The Alan Turing Institute

Model Similarity Phase 2: Dataset Similarity

October 2023
Phil Swatton, Data Scientist
Joanna Knight, Data Scientist
James Bishop, Principal Data Scientist

1 Executive Summary

In this project, we investigate the relationship between transfer attack success between two models and the similarity of the datasets on which they were trained. In particular, we examine whether similarity metrics are predictive of transfer attack success. We begin by assessing the validity of our roster of metrics. We confirm that, broadly, those metrics that did well under our assessment of their validity do predict transfer attack success. We conclude from our results that the dataset used to train a model is a useful source of information to an attacker. We therefore recommend that opening information regarding the training dataset of a deployed model be considered as a risk when making decisions regarding the openness of models, to be balanced against considerations of the benefits of open models.

Contents

2 Introduction

When deploying neural networks, data scientists need to guard against the threat of adversarial attacks. A common approach to defending against this threat has been keeping the model and information regarding the model private. This typically includes measures such as regulating the use of the model, restricting access to information on the model's architecture, and restricting access to information on the model's dataset.

However, these measures give rise to the approach of transfer attacks. Hostile actors will train a surrogate model to solve the same or a very similar task, generate adversarial attacks on this model, and then transfer them to the target model. Intuitively, the more similar the surrogate model to the target, the more likely we would expect a successful transfer attack to be.

In a previous project, ARC investigated the relationship between transfer attack success and model similarity [5], which we will refer to throughout as Phase 1. In Phase 1, the dataset used to train the model was held constant, while model architecture was altered. Results were varied, but conditional support for the notion that model similarity predicts transfer attack success was found. This project continues that work and investigates the relationship between transfer attack success and the similarity of the datasets used to train each model. We hold the model architecture and training regime constant while varying the datasets used to train the models.

To aid in conceptual understanding of dataset similarity in terms of transferability, we propose some formal definitions in Section 3. We use these formal definitions to motivate our hypotheses in Section 4. The hypotheses define expected relationships between transfer attack success and model vulnerability, transfer attack success and dataset similarity, and transfer attack successs and the size of both datasets.

After setting out our hypotheses, we introduce the dataset similarity metrics used in this work in Section 5. We first offer a typology of these metrics, showing which pieces of information from the datasets they draw on, and how they should be understood in light of our earlier definition of dataset similarity. In particular, we discuss the (non) generalisability of these measures, as where measures are equally good their generalisability to other use cases will serve to further distinguish them.

Details of our technical approach are outlined in Section 6. We describe our choice of dataset, CIFAR-10, and how we generate increasing differences between various altered versions of CIFAR-10.

Section 6 also discusses model training and transfer attack approaches. In both cases, we draw on the past work in model similarity Phase 1. We utilise a ResNet18 architecture with the first layer modified for the small image size of CIFAR-10. Similarly, we use the fast gradient attack and boundary attack used in Phase 1.

The results are presented in Section 7, and they are analysed to determine whether they support the hypotheses. A conclusion is drawn in Section 8 and recommendations for further work are made in Section 9.

3 Dataset Similarity

To better understand the concept of dataset similarity, we begin by offering a definition of the terms 'dataset' and 'dataset similarity' in a machine learning context. A degree of formalisation is useful to avoid potential confusion arising from colloquial use of the terms.

3.1 Dataset

Adapting the definition in [1], where $\mathcal{X}$ is a feature space and $\mathcal{Y}$ is a label space we define an *unlabelled dataset* D_U as a collection of feature vectors $\boldsymbol{x} \in \mathcal{X}$, while a *labelled dataset* D_L is a collection of feature-label pairs $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$. Note that a feature space implicitly defines the number of features a dataset can have and the units of measurement for those features.

As a simplifying assumption, we consider the label and prediction spaces to be the same. If a dataset has the labels $\{0, 1\}$, we thus consider the label space to be $\mathcal{Y} = [0, 1]$. This both simplifies notation and abstracts away any decision around thresholding or whether continuous or discrete outputs are desired.

3.2 Domain

For some feature space $\mathcal{X}$, we define a *domain* as $\mathcal{D} = \{\mathcal{X}, P_{\mathcal{X}}\}$ where $P_{\mathcal{X}}$ is some marginal probability distribution over $\mathcal{X}$, that is, $P_{\mathcal{X}} : \mathcal{X} \mapsto [0, 1]$.

We can consider an unlabelled dataset to comprise feature vectors sampled from some domain $\mathcal{D} = \{\mathcal{X}, P_{\mathcal{X}}\}$. A labelled dataset will similarly have feature vectors sampled from some domain $\mathcal{D} = \{\mathcal{X}, P_{\mathcal{X}}\}$ with labels applied after sampling.

Empirically, both unlabelled and labelled datasets give rise to an empirical distribution P_X over the observed feature vectors, which can be considered an estimator of $P_{\mathcal{X}}$.

3.3 Task

Adapting [15], given a specific domain $\mathcal{D} = \{\mathcal{X}, P_{\mathcal{X}}\}$, we denote a *task* as $\mathcal{T} = \{\mathcal{Y}, \mathcal{F}, L\}$, where $\mathcal{F}$ is the set of predictive functions f that map all $\boldsymbol{x}$ sampled from $\mathcal{D}$ onto some $y \in \mathcal{Y}$, and L is a loss function.

The goal of a machine learning task is to find a function $\hat{f} \in \mathcal{F}$ that minimises the loss function L for some future dataset drawn from the same domain $\mathcal{D}$. Increasing the size of the training set can reduce overfitting and thus improve validation loss. It follows that an additional attribute of a dataset that determines the learned function $\hat{f} \in \mathcal{F}$ is the size of the dataset.

3.4 Transfer learning and domain adaptation

Given a target domain $\mathcal{D}_T = \{\mathcal{X}_T, P_{\mathcal{X}_T}\}$ with task $\mathcal{T}_T = \{\mathcal{Y}_T, \mathcal{F}_T, L_T\}$, the goal of traditional supervised machine learning is to find a function $\hat{f}_T$ that minimises L_T for the test set. The learning process is framed as an optimisation problem and requires a labelled dataset of training samples, $(\boldsymbol{x}_{T_i}, y_{T_i})$ where $\boldsymbol{x}_{T_i} \in \mathcal{X}_T$ and $y_{T_i} \in \mathcal{Y}_T$.

Transfer learning uses knowledge from a source domain $\mathcal{D}_S = \{\mathcal{X}_S, P_{\mathcal{X}_S}\}$ with task $\mathcal{T}_S = \{\mathcal{Y}_S, \mathcal{F}_S, L_S\}$ to aid in the learning of the target model $\hat{f}_T$. In order to distinguish this learning task from traditional supervised machine learning, either $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$ must hold, and potentially both may be true. Following [12], we will refer to the case where $\mathcal{X}_S = \mathcal{X}_T, \mathcal{Y}_S = \mathcal{Y}_T$ but $P_{\mathcal{X}_S} \neq P_{\mathcal{X}_T}$ as *domain adaptation*.

3.5 Transfer attacks

First highlighted in [19], an *adversarial example* is a record $\boldsymbol{x}_i \in \mathcal{X}$ that has been modified with noise $\boldsymbol{\epsilon}$ such that the learned function $\hat{f}$ makes an incorrect prediction. For an adversarial attack to be considered successful in practice, the difference between the adversarial example and the original record should be essentially imperceptible to a human.

Finding adversarial examples in the untargeted case[1] can be written as the following optimisation problem [16]:

$$\boldsymbol{x}_i + \boldsymbol{\epsilon}_i \text{ where } \boldsymbol{\epsilon}_i = \underset{\boldsymbol{\delta}}{\operatorname{argmin}}\, \hat{f}(\boldsymbol{x}_i + \boldsymbol{\delta}) \neq y_i \tag{1}$$

Notably, adversarial examples trained on one model will often successfully fool another model, even if the two models were trained on disjoint datasets [19, 8, 16]. It is speculated that this occurs as, even with different architectures and datasets, the models learned from data typically represent similar functions to one another [8].

When performing a *transfer attack*, adversaries train a surrogate model $\hat{f}_S$ with the intention of using it to develop adversarial examples $\boldsymbol{x} + \boldsymbol{\epsilon}$ by solving (1). The goal is then to generate these adversarial examples such that a target model $\hat{f}_T$ is successfully fooled into assigning the incorrect label:

$$\hat{f}_T(\boldsymbol{x}_i + \boldsymbol{\epsilon}_i) \neq y_i \text{ where } \boldsymbol{\epsilon}_i = \underset{\boldsymbol{\delta}}{\operatorname{argmin}}\, \hat{f}_S(\boldsymbol{x}_i + \boldsymbol{\delta}) \neq y_i \tag{2}$$

[1] Finding targeted ones would correspond to adding noise to produce a particular label. We focus on untargeted adversarial examples in this report and thus present this version here.

3.6 Dataset similarity

There are many different measures of dataset similarity, which will be introduced later in this report. However, given the definitions of dataset, domain and task we may provide some intuitive observations.

Consider the following examples:

1. Unlabelled datasets D_S and D_T sampled from the same domain $\mathcal{D} = \{\mathcal{X}, P_{\mathcal{X}}\}$
2. Unlabelled dataset D_S sampled from $\mathcal{D}_S = \{\mathcal{X}, P_{\mathcal{X}_S}\}$ and unlabelled dataset D_T sampled from $\mathcal{D}_T = \{\mathcal{X}, P_{\mathcal{X}_T}\}$, where $P_{\mathcal{X}_S} \neq P_{\mathcal{X}_T}$
3. Unlabelled dataset D_S sampled from $\mathcal{D}_S = \{\mathcal{X}_S, P_{\mathcal{X}_S}\}$ and unlabelled dataset D_T sampled from $\mathcal{D}_T = \{\mathcal{X}_T, P_{\mathcal{X}_T}\}$, where $\mathcal{X}_S \neq \mathcal{X}_T$ and $P_{\mathcal{X}_S} \neq P_{\mathcal{X}_T}$

Without further information, we would expect the datasets in the first example to be the most similar as they exist in the same feature space and are sampled from the same probability distribution. The datasets in the second example would be expected to be less similar than the first: despite existing in the same feature space, they are sampled from different distributions. We would anticipate the third example to exhibit the most dissimilarity as the datasets exist in different feature spaces and are drawn from different probability distributions.

Instead of relying on intuition alone, measures of dataset similarity draw on the empirical attributes to quantitatively capture the extent to which two datasets are similar. Conditional on precisely which features of a dataset are used in a dataset similarity metric, we might expect different behaviours on the part of the metrics, with different consequences for predicting transfer success. Consider the following examples:

1. Unlabelled datasets D_S and D_T sampled from domain $\mathcal{D} = \{\mathcal{X}, P_{\mathcal{X}}\}$, where $D_S \cap D_T = \emptyset$
2. Unlabelled datasets D_S and D_T sampled from domain $\mathcal{D} = \{\mathcal{X}, P_{\mathcal{X}}\}$, where $D_S \cap D_T \neq \emptyset$

As a rule of thumb, about 100 samples are enough to approximate a multivariate normal distribution. It follows that, even when features are not distributed according to a multivariate normal distribution, it may require surprisingly few examples for the empirical distribution a dataset gives rise to to approximate the domain distribution. It may, therefore, be the case that it is difficult for a similarity measure based on distribution alone to distinguish between the first and second examples above. This will be true even if the example datasets have different sizes.

4 Hypotheses

In this section, we draw on both the preceding discussion and the wider literature on transfer attacks to develop our hypotheses.

4.1 Model Vulnerability

In the Phase 1 report, ARC tested the following two hypotheses [5]:
Hypothesis 1: *Target model vulnerability to attacks directly correlates with attack transferability*
Hypothesis 2: *For a given target model, the surrogate model's vulnerability to attacks negatively correlates with attack transferability*

The Phase 1 report found evidence to fully support hypothesis 1. It found mild evidence to support hypothesis 2, with a stronger correlation in the case of boundary attacks (see Section 6 below for a description of boundary attacks) [5].

Given their continued relevance to model security, we re-test these hypotheses in this report, taking advantage of the new test cases offered by variation in dataset similarity in place of variation in model similarity. We expect to find the same results as in Phase 1.

4.2 Dataset Similarity and Attack Transferability

Past research has suggested that transfer attacks between two models are successful as they represent similar functions [8]. Based on both this and on our preceding discussion, we expect that greater similarity between surrogate and target dataset causes greater transfer attack success.

We base this expectation on the fact that holding all other relevant variables constant, datasets drawn from similar domains should be more similar to each other than to datasets drawn from dissimilar domains. We would expect to lead to greater similarity between learned functions. We expect that this, in turn, would result in greater transfer attack success.

Our third hypothesis can then be expressed as:
Hypothesis 3: *Higher dataset similarity increases transfer attack success*

4.3 Dataset Size and Attack Transferability

In our discussion in Section 3.6, we noted that often the size of the dataset can affect the function learned from data. This is evidenced by the common advice that increasing the amount of training data can reduce the validation loss. Validation loss is an indicator of the quality of the learned function, and thus changes in validation loss correspond to changes in the learned function. We therefore hypothesise that the size of the dataset is related to transfer attack success.

In particular, if two datasets are close in size, we expect that they will have learned more similar functions (holding the domains constant). Our fourth hypothesis can therefore be expressed as:
Hypothesis 4: *The closer two datasets are in size, the more successful transfer attacks between models learned from them will be*

We stress that this hypothesis in particular is conditional. First, transfer attacks in practice only take place between datasets in reasonably close domains. A transfer attack based on MNIST images on a CIFAR-10 dataset would be nonsensical. Our hypothesis therefore assumes that the transfer attack taking place is reasonable in this sense.

Second, as datasets become increasingly large, we expect that the effect of size differences will decrease. We are largely working at smaller dataset sizes in these experiments, and so do not qualify the hypothesis for the purposes of this report, but this potential lack of generalisation should be borne in mind.

5 Dataset Similarity Metrics

There are many different metrics for assessing the similarity between two datasets, and no single metric dominates the literature. Therefore, this work considers a variety of metrics with different characteristics. This section begins by presenting a typology of the metrics, identifying differences between the approaches. We then introduce the metrics that will be applied in the experiments. The descriptions of the metrics will highlight some hyperparameters that need to be selected; the choice of hyperparameters for this work will be presented later in the technical approach in Section 6.

5.1 Typology

All of our metrics are in fact assessments of *dissimilarity*, as this is typically easier to capture than similarity. Smaller values are therefore associated with greater similarity, and higher values with greater difference. Beyond this, there are some key differences between the metrics which we shall discuss first before providing descriptions of each metric.

Most metrics are *unlabelled* as they do not take into account the labels of the dataset. Others are *labelled* incorporate the information of the labels into the metric. Some approaches require a shared feature space (albeit not necessarily a shared domain), while others allow for a different feature space.

Some metrics rely on the assumption that the datasets are sampled from a domain $\mathcal{D} = \{\mathcal{X}, P_{\mathcal{X}}\}$. They use the empirical distribution generated by the samples to approximate $P_{\mathcal{X}}$. Others work directly with the

samples in the dataset and require no assumption about the probability distribution from which they are sampled.

5.2 Metrics

5.2.1 Maximum Mean Discrepancy (MMD)

The maximum mean discrepancy (MMD) was developed in [9]. This work uses an empirical estimation of the MMD which is an unlabelled metric that requires the datasets to belong to the same feature space. It does not require any assumption about the probability distribution from which the datasets are sampled, instead being defined using the records in the dataset directly. For any two datasets D_A and D_B with N_A and N_B records, respectively, that share the same feature space $\mathcal{X}$, and for some positive-definite kernel $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, the empirical estimation of the MMD is defined as:

$$\text{MMD} = \frac{1}{2N_A^2}\sum_{i=1}^{N_A}\sum_{j=1}^{N_A}K(x_{Ai}, x_{Aj}) + \frac{1}{2N_B^2}\sum_{i=1}^{N_B}\sum_{j=1}^{N_B}K(x_{Bi}, x_{Bj}) - \frac{1}{N_A N_B}\sum_{i=1}^{N_A}\sum_{j=1}^{N_B}K(x_{Ai}, x_{Bj}) \quad (3)$$

where x_{Ai} and x_{Bj} are the i^{th} and j^{th} records of D_A and D_B, respectively. There are different choices for the kernel K and it will represent a measure of similarity between any two vectors in $\mathcal{X}$. If $K(x, y) > K(x, z)$, then vectors x and y are considered more similar than x and z.

If $D_A = D_B$ then the MMD $= 0$, and this is the minimum possible value of the MMD; however, as this work uses an empirical estimation it may be that very small negative numbers occur. The more that these datasets differ, the smaller will be the similarity between them, which will lead to a larger MMD statistic.

5.2.2 Proxy $\mathcal{A}$-Distance (PAD)

The $\mathcal{A}$-distance [11] is a measure of distance between two probability distributions. This work uses an approximation of the $\mathcal{A}$-distance called the proxy $\mathcal{A}$-distance (PAD), first introduced in [2] and applied in [7, 4, 6]. It is an unlabelled metric that requires the datasets to share the same feature space.

As discussed in Section 3, we can consider two datasets D_A and D_B to be samples drawn from domains $\mathcal{D}_A = \{\mathcal{X}, P_A\}$ and $\mathcal{D}_B = \{\mathcal{X}, P_B\}$. The approach for approximating the $\mathcal{A}$-distance between P_A and P_B creates a new dataset, D, which is a union of D_A and D_B with each record from D_A labelled 0 and each record from D_B labelled 1 (or vice versa). A classifier is then trained on a subset of D with the remainder of the data being held out as a test dataset. It is necessary for the same number of data samples from D_A and D_B to be included in the test dataset.

When the classifier is trained, the mean absolute error between the test dataset labels and the predictions is calculated. As there are only two classes, the maximum value of the resulting error ϵ will be 1. The error is used to define the proxy $\mathcal{A}$-distance:

$$\text{PAD} = 2(1 - 2\epsilon) \quad (4)$$

The intuition is that when the datasets are significantly different, a classifier should be able to distinguish between them leading to a low error, ϵ. In turn, this will lead to a PAD close to 2. The PAD is bounded to $[-2, 2]$. However, if a classifier leads to a PAD less than zero, then this means it is performing worse than a random assignment of labels. Assigning random labels will lead to an error ϵ close to 0.5 and a PAD close to zero.

5.2.3 Kernel Density Estimation (KDE)

Kernel density estimation (KDE) is another unlabelled approach to measuring the similarity between datasets. It requires that datasets D_A and D_B share the same n-dimensional feature space, $\mathcal{X}$, and assumes that they belong to domains $\mathcal{D}_A = \{\mathcal{X}, P_A\}$ and $\mathcal{D}_B = \{\mathcal{X}, P_B\}$, where P_A and P_B are continuous.

The distributions P_A and P_B will have probability densities p_A and p_B. This approach assumes that D_A and D_B are independent and identically distributed samples from p_A and p_B. It is then possible to estimate the density functions using a kernel density estimator.

For a dataset of samples $D = \{x_1, \ldots, x_m\} \subset \mathcal{X}$ and an unknown density function p, kernel density estimation defines an estimator for the density as:

$$\hat{p}(x) = \frac{1}{m} \sum_{i=1}^{m} K_h(x - x_i) \tag{5}$$

where K_h is some kernel function and h is smoothing parameter called the bandwidth.

The estimated density functions $\hat{p}_A$ and $\hat{p}_B$ and a distance metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can then be used to estimate a distance between P_A and P_B using integration:

$$\int_{\mathbb{R}^n} d(\hat{p}_A, \hat{p}_B) dx \tag{6}$$

5.2.4 KL Divergence

The Kullback–Leibler (KL) divergence is another unlabelled approach that requires datasets D_A and D_B to share the same n-dimensional feature space $\mathcal{X}$ and assumes that they belong to domains $\mathcal{D}_A = \{\mathcal{X}, P_A\}$ and $\mathcal{D}_B = \{\mathcal{X}, P_B\}$. The KL divergence between P_A and P_B is defined as:

$$D_{KL}(P_A||P_B) = \int_{\mathbb{R}^n} p_A(x) \log \left(\frac{p_A(x)}{p_B(x)} \right) dx \tag{7}$$

where p_A and p_B are the associated density functions. The KL divergence is not strictly a metric as it is not guaranteed to be symmetric; that is, $D_{KL}(P_A||P_B)$ does not necessarily equal $D_{KL}(P_B||P_A)$.

A method for approximating the KL divergence for continuous functions over multiple dimensions was presented in [21]. The approach assumes that the samples in D_A and D_B are independent and identically distributed. The method then uses the records in D_A and D_B to estimate the KL divergence.

The first step of the process is to fit a $(k+1)$-nearest neighbour estimator to D_A and a k-nearest neighbour estimator to D_B for some positive integer, k. The estimated KL divergence from P_A to P_B is defined in [21] as:

$$\hat{D}_{KL}(P_A||P_B) = \frac{n}{N_A} \sum_{i=1}^{N_A} \log \frac{s_k(x_i)}{r_k(x_i)} + \log \frac{N_B}{N_A - 1} \tag{8}$$

where there are n dimensions in the shared feature space, D_A has N_A records, D_B has N_B records, and $r_k(x_i)$ and $s_k(x_i)$ are, respectively, the Euclidean distances to the k^{th} nearest-neighbour of x_i in $D_A \setminus x_i$ and D_B. The derivation of $\hat{D}_{KL}(P_A||P_B)$ and a proof that it converges almost surely to $D_{KL}(P_A||P_B)$ can be found in [21].

When $k = 1$ and D_A and D_B are identical, then $\hat{D}_{KL}(P_A||P_B)$ cannot be calculated as $s_k(x_i)$ will be zero for all i, which leads to $\log 0$ in equation 8.

5.2.5 Optimal Transport

Optimal transport is the problem of transporting mass from one distribution into another in the most efficient way possible. If the cost function c is itself a metric, then the overall cost of an optimal transport solution defines a metric between the two distributions.

The total cost of optimal transport could in and of itself be used as distance between datasets D_A and D_B by estimating the optimal transport between P_A and P_B (i.e. over the features alone if labelled). The distance between the two distributions is given by

$$d(D_A, D_B) = \inf_{\pi \in \Pi(P_A, P_B)} \int c(x_A, x_B) d\pi(x_A, x_B) \tag{9}$$

where π is a transport plan, which is a joint distribution between P_A and P_B which defines where mass is moved from P_A to P_B, and $\Pi(P_A, P_B)$ denotes all the set of all joint distributions over $\mathcal{X} \times \mathcal{X}$ with marginal distributions P_A and P_B.

If the cost function c is $\|x_A - x_B\|^p$, then this is named the p-Wasserstein or W_p distance (e.g. 2-Wasserstein or W_2 where Euclidean distance is used).

A more detailed motivation and treatment of optimal transport is given in appendix A.

A metric named optimal transport dataset distance (OTDD) was introduced in [1]. This metric uses optimal transport to compute a distance between labelled datasets. Importantly, it allows for the label spaces of both datasets to be entirely disjoint (i.e. only a shared feature space is required). Of the metrics considered in this report, it is thus the only labelled metric.

For this approach it is necessary to define a cost function between pairs of feature vectors *and* labels. If labels of the datasets belonged to metric spaces, then it would be possible to define a cost directly between two label values and a natural extension of the cost function would be:

$$c_Z(z, z') = (c_X(x, x')^p + c_Y(y, y')^p)^{1/p} \tag{10}$$

for feature-label pairs $z = (x, y)$, $z' = (x', y')$ and some $p \geq 1$, where c_X, c_Y are as before metrics such as Euclidean distance.

However, it will usually not be feasible to define the metric c_Y directly between two labels. The labels will typically represent nominal values such as 'car' or 'cat', where there is no sensible metric between the two. Instead, [1] represent the labels as conditional probability distributions: $P_{Ay} = P_A(X|Y = y)$ and $P_{By} = P_B(X|Y = y)$ and define the distance as:

$$c_Z(z, z') = (c_X(x, x')^p + W_p(P_{Ay}, P_{By})^p)^{1/p} \tag{11}$$

The OTDD between labelled datasets D_A and D_B is then defined as:

$$\text{OTDD}(D_A, D_B) = \min_{\pi \in \Pi(P_A, P_B)} \int_{\mathcal{Z}_A \times \mathcal{Z}_B} c(z, z') d\pi(z, z') \tag{12}$$

where $\mathcal{Z}_A = \mathcal{X}_A \times \mathcal{Y}_A$ and $\mathcal{Z}_B = \mathcal{X}_B \times \mathcal{Y}_B$.

It is worth noting in particular that while the version of optimal transport we have discussed here requires a shared feature space, a more general version named the Gromov-Wasserstein distance relaxes this requirement. Optimal transport is thus the only family of metrics discussed in this report which does not necessarily require a shared feature space.

6 Technical Approach

In this section, we set out the technical approach taken in this report.

Transfer attack success is likely a function of many variables such as model similarity, learning algorithm, and dataset similarity. In this report, we constrain our focus on the last of these variables, and thus hold other variables constant in our experiments by re-using the same model and training regime across all of our experiments. Only the datasets are varied between experiments.

6.1 Data

In our experiments we reuse the CIFAR-10 dataset [13] of low resolution images from model similarity Phase 1 [5]. To generate pairs of datasets with differences between them, we apply various dataset transformations to CIFAR-10. The transformations we use are:

1. Null (no transformation)
2. Grayscaling
3. A Gaussian blur with standard deviation 1 and kernel size 3 ('little blur')
4. A Gaussian blur with standard deviation 3 and kernel size 3 ('big blur')
5. Rotating the images 180 degrees

Examples of the transformations of images are shown in Figure 1. Although there is little observable difference between the two blurred transforms, the difference in variance between the two is significant and this work will investigate whether the metrics can distinguish between the two transforms.

We produce datasets for each of these transforms. For each pair we denote one dataset as A and one as B. Dataset B is always the dataset with transforms applied to it. We manipulate the datasets for each transform group further by dropping observations. When dropping from both A and B, drop indices are selected randomly, but we ensure that data dropped from A is kept in B and vice versa. Dropping data in this way allows us to assess the impact of dataset size independently of the similarity of dataset distributions.

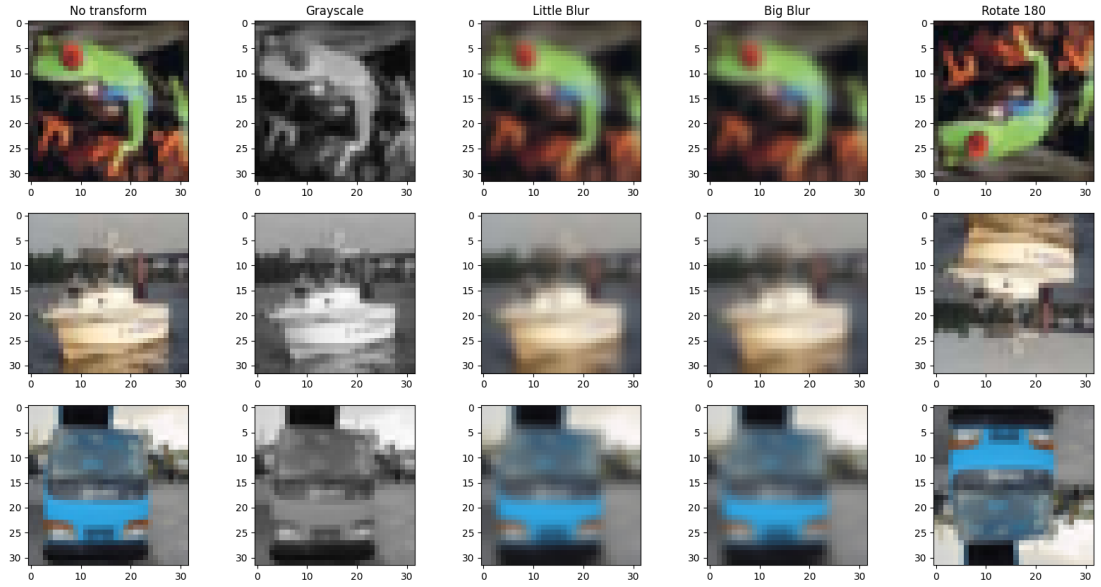

Figure 1: The four transformations used in this work, applied to three images in the CIFAR-10 dataset

We have six data drop combinations, which are (0%, 0%), (0%, 50%), (50%, 0%), (50%, 50%), (25%, 75%), and (75%, 25%), where the first number denotes the percentage of data dropped from A and the second the percentage of data dropped from B.

For each drop combination, we separately apply each of the noted 5 transforms to dataset B, giving us 30 unique dataset pairs. We drop the (0%, 0%) with null transform as this produces an identical A and B, leaving 29 pairs. We generate each combination three times with a different seed, for a total of 87 pairs and 174 models.

6.2 Model Training

As with dataset choice, for purposes of continuity we reuse a model architecture from Phase 1. We selected the ResNet18 architecture from among the available choices to fix in this phase [5].

We similarly reuse the training regime from Phase 1 with some small alterations. We train each model for 10 epochs with stochastic gradient descent. We use a weight decay of 0.0005 and a momentum value of 0.9. As in Phase 1, we use one-cycle annealing with a maximum learning rate of 0.1.

6.3 Metric Implementations

6.3.1 Embeddings

The CIFAR-10 dataset consists of images that of size 32x32 pixels. With three colour channels (red, green and blue) this leads to a total number of 3,072 features in each image. We discuss in appendix B why it is possible to meaningfully compute dataset similarity metrics on the raw pixel values. However, many of the metrics could not be calculated in a reasonable time-frame with such a larger number of dimensions, and several dimension reduction techniques have been considered in this work.

The first is the Inception V3 embedding [18], which uses a pre-trained neural network to produce a feature vector of size 2,048 for each image. The second option uses the Inception V3 model to embed the data and then applies principal component analysis (PCA) to further reduce the number of features down to 50. The third method also uses the Inception V3 model to embed the data and then applies uniform manifold approximation (UMAP) [14] to reducing the embedding down to two dimensions.

Reducing the number of dimensions to two means that the datasets can be visualised, as shown in Figure 2. The heatmaps in the figure represent the probability densities of the whole dataset, having had the relevant transform and then the Inception V3 and UMAP embeddings applied. Figure 3 represents the CIFAR training dataset with records dropped according to the values of the drop combinations applied to the experiment groups. The figure shows how the probability density of the dataset remains similar, but not identical, as an increasing number of records are dropped.

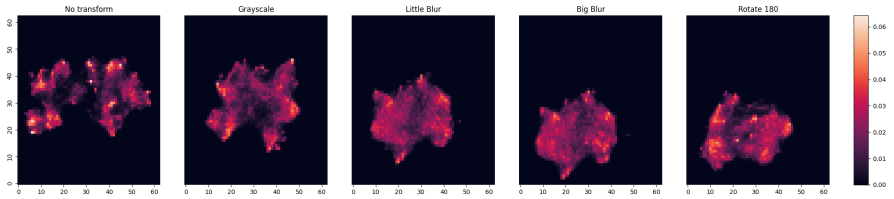

Figure 2: The four transformations used in this work, applied to the whole CIFAR training dataset shown in two dimensions using the Inception V3 embedding and then UMAP. Each heatmap shows the probability density in each bin.

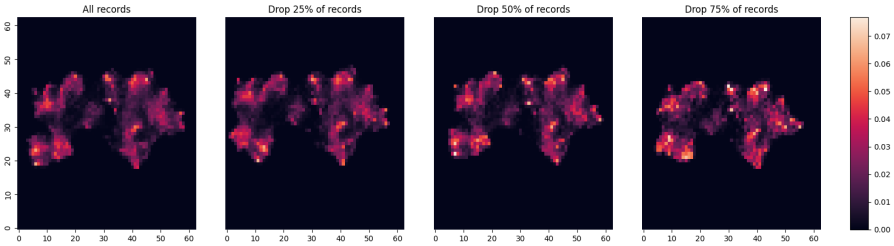

Figure 3: The CIFAR training dataset shown in two dimensions using the Inception V3 embedding and then UMAP. The number of records dropped from the dataset is varied between the plots. Each heatmap shows the probability density in each bin.

6.3.2 Hyperparameters

Our high-level choices of hyperparameters for the various metrics are discussed below, and are summarised in Table 1 along with the embedding options that have been applied for each metric. Note that some metrics

have been applied to the dataset pairs multiple times for alternative hyperparameters. Our complete set of hyperparameter choices can be found by investigating the relevant metric config files in https://github.com/alan-turing-institute/arc-model-similarities-phase-2/tree/main/configs/metrics.

Table 1: Similarity metric summary

Metric	Implementation	No embedding	Inception	Inception + PCA	Inception + UMAP
MMD	RBF kernel	Y	Y	Y	Y
PAD	Linear kernel	N	N	Y	Y
PAD	RBF kernel	N	N	Y	Y
KDE	L2 distance	N	N	N	Y
KDE	Total variation distance	N	N	N	Y
KL	$k = 1$	N	N	Y	Y
OT	OTDD	Y	Y	Y	Y

Metrics in this work with the hyperparameters and embeddings that were applied.

MMD

Our choice of kernel for maximum mean discrepancy (MMD) is the radial basis function (RBF):

$$K(x, y) = \exp\left(-\gamma||x - y||^2\right) \tag{13}$$

where $\gamma = \frac{1}{n}$ and x and y both have n features. The MMD is one metric that can be run for the three embedding options, as well as for the raw dataset without embeddings.

PAD

Following other work using the proxy $\mathcal{A}$-distance (PAD) metric [7, 6], our classifier of choice for the is a support vector machine (SVM). An SVM itself has many hyperparameters to be selected. Two different kernels were applied: the linear and RBF kernels. For each kernel, we performed a search over regularisation parameters of 0.01, 1, and 10, and selected the model that produced the minimum error.

We used PAD with two of our embedding options: Inception + UMAP, and Inception + PCA. We did not run it without embeddings or with Inception alone as found that the method became too slow as the number of features increased.

KDE

For kernel density estimation metrics, we used the Gaussian kernel for all implementations. We applied two different distance metrics:

L_2 norm:

$$||P - Q||_2 = \left(\int_{\mathbb{R}^n} (p(x) - q(x))^2 dx\right)^{1/2} \tag{14}$$

Total variation:

$$TV(P, Q) = \frac{1}{2}\int_{\mathbb{R}^n} |p(x) - q(x)|dx \tag{15}$$

The KDE metrics are only calculated for the Inception + UMAP embedding, as integrating over a large number of dimensions was computationally expensive. We abandoned PCA after early experimentation as it showed poor performance early on, as we preferred to focus on more promising metrics for expediency.

KL-divergence

The implementation of the estimation of the Kullback–Leibler (KL) divergence used a value of $k = 1$. The KL divergence is calculated only for the embeddings with Inception V3 and then PCA and UMAP as early experimentation showed the k-nearest neighbour algorithm was not effective with a larger number of features.

OTDD

We used Microsoft's implementation of the optimal transport dataset distance (OTDD)[2]. This uses the p-Wasserstein distance as the cost function and we have set the value of $p = 2$. We made a small number of adjustments to Microsoft's default set of parameters for the metric. These changes resulted in much faster performance without any substantial change in early results[3]. We ran the OTDD both without any embedding and with all three embeddings applied.

6.4 Transfer Attacks

For purposes of continuity, we reuse the transfer attacks from Phase 1 [5]:

1. A **fast gradient attack**, which uses gradient descent to optimise an attack to maximise loss [8]
2. A **boundary attack** which perturbs the image then decreases the perturbation to be as close to the boundary between classes as possible

In the case of the fast gradient attack, the L_2 norm was used to measure the magnitude between the original and perturbed image. For both attacks, we use a perturbation value of 1.

There is one change from the Phase 1 approach, which reflects this report's emphasis on dataset similarity. We use test images from datasets A and B to create adversarial images. When transferring from A to B (and from B to A), we faced the question of whether to base the attack images on the surrogate or target dataset. While, in the real world, we may expect attackers to use the surrogate dataset, it may still be informative to assess whether results change when using images from the target distribution. We therefore generated adversarial examples from both the test images of A and the test images of B (both for transferring from A to B and from B to A), to assess whether this also makes a difference on the final transfer success rates.

6.5 Attack Metrics

We reuse both attack metrics from Phase 1 [5]:

1. **Success rate**: The percentage of images misclassified by a model, given that the model correctly identifies the original image.
2. **Mean loss increase**: The mean value of the difference between the loss that a model assigns to an adversarial image and the loss that it assigns to the original image.

We use these metrics both as transfer metrics (i.e. if adversarial images were created using model A, we test them on model B) and as model vulnerability metrics (i.e. if created using model A, tested on model A).

7 Results

In this section, we present the results of our experiments. In our experiments we had both a large number of similarity metrics and a large number of transfer attacks to examine. This is because we have two kinds of attack (fast gradient and boundary attack), two directions of attack (with A as surrogate and B as target, and with B as surrogate and A as target), two distributions of images on which adversarial attacks are generated (taken from A's test set, taken from B's test set), and two attack metrics. This results in 8 distinct transfer attacks and 16 distinct attack success metrics in total.

7.1 Validity of Metrics

Our report has two goals - to discover the quality of the various candidate metrics, and to use those metrics to confirm or reject our hypotheses. To avoid the risk of following a circular logic (our metrics work because

[2] `https://github.com/microsoft/otdd`

[3] These changes are noted in the relevant config file for the OTDD metrics, which can be found at this link: https://github.com/alan-turing-institute/arc-model-similarities-phase-2/blob/main/configs/metrics/otdd.yaml

they confirm our hypotheses, our hypotheses are true because our metrics confirm them), we draw on a psychometric framework of measure validity.

The Trinitarian concept of validity we draw on defines three types of measure validity: construct validity, content validity, and criterion validity. The first assesses the extent to which a measure captures the concept it is supposed to be measuring. For assessment it can be subdivided into two sub forms of validity: convergent validity, where measures are correlated with other measures that capture the same concepts, and discriminant validity, where measures are uncorrelated with measures that capture theoretically orthogonal concepts.

Construct validity is essential for establishing the other two. Table 2 presents a matrix of Pearson's correlations between each of the similarity metrics. We use these correlations to assess the convergent validity of our metrics: if they all capture dataset similarity, they should broadly be correlated with one another. We tested each correlation, and correlations significant at the 95% confidence level are reported in bold.

Table 2: Similarity Metrics Correlation Matrix

	MMD (None)	MMD (Inception)	MMD (UMAP)	MMD (PCA)	OTDD (None)	OTDD (Inception)	OTDD (UMAP)	OTDD (PCA)	KL Approx (UMAP)	KL Approx (PCA)	KDE - L2 (UMAP)	KDE - TV (UMAP)	PAD - Linear (UMAP)	PAD - RBF (UMAP)	PAD - Linear (PCA)	PAD - RBF (PCA)
MMD (None)		**-0.43**	**0.54**	-0.19	**0.73**	**0.41**	**0.31**	**0.31**	**0.51**	**0.28**	**0.48**	**0.49**	**0.57**	**0.49**	-0.02	-0.13
MMD (Inception)	**-0.43**		**0.23**	**0.93**	**-0.50**	**0.49**	0.16	**0.62**	**0.29**	**0.50**	**0.44**	**0.38**	0.12	**0.38**	**0.28**	**0.41**
MMD (UMAP)	**0.54**	**0.23**		**0.43**	**0.26**	**0.66**	**0.87**	**0.61**	**0.98**	**0.57**	**0.93**	**0.97**	**0.98**	**0.97**	0.09	0.09
MMD (PCA)	-0.19	**0.93**	**0.43**		**-0.28**	**0.71**	**0.31**	**0.83**	**0.50**	**0.71**	**0.66**	**0.59**	**0.32**	**0.59**	**0.28**	**0.44**
OTDD (None)	**0.73**	**-0.50**	**0.26**	**-0.28**		**0.42**	0.05	0.19	0.18	0.09	**0.23**	0.2	**0.27**	0.21	-0.12	**-0.40**
OTDD (Inception)	**0.41**	**0.49**	**0.66**	**0.71**	**0.42**		**0.40**	**0.93**	**0.67**	**0.76**	**0.83**	**0.75**	**0.57**	**0.76**	0.13	0.13
OTDD (UMAP)	**0.31**	0.16	**0.87**	**0.31**	0.05	**0.40**		**0.36**	**0.88**	**0.36**	**0.74**	**0.81**	**0.90**	**0.81**	0.06	0.13
OTDD (PCA)	**0.31**	**0.62**	**0.61**	**0.83**	0.19	**0.93**	**0.36**		**0.66**	**0.89**	**0.84**	**0.76**	**0.52**	**0.76**	**0.25**	**0.36**
KL Approx (UMAP)	**0.51**	**0.29**	**0.98**	**0.50**	0.18	**0.67**	**0.88**	**0.66**		**0.62**	**0.95**	**0.98**	**0.97**	**0.98**	0.14	0.21
KL Approx (PCA)	**0.28**	**0.50**	**0.57**	**0.71**	0.09	**0.76**	**0.36**	**0.89**	**0.62**		**0.77**	**0.71**	**0.48**	**0.72**	**0.22**	**0.53**
KDE - L2 (UMAP)	**0.48**	**0.44**	**0.93**	**0.66**	**0.23**	**0.83**	**0.74**	**0.84**	**0.95**	**0.77**		**0.99**	**0.88**	**0.99**	0.19	**0.24**
KDE - TV (UMAP)	**0.49**	**0.38**	**0.97**	**0.59**	0.2	**0.75**	**0.81**	**0.76**	**0.98**	**0.71**	**0.99**		**0.93**	**1.00**	0.17	**0.23**
PAD - Linear (UMAP)	**0.57**	0.12	**0.98**	**0.32**	**0.27**	**0.57**	**0.90**	**0.52**	**0.97**	**0.48**	**0.88**	**0.93**		**0.93**	0.11	0.06
PAD - RBF (UMAP)	**0.49**	**0.38**	**0.97**	**0.59**	0.21	**0.76**	**0.81**	**0.76**	**0.98**	**0.72**	**0.99**	**1.00**	**0.93**		0.18	**0.23**
PAD - Linear (PCA)	-0.02	**0.28**	0.09	**0.28**	-0.12	0.13	0.06	**0.25**	0.14	**0.22**	0.19	0.17	0.11	0.18		0.06
PAD - RBF (PCA)	-0.13	**0.41**	0.09	**0.44**	**-0.40**	0.13	0.13	**0.36**	0.21	**0.53**	**0.24**	**0.23**	0.06	**0.23**	0.06	

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Broadly, most of the metrics reported in table 2 are in agreement with each other: their correlations are positive and significant at the 95% confidence level. A small handful are negative and significant: notably these correlations always involve a metric which uses either the Inception-only or the Inception plus PCA embedding. There are also a number of null results: these are concentrated on the PAD metrics using Inception plus PCA and the OTDD metric without any embeddings. Despite these disagreements and the wide range in the sizes of the correlations, we are sufficiently satisfied that these correlations show convergent validity to continue with our analysis.

Content validity is a subjective assessment of how well the measure captures all attributes of the concept in question. Our preceding discussions have emphasised that the similarity measures primarily draw on the empirical distribution of the datasets which is arguably the most salient attribute for determining similarity. However, our metrics also do not (explicitly) capture attributes such as dataset size. While they have some content validity, we do therefore highlight that they are not perfect in this regard.

Finally, criterion validity is assessed by assessing how well a measure predicts other related concepts. Typically,

this should be assessed against an already-known result, lest we fall into the trap of the circular logic outlined above. In our case, a standard that has already been assessed is that many of these metrics have been shown to predict successful transfer learning [1, 10]. Since to compute our transfer success rate metrics we need to first compute the base transfer of the images, we are able to assess this.

7.2 Base Transfer Success Rates

Prior to generating adversarial images, we take the images from the test dataset of either dataset A or B, then pass them through the target model. Where the test images come from the surrogate model, this is effectively evaluating the transfer performance of the target model on the surrogate model's distribution without tuning. Since our similarity metrics are in practice dissimilarity metrics, they should be negatively correlated with base transfer success in this case. This serves to act as a test of criterion validity.

Where the images come from the target model dataset's test set, there should be no particular pattern as this will have no relationship with the similarity of the two training datasets, as the surrogate dataset is not used at all. We use this correlation to assess the discriminant validity of the metrics, as these two things should be independent of one another by construction. We present these results for both of these scenarios in table 3.

Table 3 is divided into four columns. The first two columns show transfer success using model A as the surrogate and model B as the target (recall that where transforms have been applied, they have been applied to the training dataset for model B), while the next two show attacks from model B to model A. The direction of attack is noted in the first layer of heading.

We then further divide into attacks based on the distribution of model A (i.e. drawn from dataset A's otherwise unused test set) and attacks based on the distribution of model B. The outer columns thus use the distribution of the surrogate model, and the inner columns use the distribution of the target model. We repeat this structure in terms of columns across all further results tables in this report.

On the rows of table 3 are the labels for our various dataset similarity metrics. The table contains Pearson's correlations between the similarity metrics and the base transfer success rates, with results significant at the 95% confidence level reported in bold.

Table 3: Base Transfer Success

Similarity Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
MMD (None)	**-0.81**	**0.21**	0.02	**-0.62**
MMD (Inception)	**0.27**	**-0.22**	0.06	0.05
MMD (UMAP)	**-0.74**	-0.19	**0.21**	**-0.76**
MMD (PCA)	-0.01	**-0.23**	0.04	-0.2
OTDD (None)	**-0.57**	-0.11	**-0.33**	**-0.38**
OTDD (Inception)	**-0.50**	**-0.24**	-0.16	**-0.55**
OTDD (UMAP)	**-0.53**	-0.09	**0.22**	**-0.59**
OTDD (PCA)	**-0.37**	-0.16	-0.01	**-0.44**
KL Approx (UMAP)	**-0.69**	-0.09	**0.25**	**-0.60**
KL Approx (PCA)	**-0.34**	**-0.27**	-0.21	0.16
KDE - L2 (UMAP)	**-0.64**	-0.21	0.17	**-0.68**
KDE - TV (UMAP)	**-0.66**	-0.18	**0.23**	**-0.70**
PAD - Linear (UMAP)	**-0.72**	-0.1	0.21	**-0.74**
PAD - RBF (UMAP)	**-0.66**	-0.19	**0.23**	**-0.69**
PAD - Linear (PCA)	0.08	0.04	-0.01	0.05
PAD - RBF (PCA)	0.13	**0.23**	**0.50**	0.08

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

As expected, in the outer columns (i.e. those results based on the test sets of the surrogate datasets), where statistically significant at the 95% confidence level our metrics are for the most part negatively correlated with base transfer success, thus confirming their criterion validity.

There are some notable exceptions: for the MMD and PAD metrics, where they use the Inception-only or Inception plus PCA embeddings results are either null results or even positive and significant. This is also partially true for the approximate KL metric using the Inception plus PCA embedding. We therefore note an early absence of criterion validity for these metrics using these embeddings. Notably, these were also the metrics producing many of the significant negative correlations in the earlier tests of construct validity. For these metrics there is therefore a clear suggestion that the use of these embeddings produces invalid measures.

Also as expected, all of the correlations in the middle two columns are either null results, or are for the most part relatively weak albeit significant correlations, though with no overall pattern in direction. We note that roughly half of these significant results belong to the same metrics that failed the earlier assessment of criterion validity. Overall, we take this set of null results to demonstrate the discriminant validity of our metrics.

Beyond our primary concern of establishing the validity of our roster of metrics, there is a second relevance for this report. While our emphasis is on the use of information regarding a model's training data to train a similar surrogate to generate adversarial examples on, this also illustrates that an attacker could instead exploit the same information (or even surrogate model) to find out of distribution examples a model is unlikely to correctly classify without modification. This represents a secondary consideration for the protection of information regarding a model's dataset.

With the varying degrees of validity of our various metrics established, we now turn to presenting the results for each of our hypotheses.

7.3 Hypotheses 1 and 2

Hypotheses 1 and 2 are closely related to one another and so we present their results together.

For each hypothesis, we present two tables containing correlations between attack vulnerability metrics as generated for the model when using that model to create adversarial images, and transfer attack success. We always relate like-for-like metrics - for instance, we correlate transfer success metrics based on the fast gradient attack with images from the distribution of model A with model vulnerability from generating fast gradient attack images based on the distribution of model A.

Each row of the tables for hypotheses 1 and 2 corresponds to a single attack. Each table corresponds to a single attack success metric, where the corresponding vulnerability and transfer metrics have been correlated with one another. We present correlations statistically significant at the 95% confidence level in bold. All correlations are Pearson's correlations.

Table 4 contains the first set of results for hypothesis 1, which suggests that the target model's vulnerability is predictive of transfer success rate. Table 4 presents these results for the case of success rate as the attack metric of choice.

Table 4: Target Vulnerability - Success Rate

Attack	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Fast Gradient Attack	**-0.36**	**-0.25**	-0.07	**-0.35**
Boundary Attack	**0.94**	**0.95**	**0.33**	**0.95**

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Interestingly, while for results based on the success rate boundary attacks conform to our expectations, the results for fast gradient attacks do not. For the former, all results show positive and mostly large correlations

significant at the 95% confidence level. By contrast, for the latter where results are significant at the 95% confidence level they are negative.

The discrepancy is particularly surprising as the idea that the target model being *less* vulnerable defies initial intuitions. It should be borne in mind however that in our experiments, while dataset similarity and dataset size are causally independent constructs we can directly correlate with transfer attack success, the same is not true of vulnerability, as this will stem from a large number of the earlier decisions we made. We speculate on some potential drivers of this inverted relationship in the conclusion section.

Table 5 also presents results relating to hypothesis 1, but this time using the mean loss increase attack metric.

Table 5: Target Vulnerability - Mean Loss Increase

Attack	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Fast Gradient Attack	**0.85**	**0.28**	**0.37**	**0.89**
Boundary Attack	**0.91**	**0.97**	**0.35**	**0.32**

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Here, all results presented confirm our hypothesis. All correlations are positive and statistically significant at the 95% confidence level. However, we note that transfer success rate more directly captures the goal of an attacker in the context of transfer attacks, so given our mixed results in table 4, we ultimately find mixed evidence for hypothesis 1. We confirm it in the case of boundary attacks, but fail to confirm it in the case of fast gradient attacks.

Table 6 presents the first set of results for hypothesis 2, which suggests that surrogate model vulnerability will be positively associated with transfer attack success. As before, we start by presenting the results in the case of attack success rate.

Table 6: Surrogate Vulnerability - Success Rate

Attack	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Fast Gradient Attack	-0.15	-0.21	**-0.71**	-0.14
Boundary Attack	-0.13	**0.28**	**-0.54**	**0.38**

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Here, while half of the results are statistically significant, they are in opposite directions, and without any particular pattern beyond the two negative results being in the case where B is the surrogate and A is the target, and attacks are based on the distribution of A. We are therefore unable to draw any particular conclusions in terms of the relationship between surrogate vulnerability and attack success where success rate is the metric in question.

Finally, table 7 presents the second set of results for hypothesis 2, this time with mean loss increase as the metric being used.

Table 7: Surrogate Vulnerability - Mean Loss Increase

Attack	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Fast Gradient Attack	0.01	**0.57**	**0.72**	0.08
Boundary Attack	0.07	**0.26**	**0.36**	**-0.44**

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Here as before for hypothesis 1, a clearer pattern emerges. Where the distribution of images belongs to the target model, the correlations are all positive and significant at the 95% confidence level. We therefore conclude that where using mean loss increase as the metric of attack success and basing the attacks on the distribution of the target model, surrogate vulnerability is correlated with attack transfer success. However, when using images from the distribution of the surrogate model, this no longer holds.

As before however, attack success rate overall is a more useful measure for understanding transfer attack success as it conditions on the image previously being successfully classified prior to being converted into an adversarial image. We therefore conclude that we find little overall evidence to support hypothesis 2.

Appendix C gives additional context to these results. It contains plots showing the distribution of model vulnerabilities, and makes clear that generally models were more vulnerable to the fast gradient attack than to the boundary attacks. It also provides visual representations of the results discussed here.

7.4 Hypothesis 3

For hypothesis 3, we present four tables containing correlations between dataset similarity metrics and transfer attack success. Each table corresponds to one attack and one transfer attack success metric.

As for our earlier hypotheses, we have divided these tables into four columns denoting both the direction of attack and the distribution on which the attack was based.

The dataset similarity metrics in question are on the rows of the tables. This will contain the name of the metric, and where relevant a label describing the hyperparameters used (see Section 6.3 above). The embedding used is listed after the name - label pair in brackets.

It is important to note that since most of our similarity metrics are symmetrical (i.e. $sim(A, B) = sim(B, A)$), each metric is reused across the attack directions. More precisely, it is reused in the calculation of correlations between attack success and similarity for both attacks from A to B and for attacks from B to A. The exception for this is the KL divergence, which has separate values for A and B. As before, we present correlations statistically significant at the 95% confidence level in bold and all correlations are Pearson's correlations. Early experimenting with Spearman's correlations for these results did not produce meaningfully different results and inferences.

Table 8 shows the correlation between our dataset similarity metrics and transfer attack success in the case of the fast gradient attack.

Table 8: Fast Gradient Attack - Success Rate

Similarity Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
MMD (None)	-0.17	**-0.69**	**-0.63**	**-0.53**
MMD (Inception)	0.06	**0.41**	0.21	**0.47**
MMD (UMAP)	**-0.48**	**-0.59**	**-0.64**	**-0.23**
MMD (PCA)	-0.09	0.16	-0.05	**0.26**
OTDD (None)	**-0.42**	**-0.76**	**-0.70**	**-0.77**
OTDD (Inception)	**-0.39**	**-0.42**	**-0.57**	**-0.32**
OTDD (UMAP)	**-0.37**	**-0.42**	**-0.45**	-0.12
OTDD (PCA)	-0.2	-0.21	**-0.38**	-0.09
KL Approx (UMAP)	**-0.37**	**-0.51**	**-0.55**	-0.2
KL Approx (PCA)	-0.2	-0.2	**0.24**	**0.22**
KDE - L2 (UMAP)	**-0.39**	**-0.47**	**-0.56**	-0.16
KDE - TV (UMAP)	**-0.40**	**-0.49**	**-0.56**	-0.14
PAD - Linear (UMAP)	**-0.43**	**-0.60**	**-0.62**	**-0.26**
PAD - RBF (UMAP)	**-0.40**	**-0.49**	**-0.57**	-0.15
PAD - Linear (PCA)	0.04	0.05	0.01	0.04
PAD - RBF (PCA)	**0.42**	**0.40**	**0.39**	**0.60**

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Broadly, where statistically significant at the 95% confidence level the pattern of results shown in table 8 suggest a negative relationship between our metrics and transfer success rate. As discussed in Section 6.3, all of our metric implementations are dissimilarity metrics and thus these results suggest a positive relationship between dataset similarity and transfer success, in line with hypothesis 3.

This pattern broadly holds across MMD, OTDD, KDE, and PAD metrics. This pattern also holds across embeddings including with the raw data, with the sole exception of the Inception-only and the Inception plus PCA embeddings. Where using these embeddings, the results for MMD, approximate KL divergence, and PAD, the result are flipped in direction where still significant at the 95% confidence level. Notably, where using these embeddings the OTDD metric appears robust to this change in direction.

The flipped direction on these metrics in particular is consistent with our earlier assessment of the construct and criterion validity of many of these metrics when using the Inception-only and Inception plus PCA embeddings. Their failure to predict successful transfer learning and negative correlations leads us to disregard these results in assessing hypothesis 3.

Table 9 presents the second set of results for the fast gradient attack, this time with mean loss increase as the transfer success metric.

Table 9: Fast Gradient Attack - Mean Loss Increase

Similarity Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
MMD (None)	**-0.59**	**-0.64**	**-0.61**	**-0.48**
MMD (Inception)	**0.22**	**0.35**	**0.23**	0.11
MMD (UMAP)	**-0.69**	**-0.60**	**-0.57**	**-0.60**
MMD (PCA)	-0.05	0.11	-0.02	-0.14
OTDD (None)	**-0.61**	**-0.75**	**-0.74**	**-0.44**
OTDD (Inception)	**-0.52**	**-0.46**	**-0.56**	**-0.52**
OTDD (UMAP)	**-0.52**	**-0.42**	**-0.39**	**-0.49**
OTDD (PCA)	**-0.33**	**-0.24**	**-0.35**	**-0.36**
KL Approx (UMAP)	**-0.61**	**-0.50**	**-0.48**	**-0.45**
KL Approx (PCA)	**-0.30**	**-0.24**	0.2	0.1
KDE - L2 (UMAP)	**-0.59**	**-0.49**	**-0.50**	**-0.53**
KDE - TV (UMAP)	**-0.60**	**-0.50**	**-0.49**	**-0.53**
PAD - Linear (UMAP)	**-0.66**	**-0.59**	**-0.55**	**-0.58**
PAD - RBF (UMAP)	**-0.60**	**-0.51**	**-0.50**	**-0.53**
PAD - Linear (PCA)	0.07	0.07	0.01	0.05
PAD - RBF (PCA)	**0.31**	**0.40**	**0.46**	**0.24**

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Table 9 corroborates the earlier results in table 8 showing the correlation between dataset similarity and transfer success rate in the case of the fast gradient attack. Our results for the fast gradient attack are thus not sensitive to the choice of attack success metric, and hold for both success rate and mean loss increase.

As before, across MMD, OTDD, approximate KL, KDE, and PAD dataset similarity metrics, where statistically significant at the 95% confidence level the correlations are negative, giving further evidence to support hypothesis 3.

However, also as before, most metrics using the Inception-only and Inception plus PCA embedding show the same reversed direction in results. Likewise, their earlier poor performance in assessments of their validity again leads us to disregard these results in assessing hypothesis 3.

Table 10 presents the correlations for success rate for the boundary attacks.

Table 10: Boundary Attack - Success Rate

Similarity Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
MMD (None)	**0.69**	**-0.50**	**-0.47**	0.18
MMD (Inception)	-0.15	**0.49**	0.01	**0.24**
MMD (UMAP)	**0.53**	-0.09	**-0.59**	**0.68**
MMD (PCA)	0.06	**0.43**	-0.16	**0.39**
OTDD (None)	**0.27**	**-0.63**	-0.12	-0.12
OTDD (Inception)	**0.36**	-0.0	**-0.31**	**0.36**
OTDD (UMAP)	**0.41**	0.04	**-0.47**	**0.64**
OTDD (PCA)	**0.34**	0.11	**-0.33**	**0.35**
KL Approx (UMAP)	**0.58**	-0.01	**-0.48**	**0.54**
KL Approx (PCA)	**0.30**	0.03	0.18	-0.04
KDE - L2 (UMAP)	**0.49**	-0.01	**-0.52**	**0.61**
KDE - TV (UMAP)	**0.51**	-0.03	**-0.55**	**0.66**
PAD - Linear (UMAP)	**0.54**	-0.14	**-0.57**	**0.64**
PAD - RBF (UMAP)	**0.51**	-0.03	**-0.54**	**0.65**
PAD - Linear (PCA)	-0.04	0.02	0.13	0.01
PAD - RBF (PCA)	0.21	**0.29**	-0.13	**0.27**

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

With table 10, our results begin to diverge from those reported for the fast gradient attack. When using attack images based on the distribution of the dataset the target model was trained on (inner columns of the table), the results are broadly as before (negative, except those using the Inception plus PCA embedding).

However, when using images from the surrogate distribution, the relationship is 'flipped': the correlations become positive, implying that greater *dissimilarity* is associated with transfer attack success, thus completely contradicting hypothesis 3.

We speculate based on this result that it may be the case that dataset similarity plays a role in two ways. First, greater similarity leads to a more similar function being learned in the surrogate in all cases. This is useful for an attacker to better approximate the target model. This result is evidenced by the fact that when using adversarial examples generated on the target model's distribution (inner columns), we observe the same relationship as before.

However, from here, for some attacks, using out of distribution images may result in greater success as they are already harder for the target model. This dynamic may drive the strange results in the outer columns for boundary attacks in the case of transfer success rate.

We note that most of the Inception-only and Inception plus PCA results (other than with OTDD) are not statistically significant for the attacks based on the surrogate distribution. However, a small number are, and have not changed direction, adding to the puzzle of the behaviour of the embeddings based on PCA. Again, we ultimately conclude that based on their poor validity, we may disregard these results.

Table 11 is our final table for hypothesis 3, and presents the correlations with mean loss increase in the case of boundary attacks.

Table 11: Boundary Attack - Mean Loss Increase

Similarity Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
MMD (None)	**-0.37**	**-0.43**	**-0.51**	-0.05
MMD (Inception)	**0.36**	**0.43**	**0.22**	0.07
MMD (UMAP)	**-0.31**	-0.06	**-0.51**	-0.16
MMD (PCA)	**0.24**	**0.39**	0.0	-0.03
OTDD (None)	**-0.65**	**-0.62**	**-0.49**	-0.08
OTDD (Inception)	-0.17	-0.02	**-0.39**	-0.13
OTDD (UMAP)	-0.2	0.06	**-0.42**	-0.14
OTDD (PCA)	0.03	0.11	**-0.26**	-0.11
KL Approx (UMAP)	-0.19	0.02	**-0.42**	-0.03
KL Approx (PCA)	0.01	0.03	**0.23**	0.02
KDE - L2 (UMAP)	-0.18	0.0	**-0.41**	-0.11
KDE - TV (UMAP)	-0.2	-0.01	**-0.43**	-0.1
PAD - Linear (UMAP)	**-0.31**	-0.1	**-0.51**	-0.14
PAD - RBF (UMAP)	-0.2	-0.01	**-0.43**	-0.11
PAD - Linear (PCA)	0.1	0.06	0.1	0.1
PAD - RBF (PCA)	**0.40**	**0.30**	**0.31**	0.17

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

With table 11 our results partially return to their earlier pattern. Where significant at the 95% confidence level, metrics other than those using the Inception-only or the Inception plus PCA embeddings (again, excepting OTDD) are negative.

This gives further overall support to hypothesis 3. However, there are a large number of null results in this table, which should be taken into account. It is broadly only the MMD and OTDD metrics without embeddings that provide a large amount of support to hypothesis 3 for this table. All other metrics (not previously ruled out as possible to disregard given poor validity and strange results earlier on) where significant are only significant in one or two of the columns.

We note in particular the B to A boundary attacks based on the distribution of B, which is typically those with the transforms applied. For all metrics we observe a null result in this column We do not know why this has produced an overall null result (if again ignoring the PAD with Inception plus PCA). Given the construction of our experiments, the fact that dataset B is always that with a transform applied to it if any could be the main driver of this large null result.

The difference between table 10 and table 11 mean that for boundary attacks, our attack success metrics do not entirely agree with one another. Bearing this in mind, the success rate counts only previous successes for the model being converted into failures, and so is arguably the more relevant metric to consider. It may be that more out of distribution images are already close to the Boundary, hence did not require as much increase in loss, thus leading to the large number of null results evidence here and the fact that we observe the same negative correlations as for the fast gradient attacks in the outer columns.

Appendix D presents visualisations of the relationships discussed for hypothesis 3.

7.5 Hypothesis 4

As with our earlier hypotheses, we use the same four column table format to present our results for hypothesis 4. Here, we have attack success metrics on rows, and two tables - one for each kind of attack approach.

To capture the notion that datasets with similar sizes should see more successful transfer attacks, we took the absolute difference in size between both surrogate and target as our measure. As this increases in size as the size difference between datasets increases, this means negative correlations will be in line with our theory and hypothesis.

Table 12 presents the first set of results for hypothesis 4. These are the correlations between absolute difference in dataset size between target and surrogate, and transfer success rate in the case of fast gradient attack.

Table 12: Dataset Size, Absolute Difference - Fast Gradient Attack

Attack Success Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Success Rate	**-0.44**	**-0.28**	**-0.36**	**-0.40**
Mean Loss Increase	**-0.29**	**-0.30**	**-0.41**	-0.2

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Here, all results bar one are negative and significant at the 95% confidence level. It is therefore clear that across attack success metrics, we find evidence for hypothesis 4.

Table 13 presents the same results in the case of boundary attack.

Table 13: Dataset Size, Absolute Difference - Boundary Attack

Attack Success Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Success Rate	**-0.22**	-0.09	0.07	-0.01
Mean Loss Increase	**-0.30**	-0.09	**-0.26**	-0.02

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Unlike table 12, only three out of eight correlations are statistically significant at the 95% confidence level, although all three are negative. Notably, two of the three belong to the mean loss increase. The evidence for hypothesis 4 is thus much milder in the case of boundary attack, although it does not fully contradict the hypothesis.

While our theory emphasises the similarity in terms of the size of the datasets, we also re-ran the same analyses, albeit changing the measure from absolute difference to the size of the target subtracted from the size of the surrogate, and the ratio of the size of the target over the size of the surrogate. The goal here was to discover if the surrogate dataset being larger was also predictive in some way of transfer attack success.

These results are presented and analysed in appendix E, but broadly support the notion that this is true in the case of boundary attack - although the precise relationship remains ambiguous. We also present visual representations of the results discussed here in appendix E.

7.6 Dataset Similarity Using the Target Model

We carried out further analysis by using the models trained for the experiments to develop an additional similarity measure. The results from this provide additional support for hypotheses 3 and 4.

7.6.1 Test Accuracy Ratio as an Approximator of Similarity

Figure 4 shows the accuracy of the A and B test datasets on each of the models prior to any transfer attacks. These accuracies were computed as part of the transfer attacks and so were available for use without further work. Each point on the plot represents one experiment group. The model's accuracy on its own test set is on the x-axis, and the model's accuracy on the other test set in the experiment pair is on the y-axis. If both datasets in an experiment pair have the same accuracy, then the point will fall on the dashed line.

Comparing the accuracy of the two datasets in an experiment pair using one model provides an implicit assessment of the similarity of the datasets. This is because as the datasets become more similar, the model will classify both datasets with a similar accuracy. If the datasets are very different, then it is reasonable to

expect that the model will classify its own test dataset with a higher accuracy than the one that is out of distribution. Given our prior discussion and results, it is clear that dataset similarity contributes to the overall similarity of learned models. We can thus assume that datset similarity will at least partially drive the closeness in overall accuracy.

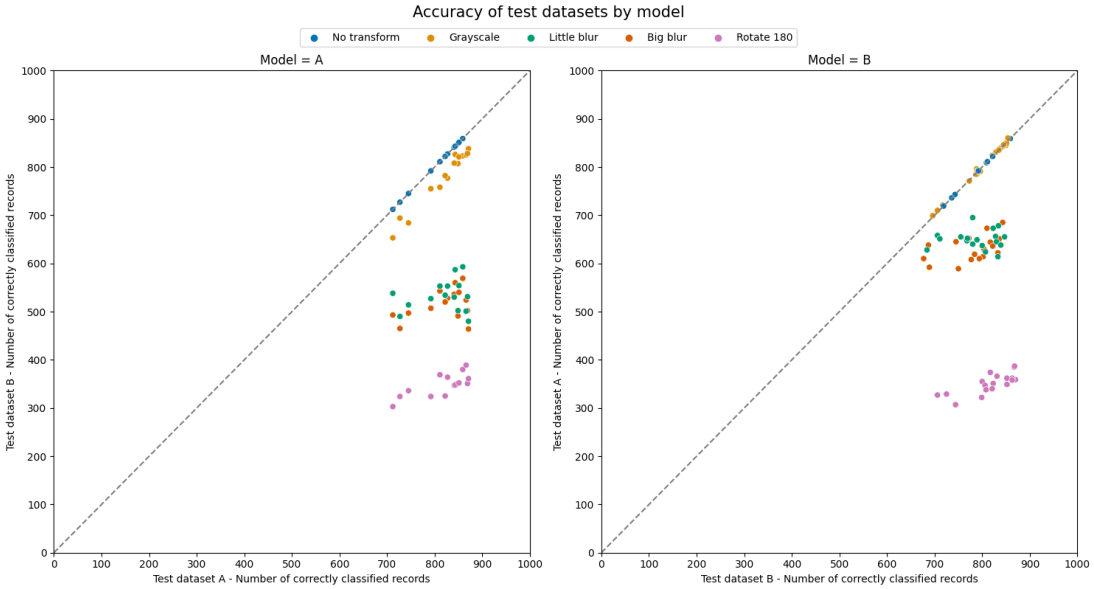

Figure 4: Accuracy of test datasets on A and B on each of the models, grouped by the transform applied to dataset B. If datasets A and B result in the same accuracy, then the point will fall on the dashed line.

The plots show that, broadly, the experiment groups without any transforms are always close to or on the dashed line. When the model trained on dataset B has been trained on grayscale images, it can classify coloured images from the test dataset A with similar accuracy. However, when the model trained on dataset A is used to classify grayscale images (which is was not trained on as it is always dataset B that undergoes the transforms) it is not able to classify them with as high accuracy as the colour images. This indicates that the clear separation of transforms into group B of the datasets may drive some differences in results between those groups discussed earlier in this section.

The plots further show that the models are most dissimilar in their classifications of the experiment pairs with the rotate 180 group of transforms, as the accuracy of the out of distribution dataset is far lower than the in distribution dataset in both cases. Both models also show some dissimilarity between the original images and the images with blurred transforms. As with the grayscale transforms, when model B has been trained on the blurred images, it is more successful at classifying the original images than the model trained on the original images (model A) is at classifying blurred images.

7.6.2 The Role of Dataset Size

Figure 5 shows the same plots, but grouped by the drop combinations to demonstrate how the learned models can be affected by the size of the training dataset.

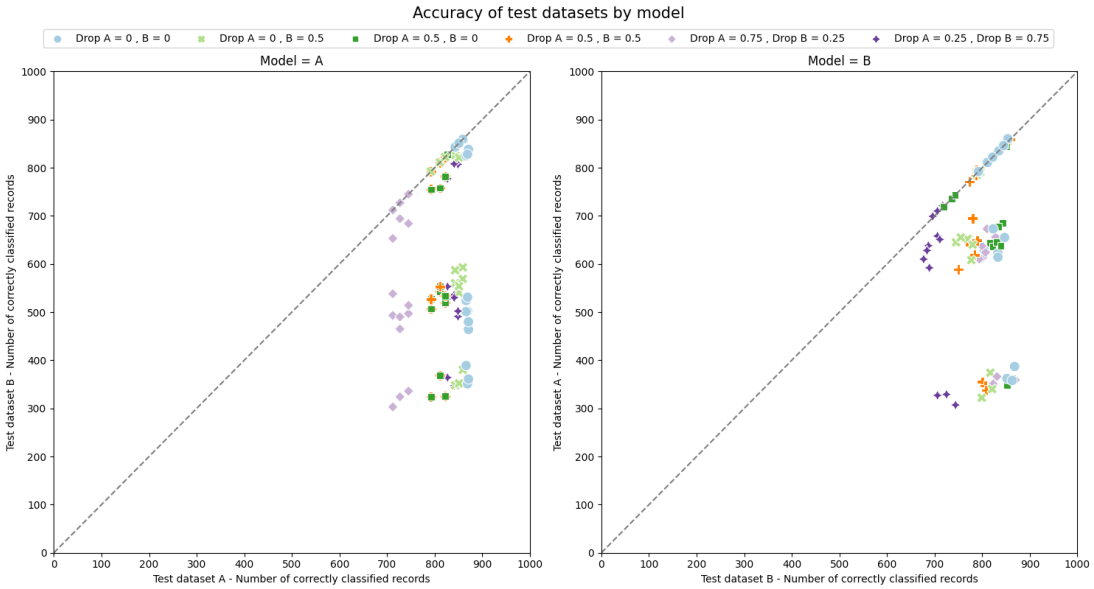

Figure 5: Accuracy of test datasets on A and B on each of the target models, shown by drop combination. If datasets A and B result in the same accuracy, then the point will fall on the dashed line.

The plots in figure 5 show that within the three clusters, variation in both model accuracy on the test datasets can be driven in part by training dataset size. On the model's own test set, this relationship is clear in all three clusters. On its partner's test set however, the surrogate accuracy has the strongest relationship with size for the no transform and grayscale cluster on or near the line. This implies that for hypothesis 4, dataset similarity may be a necessary condition for dataset size to have a full impact.

7.6.3 Accuracy Ratio Measure

Given the pattern of results shown in figures 4 and 5, it follows that the ratio of test dataset accuracies is in itself a measure of the similarity of both datasets. Using the target model from our experiments, we construction an additional measure of similarity:

$$\text{Accuracy Ratio Measure} = 1 - \frac{\text{Accuracy of surrogate distribution}}{\text{Accuracy of target distribution}} \tag{16}$$

As with the metrics introduced in Section 5, this measure captures the *dissimilarity* of both datasets. A value close to zero shows that the model has performed similarly on both test datasets, and so in turn suggests that they are drawn from similar distributions. Likewise, greater values (i.e. above 0) indicate that the model has performed worse on the surrogate test dataset, and so suggests that the datasets are drawn from more dissimilar distributions.Technically, a value less than zero could be achieved if the surrogate accuracy is higher than the target accuracy, although in practice this is rare (and where it does occur, is very close to the line on the above figures). As with the KL divergence, this is not technically a metric as it is not symmetric. That is, the target model similarity for two datasets is not necessarily the same for both target models.

One key difference between the accuracy ratio measure and the metrics used earlier in this work is that the other metrics do not use a model trained on the training set of the target dataset. Because this measure does use this model, it implicitly accounts for the size of the dataset in its measurement of similarity. As the models in this work all have the same architecture and are trained for the same number of epochs, the learned parameters of the model are likely to differ between a model trained on 25% of the dataset compared to 100% of the dataset. The target model dataset similarity is the only measure that is capable of reflecting this in its value.

Figure 6 shows the accuracy ratio measure for the target model against the transfer attack success rate for fast gradient attacks. There are four plots: one for each attack direction and distribution of test images for that

attack direction. Each plot has the transfer attack success rate on the x-axis, and the accuracy ratio measure on the y-axis. The points an each plot are grouped by the transform applied in each experiment group.

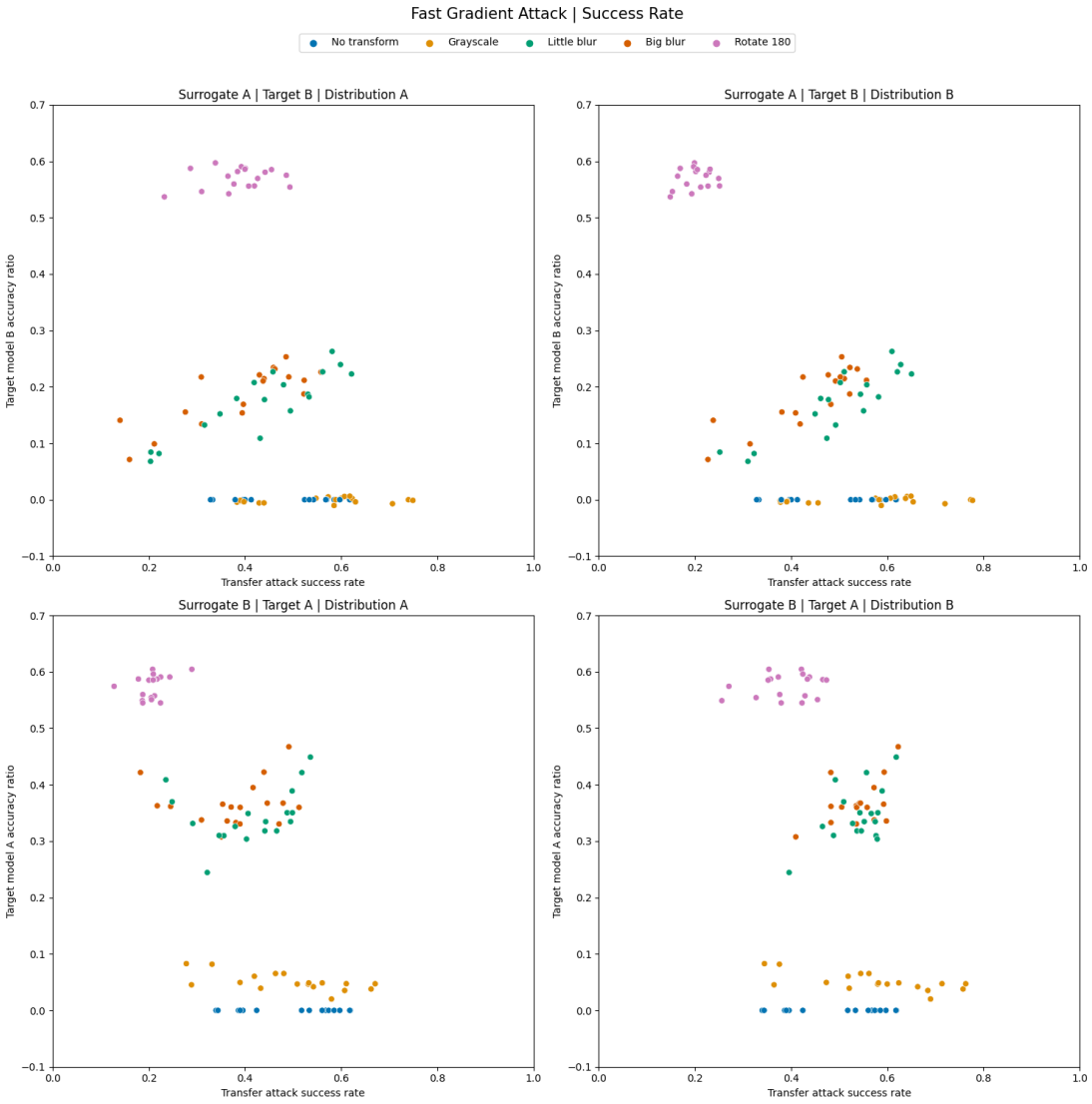

Figure 6: Accuracy ratio measure computed for the target model against transfer attack success rate for fast gradient attacks grouped by the transform applied to dataset B.

Figure 6 reveals a fascinating pattern of results. *Overall* and in line with hypothesis 3 there is a clear negative relationship between the measure and transfer attack success, in line with out earlier results. However, three distinct clusters of results emerge in each of the plots, showing interesting patterns of within-transform-group variation.

There is wide variation in the transfer attack success from around 30% to around 80% of the no transform and grayscale transform groups. Within the blur groups, the relationship with dataset similarity appears to be inverted: increased dissimilarity *correlates* with increased transfer attack success. Finally, for the rotate-180 transform group which is the most dissimilar, the results cluster in a low similarity (high dissimilarity) low transfer attack success corner of the plot.

To further explore the relationship of the accuracy ratio measure with dataset size, the same plots are shown in Figure 7, but points are instead grouped by drop combination.

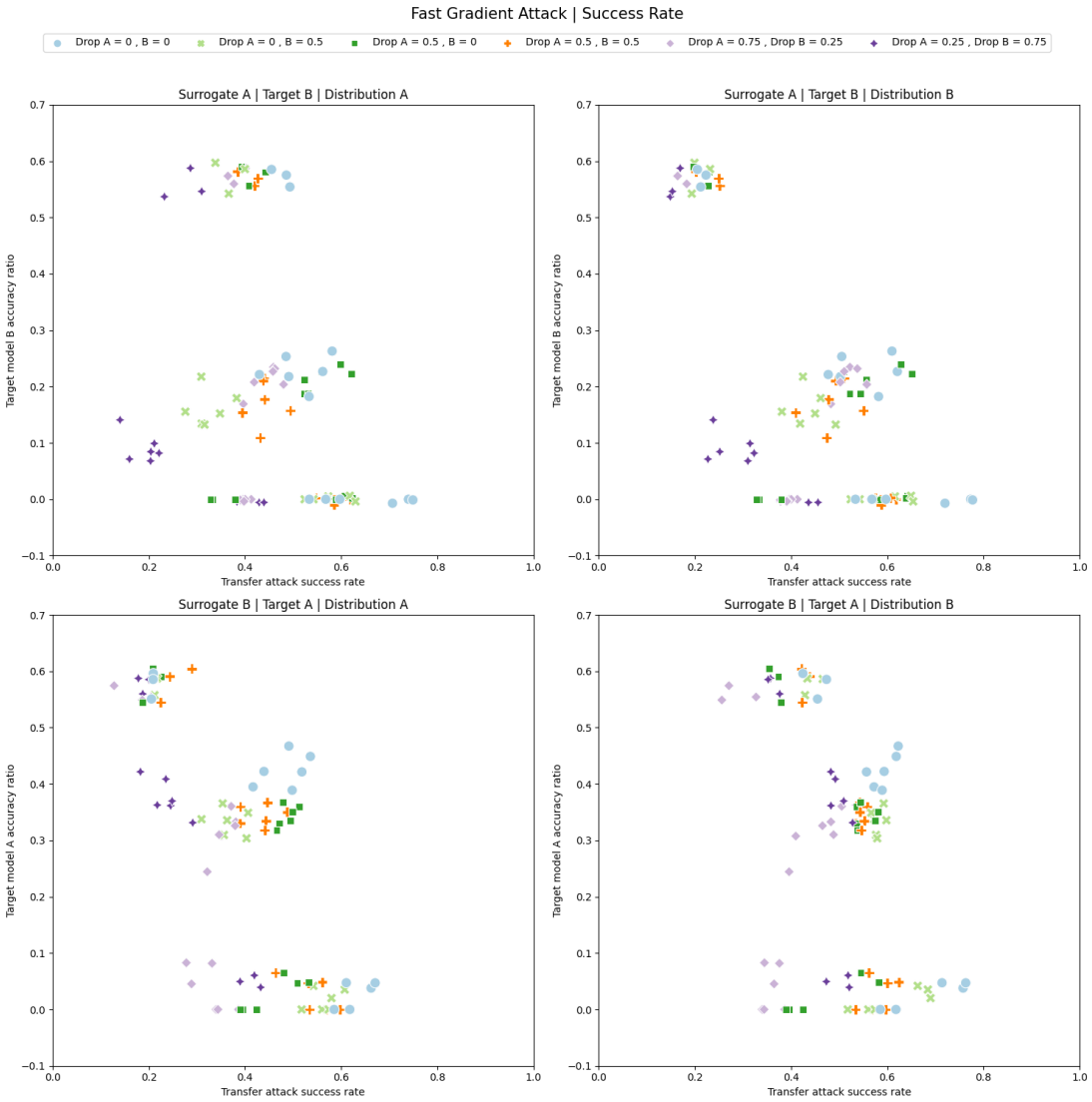

Figure 7: Target model similarity against transfer attack success rate for fast gradient attacks grouped by the drop combination of datasets A and B.

Figure 7 offers some explanation for the within-transfer-group patterns observed in figure 6. Consistent with hypothesis 4 transfer success rate is typically at its highest for the groups where no data has been dropped, while it is at its highest for the drop groups with a 75:25 ratio in either direction. Notably, the precise order shifts depending on which dataset is the surrogate or target: interestingly, more data dropped from the target dataset appears to correlate both with the accuracy ratio measure and with transfer attack success. Overall, this pattern of results suggests that dataset similarity and dataset size do in fact interact with one another in determining the success of transfer attacks.

Figure 8 presents the same set of results for boundary attacks, with points once again coloured by transform group. The different scales on the x-axis to the earlier figures should be noted - the boundary attacks were in general less successful than the fast gradient attacks. This is in line with the general pattern of model vulnerabilities discussed in appendix C.

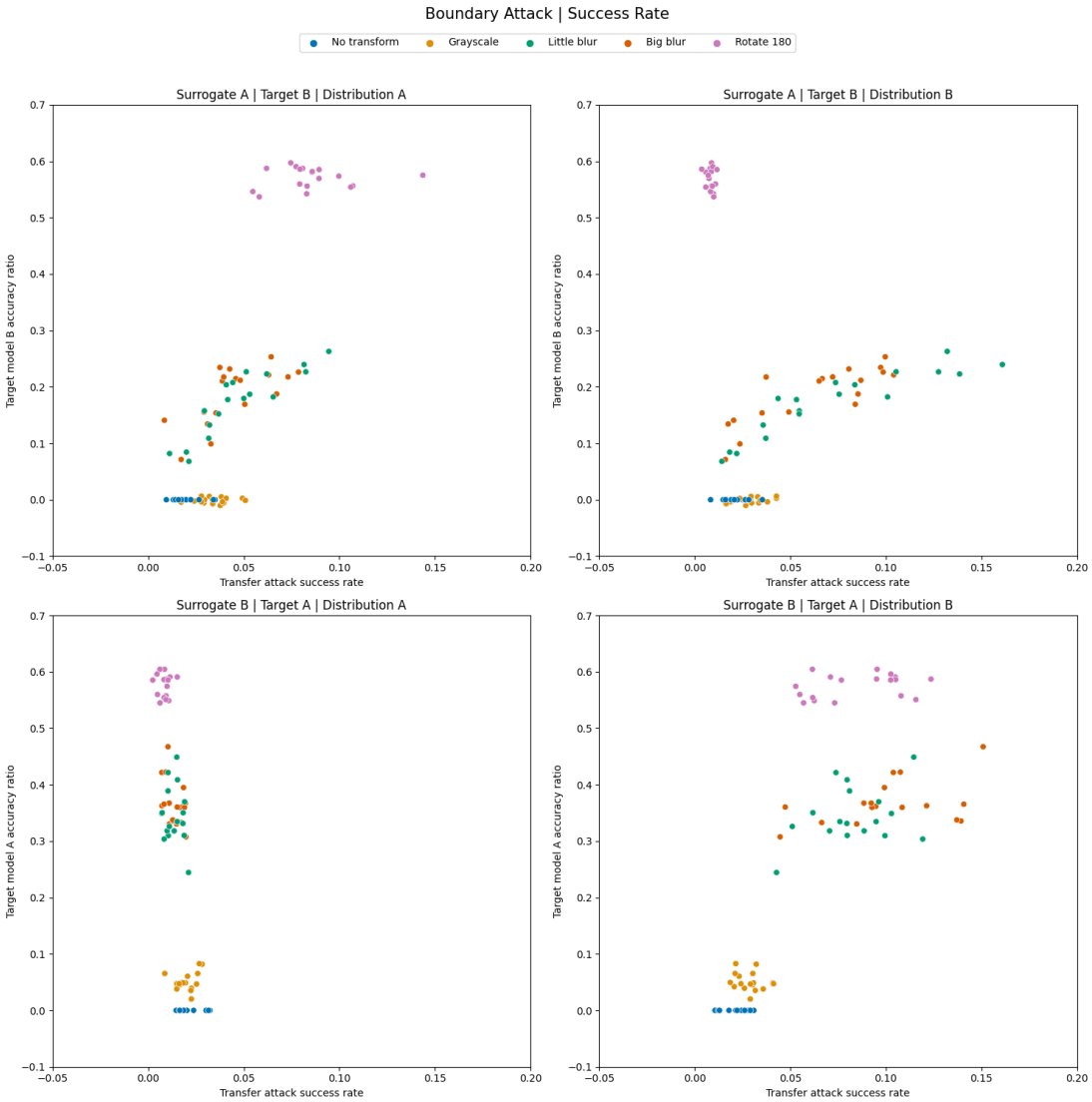

Figure 8: Target model similarity against transfer attack success rate for boundary attacks grouped by the transform applied to dataset B.

Similar to the correlations shown in table 10, overall pattern of relationships here is inverted and the measure is positively correlated with transfer attack success where the attack images come from the surrogate model's distribution. Where the images come from the target model's distribution, the relationship is similar to those relationships in table 10 the same as the earlier results (i.e. overall negative relationship), although the exact pattern appears different where A is the target model.

Figure 9 repeats the visualisation for boundary attacks, but as before groups points by drop combinations.

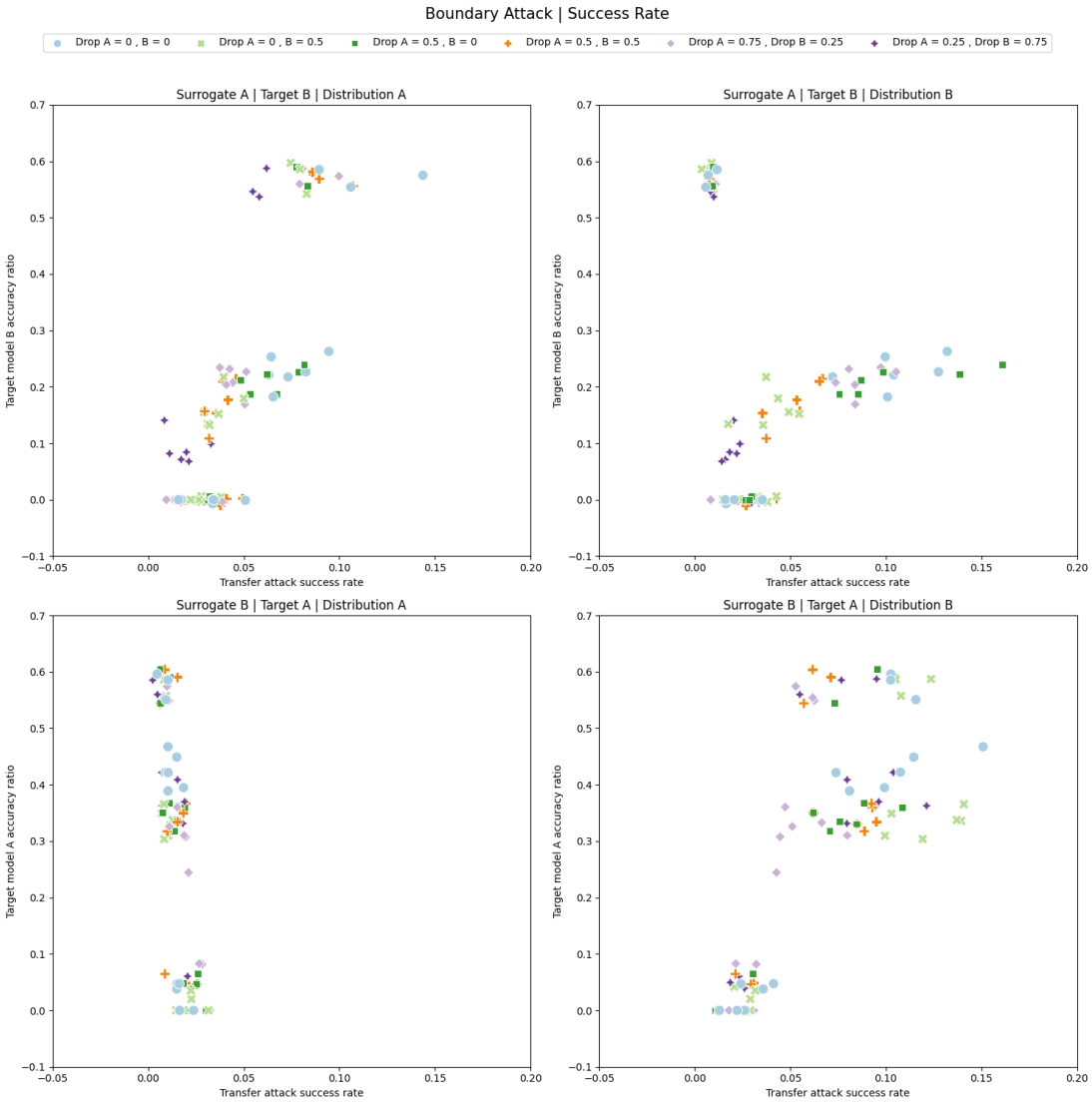

Figure 9: Target model similarity against transfer attack success rate for boundary attacks grouped by the drop combination of datasets A and B.

A similar relationship between size and within-transfer-group variation in transfer attack success also exists here, although it is less clear, perhaps due to the lower transfer success rates of boundary attacks in general.

In conclusion, while the target model dataset similarity provides support for hypotheses 3 and 4, it also demonstrates that there is an interaction between dataset size and similarity.

8 Conclusions

In this section, we summarise the conclusions from our results.

8.1 Validity and Performance of Similarity Metrics

Our assessment of our similarity metrics led to us concluding that broadly, all of the metrics clearly possess construct, content, and criterion validity. However, the use of embeddings may negatively impact the validity of some metrics.

In particular, our Inception-only and Inception plus PCA embeddings resulted in poor validity, and strange results following our assessment of validity. We therefore recommend careful consideration before using embeddings in conjunction with dataset similarity metrics.

We provide in the appendix some results and discussion of why counter to what some data scientists may expect the raw pixel values are sufficient for dataset similarity metrics.

Notably, the Inception plus UMAP embedding did not cause the same validity issues or strange results, and the OTDD metric was entirely robust to these issues.

8.2 Hypotheses 1 and 2

1. Target model vulnerability to attacks directly correlates with attack transferability
2. For a given target model, the surrogate model's vulnerability to attacks negatively correlates with attack transferability

First, as in the Phase 1 report [5] we find evidence that target model vulnerability predicts transfer attack success. In our results this is true where boundary attacks are being used, and true for fast gradient attacks where the metric of interest is mean loss increase. However, we find weaker but nonetheless statistically significant correlations in the opposite direction for fast gradient attacks where we use attack success rate as the metric of choice.

We speculate that this reversal is driven by two things. First, in our experiments while dataset similarity and dataset size are independent of one another (and broadly any other variable) by construction, this is not true of model vulnerability. This is in part a product of variables we have manipulated such as the transforms applied and dataset size, and so the correlations presented cannot be interpreted as causal in the way later correlations in the report are. In particular, since in the no-transform and grayscale transform groups datasets and thus learned models are similar and so have similar vulnerabilities, this could drive the inverted relationship. We do not know why these transform groups in particular showed higher vulnerability, but this would plausibly have been enough to produce the overall correlation.

Second, alongside the above reasoning (which applies across results), it may be that the general relationship of target vulnerability with transfer attack success is weaker in the case of the fast gradient attack, and hence despite the similar pattern of vulnerabilities across experiment groups we only saw the inverted relationship for fast gradient attack in terms of success rate. Due to time constraints, we have not further investigated potential drivers of this result further, but overall we do not take the view that this reversed relationship suggests that increased target model vulnerability decreases the likelihood of transfer attack success.

We therefore conclude overall in favour of hypothesis 1 for boundary attacks, but find mixed evidence in the case of fast gradient attacks.

As for hypothesis 2, our results were mixed, with significant correlations pointing in both negative and positive directions without any particular overall pattern. We therefore conclude against hypothesis 2 and do not take the view that surrogate model vulnerability is particularly predictive of transfer attack success.

8.3 Hypothesis 3

3. Higher dataset similarity increases transfer attack success

Overall, we find evidence to confirm hypothesis 3, though not without caveats. While the overall evidence is clear that greater dataset similarity predicts transfer attack success, this relationship is inverted for boundary attacks when using images from the surrogate model's training distribution to create the adversarial images. Mean loss increase largely saw null results for the boundary attack, with some results still in line with our hypothesis. Notably, the results that still supported our hypothesis were the MMD and OTDD metrics without embeddings. All B to A attacks using the distribution of B for the adversarial images were however null results.

We speculate that in all cases greater dataset similarity leads to more similar functions being learned during model training. However, for the boundary attack success is overall more likely in the first place given that

the image in question is already out of distribution. This is because it requires less work to be moved over the boundary.

8.4 Hypothesis 4

4. The closer two datasets are in size, the more successful transfer attacks between models learned from them will be

Finally, we find evidence in favour of hypothesis 4 in the case of fast gradient attacks. The absolute size difference between surrogate has a mild but clear negative effect on transfer attack success. We do not reach the same conclusion for boundary attacks as the available evidence was much weaker. However, in our appendices we do find evidence that dataset size has an impact on transfer attack success for boundary attack, albeit along different lines to those predicted by hypothesis 4.

8.5 Target Model Accuracy Ratio

We further develop an additional measure labelled 'accuracy ratio measure', where we use the target model to calculate the accuracy of the surrogate and target test datasets. We then subtract the ratio of the surrogate dataset accuracy to the target dataset accuracy from one. This differs from other similarity metrics presented in this report as instead of exploiting the empirical distribution of the datasets or how easily they can be discriminated between, and instead examines the accuracy of the test datasets using the target model. In practice, this captures not just dataset similarity, but also the size of the target training dataset as it exploits the learned function rather than the datasets in and of themselves. It therefore speaks to both hypotheses 3 and 4.

Plotting transfer attack success against this measure reveals some interesting results. As with other similarity metrics, the overall relationship is negative. However, unlike other metrics, within-transform-group (i.e. holding any data transformations constant other than the dropping of data from the datasets) variation emerges when considering the relative sizes of datasets. Second, the results appeared to show within-transform-group changes in transfer effect across combinations of dataset similarity and size, suggesting an interaction between the two attributes of a dataset in driving transfer attack success. This serves to demonstrate the importance of understanding which parts of a dataset a similarity metric exploits.

8.6 Limitations

Our conclusions do however have some clear limitations which should be made explicit. First, we cannot say whether our results will generalise beyond the experiments we have conducted. Our primary method of controlling relevant variables was by not varying them, instead of randomly manipulating them. This had the advantage of preventing scope creep and ensuring the project remained manageable, but comes at the cost of meaning our results are particular to the case of CIFAR-10.

CIFAR-10 is of course a relatively simple dataset, with a low number of features (compared to many real-world datasets) and just ten classes. As these variables are changed, the relationship between dataset similarity and transfer attack success may change. In particular, we suspect that our confirmation of hypothesis 4 is least likely to generalise - especially as datasets become substantially larger and underfitting becomes less of a concern.

Second, our experiment design has allowed one form of imbalance to creep in: we applied all of the dataset transformations on the 'B' datasets. This means that were results to do differ between transfer attack directions, we cannot rule out the role of the transforms.

Finally, we have only examined only two kinds of attack: the fast gradient attack and the boundary attack. While our results between these attacks have mostly corroborated one another, some differences have emerged, implying that the relationships we have explored are partly conditional on how the adversarial attack is constructed. Reproducing the analysis across a wider range of attack types may reveal greater insights into how changing the attack method may affect the relationship between dataset similarity and transfer attack success.

8.7 Considered but not Pursued

We considered, but ultimately did not pursue, several possible extensions of the work presented here. In all cases, we believed the additional work may have been interesting but did not add enough to the central goal of the project to justify the additional work involved.

First, we considered calculating each metric by class (i.e. calculate a value for the observations belonging to the same class between both datasets, for each class). This may have helped decompose exactly what each metric was picking up on from each dataset.

Second, we considered introducing a second set of experiments with a second dataset, such as MNIST. We felt this would have given further confirmation to our results and would have allayed some of our concerns around generalisation, but we ultimately did not pursue this due to time constraints.

Finally, we considered introducing further transforms. This would have provided more variation in datasets and thus in dataset similarity. Ultimately, we felt we had enough data points to avoid type 2 errors in our hypothesis testing and so did not pursue further transforms.

9 Recommendations

In this section, we describe the main recommendations we believe follow from the results and conclusions of this report.

9.1 Protection of Information Regarding Model Training Data

In light our results, our first and most important recommendation is to stress the importance of protecting information regarding the dataset a model is trained on as a policy consideration. The primary angle through which our results reach this conclusion is the positive relationship between dataset similarity and transfer attack success. If an attacker has some idea of the data a model was trained on, they will be better able to construct a more similar model to generate their attacks on.

A second angle that may emerge from our results however is that model owners also need to prevent attackers from discovering which examples are out of distribution for their training data. This is displayed both by the relationship between base transfer success and the similarity metrics; and by the difference in results for the boundary attack when using success rate as the metric in question - basing attack images on the surrogate distribution of images inverted the relationship between dataset similarity and transfer attack success.

This dynamic - using similar datasets to train a surrogate, then using dissimilar images to achieve transfer attack success - may be worth exploring in future work. After all, the use of dataset information to an attacker is to inform them how they can cause a target model to misclassify an image. Whether this is achieved by generating adversarial examples, or passing difficult to classify examples for the network is not of as much concern to an attacker as the overall likelihood that an image will be misclassified.

In the age of foundation models, this recommendation will likely become increasingly important. Just as foundation models mean that deployed models will share the same biases [3], so too will they share some common vulnerabilities. We highlight for example that many image foundation models are trained on ImageNet. In line with the weak results of model similarity in the Phase 1 report [5], variation in model architecture is unlikely to be sufficient mitigation against adversarial attacks.

We stress that we recommend these points as considerations, not directly as policy. Policy-makers should weigh the risks of making model information open, or the risks of using an open foundation model, against the benefits that come with openness. There will be no one-size-fits-all policy.

9.2 Improving Target Model Vulnerability

Alongside our main recommendation, we also re-confirm with some caveats the finding of the Phase 1 report that a simple focus on reducing intrinsic model vulnerability is clearly beneficial to reducing the risk of successful transfer attacks.

9.3 Dataset Similarity for Assessing Domain Adaptation and Transfer Learning

We hope that our theoretical discussion in Section 3 illustrates the ways in which different attributes of a dataset drive both learning and transfer attack success. These metrics however are also relevant to the claims many papers make regarding domain adaptation and (zero-shot) transfer learning. Dataset similarity metrics give one means of assessing how dissimilar two domains really are, and thus how impressive particular claims being made are.

9.4 Generalisability of Dataset Similarity Metrics

With this in mind, we conclude with a final comment on the generalisability of some of the dataset similarity metrics. While optimal transport dataset distance (OTDD) has arguably been the stand-out metric given its robustness to the pattern of poor results shown when using the Inception plus PCA embedding (MMD also showed strong results, although less robustness to embedding choice), we assess that all of our metrics performed reasonably well and have good validity conditional on the choice of embedding.

We wish however to highlight an additional consideration beyond performance in this report. Almost all of these metrics either have a hard requirement of shared features (e.g. Proxy-A Distance cannot be run without them), or a soft requirement in that while an operation is possible in numeric terms it must still pass unit analysis in order to be a valid operation.

To understand this point, consider attempting to take a distance between a length measured in meters and some volume of liquid measured in litres. Such an operation is clearly nonsensical. This requirement exists for all similarity metrics presented in this report. However, optimal transport (OT) can be implemented as a Gromov-Wasserstein distance (as opposed to a Wasserstein distance). This distance does not have this requirement. As the creators of the OTDD note, it can also be adapted in this direction [1].

We thus conclude our recommendations by suggesting that optimal transport similarity measures - both raw and in the form of OTDD - represent the most promising avenue for the measurement of dataset similarity.

Acknowledgements

The computations described in this research were performed using the Baskerville Tier 2 HPC service (https://www.baskerville.ac.uk/). Baskerville was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham.

References

[1] David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020.

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

[3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[4] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012.

[5] Alena Frankel and Marcos Charalambides. Model similarity for attack transferability. *Applied Research Centre for Defence and Security*, 2021.

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[7] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

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

[9] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[10] Robin Hirt, Akash Srivastava, Carlos Berg, and Niklas Kühl. Sequential transfer machine learning in networks: Measuring the impact of data and neural net similarity on transferability. *arXiv preprint arXiv:2003.13070*, 2020.

[11] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, page 180–191. VLDB Endowment, 2004.

[12] Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[14] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[15] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[16] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[17] Michael Snow et al. The monge-kantorovich optimal transport distance for image comparison. *arXiv preprint arXiv:1804.03531*, 2018.

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

[19] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[20] Junqi Wang, Pei Wang, and Patrick Shafto. Efficient discretization of optimal transport. *Entropy*, 25(6):839, 2023.

[21] Qing Wang, Sanjeev R. Kulkarni, and Sergio Verdu. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, 2009.

A Optimal Transport

Optimal transport is the problem of transporting mass from one distribution into another in the most efficient way possible.

A.1 Monge Formulation

Where P_α and P_β are distributions over $\mathcal{X}_\alpha$ and $\mathcal{X}_\beta$, the Monge formulation of the optimal transport problem is to find a function $T : \mathbb{R}^d \to \mathbb{R}^d$ called the transport map, that achieves the infimum:

$$\inf_{T, T_\# P_\alpha = P_\beta} \int c(x, T(x)) dP_\alpha(x) \tag{17}$$

where c is some cost function and $T_\# P_\alpha = P_\beta$ denotes that P_β is the push-forward of P_α under T, that is every $x \in \mathcal{X}_\alpha$ maps onto exactly one $x' \in \mathcal{X}_\beta$. A map T^* that is a minimiser is called the *optimal* transport map. However, there is no guarantee that such a T^* will exist in this formulation.

A.2 Kantorovich Formulation

The Kantorovich formulation of OT overcomes the problem of the Monge formulation where a solution may not exist by allowing 'mass splitting'. The Kantorovich formulation of OT is to find a joint distribution, π with marginal distributions P_α and P_β, that achieves the infimum:

$$\inf_{\pi \in \Pi(P_\alpha, P_\beta)} \int c(x, x') d\pi(x, x') \tag{18}$$

Where $\Pi(P_A, P_B)$ denotes all the joint distributions π for $(\mathcal{X}_A, \mathcal{X}_B)$ with marginals P_A and P_B, that is $T_{X\#} J = P_A$ and $T_{Y\#} J = P_B$. This formulation allows every $x \in \mathcal{X}_A$ to be split and mapped to multiple locations in $\mathcal{X}_B$. A minimiser π^* will exist for the Kantorovich formilation and is called the optimal transport plan or the optimal coupling.

A.3 Wasserstein Distance

Where the cost function c in 18 is a distance metric, then the overall cost of optimal transport is itself a distance metric between P_α and P_β.

Where the cost function c is defined by the distance

$$c(x, x') = \|x - x'\|^p \tag{19}$$

then the distance becomes known as the p-Wasserstein distance, or W_p. Where for instance p is 2, this becomes the 2-Wasserstein distance or W_2. Many commonly-used metrics in data science such as the word mover's distance or Fréchet inception distance either simplify to or are applications of the p-Wasserstein distance.

A.4 2-Wasserstein Solution

We typically do not directly observe P_α and P_β, but instead have access to finite samples $x_a \in \mathcal{X}, x_b \in \mathcal{X}$ (e.g. the features of a dataset). 18 thus often cannot be directly computed in the continuous case [20].

However, the 2-Wasserstein distance between two Gaussian distributions has the analytic form [1]:

$$W_2^2(P_\alpha, P_\beta) = \|\mu_\alpha - \mu_B\|_2^2 + tr(\Sigma_\alpha + \Sigma_B - 2(\Sigma_\alpha^{\frac{1}{2}} \Sigma_\beta \Sigma_\alpha^{\frac{1}{2}})^{\frac{1}{2}}) \tag{20}$$

where μ_α and μ_β are the means of the Gaussian distributions P_α and P_β, Σ_α and Σ_β are the covariances of P_α and P_β, and $\Sigma^{\frac{1}{2}}$ denotes the matrix square root. One solution for computing OT is thus to use the sample means and covariance matrices to compute 20.

A.5 Discrete Solution

Another common solution is to discreteise the data [17, 20]. We compute histograms $\boldsymbol{a}$ and $\boldsymbol{b}$ on the probability simplex, where each value represents the mass in that particular bin. Of course, if the data are already discrete then this step need not be performed.

The discrete version of the problem can be described as attempting to find the transport matrix T between distributions $\boldsymbol{a}$ and $\boldsymbol{b}$ that minimises the total cost $\langle \boldsymbol{T}, \boldsymbol{C} \rangle$:

$$\min_{\boldsymbol{T} \in \Pi(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{C}, \boldsymbol{T} \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \boldsymbol{C}_{ij} \boldsymbol{T}_{ij} \tag{21}$$

where $\boldsymbol{C}$ is a cost matrix defining the cost of moving an element from one part of $\boldsymbol{a}$ to $\boldsymbol{b}$. The margins of $\boldsymbol{T}$ must be $\boldsymbol{a}$ and $\boldsymbol{b}$, and each element of $\boldsymbol{T}$ must be 0 or greater. This gives us the constraints:

$$\boldsymbol{T}\mathbb{1} = \boldsymbol{a} \tag{22}$$

$$\boldsymbol{T}'\mathbb{1} = \boldsymbol{b} \tag{23}$$

$$\forall\, i, j\ \boldsymbol{T}_{ij} \geq 0 \tag{24}$$

This problem can be solved as a linear program. In practice it is often solved with a regularisation term, but this goes beyond the scope of this appendix.

B Dataset Similarity Without Embeddings

When no embeddings are applied to the datasets, the similarity metrics use the pixel values to compare the datasets. Since pixel values on their own do not contain any semantic information but only do so in combination with one another and their positions, we explored in this appendix how this remains appropriate for use in measures of dataset similarity.

Figure 10 shows a grid of histograms for six pixels in the CIFAR-10 training dataset for each of the transforms applied in this work. The rows of the grid correspond to a particular pixel in the images, while the columns correspond to each of the transforms. The pixel values (shown on the x-axis) have been scaled so that they fall between zero and one. The y-axis show normalised values so that the total value of all bar heights sum to one.

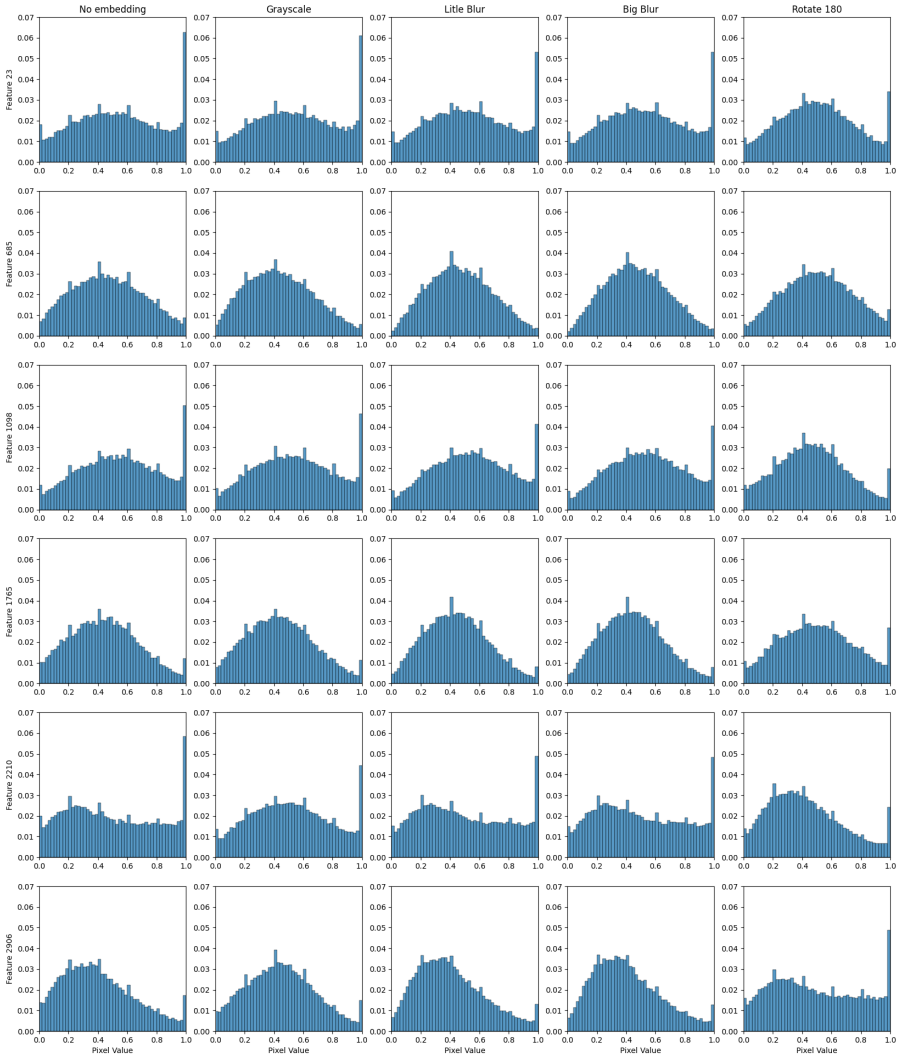

Figure 10: Histograms for six pixels in the CIFAR-10 dataset by transform

The figure shows how the transforms can affect the distribution of each pixel value, with some of the distributions being noticeably different between the transforms. It follows that any similarity metric capable of capturing the differences between these distributions will be capable of determining the differences between datasets.

Conversely, Figure 11 shows a grid of histograms for the same six pixel values without any transformations applied, but instead with an increasing number of records dropped from the training dataset. As before, the pixel values are scaled and the y-axis show the normalised values.

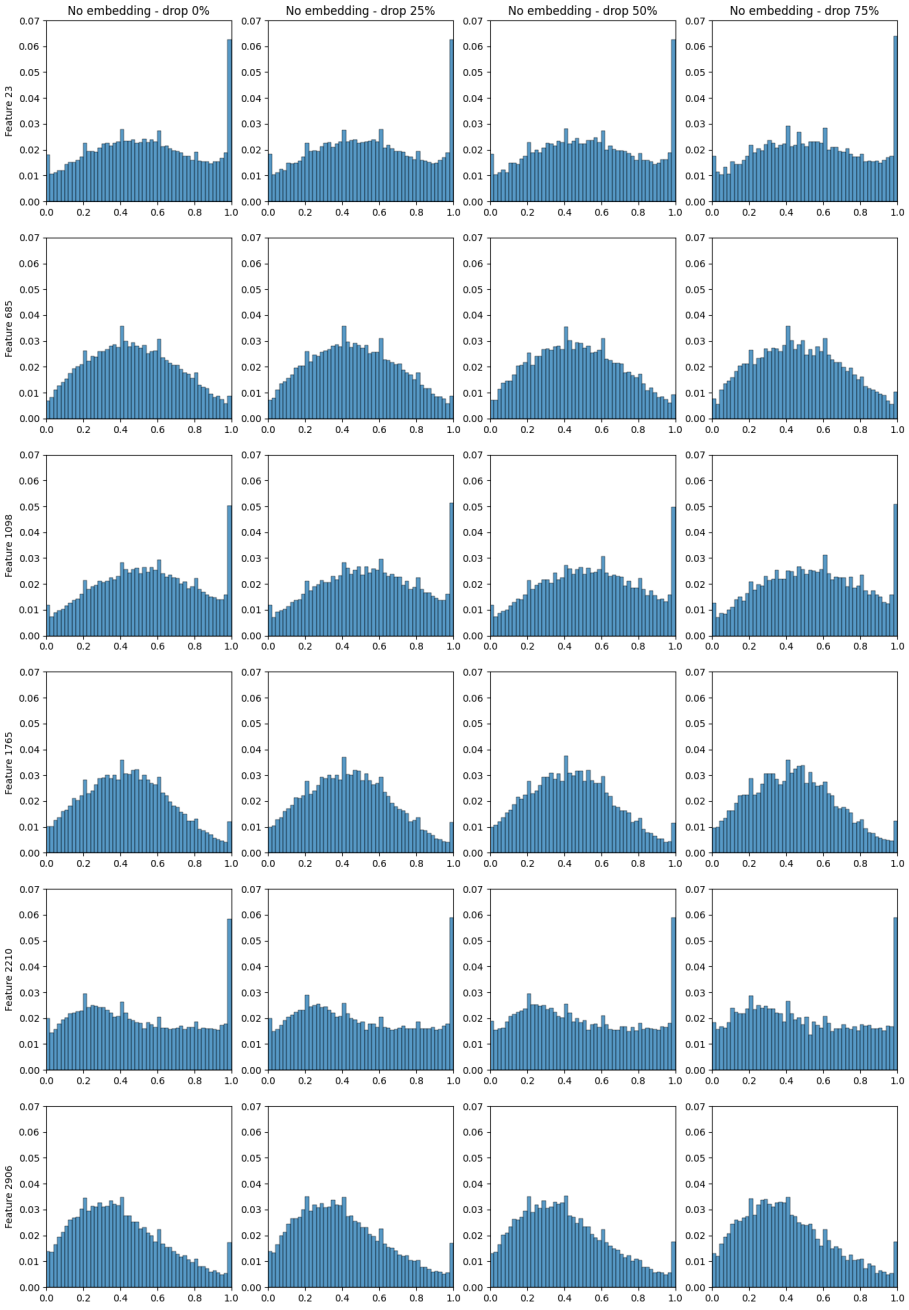

Figure 11: Histograms for six pixels in the CIFAR-10 dataset by increasing number of records dropped from the dataset

This figures shows how dropping records does not significantly change the distribution of each pixel. In line

with our argument set out in section 3, any dataset similarity metric based on distributions alone will not capture this difference between datasets.

Not applying any embeddings to reduce the number of features in the dataset meant that each record in the dataset contained 3,072 features. Not all the similarity metrics could manage so many features in a reasonable computational timeframe, and only MMD and OTDD were applied to the raw datasets without embeddings.

Figure 22 in appendix D shows that both MMD and OTDD were able to identify the experiment pairs where dataset B had a rotate transform applied as being the most dissimilar. However, OTDD did not evaluate the experiment pairs where no transforms were applied to dataset B as being the most similar. MMD evaluated experiment pairs where no transforms or blurred transforms were applied to dataset B as all being similar (with a value very close to zero).

C Hypotheses 1 and 2

C.1 Model Vulnerabilities

Figure 12 presents the distribution of model vulnerabilities to fast gradient attack used in hypothesis 1. It does this by presenting two sets of boxplots - one for each attack success metric. A boxplot for each target model and attack image distribution.

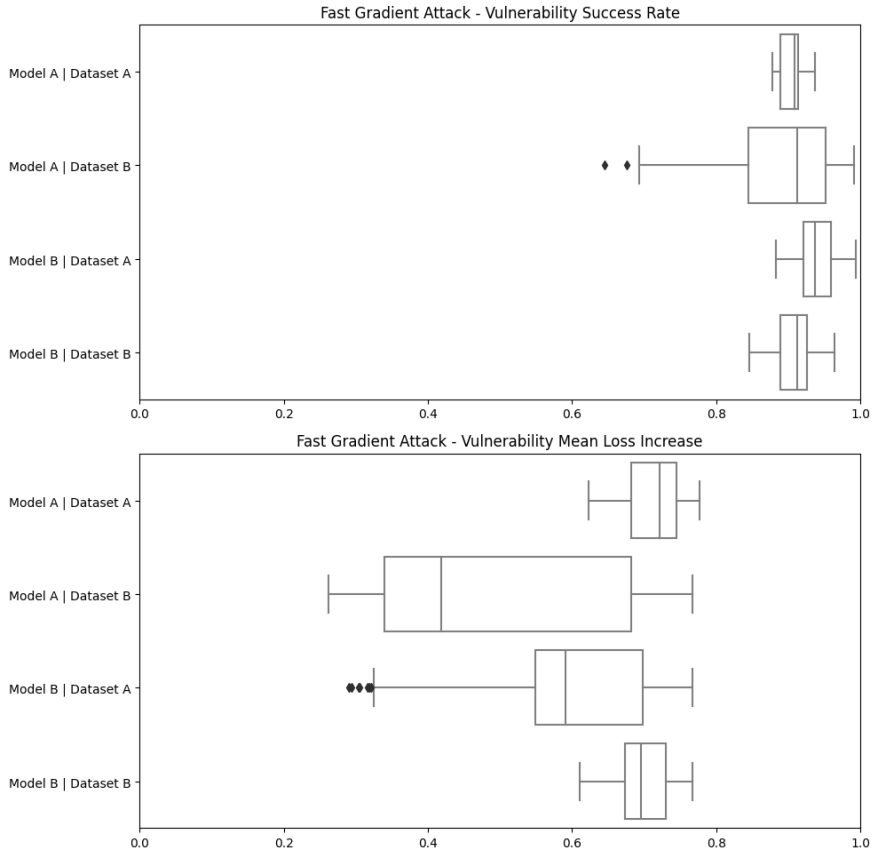

Figure 12: Model vulnerability for fast gradient attacks split by models A and B, and the dataset used for the attack

Across the board figure 12 shows larger success rates than mean loss increases in the vulnerabilities of the models, regardless of direction of transfer or which images the models were based on. Interestingly, distribution of mean loss increase has a much larger variance when using images from the target distribution rather than from the surrogate distribution.

Figure 13 in turn presents the same results in the case of the boundary attacks. The smaller scale on these boxplots (0 to 0.5, instead of 0 to 1) should be noted.

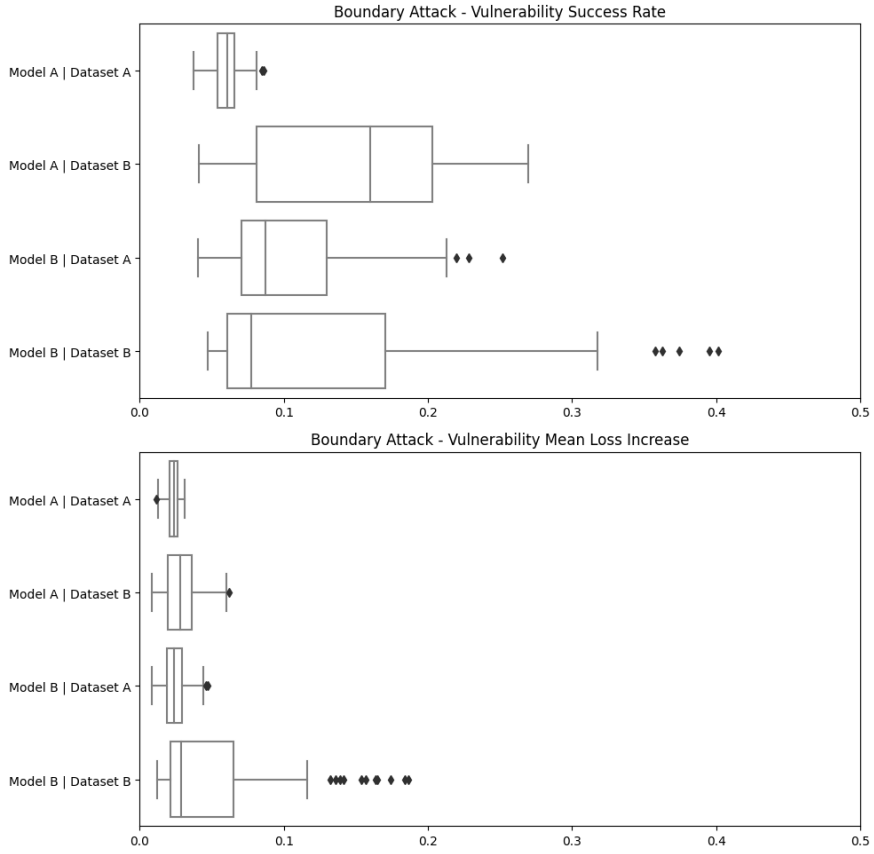

Figure 13: Model vulnerability for boundary attacks split by models A and B, and the dataset used for the attack

Relative to the vulnerabilities for the fast gradient attack, the vulnerabilities in 13 are noticably smaller, even at the extremes of both distributions. We do not know why in our experiments the fast gradient attacks have performed better in this regard.

C.2 Visualising Hypotheses 1 and 2

To give visual representation to the results for hypotheses 1 and 2, figures 14, 15, 16, 17, 18, 19, 20, and 21 present the same results in graph form. In each case, the model vulnerability metric is on the x-axis and the transfer success metric is on the y-axis. As in the main results we match metrics like-for-like: where model vulnerability is measured with the success rate metric, so is transfer success.

Figures 14, 15, 16, and 17 present these results for hypothesis 1. The first two present results in the case of success rate as the attack metric of interest, with fast gradient attack first and boundary attack second. The latter two present these results in the case of mean loss increase, with attacks again in the same order.

Similarly, figures 18, 19, 20, and 21 present these results for hypothesis 2, in the same order.

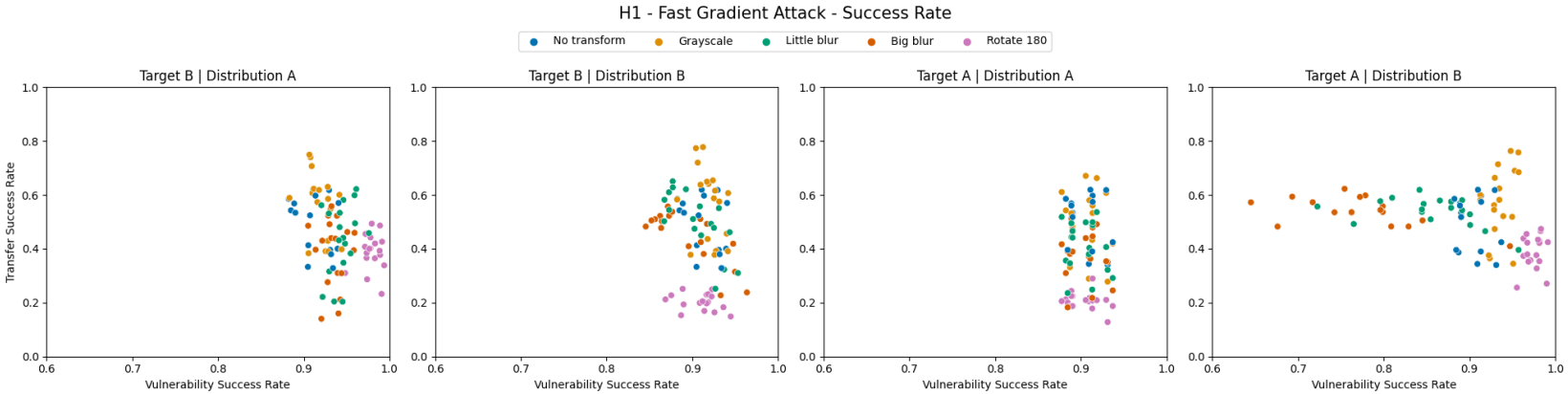

Figure 14: H1 - Vulnerability of the target model to fast gradient attacks using the success rate metric - visualises the correlations in the first row of Table 4

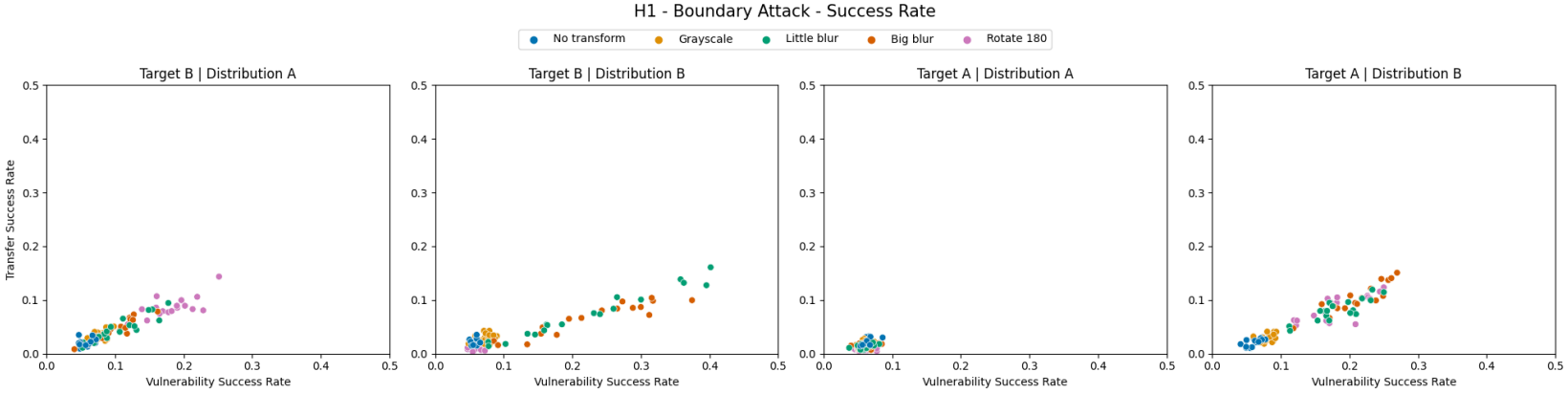

Figure 15: H1 - Vulnerability of the target model to boundary attacks using the success rate metric - visualises the correlations in the second row of Table 4

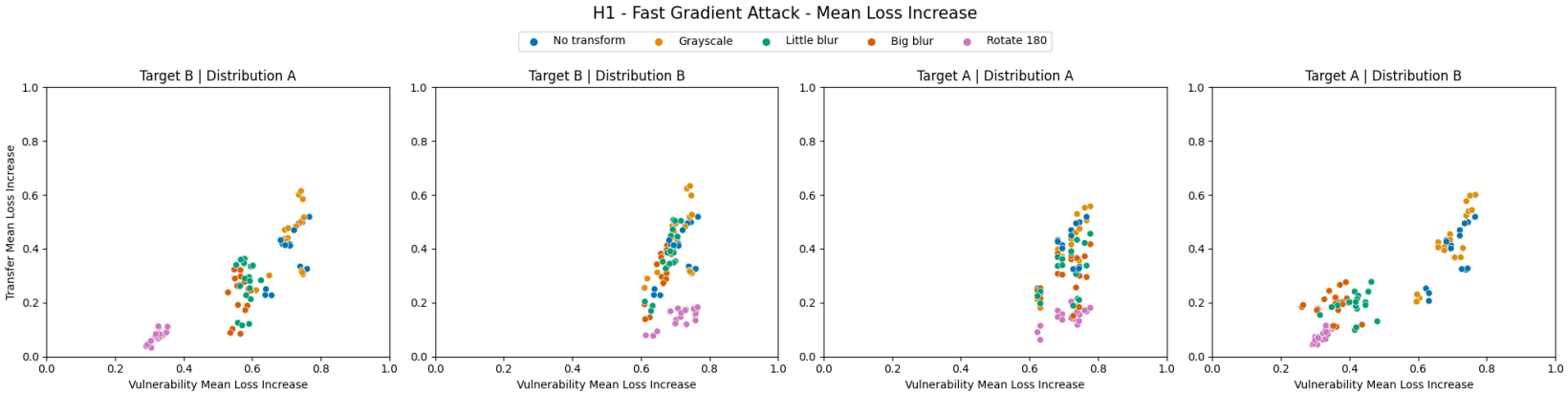

Figure 16: H1 - Vulnerability of the target model to fast gradient attacks using the mean loss increase metric - visualises the correlations in the first row of Table 5

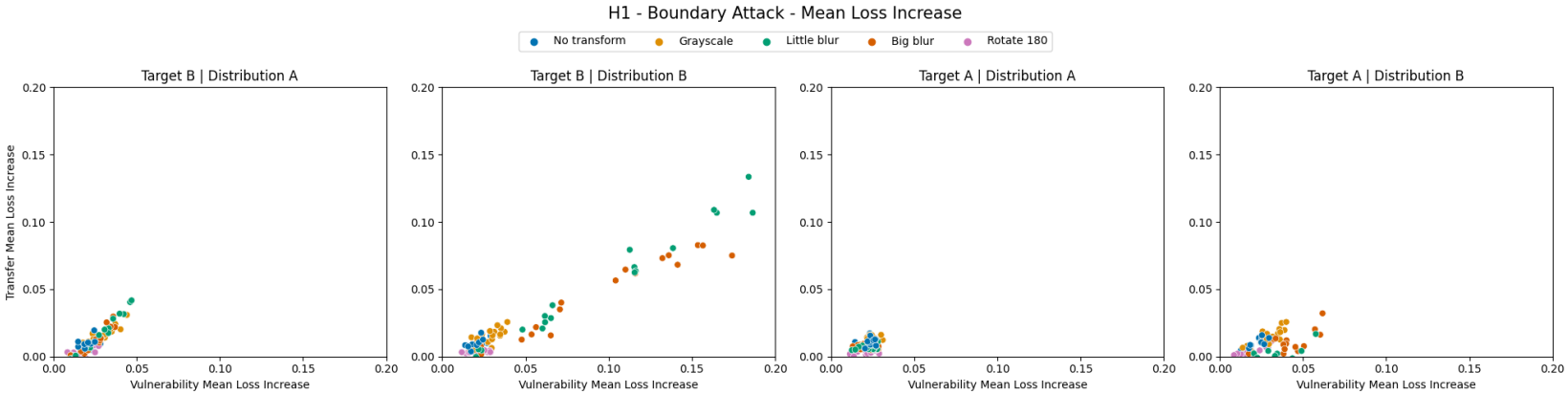

Figure 17: H1 - Vulnerability of the target model to boundary attacks using the mean loss increase metric - visualises the correlations in the second row of Table 5

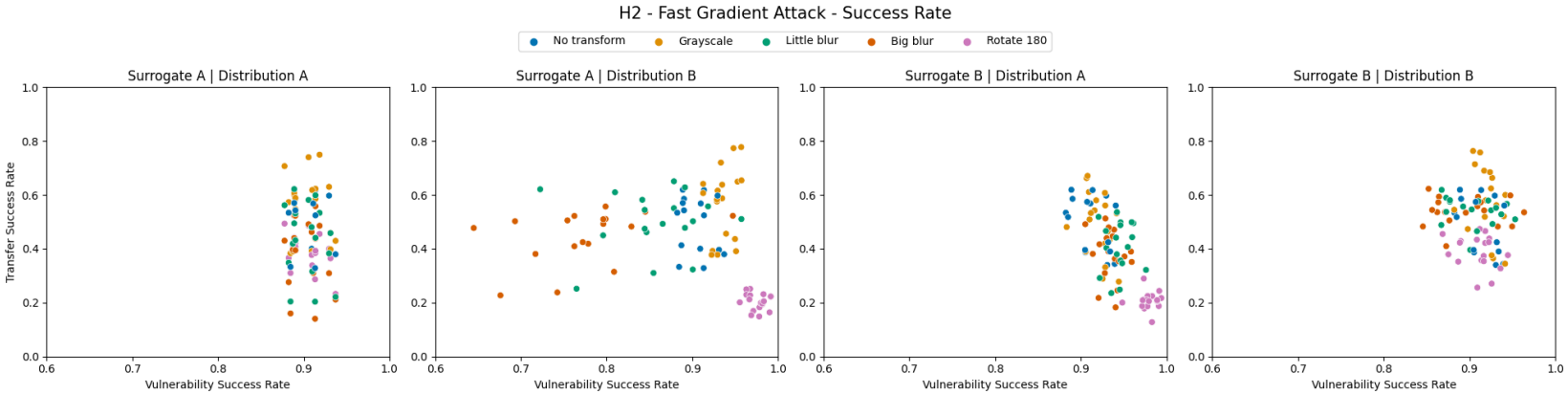

Figure 18: H2 - Vulnerability of the surrogate model to fast gradient attacks using the success rate metric - visualises the correlations in the first row of Table 6

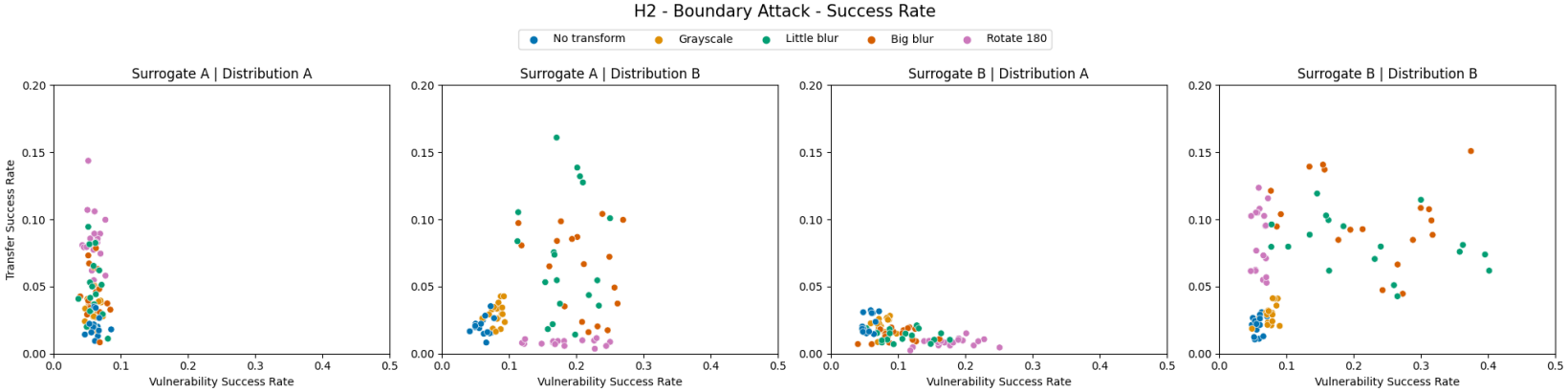

Figure 19: H2 - Vulnerability of the surrogate model to boundary attacks using the success rate metric - visualises the correlations in the second row of Table 6

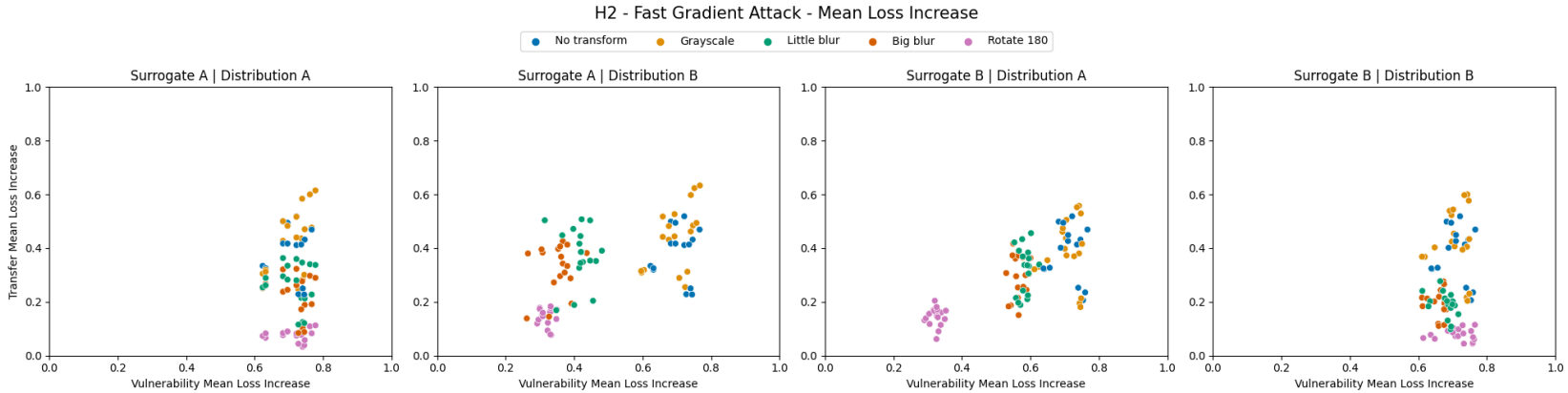

Figure 20: H2 - Vulnerability of the surrogate model to fast gradient attacks using the mean loss increase metric - visualises the correlations in the first row of Table 7

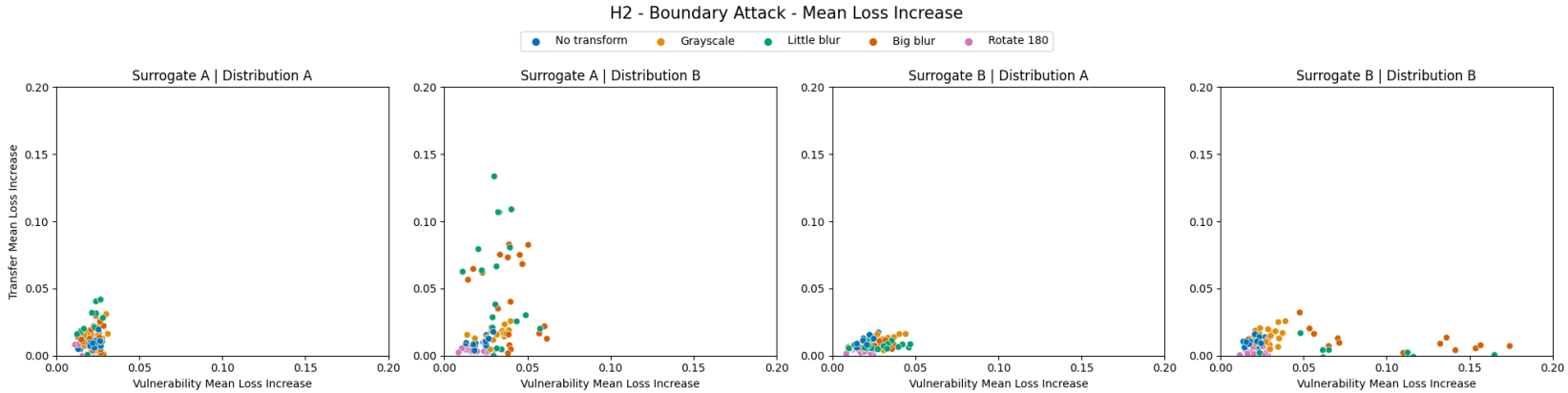

Figure 21: H2 - Vulnerability of the surrogate model to boundary attacks using the mean loss increase metric - visualises the correlations in the second row of Table 7

D Hypothesis 3

The figures presented in this section visualise the relationships discussed in the results for hypothesis 3. Each figure presents a grid of scatter plots. On each scatter plot, the transfer success metric is on the x-axis and the dataset similarity metric is on the y-axis. Points are coloured by which transform group they belong to. Each grid is organised in rows according to dataset similarity metric, and in columns according to choice of embedding (note that this means there are some empty 'slots' in the grid). Each figure notes in its title which attack direction and distribution of images it corresponds to.

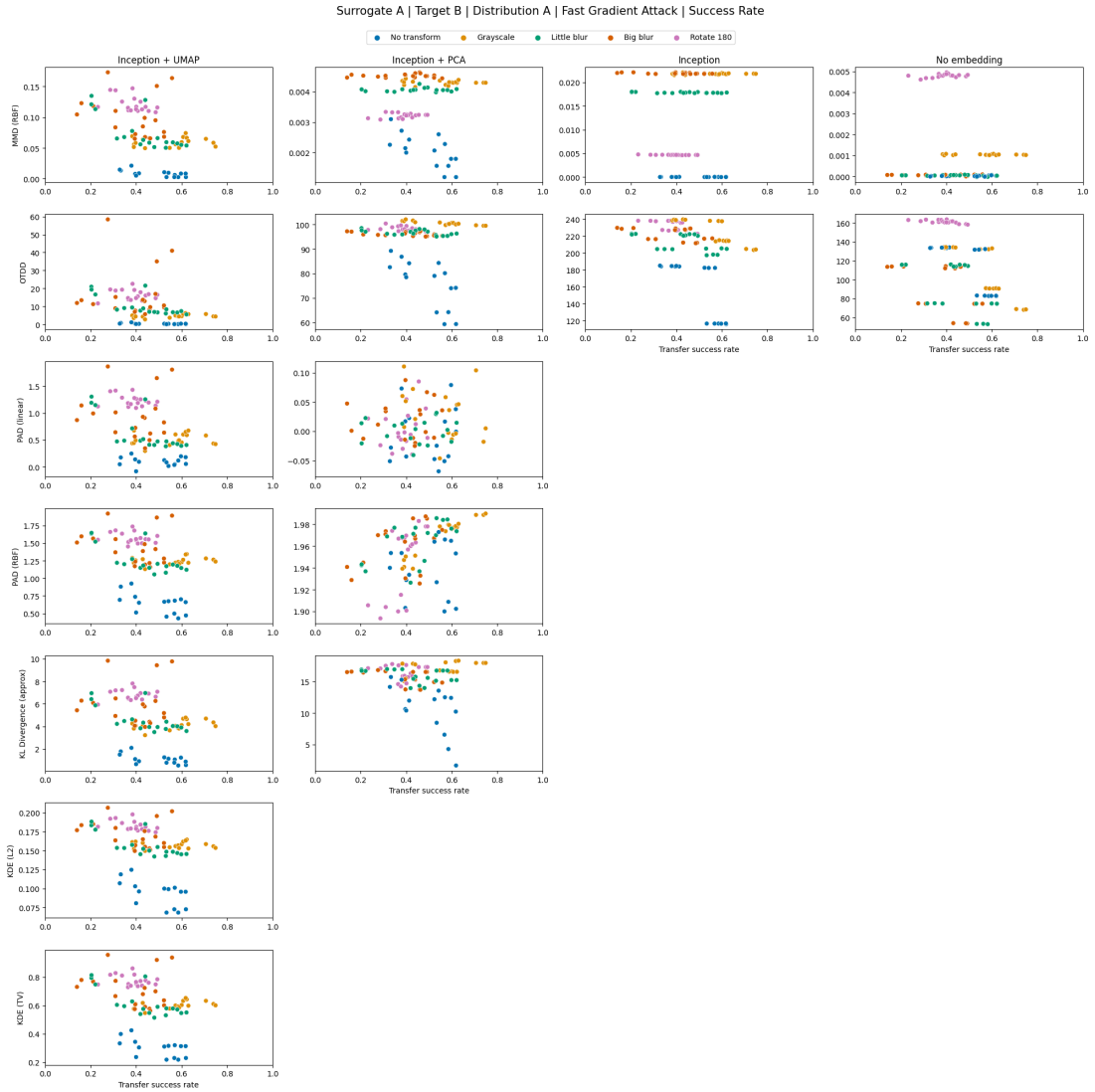

Figure 22: Dataset similarity metrics plotted against transfer attack success rate for fast gradient attacks using surrogate model A, target model B using the data from distribution A to generate the adversarial images.

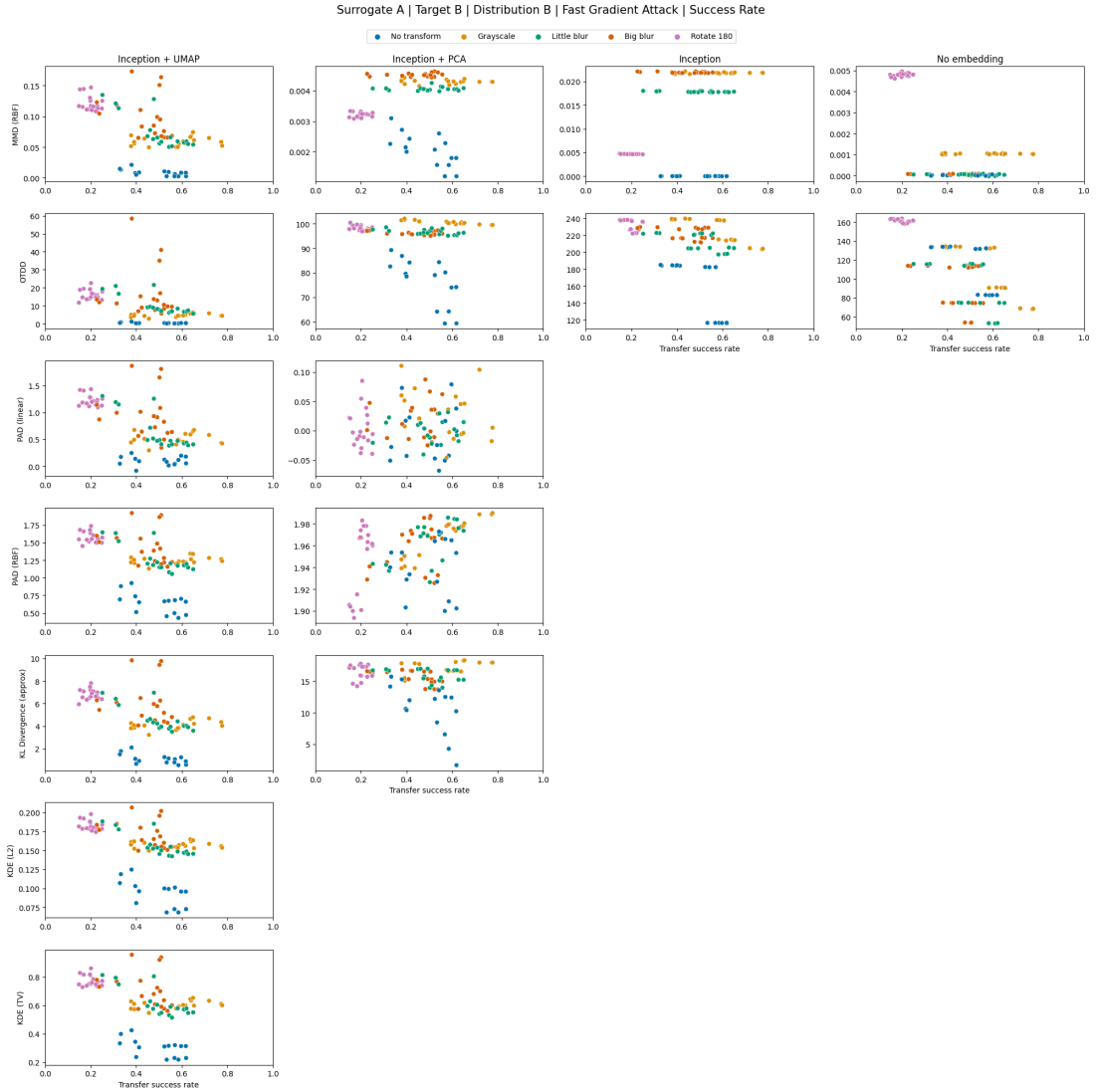

Figure 23: Dataset similarity metrics plotted against transfer attack success rate for fast gradient attacks using surrogate model A, target model B using the data from distribution B to generate the adversarial images.

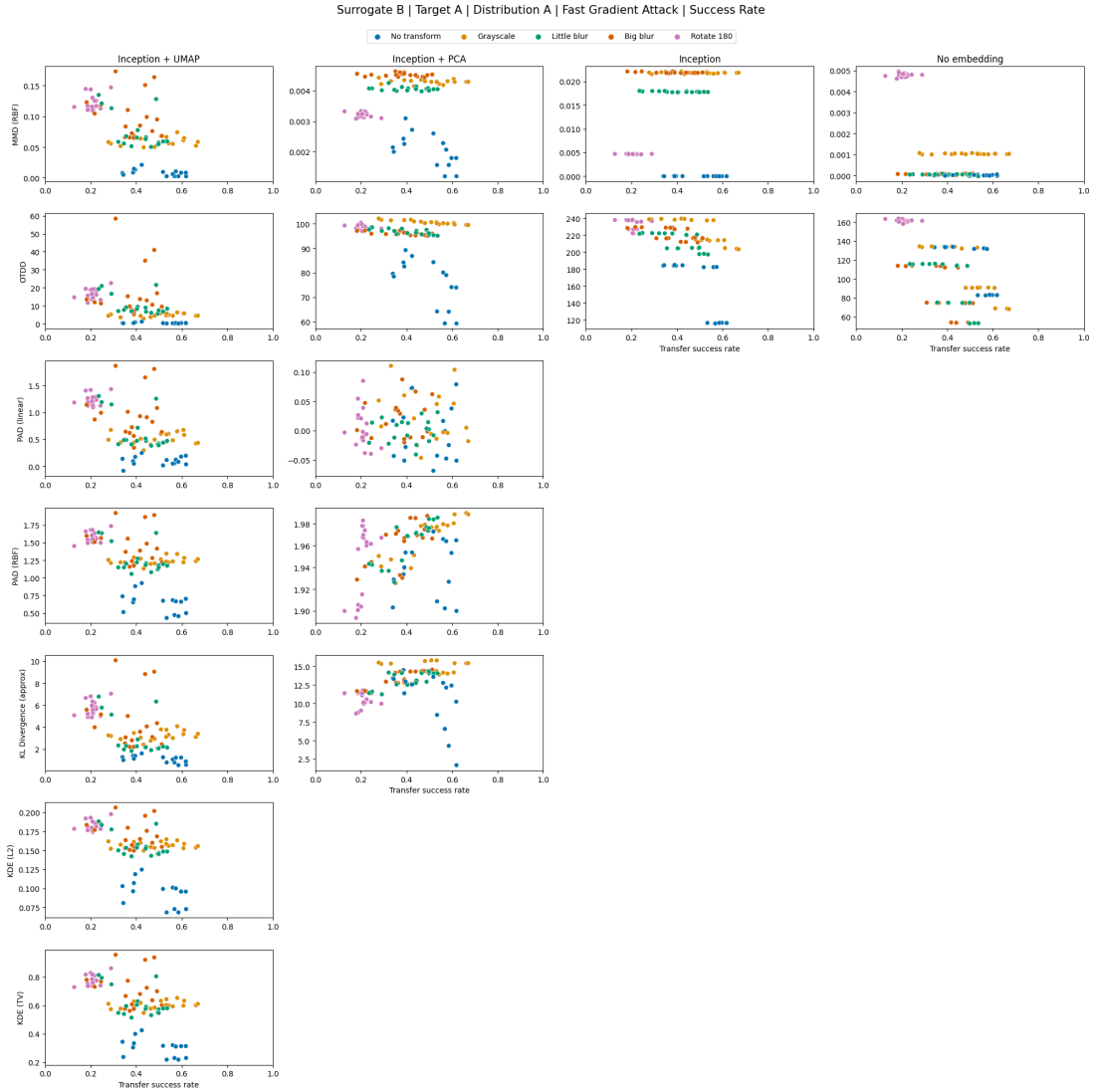

Figure 24: Dataset similarity metrics plotted against transfer attack success rate for fast gradient attacks using surrogate model B, target model A using the data from distribution A to generate the adversarial images.

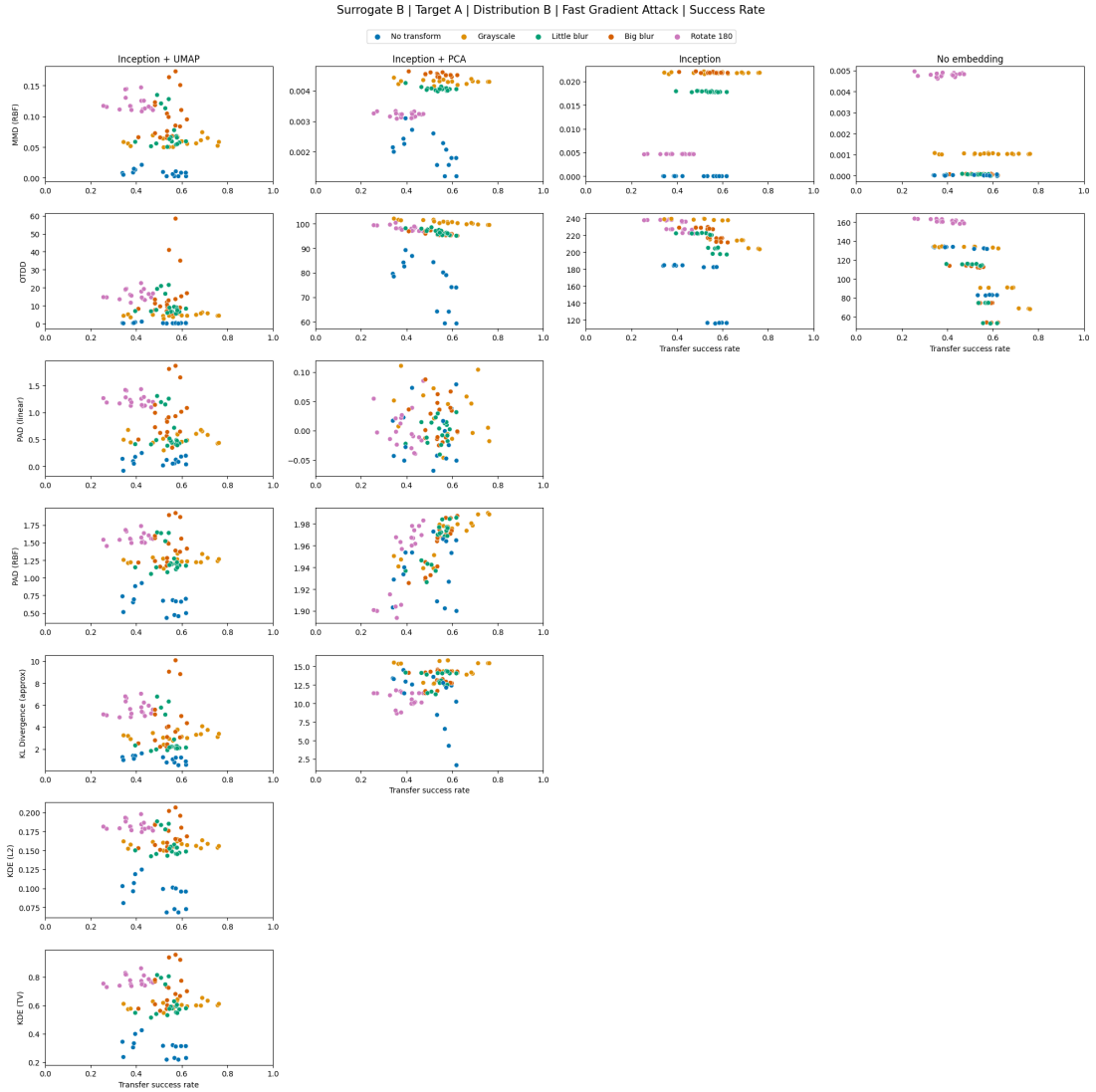

Figure 25: Dataset similarity metrics plotted against transfer attack success rate for fast gradient attacks using surrogate model B, target model A using the data from distribution B to generate the adversarial images.

Figure 26: Dataset similarity metrics plotted against transfer attack mean loss increase for fast gradient attacks using surrogate model A, target model B using the data from distribution A to generate the adversarial images.

Figure 27: Dataset similarity metrics plotted against transfer attack mean loss increase for fast gradient attacks using surrogate model A, target model B using the data from distribution B to generate the adversarial images.

Figure 28: Dataset similarity metrics plotted against transfer attack mean loss increase for fast gradient attacks using surrogate model B, target model A using the data from distribution A to generate the adversarial images.

Figure 29: Dataset similarity metrics plotted against transfer attack mean loss increase for fast gradient attacks using surrogate model B, target model A using the data from distribution B to generate the adversarial images.

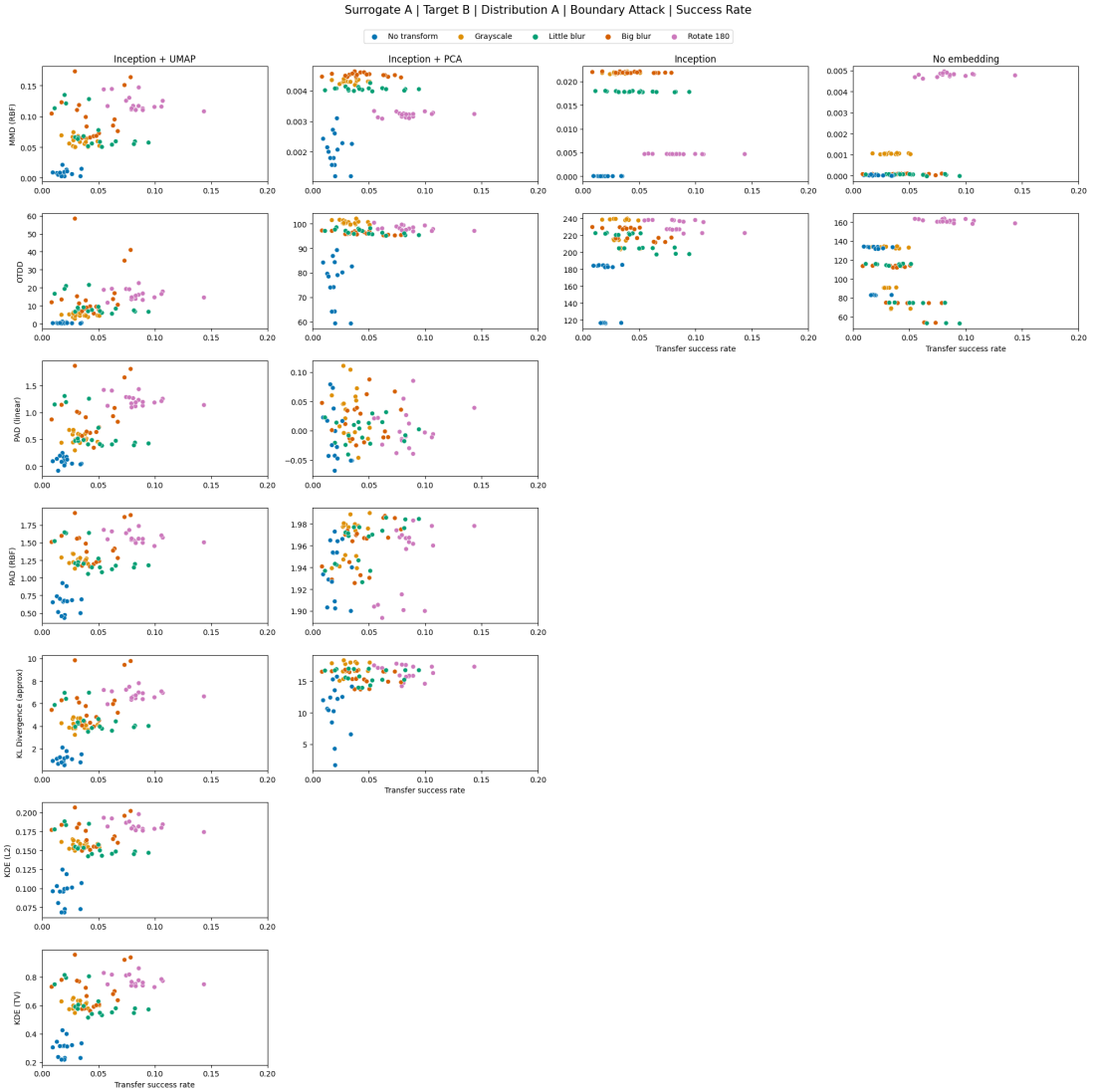

Figure 30: Dataset similarity metrics plotted against transfer attack success rate for boundary attacks using surrogate model A, target model B using the data from distribution A to generate the adversarial images.

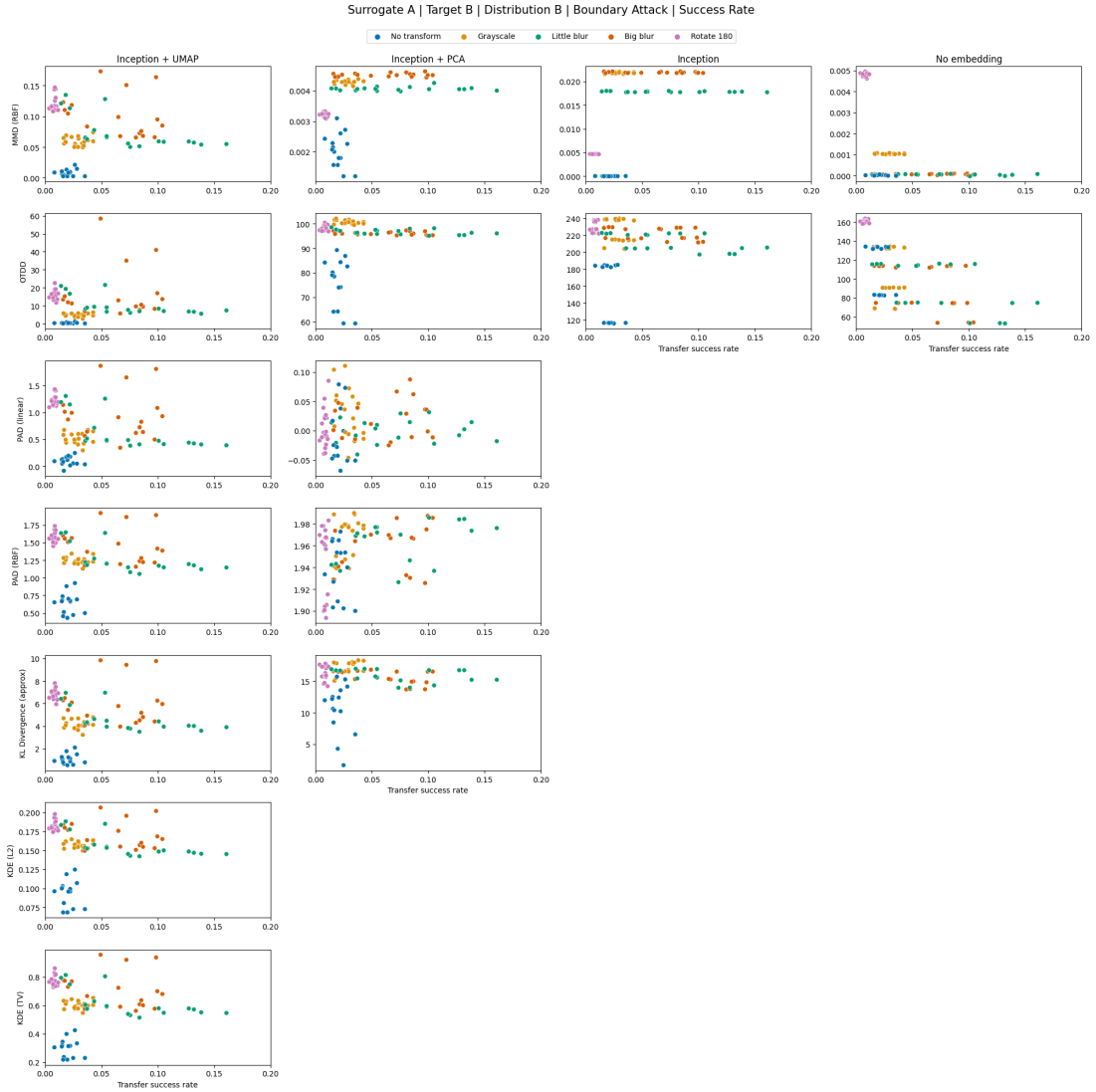

Figure 31: Dataset similarity metrics plotted against transfer attack success rate for boundary attacks using surrogate model A, target model B using the data from distribution B to generate the adversarial images.

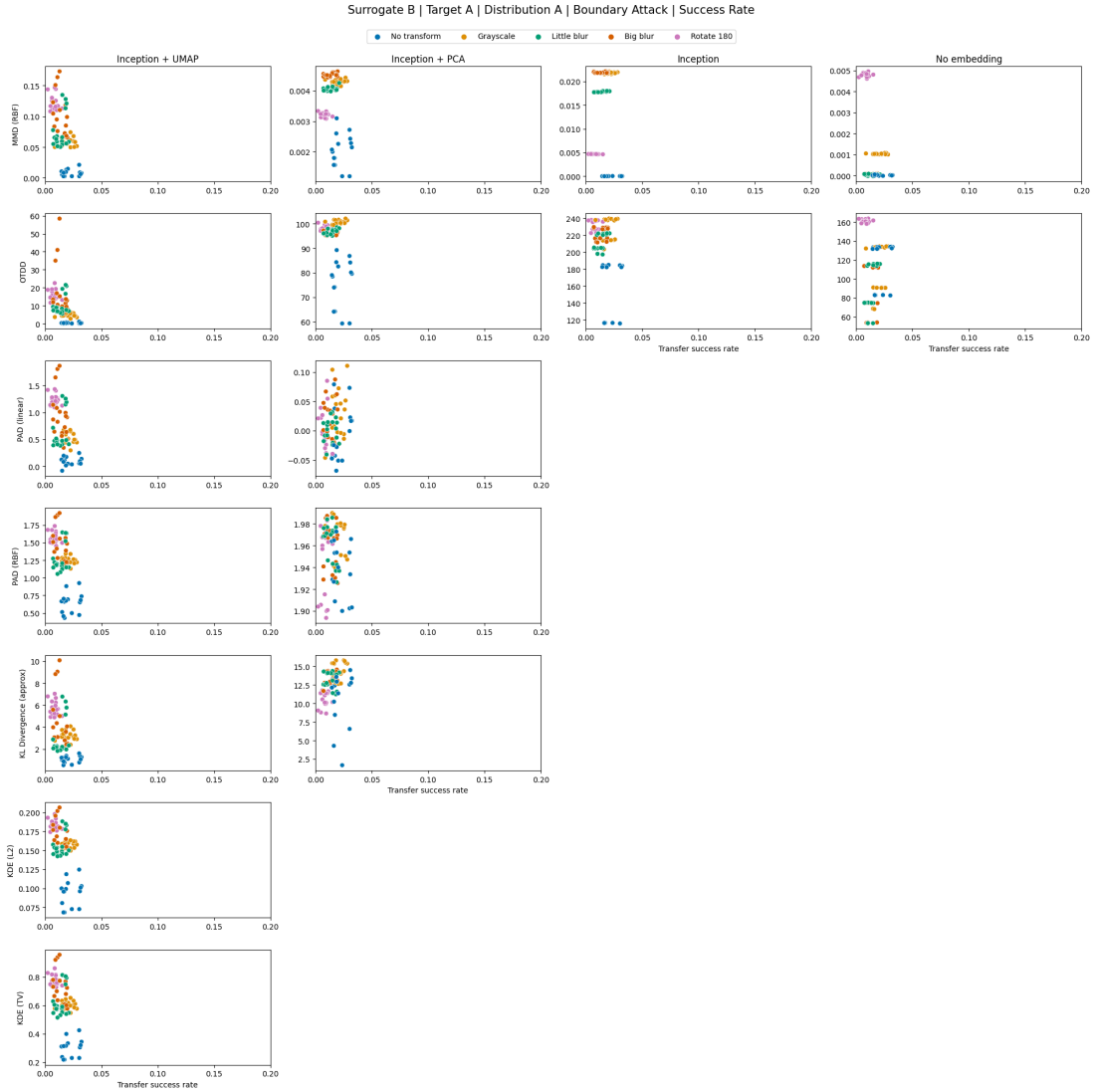

Figure 32: Dataset similarity metrics plotted against transfer attack success rate for boundary attacks using surrogate model B, target model A using the data from distribution A to generate the adversarial images.

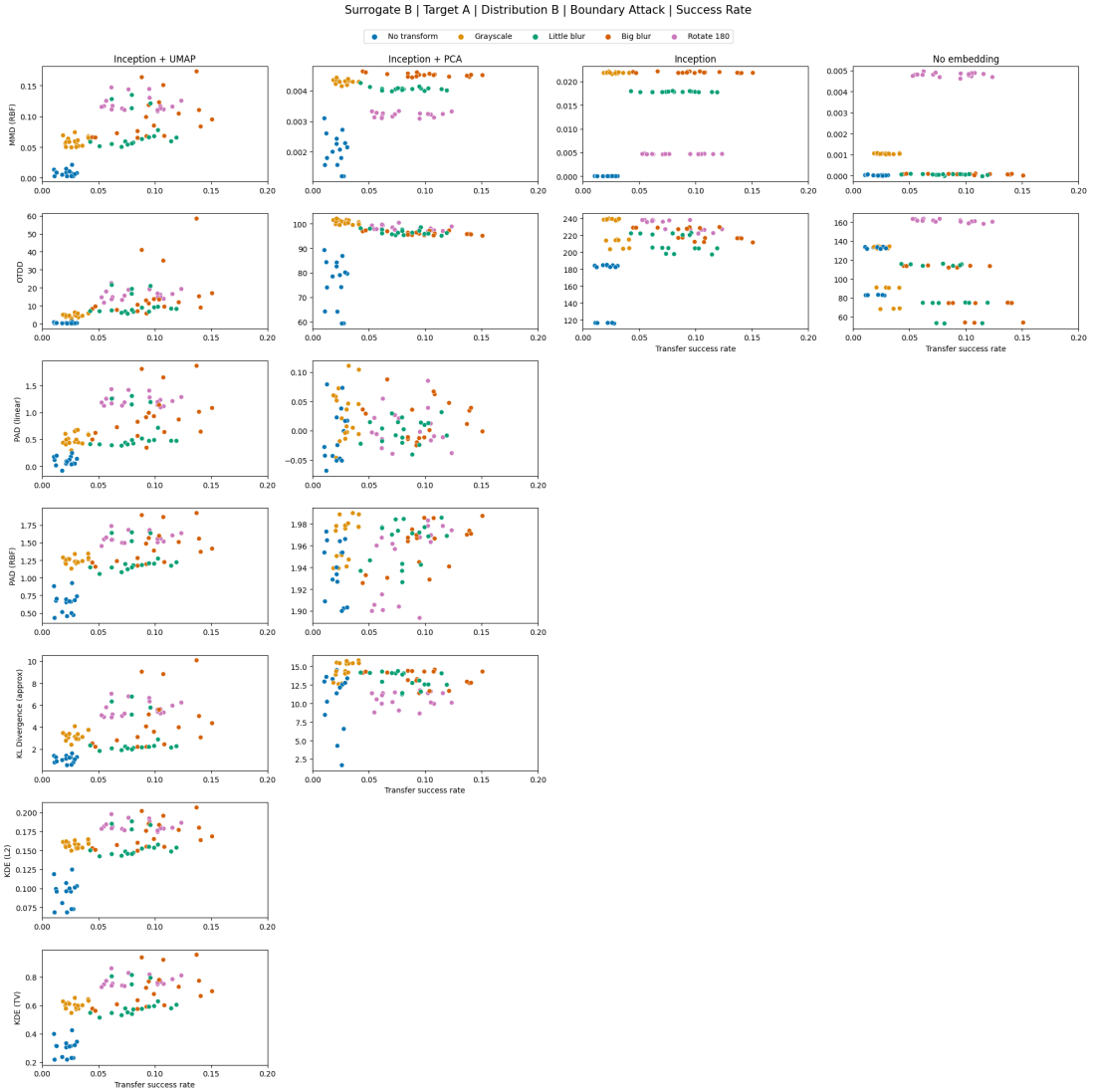

Figure 33: Dataset similarity metrics plotted against transfer attack success rate for boundary attacks using surrogate model B, target model A using the data from distribution B to generate the adversarial images.

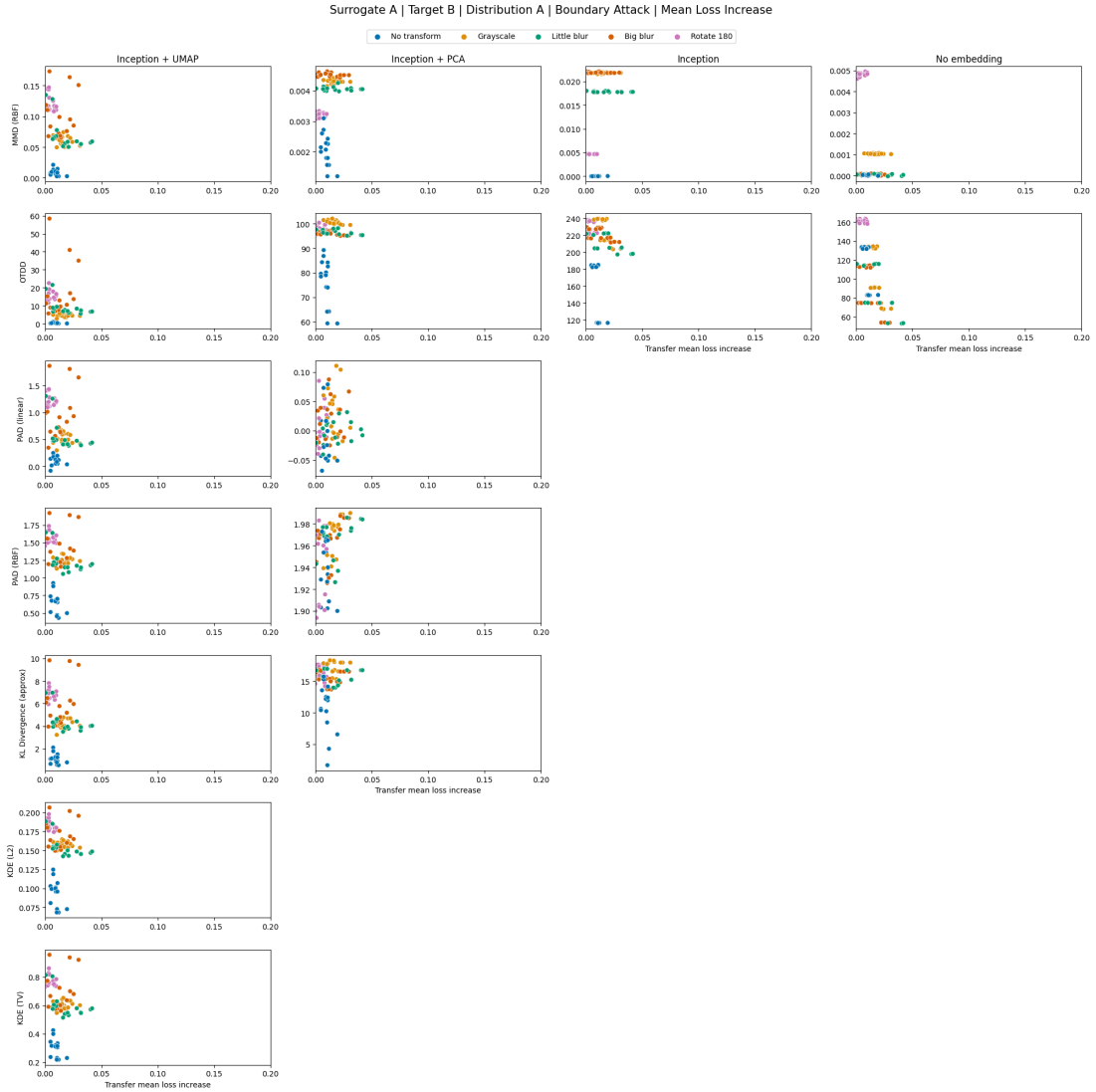

Figure 34: Dataset similarity metrics plotted against transfer attack mean loss increase for boundary attacks using surrogate model A, target model B using the data from distribution A to generate the adversarial images.

Figure 35: Dataset similarity metrics plotted against transfer attack mean loss increase for boundary attacks using surrogate model A, target model B using the data from distribution B to generate the adversarial images.

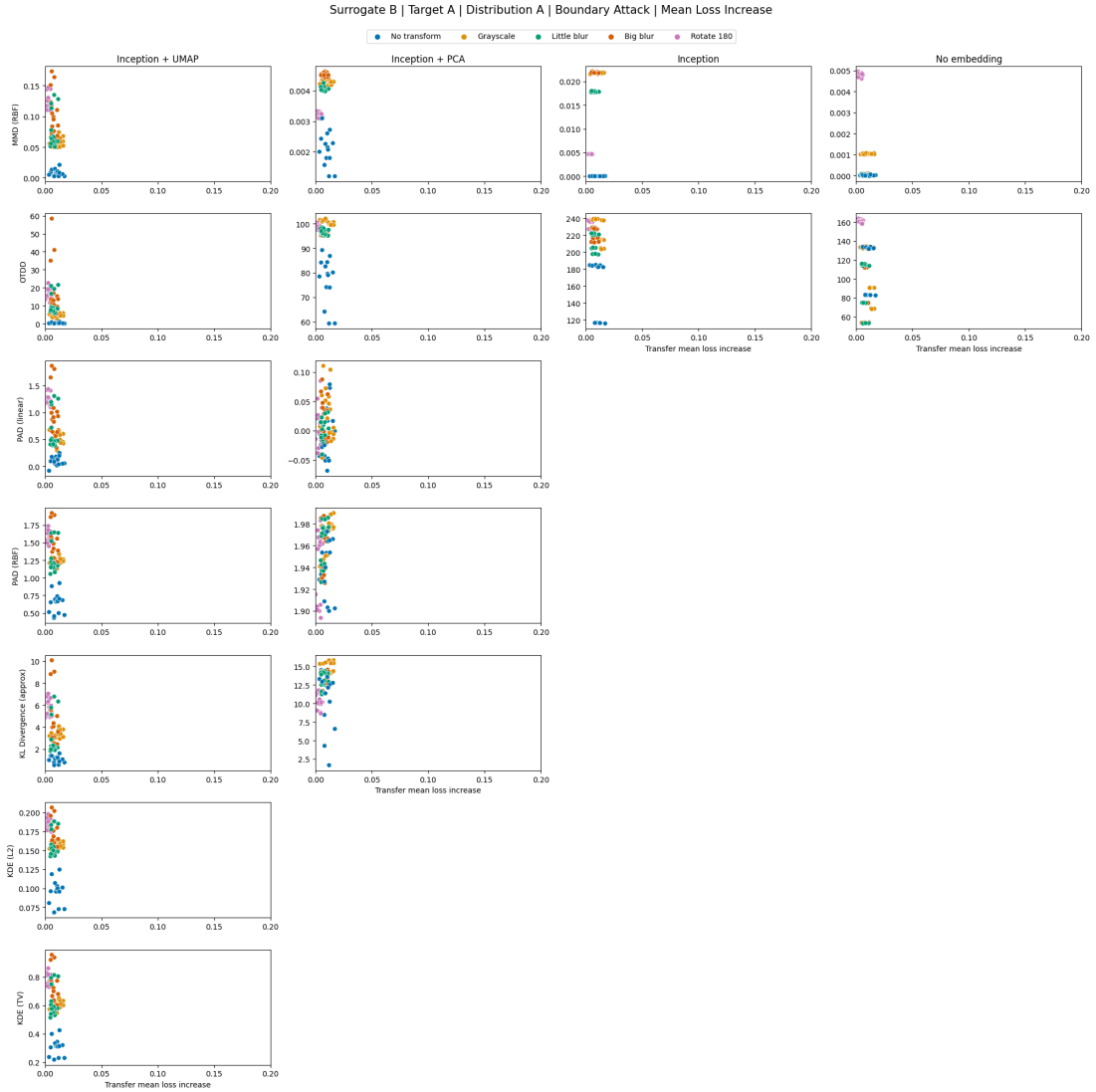

Figure 36: Dataset similarity metrics plotted against transfer attack mean loss increase for boundary attacks using surrogate model B, target model A using the data from distribution A to generate the adversarial images.

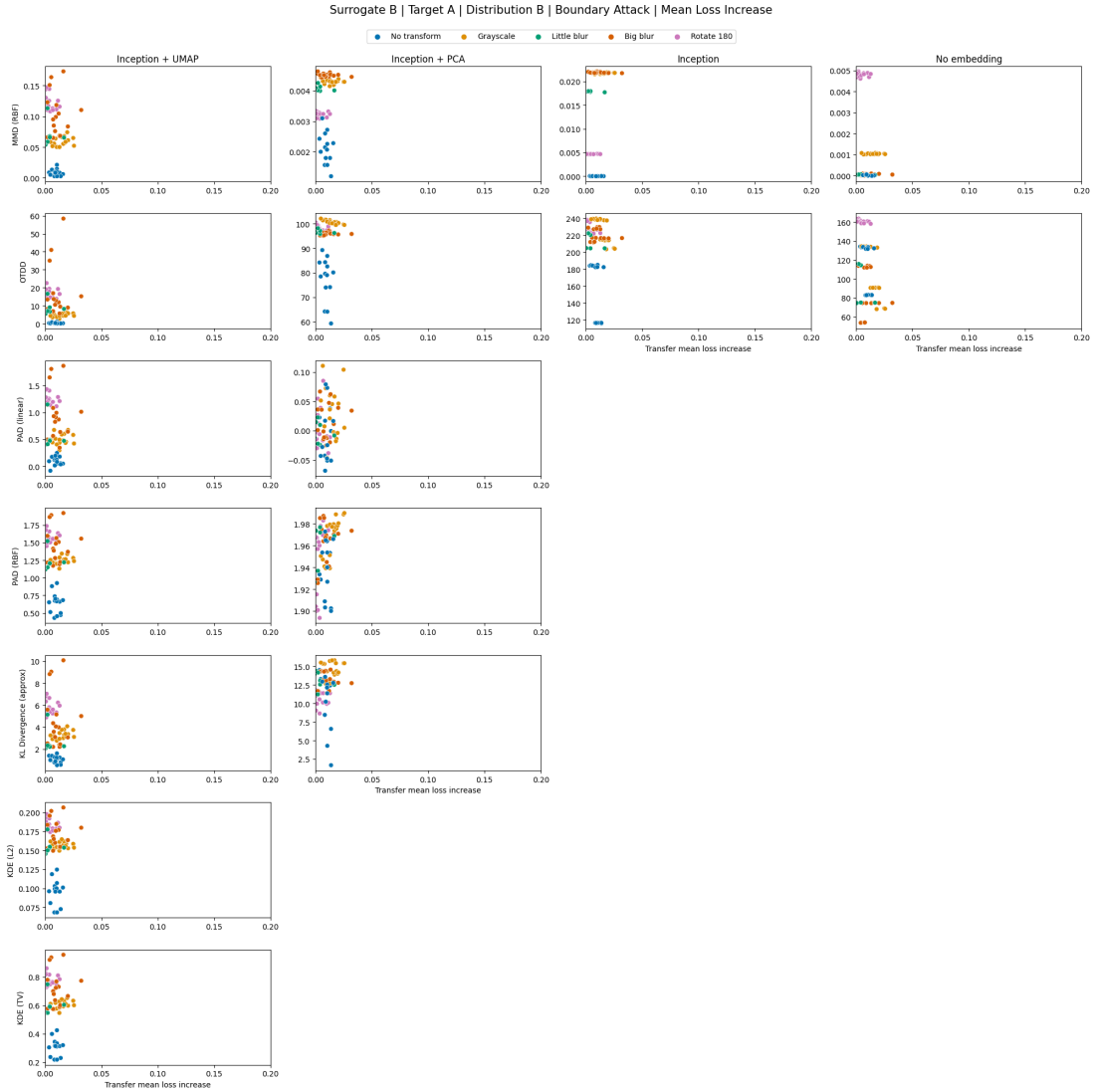

Figure 37: Dataset similarity metrics plotted against transfer attack mean loss increase for boundary attacks using surrogate model B, target model A using the data from distribution B to generate the adversarial images.

E Hypothesis 4

E.1 Additional Tables

Tables 14, 15, 16, and 17 re-present the main results for hypothesis 4, but instead of using the absolute difference in size they instead use different operationalisations of the role of size. Tables 14 and 16 take the ratio of target dataset size to surrogate dataset size in the cases of fast gradient attack and boundary attack respectively. Tables 15 and 17 show the difference between them (target subtracted from surrogate), again in the same order.

For fast gradient attack, the results are mostly null. However, where the surrogte distribution is used for attack images, there is a significant negative correlation in 3 out of 4 cases for the difference in table 15. This implies that as the surrogate dataset becomes larger than the target dataset, transfer success becomes less likely.

For the boundary attack, there is a significant positive correlation in three of the correlations between the ratio of sizes (target size divided by surrogate size). However, there are five negative and significant correlations (and one positive) between the difference in sizes between surrogate and target. This mixed set of results leads to conclusions in two opposite directions: the first suggests that as the surrogate becomes larger, transfer success becomes more likely. The second suggests that as the surrogate becomes larger, transfer success becomes less likely.

Table 14: Dataset Size, Ratio - Fast Gradient Attack

Attack Success Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Success Rate	0.11	0.03	-0.19	0.16
Mean Loss Increase	0.07	0.09	-0.09	0.11

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Table 15: Dataset Size, Difference - Fast Gradient Attack

Attack Success Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Success Rate	**-0.29**	-0.13	0.03	**-0.35**
Mean Loss Increase	-0.17	-0.2	-0.09	**-0.22**

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Table 16: Dataset Size, Ratio - Boundary Attack

Attack Success Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Success Rate	0.07	0.21	-0.19	**0.23**
Mean Loss Increase	**0.21**	**0.26**	-0.06	0.14

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

Table 17: Dataset Size, Difference - Boundary Attack

Attack Success Metric	Surrogate A, Target B		Surrogate B, Target A	
	A Distribution	B Distribution	A Distribution	B Distribution
Success Rate	-0.17	**-0.28**	**0.21**	**-0.26**
Mean Loss Increase	**-0.35**	**-0.34**	-0.04	**-0.24**

All values rounded to 2 decimal places. Values significant at the 95% confidence level highlighted in bold.

E.2 Visualising Hypothesis 4

Figures 38, 39, 40, and 41 visualise hypothesis 4 by presenting boxplots of transfer attack successes.

Each boxplot contains six distributions: one for each drop combination listed in section 6. Note that where Drop A is 0 and Drop B is 0.5, this means that drop A is the larger dataset, as Drop B corresponds to dropping 50% of the data from dataset B.

Each figure is organised into a grid of four plots. Each plot in the grid corresponds to a transfer direction (A surrogate and B target, or A target and B surrogate) and a source distribution (A or B) for the attack images. Figures 38 and 39 visualise these results for the fast gradient attack for success rate and mean loss increase respectively. Figures 40 and 41 visualise these results for the Boundary Attack, in the same order for attack success metrics.

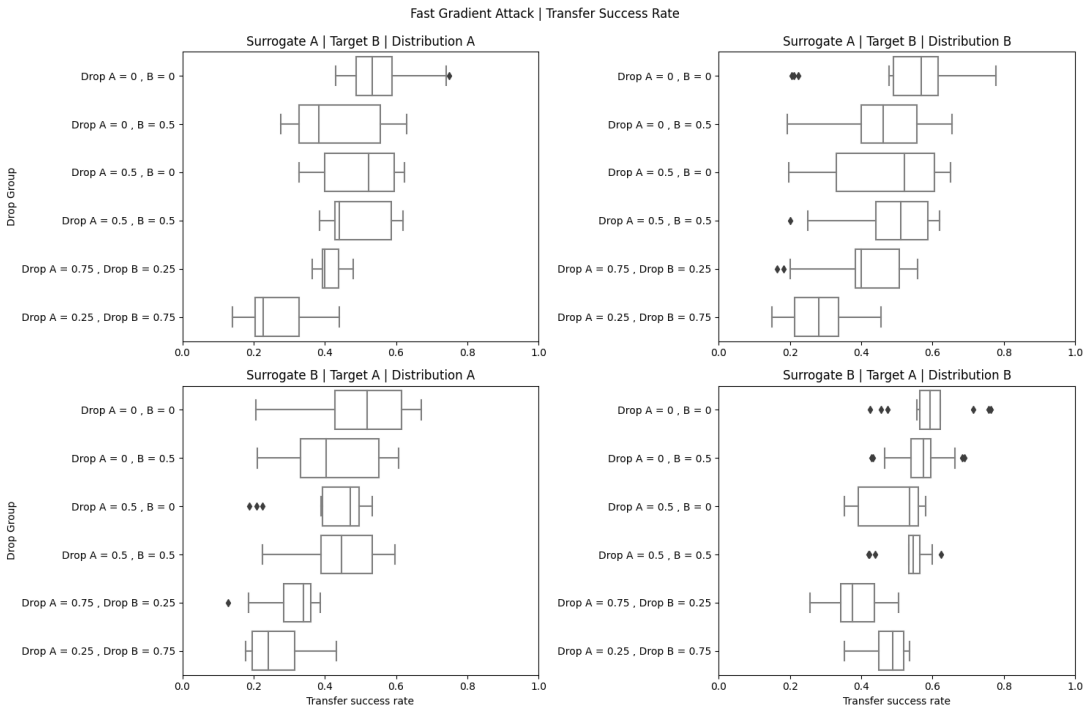

Figure 38: Transfer success rate by drop group for fast gradient attacks

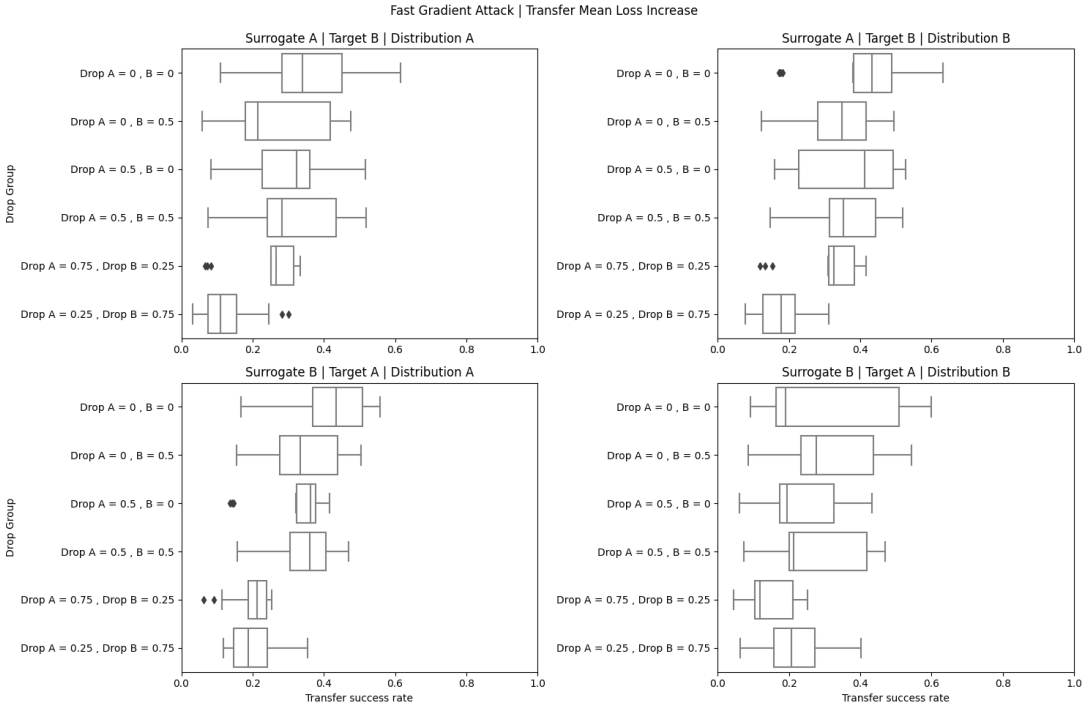

Figure 39: Transfer mean loss increase by drop group for fast gradient attacks

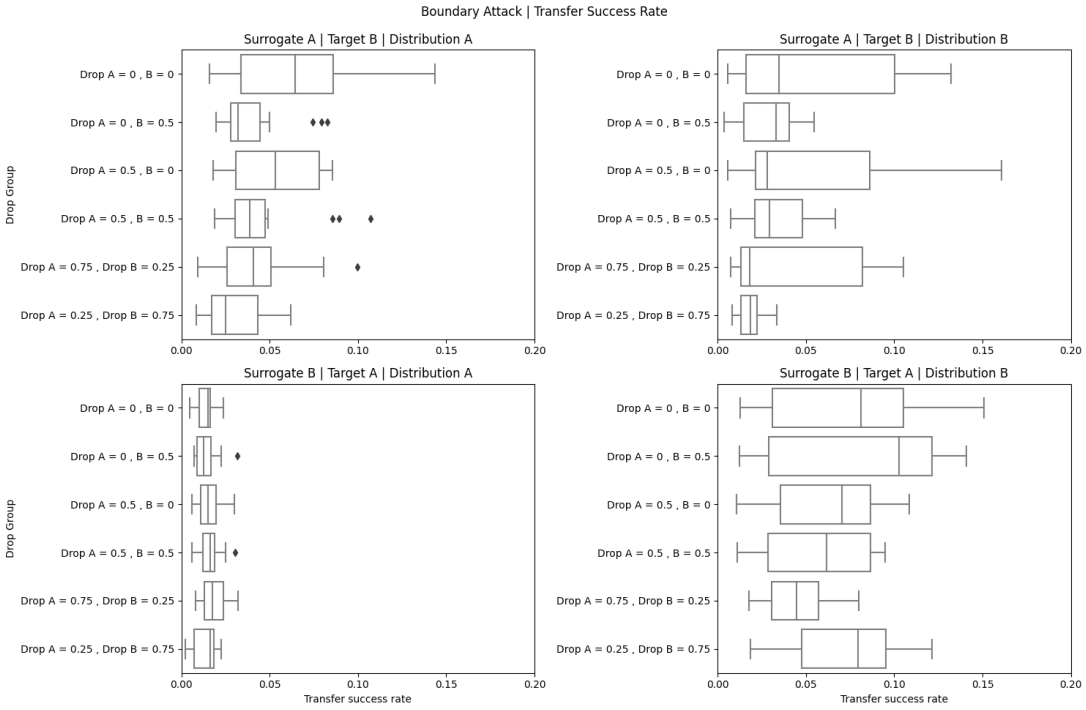

Figure 40: Transfer success rate by drop group for boundary attacks

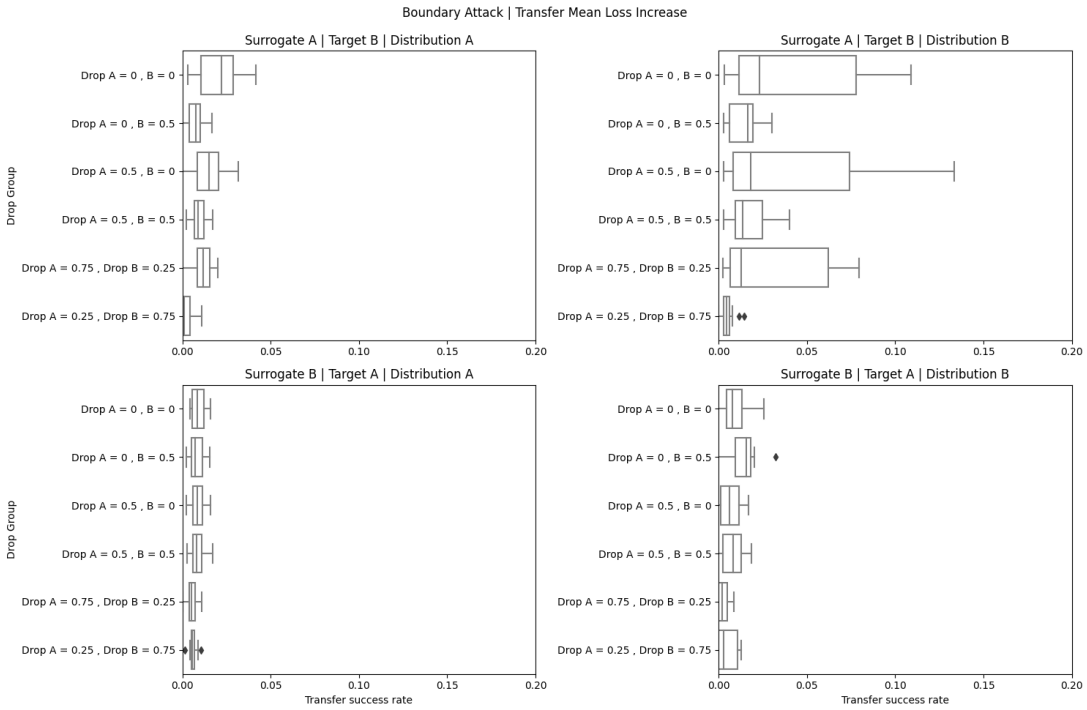

Figure 41: Transfer mean loss increase by drop group for boundary attacks