ATLAS-AL: ADAPTIVE TRUST-REGION FOR LATENT ADVERSARIAL SEARCHES VIA ACTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Security evaluation of learning-based systems requires more than just testing the system against a fixed collection of attacks. It requires adaptive mechanisms that can efficiently discover *sets* of inputs that induce model failure. We introduce ATLAS (Adaptive Trust-Regions for Latent Adversarial Searches), which is a query-based framework that discovers adversarial input sets for black-box learning systems. ATLAS casts attack generation as an active learning level set estimation problem then combines calibrated approximations with a local-global sampling architecture to find regions of the input space that contain adversarial examples. Once discovered, ATLAS is designed to sample points within these adversarial regions to build adversarial sets that accurately represent the state of robustness of the target model. When applied on toy experiments, we find that ATLAS is able to recover more of the adversarial region under a limited query budget than does previous work. When applied to standard and adversarially trained MNIST, CIFAR, and ImageNet model targets, ATLAS produces better representative attacks than other query-based black-box attacks (NES, SignHunter, BayesOpt). ATLAS represents an automated red-teaming framework that can be used for both analyzing the robustness of learning-based systems under development and continuous auditing to see how the robustness of a system changes over time.

1 INTRODUCTION

As more learning-based systems get deployed in safety critical settings (Eykholt et al., 2018; Chou et al., 2021), robust evaluation requires more than testing against a fixed collection of previously known attacks(Athalye et al., 2018; Croce and Hein, 2020). An evaluator must be able to adaptively discover diverse inputs that induce failure, especially for complex targets whose failure mechanisms are unknown (Nasr et al., 2025; Bahati et al., 2021). Existing black-box adversarial attacks are designed to discover one/few successful failures. However, in our problem, our goal is to identify a set of adversarial inputs that produces failures for a specified target system. Rather than optimizing for a single successful adversarial example, this formulation seeks to produce a wide-spread adversarial set that can be used to audit where and how a learning-based system fails (Figure 1). The challenge comes from the input space being high-dimensional, making exhaustive search infeasible, and potentially having input adversarial regions occupy only a small fraction of the overall search space, making unguided exploration unlikely to discover them.

Therefore, we introduce ATLAS, an active-learning framework for generating adversarial sets in high-dimensional spaces. One of the key innovations is that it *re-formulates adversarial discovery as a region-discovery problem and combines tools from level-set estimation, Bayesian optimization, and adversarial generation to build these sets*. Because existing approaches (Ngo et al., 2026; Eriksson et al., 2019; Deglurkar et al., 2026; Paccagnan et al., 2023) are not enough to solve this problem - these approaches aren't designed to generate diverse adversarial sets - ATLAS builds upon previous work and uses calibrated model predictions combined with a dual sampling mechanism to build both an adversarial classifier and the associated adversarial dataset. To evaluate ATLAS, we first use non-convex benchmark functions helps us analyze ATLAS' ability to recover sparse and geometrically complex adversarial sets in toy problems. After this, we apply ATLAS on several image classification models. Since existing automated attack frameworks primarily uses optimization-based approaches, only finding single points of failure at a time, we investigate whether or not the repeated application of existing attack-generation methods is enough to recover diverse adversarial

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

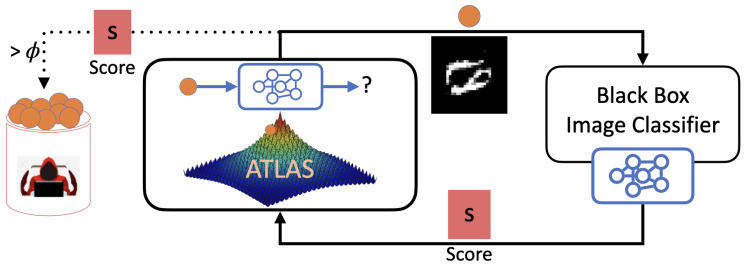


Figure 1: **ATLAS**. Given a target (black box image classifier), ATLAS searches for adversarial images (orange dot) by learning a mapping between inputs (orange dot) and scores that represent the level of adversity of those images for the target model. Using this function, ATLAS produces a set of attacks (the red bucket) for the target model.

sets or whether a method tailored to adversarial set construction would provide some advantage. When compared against popular optimization-based methods, we find that not only is ATLAS able to recover sets with more empirical coverage of the adversarial region, but it is also able to curate attacks with a success rate that is on par or better than the baselines.

Our contributions are: (1) framing adversarial generation as an active level-set estimation problem rather than an optimization problem, and showing that optimization-based attacks recover less of the adversarial set even when all of their queries are retained; (2) calibrating the surrogate's predictions with a residual model trained over the entire input space, in contrast to prior work that calibrates the sample-selection score; (3) designing a saturation trust region and acquisition that recover the interior of the adversarial set rather than only its boundary; and (4) evaluating on non-convex benchmarks with known ground-truth level sets up to 100D, and on standard and adversarially trained MNIST, CIFAR-10, and ImageNet classifiers, with ablations isolating (2) and (3).

2 Related Works

There are certain categories of methods that generate image attacks: (a) Methods that use gradient-based approaches to craft adversarial inputs (Goodfellow et al., 2014; Eykholt et al., 2018; Papernot et al., 2017; Kurakin et al., 2018; Carlini and Wagner, 2017; Ilyas et al., 2019; Athalye et al., 2018; Kurakin et al., 2016; Croce and Hein, 2020), (b) methods that use generative models to generate adversarial images (Xiao et al., 2018; Jandial et al., 2019; Song et al., 2018), (Poursaeed et al., 2018; Xu and Zhai, 2023), and (c) query-based methods that finds attacks to black box systems. In the latter, there exist Bayesian Optimization based approaches (Shukla et al., 2019; Cheng et al., 2024; Zhang et al., 2024; Ru et al., 2019) and generic querying methods like SquareAttack (Andriushchenko et al., 2020), SignHunter (Al-Dujaili and O'Reilly, 2020), and NES (Ilyas et al., 2018). Most notably, BayesOpt (Ru et al., 2019) is a well known method that is most similar to ours. In this work, they aim to find L_∞ based perturbation using similar adaptive sampling mechanisms as we do; however, they get around the curse of dimensionality by decomposing the input space into lower dimensional subspaces and assigning a local model to represent the adversarial objective in that lower-dimensional space. This is complementary to our method. In ATLAS, our input space is decomposed into local regions of the same dimension, while BayesOpt decomposes its search space into manifolds of lower dimensions. Furthermore, their method is designed to optimize an adversarial objective to find one specific attack. Our framework is designed to learn an adversarial mapping globally; thereby learning entire sets of adversarial examples that exists for a target system. Because of the differing aims, our adaptive sampling mechanisms are geared towards minimizing model error vs. generic adversarial maximization. Other approaches leverage LM-evolutionary methods (Nasr et al., 2025), (Zhan et al., 2025), and generic try-and-check methods (Gibson et al., 2024) as well.

3 Threat Model and Problem Formulation

Let $\mathcal{T} : \mathcal{X} \to \Omega$ represent a black box image classification system that maps from a bounded image space $\mathcal{X}$ to a probability distribution representing its output classification Ω. Let $\mathcal{O} : \Omega \to \iota$ represent our given adversarial objective that scores the output of the target system with respect to some adversarial criteria and let a successful attack be characterized by $\iota > \phi$. Also, assume that the

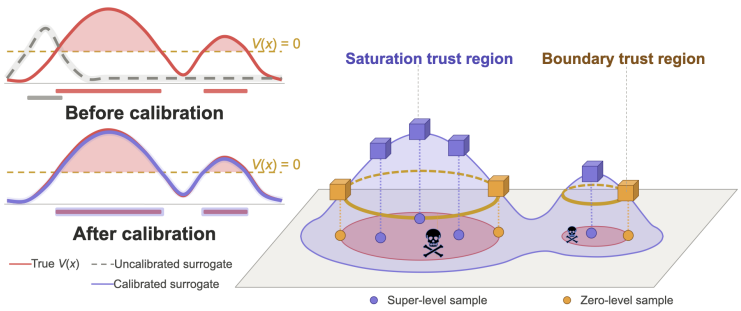


Figure 2: ATLAS performs level set estimation to find regions containing adversarial examples. It queries the target system using calibrated model predictions (purple surface) and a dual sampling strategy. The first sample points at the zero level set (gold boxes) to find the edges of adversarial regions, while the other sample within the super level set (purple boxes) to collect as many adversarial examples as possible. ATLAS produces both a model that correctly estimates which inputs are adversarial and an adversarial set that provides good coverage of target's adversarial region.

attack framework has limited query access to the target system $\mathcal{T}$ and can observe output probabilities. We are interested in producing a set of adversarial inputs, $\mathcal{A} \coloneqq \{x \in \mathcal{X} : f_{\text{adv}}(x) \geq \phi\}$, that represents a set of images that can successfully attack the system f_{adv}, where $f_{\text{adv}}(x) = (\mathcal{T} \circ \mathcal{O})(x)$. To generate this adversarial set, we formulate a level-set estimation active learning problem, where we aim to learn an adversarial value function $\mathcal{V}(x) \to \iota$ that maps candidate inputs x to adversarial scores $\iota \in \mathbf{R}$, whose super-zero level set $\mathcal{S} \coloneqq \{x : \mathcal{V}(x) \geq 0\}$ is the adversarial set we aim to recover. Previous works Ngo et al. (2026); Deglurkar et al. (2026) solves this problem by sampling near the zero level set. While sufficient for classification (are points safe or adversarial), this leaves the interior of $\mathcal{S}$ under-sampled which is unfavorable since an audit also requires a representative sample of $\mathcal{S}$ itself. However, trying to sample the interior of $\mathcal{S}$ is not trivial. Interior candidates can only be chosen from the surrogate's own prediction of where $\mathcal{S}$ lies. This prediction is often wrong and causes queries to land outside of $\mathcal{S}$ or in already highly sampled regions the model knows well. While prior work (Deglurkar et al., 2026) attempt to calibrate where samples are selected using calibrated acquisition functions, these functions combine prediction and uncertainty on an arbitrary scale, which makes it both harder to calibrate and redundant to do so: the calibration must be repeated for every acquisition function, and none of it corrects the underlying model. Our key idea is therefore to calibrate the model's prediction directly, using a residual model trained on the main model's error over the entire input space. A single correction then propagates to every acquisition built on that model. This lets us define two complementary selection rankings over a shared calibrated surrogate: one preferring candidates near the zero level set (*boundary*), which improves classification of the adversarial region, and one preferring candidates predicted to lie inside $\mathcal{S}$ (*saturation*), which curates the adversarial set.

4 ATLAS: Adaptive Trust-Region for Latent Adversarial Search

ATLAS (Fig. 2) constructs both an adversarial set $\mathcal{A}$ and an adversarial value function $\mathcal{V}(x) \to \iota$ using two components: (i) a calibration scheme for the trained value function and (ii) a local placement-sampling scheme that samples from two types of local regions. The first type (boundary regions) focuses on sampling from the zero level set for discovery, and the other (saturation regions) samples within discovered super level sets to produce $\mathcal{A}$. We refer to these local regions as 'trust regions'. A single algorithm, Algorithm 1, ties these two together: (1) fit global model, (2) fit residual model and calibrate, (3) select trust regions, (4) sample locally, and (5) refit with new local samples and repeat.

4.1 Calibration

Calibration helps turn inaccurate model estimates to accurate model estimates, helping us to efficiently leverage our sampling budget. To successfully calibrate our model, we can take a hold out set from our budget, train a second model that estimates the main model's error per sample (its residual), and apply the estimate error as a residual offset on our main model's prediction. How-

Algorithm 1: ATLAS: Adaptive Trust-Region for Latent Adversarial Search

Input : f_{adv} (oracle);
N_{bound}, N_{sat} (number of boundary / saturation regions);
b_L (queries per region per round);
b_G (total query budget)

Initialize: $\mathcal{C}_{global} \leftarrow \{(x_i, f_{\text{adv}}(x_i))\}_{i=1}^{N_{\text{cert}}},\ x_i \sim \mathcal{U}(\mathcal{X})$; (held out; for residual models only)
$\mathcal{D}_{total} = \emptyset$; $t = 0$

while $|\mathcal{D}_{total}| + |\mathcal{C}_{global}| < b_G$ **do**
  ▷ 1: Fit the global surrogate
  $h_g \leftarrow L(\mathcal{D}_{total})$ // Models $\mathcal{V}$
  ▷ 2: Fit a residual model and calibration global surrogate
  $h_r \leftarrow L\Big(\{(x_i,\, h_g(x_i) - f_{\text{adv}}(x_i)) \mid x_i \in \mathcal{C}_{global}\}\Big)$
  $\tilde{h}_g \leftarrow h_g - h_r$
  ▷ 3: Choose boundary and saturation regions centers
  for $k \leftarrow 1$ **to** N_{bound} **do**
    $\zeta_k \leftarrow \arg\max_{x \in \mathcal{X}} \mathsf{a}_{bound}(x, \tilde{h}_g) \ + \ \hat{\alpha} \prod_{\zeta_j \in \mathcal{C}_B^t} g\big(\|x - \zeta_j\|_2^2\big)$
    $\mathcal{C}_B^t \leftarrow \mathcal{C}_B^t \cup \{\zeta_k\}$
  for $k \leftarrow 1$ **to** N_{sat} **do**
    $\zeta_k \leftarrow \arg\max_{x \in \mathcal{X}} \mathsf{a}_{sat}(x, \tilde{h}_g) \ + \ \hat{\alpha} \prod_{\zeta_j \in \mathcal{C}_{sat}^t} g\big(\|x - \zeta_j\|_2^2\big)$
    $\mathcal{C}_{sat}^t \leftarrow \mathcal{C}_{sat}^t \cup \{\zeta_k\}$
  ▷ 4: Sample locally in every region
  $\mathcal{Q} \leftarrow \texttt{LocalSample}(\mathcal{C}_B^t, \hat{a}_B, \mathcal{D}_{total}) \bigcup \texttt{LocalSample}(\mathcal{C}_{sat}^t, \hat{a}_S, \mathcal{D}_{total})$
  ▷ 5: Add the new samples; the next iteration refits
  $\mathcal{D}_{total} \leftarrow \mathcal{D}_{total} \cup \mathcal{Q}$; $t \leftarrow t + 1$; $\mathcal{C}_B^t = \emptyset$; $\mathcal{C}_{sat}^t = \emptyset$
return calibrated surrogate $\tilde{h}_g$; adversarial set $\{x \in \mathcal{D}_{total} : f_{\text{adv}}(x) \geq \phi\}$

Algorithm 2: LocalSample: Calibrated Sampling inside Trust Regions

Input: Trust Region Centers $\mathcal{C}$; local acquisition $\bar{\mathsf{a}}$ ($\bar{\mathsf{a}}_{bound}$ or $\bar{\mathsf{a}}_{sat}$); samples $\mathcal{D}_{total}$; local sampling budget b_L

regions $\mathcal{TR}_i \leftarrow \mathcal{C}_i$; $\mathcal{Q} \leftarrow \emptyset$; $\bar{\mathcal{D}}_{\mathcal{TR}} \leftarrow$ candidates drawn in $\mathcal{TR}$
if $\mathcal{D}_{total} \cap \mathcal{TR} = \emptyset$ **then**
  $\mathcal{Q} \leftarrow \{(c,\, f_{\text{adv}}(c))\}$ // empty region: label its center
for $m = 1, \ldots, b_L$ **do**
  ▷ Calibrate local model by LOO-CV (Williams and Rasmussen, 2006)
  $\mathcal{C}_{local} \leftarrow (\mathcal{D}_{total} \cap \mathcal{TR}) \cup \mathcal{Q}$
  $\tilde{h}_{\mathcal{TR}} \leftarrow L(\mathcal{C}_{local}) - L\Big(\big\{\big(x_i,\, h^{(-i)}(x_i) - f_{\text{adv}}(x_i)\big) \mid x_i \in \mathcal{C}_{local}\big\}\Big), \quad h^{(-i)} = L(\mathcal{C}_{local} \setminus \{x_i\})$
  ▷ take the best candidate under the calibrated local model
  $x^\star \leftarrow \arg\max_{x \in \bar{\mathcal{D}}_{\mathcal{TR}} \setminus \mathcal{Q}} \hat{a}(x;\, \tilde{h}_{\mathcal{TR}})$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(x^\star, f_{\text{adv}}(x^\star))\}$
return $\mathcal{Q}$

ever, in our application because each local region contains its own local model, we must calibrate both the global model for accurate placement, and our local model for accurate sampling. This is challenging because every local region is temporary and moves. Successfully calibrating each and every local model with a fresh hold out set as they get reinitialized in a new part of the space will easily eat up our sampling budget. Therefore, we devise two separate calibration schemes - one for our local models and another for the global model. Our global model's calibration scheme uses a uniformly sampled held out set to train its residual model (see Algorithm 1, while our local models use leave-one-out cross-validation (LOO-CV) from (Williams and Rasmussen, 2006) using only its training points to avoid depleting the sampling budget.

4.2 Acquisition Design for Placement and Sampling

ATLAS follows the classic high dimensional global-local sampling architecture from previous works (Ngo et al., 2026; Eriksson et al., 2019), where, first, local trust regions are selected, and then, points are sampled locally within those regions. As a consequence, there are two sampling loops - an outer loop that select trust region centers $\mathcal{C}^t$ for iteration t and inner loop that selects the next best samples for our dataset. Let $\mathcal{TR} = \{x \in \mathbb{R}^d \mid \|x-c\|_\infty < \frac{l}{2}\}$ denote our trust regions, which are hypercubes centered at c with side length l. Each trust region's center ζ_i is chosen as

$$\zeta_{j+1} \;=\; \underset{x \in \mathcal{U}(\mathcal{X}) \setminus \mathcal{C}^{t-1}}{\arg\max} \; \mathsf{a}(x) \;+\; \hat{\alpha} \prod_{\zeta_j \in Z} g\big(\|x - \zeta_j\|_2\big) \tag{1}$$

where $g(r) \;=\; \frac{1}{1+e^{-(r-\eta)/(\hat{\alpha}*\eta)}}$ represents a distance penalty, parameters η and $\hat{\alpha}$ represent desired seed separation and a multiplier respectively, a represents a designed acquisition placement function, and trust region centers $\mathcal{C}^{t-1}$ are the centers previously chosen during the last iteration. Every sample is selected simply using a designed acquisition sampling function α. Designing these acquisition functions is not trivial. Because ATLAS learns both an adversarial value function while curating an adversarial dataset, each acquisition must be designed in a way that improves the model's classification and increases the diversity of the adversarial set. This is challenging because each design criterion is not necessarily aligned. Prioritizing model classification often means training on points with higher model error, whereas sampling diverse adversarial points may mean sampling from regions you already know. To solve this problem, we devise two different acquisition functions: one that selects zero level samples with high model error and another that selects super level samples with high model error. We denote these zero-level based (boundary) and super-level based (saturation) placement acquisitions with a_{bound} and a_{sat} and their corresponding sampling functions with $\bar{\mathsf{a}}_{bound}$ and $\bar{\mathsf{a}}_{sat}$. Let the Gaussian Process $\mathcal{GP}_i$ with posterior mean $\mu_i(x)$ and variance $(\sigma_i(x))^2$ represent our calibrated model $\tilde{h}$. Our boundary-focused and saturation-focused sampling function selects candidates based on:

$$\bar{\mathsf{a}}_{bound}(x, \mu_i, \sigma_i) \;=\; (1-\rho)e^{-\left|\mu_i(x)\right|/s_{\text{bnd}}} + \rho(1 - e^{-c\beta\sigma_i(x)/s_{\text{bnd}}}) \tag{2}$$

and

$$\bar{\mathsf{a}}_{sat}(x, \mu_i, \sigma_i) \;=\; 1 - e^{-(\beta\,\sigma_i(x) + w\,P)/s_{\text{sat}}} \tag{3}$$

where

$$P \;= p_s(x, \mu_i(x), \sigma_i(x)) \;=\; 1 - \Phi\left(\frac{-\mu_i(x)}{\sigma_i(x)}\right) \tag{4}$$

denotes the posterior probability that x belongs to the adversarial super-level set. Φ represents the CDF of a normal distribution. Boundary (leveraging Bryan et al. (2005)) and saturation placements are designed as:

$$\mathsf{a}_{bound}(x) = c\,\beta\,\sigma_i(x) \;- \left|\mu_i(x)\right| \tag{5}$$

and

$$\mathsf{a}_{sat}(x) = \beta\,\sigma_i(x) \;+ \beta\,\sigma_i(x)\,P\,(1-P) \;+ w\,P \tag{6}$$

Each trust region is a hypercube of fixed volume V, i.e., side length $l = V^{1/d}$ (Table 5).

5 Toy Experiments: Finding Adversarial Regions in Highly Non-Convex Functions

We design a controlled experiment in which our black-box adversarial target f_{adv} is represented by a non-convex function f_{toy} (see Appendix B) and the goal is to recover a super-level set representing the space of possible attacks for the system $\mathcal{A} \coloneqq \{x \in \mathrm{R}^d : f_{toy}(x) > \phi\}$. We compare ATLAS to AP2L (Deglurkar et al., 2026) and TRLSE (Ngo et al., 2026). We use 2 test sets consisting of 50K super level samples ($\mathcal{S}$) and 1 million mixed points (Mix), where the mixed test set contains all super level samples and 950K other uniformly sampled points. A threshold $\phi \;=\; \text{Quantile}\Big(\{f_{\text{adv}}(x_i)\}_{i=1}^{N_{\text{thres}}},\, 1 - h_{\text{ratio}}\Big)$ for each experiment is chosen such that the ratio of the volume of the super level set to the volume of the total domain matches a chosen parameter $h_{\text{ratio}} = 0.05$, where $x \sim \mathcal{U}(\mathcal{X})$. For model performance, we measure the average F1 score and

report recall (over $\mathcal{S} \subset$ Mix) and precision. For each adversarial set, we measure the spread of adversarial samples using the mean distance between each pair of samples in the set:

$$\mathrm{Spread}_U(\mathcal{A}) = \binom{M}{2}^{-1} \sum_{i \neq j} \|\mathcal{A}_i - \mathcal{A}_j\|_2 \tag{7}$$

where $M = |\mathcal{A}|$. This metric is normalized by the spread of the test set for each experiment $\mathrm{Spread}(\mathcal{A}) = \frac{\mathrm{Spread}_U(\mathcal{A})}{\mathrm{Spread}_U(\mathcal{S})}$. Because this metric biases towards sets containing boundary samples (samples at the edge of the super level set will by definition have higher average pairwise distance than samples spread within the interior), we include 2 extra metrics that measures how spread the samples' adversarial scores are within the set and how deep inside the interior each method is able to go. The spread of adversarial scores (Margin J) is measured across $K = 20$ discretized bins b_k over the interval $[0, f(x^*) - \phi]$:

$$J(\mathcal{A}) = -\frac{1}{\ln K} \sum_{k=1}^{K} p_k \ln p_k, \qquad p_k = \frac{|\{x \in \mathcal{A} : f(x) - \phi \in b_k\}|}{|\mathcal{A}|} \tag{8}$$

where $J(\mathcal{A}) \in [0, 1]$ measures the normalized Shannon entropy ($1 =$ uniform coverage, $0 =$ single bin concentration). Super level penetration is measured using the maximum adversarial score contained in each adversarial set divided by the largest possible adversarial score:

$$\mathrm{mV}(\mathcal{A}) = \max_{x \in \mathcal{A}} \frac{f(x) - \phi}{f(x^*) - \phi} \tag{9}$$

This metric is reported alongside with the $\%$ of samples inside the adversarial set that exceeds the test set's ($\mathcal{S}$) largest adversarial score. The following tables report two ablation experiments and our main benchmark experiment. Table 1 and Table 2 report representative ablation results across all toy functions in 100D, while Table 3 contains the full results of our benchmark experiment; all extended analysis is in Appendix A. All experiments use 30 seeds and a budget of 650, with the best-performing calibration budget from our sweep (17%, Table 6). Calibration queries count against ATLAS's budget; AP2L receives its calibration set outside its budget. Full settings for ATLAS and the baselines are in Table 5, and Table 8 reports ATLAS with a single default weight setting shared across all benchmarks.

Table 1: **Calibration is what makes ATLAS work in high dimensions.** We vary the number of calibration samples N_C drawn from the fixed total budget $B = 650$, leaving $B - N_C$ for sampling. We use 30 seeds. At 100D, adding just 50 calibration samples restores classification performance (from .163, .003, and .100 to .322, .288 and .457 for Ack, Rast, Rosen). The benefit grows with dimension (See table 6 for full analysis).

Func	Dim	Calib. N_C (% of B)	F1 (Mix) ↑	Rec $\mathcal{S}$ % ↑	Prec (Mix) % ↑[1]
Ackley	100D	0 (0%)	$0.163_{.020}$	$9.2_{1.3}$	$\mathbf{77.1}_{2.7}$
		50 (8%)	$\mathbf{0.322}_{.069}$	$\mathbf{22.8}_{6.2}$	$60.3_{6.3}$
		100 (15%)	$0.305_{.056}$	$21.0_{5.7}$	$60.6_{5.1}$
		250 (38%)	$0.199_{.050}$	$12.1_{3.7}$	$63.4_{5.4}$
		375 (58%)	$0.029_{.028}$	$1.5_{1.5}$	$72.9_{10.6}$
Rast.	100D	0 (0%)	$0.003_{.002}$	$0.2_{0.1}$	$\mathbf{94.6}_{3.2}$
		50 (8%)	$\mathbf{0.288}_{.051}$	$\mathbf{20.3}_{5.2}$	$54.0_{6.1}$
		100 (15%)	$0.278_{.035}$	$19.1_{3.5}$	$53.7_{4.3}$
		250 (38%)	$0.251_{.030}$	$16.8_{2.7}$	$51.0_{3.3}$
		375 (58%)	$0.214_{.046}$	$13.7_{3.8}$	$51.6_{4.1}$
Rosen.	100D	0 (0%)	$0.100_{.035}$	$5.3_{2.0}$	$\mathbf{90.2}_{1.6}$
		50 (8%)	$\mathbf{0.457}_{.037}$	$\mathbf{43.6}_{9.6}$	$52.2_{9.9}$
		100 (15%)	$0.455_{.041}$	$41.9_{8.0}$	$52.8_{8.9}$
		250 (38%)	$0.447_{.026}$	$39.0_{4.6}$	$53.9_{7.1}$
		375 (58%)	$0.423_{.025}$	$36.1_{4.1}$	$52.7_{7.5}$

[1] No calibration attains high precision by predicting few super-level points, at the cost of recall. This is undesirable for a security analysis tool, whose purpose is to catch as many adversarial inputs as possible.

Table 2: **In 100D, saturation and boundary sampling allow ATLAS to produce a more diverse adversarial set while having robust model estimation.** All variants share the same sampling budget (B = 538) and number of trust regions; ATLAS-b uses only boundary regions, ATLAS-s only saturation, and ATLAS a 1:1 split. Subscripts are standard deviations. Each single-mode variant performs as designed. Saturation sampling harvests adversarial points (nSuper/B $\approx 90\%$), while boundary sampling gives the widest spatial spread. Each fails somewhere: ATLAS-s collapses in F1 on Rastrigin and Rosenbrock (≈ 0.01), and ATLAS-b finds almost no adversarial points on Rastrigin. Only ATLAS is robust across all landscapes. See Table 7 in the Appendix for full analysis.

			Learned Model Metrics			Sampling Set Metrics				
Func	**Dim**	**Arm**	**F1 (Mix) ↑**	**Rec $\mathcal{S}$ % ↑**	**Prec (Mix) % ↑**	**mV (%) ↑**	**Margin J ↑**	Spread ↑	**nSuper/B (%) ↑**	**B**
Ackley	100D	ATLAS-b	$0.223_{.082}$	$14.0_{6.0}$	$\mathbf{68.9}_{9.1}$	$0.039_{.022}$ (1.9)	$0.001_{.004}$	$\mathbf{0.913}_{.026}$	$48.3_{6.5}$	538
		ATLAS-s	$\mathbf{0.311}_{.067}$	$\mathbf{21.6}_{6.4}$	$60.4_{4.8}$	$\mathbf{0.053}_{.005}$ (21.9)	$\mathbf{0.006}_{.005}$	$0.825_{.010}$	$\mathbf{85.7}_{2.8}$	538
		ATLAS	$0.307_{.038}$	$20.8_{3.6}$	$60.6_{3.7}$	$0.051_{.006}$ (20.3)	$\mathbf{0.006}_{.007}$	$0.837_{.007}$	$52.0_{2.2}$	538
Rast.	100D	ATLAS-b	$0.255_{.059}$	$16.2_{5.1}$	$67.1_{5.7}$	$0.119_{.063}$ (0.0)	$0.249_{.128}$	$0.816_{.143}$	$1.5_{1.1}$	538
		ATLAS-s	$0.011_{.010}$	$0.6_{0.5}$	$\mathbf{83.2}_{8.1}$	$\mathbf{0.317}_{.017}$ (4.2)	$\mathbf{0.532}_{.013}$	$\mathbf{1.080}_{.021}$	$\mathbf{90.9}_{2.0}$	538
		ATLAS	$\mathbf{0.273}_{.036}$	$\mathbf{18.7}_{3.8}$	$52.9_{3.8}$	$0.304_{.023}$ (3.6)	$0.515_{.017}$	$1.032_{.018}$	$50.0_{1.7}$	538
Rosen.	100D	ATLAS-b	$\mathbf{0.503}_{.052}$	$\mathbf{62.5}_{11.3}$	$45.2_{10.9}$	$0.586_{.150}$ (3.8)	$0.649_{.080}$	$\mathbf{0.959}_{.027}$	$22.3_{8.6}$	538
		ATLAS-s	$0.015_{.011}$	$0.8_{0.6}$	$\mathbf{94.9}_{3.0}$	$0.657_{.030}$ (21.2)	$0.786_{.011}$	$0.799_{.007}$	$\mathbf{90.9}_{1.7}$	538
		ATLAS	$0.461_{.028}$	$41.0_{5.7}$	$55.0_{8.4}$	$\mathbf{0.844}_{.028}$ (32.8)	$\mathbf{0.914}_{.011}$	$0.839_{.007}$	$47.6_{2.8}$	538

6 Image Experiments: Mapping Adversarial Regions through Latent-Space Search

We evaluate whether ATLAS can recover adversarial sets for image-classification models. Following the generator-constrained attack setting of (Song et al., 2018), we search the latent space of a fixed class-conditional generator. We consider only the space of natural images for candidate adversaries and learn a value function, $\mathcal{V}(x)$ whose super level set represents latent samples $z \in \mathcal{Z}$ of a fixed generator, $G(z, s) \to I$ in which the decoded image represents a chosen source class $s \in \mathbf{R}^{n_c}$ that gets classified by the target as a chosen target class. For each dataset–model setting, we evaluate 20 source–target class pairs. We restrict the latent domain to a bounded region around the generator prior mean, $\mathcal{Z} = \left\{z \in \mathbb{R}^d \ : \ \|z - \mu_{latent}\|_\infty \leq z_{score}\sigma_{latent}\right\}$, where z_{score} indicated z_{score}-deviations away from the latent prior's mean μ_{latent}. To align successful targeted attacks with a super-zero level set, we define the negated objective function which takes from Carlini&Wagner (Carlini and Wagner, 2017):

$$\mathcal{O}(x'; t) = \widehat{Z}_t(x') - \max_{i \neq t} \widehat{Z}_i(x') \tag{10}$$

where $\widehat{Z}(x) = z$ be the output of all neural layers of the target system (the logits before softmax is applied) so $\mathcal{T}(x) = \text{softmax}(\widehat{Z}(x))$. Under this formulation, the successful targeted attack region $\mathcal{A} \coloneqq \{x \in \mathcal{X} : f_{\text{adv}}(x) \geq \phi\}$ corresponds directly to the non-negative super-level set of Eq. 10. We further use a nearest neighbor metric (Clark and Evans, 1954) to measure the ratio of mean nearest-neighbour distance to mean pairwise distance within the set:

$$\text{Spacing}(\mathcal{A}) = \frac{\frac{1}{M}\sum_i \min_{j \neq i} \|\mathcal{A}_i - \mathcal{A}_j\|_2}{\text{Spread}_U(\mathcal{A})} \tag{11}$$

Values near 1 indicate points spread evenly through the region they occupy; values near 0 indicate the set collapses into a few tight clusters. We compare ATLAS to NES, SignHunter, BayesOpt, and uniform random sampling (see Appendix D for details) and report the results in Table 4.

At 650-budget, the results show that ATLAS recovers the most representative adversarial set. Although ATLAS recovers fewer raw points than a couple of the baselines, it distributes its points far better than any of the other baselines. On the one hand, uniform random sampling clusters the least (by construction) with $\text{Spacing} = 0.88\text{-}0.93$ but finds the fewest adversarial points (1-19). On the other, NES and SignHunter finds the most adversarial points, but with either less spread $\text{Spread} = 1.14\text{-}2.43$ (v. 1.51-3.82) or more clustering $\text{Spacing} = 0.24\text{-}0.31$ (v. 0.55–0.65) respectively, implying that their points collapse into smaller, tighter neighborhoods. ATLAS is the only method that achieves best on Margin J in 4 of 5 targets with good size, ASR, spatial spread and spacing. Although, it may be slightly slower to the first success than BayesOpt ($Q_{first} = 83\text{–}229$ vs. 36–102 for BayesOpt), it spends its budget effectively finding more adversarial regions to find

Table 3: **ATLAS produces better recall and F1 while recovering more adversarial points per query, using a smaller sampling budget (B).** We compare ATLAS's calibration (112 samples) and 1:1 boundary-to-saturation sampling against our baselines at $B = 538$ vs. 650. Subscripts are standard deviations. ATLAS trades precision for recall (recall best in 9 of 9 and F1 (Mix) best in 8 of 9), yielding a model that misses fewer adversarial inputs. It also finds the most super-level points per unit budget (8 of 9) and recovers a better-distributed set — the highest margin spread and depth in every experiment, with dispersion within 30% of the best.

			Learned Model Metrics			Sampling Set Metrics				
Func	**Dim**	**Method**	**F1 (Mix)** ↑	**Rec $\mathcal{S}$ %** ↑	**Prec%** ↑	**mV (%)** ↑	**Margin J** ↑	Spread ↑	**nSuper/B (%)** ↑	**B**
Ackley	10D	AP2L	$0.088_{.203}$	$14.0_{29.9}$	$49.3_{37.9}$	$0.096_{.033}$ (0.0)	$0.128_{.072}$	$0.957_{.102}$	$5.5_{9.1}$	650
		TRLSE	$\mathbf{0.681}_{.005}$	$59.2_{1.5}$	$\mathbf{80.2}_{1.5}$	$0.113_{.044}$ (0.0)	$0.042_{.011}$	$\mathbf{1.019}_{.005}$	$26.0_{2.0}$	650
		ATLAS	$0.664_{.020}$	$\mathbf{63.6}_{4.9}$	$69.8_{4.1}$	$\mathbf{0.304}_{.048}$ (0.0)	$\mathbf{0.368}_{.016}$	$0.735_{.033}$	$\mathbf{53.7}_{3.3}$	538
	50D	AP2L	$0.010_{.024}$	$0.5_{1.3}$	$\mathbf{81.6}_{4.5}$	$0.021_{.009}$ (0.0)	$0.000_{.000}$	$\mathbf{0.991}_{.009}$	$2.9_{3.5}$	650
		TRLSE	$0.391_{.022}$	$27.4_{2.4}$	$68.6_{2.1}$	$0.035_{.008}$ (0.1)	$0.001_{.003}$	$0.943_{.003}$	$26.8_{1.2}$	650
		ATLAS	$\mathbf{0.454}_{.016}$	$\mathbf{61.9}_{4.4}$	$36.2_{3.2}$	$\mathbf{0.077}_{.009}$ (18.0)	$\mathbf{0.093}_{.018}$	$0.784_{.007}$	$\mathbf{52.0}_{1.1}$	538
	100D	AP2L	$0.002_{.002}$	$0.1_{0.1}$	$7.0_{1.6}$	$0.010_{.003}$ (0.0)	$0.000_{.000}$	$\mathbf{1.001}_{.004}$	$3.5_{2.6}$	650
		TRLSE	$0.182_{.031}$	$10.3_{2.0}$	$\mathbf{79.0}_{2.4}$	$0.021_{.003}$ (0.1)	$0.000_{.000}$	$0.928_{.002}$	$39.2_{1.2}$	650
		ATLAS	$\mathbf{0.307}_{.038}$	$\mathbf{20.8}_{3.6}$	$60.6_{3.7}$	$\mathbf{0.051}_{.006}$ (20.3)	$\mathbf{0.006}_{.007}$	$0.837_{.007}$	$\mathbf{52.0}_{2.2}$	538
Rast.	10D	AP2L	$0.259_{.192}$	$31.5_{30.5}$	$53.3_{23.7}$	$0.474_{.110}$ (0.1)	$0.643_{.061}$	$0.900_{.081}$	$12.9_{11.4}$	650
		TRLSE	$0.366_{.034}$	$26.0_{3.5}$	$\mathbf{62.9}_{1.5}$	$0.489_{.051}$ (0.0)	$0.608_{.020}$	$\mathbf{0.989}_{.012}$	$30.0_{2.0}$	650
		ATLAS	$\mathbf{0.432}_{.025}$	$\mathbf{51.1}_{10.0}$	$39.3_{5.7}$	$\mathbf{0.644}_{.061}$ (0.2)	$\mathbf{0.729}_{.014}$	$0.802_{.031}$	$\mathbf{48.7}_{3.0}$	538
	50D	AP2L	$0.245_{.201}$	$21.1_{21.1}$	$61.4_{14.0}$	$0.221_{.049}$ (0.0)	$0.388_{.067}$	$\mathbf{0.991}_{.015}$	$12.8_{11.7}$	650
		TRLSE	$0.396_{.019}$	$27.1_{1.8}$	$\mathbf{73.7}_{1.3}$	$0.257_{.032}$ (0.0)	$0.443_{.015}$	$0.952_{.004}$	$35.2_{1.5}$	650
		ATLAS	$\mathbf{0.471}_{.024}$	$\mathbf{38.8}_{4.6}$	$60.9_{3.3}$	$\mathbf{0.339}_{.029}$ (0.9)	$\mathbf{0.541}_{.021}$	$0.989_{.021}$	$\mathbf{42.4}_{2.6}$	538
	100D	AP2L	$0.227_{.144}$	$14.9_{9.6}$	$\mathbf{67.1}_{2.7}$	$0.150_{.029}$ (0.0)	$0.276_{.054}$	$0.995_{.005}$	$15.5_{10.2}$	650
		TRLSE	$0.000_{.000}$	$0.0_{0.0}$	–	$0.183_{.021}$ (0.0)	$0.327_{.015}$	$0.922_{.002}$	$47.1_{3.1}$	650
		ATLAS	$\mathbf{0.273}_{.036}$	$\mathbf{18.7}_{3.8}$	$52.9_{3.8}$	$\mathbf{0.304}_{.023}$ (3.6)	$\mathbf{0.515}_{.017}$	$\mathbf{1.032}_{.018}$	$\mathbf{50.0}_{1.7}$	538
Rosen.	10D	AP2L	$0.019_{.044}$	$10.5_{30.4}$	$50.3_{49.7}$	$0.676_{.127}$ (0.0)	$0.742_{.168}$	$0.993_{.030}$	$4.0_{3.1}$	650
		TRLSE	$0.682_{.016}$	$57.6_{2.3}$	$\mathbf{83.8}_{2.0}$	$0.812_{.059}$ (0.0)	$0.811_{.015}$	$\mathbf{1.086}_{.015}$	$32.8_{5.8}$	650
		ATLAS	$\mathbf{0.700}_{.015}$	$\mathbf{79.4}_{4.1}$	$63.1_{4.7}$	$\mathbf{0.879}_{.045}$ (0.0)	$\mathbf{0.913}_{.014}$	$0.959_{.020}$	$\mathbf{33.1}_{5.9}$	538
	50D	AP2L	$0.006_{.024}$	$3.6_{18.3}$	$60.7_{45.8}$	$0.326_{.058}$ (0.0)	$0.528_{.039}$	$\mathbf{0.997}_{.007}$	$3.8_{2.8}$	650
		TRLSE	$0.185_{.024}$	$10.6_{1.5}$	$75.2_{2.2}$	$0.536_{.053}$ (0.0)	$0.712_{.015}$	$0.921_{.004}$	$37.2_{2.2}$	650
		ATLAS	$\mathbf{0.300}_{.055}$	$\mathbf{18.8}_{4.6}$	$\mathbf{78.4}_{4.7}$	$\mathbf{0.730}_{.027}$ (3.0)	$\mathbf{0.852}_{.008}$	$0.846_{.006}$	$\mathbf{59.5}_{1.1}$	538
	100D	AP2L	$0.006_{.022}$	$3.6_{18.3}$	$4.1_{3.8}$	$0.250_{.055}$ (0.0)	$0.417_{.065}$	$\mathbf{0.998}_{.008}$	$3.5_{2.6}$	650
		TRLSE	$0.000_{.000}$	$0.0_{0.0}$	–	$0.397_{.025}$ (0.0)	$0.603_{.011}$	$0.937_{.001}$	$\mathbf{66.3}_{1.7}$	650
		ATLAS	$\mathbf{0.461}_{.028}$	$\mathbf{41.0}_{5.7}$	$\mathbf{55.0}_{8.4}$	$\mathbf{0.844}_{.028}$ (32.8)	$\mathbf{0.914}_{.011}$	$0.839_{.007}$	$47.6_{2.8}$	538

[1] Sampling deep inside the super level set sacrifices average dispersion, since samples there are closer together than samples along the boundary. As ATLAS goes deep into the super level set to produce a set with more representative margin spread, its set's coverage decreases on average, which is why our dispersion spread is somewhat lower than our baselines. Although it loses coverage, it gains coverage of adversarial scores that would not otherwise be recoverable with the other baselines or uniform sampling. At 50D and 100D, ATLAS recovers adversarial scores that are higher than every sample in the super level test set.

as many diverse model failures as possible as opposed to cheaply sampling the same attacks near already known regions. The former is much better for a robust security evaluation. Figure 3 shows ATLAS's representative sets for each target and Appendix F reports an example security audit.

7 Conclusion

In conclusion, this paper presents ATLAS, a framework that reformulates adversarial discovery as a region-recovery problem, leverages active learning and level-set estimation for this problem, and introduces new mechanisms that are specifically required for adversarial-set and adversarial classification construction. In this work, we find that ATLAS can construct more diverse adversarial sets, which provides a useful foundation for building automated red-teaming tools. There is still some variance in performance between function landscapes. Future work involves investigating on more targets and training target models on ATLAS's generated adversarial sets.

Table 4: **ATLAS matches the best attack success rate while returning the most representative adversarial set.** Each method is given $B = 650$ oracle queries per experiment, averaged over 20 source–target class pairs per target model; subscripts are standard deviations across pairs and the best value per metric and target is in bold. ASR is the fraction of pairs with at least one success; Q_{first} is queries to first success, computed over succeeding pairs only. Margin J, Spread and Spacing describe how the recovered points are distributed (higher = more of the region covered, covered more evenly). ATLAS is the only method achieving the best distribution of adversarial scores (Margin J) with good spatial spread and competitive set size. Standard CIFAR-10 and ImageNet targets are in Table 9.

			Attack		Adversarial Set			
Data	**Target**	**Method**	**ASR% ↑**	Q_{first} **↓**	**Margin J ↑**	Spread ↑	**Adv. Count ↑**	Spacing ↑
MNIST	Standard	**ATLAS**	95	164_{85}	$\mathbf{0.859}_{0.091}$	$1.94_{0.15}$	$174.4_{109.1}$	$0.63_{0.04}$
		NES	75	181_{162}	$0.745_{0.238}$	$1.14_{0.12}$	$225.9_{199.2}$	$0.58_{0.13}$
		SignHunter	**100**	60_{48}	$0.816_{0.075}$	$\mathbf{2.99}_{0.31}$	$\mathbf{255.4}_{99.4}$	$0.27_{0.03}$
		Uniform	26	408_{211}	$0.136_{0.207}$	$2.25_{0.03}$	$1.2_{3.5}$	$\mathbf{0.92}_{0.10}$
		BayesOpt[a]	**100**	$\mathbf{36}_{90}$	–	–	–	–
CIFAR-10	Adv.	**ATLAS**	**100**	83_{81}	$\mathbf{0.823}_{0.114}$	$1.88_{0.18}$	$203.4_{96.9}$	$0.57_{0.03}$
		NES	95	121_{95}	$0.803_{0.162}$	$1.24_{0.13}$	$282.6_{172.2}$	$0.52_{0.08}$
		SignHunter	**100**	47_{67}	$0.770_{0.109}$	$\mathbf{3.00}_{0.31}$	$\mathbf{306.6}_{142.4}$	$0.24_{0.05}$
		Uniform	85	96_{163}	$0.356_{0.204}$	$2.24_{0.09}$	$10.9_{15.8}$	$\mathbf{0.87}_{0.07}$
		BayesOpt[a]	**100**	$\mathbf{44}_{45}$	–	–	–	–
ImageNet	Adv.	**ATLAS**	**100**	162_{158}	$\mathbf{0.745}_{0.206}$	$3.82_{0.75}$	$115.1_{104.4}$	$0.55_{0.14}$
		NES	85	98_{133}	$0.703_{0.194}$	$2.43_{0.23}$	$144.6_{180.2}$	$0.61_{0.09}$
		SignHunter	85	147_{150}	$0.583_{0.189}$	$3.37_{0.97}$	$\mathbf{195.4}_{112.1}$	$0.26_{0.08}$
		Uniform	55	$\mathbf{25}_{118}$	$0.391_{0.264}$	$\mathbf{4.48}_{0.12}$	$15.3_{39.9}$	$\mathbf{0.93}_{0.05}$
		BayesOpt[a]	**100**	102_{121}	–	–	–	–

[a] BayesOpt is run in its published configuration, which stops at its first adversarial point, so it is read for ASR and Q_{first} only: its Count is 1 by construction.

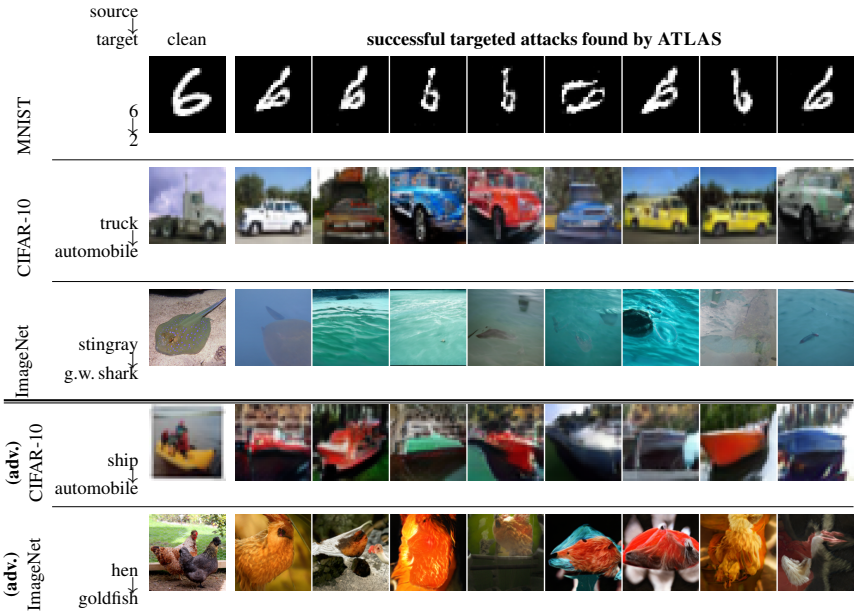


Figure 3: **Adversarial sets expose shared failure modes that single attacks cannot.** Representative examples discovered by ATLAS under a 650-query budget; rows below the double rule are adversarially trained targets. Each row is one source→target pair: the leftmost image is a clean source-class sample, the remainder are members of the recovered adversarial set. Read jointly, each set reveals what the target model keys on. For stingray→great white (g.w.) shark, every recovered image contains an open ocean rather than a shark's physical features; for goldfish targets, the recovered images contain flower petals and feathers perhaps standing in for fin-like features. These patterns point to areas for more needed study and analysis for the target model. Appendix E gives the full grid and analysis.

REFERENCES

NIST AI. Artificial intelligence risk management framework (ai rmf 1.0). *URL: https://nvlpubs. nist. gov/nistpubs/ai/nist. ai*, 1(0):0–1, 2023.

Abdullah Al-Dujaili and Una-May O'Reilly. Sign bits are all you need for black-box attacks. In *International conference on learning representations*, 2020.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pages 484–501. Springer, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

Gilbert Bahati, Marsalis Gibson, and Alexandre Bayen. Multi-adversarial safety analysis for autonomous vehicles. *arXiv preprint arXiv:2112.14344*, 2021.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Brent Bryan, Robert C Nichol, Christopher R Genovese, Jeff Schneider, Christopher J Miller, and Larry Wasserman. Active learning for identifying function threshold boundaries. *Advances in neural information processing systems*, 18, 2005.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

Shuyu Cheng, Yibo Miao, Yinpeng Dong, Xiao Yang, Xiao-Shan Gao, and Jun Zhu. Efficient black-box adversarial attacks via bayesian optimization guided by a function prior. *arXiv preprint arXiv:2405.19098*, 2024.

Fang-Chieh Chou, Marsalis Gibson, Rahul Bhadani, Alexandre M Bayen, and Jonathan Sprinkle. Reachability analysis for followerstopper: Safety analysis and experimental results. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8607–8613. IEEE, 2021.

Philip J Clark and Francis C Evans. Distance to nearest neighbor as a measure of spatial relationships in populations. *Ecology*, 35(4):445–453, 1954.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

Sampada Deglurkar, Ebonye Smith, Jingqi Li, and Claire J Tomlin. Active calibration of reachable sets using approximate pick-to-learn. *IEEE Control Systems Letters*, 2026.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

Marsalis Gibson, David Babazadeh, Claire Tomlin, and Shankar Sastry. Hacking predictors means hacking cars: Using sensitivity analysis to identify trajectory prediction vulnerabilities for autonomous driving security. In *6th Annual Learning for Dynamics & Control Conference*, pages 745–757. PMLR, 2024.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pages 2137–2146. PMLR, 2018.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Surgan Jandial, Puneet Mangla, Sakshi Varshney, and Vineeth Balasubramanian. Advgan++: Harnessing latent layers for adversary generation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2045–2048. IEEE, 2019.

Minguk Kang, Joonghyuk Shin, and Jaesik Park. Studiogan: a taxonomy and benchmark of gans for image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15725–15742, 2023.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Alexey Kurakin, Ian J Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world. *Artificial intelligence safety and security*, 2, 2018.

MITRE Corporation. Atlas: Adversarial threat landscape for artificial-intelligence systems. `https://atlas.mitre.org/`, 2021. Accessed: 2026-09-04.

Milad Nasr, Nicholas Carlini, Chawin Sitawarin, Sander V Schulhoff, Jamie Hayes, Michael Ilie, Juliette Pluto, Shuang Song, Harsh Chaudhari, Ilia Shumailov, et al. The attacker moves second: Stronger adaptive attacks bypass defenses against llm jailbreaks and prompt injections. *arXiv preprint arXiv:2510.09023*, 2025.

Giang Ngo, Dat Phan Trong, Dang Nguyen, and Sunil Gupta. High-dimensional level set estimation with trust regions and double acquisition functions. *arXiv preprint arXiv:2602.12391*, 2026.

Dario Paccagnan, Marco Campi, and Simone Garatti. The pick-to-learn algorithm: Empowering compression for tight generalization bounds and improved post-training performance. *Advances in Neural Information Processing Systems*, 36:18165–18185, 2023.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pages 4422–4431. IEEE, 2018.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

Binxin Ru, Adam Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. In *International conference on learning representations*, 2019.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in neural information processing systems*, 33:3533–3545, 2020.

Satya Narayan Shukla, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Black-box adversarial attacks with bayesian optimization. *arXiv preprint arXiv:1909.13857*, 2019.

Chandan Singh. Pretrained GANs in PyTorch for MNIST/CIFAR, May 2019a. URL `https://doi.org/10.5281/zenodo.2778737`.

Chandan Singh. gan-vae-pretrained-pytorch: Pretrained GANs + VAEs + classifiers for MNIST/CIFAR in PyTorch. `https://github.com/csinva/gan-vae-pretrained-pytorch`, 2019b. Accessed: 2026-09-25.

Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. *Advances in neural information processing systems*, 31, 2018.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

Lei Xu and Junhai Zhai. Dcvae-adv: A universal adversarial example generation method for white and black box attacks. *Tsinghua Science and Technology*, 29(2):430–446, 2023.

Qiusi Zhan, Richard Fang, Henil Shalin Panchal, and Daniel Kang. Adaptive attacks break defenses against indirect prompt injection attacks on llm agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7116–7132, 2025.

Shudong Zhang, Haichang Gao, Chao Shu, Xiwen Cao, Yunyi Zhou, and Jianping He. Black-box bayesian adversarial attack with transferable priors. *Machine Learning*, 113(4):1511–1528, 2024.

A Full Toy Experiment Evaluation

This supplementary appendix contains the comprehensive, fully expanded empirical tables across all benchmark geometry configurations: Ackley, Rastrigin, and Rosenbrock across 10D, 50D, and 100D settings. Our adversarial objective function for this experiment is: $\mathcal{O}(x) = -f_{toy}(x - p)$, where f_{toy} is the function being minimized and p is a design parameter used to offset the minimizer. Each method is run with 650 queries. Dataset sizes, calibration set sizes, and the number of trust regions are kept the same across methods. We use a 1:1 ratio of saturation to boundary regions and maintain 4 trust regions in all experiments. Full settings, including footnotes on how weights were chosen and where they deviate, are in Table 5. We do not alter the baseline implementations, since doing so would require re-tuning them. ATLAS includes several speed-ups to its optimization (Appendix C). Table 6 reports the full calibration sweep and Table 7 the full boundary/saturation ablation. Table 8 repeats the main benchmark using a single default weight setting (c=2, w=40) shared across all benchmarks in place of per-benchmark weights. ATLAS retains the best recall in 8 of 9 settings under these defaults.

Table 5: **Experiment settings for ATLAS and baselines.** Shared settings are identical across methods; matched settings are set equal between ATLAS and the baseline that has the counterpart. Baseline-specific values follow each method's reference implementation. d-dependent values are listed as 10D / 50D / 100D. Baseline-specific values follow each method's reference implementation. Weights c and w are set per benchmark to match the scale of $\beta\sigma$ (footnote 2). With a single default setting (c=2, w=40) across all benchmarks, ATLAS still achieves the best recall in 8 of 9 and the best F1 in 7 of 9 (Table 8), so the reported comparison does not rest on this choice. d-dependent values are listed as 10D / 50D / 100D.

Parameter	ATLAS	TRLSE	AP2L
Experimental Params			
query budget b_G	650 (112 calib. + 538)	650	650 (+112 calib.)[1]
seeds; test sets	30; 50K ($\mathcal{S}$), 1M (Mix)		
super-level fraction; functions	0.05; same domain and minimizer offset		
points per acquisition	1	1	1
surrogate	GP, Matérn-5/2	GP, Matérn-5/2 (ARD)	GP, Matérn-5/2
Common Params			
trust regions	2 bound. + 2 sat.	4	–
uncertainty weight β	1.96[4]	1.96[3]	1.28[3]
calibration samples	112	–	112
region volume V	0.1 / 0.1 / 0.5^{100} (fixed)	init 0.033/0.033/0.5^{100}; max 0.1/0.1/0.01	–
candidate pool[5]	20K	100K (Sobol)	100K
Method-specific			
ATLAS	c=2 (Rast. 100D: 16); w per setting[2]; ρ=0.5; $\hat{\alpha}$=10, w_g=0.05, η=2l; GP refit every 4 rounds		
TRLSE	$V_{\min} = V_{\text{init}}/2$; regions persist and resize; classify at threshold 0		
AP2L	$\hat{e}$ threshold 0.3; decay 0.95; conformal α=0.05		

[1] AP2L's calibration set is drawn outside its query budget; ATLAS's is counted against it.
[2] For Rosenbrock and Rastrigin, w scales with dimension. Ack. 40; Rosen. 10^3 / 10^5 / 5×10^6; Rast. 100 / (40) / 10^4. The saturation acquisition (Eq. 6) adds wP to $\beta\sigma_i(x)$, where σ_i is expressed in the target function's output units. These units differ by roughly five orders of magnitude across our benchmarks and grow with dimension, so no single w balances the two terms across settings. Therefore, to avoid a saturation acquisition dominated by the exploration term $\beta\sigma_i(x)$, we set w per benchmark to be at least the observed magnitude of $\beta\sigma_i$ ($w/\beta\sigma_i \in [1, 140]$), yielding the values above. Rastrigin 50D departs from this ($w = 40$) because raising w to that scale shifted budget from boundary sampling and reduced F1 (0.54 to 0.20). With a single default setting (c=2, w=40) for all benchmarks (Table 8), ATLAS still leads recall in 8 of 9 settings and F1 in 7 of 9, so the reported comparison does not rest on this choice.
[3] Method's original setting.
[4] Matched to TRLSE's setting.
[5] ATLAS draws from a smaller candidate pool (20K vs. 100K), which if anything disadvantages it

Table 6: **In general, calibration lets ATLAS learn a better model with fewer samples.** We vary the number of calibration samples N_C drawn from the fixed total budget $B = 650$, leaving $B - N_C$ for sampling. We evaluate over 30 seeds. A small calibration set (N_C = 50–100, 8–15% of B) gives the best or near-best F1 in every experiment and higher recall than no calibration in all nine. Thus, for our main experiment, we use $N_C = 112(17\%)$, inside of this range. The benefit grows with dimension. At 10D, the GP may learn well without calibration (Rosenbrock 10D), while at 100D, removing calibration may collapse performance (F1 for Rastrigin 0.003, Rosenbrock 0.100). While calibration is helpful, over-allocating the budget to calibration (58%) degrades performance.

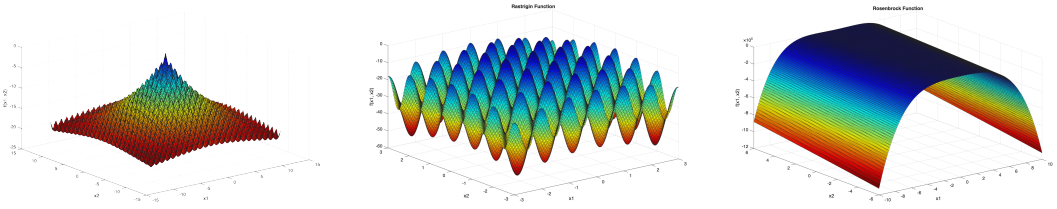
Func	Dim	Calib. N_C (% of B)	F1 (Mix) ↑	Rec $\mathcal{S}$ % ↑	Prec (Mix) % ↑[1]
Ackley	10D	0 (0%)	$0.678_{.012}$	$60.3_{2.1}$	$\mathbf{77.5}_{1.9}$
		50 (8%)	$0.663_{.022}$	$62.7_{5.4}$	$71.0_{4.1}$
		100 (15%)	$\mathbf{0.687}_{.025}$	$\mathbf{66.6}_{5.0}$	$71.4_{3.8}$
		250 (38%)	$0.664_{.016}$	$62.2_{5.6}$	$72.1_{5.8}$
		375 (58%)	$0.662_{.020}$	$63.3_{5.3}$	$70.3_{5.4}$
	50D	0 (0%)	$0.226_{.075}$	$14.1_{6.4}$	$\mathbf{72.0}_{9.1}$
		50 (8%)	$0.448_{.019}$	$\mathbf{63.5}_{4.6}$	$34.9_{3.3}$
		100 (15%)	$0.450_{.018}$	$63.1_{4.4}$	$35.2_{3.3}$
		250 (38%)	$\mathbf{0.460}_{.012}$	$59.8_{5.2}$	$37.8_{3.4}$
		375 (58%)	$0.367_{.075}$	$32.2_{10.8}$	$47.2_{5.0}$
	100D	0 (0%)	$0.163_{.020}$	$9.2_{1.3}$	$\mathbf{77.1}_{2.7}$
		50 (8%)	$\mathbf{0.322}_{.069}$	$\mathbf{22.8}_{6.2}$	$60.3_{6.3}$
		100 (15%)	$0.305_{.056}$	$21.0_{5.7}$	$60.6_{5.1}$
		250 (38%)	$0.199_{.050}$	$12.1_{3.7}$	$63.4_{5.4}$
		375 (58%)	$0.029_{.028}$	$1.5_{1.5}$	$72.9_{10.6}$
Rast.	10D	0 (0%)	$0.390_{.020}$	$30.0_{2.5}$	$56.2_{3.4}$
		50 (8%)	$\mathbf{0.430}_{.023}$	$\mathbf{50.6}_{12.4}$	$41.2_{10.3}$
		100 (15%)	$0.421_{.050}$	$35.1_{11.1}$	$\mathbf{61.0}_{10.9}$
		250 (38%)	$0.428_{.018}$	$49.8_{9.4}$	$40.1_{9.1}$
		375 (58%)	$0.422_{.021}$	$48.6_{6.0}$	$37.9_{3.6}$
	50D	0 (0%)	$0.304_{.022}$	$19.2_{1.8}$	$73.8_{1.9}$
		50 (8%)	$0.471_{.039}$	$\mathbf{39.8}_{7.7}$	$60.4_{5.6}$
		100 (15%)	$\mathbf{0.473}_{.018}$	$39.0_{3.2}$	$60.8_{2.5}$
		250 (38%)	$0.420_{.078}$	$32.1_{8.6}$	$65.9_{6.4}$
		375 (58%)	$0.191_{.140}$	$12.1_{9.8}$	$\mathbf{82.8}_{10.1}$
	100D	0 (0%)	$0.003_{.002}$	$0.2_{0.1}$	$\mathbf{94.6}_{3.2}$
		50 (8%)	$\mathbf{0.288}_{.051}$	$\mathbf{20.3}_{5.2}$	$54.0_{6.1}$
		100 (15%)	$0.278_{.035}$	$19.1_{3.5}$	$53.7_{4.3}$
		250 (38%)	$0.251_{.030}$	$16.8_{2.7}$	$51.0_{3.3}$
		375 (58%)	$0.214_{.046}$	$13.7_{3.8}$	$51.6_{4.1}$
Rosen.	10D	0 (0%)	$\mathbf{0.732}_{.006}$	$73.1_{2.5}$	$73.4_{2.1}$
		50 (8%)	$0.701_{.020}$	$\mathbf{80.9}_{4.7}$	$62.3_{5.6}$
		100 (15%)	$0.698_{.015}$	$79.3_{3.9}$	$62.6_{4.2}$
		250 (38%)	$0.691_{.013}$	$68.5_{5.4}$	$70.3_{4.9}$
		375 (58%)	$0.643_{.033}$	$56.7_{8.1}$	$\mathbf{76.5}_{6.3}$
	50D	0 (0%)	$0.333_{.038}$	$21.6_{3.7}$	$75.4_{3.3}$
		50 (8%)	$\mathbf{0.342}_{.065}$	$\mathbf{23.1}_{7.1}$	$73.7_{7.2}$
		100 (15%)	$0.312_{.057}$	$20.0_{5.2}$	$76.8_{5.6}$
		250 (38%)	$0.268_{.042}$	$16.2_{3.1}$	$82.6_{3.5}$
		375 (58%)	$0.209_{.046}$	$12.0_{3.2}$	$\mathbf{87.0}_{4.0}$
	100D	0 (0%)	$0.100_{.035}$	$5.3_{2.0}$	$\mathbf{90.2}_{1.6}$
		50 (8%)	$\mathbf{0.457}_{.037}$	$\mathbf{43.6}_{9.6}$	$52.2_{9.9}$
		100 (15%)	$0.455_{.041}$	$41.9_{8.0}$	$52.8_{8.9}$
		250 (38%)	$0.447_{.026}$	$39.0_{4.6}$	$53.9_{7.1}$
		375 (58%)	$0.423_{.025}$	$36.1_{4.1}$	$52.7_{7.5}$

[1] No calibration attains high precision by predicting few super-level points, at the cost of recall. This is undesirable for a security analysis tool, whose purpose is to catch as many adversarial inputs as possible.

Table 7: **Saturation and boundary sampling allow ATLAS to produce a more diverse adversarial set while having robust model estimation across functions and dimensions.** All variants use same sampling budget (B=538) and 4 trust regions. ATLAS-b uses only boundary regions, ATLAS-s only saturation, and ATLAS a 1:1 split. ATLAS-s and ATLAS have the best margin spread and depth. Saturation, as designed, helps harvest adversarial examples, winning nSuper/B in 7 of 9 cells, often at about 90%. ATLAS-b has the best or near best spatial spread and precision[1] in most experiments, helping us explore widely and find boundary points, but loses at margin spread and depth. ATLAS is the only variant that is robust across all landscapes with model estimation and adversarial set quality. This highlights the need to balance both sampling designs.

			Learned Model Metrics			Sampling Set Metrics				
Func	**Dim**	**Arm**	**F1 (Mix)** ↑	**Rec $\mathcal{S}$ %** ↑	**Prec (Mix) %** ↑	**mV (%)** ↑	**Margin J** ↑	Spread ↑	**nSuper/B (%)** ↑	**B**
Ackley	10D	ATLAS-b	$0.613_{.060}$	$\mathbf{79.6}_{4.6}$	$51.1_{10.3}$	$0.126_{.043}$ (0.0)	$0.074_{.016}$	$\mathbf{0.973}_{.008}$	$38.8_{2.0}$	538
		ATLAS-s	$0.466_{.047}$	$71.3_{8.8}$	$35.4_{6.7}$	$\mathbf{0.315}_{.045}$ (0.0)	$0.358_{.011}$	$0.734_{.018}$	$\mathbf{97.6}_{0.4}$	538
		ATLAS	$\mathbf{0.664}_{.020}$	$63.6_{4.9}$	$\mathbf{69.8}_{4.1}$	$0.304_{.048}$ (0.0)	$\mathbf{0.368}_{.016}$	$0.735_{.033}$	$53.7_{3.3}$	538
	50D	ATLAS-b	$0.010_{.054}$	$0.6_{3.4}$	–	$0.038_{.009}$ (0.1)	$0.001_{.003}$	$\mathbf{0.929}_{.006}$	$50.9_{2.4}$	538
		ATLAS-s	$\mathbf{0.466}_{.012}$	$\mathbf{65.0}_{3.7}$	$\mathbf{36.5}_{2.5}$	$\mathbf{0.080}_{.010}$ (19.1)	$\mathbf{0.097}_{.019}$	$0.793_{.010}$	$\mathbf{90.7}_{2.8}$	538
		ATLAS	$0.454_{.016}$	$61.9_{4.4}$	$36.2_{3.2}$	$0.077_{.009}$ (18.0)	$0.093_{.018}$	$0.784_{.007}$	$52.0_{1.1}$	538
	100D	ATLAS-b	$0.223_{.082}$	$14.0_{6.0}$	$\mathbf{68.9}_{9.1}$	$0.039_{.022}$ (1.9)	$0.001_{.004}$	$\mathbf{0.913}_{.026}$	$48.3_{6.5}$	538
		ATLAS-s	$\mathbf{0.311}_{.067}$	$\mathbf{21.6}_{6.4}$	$60.4_{4.8}$	$\mathbf{0.053}_{.005}$ (21.9)	$\mathbf{0.006}_{.005}$	$0.825_{.010}$	$\mathbf{85.7}_{2.8}$	538
		ATLAS	$0.307_{.038}$	$20.8_{3.6}$	$60.6_{3.7}$	$0.051_{.006}$ (20.3)	$\mathbf{0.006}_{.007}$	$0.837_{.007}$	$52.0_{2.2}$	538
Rast.	10D	ATLAS-b	$0.409_{.096}$	$46.8_{16.6}$	$\mathbf{43.9}_{13.1}$	$0.456_{.053}$ (0.0)	$0.630_{.016}$	$\mathbf{0.976}_{.026}$	$37.2_{5.4}$	538
		ATLAS-s	$0.414_{.027}$	$41.7_{7.8}$	$43.5_{8.3}$	$\mathbf{0.655}_{.047}$ (0.1)	$\mathbf{0.765}_{.012}$	$0.669_{.032}$	$\mathbf{63.9}_{4.5}$	538
		ATLAS	$\mathbf{0.432}_{.025}$	$\mathbf{51.1}_{10.0}$	$39.3_{5.7}$	$0.644_{.061}$ (0.2)	$0.729_{.014}$	$0.802_{.031}$	$48.7_{3.0}$	538
	50D	ATLAS-b	$0.298_{.087}$	$19.0_{7.1}$	$\mathbf{79.8}_{6.2}$	$0.308_{.024}$ (0.3)	$0.512_{.030}$	$0.977_{.029}$	$\mathbf{51.5}_{4.6}$	538
		ATLAS-s	$0.414_{.046}$	$29.7_{4.8}$	$71.0_{4.1}$	$0.312_{.048}$ (0.6)	$0.515_{.021}$	$\mathbf{1.019}_{.015}$	$33.1_{3.0}$	538
		ATLAS	$\mathbf{0.471}_{.024}$	$\mathbf{38.8}_{4.6}$	$60.9_{3.3}$	$\mathbf{0.339}_{.029}$ (0.9)	$\mathbf{0.541}_{.021}$	$0.989_{.021}$	$42.4_{2.6}$	538
	100D	ATLAS-b	$0.255_{.059}$	$16.2_{5.1}$	$67.1_{5.7}$	$0.119_{.063}$ (0.0)	$0.249_{.128}$	$0.816_{.143}$	$1.5_{1.1}$	538
		ATLAS-s	$0.011_{.010}$	$0.6_{0.5}$	$\mathbf{83.2}_{8.1}$	$\mathbf{0.317}_{.017}$ (4.2)	$\mathbf{0.532}_{.013}$	$\mathbf{1.080}_{.021}$	$\mathbf{90.9}_{2.0}$	538
		ATLAS	$\mathbf{0.273}_{.036}$	$\mathbf{18.7}_{3.8}$	$52.9_{3.8}$	$0.304_{.023}$ (3.6)	$0.515_{.017}$	$1.032_{.018}$	$50.0_{1.7}$	538
Rosen.	10D	ATLAS-b	$0.650_{.057}$	$53.7_{10.4}$	$\mathbf{85.7}_{6.5}$	$0.817_{.050}$ (0.0)	$0.858_{.015}$	$\mathbf{1.041}_{.014}$	$\mathbf{49.4}_{2.8}$	538
		ATLAS-s	$0.691_{.011}$	$76.0_{4.7}$	$63.8_{4.0}$	$0.871_{.050}$ (0.1)	$0.885_{.026}$	$0.852_{.034}$	$25.7_{9.3}$	538
		ATLAS	$\mathbf{0.700}_{.015}$	$\mathbf{79.4}_{4.1}$	$63.1_{4.7}$	$\mathbf{0.879}_{.045}$ (0.0)	$\mathbf{0.913}_{.014}$	$0.959_{.020}$	$33.1_{5.9}$	538
	50D	ATLAS-b	$0.165_{.108}$	$9.7_{6.7}$	$84.4_{6.3}$	$0.529_{.080}$ (0.0)	$0.650_{.019}$	$\mathbf{0.958}_{.009}$	$38.8_{3.5}$	538
		ATLAS-s	$0.096_{.055}$	$5.2_{3.2}$	$\mathbf{90.2}_{4.6}$	$0.716_{.031}$ (1.1)	$0.794_{.015}$	$0.822_{.007}$	$\mathbf{91.3}_{3.2}$	538
		ATLAS	$\mathbf{0.300}_{.055}$	$\mathbf{18.8}_{4.6}$	$78.4_{4.7}$	$\mathbf{0.730}_{.027}$ (3.0)	$\mathbf{0.852}_{.008}$	$0.846_{.006}$	$59.5_{1.1}$	538
	100D	ATLAS-b	$\mathbf{0.503}_{.052}$	$\mathbf{62.5}_{11.3}$	$45.2_{10.9}$	$0.586_{.150}$ (3.8)	$0.649_{.080}$	$\mathbf{0.959}_{.027}$	$22.3_{8.6}$	538
		ATLAS-s	$0.015_{.011}$	$0.8_{0.6}$	$\mathbf{94.9}_{3.0}$	$0.657_{.030}$ (21.2)	$0.786_{.011}$	$0.799_{.007}$	$\mathbf{90.9}_{1.7}$	538
		ATLAS	$0.461_{.028}$	$41.0_{5.7}$	$55.0_{8.4}$	$\mathbf{0.844}_{.028}$ (32.8)	$\mathbf{0.914}_{.011}$	$0.839_{.007}$	$47.6_{2.8}$	538

[1] – denotes precision undefined (no super-level predictions).



(a) Ackley Function (b) Rastrigin Function (c) Rosenbrock Function

Figure 4: Non-convex toy functions used in our toy experiments described in Section 5.

Table 8: **ATLAS with default weights.** Same setup as Table 3, but with a single setting (c=2, w=40) for every benchmark. Subscripts are standard deviations. ATLAS still achieves the best recall in 8 of 9 experiments, the best F1 (Mix) in 7 of 9, and the highest depth (mV) in all 9, with dispersion within 30% of the best. Per-benchmark weights (Table 3) mainly improve Rastrigin 100D and the adversarial-set metrics.

			Learned Model Metrics			Sampling Set Metrics				
Func	**Dim**	**Method**	**F1** ↑	**Rec $\mathcal{S}$ %** ↑	**Prec%** ↑	**mV (%)** ↑	**Margin J** ↑	Spread ↑	**nSuper/B (%)** ↑	**B**
Ackley	10D	AP2L	$0.088_{.203}$	$14.0_{29.9}$	$49.3_{37.9}$	$0.096_{.033}$ (0.0)	$0.128_{.072}$	$0.957_{.102}$	$5.5_{9.1}$	650
		TRLSE	$\mathbf{0.681}_{.005}$	$59.2_{1.5}$	$\mathbf{80.2}_{1.5}$	$0.113_{.044}$ (0.0)	$0.042_{.011}$	$\mathbf{1.019}_{.005}$	$26.0_{2.0}$	650
		ATLAS	$0.664_{.020}$	$\mathbf{63.6}_{4.9}$	$69.8_{4.1}$	$\mathbf{0.304}_{.048}$ (0.0)	$\mathbf{0.368}_{.016}$	$0.735_{.033}$	$\mathbf{53.7}_{3.3}$	538
	50D	AP2L	$0.010_{.024}$	$0.5_{1.3}$	$\mathbf{81.6}_{4.5}$	$0.021_{.009}$ (0.0)	$0.000_{.000}$	$\mathbf{0.991}_{.009}$	$2.9_{3.5}$	650
		TRLSE	$0.391_{.022}$	$27.4_{2.4}$	$68.6_{2.1}$	$0.035_{.008}$ (0.1)	$0.001_{.003}$	$0.943_{.003}$	$26.8_{1.2}$	650
		ATLAS	$\mathbf{0.454}_{.016}$	$\mathbf{61.9}_{4.4}$	$36.2_{3.2}$	$\mathbf{0.077}_{.009}$ (18.0)	$\mathbf{0.093}_{.018}$	$0.784_{.007}$	$\mathbf{52.0}_{1.1}$	538
	100D	AP2L	$0.002_{.002}$	$0.1_{0.1}$	$7.0_{1.6}$	$0.010_{.003}$ (0.0)	$0.000_{.000}$	$\mathbf{1.001}_{.004}$	$3.5_{2.6}$	650
		TRLSE	$0.182_{.031}$	$10.3_{2.0}$	$\mathbf{79.0}_{2.4}$	$0.021_{.003}$ (0.1)	$0.000_{.000}$	$0.928_{.002}$	$39.2_{1.2}$	650
		ATLAS	$\mathbf{0.307}_{.038}$	$\mathbf{20.8}_{3.6}$	$60.6_{3.7}$	$\mathbf{0.051}_{.006}$ (20.3)	$\mathbf{0.006}_{.007}$	$0.837_{.007}$	$\mathbf{52.0}_{2.2}$	538
Rast.	10D	AP2L	$0.259_{.192}$	$31.5_{30.5}$	$53.3_{23.7}$	$0.474_{.110}$ (0.1)	$0.643_{.061}$	$0.900_{.081}$	$12.9_{11.4}$	650
		TRLSE	$0.366_{.034}$	$26.0_{3.5}$	$\mathbf{62.9}_{1.5}$	$0.489_{.051}$ (0.0)	$0.608_{.020}$	$\mathbf{0.989}_{.012}$	$30.0_{2.0}$	650
		ATLAS	$\mathbf{0.413}_{.043}$	$\mathbf{34.5}_{7.2}$	$54.2_{5.8}$	$\mathbf{0.600}_{.053}$ (0.0)	$\mathbf{0.720}_{.021}$	$0.798_{.040}$	$\mathbf{45.9}_{2.6}$	538
	50D	AP2L	$0.245_{.201}$	$21.1_{21.1}$	$61.4_{14.0}$	$0.221_{.049}$ (0.0)	$0.388_{.067}$	$\mathbf{0.991}_{.015}$	$12.8_{11.7}$	650
		TRLSE	$0.396_{.019}$	$27.1_{1.8}$	$\mathbf{73.7}_{1.3}$	$0.257_{.032}$ (0.0)	$0.443_{.015}$	$0.952_{.004}$	$35.2_{1.5}$	650
		ATLAS	$\mathbf{0.470}_{.024}$	$\mathbf{38.8}_{4.6}$	$60.9_{3.3}$	$\mathbf{0.339}_{.029}$ (0.9)	$\mathbf{0.541}_{.021}$	$0.989_{.021}$	$\mathbf{42.2}_{2.6}$	538
	100D	AP2L	$\mathbf{0.227}_{.144}$	$\mathbf{14.9}_{9.6}$	$67.1_{2.7}$	$0.150_{.029}$ (0.0)	$0.276_{.054}$	$0.995_{.005}$	$15.5_{10.2}$	650
		TRLSE	$0.000_{.000}$	$0.0_{0.0}$	–	$0.183_{.021}$ (0.0)	$0.327_{.015}$	$0.922_{.002}$	$\mathbf{47.1}_{3.1}$	650
		ATLAS	$0.066_{.030}$	$3.5_{1.7}$	$\mathbf{78.6}_{6.5}$	$\mathbf{0.203}_{.037}$ (0.1)	$\mathbf{0.367}_{.046}$	$\mathbf{1.089}_{.037}$	$29.0_{8.7}$	538
Rosen.	10D	AP2L	$0.019_{.044}$	$10.5_{30.4}$	$50.3_{49.7}$	$0.676_{.127}$ (0.0)	$0.742_{.168}$	$0.993_{.030}$	$4.0_{3.1}$	650
		TRLSE	$0.682_{.016}$	$57.6_{2.3}$	$\mathbf{83.8}_{2.0}$	$0.812_{.059}$ (0.0)	$0.811_{.015}$	$\mathbf{1.086}_{.015}$	$\mathbf{32.8}_{5.8}$	650
		ATLAS	$\mathbf{0.693}_{.013}$	$\mathbf{79.3}_{3.9}$	$61.9_{4.0}$	$\mathbf{0.877}_{.051}$ (0.0)	$\mathbf{0.910}_{.020}$	$0.952_{.028}$	$28.8_{2.8}$	538
	50D	AP2L	$0.006_{.024}$	$3.6_{18.3}$	$60.7_{45.8}$	$0.326_{.058}$ (0.0)	$0.528_{.039}$	$\mathbf{0.997}_{.007}$	$3.8_{2.8}$	650
		TRLSE	$0.185_{.024}$	$10.6_{1.5}$	$\mathbf{75.2}_{2.2}$	$0.536_{.053}$ (0.0)	$\mathbf{0.712}_{.015}$	$0.921_{.004}$	$\mathbf{37.2}_{2.2}$	650
		ATLAS	$\mathbf{0.498}_{.024}$	$\mathbf{43.8}_{4.7}$	$58.5_{3.5}$	$\mathbf{0.567}_{.071}$ (0.1)	$0.680_{.026}$	$0.955_{.008}$	$17.3_{1.3}$	538
	100D	AP2L	$0.006_{.022}$	$3.6_{18.3}$	$4.1_{3.8}$	$0.250_{.055}$ (0.0)	$0.417_{.065}$	$\mathbf{0.998}_{.008}$	$3.5_{2.6}$	650
		TRLSE	$0.000_{.000}$	$0.0_{0.0}$	–	$0.397_{.025}$ (0.0)	$0.603_{.011}$	$0.937_{.001}$	$\mathbf{66.3}_{1.7}$	650
		ATLAS	$\mathbf{0.371}_{.020}$	$\mathbf{67.4}_{4.8}$	$\mathbf{25.7}_{2.1}$	$\mathbf{0.672}_{.153}$ (21.0)	$\mathbf{0.797}_{.122}$	$0.834_{.057}$	$13.4_{5.0}$	538

[1] Sampling deep inside the super level set sacrifices average dispersion, since samples there are closer together than samples along the boundary. As ATLAS goes deep into the super level set to produce a set with more representative margin spread, its set's coverage decreases on average, which is why our dispersion spread is somewhat lower than our baselines. Although it loses coverage, it gains coverage of adversarial scores that would not otherwise be recoverable with the other baselines or uniform sampling. At 50D and 100D, ATLAS recovers adversarial scores that are higher than every sample in the super level test set.

B Functions Used In Toy Experiments

B.1 Ackley Function

This function is described by the following:

$$f_{toy}(\mathbf{x}) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1) \tag{12}$$

where $\mathbf{x} \in \mathbf{R}^d$, and $a = 20$, $b = 0.2$, and $c = 2\pi$. The global minimum has value 0 at $\mathbf{x} = \mathbf{0}$.

B.2 Rastrigin Function

This function is described by the following:

$$f_{toy}(\mathbf{x}) = Ad + \sum_{i=1}^{d}\left[x_i^2 - A\cos(2\pi x_i)\right] \tag{13}$$

where $\mathbf{x} \in \mathbf{R}^d$, and $A = 10$. The global minimum has value 0 at $\mathbf{x} = \mathbf{0}$.

B.3 Rosenbrock Function

This function is described by the following:

$$f_{toy}(\mathbf{x}) = \sum_{i=1}^{d-1} \left[b(x_{i+1} - x_i^2)^2 + (a - x_i)^2 \right] \tag{14}$$

where $\mathbf{x} \in \mathbf{R}^d$, and $a = 1$, $b = 100$. The global minimum has value 0 at $\mathbf{x} = \mathbf{1} = (1, \ldots, 1)$.

C Implementation Speed up

In large dimensions, most of ATLAS's runtime comes from training a new global model on samples and choosing the new TR centers using limited memory BFGS with bound constraints (L-BFGS-B) and warm starts. To address the latter, we ultimately use analytic gradients during optimization[1] and parallelize specific calculations. Both mechanisms give us tremendous speed ups, which are most beneficial after the model starts acquiring a large amount of samples during its run. The timing metrics shown are runtimes when using CPU. Depending on the choice of model and learning algorithm, further speeds up can be realized by using GPU resources.

D Image Experiment Details

The models chosen are ResNet50 (He et al., 2016) for CIFAR's and ImageNet's standard target with 93.65% and 76.13% accuracy respectively, WRN-28-10 adversarially trained (**AT**) with DDPM (Rebuffi et al., 2021) with 60.75% accuracy for CIFAR's **AT** target, ResNet50 **AT** on L_{inf} perturbations (Salman et al., 2020) with 54.53% accuracy for ImageNet's **AT** target, and a pre-trained LeNet5 from (Singh, 2019a) with 99% accuracy. We use a conditional DCGAN (Radford et al., 2015) and BIGGAN and BIGGAN-256 (Brock et al., 2018) for our image generators for MNIST, CIFAR, and ImageNet respectively. BIGGAN and the conditional DCGAN was retrained from scratch using (Singh, 2019b) implementation to take in 32 dimensions and a source class, while the implementation and trained weights for BIGGAN-256 comes from StudioGAN(Kang et al., 2023). BayesOpt is run in its native early-stopping configuration, while NES and SignHunter is ran until the full budget generating its version of an adversarial set. We tried to run BayesOpt similarly; however, we found it to be more computationally infeasible to do. We therefore report ASR and query-to-success for BayesOpt, and omit the dispersion statistics, which require the run-to-budget super-level sets. Full analysis of ATLAS, NES, SignHunter, BayesOpt, and uniform sampling is in Table 9.

E Interpreting ATLAS Image Results

For each experiment, we ask: (1) whether the generated adversarial sets contain images with visual semantics that are common under the source class with little to no features of the target class or (2) whether there are any particular characteristics or patterns that emerge from all images (realistic or not) that could indicate which features the model's boundary region for the target class is built on. The generated adversarial sets for both MNIST and CIFAR represent the former. These are interesting failure cases because it represents manifold-constrained adversarial images which could present as benign or containing imperceptible perturbations but that gets misclassified by the model (most similar to L_∞ attacks). On the other hand, the generated adversarial sets for ImageNet represent more of the latter. The latter is important analysis because it tells us what features the models may hone in on for a particular classification and tells us what features push the model's decision boundary from one class to another. Take as example the "stingray $\rightarrow$ great white shark (g.w. shark)" row in the "Imagenet" section in Figure 5. All images contain a birds-eye view of the ocean with sightings of a stingray in the water which are all reasonable images for the stingray class. Because these images are being classified as a great white shark, this tells us that the model is most likely to key in on the ocean (a shark's environment) rather than a shark's features for its classification.

[1] We experiment between using analytic gradient of our placement acquisition functions and using batched finite difference as gradients. While each have their specific speed up benefits, the former displayed higher speeds for larger dimensions during longer runs.

Table 9: **At a matched query budget, ATLAS returns better distributed adversarial sets than the query-based attacks, at comparable attack success.** We compare ATLAS to NES, SignHunter, BayesOpt, and uniform random sampling at finding generated images under a source class that gets misclassified into a target class by the target model. We average each metric over 20 different source-target pairs per target model, and give each method a budget of $B = 650$ oracle queries per experiment. Subscripts are standard deviations across pairs. Best per metric and target in bold. ASR is the fraction of pairs for which the method finds at least one adversarial example. Q_{first} is the median queries to the first adversarial example and is computed over the pairs on which the method succeeded. Thus, it is not comparable across methods with different ASR. 'Adv. Count' is the number of adversarial examples found. Margin J, Spread and Spacing describe how those points are distributed (higher = more of the region covered, and covered more evenly). At 650-budget, ATLAS recovers the most representative adversarial set. It is best on Margin J in 4 of 5 targets with good spatial spread and spacing. ATLAS recovers fewer raw points than the query-based attacks but distributes them far better. The other query baselines are good at finding adversarial examples but poor at characterizing the adversarial region. SignHunter finds the most points, yet its Spacing of 0.24–0.31 versus ATLAS's 0.55–0.65 shows they are in tight clusters; NES's Spread of 1.14–2.43 versus ATLAS's 1.51–3.82 shows its points are in smaller neighbourhoods than ATLAS. Uniform random sampling has more spread by construction, but finds only 1–19 points on average.

			Attack		**Adversarial Set**			
Data	**Target**	**Method**	**ASR%** ↑	Q_{first} ↓	**Margin J** ↑	Spread ↑	**Adv. Count** ↑	Spacing ↑
MNIST	Standard	**ATLAS**	95	164_{85}	$\mathbf{0.859}_{0.091}$	$1.94_{0.15}$	$174.4_{109.1}$	$0.63_{0.04}$
		NES	75	181_{162}	$0.745_{0.238}$	$1.14_{0.12}$	$225.9_{199.2}$	$0.58_{0.13}$
		SignHunter	**100**	60_{48}	$0.816_{0.075}$	$\mathbf{2.99}_{0.31}$	$\mathbf{255.4}_{99.4}$	$0.27_{0.03}$
		Uniform	26	408_{211}	$0.136_{0.207}$	$2.25_{0.03}$	$1.2_{3.5}$	$\mathbf{0.92}_{0.10}$
		BayesOpt[a]	**100**	$\mathbf{36}_{90}$	–	–	–	–
CIFAR-10	Standard	**ATLAS**	95	203_{139}	$\mathbf{0.818}_{0.200}$	$1.51_{0.30}$	$71.8_{64.7}$	$0.63_{0.13}$
		NES	65	143_{214}	$0.804_{0.267}$	$1.18_{0.15}$	$94.2_{119.4}$	$0.57_{0.10}$
		SignHunter	**100**	72_{104}	$0.762_{0.078}$	$\mathbf{2.92}_{0.41}$	$\mathbf{256.6}_{122.8}$	$0.27_{0.09}$
		Uniform	60	106_{103}	$0.492_{0.259}$	$2.23_{0.15}$	$5.2_{9.0}$	$\mathbf{0.88}_{0.07}$
		BayesOpt[a]	95	$\mathbf{57}_{54}$	–	–	–	–
CIFAR-10	Adv.	**ATLAS**	**100**	83_{81}	$\mathbf{0.823}_{0.114}$	$1.88_{0.18}$	$203.4_{96.9}$	$0.57_{0.03}$
		NES	95	121_{95}	$0.803_{0.162}$	$1.24_{0.13}$	$282.6_{172.2}$	$0.52_{0.08}$
		SignHunter	**100**	47_{67}	$0.770_{0.109}$	$\mathbf{3.00}_{0.31}$	$\mathbf{306.6}_{142.4}$	$0.24_{0.05}$
		Uniform	85	96_{163}	$0.356_{0.204}$	$2.24_{0.09}$	$10.9_{15.8}$	$\mathbf{0.87}_{0.07}$
		BayesOpt[a]	**100**	$\mathbf{44}_{45}$	–	–	–	–
ImageNet	Standard	**ATLAS**	85	229_{168}	$0.609_{0.272}$	$3.33_{0.93}$	$86.4_{112.8}$	$0.65_{0.16}$
		NES	90	$\mathbf{97}_{200}$	$\mathbf{0.720}_{0.250}$	$2.41_{0.34}$	$175.6_{207.4}$	$0.63_{0.13}$
		SignHunter	**95**	151_{176}	$0.538_{0.242}$	$2.47_{1.16}$	$\mathbf{184.1}_{94.8}$	$0.31_{0.17}$
		Uniform	60	107_{185}	$0.342_{0.262}$	$\mathbf{4.49}_{0.13}$	$18.9_{40.7}$	$\mathbf{0.93}_{0.06}$
		BayesOpt[a]	90[b]	130_{121}	–	–	–	–
ImageNet	Adv.	**ATLAS**	**100**	162_{158}	$\mathbf{0.745}_{0.206}$	$3.82_{0.75}$	$115.1_{104.4}$	$0.55_{0.14}$
		NES	85	98_{133}	$0.703_{0.194}$	$2.43_{0.23}$	$144.6_{180.2}$	$0.61_{0.09}$
		SignHunter	85	147_{150}	$0.583_{0.189}$	$3.37_{0.97}$	$\mathbf{195.4}_{112.1}$	$0.26_{0.08}$
		Uniform	55	$\mathbf{25}_{118}$	$0.391_{0.264}$	$\mathbf{4.48}_{0.12}$	$15.3_{39.9}$	$\mathbf{0.93}_{0.05}$
		BayesOpt[a]	**100**	102_{121}	–	–	–	–

[a] BayesOpt is run in its published configuration, which stops at its first adversarial point, so it is read for ASR and Q_{first} only: its Count is 1 by construction.

[b] 2 of the 20 batches for BayesOpt targeting the standard ImageNet were terminated. We cap a single batch at ∼12 h of wall-clock time, which they exceeded having completed only ∼490 and ∼585 of 650 queries. Neither had found an adversarial point, and both are counted as failures. No other batch of any arm reached this cap. The slowest ATLAS batch took 37 min, and no NES or SignHunter batch exceeded 2 min. This illustrates the computational inefficiency of BayesOpt in this setting.

As another example, take every row in the "(Adv.) ImageNet" section displaying adversarial sets for a targeted goldfish classification. The images that ATLAS recovers contains both visual features that you'd expect to see in a goldfish image - orange/gold coloration, strong high-contrast colors, curved/elongated flowing structures, structures that vaguely resemble fins - and visual features that you would not expect in a typical goldfish image - structures that resembles flowers or petals, feathers, bird parts, etc. This suggests that the model may accept non-goldfish structures as reasonable substitutes when enough lower-level or mid-level goldfish visual cues are present. For example, *an image with a flower petal or feather may be accepted as an image substitute because it provides a structure that is visually similar to a fin without actually being a fin*. This sort of observation can be used as a guideline for producing more sets of adversarial image failures for this particular model.

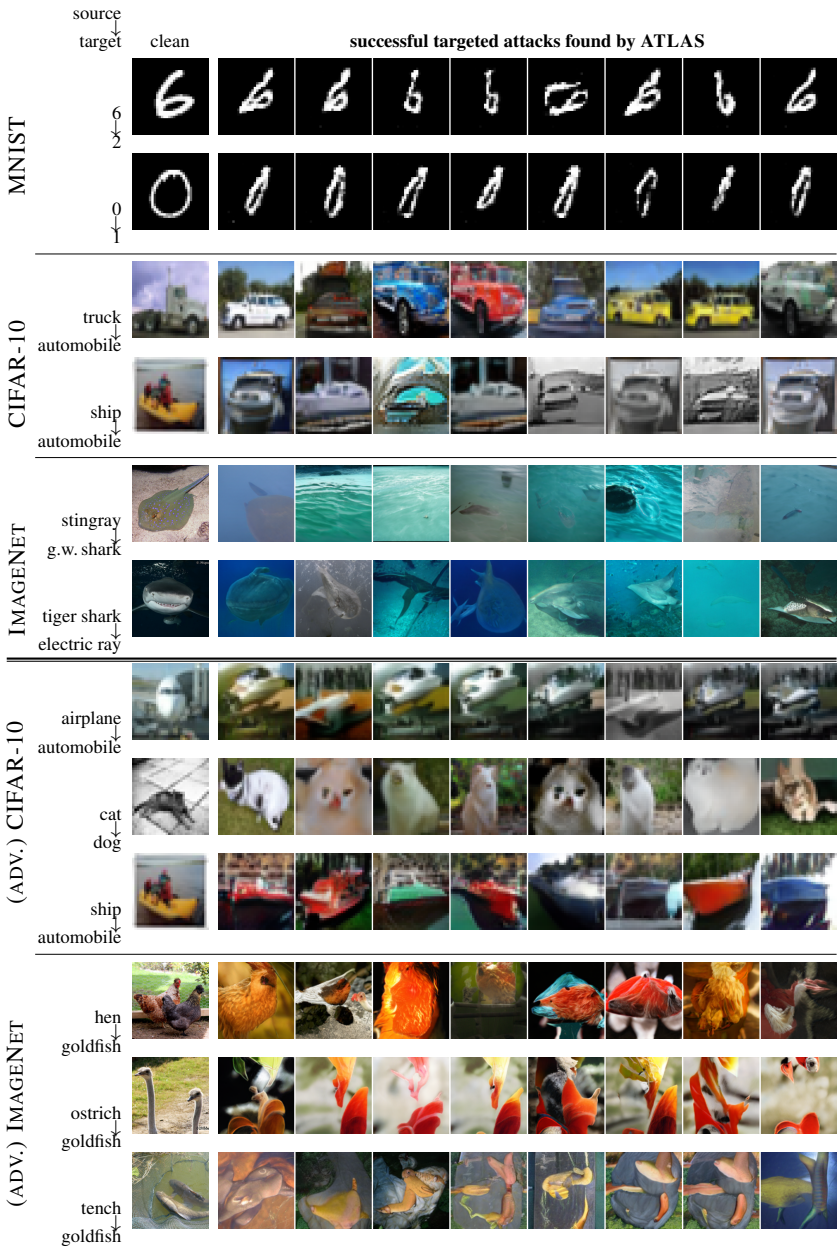


Figure 5: Representative adversarial examples discovered by ATLAS under a 650-query budget. Each row specifies a source and target class. The leftmost image is a clean source-class sample, followed by representative samples from the generated adversarial set. The examples for MNIST, CIFAR, and (adv.) CIFAR contains adversarial images which could present as benign but that gets misclassified by the model. The examples for ImageNet and (adv.) ImageNet ALL contain distinct visual cues which pushes that model's decision boundary towards that specific class. For the great white (g.w.) shark, the model is most likely to key in on the ocean (a shark's environment) rather than a shark's features for its classification. For goldfish, this model accepts non-goldfish structures as reasonable substitutes (a flower petal) when enough lower-level or mid-level goldfish visual cues are present. See Section E for details.

F Security Audit Summary

We interpret ATLAS as a model-level security auditing tool and summarize our audit findings using the NIST AI Risk Management Framework (AI RMF) (AI, 2023) and the MITRE ATLAS threat framework (MITRE Corporation, 2021). The evaluation models an attacker who can query a classi-

fier but does not require access to its weights or training infrastructure. In MITRE ATLAS terminology, the procedure generates *adversarial data* (AML.T0043) to *evade an AI model* (AML.T0015). The potential consequences of successful misclassification are context dependent and should be assessed separately.

MAP. We identify targeted adversarial attacks that represent security risks using a chosen image generator. We characterize "failure regions" within this generator's input latent space that contain latents that map to images containing the visual features of a specified source class but for which gets misclassified by the target model.

MEASURE. ATLAS measures the accessibility and size of each discovered failure region under a fixed query budget (see Table 4). Across all reported settings, ATLAS recovered nonempty sets, including against adversarially trained CIFAR-10 and ImageNet models, indicating that the adversarial training process for each adversarially trained model did not eliminate the existence of targeted failure regions.

MANAGE. The resulting adversarial sets point to model vulnerabilities that are fairly accessible for outside users. The examples within these sets should be used to guide additional robustness evaluation, adversarial training, or the creation of monitoring and security filters as potential defenses.

GOVERN. Finally, the recovered adversarial sets, ATLAS's trained surrogate model, and quantitative measurements, all encodes the state of security for each subject target, while the threat model, query budget, source-target pairs, and experimental settings provide documentation for how to repeat this security analysis after the model has been robustified with better security defenses. Each of these resulting audit artifacts should be used to inform deployment decisions.

Ethical Considerations

This system, while designed for developers to robustify their own systems, may be leveraged by outside actors to attack other systems they do not own. While this may be the case, to mitigate the potential damage of such cases, future work includes analyzing the limitations of this system, which includes finding the functional characteristics (defenses) for which similar algorithms are unable to efficiently map inputs their associated adversarial scores. Other possible mitigations include limiting query access to outside systems.

AI Statement

AI was used to help run smaller debugging experiments, analysis, and sweeps for parameter settings; help set up experimental scripts; help create figures and assist with writing revising; and give feedback on the authors methodological ideas and data interpretations. All core experimental, methodological, and analytical designs were done by the authors, along with their scripts; however, bug editing and enhancements for parallelization was assisted with AI. Core analysis was done by the authors from logs and data generated by the authors with assistance from AI. All code and any artifacts generated by AI were looked over and edited by the authors.

Open Science

Complete hyperparameters for ATLAS and all baselines, including shared, matched, and method-specific settings, are given in Table 5; experimental protocols are in Appendices A and D. All source code, configurations, and scripts to reproduce every reported result will be released publicly upon publication.