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:
Andreas Plesner
Prof. Dr. Roger Wattenhofer

Alireza Furutanpey
Prof. Dr. Schahram Dustdar

December 30, 2024

Acknowledgements

> *"Well, at a high level, I think that the goal of computer security is to ensure that systems do the right thing, even in the presence of malicious inputs. Now, achieving this goal in the context of machine learning is exceptionally challenging for two reasons. [...] So first of all, computer scientists lack a deep mathematical understanding of how machine learning actually learns and predicts. [...] And second, many people who deploy machine learning don't actually care about the first problem."*
>
> *"And if somebody asks you why the stuff worked, you just say the stuff is what the stuff is brother, accept the mystery. Okay. And so basically machine learning is like this, right? So we've invented a bunch of techniques that kind of work, like in some cases, but we're not really sure what's going on. [...] 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'18 [1]

The most rewarding part of this project was working on a problem that truly matters to me, alongside people who genuinely care. For the first time, I felt a sense of belonging.

I'm deeply grateful to those who supported me along the way. My parents, Shima and Florian and my family, for their unconditional support – even when I quit my job to pursue my passion. My partner, Laura, whose love and encouragement crossed the Atlantic and carried 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, for keeping me grounded and focused.

I also value the friendships I made throughout this journey. Prof. Nils Lukas, who first introduced me to ML-Security and was always there to discuss ideas. Alireza Furutanpey, for his camaraderie and sharing his boundless passion.

Thanks as well to those whose paths have diverged from mine but whose impact remains with me, including Prof. Schahram Dustdar, who enabled me to study abroad.

To me, adversarial examples are also a metaphor for having a strong character by being open-minded. They show how subtle differences in perspective can lead to vastly different interpretations and outcomes.

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

Abstract

Adversarial machine learning has traditionally focused on imperceptible perturbations that fool deep neural networks. This thesis challenges that narrow view by examining unrestricted adversarial examples – a broader class of manipulations that can compromise model security while preserving semantics.

Through extensive experiments, we make three key contributions: First, we demonstrate that the standard imperceptibility constraint is insufficient for characterizing real-world adversarial threats through a comprehensive survey of current research. Second, we develop a novel and computationally efficient method for generating adversarial examples using geometric masks inspired by hCAPTCHA challenges. Our approach creates adversarial examples that are (1) effective, (2) transferable between models and (3) more traceable in the model's decision space – achieving comparable misclassification rates to existing techniques while requiring significantly less compute. Finally, we investigate improving model robustness by creating ensembles from intermediary ResNet layers using linear probes, combined with nature-inspired noise during training. While this architectural approach shows promise, we find that achieving "zero-cost robustness" remains elusive without adversarial training.

This work advances our understanding of adversarial examples beyond pixel-space perturbations and provides practical tools for both generating and defending against them. Our findings highlight the need to rethink how we conceptualize and evaluate adversarial robustness in machine learning systems.

Keywords: Reliability, Robustness, Security, Algorithmic Models

Zusammenfassung

Feindliche Angriffe wurden traditionell auf nicht wahrnehmbare Störungen beschränkt, die tiefe neuronale Netzwerke täuschen. Diese Arbeit stellt diese enge Sichtweise in Frage, indem sie unbeschränkte feindliche Angriffe untersucht, eine breitere Klasse von Manipulationen, die die Sicherheit des Modells gefährden können, ohne die Semantik von Eingaben zu beeinträchtigen.

Durch umfangreiche Experimente leisten wir drei wichtige Beiträge: Erstens zeigen wir, dass die Standardbedingung der Unwahrnehmbarkeit nicht ausreicht, um reale Bedrohungen zu charakterisieren, indem wir einen umfassenden Überblick über die aktuelle Forschung geben. Zweitens entwickeln wir eine neuartige und rechnerisch effiziente Methode zur Generierung von feindlichen Beispielen mit geometrischen Masken, die von hCAPTCHA-Herausforderungen inspiriert sind. Unser Ansatz erzeugt feindliche Beispiele, die (1) effektiv, (2) zwischen Modellen übertragbar und (3) im Entscheidungsraum des Modells nachvollziehbarer sind und vergleichbare Fehlklassifizierungsraten wie bestehende Verfahren erreichen, während sie deutlich weniger Rechenaufwand erfordern. Schließlich untersuchen wir die Verbesserung der Modellrobustheit durch die Erstellung von Ensembles aus ResNet-Zwischenschichten unter Verwendung von linearen Sonden in Kombination mit naturinspiriertem Rauschen während des Trainings. Während dieser architektonische Ansatz vielversprechend ist, stellen wir fest, dass sie keine "kostenlose Robustheit" erreicht.

Diese Arbeit erweitert unser Verständnis von feindlichen Beispielen über Pixel-Raum-Störungen hinaus und bietet praktische Werkzeuge, um sie zu erzeugen und sich gegen sie zu verteidigen. Unsere Ergebnisse unterstreichen die Notwendigkeit, die Art und Weise, wie wir die Robustheit von Systemen des maschinellen Lernens konzeptualisieren und bewerten, zu überdenken.

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.

I further declare that I have used generative AI tools only as an aid, and that my own intellectual and creative efforts predominate in this work. In the appendix "Overview of Generative AI Tools Used" I have listed all generative AI tools that were used in the creation of this work, and indicated where in the work they were used. If whole passages of text were used without substantial changes, I have indicated the input (prompts) I formulated and the IT application used with its product name and version number/date.

Yahya Jabary, December 30, 2024. Signed digitally.

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 locked dependencies.

Due to the exploratory nature of the work, many of the software built and experiments conducted had to be discarded.

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 build ResNet 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, running an Apptainer to provide root privileges and redirecting all file pointers to the EXT4 filesystem to avoid out-of-memory OS errors. Also runs a Jupyterlab instance on the intranet for ease of use.
`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. Also a simple job watchdog for long-running processes.
`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 I writing the majority of the text. Turlan and I conducting all experiments. The TU Wien DSG lab provided computational resources for robustifying a ResNet model, which we had to discard. Alireza Furutanpey suggested using LPIPS as a metric to evaluate the perceptual quality of adversarial examples, which we incorporated into our weighted objective function. Additionally, he helped with general advice on PyTorch.

Regarding the developed software, all contributions are my own, unless stated otherwise in the repository. A prototype of the self-ensembled ResNet 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 and Maximilian Seeliger 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 [2] 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].

1.1.1 Perturbation Methods

The origin of perturbations can be traced back to the early days of computational geometry by Seidel et al. in 1998 [3]. 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 [4]. 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) [5–10]. SoS systematically perturbs input coordinates using powers of a symbolic infinitesimal. For a point $p_i = (x_i, y_i)$, the perturbed coordinates become:

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

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

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.

1.1.2 Imperceptible Adversarial Examples

Adversarial examples, first introduced by Szegedy et al. in 2014 [11], 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. [12] 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 [12, 13]. 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.

Digression: Pixel-space constraints do not guarantee imperceptibility

Traditionally, adversarial examples are expected to have two key properties: (1) they should successfully cause misclassification in targeted models while (2) remaining imperceptible to human observers [14].

However, the concept of "imperceptibility [to humans]" as originally proposed by Szegedy et al. [11] 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 [15, 16].

Humans can detect forged low-ε adversarial examples with high accuracy in both the visual (85.4%) [17] and textual ($\geq$70%) [18] domain. It's worth mentioning that invertible neural networks can partially mitigate this issue in the visual domain [19].

Additionally, small ε-bounded adversarial perturbations are found to cause misclassification in time-constrained humans [20] and primates [21].

Intuition: The Deep Learning Hypothesis

Ilya Sutskever, in his "Test of Time" award talk at NeurIPS 2024 [22], revisited an idea he had previously only hinted at in interviews. This heuristic has since gained widespread acceptance and is now even taught in introductory machine learning courses [23].

The idea is straightforward: Human perception operates at a rapid pace. Neurons in the human brain can fire up to 100 times per second. Humans can complete simple perceptual tasks within 0.1 seconds. This implies that neurons fire in a sequence of at most 10 times for such tasks. Consequently, any task that a human can perform in 0.1 seconds can also be accomplished by a deep neural network with approximately 10 layers [22].

This could explain why adversarial examples transfer to time-constrained humans [20]. It also suggests that there may be a fundamental limit to the robustness of deep learning models, as they are inherently limited by the speed of their computations.

At first glance this idea might seem contradictory to the universal approximation theorem (UAT). However, the UAT only guarantees the existence of a network that can approximate any continuous function, not the efficiency or speed of computation [24].

While initially discovered in computer vision applications, the attack can be crafted for any domain or data type, even graphs [25]. Natural language processing models can be attacked by circumventing the discrete nature of text data [26–28]. Speech recognition systems are vulnerable to audio-based attacks, where crafted noise can cause system failure [29]. Deep reinforcement learning applications, including pathfinding and robot control, have also shown susceptibility to adversarial manipulations that can compromise their decision-making capabilities [30].

1.1.3 Semantics Preserving Adversarial Examples

Imperceptible noise-based adversarial examples are just one type of semantics-preserving adversarial examples. Other examples include rotating an image by a few degrees or capturing it from a different angle, which can also cause misclassification. These broader categories of adversarial examples are often referred to as "unrestricted" [31, 32] or "semantics-preserving" [33–35]. The comparison in Fig. 1.1 and the illustration in Fig. 1.2 highlight the differences between various kinds of adversarial examples. Fig. 1.3 shows a collection of naturally occurring adversarial examples, also known as "natural adversarial examples" [36, 37].

This shift in defining adversarial examples, popularized by the "Unrestricted Adversarial Examples

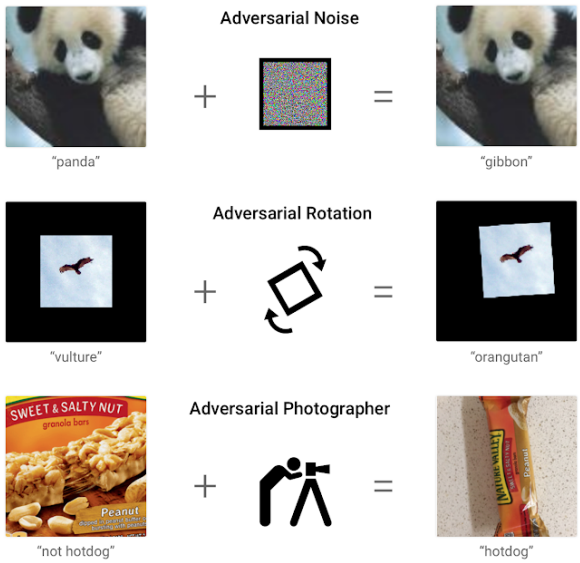


Figure 1.1: Unrestricted adversarial examples [32].

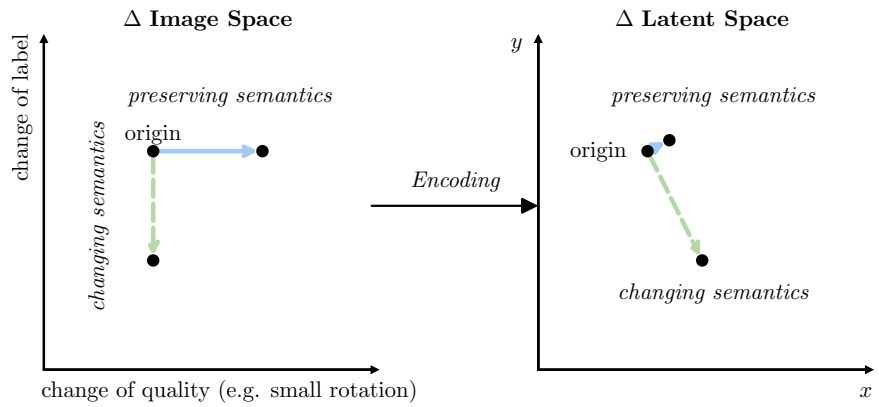


Figure 1.2: Semantics preserving/changing perturbations in pixel/latent-space (assuming full accuracy).

Challenge" [32] by Google in 2018, has led to a more nuanced understanding of the phenomenon. It acknowledges that real-world applications, especially in safety-critical contexts, are subject to a broader range of adversarial attacks than previously assumed and do not always adhere to the "small perturbation" constraint initially proposed [32].

This paradigm shift towards seeking more meaningful adversarial examples and "spatial robustness" was first proposed by Gilmer et al. in 2018 [38] and further explored by Engstrom et al. in 2019 [39]. These works lay the theoretical foundation for our research and we believe this approach to be the most promising for future research in adversarial machine learning.

The challenge of defining semantics is central to this discussion. Without perfect representations that align with human judgment functions, we must rely on the best available encoders or semantics preservation metrics [18, 39] as proxies. This pragmatic approach acknowledges the limitations of current technology while striving for more meaningful adversarial examples.

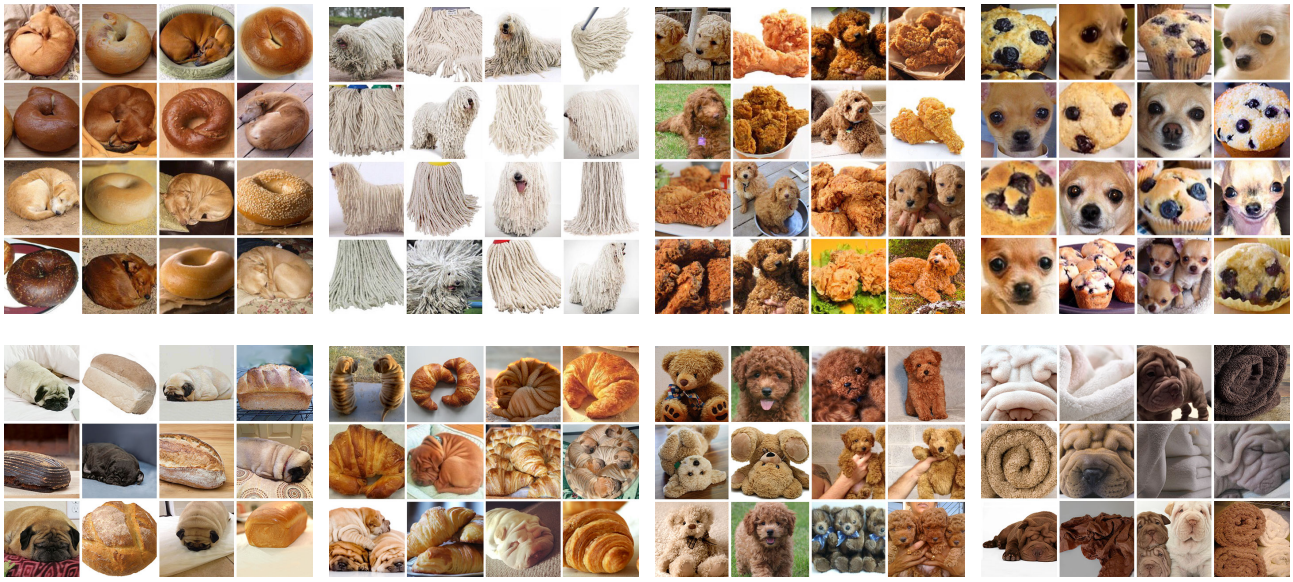


Figure 1.3: Natural adversarial examples [37]: Dog vs. Similar looking objects.

1.2 Motivation

Machine learning systems are growing rapidly in scale, capability[2] and are increasingly being deployed in critical applications [43–53].

Ensuring the safety of these systems is widely recognized as one of the most impactful fields for addressing global challenges [54–56].

Neglectedness. The field of ML safety – which encompasses research areas such as robustness (resilience to hazards), monitoring (hazard detection), alignment (guiding ML systems' behavior) and systemic safety (minimizing deployment risks) – remains significantly overlooked compared to other domains of machine learning [54, 56]. As estimated by Hilton et al. only about 0.1% of the resources dedicated to advancing AI capabilities in 2021 were allocated toward mitigating AI risks [57]. That is, despite a growing consensus that the risks posed by AI systems are significant and warrant urgent attention [55, 56] and a near-exponential surge in academic interest in adversarial machine learning since 2014 [58].

Impact. Adversarial attacks and machine learning security extend beyond academic curiosity. Algorithmic models like deep neural networks face several challenges when deployed in high-stakes scenarios. (1) They fail to provide clear explanations for decisions, which makes their outcomes hard to trust. (2) They create additional security risks by expanding the potential attack surface, which we do not yet fully know how to defend. (3) They can be weaponized to uncover unknown vulnerabilities (zero-days) at an unprecedented scale.

The issue with a lack of interpretability in high-stakes decisions and model vulnerability appears clearly in areas like autonomous weapons [59], critical national infrastructure [60–64], financial fraud detection [65–67], healthcare diagnostics [68–70], autonomous vehicles [71–73] and cybersecurity [74–76]. Another, more day-to-day example of a model's vulnerability is the exploitation of conference paper-reviewer assignment systems, enabling the adversary to preselect reviewers to gain a competitive advantage [76].

[2]Today's LLMs are no longer mere "stochastic parrots", as they demonstrate compositional generalization [40–42].

Finally, tools such as the VulnHuntr [77] and Big Sleep [78] models have high potential to be misused for malicious purposes. The latter has recently been particularly successful in automatically detecting zero-day exploits in SQLite [78].

This has lead to major companies investing heavily in adversarial machine learning research and security.

Funding. Microsoft has taken a leading position, spending over $20 billion on cybersecurity initiatives, with a significant portion dedicated to machine learning security research and their specialized ML red team operations [79].

Open Philanthropy has provided $330,000 and $343,235 in funding to Carnegie Mellon University dedicated to adversarial machine learning research [80].

The MITRE corporation is now cooperating with Microsoft, Bosch, IBM, NVIDIA, Airbus, Deep Instinct and PricewaterhouseCoopers to develop the Adversarial Machine Learning Threat Matrix for threat modeling and risk assessment [81].

The Defense Advanced Research Projects Agency (DARPA) has granted nearly $1 million to the computer vision and adversarial machine learning research team at UC Riverside [82]. Booz Allen Hamilton, the largest provider of machine learning services for the Federal government, invested in HiddenLayer, Robust Intelligence [83, 84] Shift5, Credo, Hidden Level, Latent, Synthetaic and Reveal Technology [85, 86], all of which are dedicated to machine learning security and robustness research.

These investments reflect a growing recognition of the importance of adversarial machine learning research and the need for robust, secure and reliable machine learning systems in the industry.

1.3 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 [87]. 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 [88]. Interestingly, research has shown that in some cases, black-box attacks can be more effective than white-box approaches at compromising model security [88].

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 [25, 88]. 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 [89].

The extent to which adversarial examples are transferable – meaning their ability to fool multiple different models or the human vision system [20] – 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 [90, 91]. Recent research has shown that adversarial examples are more readily transferable between vanilla neural networks than between defended ones [92, 93].

Finally, attacks can either focus on preserving the semantic meaning of inputs or exploit the mathematical properties of models without regard for semantic interpretation [33].

1.4 Latent Representations

The internal latent representations of neural networks, their alignment with human understanding and the resulting gap between the two (the human-machine vision gap [94]) 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 [95]. 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 [96]. 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 [97, 98]. 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 an 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 [99].

Recent research [100] 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", which suggests that neural networks are converging toward a shared statistical model of reality, regardless of their training objectives or architectures [101]. 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[3] 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 underlying statistical patterns and relationships that exist in reality [101].

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.

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

Recent work by Moosavi-Dezfooli et al. [102] 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 Mental Models

The question discussed in the previous section is just one of many that remain open and yet have to be fully explained [103]. Among them are:

- What are adversarial examples?
- Why are the adversarial examples so close to the original images?
- Why do the adversarial perturbations not resemble the target class?
- Why do robustness and accuracy trade-off [104]?
- Why do adversarial examples transfer between models, even on disjoint training sets [11]?
- Why do adversarial examples transfer between models [11]?
- Why do adversarial examples transfer between models and time-limited humans [20]?

Initially, when Szegedy et al. [11] 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 by Szegedy et al. [11] used the term "manifold", while referring to a data submanifold.

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 [103, 105–108].

A more rigorous definition of the manifold hypothesis is provided by Khoury et al. [105].

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 [109]:

$$U(x) = -\log P(x) - K(x)$$

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 [109].

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 [109].

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 [110].

The challenge of quantifying realism remains a fundamental problem in machine learning [109].

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 [111].

Non-robust features & concentration of measure in high dimensions. Most popularly, Ilyas et al. [111] 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 [112–121].

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 [111] they add a low-magnitude weight to sensitive variables that can get overamplified by adversarial examples [122, 123]. The assumption is that this happens due to computational constraints or model complexity.

Insufficient data. Schmidt et al. argue [124] 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. [111]. 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. [125, 126] 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 [116, 127–129]. 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. [12, 121] 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 are 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 [130].

Piecewise-linear decision boundaries. In the "dimpled manifold hypothesis" [103] 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 will not be judged, which enables them exist roughly perpendicularly from the true location of the low-dimensional natural image, by using large gradients. In this mental model adversarial examples can be on-manifold or off-manifold, based on the angle of the gradients relative to the data manifold.

The authors also 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 gradient-descent-based process 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 was able to successfully reproduce the experiments from the "Dimpled Manifold Hypothesis" paper in 2023 [131]. He additionally demonstrated that dimensionality reduction increases the interpretability of the perturbations to humans [131].

However, despite the experiments being carried out correctly themselves, the chain of reasoning might be flawed, as shown by a succinct (<100 LoC) counterexample by Yannik Kilcher in 2021 [132, 133]. While the "Dimpled Manifold Hypothesis" implies a relatively uniform vulnerability across all dimensions the counter experiment 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.

To summarize, there is no consensus on the root cause of adversarial examples and the field remains an active area of research. The mental models proposed by different researchers are not necessarily mutually exclusive and it is likely that the true explanation involves a combination of these factors.

1.6 Defenses

Having discussed the various theories and approaches in explaining adversarial examples, we can now turn our attention to the countermeasures that have been proposed to mitigate their impact.

1.6.1 Train- and Test-time defenses

A leaderboard of adversarial robustness can be found on the RobustBench platform [134], which provides a standardized evaluation of adversarial robustness across a wide range of models and datasets. The platform includes a variety of metrics for evaluating robustness, such as the ℓ_∞ and ℓ_2 adversarial perturbation sizes, as well as the robust accuracy under different attack settings.

Several effective strategies have been developed. This collection is by no means exhaustive.

Adversarial Training. One of the least invasive methods to improve adversarial robustness is adversarial training. Incorporating adversarial examples into the training process improves model resilience by learning from potential attack patterns and helps maintain performance on clean data [135, 136]. However, this requires the anticipated attacks to be known in advance. An alternative would be introducing derived variables for controlled randomness to input data during training, which is still effective [137].

Quality Assessment Integration. Implementing image quality assessment combined with knowledge distillation helps detect potentially harmful inputs that could cause incorrect model predictions [138]. Another alternative preprocessing technique is using brain-inspired encoders [139]. This

method is particularly effective as it does not require model retraining, but depending on the preprocessing technique used, it can be computationally expensive.

Moving Target Defense. Using heterogeneous models, diversifying the model structure, using ensembles and dynamic model switching can protect against white-box adversarial attacks. This approach will make attack vectors that work on one model ineffective on others [140].

Statistical Detection. Statistical tests can be employed for some signal-based deep learning systems to detect adversarial examples. This includes analyzing a peak-to-average-power ratio and examining softmax outputs of the model [141].

Enhanced Transformation. Transformation-based defense strategies, such as using generative adversarial networks (GANs), can help recover from adversarial examples. These methods can counteract adversarial effects while maintaining or even improving classification performance [142].

The countermeasures discussed so far provide a diverse array of techniques to mitigate the impact of adversarial examples. Each method addresses specific aspects of the problem, ranging from input preprocessing to model architecture adjustments and training methodologies. Notably, hybrid strategies that combine multiple techniques often yield the best results, with some implementations achieving reliable performance even under sophisticated attack benchmarks [143].

1.6.2 Architectural Defenses

Assuming that robustness and generalizability are not competing objectives but complementary goals, the ultimate defense lies in designing architectures that inherently integrate robustness and interpretability [144]. By prioritizing these objectives at the core of model development, we can create systems that not only withstand adversarial attacks but also offer more trustworthy and transparent decision-making.

Fermi-Bose Machine. One noteworthy example is the Fermi-Bose Machine [145]. Unlike traditional neural networks that rely on backpropagation, this method introduces a local contrastive learning mechanism inspired by quantum mechanics principles. The system works by making representations of inputs with identical labels cluster together (like bosons), while representations of different labels repel each other (like fermions). This layer-wise learning approach is considered more biologically plausible than traditional backpropagation [145]. The researchers demonstrated the effectiveness of their method on the MNIST dataset, showing that by adjusting the target fermion-pair-distance parameter, they could significantly reduce the susceptibility to adversarial attacks that typically disturb standard perceptrons [145]. The key innovation lies in controlling the geometric separation of prototype manifolds through the target distance parameter, as revealed by statistical mechanics analysis [145].

Ensemble everything everywhere. A recent (August 2024) state-of-the-art approach works by multi-resolution input representations and dynamic self-ensembling of intermediate layer predictions [146]. The researchers introduced a robust aggregation mechanism called CrossMax, based on Vickrey auction, which combines predictions from different layers of the network [146].

The method achieved impressive results without requiring adversarial training or additional data, reaching approximately 72% adversarial accuracy on CIFAR-10 and 48% on CIFAR-100 using the RobustBench AutoAttack suite [146]. When combined with simple adversarial training, the performance

improved further to 78% on CIFAR-10 and 51% on CIFAR-100, surpassing the current state-of-the-art by 5% and 9% respectively [146].

An interesting secondary outcome of this research was the discovery that gradient-based attacks against their model produced human-interpretable images of target classes [146]. Additionally, the multi-resolution approach enabled the researchers to transform pre-trained classifiers and CLIP models into controllable image generators, while also developing successful transferable attacks on large vision language models [146].

1.7 Future Directions

Perhaps we should be rethinking unrestricted adversarial examples not as attacks but as indicators of insufficient generalization, which cannot always be measured by accuracy on a predefined test set alone.

The most promising path forward may not lie in defending against these examples, but rather in fundamentally reimagining model architectures with reliability, robustness and interpretability as core design principles. This way, robustness becomes a natural byproduct of the model's structure, at no additional cost.

This perspective suggests that enhancing adversarial robustness requires developing new architectures from the ground up that inherently exhibit these properties, rather than patching existing systems.

Chapter 2

Experiments: HCaptcha Inspired Geometric Masks

When studying adversarial examples in computer vision, we are essentially dealing with a black box. Top-performing models are all deep neural networks, which are notoriously hard to interpret. This lack of interpretability means we cannot easily pinpoint why these models make certain decisions or more importantly, why they fail. This presents a challenge when trying to understand and mitigate adversarial vulnerabilities.

To address this challenge, we've adopted two guiding principles: (1) keeping it simple by using basic geometric shapes to overlay on images and (2) leveraging intermediary layers of neural networks, all to enhance interpretability.

Each principle will be discussed in a separate chapter.

2.1 Research Motivation

We noticed a gap in research when it comes to unrestricted adversarial examples that exploit the human-machine vision gap. One practical application for these types of attacks is in developing robust CAPTCHAs[1], which are widely used to differentiate between humans and bots. The aim is to create images that machines struggle or ideally fail to recognize, while remaining easily solvable by humans, to prevent denial-of-service attacks, spam, open-source intelligence scraping and other malicious activities.

Leveraging CAPTCHAs for adversarial research offers several key benefits: (1) Their real-world effectiveness is well-established, as demonstrated by widespread adoption from major providers like Google's reCAPTCHA and hCaptcha. Studying these systems provides a robust baseline with proven resilience, allowing researchers to build on existing strengths rather than starting from scratch. (2) CAPTCHAs present a well-defined challenge with clear success criteria, facilitating future studies on their transferability to human users. Moreover, this research can be conducted with fewer ethical, legal, and financial constraints compared to many other experimental setups.

A demo, shown in Figure 2.1, was built to showcase the potential of adversarial examples in CAPTCHAs and provide an emulation framework of reCAPTCHAv2 for penetration testing purposes. It is composed of two containers responsible for the challenger and the solver, respectively.

The first step in this direction was studying the effectiveness of CAPTCHA solvers for each provider, in the order of their market share.

Despite Google's reCAPTCHA having a global market share of at least 99% [147], they have been shown to be solvable with 100% accuracy using publicly available computer vision models on consumer hardware, by Plesner et al. [148].

[1] Completely Automated Public Turing test to tell Computers and Humans Apart

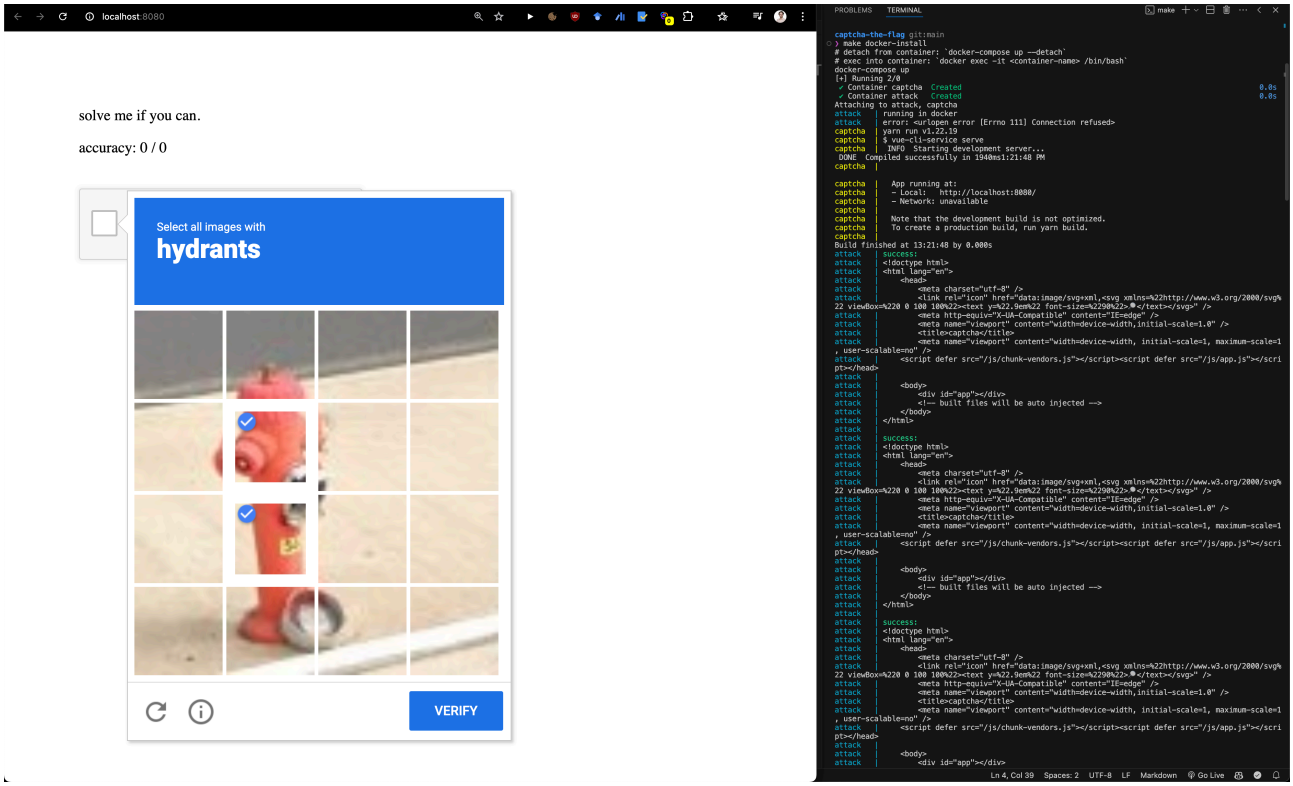


Figure 2.1: ReCAPTCHAv2 cybersecurity emulation framework
`https://github.com/ETH-DISCO/captcha-the-flag`

HCaptchas, on the other hand, have remained undefeated in the ongoing attack-defense arms race. In fact, several dedicated open-source communities collaborating on building a solver for hCaptcha report low success rates [149,150]. This is coupled with our observation on sophisticated defenses being rolled out by hCaptcha on an almost weekly basis. Within the last 6 months we were studying hCaptcha, we observed the obfuscation of metadata and payloads, the introduction of new reasoning-based challenges through question answering and the introduction of new classification and segmentation challenges. This leads us to believe that hCaptcha is the most robust CAPTCHA provider on the market today.

Two hCaptcha challenges were selected for our experiments: a classification challenge and a segmentation challenge. The classification challenge overlays images with simple, predictable patterns like grids of colored geometric shapes (e.g., circles or squares). The segmentation challenge works by embedding images within a Perlin-noise-like pattern and overlaying various unrestricted perturbations.

In this scenario, the solver / identity provider acts as the adversary, while the CAPTCHA provider is the challenger. The adversary aims to bypass the CAPTCHA, and the challenger's goal is to prevent this. The adversary's success is gauged by the solver's accuracy, whereas the challenger's success is measured by the CAPTCHA's effectiveness.

However, in traditional settings like automated content moderation on social media, the adversary aims to bypass the system by posting harmful content. Discovering simple geometric masks as robust black-box adversarial examples that transfer well between moderation systems would enable mass scale evasion at a fraction of the cost of traditional adversarial examples.

> **Idea:** Evaluation on Synthetic Data
>
> One interesting idea we explored but ultimately set aside was the use of synthetic images for

Figure 2.2: Evaluation on synthetic data.

adversarial training and evaluation of (natural unrestricted) adversarial examples. Although promising, it did not quite align with our primary goals. We developed a pipeline that chains together several advanced models: starting with stable diffusion to generate an image, then using GPT-2 to caption it. These captions were used as text queries for zero-shot classification and detection models like CLIP and ViT, which perform exceptionally well but need a prompt to work. Finally, we used SAM1 for segmentation. This process is illustrated in Figure 2.2. This is an exciting direction, especially with the advancements brought by FLUX.1 [151], not sufficiently explored in the current literature, based on our preliminary review.

```
img = gen_stable_diffusion("an astronaut on mars riding a horse")
query = caption_gpt2(img)
probs = classify_clip(img, query)
boxes, scores, labels = detect_vit(img, query, 0.1)
masks = segment_sam1(img, boxes)
```

2.2 Experimental Setup

The first series of experiments focused on evaluating the performance of state-of-the-art computer vision models on hCaptcha challenges. The goal was to assess the performance of solvers and identify potential vulnerabilities that could be exploited to generate CAPTCHA based adversarial examples. We targeted a single type of challenge: The classification challenge, which involves overlaying images with simple, predictable patterns like grids of colored geometric shapes (e.g., circles or squares).

We also studied and reconstructed the segmentation challenge, which embeds images within a Perlin-noise-like pattern, smooths the edges, adds a slight blur and sometimes incorporates the geometric masks from the classification challenge on either individual segments or the entire image. However, given the number of variables involved, we decided to focus on the classification challenge for our initial experiments and leave the other tasks for future work. Our goal was to establish a reliable baseline, not to exhaustively explore all possibilities. However, we noticed through small-scale experiments using SAM1 and SAM2 that it was significantly more difficult to solve than the classification challenge.

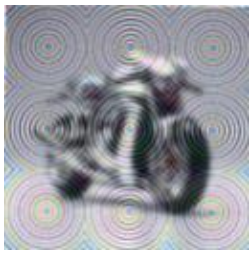
Mask Generation. We generated adversarial examples using 4 geometric shapes.

We developed a sublibrary that allows for the parametrizable reconstruction of geometric masks used in the classification task using the rendering engine `pycairo`. We created four distinct masks to overlay on images at varying intensities: “Circle”, “Diamond”, “Square” and “Knit” (the “Word” mask was

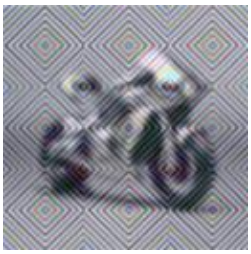
also reconstructed, but omitted as they have been proven to be easy to mitigate [152–154]).

These masks were chosen based on an experiment where we hand-labeled 1600 images from hCaptcha[2]. The opacities (alpha values) and densities of these masks were determined through an initial hyperparameter search, which we'll discuss later. Figure 2.3 showcases the optimized reconstructions we used to benchmark the models.

For the segmentation task, we experimented with various background textures, including Perlin noise and color-encoded multivariate Gaussian distributions. Figure 2.4 shows both a selected example and its reconstruction. But as mentioned earlier, we decided to focus on the classification challenge for our initial experiments.



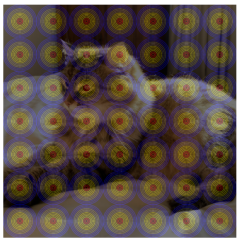
(a)"Circle" mask



(b)"Diamond" mask

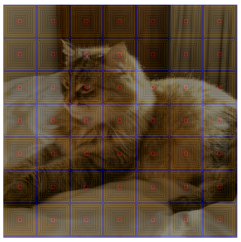
(c)"Square" mask

(d)"Word" mask

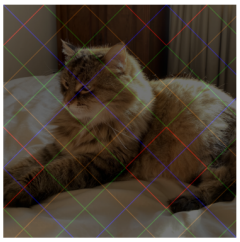


(e)"Circle" mask (reconstructed)

(f)"Diamond" mask (reconstructed)



(g)"Square" mask (reconstructed)



(h)"Knit" mask (custom)

(i)Original image (for reconstructions)

Figure 2.3: Selected examples by hCAPTCHA and their optimized reconstructions. The "Word" overlay was omited and replaced with a custom "Knit" mask.

Figure 2.4: Segmentation Challenge.

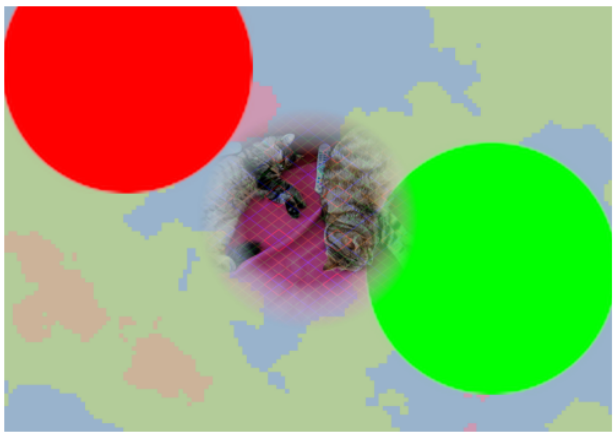


Figure 2.5: Reconstruction (arbitrary parameters).

[2]Credits to Turlan Kuzhagaliyev.

Model Selection. We benchmarked 5 SOTA models, that can run on consumer hardware.

To identify the best zero-shot open vocabulary classification, object detection and segmentation models that can run on consumer hardware, we did a near exhaustive literature review. We looked at focused research papers [155, 156] and checked out public leaderboards from HuggingFace, TIMM, PapersWithCode, GitHub Trends, PyTorch benchmarks and more. Our goal was to ensure that the solver could break CAPTCHAs on any machine with minimal setup.

After the literature review we clustered the models by their architecture family and clustered them on a scatter plot based on their performance, inspired by [157]. We then selected the top models from each family and attempted to run them on a consumer machine, assuming that they are publicly available. This was to test the feasibility of running these models in a real-world scenario.

We chose several models to test, including "ConvNeXt_XXLarge" [158], Open CLIP's "EVA01-g-14-plus" [159] and "EVA02-L-14" [160], the original "ViT-H-14-378-quickgelu" [161] and "ResNet50x64" [162]. We highlight results for a subset of these models, specifically ConvNeXt, EVA01, EVA02, ViT-H-14 and ResNet50. These models were picked to cover key architectures in both convolutional and transformer-based approaches, allowing us to see how well our masks work across different architecture families. Additionally, we also evaluated a custom adversarially trained ResNet-50, but due to its poor performance, we decided to exclude it from the final results.

When specific models are not mentioned in tables, experiments or figures, we are implicitly referring to the average performance of all the models listed.

Dataset Selection. We evaluated against an enriched ImageNet dataset.

For experiments we utilized the enriched ImageNet dataset provided by "visual-layer" on HuggingFace. This dataset includes 1,000 distinct classes, offering a comprehensive range of categories for our experiments.

Evaluation Metrics. We measured the drop in accuracy and perceptual quality using a proxy metric.

Perceptual quality is about how visually similar the adversarial examples are to the original images. We used a weighted average metric to get a comprehensive view of image quality. This metric combines cosine similarity (15% weight) [163], Peak Signal-to-Noise Ratio (PSNR, 25% weight) [164], Structural Similarity Index (SSIM, 35% weight) [165] and Learned Perceptual Image Patch Similarity (LPIPS, 25% weight) [166]. These weights were chosen to balance the importance of each component in assessing overall image quality. We decided to omit the Fréchet Inception Distance (FID), due to its high computational cost. Overall, we want this number to be as high as possible.

$$\text{Perceptual Quality} = 0.15 \times \text{Cosine Similarity} + 0.25 \times \text{PSNR} + 0.35 \times \text{SSIM} + 0.25 \times \text{LPIPS}$$

On the accuracy front, we looked at how well the models performed in predicting the correct class. The models output a list of classes sorted by likelihood. We measured accuracy using the top-k accuracy metric, specifically accuracy@1 (Acc@1) and accuracy@5 (Acc@5). This tells us how often the correct label is in the top 1 or top 5 predicted classes, respectively. Finally, we calculated the adversarial accuracy (AdvAcc) as the difference in accuracy between adversarial and benign examples. This metric gives us a sense of how much the performance decreases relative to the original accuracy. We want this number to be as low as possible.

$$\text{Adversarial Accuracy} = \frac{(\text{Acc@1}_{\text{adv}} + \text{Acc@5}_{\text{adv}})/2 - (\text{Acc@1} + \text{Acc@5})/2 + 1}{2}$$

This brings us to our final evaluation metric, the final score (Score). It is calculated as the difference between perceptual quality and adversarial accuracy. Ideally, we want this number to be as high as possible.

$$\text{Final Score} = \text{Perceptual Quality} - \text{Adversarial Accuracy}$$

Hyperparameter Search. A grid search attempted to find the best mask parameters.

Having determined our evaluation metrics, dataset and models, we moved on to the hyperparameter search to find the optimal adversarial masks to benchmark with. We parameterized three variables: "opacity" (alpha value of the overlay), "density" (shapes per row/column and nesting, ranging from 0-100) and "epsilon" (for white-box FGSM attacks with CLIP-ViT on ImageNet). Using a grid search, we aimed to find the optimal values for these parameters.

We ran a hyperparameter grid search using the "visual-layer/imagenet-1k-vl-enriched" dataset on HuggingFace, testing 5-20 examples per combination on the validation set. For this phase, we exclusively used the CLIP ViT model due to its strong adversarial robustness, as highlighted by [167].

Our optimization metric combined the difference in model accuracy before and after applying the mask with an average of three perceptual quality metrics. To find the best parameters, we selected examples with the highest perceptual quality for each level of accuracy difference and performed a linear regression. We then focused on samples above the regression line in multidimensional space. This approach was more manageable than our attempts with multi-objective optimization involving multiple variables.

Our initial grid search revealed some interesting patterns. We found that achieving high perceptual quality generally required a combination of low opacity and high density. Specifically, the optimal opacity range appeared to be between 50 and 200. Additionally, higher density values consistently yielded better results. Among the various masks we tested, the diamond mask stood out as offering the best balance between effectiveness and perceptual quality. Based on these findings, the diamond mask seemed most promising for the following steps of our experiments. Figure 2.6 shows the trade-off between accuracy and perceptual quality for the strongest masks, where masks are visually encoded through shapes, opacities through alpha values and densities through the size of the shapes. A linear regression line is shown to highlight the areas of interest.

Our second grid search, which combined FGSM perturbations with the chosen masks, revealed some interesting insights. We found that combining perturbations with the masks generally led to worse results. The best results were obtained using the diamond mask with FGSM disabled, a density of 50 and opacities of 150 or 170. This further validated our previous findings and hinted that FGSM should be used with caution when combined with masks. Figure 2.7 shows the trade-off between accuracy and perceptual quality for the diamond mask combined with different opacities, densities and perturbation settings.

While these results are not representative of the entire dataset or model space, they provide a good starting point for our experiments. We can conclude not to use perturbations with the masks. These initial findings also helped us identify the best configurations in opacity and density, to be used in the following experiments. The final configurations of each mask are shown in figure 2.3.

2.3 Results

Attack Transferability. We evaluated the individual masks for their transferability between different models by averaging the "final score" across all models. This score serves as a proxy for the trade-off between adversarial effectiveness and perceptual quality. We found that opacity was the most

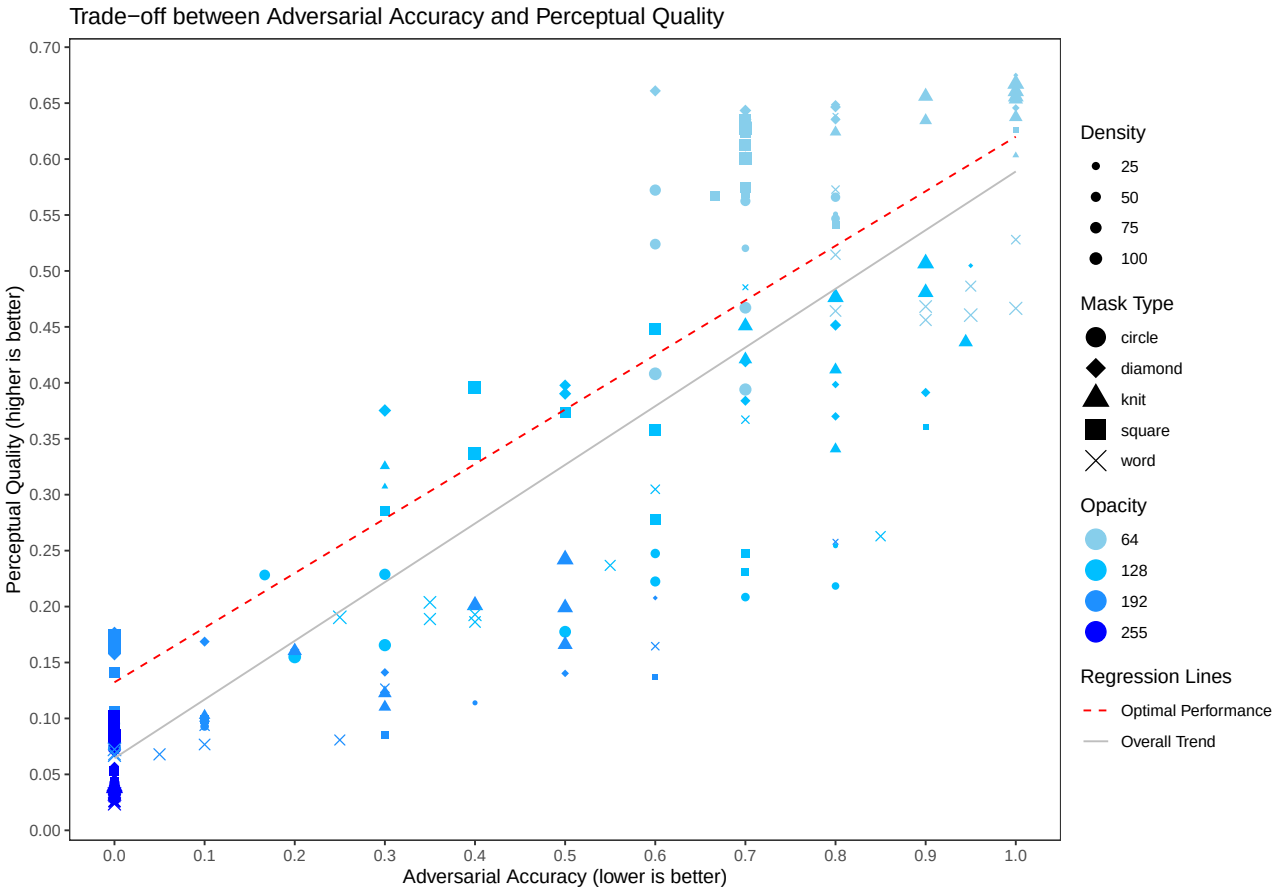


Figure 2.6: Accuracy-perceptibility trade-off: We compare the strength of all our masks with different opacities and densities against their perceptibility based on our proxy metric.

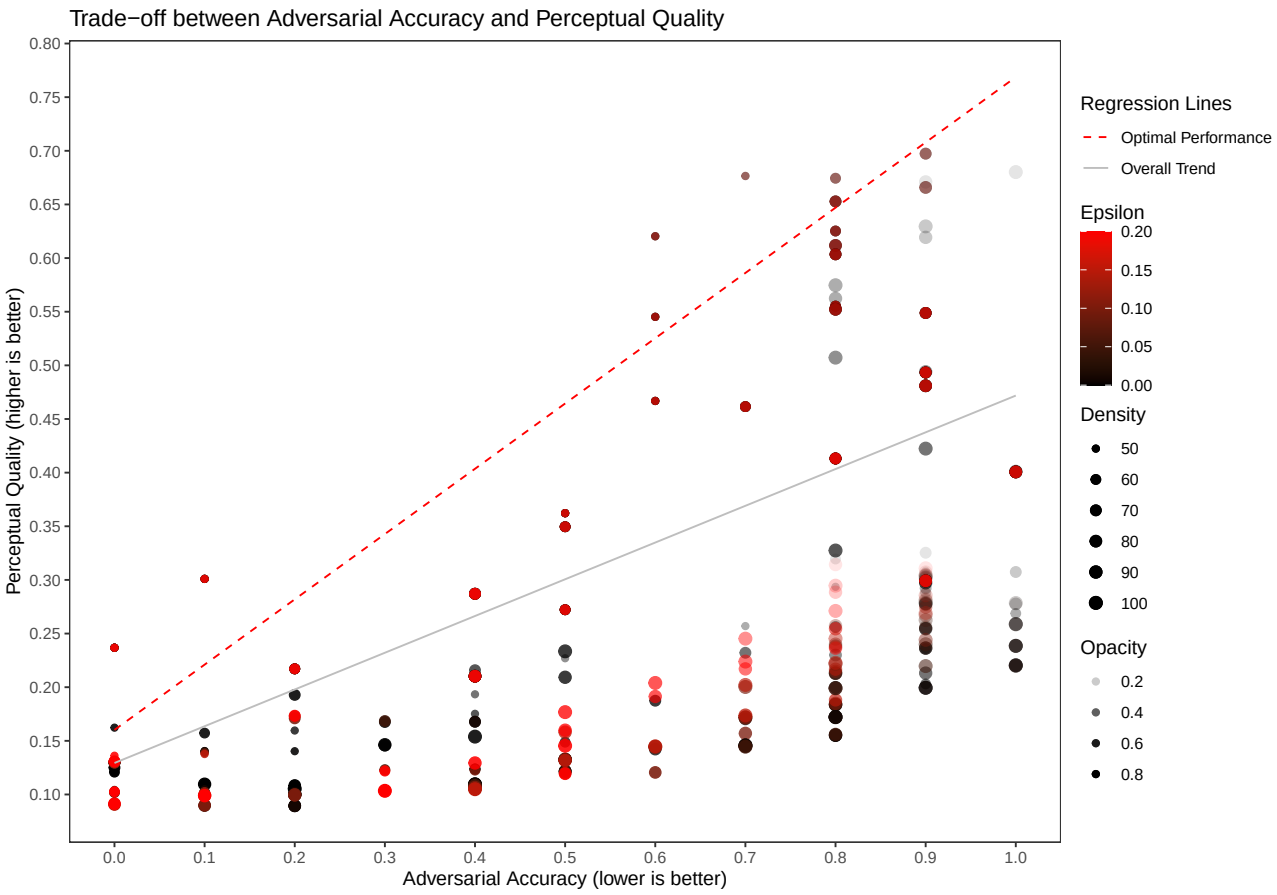


Figure 2.7: Accuracy-perceptibility trade-off: We compare the strength of a diamond mask with different opacities, densities and FGSM perturbation settings against their perceptibility based on our proxy metric.

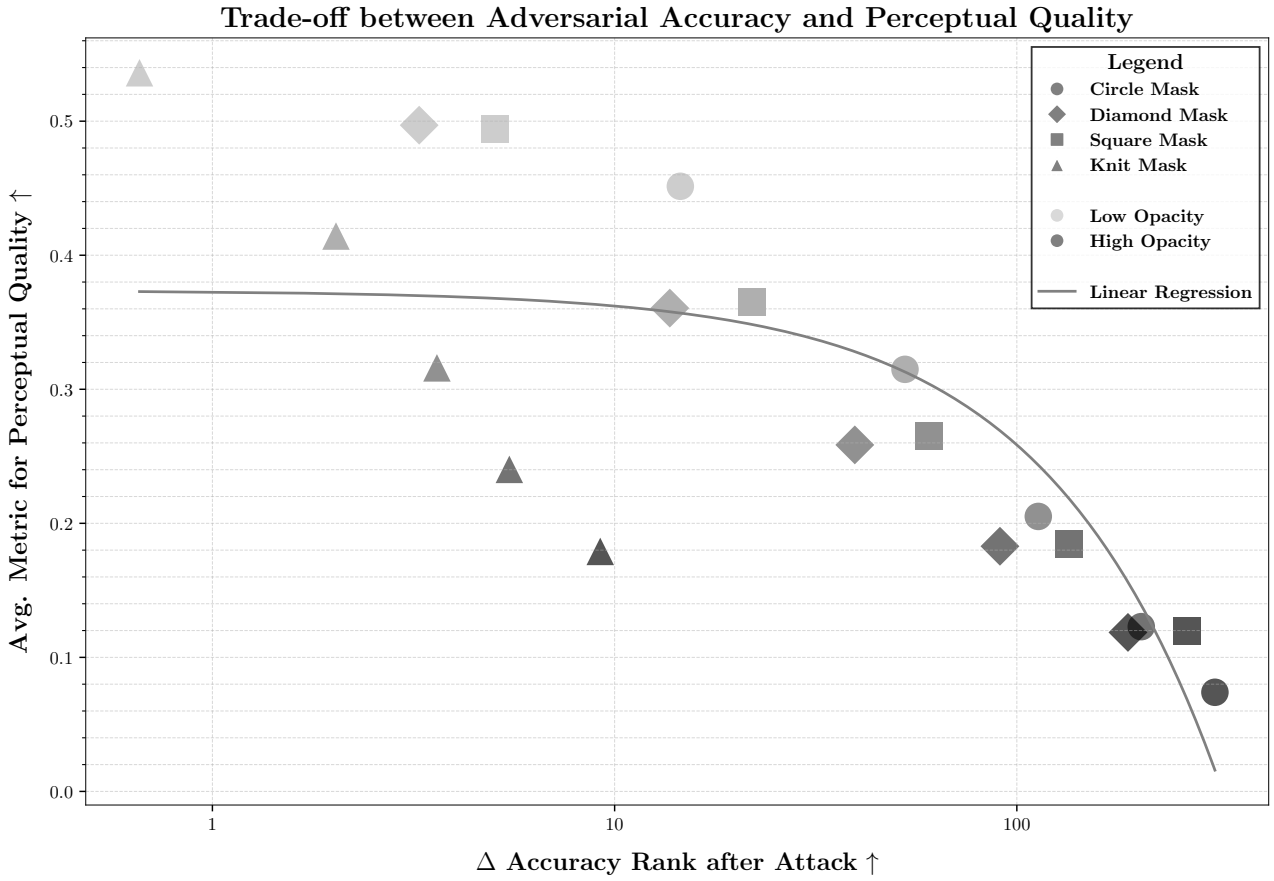


Figure 2.8: Accuracy vs. Perceptual Quality Trade-off

significant parameter, regardless of mask specifics. Therefore, we chose the best-performing density parameter from the hyperparameter search and applied different masks and opacities to the models.

The results of our experiment are visualized in Figure 2.8 and summarized in Table 2.1. The figure shows a clear trend in the trade-off between adversarial effectiveness and perceptibility. The plot highlights an inverse relationship between these two factors, as indicated by the polynomial regression curve of degree 1. This relationship suggests that as the effectiveness of the adversarial attack increases (lower Δ Accuracy Rank), the perceptual quality of adversarial examples tends to decrease. While this is somewhat expected, we do see instances where there's a significant drop in rank (>10) while maintaining relatively high perceptual quality (>0.4).

The different mask types (circle, square, diamond and knit) and opacity levels show a range of performance across this trade-off spectrum. The scatter plot highlights clusters of points for each mask type, with some masks consistently striking a better balance between attack effectiveness and perceptual quality. Most importantly, these geometric pattern masks transfer well between state-of-the-art models.

The circle masks, contrary to our initial hyperparameter search, seem the most effective at reducing accuracy across all models and opacities. The knit mask, on the other hand, has a much smaller impact on accuracy.

These results demonstrate that using geometric masks is a practical method for generating adversarial examples with minimal computational demands for benchmarking in future experiments.

Model Robustness. We inspected individual models, examining both the drop in acc@1 and acc@5. This will give us a more detailed understanding of how the masks impact the individual models. The

Opacity	Mask	Δ Acc Rank	Quality	Score
50	Circle	-14.57	0.45	15.02
	Diamond	-3.27	0.50	3.76
	Knit	-0.66	0.54	1.19
	Square	-5.04	0.49	5.54
80	Circle	-52.72	0.31	53.03
	Diamond	-13.72	0.36	14.08
	Knit	-2.03	0.41	2.44
	Square	-22.01	0.37	22.37
110	Circle	-113.07	0.21	113.27
	Diamond	-39.55	0.26	39.81
	Knit	-3.62	0.32	3.93
	Square	-60.57	0.27	60.84
140	Circle	-203.89	0.12	204.01
	Diamond	-90.79	0.18	90.97
	Knit	-5.47	0.24	5.71
	Square	-134.75	0.18	134.94
170	Circle	-310.80	0.07	310.88
	Diamond	-188.92	0.12	189.04
	Knit	-9.21	0.18	9.39
	Square	-264.90	0.12	265.02

Table 2.1: Transferability of Masks

results are summarized in Table 2.2 and Table 2.3 for top-1 and top-5 accuracy, respectively.

A negative value indicates an improvement in accuracy, while a positive value indicates a decrease. Larger values indicate a stronger effect of the adversarial examples.

While the previous perspective on the data compared masks against each other, this view allows us to compare the models against each other. We can see that while all models show very similar overall trends in their behavior, the ResNet model seems particularly sensitive to the Diamond and Knit pattern in contrast to the others. This is particularly interesting, given that the Knit pattern is the least effective mask overall, as seen in the previous step.

2.4 Conclusion

In our first set of experiments, we tested geometric masks against state-of-the-art models. While the results yielded several insights, two key takeaways stand out as especially relevant for the rest of this work.

Geometric mask attacks are simple, effective and transferable. Our experiments show that hCAPTCHA-inspired geometric masks are (1) simple, (2) effective, noticeably reducing model accuracy, and (3) transferable, as they perform similarly across different models. The advantage they offer over traditional imperceptible black-box adversarial examples is their simplicity, which makes it feasible to reason about their influence on models, even in latent spaces. This enables us to break down the complexity of adversarial vulnerabilities and understand the underlying mechanisms that make them effective.

Model	Mask	Opacity 19%	31%	43%	54%	66%
ConvNeXt	Circle	13.0	33.6	51.2	64.6	69.2
	Diamond	4.8	13.6	31.8	49.6	64.6
	Knit	2.2	3.2	8.0	11.4	18.0
	Square	6.8	18.4	36.4	52.0	65.6
EVA01	Circle	7.2	15.4	33.0	49.2	65.0
	Diamond	2.6	8.6	19.6	33.0	54.8
	Knit	1.2	1.2	4.4	6.6	10.6
	Square	4.2	9.0	17.4	31.4	55.8
EVA02	Circle	9.4	19.0	31.4	50.4	63.8
	Diamond	2.4	5.6	10.6	19.0	38.0
	Knit	2.8	4.8	5.2	6.8	8.8
	Square	6.8	12.4	20.8	37.4	61.8
ResNet	Circle	31.0	54.6	60.0	62.4	63.4
	Diamond	13.2	31.6	50.4	59.4	62.2
	Knit	5.0	11.2	14.4	19.4	27.6
	Square	15.2	38.8	56.0	62.2	63.4
ViT-H-14	Circle	5.8	20.6	48.2	70.8	80.2
	Diamond	2.0	5.4	15.2	34.4	61.8
	Knit	1.6	2.4	2.8	6.2	8.0
	Square	3.2	9.6	25.0	54.2	77.2

Table 2.2: Change of Acc@1 in [%].

Model	Mask	Opacity 19%	31%	43%	54%	66%
ConvNeXt	Circle	7.60	29.60	54.80	73.40	85.00
	Diamond	2.60	8.80	24.20	51.40	71.60
	Knit	1.80	2.20	4.60	7.80	13.20
	Square	4.80	13.20	28.80	54.80	76.80
EVA01	Circle	4.80	14.00	27.80	50.60	75.40
	Diamond	2.40	6.60	14.80	31.00	57.60
	Knit	1.40	2.80	4.60	6.20	8.00
	Square	3.40	7.00	12.60	28.20	61.00
EVA02	Circle	4.60	12.20	24.40	44.60	65.00
	Diamond	1.40	3.60	6.60	14.80	34.80
	Knit	0.40	0.40	1.80	3.20	4.80
	Square	2.20	6.60	15.00	31.40	63.60
ResNet	Circle	34.20	67.40	80.40	85.40	86.20
	Diamond	12.20	28.80	56.00	75.60	85.00
	Knit	4.40	8.00	10.20	15.00	20.80
	Square	15.20	40.20	66.40	82.20	86.60
ViT-H-14	Circle	2.60	16.20	46.00	77.60	90.80
	Diamond	0.20	2.20	10.60	28.60	61.20
	Knit	-0.60	0.60	1.00	2.00	3.20
	Square	1.40	6.40	18.60	50.20	82.80

Table 2.3: Change of Acc@5 in [%].

ResNet is an outlier. We observed that most state-of-the-art architectures exhibited similar accuracy distributions when subjected to geometric masks. However, ResNet stood out as unusually sensitive to the Diamond and Knit pattern – a curious finding, given that this pattern is the least effective mask overall when averaged across all models. This suggests that the ResNet model results should be interpreted with caution, as they may not fully represent the behavior of other models under geometric mask attacks.

Chapter 3

Experiments: Self-Ensembled ResNet

In this chapter, we present our findings on the self-ensembled ResNet model.

3.1 Research Motivation

The self-ensembled ResNet [146], as presented in the introduction, is a novel approach to enhancing adversarial robustness. It combines multi-resolution inputs with dynamic self-ensembling of predictions from intermediate layers of the neural network. The multi-resolution input strategy involves feeding the model multiple versions of an image at different resolutions, inspired by biological mechanisms like eye saccades, which enhances robustness by forcing the network to process diverse representations simultaneously. The self-ensembling aspect leverages the inherent robustness of the intermediate layer predictions, which are less affected by adversarial attacks targeting the final classifier. These predictions are aggregated using a consensus algorithm called "CrossMax", inspired by Vickrey auctions, which dynamically selects consistent outputs across layers:

```python
def get_cross_max_consensus_logits(outputs: torch.Tensor, k: int) -> torch.Tensor:
    # subtract the max per-predictor over classes
    Z_hat = outputs - outputs.max(dim=2, keepdim=True)[0]
    # subtract the per-class max over predictors
    Z_hat = Z_hat - Z_hat.max(dim=1, keepdim=True)[0]
    # get highest k values per class
    Y, _ = torch.topk(Z_hat, k, dim=1)
    # get the k-th highest value per class
    Y = Y[:, -1, :]
    assert Y.shape == (outputs.shape[0], outputs.shape[2])
    assert len(Y.shape) == 2
    return Y
```

Architecturally, what makes this approach interesting is its ability to use a single ResNet model to create a "self-ensemble" by decoupling and aggregating predictions from intermediate layers. This eliminates the need for multiple independent models while still achieving ensemble-like robustness. Additionally, the use of multi-resolution inputs and stochastic augmentations reduces the attack surface for adversarial perturbations.

To the best of our knowledge, we are the first team to reproduce, evaluate and improve upon this approach.

3.2 Experimental Setup

We conducted a series of experiments on the self-ensembled ResNet and the standard ResNet-50 as a baseline with the same weights, trained on ImageNet to evaluate the new architecture against. We build upon our findings from the previous chapter.

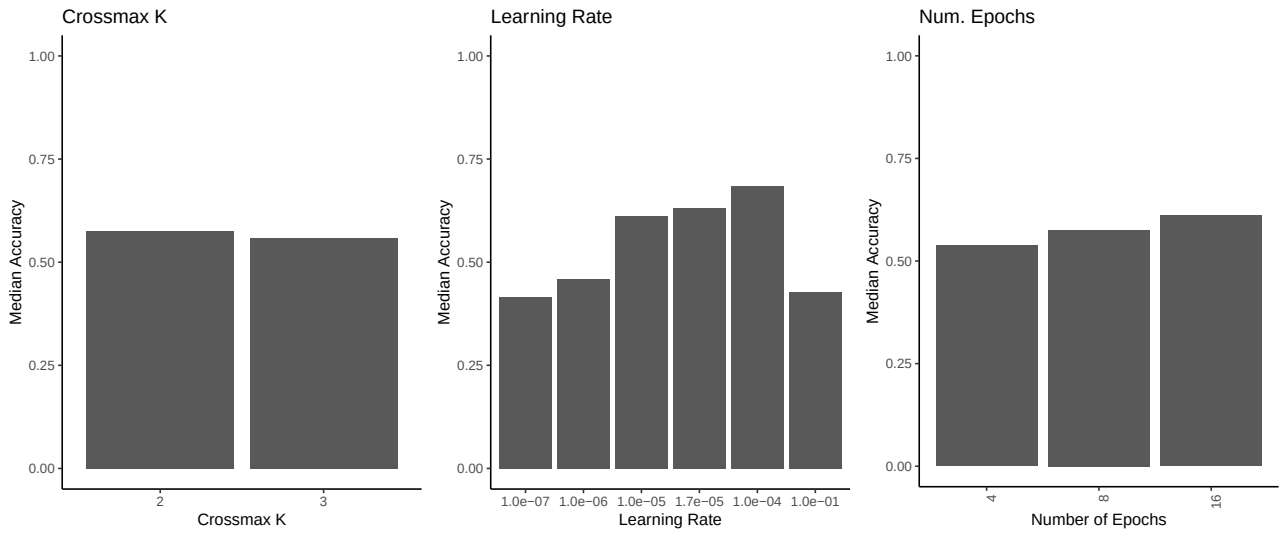


Figure 3.1: Hyperparameter search on the reproduced architecture.

Reproduction. We discarded our reproduction after the authors shared their code.

We chose to recreate the architecture[1], the CrossMax consensus algorithm and our own distributed multi-GPU training setup from the ground up, as the authors had not yet shared their codebase, when we started our experiments. This allowed us to gain a deeper understanding of the model and validate the hyperparameters used in the original paper.

We ran a set of experiments on CIFAR10 and CIFAR100 to confirm our reproduction was sensible and to find the best hyperparameters. We used the same ResNet architecture as the authors but slightly adjusted the training pipeline. Our hyperparameter search space consisted of: the dataset (CIFAR10, CIFAR100), learning rate (1e-1, 1e-4, 1e-5, 1.7e-5, 1e-6, 1e-7), number of epochs (4, 8, 16) and CrossMax k (2, 3).

We found the best results with a learning rate of 1e-4, running for 16 epochs and setting CrossMax k to 2. Interestingly, running more epochs only bumped up performance by less than 1%, so we stuck with 16 epochs. For CrossMax k, 2 was the sweet spot for both datasets, although CIFAR10 showed a tiny edge with $k = 3$ by just 0.1%. These results are visualized in Figure 3.1. When the authors later shared parts of their implementation and hyperparameters, our findings were in line with theirs, except for slight differences in the learning rate (3.3e-5 vs 1e-4) and the number of epochs (6 vs 16). We believe these differences to be due to the computational limits they set for their experiments. Figure 3.1 shows the results of our hyperparameter search on our reproduction of the self-ensembled ResNet.

After the authors shared their codebase, we transitioned to their implementation to maintain consistency and ensure our results were directly comparable.

Training Strategy Selection. We compared 2^3 training configurations.

The author's code included several improvements not mentioned in the paper, such as random resolution shuffling, which were recommended in the code comments for better results.

We aimed to keep as many hyperparameters unchanged as possible while comparing the impact of three specific training enhancements: (1) nature-inspired noise, (2) channel shuffling and (3) light FGSM adversarial training, resulting in 2^3 possible training configurations.

The nature-inspired noise works by adding noise, applying contrast adjustments, shifting pixel posi-

[1]Credits to Andreas Plesner.

tions (jittering) and modifying resolutions. In the last step it combines these augmented outputs into a multichannel input to enhance training. The random channel resolution shuffling randomly shuffles the resolutions in the stack, while the light FGSM adversarial training applies FGSM adversarial training for a few epochs to improve adversarial robustness.

Initially, we experimented with each technique independently to evaluate their effects. For example, applying random shuffling alone reduced adversarial robustness, whereas combining all three consistently produced the best results. We did not observe any negative interactions between these techniques, as they all seemed to complement each other. The results of these experiments are visualized in Figure 3.2 and Figure 3.3. This led us to either exclusively use all three techniques or none at all in our subsequent experiments. We simply refer to this enhanced training strategy as "natural training" (or just adversarial training) in the following.

Our initial evaluations of the training strategies focused on CIFAR-10 and CIFAR-100, as specified in the authors' codebase. While the original paper also utilized MNIST, we opted for Imagenette due to its relevance to higher-resolution tasks. However, the aggressive downsampling and upsampling in the code proved problematic for Imagenette, leading us to exclude it from our preliminary evaluation. These issues were addressed in our codebase for future experiments.

Dataset Selection. We evaluated on CIFAR-10 only.

Despite implementing and configuring our evaluation pipeline for 3 datasets (CIFAR-10, CIFAR-100 and Imagenette), we only evaluated CIFAR-10 due to time constraints. These constraints were due to the limited availability of GPUs with at least 80 GB memory, which is required by the self-ensembled ResNet model. We were unable to reduce the model's memory requirements, but we did identify compute bottlenecks in the codebase, some of which we also addressed as a pull request to the authors.

However, as visible in Figure 3.2 vs. Figure 3.3, classifying CIFAR-10 is clearly less challenging than other datasets for the self-ensembled ResNet and this dataset is not representative for the others. Therefore, our results should be taken with a grain of salt and further evaluation is necessary for more conclusive results. This should however not diminish the value of our findings on CIFAR-10 and our methodological contributions.

Attack Selection. We attacked the model with FGSM, PGD and 3 geometric masks.

The authors use a general benchmarking suite and visualize their attacks against FGSM. However, after some consultation[2] and manual inspection, we decided to also include Projected Gradient Descent (PGD) attacks in our evaluation.

For the geometric masks, we extended our rendering pipeline from previous experiments. Building on the observation that opacity and density of a mask are generally relevant across all models and the effectiveness of the "Knit" mask on the vanilla ResNet architecture (which we also kept, see Figure 3.4), we parametrized our mask generator with the following parameters: the number of sides (3, 4, 6, 10), the shapes per row and column (2, 4, 10), the number of concentric shapes (1, 2, 3, 4) and colors (True, False). We generated masks for all combinations of these parameters for our experiments. Selected examples of these masks are shown in Figure 3.4.

To narrow down the 96 geometric mask attacks, we did a preliminary exploratory analysis. We focused on the stronger ensemble accuracy (with and without natural training) since the ensemble consistently outperformed the last layer in all challenging experiments. This is further discussed in the results section. Figure 3.5 shows the visual results of this analysis. We found that having 4 shapes per column and row, combined with 2 concentric shapes, was the most effective. Surprisingly, the number of sides had little impact on the mask's effectiveness, although triangles performed best

[2]Thanks to Nicholas Carlini for his advice.

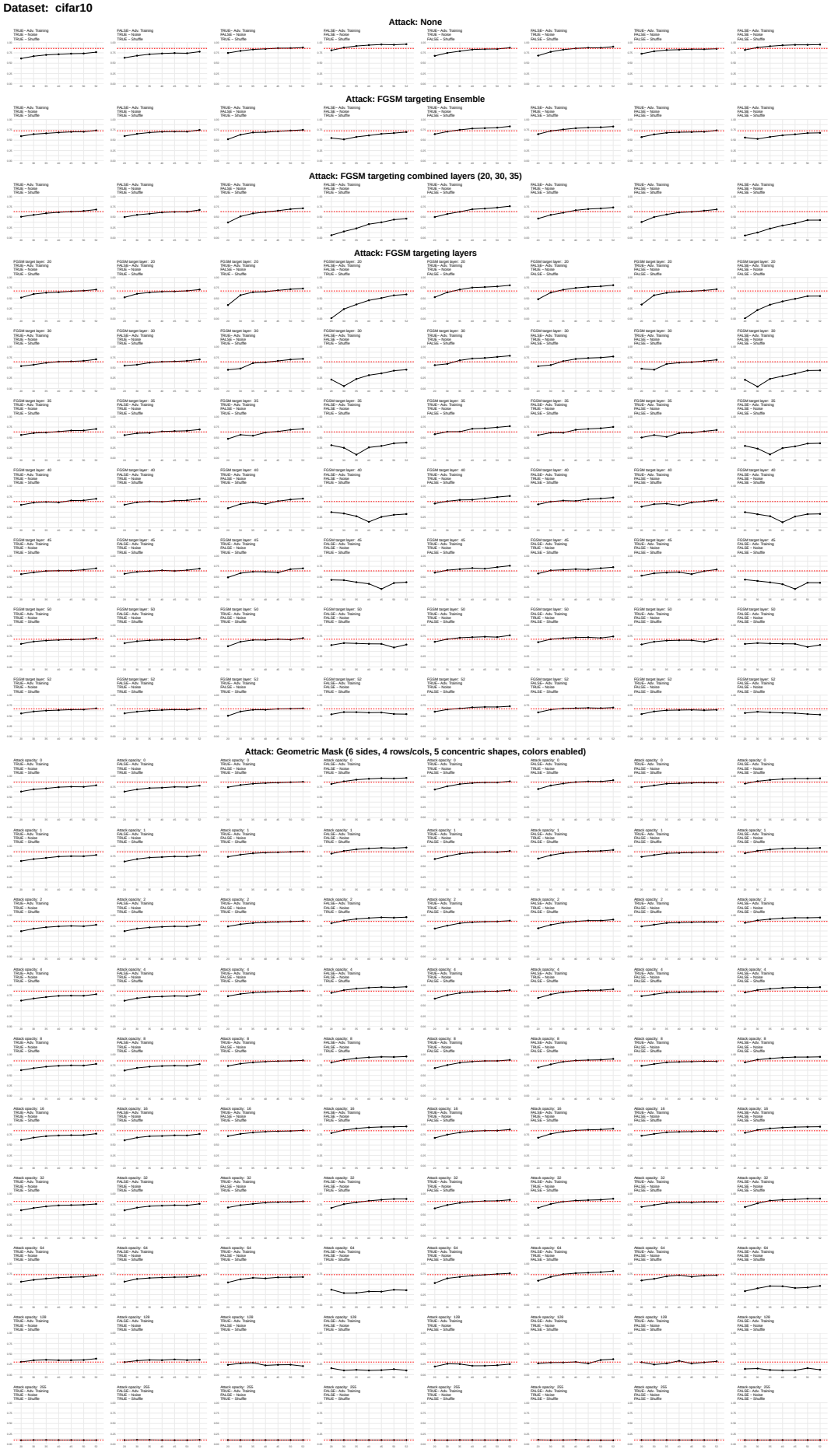


Figure 3.2: Training strategy combinations on CIFAR10, with the red dotted line for the ensemble. The X-axis shows layers, the Y-axis shows accuracy, the X-grid shows train configs and the Y-grid shows different attacks.

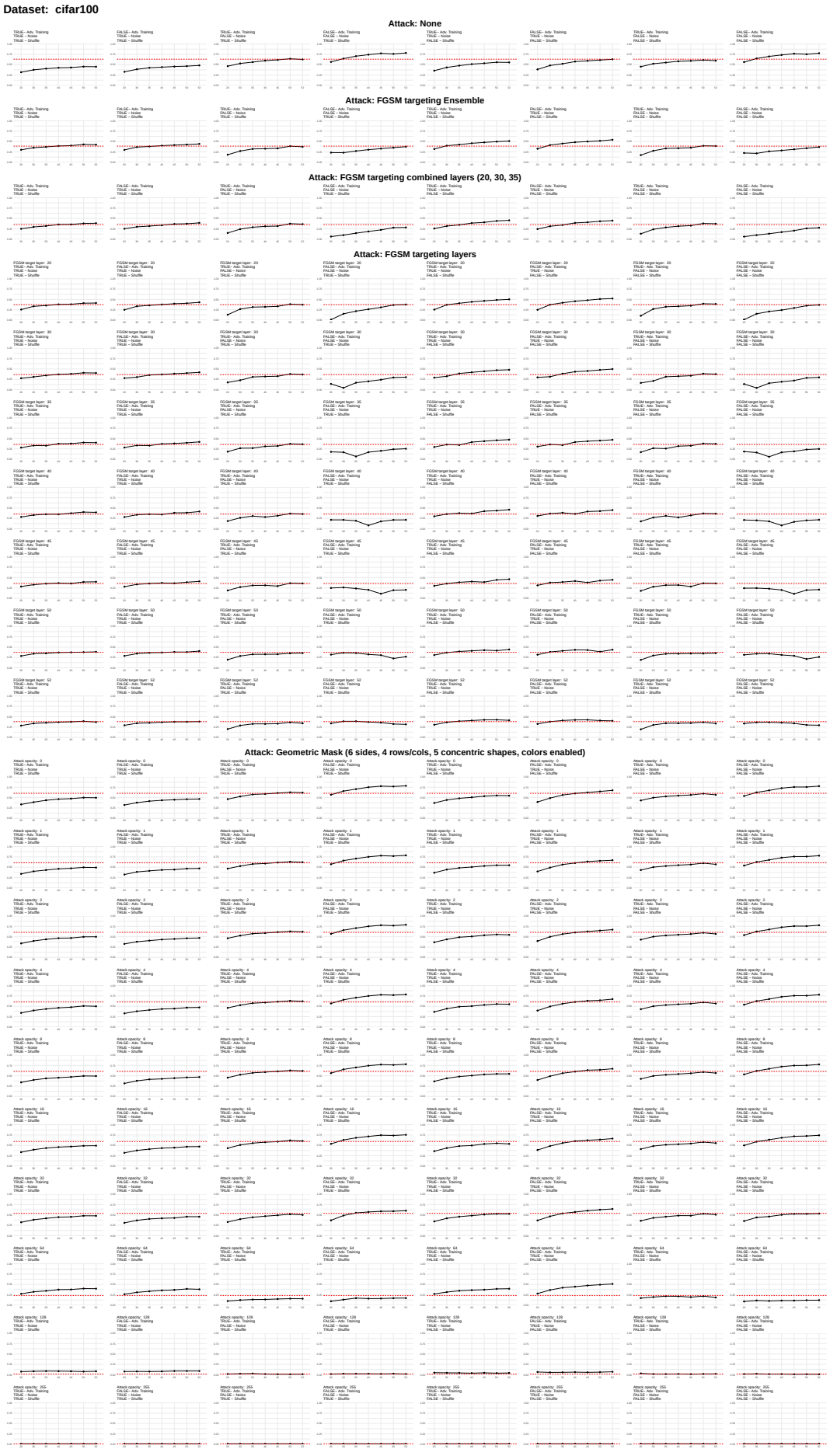
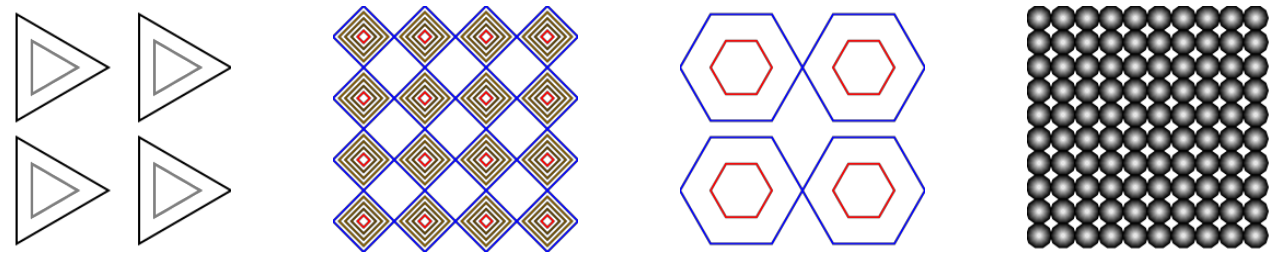


Figure 3.3: Training strategy combinations on CIFAR100, with the red dotted line for the ensemble. The X-axis shows layers, the Y-axis shows accuracy, the X-grid shows train configs and the Y-grid shows different attacks.



(a)Triangle mask (b)Square mask (c)Hexagon mask (d)Decagon mask

Figure 3.4: Selected samples of all geometric masks used in our experiments. Masks have varying sides, count of shapes per row/column, number of concentric shapes and colors.

overall. In every opacity, color, and side combination we tested, using colored masks (like those used by hCAPTCHA) outperformed monochrome masks by a wide margin. The most important parameters to vary in the remaining analyses were opacity, the number of shapes per row/column and whether the model was naturally trained. This leaves us with 3 masks to evaluate on, shown in Figure 3.6.

Backbone Evaluation. We attacked the backbone with the same 3 geometric masks.

The self-ensembler uses the default ResNet-50 implementation from `torchvision`, pretrained on ImageNet, as its backbone. To provide a reference point for the self-ensemble, we evaluated the backbone independently with the three masks to which the self-ensemble showed the highest sensitivity, applying the same range of opacities.

For each mask and opacity combination, we evaluated 9 variants of the backbone by varying the training conditions. Specifically, we used 3 training epochs (0, 2, 6) and 3 adversarial training fractions (0, 0.1, 0.2). Each variant was tuned and evaluated with the same opacity. The results of these experiments are presented in Figure 3.9. We believe that these adjustments, including adversarial training and tuning, create a fair comparison. This aligns with the self-ensemble, where tuning is required and linear probes for each intermediate layer are retrained from scratch on the dataset for each experiment.

3.3 Results

The experiments discussed in this section are illustrated in Figures 3.7, 3.8 and 3.9.

Natural Training. Looking broadly at the self-ensembled model's robustness against various attacks, we observe a consistent trend.

Natural training was not given much attention in the original paper we reproduced, even though our experiments consistently show that it plays a key role in achieving robustness. For example, while self-ensembling of the ResNet architecture is effective, it falls short when dealing with more sophisticated attacks beyond simple FGSM. When comparing naturally trained models (marked in black in the plots) to non-naturally trained ones (marked in gray), an interesting pattern emerges: initially, the non-naturally trained model underperforms on benign datasets by around 10%. However, as attack strength increases—whether through FGSM, PGD, or masks with opacity above 32 – we see the naturally trained model gain a substantial advantage. This is particularly striking in PGD attacks, where the accuracy of non-naturally trained models can drop close to zero when targeting the final or intermediate layers. Meanwhile, naturally trained models experience only minor accuracy reductions under the same conditions.

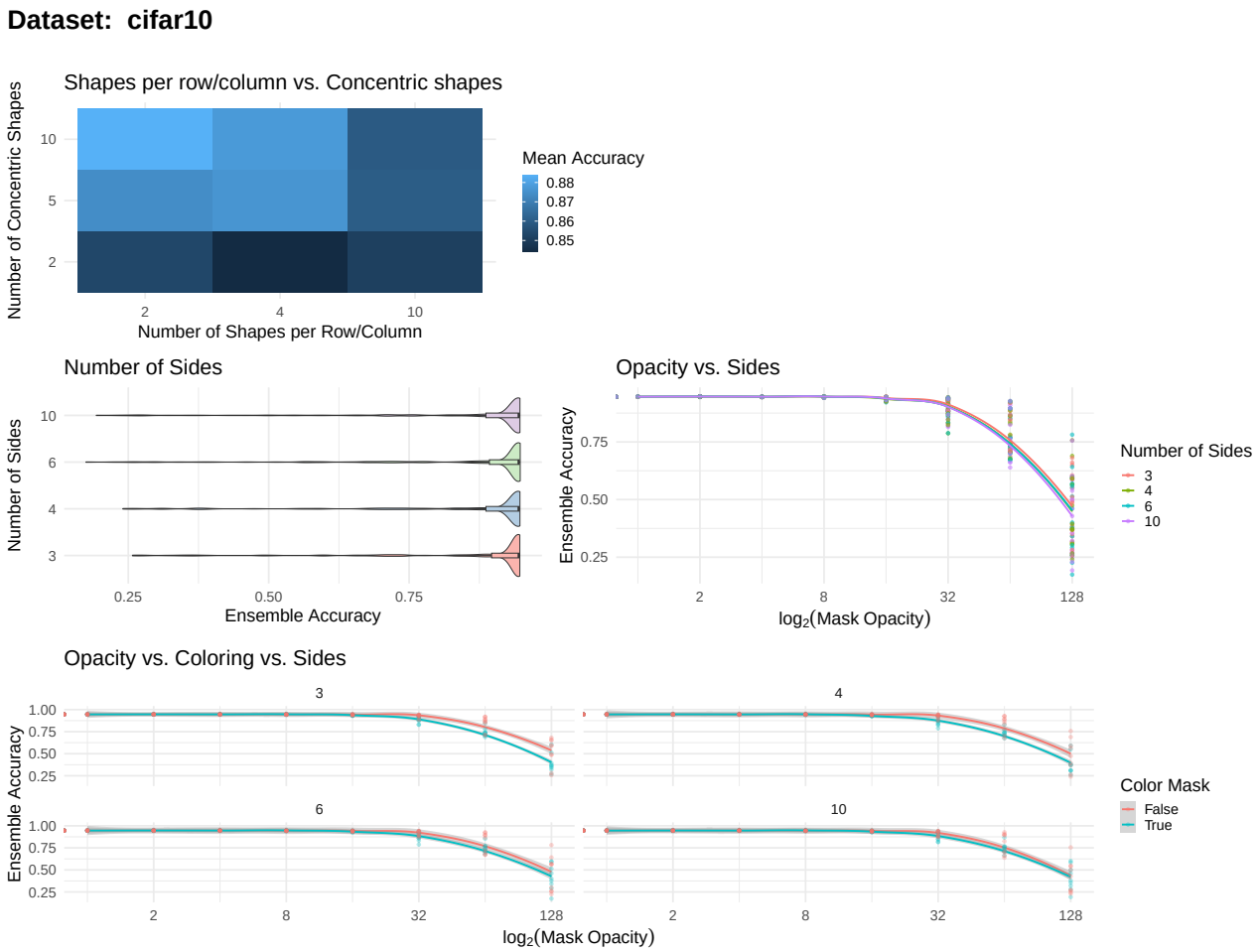
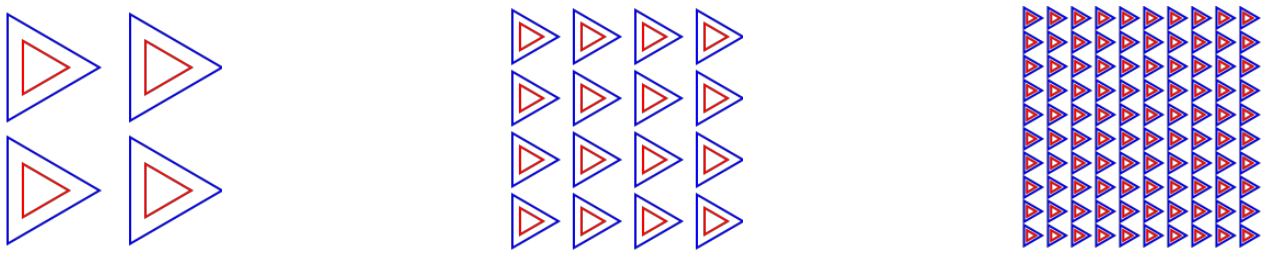


Figure 3.5: Limiting mask parameters to the most effective ones for CIFAR10. The lower most plot encodes the opacity in the x-axis, the better performing ensemble accuracy in the y-axis, the number of sides in the individual subplots and whether masks are colored or not as the color.



(a)Two shapes per row/column

(b)Four shapes per row/column

(c)Ten shapes per row/column

Figure 3.6: The 3 most effective geometric masks against the self-ensembled ResNet, based on preliminary analysis. All masks are colored triangles with 2 concentric shapes and a varying number of per row and column.

That said, this does not undermine the value of self-ensembling. In fact, it's highly effective in most scenarios, with the ensemble consistently outperforming the final layer. Only a few cases deviate from this trend. Attacks specifically targeting the ensemble itself are generally ineffective and combining natural training with self-ensembling produces exceptional results. When natural training is used, the ensemble's accuracy declines much more gradually as attack strength ramps up, compared to models lacking natural training.

However, we had hoped that this new architecture alone might achieve state-of-the-art robustness without requiring any form of adversarial training – a concept we refer to as "zero-cost robustness". This represents an exciting vision for the future of adversarial robustness research. Unfortunately, our findings suggest we're not there yet. While self-ensembling with ResNet is a promising step forward, it's not sufficient on its own to achieve state-of-the-art robustness without adversarial training.

FGSM vs. PGD. As expected, the PGD attack proves a lot stronger than FGSM when tested against the self-ensembled ResNet. Despite this, both attacks exhibit similar patterns in terms of which intermediary layers they impact and the overall shape of intermediary accuracy, as shown in Figure 3.7. In every example, the PGD attack amplifies the accuracy drop seen from FGSM, particularly for the non-naturally trained model. While FGSM reduces accuracy to around 25%, PGD pushes it nearly to zero. This disparity is most pronounced in the final layer, where PGD leads to a drop in accuracy exceeding 40%.

Geometric Masks. In this set of experiments, we did not rely on perceptibility proxy metrics. Instead, we manually inspected the masks to evaluate their perceptibility. We observed that the unrestricted geometric adversarial examples used as masks were not only clearly perceptible but also caused information loss. Specifically, with an opacity of 128, which left only the contours and rough edges of shapes visible. Nonetheless we could consistently and accurately identify the underlying class of the image in every case.

Among the variables involved in geometric mask attacks – such as the number of shape sides, number of shapes per row and column, number of concentric shapes, enabling or disabling colors and opacity – we identified the three most effective masks through exploratory analysis. We then fixed those variables as constants and focused our investigation on two variables: the number of shapes per row and column and opacity, as they had the most significant impact.

Interestingly, while varying the number of shapes per row and column yielded only marginal differences of 1-5% in accuracy (as shown in Figure 3.8), changes in opacity consistently led to noticeable drops in accuracy. Doubling the opacity always resulted in at least a 10% drop in accuracy within our tested range. Additionally, we observed a "flipping" effect between naturally trained and non-naturally

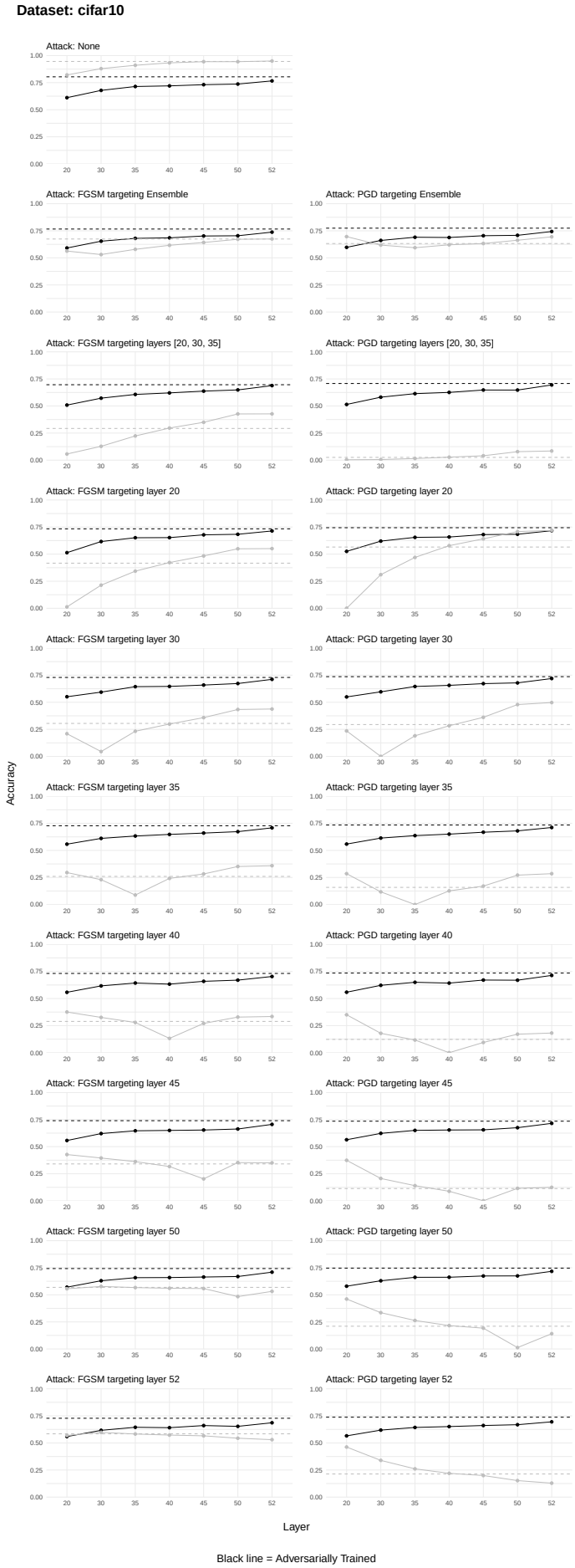


Figure 3.7: Comparing FGSM and PGD attacks on CIFAR10. The X-axis shows intermediary layers, the Y-axis shows accuracy, the black line indicates light natural training, dotted horizontal lines show ensemble performance.

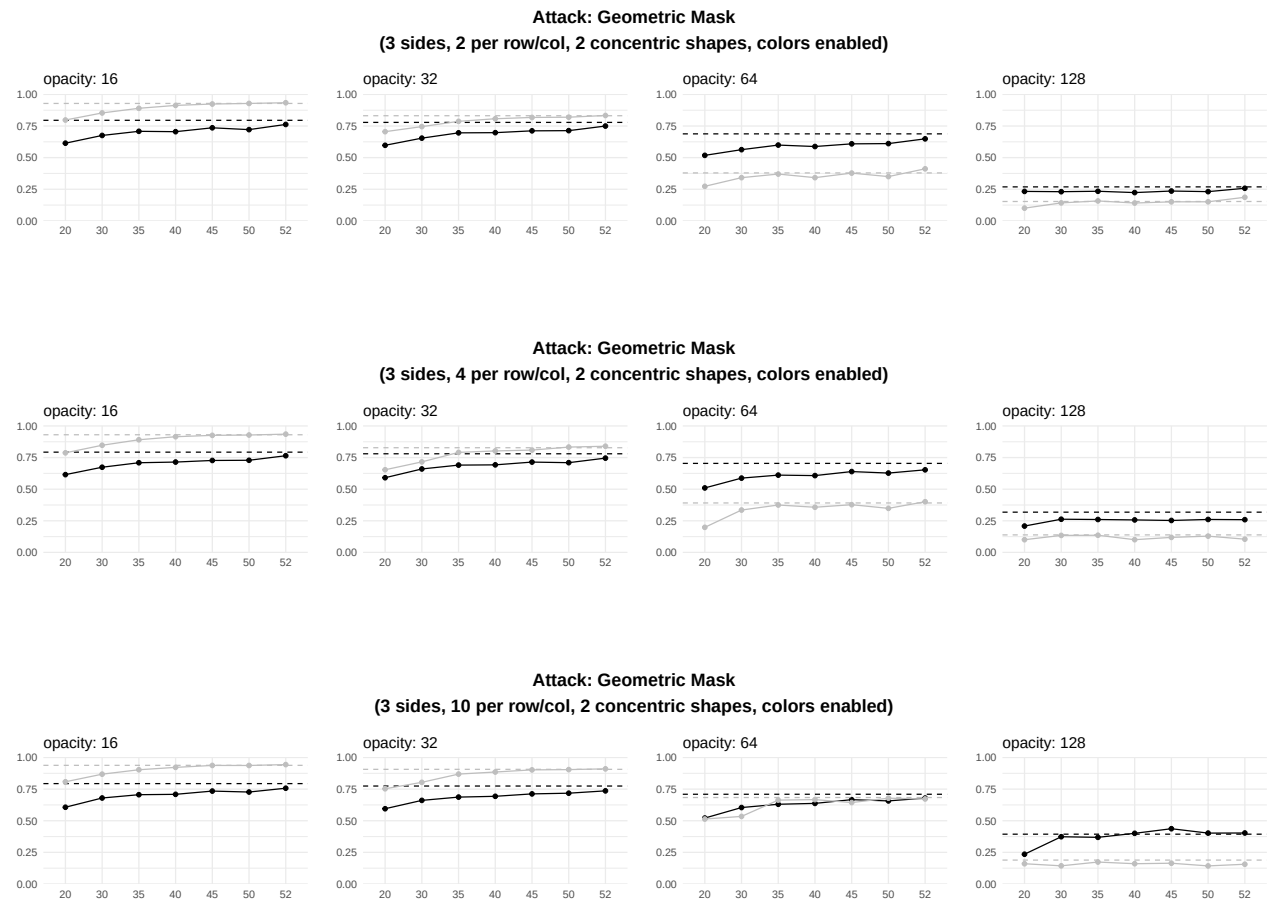


Figure 3.8: Evaluating the most effective masks on CIFAR10. The X-axis shows intermediary layers, the Y-axis shows accuracy, the black line indicates natural training, dotted horizontal lines show ensemble performance.

trained models, which consistently occurred at opacity levels between 32 and 64. This behavior stood out as the most intriguing finding in this set of experiments and aligns with our current understanding of the "robustness-accuracy trade-off", explained in the introduction.

Improvements over the Backbone. Similar to the large accuracy drop observed in the self-ensembled ResNet, especially from opacity 32, we find that simply tuning the backbone on the CIFAR10 dataset without incorporating adversarial examples also results in a noticeable accuracy decline. However, when the backbone is adversarially trained using the same masks it is later evaluated on, its accuracy significantly improves, outperforming the self-ensembled ResNet and reaching nearly 80% accuracy. This trend is illustrated in Figure 3.9.

The key insight here is that when adversaries gain access to the mask used for generating adversarial examples and train their model against it (along with the dataset), the effectiveness of the attack diminishes considerably. This highlights that a "robustified" backbone, trained with adversarial examples in its tuning set, is not directly comparable to the self-ensembled model in a fair manner.

The self-ensembling approach and naturally trained self-ensemble exhibit an additional advantage: robustness across all masks, regardless of their specific properties. In contrast, the robustified backbone shows robustness only to the masks it was specifically tuned with. This finding underscores the

versatility and potential of the self-ensembled ResNet.

However, in a specific case – using 3-sided shapes, 10 shapes per row/column, 2 concentric shapes, with colors enabled – we observe an anomaly among the high-opacity experiments. Here, even the naturally trained ensemble underperforms compared to the tuned but non-robustified backbone. While such exceptions exist, they were observed in less than 1% of all experiments and only under particularly strong attacks.

3.4 Conclusion

Our key takeaways can be summarized in two important concepts.

Limited Zero-Cost Robustness. One of the central aspirations of this research was to assess whether self-ensembling architectures could achieve state-of-the-art adversarial robustness without incurring the computational expense of adversarial training – a concept we refer to as "zero-cost robustness". While self-ensembling demonstrated clear potential, particularly in its ability to outperform non-ensembled layers in most scenarios, our results indicate that it falls short of achieving robustness on par with adversarially trained models.

Attacks such as PGD revealed critical limitations of the self-ensembled ResNet, particularly when natural training was absent. The accuracy of non-naturally trained models plummeted close to zero under stronger attacks, emphasizing the need for additional mechanisms beyond architectural improvements to achieve true zero-cost robustness. Although natural training improved the model's resilience massively (maintaining accuracy by over 50% for the highest tested opacity), it alone could not bridge the gap entirely. This suggests that while self-ensembling remains a promising approach, the vision of achieving robust, attack-resistant models without adversarial training is not yet a reality, underscoring the importance of continued research in this direction.

Accuracy Flipping through Natural Training. One of the most intriguing discoveries in our experiments was the "accuracy flipping" phenomenon observed under increasing attack strength. When comparing naturally trained and non-naturally trained models, we identified a consistent pattern: non-naturally trained models initially performed better on benign datasets but suffered dramatically under attack. Naturally trained models, on the other hand, demonstrated a striking resilience, with their accuracy surpassing that of non-naturally trained models as attack strength increased.

This behavior was especially pronounced when varying the opacity of geometric masks. A critical threshold emerged between opacity levels of 32 and 64, where naturally trained models not only maintained more consistent performance but also began outperforming their non-naturally trained counterparts. This "flipping" of accuracy highlights the nuanced interplay between robustness and accuracy within naturally trained models.

The phenomenon aligns with the broader "robustness-accuracy trade-off", suggesting that natural training manipulates a model's ability to generalize across both benign and adversarial examples. Moreover, it underscores the importance of taking natural training into account as a foundational component of robustness, especially when designing models aimed at withstanding unrestricted or perceptual adversarial attacks.

Outlook. In summary, our experiments highlight both the promise and the limitations of self-ensembling as a pathway toward robust machine learning. While natural training emerges as a surprisingly effective tool, particularly in counteracting stronger attacks like PGD and high-opacity geometric masks, achieving true zero-cost robustness remains an open challenge.

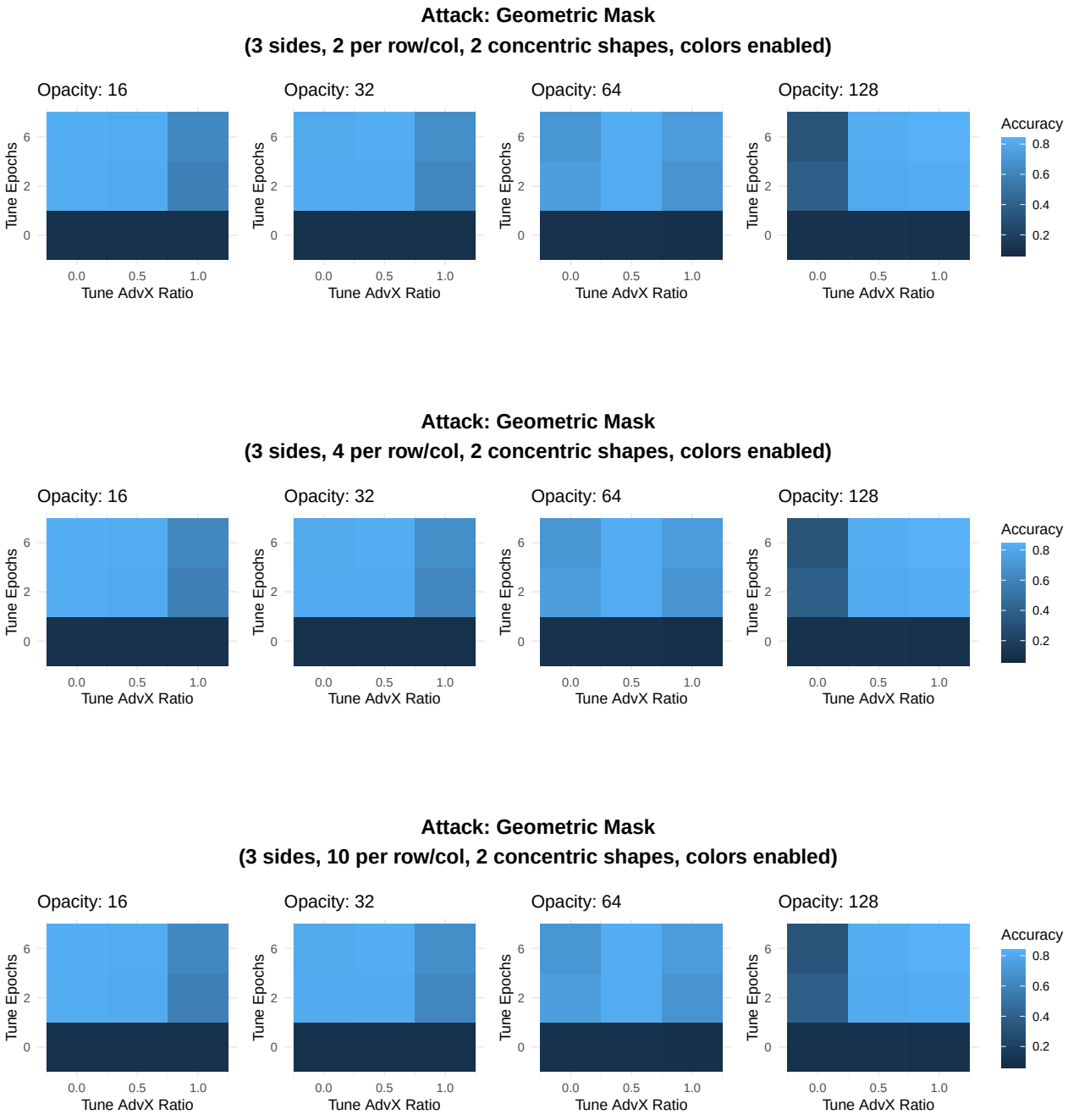


Figure 3.9: Evaluating the most effective masks on the Backbone. The X-axis shows the ratio of perturbed masks in the tuning set, the Y-axis shows the number of epochs the model was tuned with.

Future work could explore the use of MINE [168] and Rényi's α-order matrix-based functional [169] to estimate mutual information between self-ensemble layers and track the movement of latent representations across these intermediate layers. Ideally, these finer-grained metrics, combined with simple and effective black-box attacks, could lay the groundwork for foundational research in the intersection of machine learning interpretability and adversarial robustness.

Bibliography

[1] 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

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

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

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

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

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

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

[8] 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.

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

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

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

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

[13] 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.

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

[15] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson, "High-fidelity generative image compression," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] L. Yuan, W. Xiao, G. Dellaferrera, G. Kreiman, F. E. Tay, J. Feng, and M. S. Livingstone, "Fooling the primate brain with minimal, targeted image manipulation," *arXiv preprint arXiv:2011.05623*, 2020.

[22] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Neural Information Processing Systems (NeurIPS) Test of Time Award*, West Exhibition Hall C, B3, 2024, presented at NeurIPS 2024 Test of Time Award Session.

[23] M. Guerzhoy, "Understanding how neural networks see," University of Toronto, Lecture Notes, 2020. [Online]. Available: https://www.cs.toronto.edu/~guerzhoy/201s20/lec/W12/ann.pdf

[24] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[25] 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

[26] 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

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

[28] 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.

[29] 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.

[30] 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

[31] D. Fazlija, A. Orlov, J. Schrader, M.-M. Zühlke, M. Rohs, and D. Kudenko, "How real is real? a human evaluation framework for unrestricted adversarial examples," *arXiv preprint arXiv:2404.12653*, 2024.

[32] T. B. Brown and C. Olsson. (2018, 9) Introducing the unrestricted adversarial examples challenge. Google Brain Team.

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

[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] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 262–15 271.

[37] K. Zack. Archive of deleted tweet by @teenybiscuit. [Online]. Available: https://imgur.com/a/deep-learning-training-set-K4RWn

[38] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research," *arXiv preprint arXiv:1807.06732*, 2018.

[39] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," in *International conference on machine learning*. PMLR, 2019, pp. 1802–1811.

[40] Z. Qiu, W. Liu, H. Feng, Z. Liu, T. Z. Xiao, K. M. Collins, J. B. Tenenbaum, A. Weller, M. J. Black, and B. Schölkopf, "Can large language models understand symbolic graphics programs?" *arXiv preprint arXiv:2408.08313*, 2024.

[41] D. Keysers, N. Schärli, N. Scales, H. Buisman, D. Furrer, S. Kashubin, N. Momchev, D. Sinopalnikov, L. Stafiniak, T. Tihon *et al.*, "Measuring compositional generalization: A comprehensive method on realistic data," *arXiv preprint arXiv:1912.09713*, 2019.

[42] A. Ananthaswamy, "New theory suggests chatbots can understand text," *Quanta Magazine*, January 2024.

[43] S. Chahar, S. Gupta, I. Dhingra, and K. S. Kaswan, "Adversarial threats in machine learning: A critical analysis," in *2024 International Conference on Computational Intelligence and Computing Applications (ICCICA)*, vol. 1, 2024, pp. 253–258.

[44] H. Çifci, "Analysis of turkey's cybersecurity strategies: Historical developments, scope, content and objectives," *Sakarya University Journal of Science*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:268035521

[45] K. Sadeghi, A. Banerjee, and S. K. S. Gupta, "A system-driven taxonomy of attacks and defenses in adversarial machine learning," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 450–467, 2020.

[46] A. Khadka, S. Sthapit, G. Epiphaniou, and C. Maple, "Resilient machine learning in space systems: Pose estimation as a case study," *2022 IEEE Aerospace Conference (AERO)*, pp. 1–9, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251472990

[47] I. Yilmaz, K. Kapoor, A. Siraj, and M. Abouyoussef, "Privacy protection of grid users data with blockchain and adversarial machine learning," in *proceedings of the 2021 ACM workshop on secure and trustworthy cyber-physical systems*, 2021, pp. 33–38.

[48] G. Apruzzese, H. S. Anderson, S. Dambra, D. Freeman, F. Pierazzi, and K. Roundy, ""real attackers don't compute gradients": bridging the gap between adversarial ml research and practice," in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 2023, pp. 339–364.

[49] R. S. S. Kumar, J. Penney, B. Schneier, and K. Albert, "Legal risks of adversarial machine learning research," *arXiv preprint arXiv:2006.16179*, 2020.

[50] R. Cao and R. K.-W. Lee, "Hategan: Adversarial generative-based data augmentation for hate speech detection," in *International Conference on Computational Linguistics*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:227230383

[51] D. B. Nurseitov, K. A. Bostanbekov, G. Abdimanap, A. Abdallah, A. N. Alimova, and D. Kurmangaliyev, "Application of machine learning methods to detect and classify core images using gan and texture recognition," *ArXiv*, vol. abs/2204.14224, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:265674547

[52] M. Zolotukhin, D. Zhang, P. Miraghaie, T. Hämäläinen, W. Ke, and M. Dunderfelt, "Attacks against machine learning models in 5g networks," *2022 6th European Conference on Electrical Engineering & Computer Science (ELECS)*, pp. 106–114, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:259102662

[53] H. Face. (2024) Hugging face partners with wiz research to improve ai security. Hugging Face. Blog post discussing security improvements, pickle file security concerns, and partnership with Wiz Research. [Online]. Available: https://huggingface.co/blog/hugging-face-wiz-security-blog

[54] . Hours. (2024) Information security in high-impact areas career review. 80000 Hours. [Online]. Available: https://80000hours.org/career-reviews/information-security/

[55] Future of Life Institute, "Pause giant AI experiments: An open letter," Future of Life Institute, March 2023, open letter calling for pause in AI development.

[56] D. Hendrycks, N. Carlini, J. Schulman, and J. Steinhardt, "Unsolved problems in ml safety," *arXiv preprint arXiv:2109.13916*, 2021.

[57] . Hours. (2024) Preventing an ai-related catastrophe. 80000 Hours. [Online]. Available: https://80000hours.org/problem-profiles/artificial-intelligence/

[58] N. Carlini. (2019, 6) A complete list of all adversarial example papers. [Online]. Available: https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html

[59] I. Miller, "Computer vision on the battlefield: Can machines distinguish between enemy and civilian in military urban operations?" *Mechanical Engineering eJournal*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:237530499

[60] N. Moradpoor, L. A. Maglaras, E. Abah, and A. Robles-Durazno, "The threat of adversarial attacks against machine learning-based anomaly detection approach in a clean water treatment system," *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, pp. 453–460, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:262980912

[61] V. Chevardin, O. Yurchenko, O. V. Zaluzhnyi, and Y. Peleshok, "Analysis of adversarial attacks on the machine learning models of cyberprotection systems." *Communication, informatization and cybersecurity systems and technologies*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266487123

[62] D. A. Ulybyshev, I. Yilmaz, B. Northern, V. Kholodilo, and M. Rogers, "Trustworthy data analysis and sensor data protection in cyber-physical systems," *Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233384629

[63] B. Halak, C. Hall, S. Fathir, N. Kit, R. Raymonde, M. Gimson, A. Kida, and H. Vincent, "Towards autonomous physical security defenses using machine learning," *IEEE Access*, vol. PP, pp. 1–1, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248849785

[64] Rudolph, R. V. Matalucci, and J. T. Matalucci, "Developing protective strategies forcriticalbuilding infrastructures potentially subjected to m alevolentthreats* by," 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:111438592

[65] K. Gu, "Deep learning techniques in financial fraud detection," *Proceedings of the 7th International Conference on Cyber Security and Information Engineering*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253120915

[66] A. Agarwal and N. K. Ratha, "Black-box adversarial entry in finance through credit card fraud detection," in *CIKM Workshops*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:245540840

[67] M.-Y. Tsai, H.-H. Cho, C.-M. Yu, Y.-C. Chang, and H.-C. Chao, "Effective adversarial examples identification of credit card transactions," *IEEE Intelligent Systems*, vol. 39, pp. 50–59, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268628851

[68] V. Jogani, J. Purohit, I. Shivhare, and S. C. Shrawne, "Analysis of explainable artificial intelligence methods on medical image classification," *arXiv preprint arXiv:2212.10565*, 2022.

[69] M. H. Najafi, M. Morsali, M. Vahediahmar, and S. B. Shouraki, "Dft-based adversarial attack detection in mri brain imaging: Enhancing diagnostic accuracy in alzheimer's case studies," *arXiv preprint arXiv:2408.08489*, 2024.

[70] A. Rahman, M. S. Hossain, N. A. Alrajeh, and F. Alsolami, "Adversarial examples—security threats to covid-19 deep learning systems in medical iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9603–9610, 2021.

[71] N. Patel, P. Krishnamurthy, S. Garg, and F. Khorrami, "Adaptive adversarial videos on roadside billboards: Dynamically modifying trajectories of autonomous vehicles," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5916–5921, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:210971572

[72] X. Ji, Y. Cheng, Y. Zhang, K. Wang, C. Yan, W. Xu, and K. Fu, "Poltergeist: Acoustic adversarial machine learning against cameras and computer vision," *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 160–175, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235601506

[73] C. W. Axelrod, "Cybersecurity challenges of systems-of-systems for fully-autonomous road vehicles," *2017 13th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT)*, pp. 1–6, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:29935654

[74] Y. Yuan, G. Apruzzese, and M. Conti, "Multi-spacephish: Extending the evasion-space of adversarial attacks against phishing website detectors using machine learning," *Digital Threats: Research and Practice*, vol. 5, pp. 1 – 51, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266363431

[75] T.-N. To, D. L. Kim, D. T. T. Hien, N. H. Khoa, H. D. Hoang, P. T. Duy, and V.-H. Pham, "On the effectiveness of adversarial samples against ensemble learning-based windows pe malware detectors," *arXiv preprint arXiv:2309.13841*, 2023.

[76] 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.

[77] ProtectAI, "vulnhuntr," https://github.com/protectai/vulnhuntr, 2024, gitHub repository.

[78] Big Sleep Team. (2024, 10) From naptime to big sleep: Using large language models to catch vulnerabilities in real-world code. Google Project Zero. A collaboration between Google Project Zero and Google DeepMind.

[79] Coursera Editorial Team. (2024) What is adversarial machine learning? [Online]. Available: https://www.coursera.org/articles/adversarial-machine-learning

[80] Open Philanthropy, "Carnegie mellon university — research on adversarial examples," 2024, grant of $343,235 to support research on adversarial examples led by Professor Aditi Raghunathan.

[81] B. Eidson, "Mitre, microsoft, and 11 other organizations take on machine-learning threats," *MITRE News and Insights*, 2024, impact Story on the Adversarial Machine Learning Threat Matrix initiative. [Online]. Available: https://www.mitre.org/news-insights/impact-story/mitre-microsoft-and-11-other-organizations-take-machine-learning-threats

[82] A. Roy-Chowdhury, S. Krishnamurthy, C. Song, and S. Asif. (2020, 7) ECE and CSE faculty receive new DARPA grant on adversarial machine learning. University of California, Riverside. DARPA Machine Vision Disruption program grant announcement.

[83] Robust Intelligence. (2024) Ai application security. Robust Intelligence. [Online]. Available: https://www.robustintelligence.com/ai-application-security

[84] K. Cai, "Robust intelligence raises $14 million series a led by sequoia to build platform for testing machine learning applications," *Forbes*, October 2020.

[85] Booz Allen Hamilton. (2023, 9) Booz allen doubles down on adversarial ai capabilities with new investment. Business Wire. Press Release.

[86] MSSPAlert. (2023, September) Booz allen hamilton expands adversarial ai capabilities.

[87] 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

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

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

[90] 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

[91] 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.

[92] 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.

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

[94] 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.

[95] 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.

[96] 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.

[97] 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

[98] 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

[99] 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.

[100] 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.

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

[102] 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.

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

[104] 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.

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

[106] 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

[107] 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

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

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

[110] 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.

[111] 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.

[112] 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.

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

[114] 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.

[115] 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.

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

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

[118] 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.

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

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

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

[122] 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.

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

[124] 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.

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

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

[127] 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.

[128] 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.

[129] 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.

[130] 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.

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

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

[133] 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].

[134] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "RobustBench: a standardized adversarial robustness benchmark," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. [Online]. Available: https://openreview.net/forum?id=SSKZPJCt7B

[135] A. Araujo, L. Meunier, R. Pinot, and B. Negrevergne, "Advocating for multiple defense strategies against adversarial examples," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2020, pp. 165–177.

[136] C. Ren, X. Du, Y. Xu, Q. Song, Y. Liu, and R. Tan, "Vulnerability analysis, robustness verification, and mitigation strategy for machine learning-based power system stability assessment model under adversarial examples," *IEEE Transactions on Smart Grid*, vol. 13, pp. 1622–1632, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:245103286

[137] J. M. Adeke, G. Liu, J. Zhao, N. Wu, and H. M. Bashir, "Securing network traffic classification models against adversarial examples using derived variables," *Future Internet*, vol. 15, p. 405, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266389288

[138] Y. Feng and Y. Cai, "Towards robust classification with image quality assessment," *arXiv preprint arXiv:2004.06288*, 2020.

[139] Z. Rakhimberdina, X. Liu, and T. Murata, "Strengthening robustness under adversarial attacks using brain visual codes," *IEEE Access*, vol. 10, pp. 96 149–96 158, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252393135

[140] Y. Li, Q. Zhou, S. Li, and B. Li, "wadvmtd: A mitigation to white-box adversarial examples using heterogeneous models and moving target defense," *2023 3rd Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*, pp. 592–597, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259122131

[141] S. Kokalj-Filipovic, R. Miller, and G. M. Vanhoy, "Adversarial examples in rf deep learning: Detection and physical robustness," *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1–5, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:210971684

[142] J. Zhao, J. Wu, J. M. Adeke, G. Liu, and Y. wei Dai, "Eitgan: A transformation-based network for recovering adversarial examples," *Electronic Research Archive*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:264180609

[143] Q. Ji, L. Wang, C. Shi, S. Hu, Y. Chen, and L. Sun, "Benchmarking and analyzing robust point cloud recognition: Bag of tricks for defending adversarial examples," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4295–4304.

[144] T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell, "Toward transparent ai: A survey on interpreting the inner structures of deep neural networks," in *2023 ieee conference on secure and trustworthy machine learning (satml)*. IEEE, 2023, pp. 464–483.

[145] M. Xie, Y. Wang, and H. Huang, "Fermi-bose machine," *arXiv preprint arXiv:2404.13631*, 2024.

[146] S. Fort and B. Lakshminarayanan, "Ensemble everything everywhere: Multi-scale aggregation for adversarial robustness," *arXiv preprint arXiv:2408.05446*, 2024.

[147] AuthKong. (2024) Top 10 user-friendly recaptcha alternatives for 2024. AuthKong. [Online]. Available: https://authkong.com/blog/best-recaptcha-alternatives/

[148] A. Plesner, T. Vontobel, and R. Wattenhofer, "Breaking recaptchav2," in *48th IEEE International Conference on Computers, Software, and Applications (COMPSAC 2024)*. IEEE, 2024.

[149] 12189108 *et al.*, "url 'https://api.hcaptcha.com/getcaptcha/' returning base64 instead json," GitHub Repository: QIN2DIM/hcaptcha-challenger, February 2024.

[150] ForestCrazy, "hcaptcha-challenger," https://github.com/ForestCrazy/hcaptcha-challenger, 2024.

[151] BlackForestLabs, "Flux.1: A suite of text-to-image models," BlackForestLabs Website, 2024, accessed: December 09, 2024.

[152] X. Zhang, H. Hong, Y. Hong, P. Huang, B. Wang, Z. Ba, and K. Ren, "Text-crs: A generalized certified robustness framework against textual adversarial attacks," *arXiv preprint arXiv:2307.16630*, 2023.

[153] Y. Dong, H. Chen, J. Chen, Z. Fang, X. Yang, Y. Zhang, Y. Tian, H. Su, and J. Zhu, "How robust is google's bard to adversarial image attacks?" *arXiv preprint arXiv:2309.11751*, 2023.

[154] E. Shayegani, Y. Dong, and N. Abu-Ghazaleh, "Plug and pray: Exploiting off-the-shelf components of multi-modal models," *arXiv preprint arXiv:2307.14539*, 2023.

[155] C. Wang, R. Jia, X. Liu, and D. Song, "Benchmarking zero-shot robustness of multimodal foundation models: A pilot study," *arXiv preprint arXiv:2403.10499*, 2024.

[156] M. Goldblum, H. Souri, R. Ni, M. Shu, V. Prabhu, G. Somepalli, P. Chattopadhyay, M. Ibrahim, A. Bardes, J. Hoffman *et al.*, "Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[157] J. Howard. (2024) Which image models are best? [Online]. Available: https://www.kaggle.com/code/jhoward/which-image-models-are-best

[158] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 976–11 986.

[159] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 19 358–19 369.

[160] Y. Fang, Q. Sun, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva-02: A visual representation for neon genesis," *Image and Vision Computing*, vol. 149, p. 105171, 2024.

[161] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[162] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[163] A. Singhal *et al.*, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.

[164] O. S. Faragallah, H. El-Hoseny, W. El-Shafai, W. A. El-Rahman, H. S. El-Sayed, E.-S. M. El-Rabaie, F. E. A. El-Samie, and G. G. N. Geweid, "A comprehensive survey analysis for present solutions of medical image fusion and future directions," *IEEE Access*, vol. 9, pp. 11 358–11 371, 2021.

[165] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[166] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[167] C. Wang, R. Jia, X. Liu, and D. Song, "Benchmarking zero-shot robustness of multimodal foundation models: A pilot study," *arXiv preprint*, 2024.

[168] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 531–540. [Online]. Available: https://proceedings.mlr.press/v80/belghazi18a.html

[169] L. G. Sanchez Giraldo, M. Rao, and J. C. Principe, "Measures of entropy from data using infinitely divisible kernels," *IEEE Transactions on Information Theory*, vol. 61, no. 1, pp. 535–548, 2015.

[170] Association for Computing Machinery, "Acm policy on authorship: Frequently asked questions," https://www.acm.org/publications/policies/frequently-asked-questions, 2024, accessed: 2024-12-19.

Overview of Generative AI Tools Used

In this work language models were not used to generate original content. Instead, they solely enhanced spelling, grammar and style.

The ACM guidelines [170] suggest that using generative models for editing – such as improving grammar, clarity or engagement – does not require disclosure, likening it to traditional tools like Grammarly[3]. However, in the spirit of transparency, we disclose the specific models used and the prompt given to them for each paragraph.

The following models were used:

- JetBrains Grazie (version 1.7.3)
- Microsoft Github Copilot Pro (version 0.23)
- Antrophic Claude Sonnet 3.5, wrapped by Perplexity (version 2)
- OpenAI GPT-4o, wrapped by Perplexity (version omni)

In almost all cases, the following prompt was used, before minor adjustments were made:

```
rewrite.
do not alter the meaning of this text.

use full sentences.
do not use enumerations, itemizations, headings

use a simple, clear, to the point style.
do not use useless technical jargon.
do not use flowery language.
prefer transitive phrasal (noun comes inbetween the verb and the preposition).
do not use standard inversions.
do not use semicolons.
do not use serial commas / oxford commas.

do not use dancing metaphores.
do not use the words: AI, significant, delve, dive, deep, uncover, discover, explore, revoluti
```

Alternatively, for quick edits, the following prompt was used:

```
rewrite. use a casual but academic tone.
```

[3]Credits to Maximilian Kleinegger for this finding.