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 27, 2024

Acknowledgements

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

...

Keywords: Reliability, Robustness, Security, Algorithmic Models

Originality

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

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.

Papers

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

Open source software

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

Due to the exploratory nature of the work, many of the software built and experiments conducted 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 shell 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 can run a Jupyterlab instance on the intranet.
`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

Acknowledgements **i**

Abstract **ii**

1 Introduction **1**

1.1 Definition . . . 1

1.1.1 Perturbation Methods . . . 1

1.1.2 Imperceptible Adversarial Examples . . . 2

1.1.3 Semantics Preserving Adversarial Examples . . . 3

1.2 Motivation . . . 4

1.3 Threat Modeling . . . 6

1.4 Latent Representations . . . 6

1.5 Mental Models . . . 7

1.6 Defenses . . . 10

1.6.1 Train- and Test-time defenses . . . 10

1.6.2 Architectural Defenses . . . 11

1.7 Future Directions . . . 12

2 Results: HCaptcha Inspired Geometric Masks **13**

2.1 HCaptcha Inspired Geometric Masks . . . 13

2.1.1 Research Motivation . . . 13

2.1.2 Experimental Setup . . . 15

2.1.3 Results . . . 19

3 Results: Self-Ensembled ResNet **22**

3.0.1 Research Motivation . . . 22

3.0.2 Experimental Setup . . . 22

3.0.3 Results . . . 23

Bibliography **24**

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 [**?**] 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 [**?**]. 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 [**?**]. 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) [**?**, **?**, **?**, **?**, **?**, **?**]. SoS systematically perturbs input coordinates using powers of a symbolic infinitesimal. For a point $p_i = (x_i, y_i)$, the perturbed coordinates become:

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

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

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

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 [**?**], 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. [**?**] 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 [**?**, **?**]. 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 don't 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 [**?**].

However, the concept of "imperceptibility [to humans]" as originally proposed by Szegedy et al. [**?**] 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 [**?**, **?**].

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

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

Intuition: The Deep Learning Hypothesis

Ilya Sutskever, in his "Test of Time" award talk at NeurIPS 2024 [**?**], 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 [**?**].

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 [**?**].

This could explain why adversarial examples transfer to time-constrained humans [**?**]. 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 [**?**].

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

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" [**?**, **?**] or "semantics-preserving" [**?**, **?**, **?**]. The comparison in Fig. **??** and the illustration in Fig. **??** highlight the differences between various kinds of adversarial examples. Fig. **??** shows a collection of naturally occurring adversarial examples, also known as "natural adversarial examples" [**?**, **?**].

This shift in defining adversarial examples, popularized by the "Unrestricted Adversarial Examples Challenge" [**?**] 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 [**?**].

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

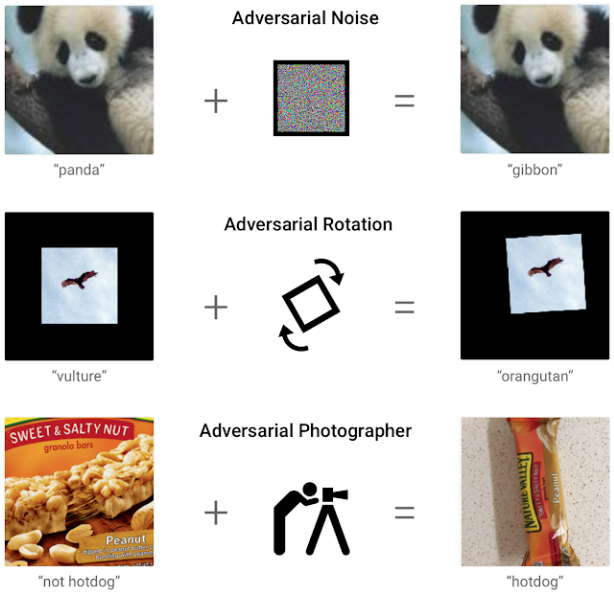


Figure 1.1: Unrestricted adversarial examples [?].

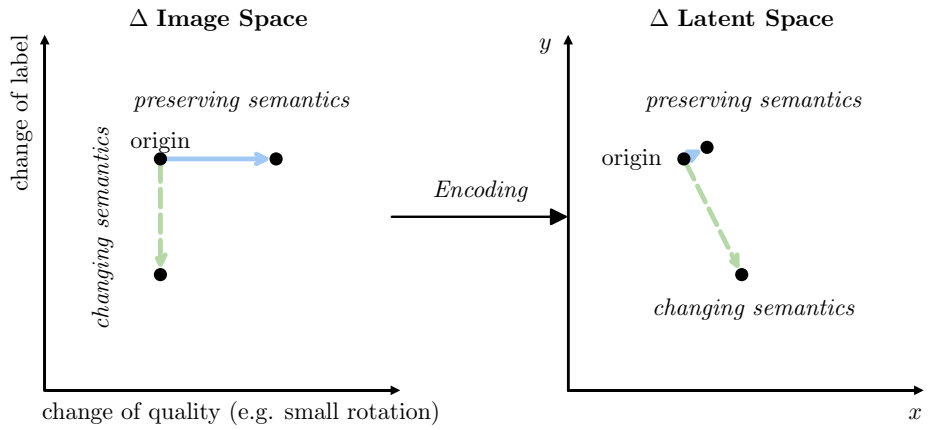


Figure 1.2: Semantics preserving/changing perturbations in pixel/latent-space (assuming full accuracy).

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 [?, ?] as proxies. This pragmatic approach acknowledges the limitations of current technology while striving for more meaningful adversarial examples.

1.2 Motivation

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

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

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

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

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 [?, ?]. 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 [?]. That is, despite a growing consensus that the risks posed by AI systems are significant and warrant urgent attention [?, ?] and a near-exponential surge in academic interest in adversarial machine learning since 2014 [?].

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 [?], critical national infrastructure [?, ?, ?, ?, ?], financial fraud detection [?, ?, ?], healthcare diagnostics [?, ?, ?], autonomous vehicles [?, ?, ?] and cybersecurity [?, ?, ?]. 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 [?].

Finally, tools such as the VulnHuntr [?] and Big Sleep [?] models have high potential to be misused for malicious purposes. The latter has been particularly successful in automatically detecting zero-day exploits in SQLite in late 2024 [?].

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 [?].

Open Philanthropy has provided $330,000 and $343,235 in funding to Carnegie Mellon University dedicated to AdvX research [?].

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 [?].

The Defense Advanced Research Projects Agency (DARPA) has granted nearly $1 million to the CV AdvX team at UC Riverside [?]. Booz Allen Hamilton, the largest provider of machine learning services for the Federal government, invested in HiddenLayer, Robust Intelligence [?, ?] Shift5, Credo, Hidden Level, Latent, Synthetaic and Reveal Technology [?, ?], 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 [?]. 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 [?]. Interestingly, research has shown that in some cases, black-box attacks can be more effective than white-box approaches at compromising model security [?].

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 [?, ?]. 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 [?].

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

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

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 [?]) 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 [?]. 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 [?]. 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 [**?**, **?**]. 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 [**?**].

Recent research [**?**] 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 [**?**]. 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 [**?**].

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

Recent work by Moosavi-Dezfooli et al. [**?**] 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 [**?**]. Among them are:

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

Initially, when Szegedy et al. [**?**] 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

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

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. [**?**] 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 [**?**, **?**, **?**, **?**, **?**].

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

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 [**?**]:

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

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 [**?**].

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 [**?**].
>
> 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 [**?**].
>
> The challenge of quantifying realism remains a fundamental problem in machine learning [**?**].

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 [**?**].

Non-robust features & concentration of measure in high dimensions. Most popularly, Ilyas et al. [**?**] 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 [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**].

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

Insufficient data. Schmidt et al. argue [**?**] 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. [**?**]. 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. [**?**, **?**] 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 [**?**, **?**, **?**, **?**]. 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. [?, ?] 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 [?].

Piecewise-linear decision boundaries. In the "dimpled manifold hypothesis" [?] the central claim is that adversarial examples emerge because we attempt to fit high $n-1$ dimensional decision boundaries to inherently low-dimensional data like images (which can be losslessly projected to $k \ll n$ dimensions). This leaves redundant dimensions on which adversarial examples won't be judged, which enables them exist roughly perpendicularly from the true location of the low-dimensional natural image, by using large gradients. 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 [?]. He additionally demonstrated that dimensionality reduction increases the interpretability of the perturbations to humans [?].

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 [?, ?]. 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 [?], 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 [**?**, **?**]. 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 [**?**].

Quality Assessment Integration. Implementing image quality assessment combined with knowledge distillation helps detect potentially harmful inputs that could cause incorrect model predictions [**?**]. Another alternative preprocessing technique is using brain-inspired encoders [**?**]. This method is particularly effective as it doesn't 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 [**?**].

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 [**?**].

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 [**?**].

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 [**?**].

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 [**?**]. 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 [**?**]. 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 [**?**]. 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 [**?**]. The key innovation lies in controlling the geometric separation of prototype manifolds through the target distance parameter, as revealed by statistical mechanics analysis [**?**].

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 [**?**]. The researchers introduced a robust aggregation mechanism called CrossMax, based on Vickrey auction, which combines predictions from different layers of the network [**?**].

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 [**?**]. 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 [**?**].

An interesting secondary outcome of this research was the discovery that gradient-based attacks against their model produced human-interpretable images of target classes [**?**]. 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 [**?**].

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

Results: HCaptcha Inspired Geometric Masks

> *"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 [**?**]

When studying unrestricted 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 can't 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 masks to overlay on images, avoiding complex spatial transformations or advanced attack techniques and (2) leveraging intermediary layers of neural networks to enhance interpretability.

2.1 HCaptcha Inspired Geometric Masks

2.1.1 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.

The biggest advantage of using CAPTCHAs for adversarial research is their proven effectiveness. By studying the challenges of top providers like Google's reCAPTCHA and hCaptcha, we can start off with a strong baseline – with a proven track record of robustness – to build upon. This saves us from having to reinvent

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

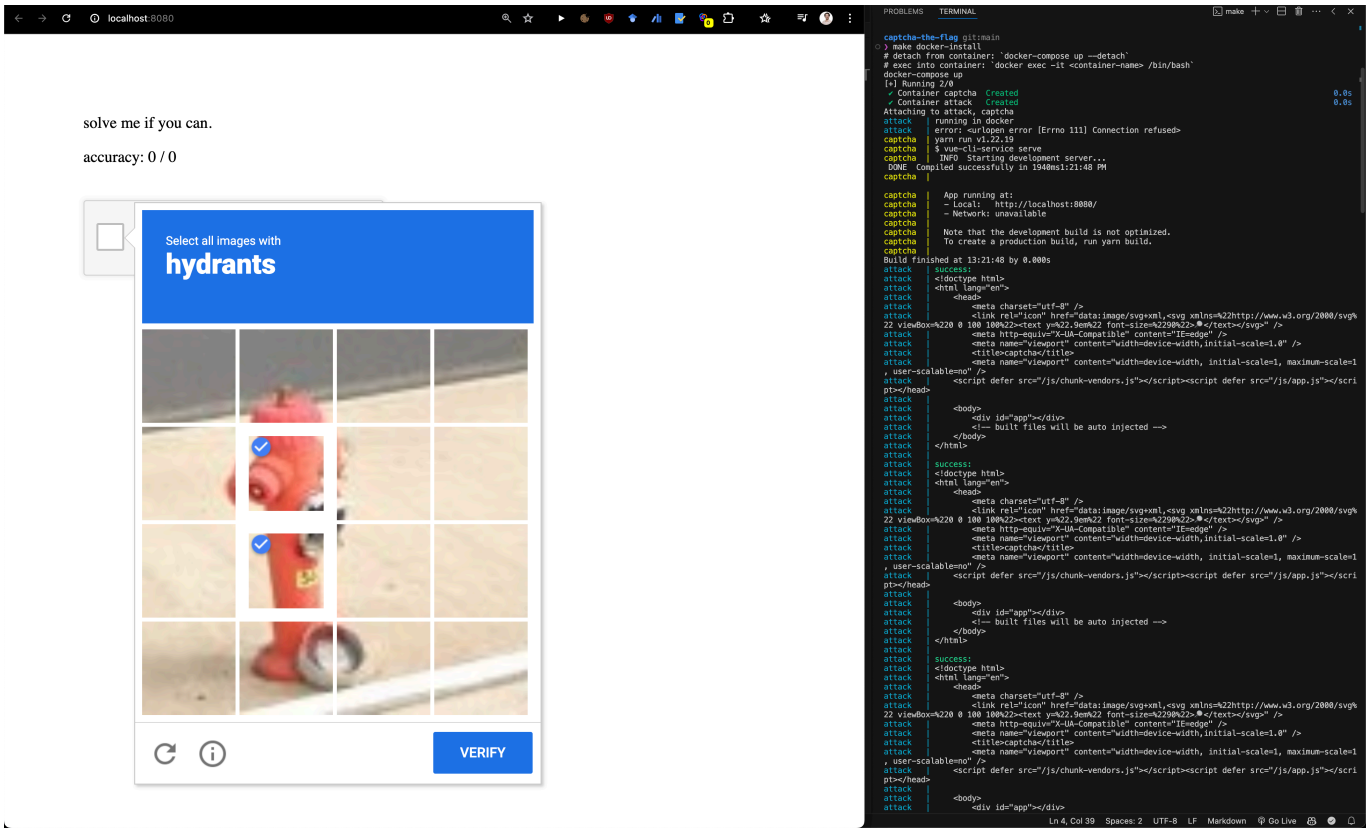


Figure 2.1: ReCAPTCHAv2 cybersecurity emulation framework
`https://github.com/ETH-DISCO/captcha-the-flag`

the wheel. Additionally a CAPTCHA challenge is a well-defined problem with clear success criteria and would enable future studies to conduct adversarial research and their transferability to humans on a large scale with significantly fewer ethical, legal and financial constraints.

A demo, shown in Figure **??**, 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% [**?**], they have been shown to be solvable with 100% accuracy using publicly available computer vision models on consumer hardware, by Plesner et al. [**?**].

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 [**?**, **?**]. 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 overlaying images with more complex patterns, such as a grid of colored geometric shapes with varying sizes and rotations. These challenges are designed to be easily solvable by humans, but difficult for machines to classify accurately.

In this scenario, the solver / identity provider acts as the adversary, while the CAPTCHA provider is the

Figure 2.2: Evaluation on synthetic data.

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 adversarial training and evaluation of (natural unrestricted) adversarial examples. Although promising, it didn't 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 **??**. This is an exciting direction, especially with the advancements brought by FLUX.1 [**?**], 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)
```

2.1.2 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).

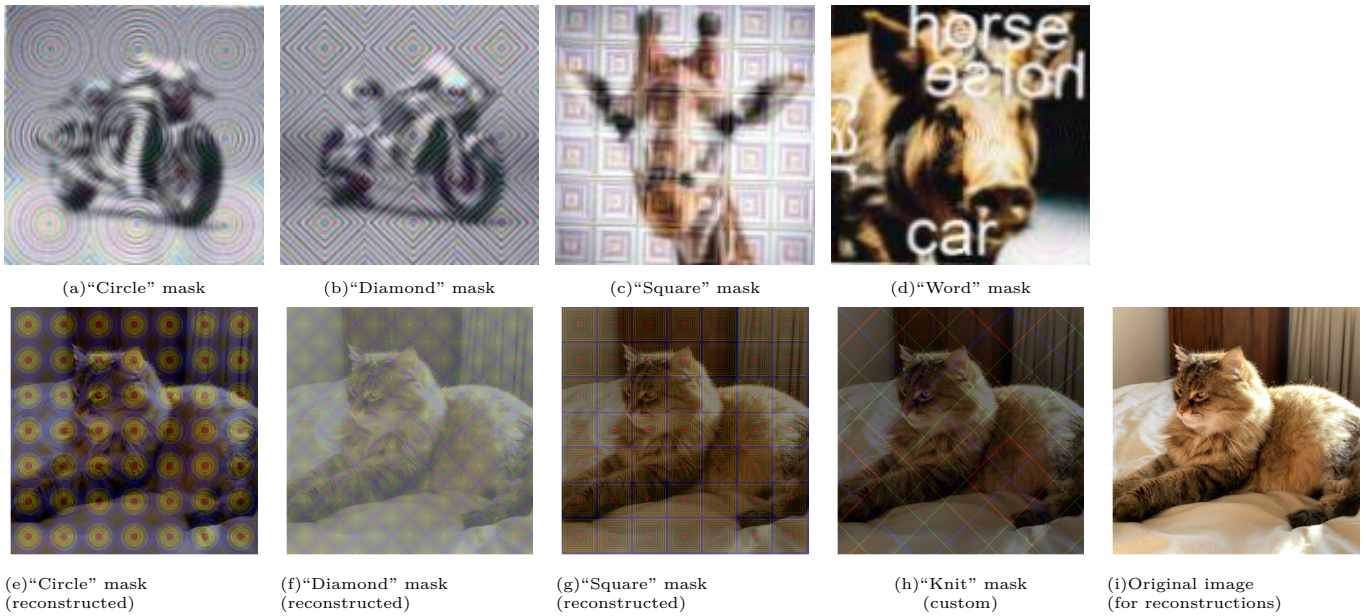
We also studied and reconstructed the segmentation challenge, which embeds images within a Perlin-noise-like pattern, smooths the edges, adds a slight blur and sometimes incorporates the geometric masks from the classification challenge on either individual segments or the entire image. However, given 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 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 [**?**, **?**, **?**]).

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 **??** 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 **??** 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.

Model Selection. 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 [**?**, **?**] 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 [**?**]. 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" [**?**], Open CLIP's "EVA01-g-14-plus" [**?**] and "EVA02-L-14" [**?**], "DFN5B-CLIP-ViT-H" by Apple [**?**], the original "ViT-L-14-378" and "ViT-H-14-378-quickgelu" [**?**], "ResNet50x64" [**?**] and RoBERTa-B and RoBERTa-L [**?**]. We highlight results for a

[2]Credits to Turlan Kuzhagaliyev.