafted noise can cause system failure [18]. Deep reinforcement learning applications, including pathfinding and robot control, have also shown susceptibility to adversarial manipulations that can compromise their decision-making capabilities [19].

1.2 Types of Adversarial Examples

Having established the general concept of adversarial examples, we can now explore the various ways they can be categorized. Our system is not exhaustive: The field continues to evolve, with new attack vectors emerging regularly [20].

We can differentiate between white-box and black-box attacks. White-box attacks assume complete knowledge of and access to the target model, while black-box attacks operate with limited or no access to the model's internal workings [21]. Interestingly, research has shown that in some cases, black-box attacks can be more effective than white-box approaches at compromising model security [21].

An attack can be targeted or untargeted. Targeted attacks aim to manipulate the model into producing a specific, predetermined output, whereas untargeted attacks simply seek to cause any misclassification or erroneous output [21, 13].

This distinction is particularly relevant in security-critical applications, where the attacker's goals may vary from causing general disruption to achieving specific malicious outcomes.

The method used to generate adversarial examples can be gradient-based, optimization-based or search-based strategies. For example, some text-based attacks leverage language models to generate alternatives for masked tokens, ensuring grammatical correctness and semantic coherence [22].

The extent to which adversarial examples are transferable – meaning their ability to fool multiple different models – is another way to differentiate them. Some adversarial examples demonstrate high transferability across various model architectures, while others are more model-specific in their effectiveness [23, 24]. Recent research has shown that adversarial examples are more readily transferable between vanilla neural networks than between defended ones [25, 26].

Finally, the attacks can be either focused on preserving the semantic meaning of inputs or exploit the mathematical properties of models without regard for semantic interpretation provides [27].

1.3 Semantics Preservation

Initially, Szegedy and colleagues proposed using epsilon-bounded perturbations in pixel space to create changes "imperceptible [to humans]", that could fool neural networks [10]. However, the term "imperceptibility" is indeed more accurately described as "semantics preserving" as demonstrated by recent work [28]. The assumption that limiting pixel-space modifications ensures semantic preservation is fundamentally flawed, because substantial pixel-space changes can occur without affecting semantic meaning, such as changes in lighting, exposure, sharpness, angle, shadow, field of view, ..., or the the orientation of an animal's fur 1.1.

Recent research has made significant progress in developing frameworks for creating truly semantics-preserving adversarial examples. These approaches focus on learning low-dimensional manifolds that capture semantic information rather than relying on pixel-space constraints [29]. This shift in methodology acknowledges that semantics are better represented in the embedding space of inputs rather than in high-level feature spaces by Kariyappa et. al. [28].

Additionally, the transferability of adversarial examples to human perception under time constraints raises profound questions about the nature of neural network vulnerabilities. This observation aligns with the perspective that neural network depth might correlate with human processing time, suggesting a deeper connection between artificial and human visual processing systems.

The challenge of defining semantics remains central to this discussion. Without

perfect representations that align with human judgment functions, we must rely on the best available encoders as proxies for semantic preservation [30]. This pragmatic approach acknowledges the limitations of current technology while striving for more meaningful adversarial examples.

Recent work has shown that many generated adversarial examples fail to preserve semantics, with up to 70% of examples potentially requiring rejection [30]. This finding emphasizes the importance of developing better methods for ensuring semantic preservation in adversarial attacks, leading to new approaches like Semantics-Preserving-Encoder (SPE) and other manifold-based techniques [29].

Moving forward, the field should continue to prioritize semantic preservation over simple pixel-space constraints, acknowledging that true imperceptibility requires a deeper understanding of semantic representation in both human and machine perception systems.

write a paper in a casual but academic tone. write full sentences. don't use enumerations, itemizations, headings. use academic papers as citations.

"‘ In the paper where adversarial examples are introduced by Szegedy, he says that they're using an epsilon value to limit the changes in the pixel-space, in order to ensure "imperceptibility [to humans]", however this doesn't make sense.

first of all the term "imperceptibility" here is interchangable with the much more well-defined term "semantics preserving".

second of all you can't preserve semantics through changes in the pixel space, because: (1) you can have very few pixels changed and still have a change in semantics, in time constrained humans and the authors didn't conduct any experiments with humans and this constraint doesn't ensure anything and (2) you can have much more pixsels changed and have no change in the semantic space (i.e. changing exposure, sharpness, brightness, angle, shadow, field of view, ... the orientation of the fur of a pet).

recent experiments on the transferability of adversarial examples to the human vision system might sugegst that the core assumptions under which adversarial examples were introduced are flawed, because these attacks do transfer to time constrained humans. this raises the philosophical question of whether it's then justified for a model to be fooled by an attack by which time constrained humans are fooled and whether or what kinds of flaws of the human judgement function we are learning should be learned by the model. this is particularly interesting because ilya sutskever briefly mentions in an interview that he believes an increased depth in a neural network is equivalent to a human with more "time" available to perceive something. so this would in some sense further underline.

in order order to be able to decide whether an attack is semantically preserving or not, we need to have a clear definition of what semantics are. in the paper "what makes an image realistic" by theis this is discussed further, but in short, as long as we don't have representations that perfectly align with the human

judgement function, we can't have a clear definition of what semantics are and to which extent our attacks are working as intended.

therefore for the remainder of this paper we will use the term "semantics preserving" instead of "imperceptibility" and we refer to the best possible encoder we have at our disposal because we can't have a perfect representation of the human judgement function and did not conduct any experiments with humans. "'

idea: adversarial examples are a problem of misaligned internal representations and not necessarily a problem of attack vectors.

"' having introduced the concepts of an "Image Space" a subset of which are reasonably realistic images, a "latent space" of how an encoder's latent representations are stored, the reader might reasonably ask why adversarial examples exist to begin with and why the "latent space" isn't an accurate depiction of the "semantic space" such that our encoder and internal latent representation is entirely undisturbed by any non-semantics-changing perturbations.

we believe that this is further evident by the fact that adversarial exampels transfer across models, datasets, architectures, training procedures. the reason adversarial exampels transfer across models, is because the human visual system they're trying to approximate is, we assume, the same. this is there the theory of platonic representations comes in, that are shared both across modailities (text, image) and models, independent of their architecture, when you take certain slices.

we believe that in it's essence any adversarial vulnerability is therefore not necessarily a vulnerability to attack vectors but just an insufficiently aligned model and a problem of misaligned internal representations. this drift between the human judgement function that the models are trying to learn and the model's internal representation is what causes adversarial examples to exist. this is called the "human-machine vision gap" and this is why we believe that the term "alignment" is a much more accurate term to describe the problem of adversarial examples, and that the term "robustness" is a much more accurate term to describe the solution to this problem.

we make the bold claim that therefore hypothetically, a correctly robustified model that can withstand adversarial attacks is infact a simply better generalizing model to downstream tasks, although it might perform worse on the test set (this is described as the bias-variance tradeoff). our second claim is that therefore the robustness-accuracy tradeoff is a false dichotomy and that it's essentially the same as the bias-variance tradeoff.

therefore what we're studying here isn't necessarily adversarial examples but the human-machine vision gap and the alignment of the model's internal representation with the human judgement function. this is why we believe that the term "adversarial examples" is a misnomer and that the term "misaligned examples" is

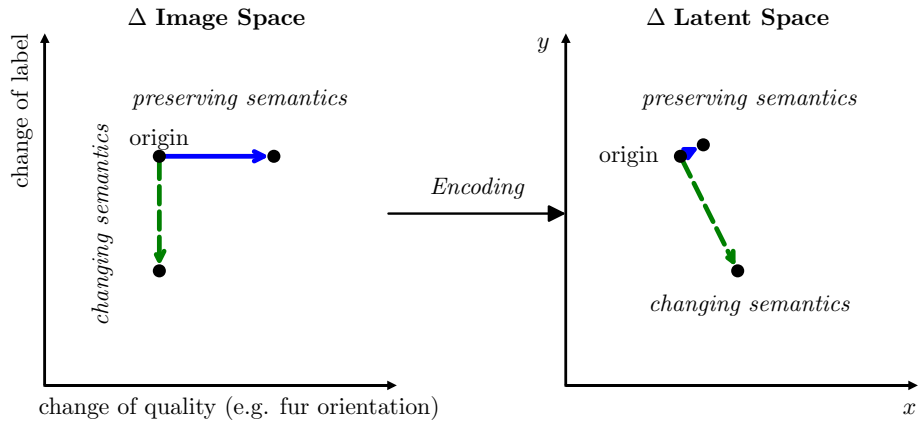

Figure 1.1: Semantics Preserving vs. Semantics Changing Perturbations (assuming the latent representations are perfectly aligned with the human judgement function).

a much more accurate term to describe the problem of insufficiently generalizing models.

but it doesn't stop here. "'

idea: transferability to humans

idea: transferability across models

figure 1.1

realism: [31]

1.4 Counterintuitive Properties

Despite how important and useful they are, there's still a bunch we don't really know about them.

There are a bunch of stuff we don't know about them

1.5 Mental Models

In 2013 Szegedy et al. [32] discovered that deep neural networks are vulnerable to adversarial examples – inputs with imperceptible perturbations that cause misclassification, revealing blind spots where the models' decision boundaries are brittle, despite appearing to generalize well on normal inputs. Since then

we have made a lot of progress in both finding attack vectors [11] [33] [34] and adversarially training models to be more robust [35] [33] [36] to these attacks.

However, despite the progress, we still don't fully understand:

- What they are, why they exist and how they work.
- How to defend against them without sacrificing accuracy (robustness vs accuracy trade-off).
- Why they transfer between different models, datasets, architectures, training procedures and even to the human visual system [37].

There have been many attempts at explaining adversarial examples, each with limitations and assumptions – some complementary, some contradictory [1].

For example, the *dimpled manifold hypothesis* [39] suggests that the decision boundary of deep neural networks is close to the data manifold, making it easy to find adversarial examples, while the *non-robust features hypothesis* [38] suggests that models exploit non-robust features that are imperceptible to humans, leading to a vulnerability against small perturbations

The *dimpled manifold hypothesis* is a particularly controversial one, as it was criticized by Yannik Kilcher [40] in 2021, who also provided a counterexample in less than 100 lines of code [41]. Despite the lack of generalizability, a master's student from the University of Vienna, Lukas Karner, successfully verified and replicated all experiments detailed in the dimples paper [42] in 2023. Lukas allowed me to use his results in my work. He mentioned that there is currently no paper or thesis based on his work and that he would be happy to see his results being used in a meaningful way.

This leaves us with the possibility that the experiments carried out are correct in themselves, but that the chain of reasoning is inconclusive and therefore doesn't generalize. Investigating this further would require more rigor by formalizing falsifiable hypotheses based on the paper and conducting experiments to test them.

Another aspect of the *dimpled manifold hypothesis* worth exploring is the idea of projecting the perturbations on the data manifold before applying them to the input space. Reducing the dimensionality of the perturbations or compressing them makes them visible to the human eye and interpretable as demonstrated by Karner [42].

1.6 Counterintuitive Properties

[1] For a comprehensive overview of the hypotheses, see the Addendum of Ilyas et al. [38]

Chapter 2

Methodology

Chapter 3

Results

Chapter 4

Stuff

Figure 4.1: This is an example graphic.

> "The stuff is what the stuff is, brother. Okay. We don't ask questions about the weights. We just wake up, we go to work, we use the weights, we go back home. Okay. If we change the weights, the predictions would be different and less good, probably... depending on the weather... so we don't ask about the weights."
>
> — *James Mickens, USENIX Security 18 [43]*

4.1 First Section Title

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

4.1.1 First Subsection Title

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Theorem 4.1 (First Theorem)**.** *This is our first theorem.*

Proof. And this is the proof of the first theorem with a complicated formula and a reference to Theorem 4.1. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

$$\frac{\mathrm{d}}{\mathrm{d}x}\arctan(\sin(x^2)) = -2 \cdot \frac{\cos(x^2)x}{-2 + (\cos(x^2))^2} \tag{4.1}$$

□

And here we cite some external documents [44, 45]. An example of an included graphic can be found in Figure 4.1. Note that in LaTeX, "quotes" do not use the usual double quote characters.

Bibliography

[1] Y. Jabary, A. Plesner, T. Kuzhagaliyev, and R. Wattenhofer, "Seeing through the mask: Rethinking adversarial examples for captchas," *arXiv preprint arXiv:2409.05558*, 2024.

[2] R. Seidel, "The nature and meaning of perturbations in geometric computing," *Discrete & Computational Geometry*, vol. 19, pp. 1–17, 1998.

[3] M. De Berg, *Computational geometry: algorithms and applications*. Springer Science & Business Media, 2000.

[4] W. R. Franklin and S. V. G. de Magalhães, "Implementing simulation of simplicity for geometric degeneracies," *arXiv preprint arXiv:2212.08226*, 2022.

[5] Edelsbrunner, Letscher, and Zomorodian, "Topological persistence and simplification," *Discrete & computational geometry*, vol. 28, pp. 511–533, 2002.

[6] H. Edelsbrunner and D. Guoy, "Sink-insertion for mesh improvement," in *Proceedings of the seventeenth annual symposium on Computational geometry*, 2001, pp. 115–123.

[7] H. Edelsbrunner and E. P. Mücke, "Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms," *ACM Transactions on Graphics (tog)*, vol. 9, no. 1, pp. 66–104, 1990.

[8] B. Lévy, "Robustness and efficiency of geometric programs: The predicate construction kit (pck)," *Computer-Aided Design*, vol. 72, pp. 3–12, 2016.

[9] P. Schorn, "An axiomatic approach to robust geometric programs," *Journal of symbolic computation*, vol. 16, no. 2, pp. 155–165, 1993.

[10] C. Szegedy, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[12] J. Zhang and C. Li, "Adversarial examples: Opportunities and challenges," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2578–2593, 2019.

[13] S. Kashyap, A. Sharma, S. Gautam, R. Sharma, S. Chauhan, and Simran, "Adversarial attacks and defenses in deep learning," *2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP)*, pp. 318–323, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272716335

[14] X. Han, Y. Zhang, W. Wang, and B. Wang, "Text adversarial attacks and defenses: Issues, taxonomy, and perspectives," *Security and Communication Networks*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248369346

[15] Z. Meng and R. Wattenhofer, "A geometry-inspired attack for generating natural language adversarial examples," *arXiv preprint arXiv:2010.01345*, 2020.

[16] Z. Yang, Z. Meng, X. Zheng, and R. Wattenhofer, "Assessing adversarial robustness of large language models: An empirical study," *arXiv preprint arXiv:2405.02764*, 2024.

[17] T. Eisenhofer, E. Quiring, J. Möller, D. Riepel, T. Holz, and K. Rieck, "No more reviewer# 2: Subverting automatic {Paper-Reviewer} assignment using adversarial learning," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 5109–5126.

[18] K. Rajaratnam and J. Kalita, "Noise flooding for detecting audio adversarial examples against automatic speech recognition," in *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2018, pp. 197–201.

[19] X. Bai, W. Niu, J. Liu, X. Gao, Y. Xiang, and J. Liu, "Adversarial examples construction towards white-box q table variation in dqn pathfinding training," *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 781–787, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49895854

[20] Y. L. Khaleel, M. A. Habeeb, and H. Alnabulsi, "Adversarial attacks in machine learning: Key insights and defense approaches," *Applied Data Science and Analysis*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272000855

[21] G. Capozzi, D. C. D'Elia, G. A. Di Luna, and L. Querzoni, "Adversarial attacks against binary similarity systems," *IEEE Access*, 2024.

[22] S. Garg and G. Ramakrishnan, "Bae: Bert-based adversarial examples for text classification," *arXiv preprint arXiv:2004.01970*, 2020.

[23] Y. Li, Y. Guo, Y. Xie, and Q. Wang, "A survey of defense methods against adversarial examples," *2022 8th International Conference on Big Data and Information Analytics (BigDIA)*, pp. 453–460, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252165991

[24] Y. Li, M. Cheng, C.-J. Hsieh, and T. C. Lee, "A review of adversarial attack and defense for classification methods," *The American Statistician*, vol. 76, no. 4, pp. 329–345, 2022.

[25] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, "Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3866–3876.

[26] S. Zheng, C. Zhang, and X. Hao, "Black-box targeted adversarial attack on segment anything (sam)," *arXiv preprint arXiv:2310.10010*, 2023.

[27] K. Browne and B. Swift, "Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks," *arXiv preprint arXiv:2012.10076*, 2020.

[28] S. Kariyappa and O. Dia, "Semantics preserving adversarial examples."

[29] W. Lee, H. Lee, and S.-g. Lee, "Semantics-preserving adversarial training," *arXiv preprint arXiv:2009.10978*, 2020.

[30] D. Herel, H. Cisneros, and T. Mikolov, "Preserving semantics in textual adversarial attacks," in *ECAI 2023*. IOS Press, 2023, pp. 1036–1043.

[31] L. Theis, "What makes an image realistic?" *arXiv preprint arXiv:2403.04493*, 2024.

[32] E. D. Cubuk, B. Zoph, S. S. Schoenholz, and Q. V. Le, "Intriguing properties of adversarial examples," *arXiv preprint arXiv:1711.02846*, 2017.

[33] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[34] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[35] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in neural information processing systems*, vol. 32, 2019.

[36] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[37] G. Elsayed, S. Shankar, B. Cheung, N. Papernot, A. Kurakin, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial examples that fool both computer vision and time-limited humans," *Advances in neural information processing systems*, vol. 31, 2018.

[38] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019.

[39] A. Shamir, O. Melamed, and O. BenShmuel, "The dimpled manifold model of adversarial examples in machine learning," *arXiv preprint arXiv:2106.10151*, 2021.

[40] Y. Kilcher, "The Dimpled Manifold Model of Adversarial Examples in Machine Learning (Research Paper Explained)," https://www.youtube.com/watch?v=k_hUdZJNzkU/, 2021, [Online; accessed 17-July-2024].

[41] Y. Kilcher, "dimple test," https://gist.github.com/yk/de8d987c4eb6a39b6d9c08f0744b1f64/, 2021, [Online; accessed 17-July-2024].

[42] L. Karner, "The Dimpled Manifold Revisited," https://github.com/LukasKarner/dimpled-manifolds/, 2023, [Online; accessed 17-July-2024].

[43] J. Mickens, "Q: Why do keynote speakers keep suggesting that improving security is possible? a: Because keynote speakers make bad life decisions and are poor role models," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/mickens

[44] A. One and A. Two, "A theoretical work on computer science," in *30th Symposium on Comparative Irrelevance, Somewhere, Some Country*, Jun. 1999.

[45] A. One and A. Two, "A theoretical work on computer science," in *30th Symposium on Comparative Irrelevance, Somewhere, Some Country*, Jun. 1999.

Appendix A

First Appendix Chapter Title
