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 6, 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. Roger 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

Originality

I hereby declare that I have written this thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

Papers

Seeing Through the Mask: Rethinking Adversarial Examples for CAPTCHAs
Yahya Jabary, Andreas Plesner, Turlan Kuzhagaliyev, Roger Wattenhofer
ArXiv: 2409.05558

Open source software

The majority of time working on this thesis, was spent on developing a reproducible research pipeline for experiments in a compute and GPU memory constrained, containerized environment with compiled dependencies.

Due to the exploratory nature of the work, many of the software built and experiments conducted were not directly related to the final paper.

The following projects were developed as part of this work (in chronological order, with the most recent first):

self-ensembling
All experiments related to the Self-Ensembling algorithm by Fort et al.
`https://github.com/ETH-DISCO/self-ensembling`
`https://huggingface.co/sueszli/self-ensembling-resnet152`

ensemble-everything-everywhere
Pull Request: Optimizing the official Self-Ensembler repository by Fort et al.
`https://github.com/stanislavfort/ensemble-everything-everywhere/pull/2`

vision
Pull Request: Containerizing TorchVision to recompile ResNet-50 from scratch.

https://github.com/pytorch/vision/pull/8652

advx-bench
All experiments related to the geometric masks from the paper.
https://github.com/ETH-DISCO/advx-bench
https://huggingface.co/sueszli/robustified_clip_vit

cluster-tutorial
Tutorial on how to circumvent the distributed NFS4 filesystem by attaching the terminal to an interactive SLURM job, run an Apptainer to enable admin privileges and redirect all filepointers to the EXT4 filesystem to avoid out-of-memory limits. A Jupyter notebook is then hosted on a public IP address.
https://github.com/ETH-DISCO/cluster-tutorial

python-template
Short scripts to `pip-compile` dependencies, containerize the environment and translate back and forth between Conda and Docker for different job submission systems.
https://github.com/sueszli/python-template/

captcha-the-flag
Cybersecurity emulation for CAPTCHAs: A deployable replica of Google's reCAPTCHAv2 and a scraper used to evaluate challenges against solvers.
https://github.com/ETH-DISCO/captcha-the-flag

Breakdown of contributions

For the paper, Andreas Plesner had the original idea. The written text was joint work between all authors, with Prof. Roger Wattenhofer taking the lead on creating a cohesive narrative for our experiments, Andreas and me writing the majority of the text, and Turlan providing experimental results and feedback. The TU Wien DSG lab kindly provided the computational resources for robustifying a ResNet-50 model, which unfortunately did not make it into the final version of the paper. Additionally, Alireza Furutanpey suggested using LPIPS as a metric to evaluate the perceptual quality of adversarial examples, which we incorporated into our weighted objective function.

Regarding the developed software, all contributions are my own, unless stated otherwise in the repository. A prototype of the self-ensembled ResNet-50 model was developed by Andreas Plesner, but the authors soon released their own implementation, which was then used in all experiments for consistency.

Andreas Plesner diligently proofread this manuscript for errors. As is traditional, any errors that remain are of course mine alone.

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[1].

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.

[1] Thanks to Prof. Roger Wattenhofer for sharing this piece of unorthodox history.

> 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:
>
> $$x' = x + \underbrace{\varepsilon \cdot \mathrm{sign}(\nabla_x J(\theta, x, y))}_{\text{Perturbation}}$$
>
> 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]. Speech recognition systems are vulnerable to audio-based attacks, where crafted noise can cause system failure [17]. Deep reinforcement learning applications, including pathfinding and robot control, have also shown susceptibility to adversarial manipulations that can compromise their decision-making capabilities [18].

1.2 Threat Modeling

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 [19]. This is particularly important in threat modeling, where the goal is to anticipate and defend against potential attacks.

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 [20]. Interestingly, research has shown that in some cases, black-box attacks can be more effective than white-box approaches at compromising model security [20].

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 [20, 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 [21].

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 [22, 23]. Recent research has shown that adversarial examples are more readily transferable between vanilla neural networks than between defended ones [24, 25].

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 [26].

1.3 Imperceptible $\neq$ Semantics Preserving

Adversarial examples are traditionally expected to have two key properties: (1) they should successfully cause misclassification in machine learning models while (2) remaining imperceptible to human observers [27].

However, the concept of "imperceptibility [to humans]" as originally proposed by Szegedy et al. [10] by limiting pixel-space perturbations through an ε-bounded constraint is fundamentally flawed. This is because the human visual system is not solely reliant on pixel-space information to interpret images [28]:

1. *Changing many pixels doesn't necessarily change semantics*: For instance, changes in lighting, exposure, sharpness, field of view or even the the orientation of an animal's fur [29] does not necessarily alter the semantic content of an image, despite involving many pixel-space changes. The figure 1.1 serves to illustrate this point, assuming that the latent representations are perfectly aligned with the human judgement function.

2. *Changing few pixels doesn't necessarily preserve semantics*: The assumption that minimal pixel-space changes ensure semantic preservation is also flawed. This is because small ε-bounded adversarial perturbations are found to cause misclassification in time-constrained humans [30]. Even more interestingly, humans detect forged low-ε adversarial examples with high accuracy in both the visual (85.4%) [31] and textual ($\geq$70%) [32] do-

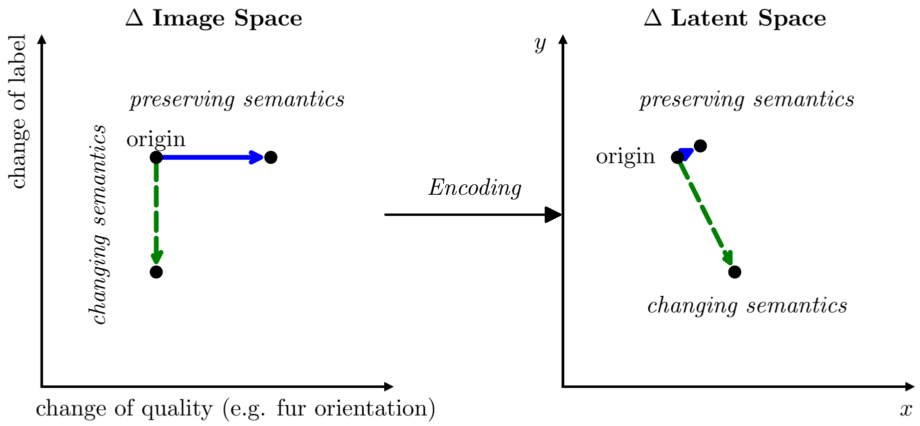

Figure 1.1: Semantics preserving/changing perturbations in pixel/latent-space.

main. Invertible neural networks can partially mitigate this issue in the visual domain [33].

> **Open Question:** If imperceptibility is not a reliable indicator of semantic preservation, then what are the true properties of adversarial examples?

Recent research has shifted towards more nuanced concepts like “semantic preservation” [34] acknowledging that the relationship between perturbation size and semantic change is not straightforward. This shift has led to the development of new methods for generating adversarial examples that focus on preserving semantic information rather than relying on pixel-space constraints [35].

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 [32]. This pragmatic approach acknowledges the limitations of current technology while striving for more meaningful adversarial examples.

1.4 Inner Representations

The internal latent representations of neural networks, their alignment with human understanding and the resuling gap between the two (the human-machine vision gap [36]) is a central theme in adversarial machine learning research. This gap has many practical implications for the robustness and interpretability of machine learning models.

Neural networks trained with topological features develop substantially different internal representations compared to those trained on raw data, though these differences can sometimes be reconciled through simple affine transformations [37]. This finding suggests that while the structural representations may differ, the underlying semantic understanding might be preserved across different training approaches.

The Centered Kernel Alignment (CKA) metric enables us to compare neural network representations, though it comes with important caveats. In biological and artificial neural networks, CKA can show artificially high similarity scores in low-data, high-dimensionality scenarios, even with random matrices [38]. This limitation is particularly relevant when comparing representations of different sizes or when analyzing specific regions of interest.

The relationship between network architecture and concept representation has also been explored. Generally higher-level concepts are typically better represented in the final layers of neural networks, while lower-level concepts are often better captured in middle layers [39, 40]. This hierarchical organization mirrors our understanding of human cognitive processing and suggests that neural networks naturally develop structured representations that align with human conceptual understanding.

The choice of objective function significantly influences how networks represent information, particularly when dealing with biased data. Networks trained with Negative Log Likelihood and Softmax Cross-Entropy loss functions demonstrate comparable capabilities in developing robust representations [41].

Recent research [42] has demonstrated that neural networks with strong performance tend to learn similar internal representations, regardless of their training methodology. Networks trained through different approaches, such as supervised or self-supervised learning, can be effectively “stitched” together without significant performance degradation. This suggests a convergence in how successful neural networks represent information.

This aligns with the “Platonic Representation Hypothesis” [43], which suggests that neural networks are converging toward a shared statistical model of reality, regardless of their training objectives or architectures. As models become larger and are trained on more diverse tasks and data, their internal representations increasingly align with each other, even across different modalities like vision and language. This convergence appears to be driven by the fundamental constraints[2] of modeling the underlying structure of the real world, similar to Plato’s concept of an ideal reality that exists beyond our sensory perceptions. The hypothesis proposes that this convergence is not coincidental but rather a natural consequence of different models attempting to capture the same under-

[2] Formally: “If an optimal representation exists in function space, larger hypothesis spaces are more likely to cover it.”

lying statistical patterns and relationships that exist in reality [43].

Should the "Platonic Representation Hypothesis" hold true, this would either mean that (a) adversarial examples as we know them are misalignments from a converged model of reality, or (b) that there exist a universal adversarial example that can fool any model, regardless of its architecture, training data or objective function, converging to a single and shared model of reality.

Recent work by Moosavi-Dezfooli et al. [44] have demonstrated the existence of a single perturbation that can fool most models for all naturally occurring images, adding weight to the latter interpretation, though the question remains open.

1.5 Open Questions & Mental Models

The question discussed in the previous section is just one of many that remain open and yet have to be fully explained [45]. Among them are:

- What are adversarial examples?
- Why are the adversarial examples so close to the original images?
- Why don't the adversarial perturbations resemble the target class?
- Why do robustness and accuracy trade-off [46]?
- Why do adversarial examples transfer between models, even on disjoint training sets [10]?
- Why do adversarial examples transfer between models [10]?
- Why do adversarial examples transfer between models and time-limited humans [30]?

Initially, when Szegedy et al. [10] coined the term they proposed that adversarial examples are caused by (1) neural networks developing internal representations that become increasingly disconnected from the input features as they progress through deeper layers, and (2) that these networks fail to maintain the smoothness properties typically assumed in traditional machine learning approaches. The idea was that this lack of smoothness gives them their expressive power, but also makes them vulnerable to these attacks.

> **Definition:** Manifold
>
> The first attempt to explain adversarial examples used the term "manifold". It refers to a data submanifold.
>
> *"In some sense, what we describe is a way to traverse the manifold represented by the network [...]. The adversarial examples represent*

> *low-probability (high-dimensional) ‘pockets’ in the manifold [...].”*
> — Szegedy et al., 2013 [10]

A manifold can be thought of as a low-dimensional structure embedded in a high-dimensional space, representing the set of valid data points (e.g., natural images) that the neural network is trained to classify. Mathematically, if the input data lies on a manifold $\mathcal{M} \subset \mathbb{R}^m$, then $\mathcal{M}$ represents the subset of the high-dimensional input space $\mathbb{R}^m$ that corresponds to meaningful or real-world data.

Szegedy et al. suggest that adversarial examples exploit the structure of this manifold and its surrounding space. Specifically, adversarial examples are small perturbations r added to an input $x \in \mathcal{M}$, such that the perturbed input $x' = x + r$ lies off the data manifold but still within the high-dimensional input space.

Formally, given a classifier $f : \mathbb{R}^m \to \{1, ..., k\}$ and its associated loss function $\text{Loss}_f(x, y)$, an adversarial example x' for an input x with true label y can be found by solving:

$$\min_r \|r\|_2 \quad \text{subject to } f(x+r) \neq y,\ x + r \in [0,1]^m$$

where r is constrained to be small (e.g., in terms of its L_2-norm). This optimization problem effectively traverses the space near x, moving off the manifold $\mathcal{M}$, to find regions where the classifier’s decision boundary behaves unexpectedly.

The paper suggests that these adversarial examples expose “blind spots” in the learned representation of the manifold by the neural network. The network’s decision boundary may extend into regions near $\mathcal{M}$ in ways that are not semantically meaningful, allowing adversarial perturbations to exploit these regions. This phenomenon arises due to the high dimensionality of the input space and the discontinuous mappings learned by deep networks, which can fail to generalize smoothly beyond the manifold [47, 48, 49, 50, 45].

A more rigorous definition of the manifold hypothesis is provided by Khoury et al. [47].

Definition: Realism

A “realistic subspace” can be understood as a subset of the data manifold

> where the images appear plausible according to human perception or a given distribution P. A simple formula that expresses this idea elegantly to quantify realism is derived from the notion of randomness deficiency in algorithmic information theory [51]:
>
> $$U(x) = -\log P(x) - K(x)$$
>
> where $P(x)$ is the probability density of the image x under the target distribution, and $K(x)$ is the Kolmogorov complexity of x, representing the shortest description of x in a universal programming language. This measure, called a "universal critic", captures how well x aligns with both the statistical properties of P and its compressibility. A low value of $U(x)$ indicates that x is realistic, while a high value suggests it is unrealistic [51].
>
> This approach generalizes prior methods by integrating both probabilistic and structural aspects of realism. It highlights that realism depends not only on adherence to statistical patterns (e.g., probabilities or divergences) but also on whether an image can be plausibly generated within the constraints of P. While directly computing $K(x)$ is infeasible due to its uncomputability, practical approximations (e.g., compression algorithms or neural network-based critics) can serve as proxies [51].
>
> The distinction between realistic and unrealistic perturbations is crucial for practical applications, as some adversarial examples may be mathematically valid but physically impossible to realize in real-world scenarios [52].
>
> The challenge of quantifying realism remains a fundamental problem in machine learning [51].

Since then there have been many attempts at finding a cohesive narrative to explain these counter-intuitive properties, each with their own limitations and assumptions – some complementary, some contradictory [53].

Non-robust features & concentration of measure in high-dimensions. Most popularly, Ilyas et al. [53] proposed that features that models learn from can be divided in 3 categories: (1) useless featurs, to be discarded by the feature extractor, (2) robust features, which are comprehensible to humans, generalize across multiple datasets and remain stable under small adversarial perturbations, and (3) non-robust features, which are incomprehensible to humans, learned by the supervised model to exploit patterns in the data distribution which are highly effective for the task at hand but also brittle and easily manipulated by adversarial perturbations. The authors suggest that the vulnerability of deep neural networks to adversarial examples is due to their reliance on non-robust features, and inherent to how the models are optimized to minimize the loss function. In essence, the authors argue that adversarial vulnerability is a property of the dataset, not the algorithm and by removing these non-robust features from the training data although the adversarial robustness of the model can be improved, due to information loss of the most predictive features, the model's overall accuracy will decrease. This view is also shared among [54, 55, 56, 57, 58, 59, 60, 61, 62, 63].

Theoretical constructions which incidentally exploit non-robust features. A complimenting hypothesis is that because models trained to maximize accuracy will naturally utilize non-robust data, regardless of whether it aligns with human perception [53] they add a low-magnitude weight to sensitive variables that can get overamplified by adversarial examples [64, 65]. The assumption is that this happens due to computational constraints or model complexity.

Insufficient data. Schmidt et al. argue [66] that adversarial vulnerabilities are intrinsic to statistical learning in high-dimensional spaces and not merely due to flaws in specific algorithms or architectures. This is a natural consequence of the mental model proposed by Ilyas et al. [53]. They also argue that due to information loss in a robust dataset, significantly more data is required during training in order to achieve comparable performance.

Boundary Tilting. A competing view by Tanay and Kim et al. [67, 68] suggests that adversarial examples exist because decision boundaries extend beyond the actual data manifold and can lie uncomfortably close to it, essentially viewing adversarial examples as a consequence of overfitting. This observation can be quantified through the concept of adversarial strength, which relates to the angular deviation between the classifier and the nearest centroid classifier. The authors also argue that this vulnerability can be addressed through proper regularization techniques.

Test Error in Noise. There might be a link between robustness to random noise and adversarial attacks [69, 70, 71, 58, 70]. This might imply that adversarial examples exploit inherent weaknesses in how models generalize under noisy or perturbed conditions.

Local Linearity. Goodfellow, Shlens, and Szegedy et al. [11, 63] argue that even though DNNs are highly nonlinear overall, their behavior in high-dimensional spaces often resembles that of linear models. This makes the models vulnerable to small, targeted perturbations similar to how they're computed by FGSM. However some adversarial examples are successful all while defying the assumption of local linearity and reducing a model's linearity does not necessarily improve its robustness either [72].

Piecewise-linear decision boundaries. In the complementary *dimpled manifold hypothesis* [45] the central claim is that adversarial examples emerge because we attempt to fit high $n-1$ dimensional decision boundaries to inherently low-dimensional data like images (which can be losslessly projected to $k \ll n$ dimensions). This leaves redundant dimensions on which adversarial examples won't be judged, which enables them exist roughly perpendicularly from the true location of the low-dimensional natural image, by using large gradients. Additionally the authors differentiate between on-manifold and off-manifold adversarial examples, based on the angle of the gradients relative to the data manifold.

The authors suggest that decision boundaries of neural networks evolve during training. This happens through two distinct phases: First, there is a rapid "clinging" phase where the decision boundary moves close to the low-dimensional image manifold containing the training examples. This is followed by a slower "dimpling" phase that creates shallow bulges in the decision boundary, pushing it to the correct side of the training examples, without shifting the plane. This natural process based on gradient descent is highly efficient, but it also leaves a brittle decision boundary that can be easily exploited.

This implies that any attempt to robustify a network by limiting all its directional derivatives will make it harder to train and thus less accurate.

It also explains why networks trained on incorrectly labeled adversarial examples can still perform well on regular test images, as the main effect of adversarial training is simply to deepen these dimples in the decision boundary.

Lukas Karner successfully reproduced the experiments from the "Dimpled Manifold Hypothesis" paper in 2023 and demonstrated that dimensionality reductions increases the interpretability of the perturbations to humans [73].

However, most likely despite the experiments carried out being correct in themselves, the chain of reasoning might be flawed, as shown by a succinct counterexample by Yannik Kilcher in 2021 in less than 100 lines of code [74, 29]. While the "Dimpled Manifold Hypothesis" implies a relatively uniform vulnerability across all dimensions the counterexperiment contradicts these assumptions through successful adversarial attacks constructed by perturbing either an arbitrary subset of selected dimensions or their complement. If the decision boundary truly "clung" to the data manifold, restricting perturbations to a subset of dimensions would not have produced successful adversarial examples. The ability to generate adversarial examples in complementary subspaces suggests the decision boundary structure is more complex than just simple dimples.

1.6 Countermeasures

Chapter 2

Motivation

Finally, having discussed the theory behind adversarial examples, we can now turn our attention to the practical implications of these attacks. Adversarial examples have been shown to be effective in a wide range of applications, from computer vision to natural language processing to reinforcement learning. They have been used to fool state-of-the-art models, including those used in security-critical applications such as malware detection and autonomous driving. As a result, researchers and practitioners are increasingly interested in developing robust machine learning models that can withstand adversarial attacks.

Chapter 3

Methodology

3.1 Geometric masks inspired by hCaptcha

3.2 Self-ensembled learning

Chapter 4
Results

Chapter 5

Summary

"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 Opening Keynote [75]

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] 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.

[18] 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

[19] 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

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

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

[22] 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

[23] 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.

[24] 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.

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

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

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

[28] P. Ning, W. Jiang, and R. Wang, "Hflic: Human friendly perceptual learned image compression with reinforced transform," in *2023 International Conference on Communications, Computing and Artificial Intelligence (CCCAI).* IEEE, 2023, pp. 188–194.

[29] 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].

[30] 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.

[31] V. Veerabadran, J. Goldman, S. Shankar, B. Cheung, N. Papernot, A. Kurakin, I. Goodfellow, J. Shlens, J. Sohl-Dickstein, M. C. Mozer *et al.*, "Subtle adversarial image manipulations influence both human and machine perception," *Nature Communications*, vol. 14, no. 1, p. 4933, 2023.

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

[33] Z. Chen, Z. Wang, J.-J. Huang, W. Zhao, X. Liu, and D. Guan, "Imperceptible adversarial attack via invertible neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 414–424.

[34] M. Careil, M. J. Muckley, J. Verbeek, and S. Lathuilière, "Towards image compression with perfect realism at ultra-low bitrates," in *The Twelfth International Conference on Learning Representations*, 2023.

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

[36] R. Geirhos, K. Narayanappa, B. Mitzkus, T. Thieringer, M. Bethge, F. A. Wichmann, and W. Brendel, “Partial success in closing the gap between human and machine vision,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 885–23 899, 2021.

[37] S. McGuire, S. Jackson, T. Emerson, and H. Kvinge, “Do neural networks trained with topological features learn different internal representations?” in *NeurIPS Workshop on Symmetry and Geometry in Neural Representations.* PMLR, 2023, pp. 122–136.

[38] A. Murphy, J. Zylberberg, and A. Fyshe, “Correcting biased centered kernel alignment measures in biological and artificial neural networks,” *arXiv preprint arXiv:2405.01012*, 2024.

[39] A. Agafonov and A. Ponomarev, “An experiment on localization of ontology concepts in deep convolutional neural networks,” *Proceedings of the 11th International Symposium on Information and Communication Technology*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254045293

[40] A. Agafonov and A. Ponomarev, “Localization of ontology concepts in deep convolutional neural networks,” *2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pp. 160–165, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:256215614

[41] G. Bangaru, L. B. Baru, and K. Chakravarthula, “Interpreting bias in the neural networks: A peek into representational similarity,” *arXiv preprint arXiv:2211.07774*, 2022.

[42] Y. Bansal, P. Nakkiran, and B. Barak, “Revisiting model stitching to compare neural representations,” *Advances in neural information processing systems*, vol. 34, pp. 225–236, 2021.

[43] M. Huh, B. Cheung, T. Wang, and P. Isola, “The platonic representation hypothesis,” *arXiv preprint arXiv:2405.07987*, 2024.

[44] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

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

[46] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning.* PMLR, 2019, pp. 7472–7482.

[47] M. Khoury and D. Hadfield-Menell, "On the geometry of adversarial examples," *arXiv preprint arXiv:1811.00525*, 2018.

[48] S. Jha, U. Jang, S. Jha, and B. Jalaeian, "Detecting adversarial examples using data manifolds," *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pp. 547–552, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:57376324

[49] W. Sha, Y. Luo, Y. Wang, and Z. Pan, "A defensive approach against adversarial examples based on manifold learning," *2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET)*, pp. 167–171, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:222222557

[50] S. Dube, "High dimensional spaces, deep learning and adversarial examples," *arXiv preprint arXiv:1801.00634*, 2018.

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

[52] S. Dyrmishi, S. Ghamizi, T. Simonetto, Y. Le Traon, and M. Cordy, "On the empirical effectiveness of unrealistic adversarial hardening against realistic adversarial attacks," in *2023 IEEE symposium on security and privacy (SP).* IEEE, 2023, pp. 1384–1400.

[53] 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.

[54] L. Engstrom, J. Gilmer, G. Goh, D. Hendrycks, A. Ilyas, A. Madry, R. Nakano, P. Nakkiran, S. Santurkar, B. Tran, D. Tsipras, and E. Wallace, "A discussion of 'adversarial examples are not bugs, they are features'," *Distill*, 2019, https://distill.pub/2019/advex-bugs-discussion.

[55] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," *arXiv preprint arXiv:1801.09344*, 2018.

[56] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International conference on machine learning.* PMLR, 2018, pp. 5286–5295.

[57] K. Y. Xiao, V. Tjeng, N. M. Shafiullah, and A. Madry, "Training for faster adversarial robustness verification via inducing relu stability," *arXiv preprint arXiv:1809.03008*, 2018.

[58] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *international conference on machine learning*. PMLR, 2019, pp. 1310–1320.

[59] A. Fawzi, H. Fawzi, and O. Fawzi, "Adversarial vulnerability for any classifier," *Advances in neural information processing systems*, vol. 31, 2018.

[60] S. Mahloujifar, D. I. Diochnos, and M. Mahmoody, "The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4536–4543.

[61] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein, "Are adversarial examples inevitable?" *arXiv preprint arXiv:1809.02104*, 2018.

[62] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow, "Adversarial spheres," *arXiv preprint arXiv:1801.02774*, 2018.

[63] A. Madry, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[64] S. Bubeck, Y. T. Lee, E. Price, and I. Razenshteyn, "Adversarial examples from computational constraints," in *International Conference on Machine Learning*. PMLR, 2019, pp. 831–840.

[65] P. Nakkiran, "Adversarial robustness may be at odds with simplicity," *arXiv preprint arXiv:1901.00532*, 2019.

[66] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," *Advances in neural information processing systems*, vol. 31, 2018.

[67] T. Tanay and L. Griffin, "A boundary tilting persepective on the phenomenon of adversarial examples," *arXiv preprint arXiv:1608.07690*, 2016.

[68] B. Kim, J. Seo, and T. Jeon, "Bridging adversarial robustness and gradient interpretability," *arXiv preprint arXiv:1903.11626*, 2019.

[69] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "Robustness of classifiers: from adversarial to random noise," *Advances in neural information processing systems*, vol. 29, 2016.

[70] N. Ford, J. Gilmer, N. Carlini, and D. Cubuk, "Adversarial examples are a natural consequence of test error in noise," *arXiv preprint arXiv:1901.10513*, 2019.

[71] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, “Certified robustness to adversarial examples with differential privacy,” in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 656–672.

[72] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in *International conference on machine learning*. PMLR, 2018, pp. 274–283.

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

[74] Y. Kilcher, “Dimpled Manifold Counter Example,” https://gist.github.com/yk/de8d987c4eb6a39b6d9c08f0744b1f64/, 2021, [Online; accessed 17-July-2024].

[75] 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