# Adversarial Transfer Attacks With Unknown Data and Class Overlap

Luke E. Richards
richards_luke@bah.com
Booz Allen Hamilton
Univ. of MD, Baltimore County
USA

André Nguyen
nguyen_andre@bah.com
Booz Allen Hamilton
Univ. of MD, Baltimore County
USA

Ryan Capps
capps_ryan@bah.com
Booz Allen Hamilton
USA

Steven Forsythe
sforsyth@nvidia.com
NVIDIA
USA

Cynthia Matuszek
cmat@umbc.edu
Univ. of MD, Baltimore County
USA

Edward Raff
raff_edward@bah.com
Booz Allen Hamilton
Univ. of MD, Baltimore County
USA

## ABSTRACT

The ability to transfer adversarial attacks from one model (the surrogate) to another model (the victim) has been an issue of concern within the machine learning (ML) community. The ability to successfully evade unseen models represents an uncomfortable level of ease toward implementing attacks. In this work we note that as studied, current transfer attack research has an unrealistic advantage for the attacker: the attacker has the exact same training data as the victim. We present the first study of transferring adversarial attacks focusing on the data available to attacker and victim under imperfect settings without querying the victim, where there is some variable level of overlap in the exact data used or in the classes learned by each model. This threat model is relevant to applications in medicine, malware, and others. Under this new threat model attack success rate is not correlated with data or class overlap in the way one would expect, and varies with dataset. This makes it difficult for attacker and defender to reason about each other and contributes to the broader study of model robustness and security. We remedy this by developing a masked version of Projected Gradient Descent that simulates class disparity, which enables the attacker to reliably estimate a lower-bound on their attack's success.

## CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Security and privacy**; • **Theory of computation** → *Adversarial learning*;

## KEYWORDS

adversarial machine learning, machine learning, machine learning security

## 1 INTRODUCTION

Given an input $x \in \mathbb{R}^d$ and an associated label $y \in \mathcal{Y}$, we wish to learn a model $V(\cdot)$ such that $V(x) = y$. This is achieved using gradient descent on a loss function $\ell(\cdot, \cdot)$, and trained over a dataset $\mathcal{D}^n$ of $n$ samples to minimize the function $\sum_{\forall x, y \in \mathcal{D}} \ell(V(x), y)$.

In this work, we are concerned with untargeted adversarial attacks against the model $V(\cdot)$. For *evasion attacks* specifically, an adversary $A$ will alter $x$ to create a new perturbed sample $\tilde{x} = A(x)$. This perturbed $\tilde{x}$ is similar enough to $x$ such that they *should* have the same label, but instead the victim $V(\cdot)$ is fooled so that $V(\tilde{x}) \neq y$. These attacks are surprisingly easy to accomplish. Just as our model $V(\cdot)$ was trained by taking the gradient of the loss with respect to model parameters $\nabla_\theta \ell(V(x), y)$, the adversary can create $\tilde{x}$ by taking the gradient of the loss with respect to the input and attempting to maximize the loss.

In the ideal circumstance, we would be able to defend $V(\cdot)$ in the white box scenario, in which the adversary has perfect knowledge of the model $V(\cdot)$, including its parameters, and the training data $\mathcal{D}$. Another common scenario is the gray-box attack, where the adversary $A$ has imperfect knowledge, and wishes to perform a *transfer* attack.

In the transfer attack scenario the adversary $A$ does not know what model the victim $V(\cdot)$ is using or its parameters, and so they construct a surrogate model $S(\cdot)$. This surrogate can have a different architecture, but is trained in the same manner: $\sum_{\forall x, y \in \mathcal{D}} \ell(S(x), y)$. This gives the adversary their own model $S(\cdot)$, which they attack to produce an adversarial sample $\tilde{x}$. The hope, which has held in practice, is that attacks against the surrogate $S(\cdot)$ generalize to the victim $V(\cdot)$ and successful fool it so that $V(\tilde{x}) \neq y$. This is how most works have proceeded in the study of this problem, which allows both the surrogate and victim to share the knowledge of the exact same label space (i.e., task) and data, as outlined in Figure 1.
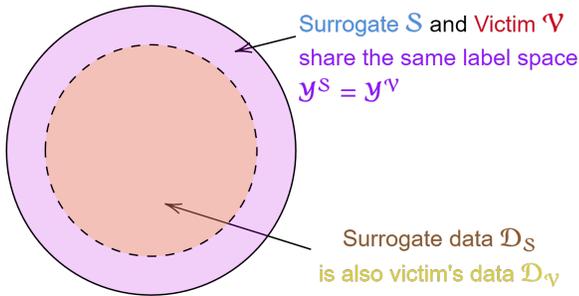
Figure 1: Solid circles show the label space, which in purple shows that the victim and surrogate use the exact same label space (perfect intersection). Dashed circles show the data used in training, which again is a perfect union (orange). This is the standard world view explored in prior works.
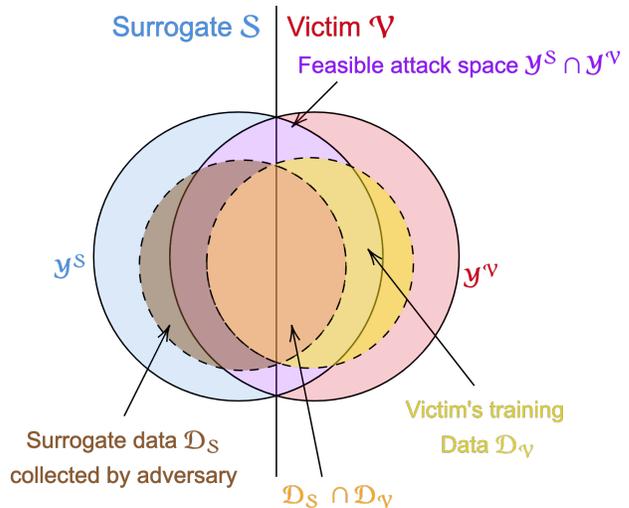


Figure 2: In a transfer attack where the adversary attacks their own surrogate model $S(\cdot)$ in order to evade a victim's model $V(\cdot)$. We study two factors that impact the attack success. The overlap between the data $\mathcal{D}_S$ used to train $S(\cdot)$ and $\mathcal{D}_V$ used to train $V(\cdot)$ (dashed lines) exists in the sub-space of the overlap in label output spaces (solid lines) $S(\cdot) :\to \mathcal{Y}^S$ and $V(\cdot) :\to \mathcal{Y}^V$. These two factors may vary in their degree of overlap, and data may have no overlap.

The contribution of our work is the recognition of an inconsistency in this threat model: in what realistic circumstance does the adversary $A$ know *nothing* about the victim's model $V$, yet has access to the same set of class labels $\mathcal{Y}$ and in non-query scenarios even the *exact* training dataset $\mathcal{D}$? We believe this to be an unrealistic threat model for many real-world scenarios, yet it is the predominant version of the transferable attack scenario studied [12, 13, 30, 34, 42]. As companies develop and label their own corpora, we should expect that the adversary is likely to know *less* about the data than they would the model type.[1] Not all ML models will be exposed to external entities, and may be used entirely for internal purposes within a company. This shared conditioning of the surrogate and victim being trained on the exact same data makes for an unrealistic evaluation.

Instead, we study a new version of the transfer attack scenario summarized in Figure 2. For simplicity we will allow the surrogate $S$ and victim $V$ to use the same architecture when using different datasets $\mathcal{D}_S$ and $\mathcal{D}_V$ each with $n$ samples. Since it is common practice to still use publicly available data, we will study scenarios where ranging from no overlap ($\mathcal{D}_S \cap \mathcal{D}_V = \emptyset$), partial overlap ($|\mathcal{D}_S \cap \mathcal{D}_V| < n$), and complete overlap ($\mathcal{D}_S = \mathcal{D}_V$). In addition we note that companies may not fully expose all of the classes they might detect, so we will also study how the degree of overlap between different output spaces $\mathcal{Y}^S$ and $\mathcal{Y}^V$ in the same fashion. In doing so, we discover that the lack of shared underlying data can significantly reduce the success of transfer attacks from the surrogate $S(\cdot)$ to the victim $V(\cdot)$ in unexpected ways, as it does not correlate with the degree of overlap in the way one may intuitively expect. This makes it difficult for the adversary to determine the likelihood an attack succeeds, and thus to plan said attack. We also show how state-of-the-art adversarial training can actually help improve the attacker's success rate due to a kind of over-fitting. To restore behaviors to an easier to understand and interpret set of patterns, we develop an augmented "masked surrogate" attack that the adversary can use which removes the variance caused by

imperfect class overlap, making it possible for $A$ to again attack victim $V(\cdot)$ with confidence.

To help motivate this problem, consider the example of pain in a medical context. A hospital using a machine learning model to try and detect different types and levels of pain from facial expressions can be useful for triage and patient care[3, 8, 19, 26, 28]. It would not be queryable by outsiders to attack, but patients engaging in drug-seeking behavior[2] would desire to attack such a system [18, 32]. An adversary could realistically create their own dataset, but because there are many different medical taxonomies and definitions around pain [43], they would not likely know the exact set of labels used by their victim.

The rest of our paper is organized as follows. In section 2 we will discuss related work which has used the unrealistic version of our threat model. In section 3 we contribute a rigorous threat model for this unrecognized, and realistic scenario, and a new masked PGD attack for the threat model. In section 4 and section 5 we contribute an extensive empirical study of the threat model showing that current adversarial methods and intuition do not hold under uncertain class and data overlap, making attacks unreliable. Our new masked PGD will restore the attacker to an intuitive behavior that allows effective attack against $V(\cdot)$. Finally we will conclude in section 7.

---

[1]Many companies will publish and discuss the types of models they use and develop, but they rarely share their labeled data.

[2]For their own use or illegal re-sale. The later group potentially well funded and motivated.

## 2 RELATED WORK

While adversarial attacks on machine learning algorithms have received increased attention in the last few years, they have been a research area for over a decade [7] and have proven to be challenging to defend against [9]. The threat of adversarial attacks applies broadly to many classification use cases, for example malware detection [24] and facial recognition [38], as well as to regression tasks [29]. In this work, we are interested in the transfer of adversarial attacks. There are two larger groups that transfer attacks have fallen into based on whether they use queries against the victim $V(\cdot)$ or not. Our work extends the study of threat models that do not query the victim.

Many have looked at the transferability of adversarial attacks that do not rely on querying the victim model [14, 15, 22, 34, 42, 45, 46] and built theory of how to improve transfer success rate [12]. All of these cited works assume, or do not discuss, the adversary having perfect knowledge of the victim's classes (i.e., $\mathcal{Y}^{\mathcal{S}} = \mathcal{Y}^{\mathcal{V}}$). The few works that have recognized that shared data ($\mathcal{D}_{\mathcal{S}} = \mathcal{D}_V$) is an unrealistic assumption have not studied how the degree of data overlap may impact transfer success, and instead replace the assumption with the case that the adversary knows how much of their data overlaps. Prior work [11, 14, 16] modified gradient-based attacks to optimize transferability between models without examining data overlap. These works used mostly 10-100 attack iterations, and found that more iterations reduced the impact of their results. When using 250 to 1,000 as we do in our work, we found these approaches had no impact on results (e.g., see Figure 8).

Our work contributes the realization that unknown class and data overlap is a likely scenario, and that if you can't query your victim you may not have certainty about class overlap either. Some have intentionally made sure there is no data overlap, but not tested the impact of no or partial overlap [40]. As far as we are aware, we are the first to study the impact of class overlap and the first to study how the degree of data overlap impacts transfer success, and the interaction of these two factors.

Many black-box attacks query the victim [12, 30, 31] with new data points to get them labeled. This means that each label pair in the surrogate's corpus $x, y \in \mathcal{D}_{\mathcal{S}}$ is really $x, V(x)$. While the attacker is limited based on the number of queries considered, they are implicitly gaining information about the entire training set $\mathcal{D}_V$ that was used to create $V(\cdot)$. The query based black-box attacks are realistic but do not cover all circumstances (e.g., models used internally or which do not provide immediate/direct feedback).

One of the only works we are aware of the recognize the adversary may have varied degrees of knowledge or overlap with $\mathcal{D}_V$ was by [41]. Their study however was focused on only poisoning based attacks, and did not explore the broader impact of the degree of overlap $|\mathcal{D}_{\mathcal{S}} \cap \mathcal{D}_V|/\max(|\mathcal{D}_{\mathcal{S}}|, |\mathcal{D}_V|)$. Instead they choose to allow the adversary to know 70% of the training corpus or 100%. Instead we will test a spectrum of overlaps of training data and class labels.

To the best of our knowledge, no work has yet considered that the adversary may not know the exact set of classes $\mathcal{Y}^{\mathcal{V}}$ used by the victim, and so must use an overlapping but different set of labels $\mathcal{Y}^{\mathcal{S}}$ for their surrogate though the scenario is realistic. Determining what family of malware (i.e., the class) a given executable belongs to is important functionality, yet all anti-virus products have wildly different sets of labels they recognize with different definitions, which are constantly changing as new families are detected [35–37]. The malware example is particularly motivating because it involves a real-live adversary (the malware author) who actively wants to avoid detection. Our model of unknown class overlap is also common to industry practice. Many companies will have alpha/beta tests or "phantom" deployments of hidden features for testing, which could include new class labels that have not been exposed. Even under normal circumstances, companies do not always disclose all classes they recognize or precisely how many exist.[3] Real-world production models can have millions and even 100 million classes in use [10, 21], making it impractical for an adversary to likely model all available classes. While we do not test to this extreme, we believe this provides further evidence that assuming a perfect model of the output space $\mathcal{Y}^{\mathcal{V}}$ is not always realistic, and thus worthy of our study.

## 3 APPROACH

As recommended by [6] we first make explicit the threat model that defines our adversary's goals, capabilities, and knowledge. We assume there is a victim model $V(\cdot)$ that they wish to provide altered inputs $\tilde{x} = A(x)$ such that $V(\tilde{x})$ will produce incorrect predictions. The adversary is able to send altered inputs $\tilde{x}$ to the victim, but is not able to observe a direct or indirect response. The adversary does not know the training data $\mathcal{D}_V$ used by the victim, or the specific classes– it is only inferred guesses or availability that they should overlap to an unknown degree (as we have established in section 2, this is realistic for cases of malware, medical, and other applications). The adversary has the capability to perturb inputs by up to $\epsilon$ under the $\infty$ norm, such that $\|\tilde{x} - x\|_{\infty} \le \epsilon$ by attacking their own surrogate model $S(\cdot)$ trained on an independently built corpus $\mathcal{D}_{\mathcal{S}}$ with an independent label set $\mathcal{Y}^{\mathcal{S}}$. Pessimistically we assume the adversary's surrogate model $S(\cdot)$ uses the exact same architecture as the victim's $V(\cdot)$ (given wide re-use of "backbone" networks such as ResNet, this is a reasonable assumption). The adversary needs to meet a minimum attack success rate of $p\%$ for their attack to be viable, and only has one chance to launch their attack campaign[4].

Our restrictions on the adversary's knowledge of the data and classes may seem extreme. This requires the adversary to invest time into understanding their victim, building their own corpus and expected label set, and then producing perturbed inputs to send to the victim. This scenario is realistic to many situations, and occurs in biometric recognition [2], medicine [18, 32], and malware detection [1, 17, 33] are all domains where the specific class labels may vary between users but have predictable high-level goals, and data is tightly held (for privacy or competitive advantage), and models are often kept private or not fully exposed to users/consumers.

---

[3]e.g., Clarifai's current documentation gives a lower bound and no comprehensive list for the number of classes available for many of their models. This behavior is relatively common amongst commerically available APIs.

[4]The assumption being a failed attack alerts the defender, who can thus make changes or react in some unfavorable way, that is ultimately a detriment to the attacker's goals.

Within the threat model, we assume that there are varying levels of intersection between the data points of $\mathcal{D}_S$ and $\mathcal{D}_V$. Within this analysis we make splits based on a number of shared classes for image classification $\mathcal{Y} = y_1, y_2, ..., y_N$ where each model is trained using classes a subset of $\mathcal{Y}$, the surrogate having $\mathcal{Y}^S$ and the victim $\mathcal{Y}^V$. Where the shared classes of $\mathcal{Y}^S \cap \mathcal{Y}^V = \mathcal{Y}^O$ represents the overlap in shared classes. We observe various sizes of intersection/overlap $|\mathcal{Y}^O|$ within the work respective to the size of $|\mathcal{Y}|$. Because we test with varying levels of class overlap, each model ($V(\cdot)$ and $S(\cdot)$) will have half as many output classes. For example, CIFAR-100 tests would only have 50 classes so that we can vary the number of shared classes from 2 to 100% without changing the number of classes.

Within that intersection of shared classes, we split the intersection further by dividing the data points of classes $\mathcal{Y}^O$ based on a varying number of data partitions such that the data points $\mathcal{D}_{y^O}$ into subsets such that $\mathcal{D}_S \neq \mathcal{D}_V$. Thus the surrogate model is trained on $\mathcal{D}_S = \mathcal{D}_{y^S} \cup \mathcal{D}_{y^O}$ and the victim model is trained on $\mathcal{D}_V = \mathcal{D}_{y^V} \cup \mathcal{D}_{y^O}$. We analyze where there is no overlap in shared data points but shared classes exist $\mathcal{D}_S \cap \mathcal{D}_V = \emptyset$, the case in which there is partial overlap of data points $|\mathcal{D}_S \cap \mathcal{D}_V| < n$, and finally the case in which there it total overlap $\mathcal{D}_S = \mathcal{D}_V$.

To better study the impact of $|\mathcal{D}_{y^O}|/\max(|\mathcal{D}_S|, |\mathcal{D}_V|)$ the shared data ratio between the surrogate $S(\cdot)$ and victim $V(\cdot)$, we will not use all of the original training data. Instead, each model will be trained only on half of the original samples (e.g., using 30,000 out of the 60,000 training samples of MNIST). This is done so that we can test a shared class overlap of $\{0, 25, 50, 75, 100\}\%$ while keeping the total training size constant, eliminating it as a factor of variation. For example, 0% overlap (i.e., $|\mathcal{D}_{y^O}|/\max(|\mathcal{D}_S|, |\mathcal{D}_V|) = 0$), would mean all of the original data is used, half by the surrogate $S$ and the other half by the victim $V$. For 100% overlap only half of the original data is used by both $S(\cdot)$ and $V(\cdot)$. This way all models trained have a corpus of the same total size. A consequence of this design choice is our results are not directly comparable with prior studies on the same datasets, since our models are trained on only half as much data.

Once both models are trained on their respective datasets, we find an adversarial attack $A$ in which the input $x$ is perturbed to be $\tilde{x} = A(x)$ for the surrogate model $S(\cdot)$. We then transfer this attack $A$ to the victim model $V(\cdot)$, which $A$ is not optimized for. The success of such a transfer attack is measured by the original accuracy of the victim model on the shared held out test data $\mathcal{D}_{y^O}$ subtracted from the new accuracy after $A$ perturbs $\mathcal{D}_{y^O}$, $A(\mathcal{D}_{y^O})$.

$$\text{success} = \text{accuracy}(V(\cdot), \mathcal{D}_{y^O}) - \text{accuracy}(V(\cdot), A(\mathcal{D}_{y^O})) \quad (1)$$

This metric allows us to examine how dependent the adversarial attack $A$ is on the data of the surrogate $\mathcal{D}_S$. Common thought from the machine learning community would assume the closer the overlap in data distribution there is between $\mathcal{D}_S$ and $\mathcal{D}_V$ the more successful the attack will be. We examine how well this initial hypothesis holds with empirical evidence in the following section.

## 3.1 Impact of Adversarial Training

We examine a variation of the threat model in which both the surrogate model and the victim model are hardened to adversarial attacks during training. Each model then is exposed to adversary examples $\tilde{x}_S$ generated by $A_S$ for model $S(\cdot)$ to have $S(\tilde{x}_S) \neq y$. The same process is used for $V(\cdot)$ to emulate the victim also attempting to safeguard the model. Under normal study this results in improved accuracy under attack, at some variable decrease in accuracy on clean samples. Our results will show this behavior does not hold under our new threat model.

## 3.2 Removing Class Uncertainty

Finally, to explore the dependence of classes being present within the surrogate model $\mathcal{Y}^S$, we introduce a modified PGD attack, Algorithm 1, that runs $T$ additional iterations with a randomly generated mask ($M$) over the logits of the class output. Randomly masking classes makes it such that the adversary cannot rely on any "easy" classes (e.g., two different breeds of dog) to obtain successful attacks. This operation is in a manner mechanically similar to Dropout [39] and Expectation over Transformation (EoT) [5]. Unlike Dropout our goal is to make the attacker more robust, and unlike EoT we are not differentiating through a defence of the victim (EoT's original purpose), but instead circumventing the adversary's uncertainty about the victim through simulation. We then compute the delta $\delta$ per the PGD attack $T$ times and then average it to have a more generalized delta, $\delta_{avg}$, that will not be dependent on $\mathcal{Y}^S$. We believe this to be a better strategy for attacking the victim model with no guarantee of any of the surrogate model classes $\mathcal{Y}^S$ being present in victim's $\mathcal{Y}^V$.

---

**Algorithm 1:** Masked PGD Attack: N number of iterations for PGD, S number of examples in dataset, T number of mask iterations

---

**for** $n = 1...S$ **do** // Samples to attack
  **for** $i = 1...N$ **do** // PGD iterations
    $\delta_{avg} = 0$
    **for** $j = 1...T$ **do** // Mask iterations
      $M = $ random mask ;    // Sample a mask for logits
      $x_p = f_\theta(x) \cdot M$ ; // Apply mask, M to model output
      $\delta = \delta + \alpha \cdot \text{sign}(\nabla_\delta \ell(x_p + \delta, y))$
      $\delta_{avg} = \delta_{avg} + \max(\min(\delta, \epsilon), -\epsilon)$
  $\delta_{avg} = \delta_{avg}/T$

---

## 3.3 Computational Considerations

A single model on CIFAR-100 takes on average 1.25 hours to train on a single graphics processing unit (GPU). For each experiment we must train the surrogate model and the victim model from scratch to preserve control for data overlap. Each with a change in the number of classes shared and data, for CIFAR-100 looking at 5 number of classes each at 5 different percentages of data shared creates 25 trains of the surrogate model and then 25 trains for the victim model. This totals in 50 total trained models. This totals to 62.5

hours for each training experiment. Due to the random selection of the shared classes and shared data, we run each experiment 5 times to account for outliers and average the performance. This increases to 312.5 GPU hours in training. This is not including the total time it takes to find and run the attack on the victim network, which in the next section we choose to take an extremely high number of iterations to properly capture the potential of the attack for our threat model. The attack cost for our masked experiments take another $\approx 3,700$ GPU hours due to the iterative masking process. The results presented constitute months of our entire compute capacity.

Performing the same study, with multiple runs to average the behavior, would take an estimated $\approx 40,000$ GPU hours ($\approx 4.5$ years), is simply beyond our resources and financial means. We have results for a reduced and single-run (no averaging) result on ImageNet in section 6 as a spot check, which supports our overall conclusions derived from the more tractable CIFAR and Mini-ImageNet datasets.

## 4 EXPERIMENTS

We examine the success rate of transferring an attack to a victim model $V(\cdot)$ from a surrogate model $S(\cdot)$ with varying levels of shared tasks/data. We define tasks as shared-classes and data as the percentage of images within that intersection of shared-classes that both models are trained on. The training data is split based on the number of shared classes leaving the non-intersecting classes to be split between the two models. From the shared class data, we then split based on a percentage shared. We then train two networks on each of these subsets.

Within our extended experiments, we focus on Mini-ImageNet since this dataset is larger, in terms of both classes and data, and diverse in the type of images and classes. We find this to be a reasonable size for our extensive computational costs of experiments compared to ImageNet (see subsection 3.3). Two variations are tested, one in which both surrogate and victim models are adversarially hardened during training and a novel variation of projected gradient descent (PGD) attack in which we mask classes to increase generalization of transfer.

In summary we conduct the following experiments:

- An analysis of success of transfer from surrogate model to victim model at various levels of shared classes and data points.
- Analysis of the effects adversarial hardening during training of both models has on the success based on varying levels of shared classes and data points.
- A novel adversarial attack meant to generalize without relying on a single class present in the surrogate model using masking based targets.

### 4.1 Standard Adversarial Transfer Attack

We use ResNet architectures [20], a ubiquitous backbone architecture, for all experiments. We choose to match the size of model to the size of dataset being learned. We used ResNet-18 for Fashion-MNIST, ResNet-50 for CIFAR-10, and Wide ResNet-50 for CIFAR-100. Each network is then modified to have a final linear layer that is

only the size of the number of classes per model. We used a ResNet-18 model for Mini-ImageNet for computational cost considerations.

The models are trained using Adam optimizer [23] with a learning rate of 0.0004 selected on empirical results, with varying batch sizes based on dataset: 12 for Fashion-MNIST, 12 for CIFAR-10, 128 for CIFAR-100, and 128 for mini-ImageNet. The number of epochs for each model were 10 for Fashion-MNIST, 20 for CIFAR-10, and 200 for both CIFAR-100 and mini-ImageNet. For CIFAR-100 and Mini-ImageNet we used a learning rate decay at epochs 60, 120, and 160 with gamma 0.2.

Once each model is trained on the respected data-subset a perturbation input $\hat{x} = A(x)$ is found using projected gradient descent (PGD) [25] by attacking the surrogate model $S(\cdot)$ with $L_\infty$ normalization and an $\epsilon$ of 0.3 with a PGD step size of 0.01. Following best practices we use as many PGD steps as computationally feasible [4], in our case 1,000 steps (though we found 100 sufficient in most cases). Then we apply that computed perturbation, $A$ to the victim model, $V(\cdot)$. We measure the success of the transferred attack based on varying combinations of shared-classes and percentage shared. Our results will show that having less shared data has a non-trivial impact on attack success rate, and that fewer shared classes has a large impact on the adversary's estimate of attack success rate and could lead to over/under confidence based on a computationally constrained attacker. We showcase this standard threat model with varying levels of attack with a small step size (see Figure 3). This allows us to broadly view the purpose of this experiment to examine the effects these factors have on transfer attacks. This example demonstrates in greater fidelity the change in transfer attack success rate as the percentage of shared classes and data change and interact in non-linear ways. We note that the specific nature of the pattern changes from one dataset to the next, and we do not have the computational capacity to repeat this high fidelity testing for each dataset. Our Masked PGD attack helps alleviate the uncertainty of what behavior will occur by removing the impact of the number of shared classes, and results in a consistent and smooth behavior that more shared data leads to high attack success rate (which is not true under standard PGD for all datasets, including Figure 3 itself).

### 4.2 Adversarial Hardening

Within the threat model, we also have to account for the victim model creators attempting to mitigate the success of attacks, and attackers trying to create more robust attacks [34]. We do this by performing adversarial training on both $S(\cdot)$ and $V(\cdot)$, and conducting the same grid of transfer attack testing, to determine how viable transfer attacks are when the adversary has the limited visibility of our specified threat model. Due to the number of train/test runs, we use the work of [44] for computationally efficient hardening of both networks during training. Within this method, the network is trained using a randomly initialized fast gradient sign method (FGSM) attack which was found to be as effective as the PGD attack. Our prior expectation is that this would hamper the adversary, but our results show the drop in attack success rate is smaller than we might have expected.
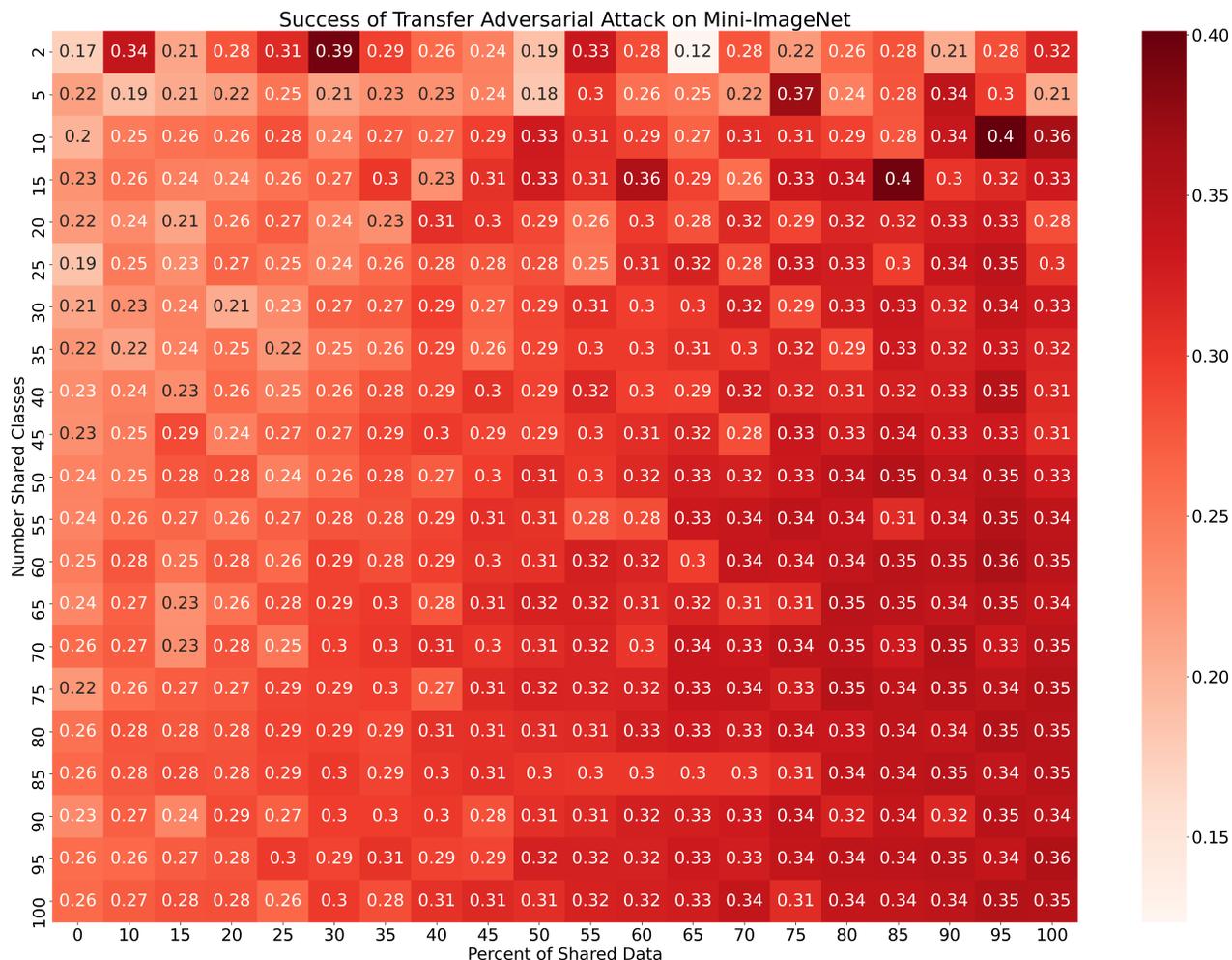
**Figure 3: This figure shows a fine-grained experiment run for a sub-sample of ImageNet using standard PGD. Examining the attack success at such scale allows greater visualization of the variance we see at a smaller scale and the relation between the two factors. With multiple runs per cell to account for the variance, this experiment took approximately 3,168 GPU hours. This dataset shows high variance when the number of shared classes $\leq 25$.**

## 4.3 Masked Transferred Attacks

The masked PGD attacks were too expensive to run to 1,000 iterations, so only 250 iterations were done instead. This number appears to still be reasonable, and was tested by running the standard attack for only 250 PGD iterations and no difference in results were observed. We note that in the masked PGD case, each iteration requires 100 sub-iterations (averaging over the random masks $M$) for an effective total of 250,000 gradient calculations. Therefore we feel this decrease is a reasonable change. We will show that this masked attack removes the number of shared classes as a source of variance, allowing the attacker to confidently estimate their attack success rate.

## 5 RESULTS

Our results shine light on behaviors outside of the intuition given the factors of shared data and classes. A priori one would expect that an attack's success should drop when class and data overlap decrease due to a lack of information, and that adversarial training would improve the victim's accuracy under attack [4, 27]. Our results show that these intuitions do not hold in practice: there is a dataset dependant pattern of correlation between class/data overlap and attack success rate, and adversarial training of the victim can counter-intuitively improve the attacker's success rate. These make it difficult for both attackers and defenders to reason about their degree of vulnerability. This adds a human-factor within the threat model of attack uncertainty minimization. Our masked PGD attack Algo. 1 restores the attacker to a more consistent behavior, removing the variability with respect to class overlap and restoring standard expectations to the attacker's success rate decreasing as the amount

of data shared with the victim decreases. Assuming an attacker would not engage in a campaign with high uncertainty, this allows the attacker (and thus, defender) to better quantify and determine their success or risk respectively. For readers viewing, we have used red for success values, blue for standard deviation, and green for differences between two methods of attack or training.

## 5.1 Standard Adversarial Transfer Attack

Looking first at the smaller dataset behaviors (CIFAR-10 and Fashion MNIST), we see an overall trend of higher success when there is more overlap in both data and classes (as originally expected). Within both however, there was an interesting dip in success with 50% shared data and 2 classes before rising again. Within CIFAR10, the model actually had more success with 0% overlap opposed to 100% overlap. Fashion-MNIST however had more consistency with success increasing as overlap increases.

For CIFAR10 (Figure 17a) the correlation coefficient between number of shared classes and success was 0.6578, for Fashion-MNIST (Figure 17b) it was 0.8213. These values point to a similar relationship, however the peak within sharing 50% of the classes in the CIAFAR10 data skewed this relationship. For the correlation coefficient for the percentage of data shared, CIFAR10 had -0.4310 and Fashion-MNIST had 0.3238. The negative coefficient is counter-intuitive going against the a priori expected trend on the small datasets, but as we look at larger datasets with more classes the behavior will regress further.
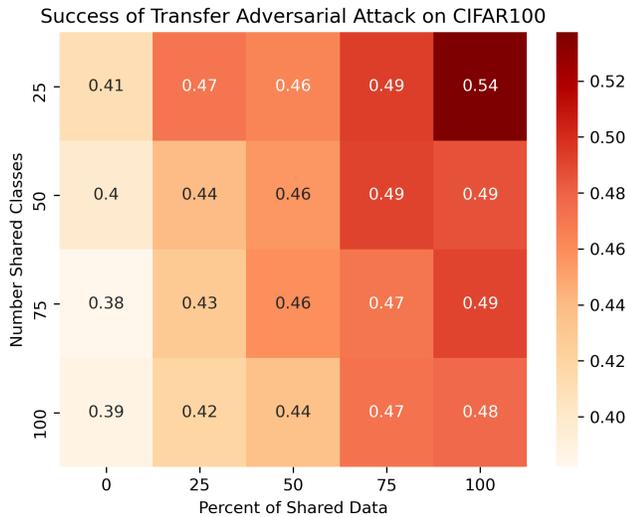


Figure 4: Success of transfer attack dependent on variations of shared classes and percentage of data shared for CIFAR-100.

CIFAR100 (Figure 4, Figure 5) showcased more nuance to the threat model, where the attacker's success improved as the number of shared classes *decreased* with an unexpected negative correlation of -0.3068, with shared data now positive at 0.8118 (in contrast to being negative for the very similar CIFAR10). Given the intermix of

adversarial attacks and out-of-distribution issues colliding, we do not yet have robust theory to explain or understand this. But the result led us to examine the variance of the success between runs for smaller class overlaps. The variance in Figure 5 showed that smaller class overlap had increasingly high variance with less overlap. The higher variance as shared classes decreases is intrinsically important to the attacker: it means they have increased uncertainty of what any individual attacks performance will be. The variance is also not large enough to explain why fewer shared classes improve the attacker's average success rate.
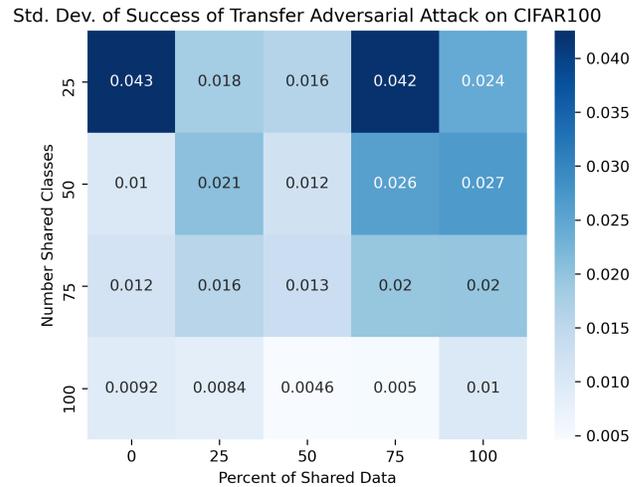


Figure 5: Standard deviation of success of transfer attack between 5 runs of each variation of shared classes and percentage of data shared for CIFAR100.

Finally we look at Mini ImageNet (Figure 6) which has 100 classes similar to CIFAR100. In this case, the data resolution is larger *(32 × 32 vs. 84 × 84)* and more diverse. At first glance it's result appear to match a priori expectations, shared classes have a positive correlation with attack success of 0.2170 and shared data a correlation of 0.8231. Again we observe the edge cases of intermediate shared percentages performing better than their extremes (e.g., 50% shared classes and 75% shared data has a 35% transfer success, higher than *all* neighboring values).

The variance of Mini-ImageNet results (see Figure 7) matches the same trend in variance from CIFAR-100: There is more variance with lower overlap in both number classes. This is the only consistent and "intuitive" behavior that occurs in this threat model, but is a pyrrhic win at best for the attacker. The higher variance may lead to greater success, but the attacker cannot rely on this to occur, and so must account for the variance in results and select from the lower tail of the distribution to make a "safe" decision about whether or not their attack will be successful.

For the results in this section, we start to gain an understanding of the behavior dynamics within this threat model. While we are unable to observe the intuition of a linear relationship for both factors, we are able to identify the nuance of variance that makes
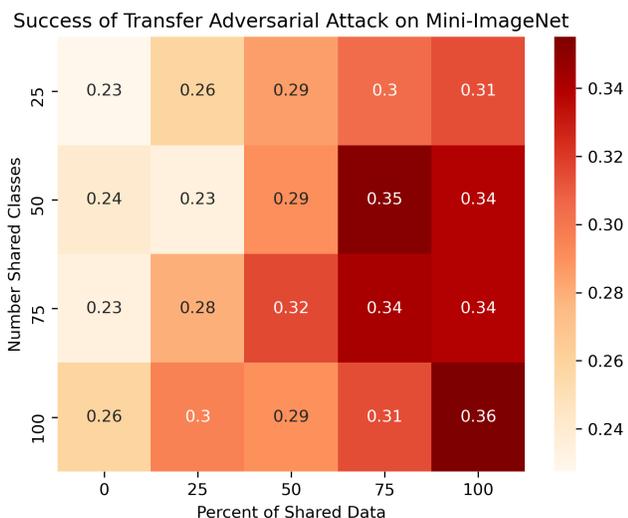
Figure 6: Success of transfer attack dependent on variations of shared classes and percentage of data shared for Mini-ImageNet.

this threat model unique and challenging. The cause of these non-intuitive behaviors is not known, and a matter of future works given the intermix of adversarial and out-of-distribution issues that blend together. Independently both issues have little existing theory, and we are aware of none that exists for the union of these problems we study.
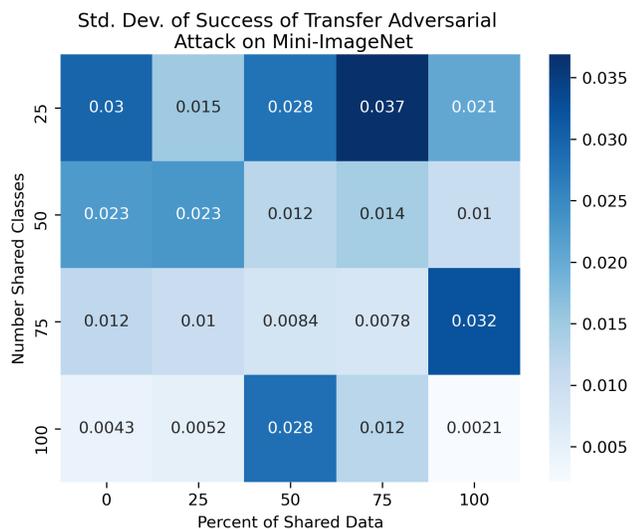


Figure 7: Standard deviation of transfer attack dependent on variations of shared classes and percentage of data shared for Mini-ImageNet.
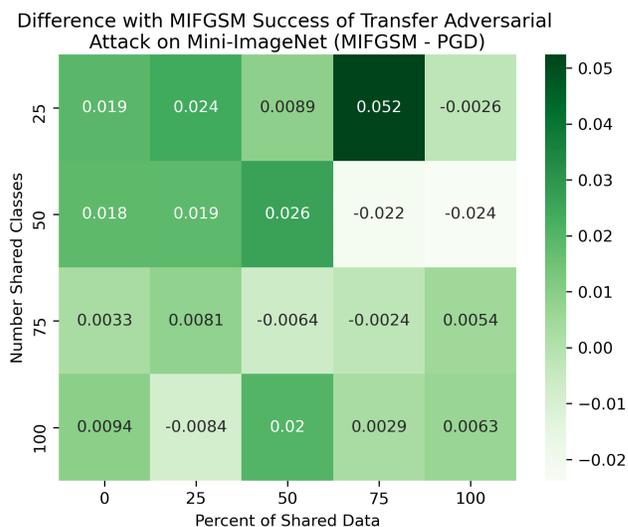


Figure 8: Difference between MI-FGSM attack and PGD attack in terms of success in attack. Lighter means PGD had more success and darker means MI-FGSM had more success.

We also examine work crafting attack objectives meant to improve transfer to a victim model. We evaluated the standard PDG attack against Momentum Iterative Fast Gradient Sign Method (MIFGSM) [14]. We found within our comparative results on mini-ImageNet (see Figure 8) that the method offered small success percentages over PGD for transfer in some cases. We defer to future work for examining transfer variants of attacks and their performance within this framework.

## 5.2 Adversarial Hardening

When training both the surrogate model and the victim model with adversarial examples included we saw two main takeaways. 1) the vulnerability of the hardened model in most cases was not as robust against attacks as the vanilla trained model. 2) we find that there is an interesting behavior profile in which the hardening is most effective with class overlap opposed to data overlap.

We find it valuable to frame the performance within the context of the difference between the vanilla model and hardened model's accuracies when under attack, as shown in Figure 9. This distinction sheds light on the trade off between robustness and accuracy that previous work faced. The correlation coefficients for these values on the Mini-ImageNet dataset for number of shared classes is -0.5537. We show both the success (Figure 14) and standard deviation between runs (Figure 10) when hardening is applied. This indicates that as the attacker and victim share fewer classes, the adversarial training becomes increasingly *detrimental* to the victim's goal of correct predictions while under attack, dropping by as much as 5.9 percentage points. The correlation with respect to shared data is only 0.0136, indicating that having the exact same data or no shared data is of comparatively little impact on the outcome.

The normal behavior is that adversarial training is purely beneficial, and usually the state-of-the-art defense against attacks (if
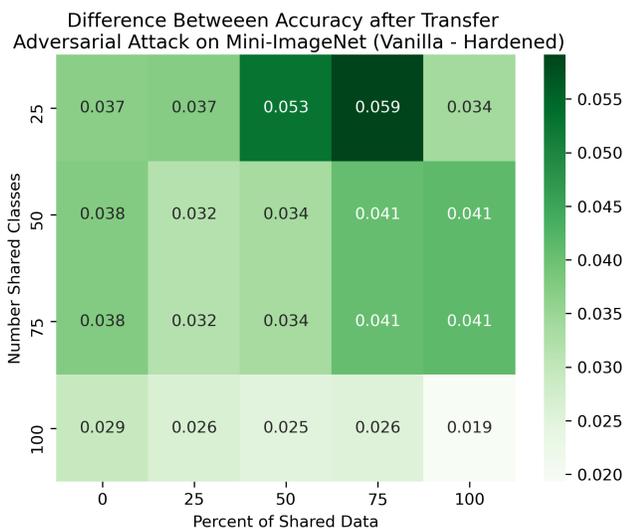
**Figure 9: The difference between vanilla and hardened models' accuracy. Overall the hardened model has lower accuracy in all cases with the least being complete overlap, the only context adversarial hardening has been studied.**
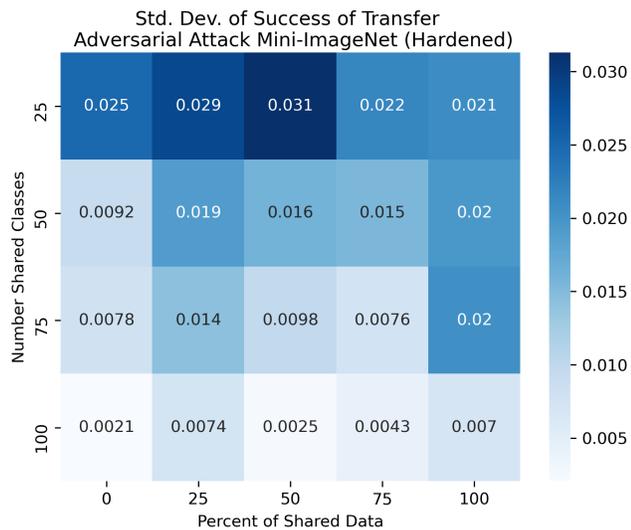


**Figure 10: Standard deviation between success of runs of transfer attack on adversarial hardened model. We again see the same level of variation as with the non-hardened model.**

within your computational constraints). However, implicit within the framework of adversarial training is that the attacker and defender have the exact same classes, the same data distribution, and thus the same manifold on which the data and attacks lie (though potentially different estimates of that manifold). Our results indicate that adversarial training can thus "overfit" to the specifics of

the current data. When attacks are generated from a highly differentiated manifold (i.e., low % shared classes) they exhibit behavior and values far outside what the defender could have generated using adversarial training, and thus, does not generalize to the attacked inputs. This places even greater importance on research into provably secure defenses for the defender's confidence, but such study is outside the scope of our current work (and compute resources).

## 5.3 Masked PGD Attack

Our results thus far have demonstrated that current intuitions and expectations on the behavior of adversarial attacks break down when there is a discordance between attacker's surrogate and the victim models. This makes reasoning for both perspectives difficult, and computationally demanding. Instead, we show that our Masked PGD attack can provide more stable estimates of attacker success that almost returns to the expected behaviors. The attack success rate is shown in Figure 12, where the variance with respect to the number of shared classes is almost entirely eliminated, and what little remains follows expectation that fewer shared classes results in a lower transfer success rate. We also see the exact same behavior profile occur on CIFAR100, as shown in Figure 11, and that the variance in results match continue to remain low when all classes are shared (Figure 13). We are unable to compare exhaustively or perform as many iterations, but we also see the same general trend on the full ImageNet corpus which is shown in section 6, which has greater variance due to the extreme computational expense which prevents us from doing more than one trial, where our CIFAR100 and Mini-ImageNet results are the average from multiple trials.
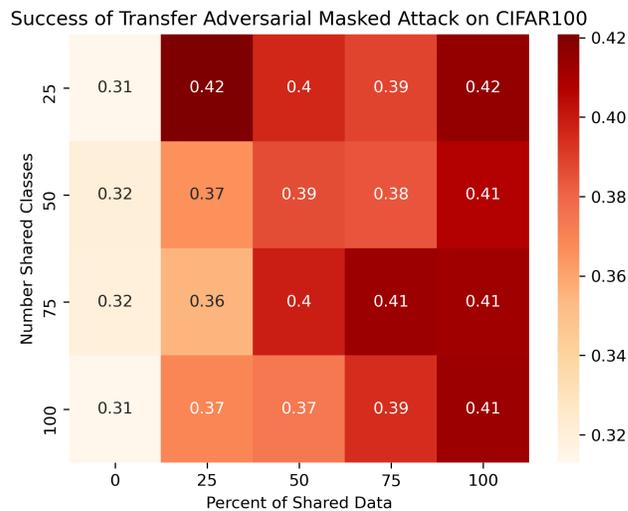


**Figure 11: Success of masked transfer attack dependent on variations of shared classes and percentage of data shared for CIFAR100.**

We note that the extreme expense for these experiments is caused by needing multiple runs of multiple difference combinations of shared class and data overlap percentages. In practice, because our Masked PGD attack restores behavior expectations, the attacker
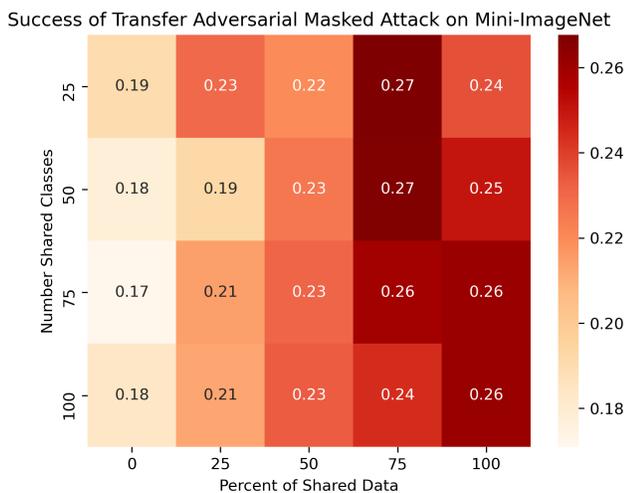
**Figure 12: Attack success with masked PGD attack dependent on variations of shared classes and percentage of data shared for Mini-ImageNet.**
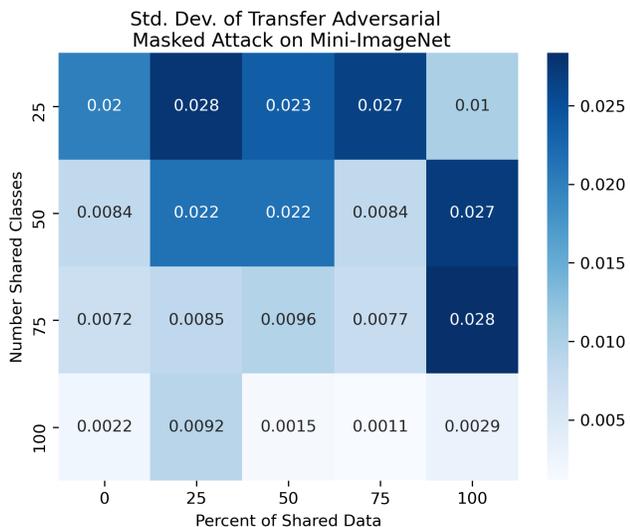


**Figure 13: Standard deviation of transfer masked-based attack dependent on variations of shared classes and percentage of data shared for Mini-ImageNet.**

or defender can simply perform the Masked PGD attack for only the case of low shared classes and 0% shared data. This provides attack and defender with a reasonable and reliable lower-bound for the attacker's probability of success without the ≥ 20× cost of evaluation for each possible combination.

Applying our Masked PGD during adversarial training is not as trivial, and in our initial experiments introduced significant decreases in stability and accuracy, as well as out-sized computational cost (the number of PGD iterations, multiplied by the number of

masked iterations). Our results thus provide a means for better estimating transfer success in our threat model, but not yet a mechanism for improved defense/robustness. This is critical for future work as our results will show that adversarial training *reduces* defender robustness in this threat model.
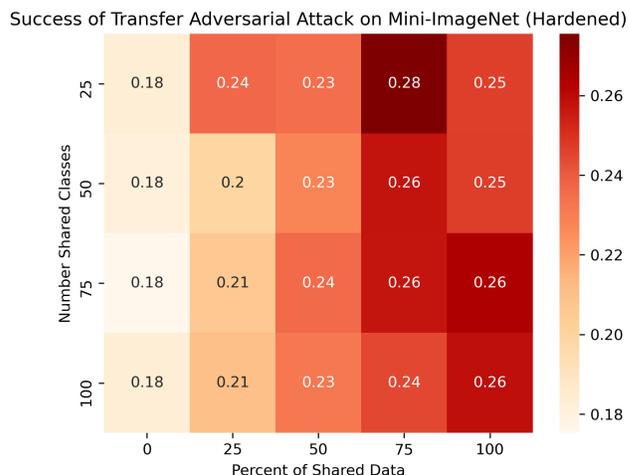


**Figure 14: Transfer success with adversarial hardening during training. Overall we saw lower accuracy and then less success. We chose to include the differences in accuracies in the paper to show that the cost of robustness only drastically assisted with the assumption of total overlap.**

## 6 INITIAL IMAGE NET RESULTS

The exploration of our results on an ImageNet scale is of extreme cost. Further understanding the results at the scale of ImageNet would also be valuable beyond the current contributions. We choose to document the initial results which are incomplete, necessarily so because they are too expensive for us to complete without external assistance. Our hope is that these incomplete results will help to illustrate the high cost for the adversary in attacking and understanding their likelihood of success as the number of classes and data increases, but also the open problem of estimating these properties for both the adversary and defender for future work.

Initial experiments were run on ImageNet with 2, 50, and 100 shared classes (2%, 5%, and 10% of all ImageNet classes) each with 0%, 25%, 50%, 75%, and 100% shared data percentages. Comparing to our experiments (Figure 15, Figure 16) with CIFAR100, we saw large variance between runs with lower number of shared classes. This may be a factor within these initial results as well as we saw the most variance at 2% of the shared classes in CIFAR100.

These experiments were run parallel on a DGX v100 server with four 32GB GPUs. With the runs for 2, 50, and 100 shared classes taking 76, 144, and 164 GPU hours respectively. The increase is due to the number of data points that are removed with cutting the number of classes to create equal training data amount per model. For the masked attack that requires more compute, the GPU hours were 396, 476, and 584.
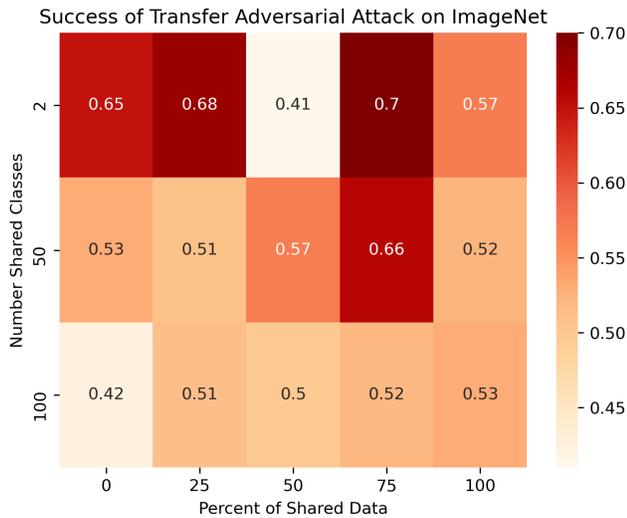
**Figure 15: Success of transfer attack dependent on variations of shared classes and percentage of data shared for partial ImageNet.**
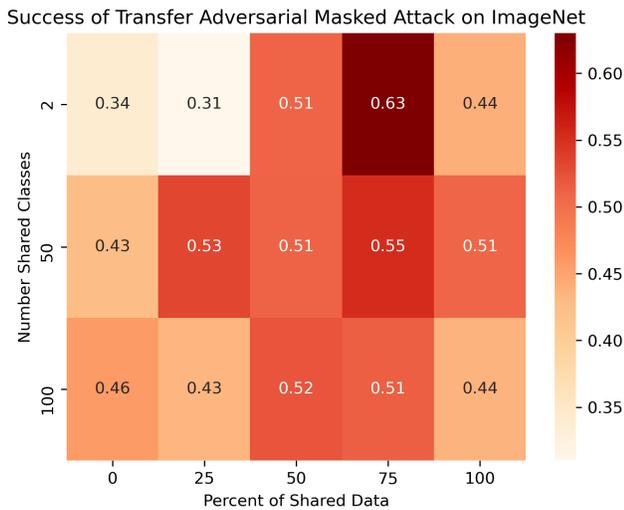


**Figure 16: Success of masked transfer attack dependent on variations of shared classes and percentage of data shared for partial ImageNet.**

These numbers account for single runs in which the variance cannot be measured like previous analysis on smaller datasets. Due to this, we can assume that each run would need to be run 5 or more times to analyze this variance. Based on the early averages, we estimate in GPU hours this would take an estimated 2,200 GPU hours (assuming linear relationship) to complete one round of 5 class step and 5 percentage step experiments. With running each iteration 5 times to measure variance, it would be 11,100 GPU hours.

With the masked version of the attack we estimate 5,980 GPU hours based on early results with a total of 29,900 GPU hours to analyze variance between runs.

We foresee scaling these experiments on a large dataset such as ImageNet to be a considerable engineering contribution in itself. Crucially, introducing the security analysis paradigm of the intersection between shared classes and shared data points itself shows promising results on smaller datasets.

## 7 CONCLUSION

We have elucidated a data-centric threat model where the adversary cannot query the victim, but must still transfer evasion attacks using a surrogate. Many prior works have implicitly used this model, with the unrealistic setting of the adversary knowing all the exact data used by the victim. This model is applicable to many real-world situations, and we have shown for the first time the asperity of attack success rate as the amount of shared data and classes vary. Attack success rate and correlations change sign depending on dataset used, and adversarial training can actually reduce defender success, making it difficult for attacker and defender to estimate overall success. Our new Masked PGD attack removes the variance with respect to shared classes, and restores an intuitive behavior to results across all tested datasets. This allows the attacker to confidently estimate attack success rate.

Considerable work still remains in fully and more accurately modeling real-world threat models under imperfect information and observation. In particular our work has taken a first step by sub-sampling existing classes, but more complex partial intersection of victim vs surrogate classes is likely in real world scenarios. The large computational cost to study these scenarios due to the intrinsically stochastic nature of class overlap also needs addressing. In particular this impacts the feasibility of adversarial training, which loses significant utility in our testing.

## REFERENCES
[1] Hyrum S. Anderson and Phil Roth. EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. *ArXiv e-prints*, 2018.
[2] Bilgehan Arslan, Mehtap Ulker, and Seref Sagiroglu. Machine Learning Methods Used in Evaluations of Secure Biometric System Components. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 448–453. IEEE, 12 2017.
[3] Ahmed Bilal Ashraf, Simon Lucey, Jeffrey F. Cohn, Tsuhan Chen, Zara Ambadar, Kenneth M. Prkachin, and Patricia E. Solomon. The painful face – Pain expression recognition using active appearance models. *Image and Vision Computing*, 27(12):1788–1796, 11 2009.
[4] 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 (ICML)*, 2018.
[5] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing Robust Adversarial Examples. *ArXiv e-prints*, 2017.
[6] Battista Biggio, Giorgio Fumera, and Fabio Roli. Security evaluation of pattern classifiers under attack. *IEEE Transactions on Knowledge and Data Engineering*, 26(4):984–996, 2014.
[7] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 12 2018.
[8] Sofia Broome, Karina Bech Gleerup, Pia Haubro Andersen, and Hedvig Kjellstrom. Dynamics Are Important for the Recognition of Equine Pain in Video. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12659–12668. IEEE, 6 2019.
[9] Nicholas Carlini and David Wagner. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pages 3–14, New York, NY, USA, 2017. ACM.
[10] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. Taming Pretrained Transformers for Extreme Multi-Label Text Classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge*

*Discovery &amp; Data Mining*, KDD '20, page 3163–3171, New York, NY, USA, 2020. Association for Computing Machinery.

[11] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. *arXiv preprint arXiv:1906.06919*, 2019.

[12] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks. *ArXiv e-prints*, pages 26–28, 2018.

[13] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 321–338, Santa Clara, CA, 8 2019. {USENIX} Association.

[14] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[15] Y Dong, T Pang, H Su, and J Zhu. Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4307–4316, 6 2019.

[16] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

[17] Manuel Egele, T Scholte, E Kirda, and Santa Barbara. A Survey On Automated Dynamic Malware Analysis Evasion and Counter-Evasion. In *Proceedings of Reversing and Offensive-oriented Trends Symposium*, 2017.

[18] Samuel G Finlayson, Isaac S Kohane, and Andrew L Beam. Adversarial Attacks Against Medical Deep Learning Systems. 2018.

[19] Zakia Hammal and Jeffrey F. Cohn. Automatic detection of pain intensity. In *Proceedings of the 14th ACM international conference on Multimodal interaction - ICMI '12*, page 47, New York, New York, USA, 2012. ACM Press.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. Slice: Scalable Linear Extreme Classifiers Trained on 100 Million Labels for Related Searches. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 528–536, New York, NY, USA, 2019. Association for Computing Machinery.

[22] Sanjay Kariyappa and Moinuddin K. Qureshi. Improving Adversarial Robustness of Ensembles with Diversity Training. *arXiv*, 2019.

[23] Diederik P Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *International Conference On Learning Representations*, 2015.

[24] Bojan Kolosnjaji, Ambra Demontis, Battista Biggio, Davide Maiorca, Giorgio Giacinto, Claudia Eckert, and Fabio Roli. Adversarial malware binaries: Evading deep learning for malware detection in executables. *European Signal Processing Conference*, 2018-Septe:533–537, 2018.

[25] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial Machine Learning at Scale. In *International Conference on Learning Representations (ICLR)*, 2017.

[26] Patrick Lucey, Jeffrey Cohn, Simon Lucey, Iain Matthews, Sridha Sridharan, and Kenneth M. Prkachin. Automatically detecting pain using facial actions. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–8. IEEE, 9 2009.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[28] Daniel Lopez Martinez, Ognjen Rudovic, and Rosalind Picard. Personalized Automatic Estimation of Self-Reported Pain Intensity from Facial Expressions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2318–2327. IEEE, 7 2017.

[29] Andre T Nguyen and Edward Raff. Adversarial Attacks, Regression, and Numerical Stability Regularization. In *AAAI 2019 Workshop on Engineering Dependable and Secure Machine Learning Systems*, 2018.

[30] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *arXiv*, 2016.

[31] 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*, ASIA CCS '17, page 506–519, New York, NY, USA, 2017. Association for Computing Machinery.

[32] Edward Raff, Shannon Lantzy, and Ezekiel J Maier. Dr. AI, Where Did You Get Your Degree? In Fernando Koch, Andrew Koster, David Riaño, Sara Montagna, Michael Schumacher, Annette ten Teije, Christian Guttmann, Manfred Reichert, Isabelle Bichindaritz, Pau Herrero, Richard Lenz, Beatriz López, Cindy Marling, Clare Martin, Stefania Montani, and Nirmalie Wiratunga, editors, *Artificial Intelligence in Health*, pages 76–83, Cham, 2019. Springer International Publishing.

[33] Edward Raff and Charles Nicholas. A Survey of Machine Learning Methods and Challenges for Windows Malware Classification. In *NeurIPS 2020 Workshop: ML Retrospectives, Surveys & Meta-Analyses (ML-RSA)*, 2020.

[34] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do Adversarially Robust ImageNet Models Transfer Better? *arXiv*, 2020.

[35] Marcos Sebastián, Richard Rivera, Platon Kotzias, and Juan Caballero. AVclass: A Tool for Massive Malware Labeling. In Fabian Monrose, Marc Dacier, Gregory Blanc, and Joaquin Garcia-Alfaro, editors, *Research in Attacks, Intrusions, and Defenses: 19th International Symposium, RAID 2016*, pages 230–253. Springer International Publishing, Paris, France, 2016.

[36] Silvia Sebastián and Juan Caballero. AVClass2: Massive Malware Tag Extraction from AV Labels. In *ACSAC*, 2020.

[37] Jeremy D. Seideman, Bilal Khan, and Antonio Cesar Vargas. Identifying malware genera using the Jensen-Shannon distance between system call traces. In *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, pages 1–7. IEEE, 10 2014.

[38] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y. Zhao. Fawkes: Protecting Privacy against Unauthorized Deep Learning Models. 2020.

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[40] Nedim Srndic and Pavel Laskov. Practical Evasion of a Learning-Based Classifier: A Case Study. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, SP '14, page 197–211, USA, 2014. IEEE Computer Society.

[41] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. When Does Machine Learning FAIL? Generalized Transferability for Evasion and Poisoning Attacks. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1299–1316, Baltimore, MD, 8 2018. {USENIX} Association.

[42] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The Space of Transferable Adversarial Examples. *arXiv*, pages 1–15, 2017.

[43] Rolf-Detlef Treede. The International Association for the Study of Pain definition of pain: as valid in 2018 as in 1979, but in need of regularly updated footnotes. *Pain reports*, 3(2):e643–e643, 3 2018.

[44] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020.

[45] C Xie, Z Zhang, Y Zhou, S Bai, J Wang, Z Ren, and A L Yuille. Improving Transferability of Adversarial Examples With Input Diversity. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2725–2734, 6 2019.

[46] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable Adversarial Perturbations. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, pages 471–486, Cham, 2018. Springer International Publishing.

## A  FIGURES REFERENCED IN MAIN TEXT
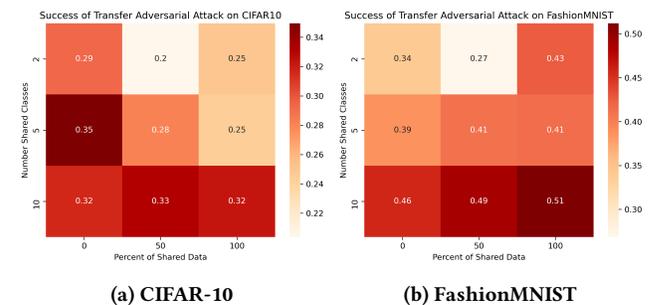


(a) CIFAR-10

(b) FashionMNIST

**Figure 17: Success of transfer attack dependent on variations of shared classes and percentage of data shared for two smaller datasets.**