It Doesn't Look Like Anything to Me: Using Diffusion Model to Subvert Visual Phishing Detectors

Qingying Hao
UIUC
qhao2@illinois.edu

Nirav Diwan
UIUC
ndiwan2@illinois.edu

Ying Yuan
University of Padua
ying.yuan@math.unipd.it

Giovanni Apruzzese
University of Liechtenstein
giovanni.apruzzese@uni.li

Mauro Conti
University of Padua
mauro.conti@unipd.it

Gang Wang
UIUC
gangw@illinois.edu

Abstract

Visual phishing detectors rely on website logos as the invariant *identity indicator* to detect phishing websites that mimic a target brand's website. Despite their promising performance, the robustness of these detectors is not yet well understood. In this paper, we challenge the invariant assumption of these detectors and propose new attack tactics, `LogoMorph`, with the ultimate purpose of enhancing these systems. `LogoMorph` is rooted in a key insight: users can neglect *large* visual perturbations on the logo as long as the perturbation preserves the original logo's semantics. We devise a range of attack methods to create semantic-preserving adversarial logos, yielding phishing webpages that bypass state-of-the-art detectors. For text-based logos, we find that using alternative fonts can help to achieve the attack goal. For image-based logos, we find that an adversarial diffusion model can effectively capture the style of the logo while generating new variants with large visual differences. Practically, we evaluate `LogoMorph` with white-box and black-box experiments and test the resulting adversarial webpages against various visual phishing detectors end-to-end. User studies ($n = 150$) confirm the effectiveness of our adversarial phishing webpages on end users (with a detection rate of 0.59, barely better than a coin toss). We also propose and evaluate countermeasures, and share our code.

1 Introduction

Phishing attacks have been a persistent threat online [19], and are still one of the leading causes of data breaches as of 2022–2023 [26, 53]. In this context, phishing websites often impersonate trusted entities (e.g., well-known brands) to gain the victims' trust. For defense, researchers have investigated various methods such as blocklists [11, 31, 48] and data-driven detectors to catch phishing webpages and URLs [52, 54, 55]

Recently, a series of systems have been proposed to counter phishing websites using *reference-based* methods [6, 21, 30, 32, 33]. The key insight is that phishing websites need to look *visually similar* to the legitimate websites of the target brands to deceive users. Therefore, these approaches treat such visual similarity as the "invariant" for robust phishing detection. They maintain a list of references (such as screenshots or logos) for popular brands and detect phishing webpages that *(i)* have a high visual similarity to the reference representation of a target brand, but *(ii)* are not hosted under the domain name of the target brand (refer to §2.1 for background). Despite the promising performance, the robustness of reference-based visual phishing detectors is not well understood, especially for the recent logo-based detectors [30, 32, 33]. Prior efforts either test such detectors against off-the-shelf adversarial algorithms [30, 32] or only focus on bypassing the logo detector [29] (which is only a single component of the detection system), neglecting the end-to-end impact on the webpages. As a result, the robustness of these detectors remains unclear.

Figure 1: **Adversarial Logo Examples**—We show the original logo and two attack examples generated by our `LogoMorph`.

New Attack Tactics. In this paper, we extensively scrutinize the security of visual phishing detectors by challenging the *invariant assumption* of these detectors. We introduce new attack tactics and discuss possible directions to enhance these detectors. We primarily focus on detectors that rely on *logos* as the website identity since there is evidence that such logo-based detectors have been deployed also in production-grade phishing solutions [7, 17]. Specifically, we present `LogoMorph`, a new attack to generate an adversarial logo that enables a webpage to bypass existing visual phishing detectors. The attack is based on a key hypothesis: *users often overlook large visual perturbations as long as the perturbation preserves the original logo "semantics."* (We test this hypothesis with a user study in §5.) Hence, our idea is to search for semantic-preserving perturbations. Figure 1 shows

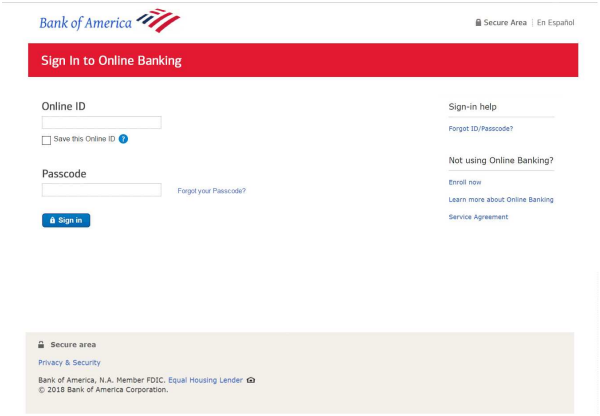

Figure 2: **Adversarial Phishing Webpage**—By using an adversarial logo crafted with LogoMorph, this phishing webpage bypasses detectors such as PhishIntention [32] and Phishpedia [30].

examples of such adversarial logos. The original logo of Bank of America (BOA) looks like an American flag in a diamond shape. The adversarial logo preserves the red color and overall diamond shape but changes the spacing and direction of the stripe patterns. Similarly, for CHASE, the adversarial logos have large visual perturbation but the original style is preserved. Figure 2 shows an example where the adversarial logo is placed on a phishing webpage.

To implement LogoMorph (§3), we categorize logos into three categories: text-only logos (e.g., Google), image-only logos (e.g., AT&T), and image-text logos (e.g., CHASE). We design different attack strategies against the image part and text part, respectively. For the text part, our strategy is to search for an alternative *font* for the logo text from a large pool of visually similar fonts while preserving the spelling, color scheme, and spacing of the original logo's text. For the image part, we propose an *Adversarial Diffusion Model* that simultaneously learns to preserve the semantics/styles of the original logo while introducing a large distance from the reference logos in the feature space of the phishing detectors.

System and User Assessment. We test LogoMorph against phishing detection systems and human users. First (§4), we use LogoMorph to generate adversarial logos, add them to webpages, and evaluate them against state-of-the-art phishing detectors; we use PhishIntention [32] as the primary target for such a *white-box* analysis. Then, we perform transferability experiments in a *black-box* setting: the attacker uses a local surrogate to generate the adversarial logos and then applies them to phishing webpages to attack a different detector (e.g., logo-based Phishpedia [30] and PhishingIntention [32], and screenshot-based VisualPhishNet [6]). We compare (§4.7) LogoMorph against a prior work [29], and explore possible defenses (§6) using gradient masking and adversarial retraining—including an "augmented" version that we specifically devise to counter LogoMorph. Finally, we (§5) conduct a user study ($n = 150$) to verify whether our LogoMorph yields adversarial logos that deceive also the human eye.

Results. We assess logos of 110 brands: we consider 18 high-ranked brands in the main paper, whereas the remaining 92 brands are covered in the Appendix. [*System:*] Our result shows that the text-based attack is highly effective. Most brands (11 out of 15 brands with text logo components) can identify over 100 attack fonts to bypass the detection at the webpage level while preserving a visual resemblance to the original logos. Similarly, for image-based logos, 9 out of 11 brands (with image logo components) can find over 80 candidate logos that transfer successfully when placed on the webpages (72%–100% success rate). More importantly, we find that adversarial logos computed with a surrogate model can transfer well to attack other detectors. For example, adversarial logos computed based on PhishIntention have a transfer rate ranging from 80% to 100% for 17 out of 18 brands against Phishpedia. Compared with prior attacks [29], we show LogoMorph is more effective end-to-end at the webpage level (i.e., by placing adversarial logos onto phishing webpages), and our adversarial logos are of a higher visual quality. [*Users:*] Adversarial phishing pages crafted with LogoMorph are slightly more noticeable by users with a true positive rate (TPR) of 0.59, compared with 0.45 TPR of unperturbed phishing pages. That being said, a 0.59 TPR means our phishing pages are still effective on users (users are performing only slightly better than random guessing) with the benefit of evading phishing detectors (while unperturbed pages cannot). The results confirm the effectiveness of LogoMorph on end users.

Contributions. Our main contributions include:

- We propose LogoMorph, a new attack against logo-based visual phishing detectors. The attack incorporates semantic-preserving manipulations to generate image- and text-based logos to create evasive phishing pages.
- By using LogoMorph, we empirically reveal the weaknesses of state-of-the-art phishing detection systems.
- We validate our attack with user studies, showing that our adversarial logos can bypass the human eye.

To facilitate future work, we make our code available [4].

2 Related Work and Preliminaries

We focus on the problem of phishing *website* detection (and evasion). Hence, studies entailing other forms of phishing (e.g., email [28], or social media [20]) are orthogonal to ours.

2.1 Phishing Website Detectors

Overview. Blocklists represent the first line of defense against phishing websites: by checking if any given URL (or domain) is contained in a set of "malicious" entries, it is possible to defuse the corresponding phishing threat [11, 31, 48]. Unfortunately, such signature-based detection approaches

only work against known phishing websites [52]—and recent studies have shown that they may be unreliable even against these [42, 43]. To protect online users against "novel" phishing websites, state-of-the-art detection methods leverage machine learning (ML) techniques [2, 7, 17]. Among these, most approaches seek to discriminate benign from malicious webpages by extracting features derived from URLs, textual web content, and/or the HTML code—and then use such features to develop ML-based detection models [15, 22, 36, 39, 46, 47, 52, 54, 55]. However, recent studies have shown that all such detectors can be *easily evaded by "perturbing" just a few features* [8,37,50,51]. For these reasons, we focus on a complementary ML-based detection approach that leverages the visual similarity of webpages. [1]

Phishing Detection via Visual Similarity. Real phishers craft their websites to be visually identical to the websites of (popular) brands: visual similarity-based methods focus on countering this quintessential property of phishing. Specifically, these methods work by *(i)* seeing if a given website is visually similar to another "reference" website (taken from a list of "protected brands"); and then *(ii)* comparing the domain of the given website with the one of the reference website's brand: if the similarity exceeds a threshold and yet the input page is not hosted under the brand's domain name, it is determined as phishing.[2] Among the ways to compute such similarity, some works rely on the screenshot of a webpage [6]. Unfortunately, recent studies revealed that the comparison of *whole-page screenshots can easily produce false positives* [30, 32]. To address this shortcoming, new systems focus on comparing the logo—which is treated as the identity indicator of a brand [30, 32]. The most recent work, PhishIntention [32], uses a Siamese neural network and an Optical Character Recognition (OCR) model to measure logo similarities and rely on other context information (e.g., the presence of a password-collecting form) to improve detection accuracy. Another work [33] provides an add-on function for PhishIntention to dynamically expand the reference list and handle "brandless" pages (orthogonal to our focus, which is adversarial phishing pages). To the best of our knowledge, these logo-based detection approaches represent the state-of-the-art, and there is evidence that they have been deployed also in production-grade phishing detectors [7, 17]).

Problem Statement. The security of logo-based phishing website detectors has not been extensively scrutinized. Indeed, the very same authors of PhishIntention [32] (and also of its predecessor, Phishpedia [30]) assessed the robustness of their solution against off-the-shelf adversarial ML algorithms [30, 32]: perhaps unsurprisingly, they found that such evasion tactics can be defused with simple countermeasures such as gradient masking [30, 32]. Two months before writing this paper, the authors of PhishGAP [29] bypassed logo-based detectors (proposed by [30, 32]) by developing a surrogate detector used to craft "adversarial logos" (by adding visual noise to the logo) that evade the actual detector. However, PhishGAP [29] *only focuses on the logo itself*: as such, the attack is only shown to bypass[3] the logo-detector—which is just a single component of the phishing detection system; and even though the authors of [29] carry out an user study to validate their method, the users are only shown[4] the logo—which is just a tiny element in a (phishing) page. Put simply, the authors of PhishGAP [29] do not perform an end-to-end evaluation by placing the "adversarial logo" on a web page, and seeing its effects on *(i)* the full-fledged phishing detection system, and on *(ii)* its end-users (in §4.7, we compare our proposed attack with [29] using end-to-end experiments).

Our Goal. To this date, it is still unclear whether logo-based detectors can truly be considered as reliable (according to [30, 32], they are) or not (according to [29], they are not). In this paper, we seek to provide an answer to this dilemma.

2.2 Threat Model

`LogoMorph` targets reference-based visual phishing detectors [6, 30, 33]. We focus on detectors that rely on logo comparisons, such as PhishIntention [32]. Such comparisons are done against a *reference list* containing logos of well-known brands (n.b., any brand may be associated with many logos).

Target System. To provide the necessary context for our attack, let us summarize how logo-based visual phishing detectors work. At a high level, these systems first extract the logo image from the screenshot of an input webpage, and then they use a mix of deep learning (typically, a Siamese neural network [32]) and OCR to create the embedding for the logo image: deep learning is used for a basic visual comparison of two logo images, whereas OCR serves to better embed text-based logos. Hence, when analyzing any given webpage to infer its legitimacy, an input logo is compared with a set of legitimate logos from the reference list with the techniques mentioned above. If the similarity between the input logo and any logo in the reference list is higher than a threshold (θ), then the two logos are considered to be "similar", thereby suggesting that the two logos belong to the same brand. If the webpage is phishing, then it will *not* be hosted under the target brand's domain name, and hence labeled as malicious [32]. Given that PhishIntention [32] represents the state of the art of such detection systems, we will use this as "baseline system."

[1] These methods are robust against evasion attempts targeting the HTML of a webpage. As a proof-of-concept, we obtained adversarial webpages generated by a recent work [8], and tested them against [32]: only 27% of these webpages evade [32]. Details in supplementary materials [5].

[2] Example: VisualPhishNet [6] can detect a phishing website that looks like the webpage of the real CHASE bank but is not hosted under `chase.com`.

[3] Even after "transferring" the adversarial logos to different detectors, the average fooling rate ranges between 10–42% (see Fig. 6(d) of [29]).

[4] We argue that the adversarial perturbations of [29] are quite visible (e.g., apparent stretching/cropping, or use of non-web logos—see [5]).

Attacker: Goal and Strategy. To bypass such reference-based visual phishing detectors, we envision an attacker who seeks to create an "adversarial logo" that *(i)* preserves the semantics of an "original" logo (e.g., it must still "resemble" PayPal) while *(ii)* achieving a low similarity (i.e., below θ) with any logo in the reference list. Indeed, doing so will induce the detector to *believe that the webpage is benign*. The threshold is there to control false alarms: a low similarity implies that the input logo belongs to a brand outside those in the reference list (e.g., maybe it is of a low-ranked website; or, alternatively, a well-known brand has just updated its logo). Hence, achieving a low similarity will prevent the domain-checking step of the detector from occurring (§2.1), and the webpage will be displayed to an end user—thereby evading the detection (if the webpage is phishing). Nonetheless, when crafting the adversarial logo, the attacker should ensure that the resulting phishing webpage does not appear more suspicious (w.r.t. the "original" phishing webpage) to the end users—i.e., the true target of phishing.

We emphasize that our goal is to assess whether such adversarial phishing *webpages* (not just the logo itself) represent a threat in practice, i.e., against systems and humans. As such, we design our experiments (§4) and user study (§5) to examine the webpage-level attacks and perform "end-to-end" evaluation (i.e., we attack the entire system, embracing the recommendations of [7]) instead of just analyzing logos in isolation (as done by most prior work, such as [29]).

Remark. Attackers must simultaneously deceive the detector and humans. Achieving only one of these is not useful.

Attacker: Knowledge and Capabilities. Our attacker has complete control on their own phishing webpages. However, from a knowledge perspective, we consider various settings.

We start with a white-box setting by assuming adversaries who have *perfect knowledge* (in terms of the model architecture and dataset[5] of the target system, i.e., PhishIntention [32]. This is to follow the recommendations from prior works [9, 16], i.e., to explore the "worse-case" robustness of a defense. The setting is the same as that of [29]. In practice, real attackers can approximate this assumption by reading related papers on the system design [6, 30, 32, 33] and collect their own datasets of logos (indeed, logos of a given brand are publicly available, and attackers can get them easily).

Then, we will relax this assumption and study the impact of adversaries who have *limited knowledge* in a black-box setting. Attackers still know the reference list and the logos (which, as we argued, is a realistic assumption), but they are oblivious to the targeted detector's low-level specifications, and *cannot directly query* the target system to observe its decisions. In this case, the attacker builds a local surrogate model (to optimize the attack), and then transfer the attack to a target system. To simulate a realistic setting, we first envision adversaries that "expect" to be attacking PhishIntention [32], but we *change* the target system (with another reference-based visual phishing detector). Specifically, we consider two cases of "mismatched" knowledge: a target system that still focuses on logo comparisons, but uses a different neural network design (i.e., Phishpedia [30]); and a target system that does not focus on logos, but on screenshots (i.e., VisualPhishNet [6]). Finally, we also simulate adversaries that still attack PhishIntention [32], but without knowing the details of PhishIntentions's logo discriminator—which is approximated by the surrogate model owned by the adversary. Through these 3 transferability experiments, we assess all these realistic scenarios and show that `LogoMorph` can be effective also without perfect knowledge of the target system.

[5] N.b.: if the attacker targets a brand that is *not* contained in the defender's reference list, the attack will automatically succeed. Hence, overlap with this dataset is detrimental for the attacker (which is a realistic setting [7]).

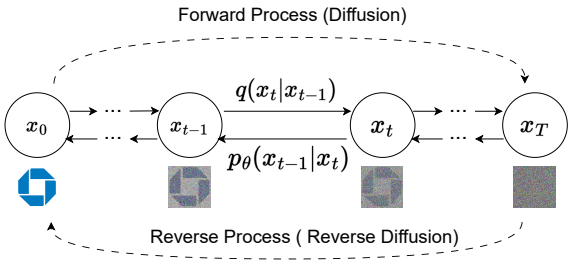

Figure 3: **Diffusion Model**—An illustration of the forward (diffusion) process and the reverse process of the diffusion model.

2.3 Diffusion Models

Given diffusion models will be used, we provide some background. Extra details are in the supplementary materials [5].

Basic Diffusion Model. The basic version of the diffusion model is called Denoising Diffusion Probabilistic Models (DDPM). DDPM (compared with generative adversarial networks or GANs) can generate high-quality images while offering fine-grained controls over image fidelity and diversity [24, 34, 40]. As a generative model, the goal of diffusion models is to learn to generate data similar to the training data. To do so, diffusion models "destroy" the training data by progressively adding Gaussian noise to the image, and then learn to recover the original image by reversing the noising process.

The idea is illustrated in Figure 3. diffusion model contains a forward (diffusion) process and a reverse (diffusion) process. The forward process (modeled by a Markov chain) is to add Gaussian noise to a clean image x_0 step by step, until reaching a version of pure Gaussian noise x_T. The reverse diffusion process is also defined by a Markov chain to transform Gaussian noise back to a clean image. The training of the diffusion model is to learn a neural network p_θ (parameterized by θ) to generate a clean image from the noise.

Improved Diffusion Model. W.r.t. DDPM, the *improved diffusion model* [40] reduces the sampling by an order of

Figure 4: **Text Logo Attack Examples**—The first row displays the brand's original logo. The second row shows attack fonts with cosine similarity (about 0.86) that is slightly below the detection threshold. The third row exhibits adversarial logos with a lower cosine similarity (about 0.79). All these fonts can bypass detection.

magnitude during the forward passes without sacrificing the quality of generated images. The idea is to learn the *variances* of the reverse diffusion process by a separate neural network to create a hybrid loss function. In §3.2, we explain how we modify this improved diffusion model for our attacks.

3 Proposed Attack (LogoMorph)

We propose to bypass logo-based phishing detectors via manipulations that operate on a logo's *semantics*—which can be either text-related (e.g., changing font) or image-related (e.g., changing the emblem), or both (cf. Figure 1). This section describes our proposed perturbation strategy in detail (against the: text in §3.1, image in §3.2, and image-text in §3.3).

Common Final Step. Ultimately, our LogoMorph attack boils down to creating an "adversarial logo" which, after replacing the "original logo" in the respective webpage, yields a phishing webpage that simultaneously deceives the detector and the human eye. Hence, for the sake of a realistic end-to-end evaluation, the last step of all our proposed attack methods entails *adding the logo to the webpage*, which we do by placing the adversarial logo in the exact same position of the original logo. In doing so, we ensure that the background color of the adversarial logo aligns with the color of its surrounding location in the "adversarial webpage" (potentially by applying post-processing techniques). We may also trim or smooth the edges of the logo for better fidelity. Note that these operations can be trivially done by our envisioned attacker, who has complete control of their phishing webpages (§2.2).

3.1 Text Logo Attack (simple)

Recall (§2.2) that our attacker wants to introduce visual perturbations that bypass detection while preserving the semantics of the original logo's text (to avoid alerting human users). This requires considering various text logo attributes such as spacing, positioning, and spelling. In other words, we cannot simply treat the text logos as "images" and blindly apply gradient-based methods [14, 27, 56], since these would yield minimal perturbations that may not fool the detector. One alternative approach is to apply generative models for text/font generation. Doing so, however, is computationally expensive; plus, the resulting text would present artifacts (e.g., poor readability or coherency) that may irremediably impair its appearance [14], thereby not fooling the users.

Idea and Preparation. We propose an easy-to-implement method to generate adversarial text logos that meet our twofold goal. The idea is to search for an *alternative font* that is visually similar to that of the original logo—while preserving its spelling, color, and spacing. This allows us to have fine-grained control over the key attributes that affect the semantics of the logo. Hence, we create a font database consisting of candidate fonts: we first include fonts collected from Google's open-source font project [1]. We then enrich this database by manually adding fonts that are used by popular brands' logos (and their variants).[6] The final database contains 2,556 fonts including 2,029 fonts from Google open-source fonts and 527 fonts from the common logo fonts.

Basic Procedure. Given the target text logo, we identify the *attack* fonts from our font database by selecting the top-K visually similar candidate fonts. We proceed as follows. **(1)** Given an "original" logo to use as the basis for our adversarial logo, we use OCR to narrow down the list of candidate fonts based on the cosine similarity between the embeddings (generated by the OCR) of any candidate font and the original font. The embedding is obtained by using ASTER [49]. We use OCR embedding due to its proficiency at processing "text." We stress that this OCR technique may not be the exact one used by the detector: this is not a problem, because this step is to search for "similar fonts inside our database" (which supposedly will deceive the actual OCR of the detector). **(2)** Then, we rank and select the top-K candidates with a cosine similarity below the target threshold (θ) as our *attack* fonts (see Figure 4). **(3)** Finally, for every candidate font, we re-create the logo by using the chosen font, and by preserving the spelling, color, and capitalization of the original logo.

Additional Variants. In addition to font replacement, attackers may apply additional semantic-preserving perturbations. For example, they can change the default capitalization as well as the bold/italics styles of the text. Such simple changes can increase the pool of candidate fonts for a given brand. Another source of variance is to change the spacing between letters. In our evaluation, we only apply the basic steps described above if adversarial logos can be identified. If the basic steps fail to find an adversarial logo that reaches the desired similarity threshold, then we consider additional variants (with a cost of potentially slightly degrading the integrity of original semantics). In our experiments, we always find candidate fonts without applying additional perturbations.

[6] E.g., *Segoe UI* is the font for the Microsoft Outlook logo. Hence, we add *Segoe UI* and its variants (*Segoe*, *Segoe-italic*, *Segoe-bold*) to the database.

3.2 Image Logo Attack (complex)

Unlike text-based logos, to the best of our knowledge, there are no semantics-preserving manipulations that are readily available (i.e., font change) for image-based logos.

Idea and Preparation. We propose a new method to learn semantics-preserving manipulations. The idea is to use a diffusion model to learn the *style* of the logos of the target brand (e.g., the logos of PayPal) and generate adversarial logos that preserve such style (to deceive users) while eluding the detector's similarity check [30, 32]. We **(1)** take the *Improved Diffusion Model* [40] as our basic pre-trained model. To adapt this model for logo generation, we **(2)** fine-tune it using logo images for the 277 protected target brands from PhishIntention's [32] logo dataset (3,061 logo images in total). This fine-tuned model will serve as our global model.[7] Finally, to learn the logo style of a *specific* target brand, we **(3)** propose and develop an *Adversarial Diffusion Model*.

Adversarial Diffusion Model. We construct the Adversarial Diffusion Model by modifying the fine-tuned Diffusion model's loss function to capture the additional attack goal. Recall that the Diffusion model is trained to generate a clean image from pure Gaussian noise (see §2.3). More specifically, as shown in Figure 3, the forward process is to progressively add Gaussian noise to a clean image x_0 step by step, until completely destroying the image to pure Gaussian noise. We train the Diffusion model by learning a neural network p_θ to capture the reverse process to generate a clean image x'_0 from the noise. We use L_{vlb} to represent the original loss function of the *Improved Diffusion Model* [40].

To make sure the generated image x'_0 can bypass phishing detectors, we insert a loss term such that the generated image x'_0 has a large latent distance from the original x_0 in the phishing detector's feature space. Using PhishIntention as an example, the similarity between x'_0 and x_0 should be lower than the detector's threshold. Here, we use the pre-trained OCR-Siamese network from PhishIntention (denoted as P_ϕ) to compute the cosine similarity between the embeddings of two images (denoted as $p_\phi(x'_0 - x_0)$). We use τ to denote the target similarity threshold. The new loss function of the adversarial diffusion model is:

$$L = L_{vlb} + \beta * L_{attack} \quad \text{(Eqn. 1)}$$

$$L_{attack} = max(0, p_\phi(x'_0 - x_0) - \tau) \quad \text{(Eqn. 2)}$$

Here, β is the scaling factor to balance L_{vlb} (controlling the *image quality and diversity* of the generated logos) and L_{attack} (controlling the *attack effectiveness*). We make β as an adaptive factor during training: it starts from small, (e.g., 0.0002) and gradually increases with the number of steps as we train the diffusion model. We set β to be small in early training iterations to ensure images are generated with high quality and close to the original logos. After that, we gradually increase β to shift the model weights towards our goal of reducing the cosine similarity (in the feature space) between the generated and original images. The intuition is to first search for the *general region* in the feature space that represents the "style" of the original logos, and then perform more *fine-grained* search to identify logos to achieve the attack goal. The adaptive scaling β can be set differently for different logo brands. By default, we set the value to be small (no larger than 0.01) to prevent our L_{attack} from overwhelming the original Diffusion model's loss term L_{vlb}. Empirically, according to our experiments, if β is too large, it will skew the model weights to generate low-quality images.

Note that, when training the model, we freeze the model weights of the OCR-Siames model (p_Φ) since it serves as a proxy for the target phishing detector, and we only update the weights of the Diffusion model p_θ. In addition, the original OCR-Siamese model has built-in modules (e.g., image resizing) that will lose gradients. We overcome this by re-implementing the OCR-Siamese model with Pytorch to store gradients along the process, to ensure gradients can be back-propagated. We will share this implementation.

Data Augmentation. A challenge of training the Adversarial Diffusion Model is that there may be only few logo images for each brand. To increase the size (and diversity) of the training data, we augment the training set via image transformations (e.g., rotating, central cropping, flipping, or Augmix[8]) on logo images. We intentionally control all transformations so that the changes preserve the semantics of the logos (i.e., the transformed logos can still be detected as the correct phishing brand by PhishIntention). For example, we only rotate 1 or 2 degrees for the logo image and then do the central cropping by trimming off 2% of the edges. In addition, we patch all images to make sure they are square images.

3.3 Image-Text Logo Attack

For logos that contain both image and text parts (e.g., BOA and Chase, as shown in Figure 1), our strategy is to first use an adversarial diffusion model to generate the image part, and then search for the corresponding fonts that allow for the combined logo to bypass the detection threshold. Here to improve efficiency, when searching for candidate fonts, we will search from the set that is already within a similarity of 0.6–0.87. These fonts retain a good chance of success (for the text part) as well as a good resemblance to the original fonts.

[7] We use a Diffusion model rather than a GAN (which inspired [29]) because the former allows fine-grained control over the image generation process to balance between fidelity and diversity [24, 34, 40]. This is important: we not only aim to preserve the "style" of the brand's logos but also need to introduce visual differences large enough to bypass the detector.

[8] Augmix [23] is a data augmentation method to interpolate and mix images to introduce variance while preserving the original images' semantics.

Logo Type	#	Brands
Text Only	7	Comcast, DocuSign, eBay, Google, Instagram, Netflix, Yahoo
Image Only	3	AT&T, Canadian Imperial Bank of Commerce (CIBC), DHL
Image-Text	8	Amazon, Bank of America (BOA), Chase, Dropbox, LinkedIn, Outlook, PayPal, Spotify

Table 1: **Experimental Logo Brands**—We select logos from 18 popular brands for our main experiments. Among these, 7 have logos that only contain text, 3 only images, and 8 both text and images.

4 Attack Evaluation

We now implement `LogoMorph` and carry out an end-to-end evaluation to assess its effectiveness. At a high level, our evaluation seeks to answer three research questions (RQ):

RQ1 (Perfect knowledge) How effective is the text- (§4.2), image- (§4.3), or image-text-logo (§4.4) attack against the state-of-the-art visual phishing detector [32]?

RQ2 (Limited knowledge) How well can the attack transfer across different detection systems? (§4.5, §4.6)

RQ3 (User Study) How well can adversarial phishing webpages deceive human eyes (i.e., end users)? (§5)

We will also compare `LogoMorph` with PhishGAP [29] in §4.7.

4.1 Experimental Setups

Dataset. We use the same dataset used by PhishIntention for our experiment [32]. The dataset contains 277 brands in the reference list (with 3,061 logo images). For a focused, in-depth analysis under various scenarios, we will consider 18 brands (prioritizing those that are most targeted by phishing websites [3]) in our main paper: these brands are reported in Table 1, showing that they cover all three categories of logos (text-only, image-only, and image-text logos). Nonetheless, in Appendix C, we have extended the "perfect knowledge" experiments to a total of 110 brands, demonstrating that `LogoMorph` is broadly applicable to a variety of brands.

When we evaluate our attack in an end-to-end fashion, we use webpage screenshots and URLs from the dataset provided by Phishpedia [30], which contains 30K phishing webpages and 25K benign webpages.[9] For this experiment, we select one phishing screenshot for each brand, ensuring that the logo of the screenshot aligns with our predefined logo categories.

Evaluation Metrics. For the perfect knowledge setting, we focus on PhishIntention to examine both the logo-level and webpage-level impact—i.e., we assume that the attacker "knows" the details of [32]. At the logo-level, we report the number of candidate logos that our method generates (i.e., similarity below the detection threshold $\theta = 0.87$ found by PhishIntention's creators). At the webpage level, we report the number and ratio of pages that bypass end-to-end detection.

[9] Among these 25k, only 21,974 have URLs for us to verify and determine the brands (which will be used in false positive assessment in §6).

Brand	# Candidate Fonts	Rate
DocuSign	2,556	1.00
LinkedIn	2,556	1.00
Yahoo	2,550	0.99
Netflix	2,546	0.99
Instagram	2,532	0.98
BOA	2,088	0.82
Comcast	1,903	0.73
PayPal	1,839	0.72
Amazon	1,290	0.50
Spotify	877	0.34
Chase	624	0.24
Outlook	597	0.23
Dropbox	447	0.18
Google	388	0.15
eBay	362	0.14

Table 2: **Logo-Level Results (Text Logo)**—For each brand, we report the number of fonts (out of 2,556 fonts in the database) with a similarity below the OCR similarity threshold of 0.87.

For the limited knowledge setting, we assess the *transferability* of our adversarial logos to other detectors that are not seen during the logo-generation process.

Generic Procedure (and ancillary user study). We adopt a similar procedure for our evaluation. We first generate adversarial logos with `LogoMorph`, and see which achieve a similarity (w.r.t. the most similar logo of the reference list) below the threshold "known" by the envisioned attacker (typically 0.87). Then, we consider a subset of logos that fall below such a threshold, and apply them to phishing webpages (by replacing the "original" logo with our adversarial variant) for end-to-end evaluation. To do so, we pick logos whose similarity falls between 0.6 and 0.87. Such a range allows us to carry out an extensive analysis, encompassing logos that are more likely to preserve the original semantics (i.e., closer to 0.87) and also those that may do so only marginally (i.e., closer to 0.6). Nonetheless, we validate this range by carrying out an *additional user study* (n=100), discussed in Appendix B, where we inquire whether users think that our adversarial logos resemble the targeted brand. The results show that, for some users, even logos with $0.6 \leq \text{Sim} < 0.7$ can resemble the target brands, but a higher similarity (e.g., $\text{Sim} \geq 0.7$) is more desirable. In practice, attackers should (and, realistically, would) use `LogoMorph` by favoring logos that bypass detection and which have higher similarity.

4.2 Text-Logo Attack Effectiveness

We start with the text logo attack. For this evaluation, we consider all the text-only logos in Table 1. To expand the evaluation set, we also include the image-text logos in Table 1 by only focusing on the text part (i.e., cropping the text part from the logo). In total, 15 brands are considered.

Logo-level. The logo-level analysis investigates the likelihood of locating an attack font with an OCR cosine similarity

Brand	# of Success Fonts	Rate	Avg. Sim.
BOA	200	1.00	0.73
Outlook	200	1.00	0.79
Spotify	200	1.00	0.75
Instagram	199	0.99	0.76
Dropbox	199	0.99	0.75
Amazon	195	0.98	0.78
Chase	194	0.97	0.82
eBay	183	0.92	0.72
DocuSign	178	0.89	0.81
Comcast	145	0.73	0.84
Google	121	0.61	0.80
Netflix	80	0.40	0.88
LinkedIn	54	0.27	0.88
Yahoo	39	0.20	0.89
PayPal	37	0.19	0.90

Table 3: **Webpage-Level Results (Text Logo)**—Number of fonts that lead to end-to-end bypass of PhishIntention. We consider Top K=200 fonts from Table 2, and report the successful ones.

lower than the similarity threshold (0.87). Given a target logo (extracted from the target brand's webpage), we search the entire database of 2,556 fonts and report the number of candidate attack fonts and ratio. The results are shown in Table 2. A higher number means attackers would have more candidate fonts to select from when crafting the attack phishing pages.

Table 2 shows that we can find at least 362 attack fonts that bypass the OCR similarity threshold for each of the target logos (on average 1,544 attack fonts). We notice that brands that use unique/uncommon font styles are easier to attack—DocuSign, Instagram, and Netflix have more candidate fonts. In comparison, brands (e.g., Dropbox, Outlook, and Ebay) that use common-looking fonts such as *Gotham Circular*, *Segoe*, and *Sans-Serif* have a lower rate of candidate fonts.

Webpage-level. Given the candidate logos, we then select those to add to the webpage for end-to-end evaluation. Considering large visual differences may raise suspicion among users, we perform some control on the *quality* of the adversarial logo using OCR similarity. Through manual inspection, we find that fonts with an OCR similarity over 0.60 would look reasonably similar. We select top-K candidate fonts with an OCR similarity *(i)* lower than 0.87 (for attack effectiveness) but *(ii)* higher than 0.60 (for visual quality). We add the logo to the corresponding brand's phishing webpage and report the number of webpages that bypass PhishIntention end-to-end.

As shown in Table 3, we identify successful attack fonts for all brands to bypass the end-to-end detection. For brands such as BOA, Outlook, and Spotify, all the attack fonts from the top-K (K=200) can successfully evade PhishIntention. Such high success rates are common in image-text logos (6 out of 8, where only the text part is used). For brands such as Yahoo, LinkedIn, and PayPal, we identify a smaller number of adversarial logos (37–54) that are successfully end-to-end. Table 3 also reports the average similarity between the testing logo and their closest reference logo computed by PhishIntention. Note that this similarity is different from the OCR similarity used to rank candidate logos (OCR only compares testing fonts with original fonts, but does not consider all the reference logos of the detectors). Not too surprisingly, Yahoo, LinkedIn, and PayPal's testing logos tend to have a higher average similarity to the reference logos (above 0.88). This suggests that these logos, although distanced enough from their original fonts, are still not distanced enough from some of PhishIntention's reference logos. We show additional successful examples in supplementary materials [5].

Takeaway. We always find at least 362 fonts per brand (out of 2,556) that bypass the logo-detector. Bypassing the end-to-end system is harder, but possible, with 37–200 fonts enabling such evasion per brand.

4.3 Image-Logo Attack Effectiveness

To evaluate the image logo attack, we consider all the image-only as well as the image-text logos (the latter after removing the textual part) of the brands in Table 1 (11 brands in total).

Logo-level. Given the detection threshold (θ=0.87) of PhishIntention, we set our target similarity to a lower value of $\tau = 0.7$ for $L_{attacck}$ in Eqn. 2, to increase the chance of success. Recall that, while training the Adversarial Diffusion Model (§3.2), we save different versions of the model at different training iterations (with different fidelity and diversity trade-offs). We use these to standardize the evaluation procedure. Specifically, given a target logo, we obtain adversarial logo images at 5 different training stages and generate 100 images at each stage, thereby yielding 500 images for each brand. Intuitively, images generated during early iterations have a better diversity, meaning they deviate more from the data seen during training (i.e., the original logo); an example would be the "Attack-2" images in Figure 1. However, as the adversarial diffusion model is trained over more iterations, the generated images have a better fidelity w.r.t. the original logos; an example would be the "Attack-1" images in Figure 1. Then, we examine how many adversarial logos can bypass the detection threshold of 0.87 of PhishIntention [32]. Table 4 shows the result in the middle column: for 10 out of 11 brands, there are over 100 candidate logo images (out of 500 generated).

Webpage-level. To select logos to add to the webpages, we again perform basic quality control. Through manual inspection, we find that a cosine similarity of 0.6 yields logo images that are suitable. Figure 5 shows example logo images at different similarity levels compared with the original (all below θ=0.87). As shown in Table 4 (the right column), nearly 40% of the generated logo images fall within this range. Therefore, we take the logos with a similarity of 0.6–0.87 (Table 4) and place them on phishing webpages.

The end-to-end detection results are reported in Table 5. For Amazon, we only used 362 logos (instead of 433) because the style (i.e., size and color scheme) of the 81 omitted logos did not match well with the corresponding webpage screenshot.

Brand	# of Success Logos (Rate)	
	Sim <0.87	0.6<Sim<0.87
Amazon	500 (1.00)	433 (0.87)
PayPal	311 (0.62)	308 (0.62)
LinkedIn	357 (0.71)	244 (0.49)
DHL	236 (0.47)	216 (0.43)
Dropbox	212 (0.42)	196 (0.39)
Chase	195 (0.39)	184 (0.37)
BOA	220 (0.44)	183 (0.37)
CIBC	188 (0.38)	152 (0.30)
AT&T	104 (0.21)	102 (0.20)
Outlook	105 (0.21)	99 (0.20)
Spotify	76 (0.15)	73 (0.15)

Table 4: **Logo-level Results (Image Logo)**—Number of generated logos images that bypass $\theta = 0.87$ threshold among 500 testing logos. We also report the number and % of logos with a similarity above 0.6 to indicate good image quality.

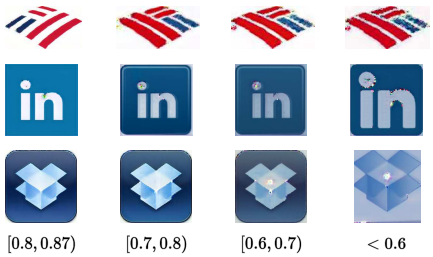

Figure 5: **Image Logo Attack Examples**—We show example logo images of different similarity levels compared with the original logos. All of them are below the detection threshold of 0.87.

We find that most of the logos that succeeded at the logo level retain their success after being added to webpages. 9 out of 11 brands have a success image rate of 0.72 or higher after transferring to the webpage level. Not too surprisingly, brands with a higher success rate, such as Amazon, Paypal, DHL, and Dropbox (success rate 0.89–0.1), often have a lower average similarity (0.67–0.70), meaning they are further away from the detection threshold. Spotify and Outlook logos have a lower rate. For Spotify, its lower rate can be explained by the relatively high similarity score to the original logos (0.83). However, for Outlook, a possible explanation is the Outlook logo size[10] is too small (about 60×60). As a validation test, we enlarge the logo size slightly to 100×100 when placing it on the webpage—the success rate goes up from 0.44 to 0.75. We show successful examples in supplementary materials [5].

Takeaway. Our method is always able to generate adversarial logo-images that bypass the logo-detector (76 in the worst case) and the end-to-end system (44 in the worst case).

4.4 Image-Text Logo Attack Effectiveness

Finally, we evaluate the attack effectiveness for image-text logos for the 8 corresponding brands in Table 1. Our method, as described in §3.3, is to combine the successful image and text parts and directly add them to the webpage screenshots [11]. Here, we combine successful text logos in Table 3 and image logos in Table 5. This immediately creates a large number of candidate webpage screenshots. For example, for Bank of America (BOA), there are 183 successful text-logos (i.e., fonts) and 200 image-logos, thereby yielding 183×200=36,600 candidate image-text-logos.

Table 6 shows the number and ratio of successful webpages that bypass PhishIntention end-to-end. The result confirms that combining text and image logos creates a large number of successful webpages for attackers to choose from. The number of successful logos end-to-end ranges from several thousand to tens of thousands. Interestingly, the average similarity of such image-text logos is usually high (Avg.Sim>0.8), which is desirable according to our logo-level user study in Appendix B. In practice, the attackers can select logos with higher visual quality and the largest distance to broadly apply them in phishing campaigns.[12]

Takeaway. For each considered brand, we find at least 3,596 image-text-logos that bypass the target system end-to-end.

Brand	# Success Logos (# Tested)	Rate	Avg. Sim
Amazon	362 (362)	1.00	0.67
PayPal	308 (308)	1.00	0.67
DHL	194 (216)	0.90	0.71
Dropbox	174 (196)	0.89	0.70
BOA	154 (183)	0.84	0.73
Chase	146 (184)	0.80	0.80
CIBC	121 (152)	0.80	0.72
AT&T	81 (102)	0.79	0.76
LinkedIn	175 (244)	0.72	0.65
Spotify	50 (73)	0.68	0.83
Outlook	44 (99)	0.44	0.75

Table 5: **Webpage-Level Results (Image Logo)**— Number of logos that bypass the end-to-end detection of PhishIntention after being placed on actual webpages. We only test logos from Table 4.

Brand	# Success Logos (# Tested)	Rate	Avg. Sim.
Amazon	37,970 (70,590)	0.54	0.857
BOA	13,479 (36,600)	0.37	0.866
Chase	18,601 (35,696)	0.52	0.869
Dropbox	29,773 (39,004)	0.76	0.807
LinkedIn	6,249 (13,176)	0.47	0.877
Outlook	11,387 (19,800)	0.58	0.849
PayPal	6,383 (11,396)	0.56	0.855
Spotify	3,596 (14,600)	0.25	0.891

Table 6: **Webpage-Level Results for Image-Text Logos**—Number of Image-Text logos that bypass the end-to-end detection of PhishIntention after being placed on actual webpages. The image-text logos are generated by combining the image part and text part generated by the respective attack algorithms.

[10] For consistency, we use the original logo size to fit the logo within the webpage layout. However, real attackers are not bound to this constraint.

[11] For this evaluation, we directly use a script to add the image and text parts to the webpage (instead of assembling a logo first) because it is easier to control the alignment of the text and image parts w.r.t the webpage layout.

[12] We show the trade-off between similarity and bypass rate in Appendix A

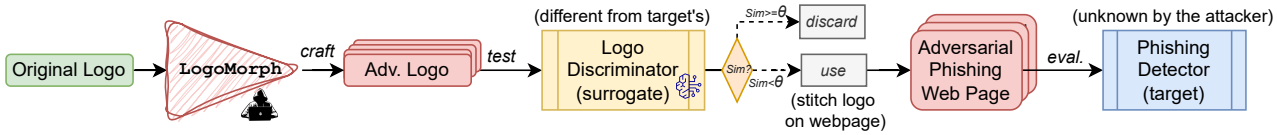

Figure 6: **Our Blackbox Experiment Setup.**—We use a surrogate logo discriminator (which is different from the one used by the target model) to generate and select adversarial logos via `LogoMorph`. Logos that bypass the surrogate discriminator (by achieving a low similarity) will be used to attack the targeted phishing detector at the webpage level.

Brand	# Bypass Phishpedia (# Tested)	Rate
DocuSign	178 (178)	1.00
Comcast	145 (145)	1.00
Yahoo	39 (39)	1.00
LinkedIn	6,172 (6,249)	0.99
Amazon	37,177 (37,970)	0.98
Google	116 (121)	0.96
Netflix	77 (80)	0.96
Instagram	192 (199)	0.96
eBay	170 (183)	0.93
Chase	17,361 (18,601)	0.93
Spotify	3,291 (3,596)	0.92
Outlook	10,361 (11,387)	0.91
AT&T	70 (81)	0.86
PayPal	5,497 (6,383)	0.86
CIBC	108 (121)	0.89
DHL	156 (194)	0.80
Dropbox	23,746 (29,773)	0.80
BOA	7,652 (13,479)	0.57

Table 7: **Transferability to Phishpedia (All Logos)**—Number of adversarial phishing webpages (bypassing PhishIntention [32]) that successfully bypass another phishing detector (Phishpedia [30]).

4.5 Blackbox Attack against Phishpedia and VisualPhishNet

To simulate an attacker in a black-box setting, we examine how likely an adversarial logo designed to bypass PhishIntention [32] can transfer to *a different phishing detector*. As illustrated in Figure 6, PhishIntention is used as the local surrogate model of the attacker, which crafts logos to create phishing webpages to attack different detectors including Phishpedia [30] and VisualPhishNet [6]. We note that this experiment is more realistic than the one carried out in PhishGAP [29], wherein the adversarial logos were assessed merely against neural networks having a different architecture.

Attacking Phishpedia [30]. For this experiment, we take all the webpage-level screenshots (including all 18 brands in Table 1) that bypass PhishIntention [32] and we test them against Phishpedia [30] (which is the predecessor of PhishIntention). The results are shown in Table 7 (recall that image-text logos have a higher number of testing screenshots due to the many combinations of the image and text parts—§4.4). We observe that the overall transferability from PhishIntention to Phishpedia is high. For text-only brands (e.g., DocuSign, Comcast, Yahoo), nearly 100% of the adversarial pages can bypass Phishpedia. Note that Phishpedia does not use OCR features for their embedding and yet the font manipulation remains successful. The transferability of image-only logos (e.g., ATT, CIBC, DHL) to Phishpedia is also high as almost 90% of the adversarial webpages are successful. For the image-text logos, the transferability is slightly lower than other categories. For most brands, at least 80% of the webpages bypass Phishpedia. The results are consistent with those in §4.4 that the successful text and image logos combined together may not guarantee success. A possible explanation is that image-text logos offer more information (w.r.t. image or text alone) for the detectors to pick up some similarity.

Attacking VisualPhishNet [6]. Next, we test the transferability from PhishIntention to VisualPhishNet [6]. Recall that VisualPhishNet evaluates the visual similarity based on the *whole-page* screenshots instead of logos. This presents a significant difference between the two detectors. However, we expect our adversarial logos to still be somewhat effective due to a simple intuition: VisualPhishNet is trained with a variety of phishing and benign webpages *from the same brand*. Hence, although these webpages may exhibit high diversity (in terms of, e.g., colors and layout), an element that is likely to be "invariant" is the brand's logo. As such, our hypothesis is that the logo is an important feature implicitly learned by VisualPhishNet. (Due to the space limit, we present these results in Table 11 in the Appendix.) The results confirm our intuition to some extent. The evaluation covers 15 brands (three brands in Table 1 are not within VisualPhishNet's reference list and thus are omitted). Out of these 15 brands, only one never transfers, while two transfer poorly (success rate 0.01). In contrast, 6 have some transferability (success rate between 0.12–0.56) while 6 have a high transferability (success rate above 0.84). Intriguingly, three brands (AT&T, Instagram, LinkedIn) have a perfect evasion rate against VisualPhishNet [6].

Takeaway. Of the 18 brands in Table 1, our adversarial logos have a transfer success rate $\geq$ 80% for 17 brands against Phishpedia [30] and 6 brands against VisualPhishNet [6].

4.6 Blackbox Attack against PhishIntention

In §4.5, we performed a black-box experiment where the attacker uses PhishIntention as the local *surrogate* to compute the attack logos, which are then used to attack two target detectors: Phishpedia [30] and VisualPhishNet [6]. In this section, we explore the feasibility of black-box attacks against

PhishIntention itself (i.e. assuming the attacker has *no access to the internal embedding of PhishIntention*).

Setup. We explore a similar idea of training a surrogate locally (as shown in Figure 6). Under the black-box setting, the attacker can train a local *logo discriminator* that aims to approximate that of PhishIntention (without the query access to PhishIntention). To simulate the scenario where the attacker has *limited knowledge*, we intentionally use *a different architecture* for the attacker's local surrogate: the local surrogate has a simple Siamese neural network to create the embedding for input images. Compared with the target PhishIntention detector, the attacker's local surrogate does not have the OCR component for the feature embedding. Given the architecture and embedding are different, the attacker would also need to pick a different similarity threshold for optimizing the attack. For our experiment, the attacker selects threshold θ such that the surrogate archives a comparable accuracy on logo classification tasks (the resulting θ=0.83, 99% identification rate). We expect such differences (threshold and architecture) will affect the attack outcome to some extent.

Method. For text-only logos, we use the local surrogate to select text fonts that bypass the similarity threshold (measured by the *local surrogate*). Among the ones that bypass the local surrogate, we rank them based on their similarity to the original fonts and select the top 200 candidates. To improve transferability, and to simulate a conservative approach by the attacker (who does not know the threshold of the targeted detector), we selected the 100 text logos ranked between 100–200 to use against the "real" PhishIntention. For image-only logos and the image part of the image-text logos, we also use the local surrogate to generate and select attack logos. Like before, we generate 200 candidate attack logos per brand and select those that bypass the local surrogate logo discriminator (with a similarity score between 0.6 and 0.83). The number of qualified image logos varies from 30 to 165 logos per brand (see Table 12 in the Appendix). Using these logos, we then perform the *webpage-level* evaluation against the real PhishIntention. For image-only logos, we add the logos directly to the corresponding webpages and test them against the real PhishIntention. For image-text logos, we combine the image parts (selected above) with 100 text logos (that bypassed the local surrogate) before adding the combined logos to the web pages and testing against PhishIntention.

Results. Table 8 shows the end-to-end testing result at the webpage level against the real PhishIntention (all logos). The black-box success rate varies across brands. For example, Instagram has a 100% success rate and AT&T has 96%. 10 of the 18 brands have a success rate over 35%. Two brands have a transferability lower than 10% (PayPal and LinkedIn). On the one hand, these results suggest that incomplete/mismatched attacker knowledge (i.e., the discrepancies between the local surrogate model and the target model) leads to reduced attack effectiveness; this is expected—e.g., for PhishGAP [29], the

Brand	# Succeed (# Tested)	Rate	Avg. Sim.
Instagram	100 (100)	1.00	0.76
AT&T	76 (79)	0.96	0.80
DHL	70 (81)	0.86	0.80
DocuSign	82 (100)	0.82	0.85
CIBC	43 (63)	0.68	0.87
BOA	5,943 (9,300)	0.64	0.82
Outlook	1,384 (3,000)	0.46	0.87
Dropbox	3,002 (6,900)	0.44	0.88
Spotify	2,989 (7,200)	0.42	0.87
Yahoo	35 (100)	0.35	0.88
Amazon	3,067 (9,900)	0.31	0.89
Chase	2,145 (7,800)	0.28	0.90
Comcast	25 (100)	0.25	0.88
eBay	24 (100)	0.24	0.87
Google	22 (100)	0.22	0.89
Netflix	10 (100)	0.10	0.89
LinkedIn	1,021 (12,800)	0.08	0.91
PayPal	319 (16,500)	0.02	0.92

Table 8: **Transferability to PhishIntention (All Logos)**—Number of adversarial webpages (using attack logos computed by a local surrogate model) that bypass the *target PhishIntention [32]*. "Avg. Sim." reports the average similarity between the testing logos and closest reference logos in PhishIntention on a webpage level.

≈95% (logo-only) fooling rate in a white-box setting drops to 10–42% in a black-box setting. On the other hand, attackers in practice may perform such black-box tests by hosting phishing webpages (with different attack logos) and later checking if the webpages get taken down or reported (e.g., on [3]) and try again. For LinkedIn and PayPal, even though their success rate is low, the number of successful logos is high (1,021 and 319), which means they have many viable logos to choose from for large-scale phishing campaign deployment.

Takeaway. Overall, our black-box experiments (vs Phishpedia in Table 7, VisualPhishNet in Table 11, and PhishIntention in Table 8) confirm the transferability of `LogoMorph`. Certain brands have high bypass rates across all three settings, e.g., Instagram (avg=0.99) and AT&T (avg=0.94).

4.7 Comparison with Existing Attacks

Objective. We now perform a hard-comparison with the attack proposed in PhishGAP [29], whose threat model aligns with ours (§2.2). Differently from our `LogoMorph` PhishGAP seek to craft an adversarial logo by generating noise—but without accounting for any "semantics" of the resulting logo. This is done by learning a "generative adversarial perturbation" by repeatedly attacking the target logo-detector (which is considered to be completely known). However, despite showing that the resulting adversarial logos can bypass the target detector (and do exhibit some form of transferability to similar detectors), no consideration is given to an end-to-end evaluation. Hence, we scrutinize whether the adversarial lo-

gos crafted via PhishGAP retain their effectiveness against the full-fledged phishing detection system of PhishIntention [32].

Method and Results. We obtain the source code (which is publicly available) and the adversarial logos (which are provided upon request) of PhishGAP [29]. We filter these logos so as to include only those of the 18 brands in Table 1. In addition, we only consider adversarial logos whose original logo can be detected by PhishIntention (otherwise, the impact of PhishGAP would be zero). This leaves us with 2,057 adversarial logos for the 18 brands. Then, we put these logos into the corresponding brand's webpages (the same ones used in our evaluation), and run PhishIntention end-to-end. We find that only 113 webpages screenshots (5.49%) bypass PhishIntention. The successful logos (compared with their original logos) have a cosine similarity of 0.77. However, due to its "unregulated" generative method, PhishGAP's adversarial logos present remarkable semantic-level differences (e.g., cropping, disproportionate stretching, unusual background) that impair their evasiveness end-to-end. We present some successful logos in supplementary materials [5].

Takeaway. Of the 2,057 adversarial logos generated by PhishGAP [29], only 5.5% evade PhishIntention [32] end-to-end (despite bypassing its logo-discriminator).

5 User Study (Does it *also* deceive humans?)

After showing that our `LogoMorph` attack can bypass a "machine", we carry out a user study to verify if our adversarial logos can also deceive "humans", i.e., the true target of phishing. Indeed, if we only wanted to bypass the detector, we could simply generate a completely different logo—but this would inevitably alert a user: real phishers do not want this. As discussed in the threat model (§2.2), for this user study, we will focus on webpage-level evaluation (instead of just analyzing the logo itself), considering that during real-world phishing attacks, logos are not presented to users in isolation.

5.1 Survey Design

We conduct two user studies. The first study shows users benign webpages and *adversarial* phishing webpages (adding the logos generated by `LogoMorph`); we denote this as *adversarial study*. The second study shows users benign webpages and *unmodified* phishing webpages, i.e., without any adversarial logo; we denote this as *baseline study*. A participant can only participate in one of the two studies (not both).

Organization. The two studies share a similar structure. **(I)** A participant starts by reading a consent form and giving their consent. Then they read a brief summary of this study. Here, we explicitly inform the participant that *the study is about detecting phishing webpages*. We expect this to have a *priming* effect, prompting participants to get ready for the phishing detection task. In practice, users may be *unprepared* when they encounter phishing websites. As such, this highly prompted setting can help estimate the upper-bound performance of users. This priming effect has been confirmed in previous phishing studies [25] where highly prompted participants have a better phishing detection performance than unprompted participants. We use this prompted setting to explore the best-case for users and the worst-case for attackers. **(II)** The participant examines 18 webpages (in the form of screenshots, without URL; we do not deploy any phishing webpages on the Web) of the 18 brands listed in Table 1: 9 brands are randomly selected to present the *legitimate* webpage of the brand, and the remaining 9 brands present the *phishing* webpages. This ensures that a participant only sees a given brand *exactly once* (e.g., if a participant sees the real page of PayPal, this participant will not see a phishing webpage for PayPal). Below each screenshot, the participant must answer two questions: (Q1) "*How do you rate the legitimacy of this webpage?*"—the participants select answers from a six-point Likert scale from 1 (definitely phishing), 2 (very probably phishing), 3 (probably phishing but not sure), 4 (probably legitimate but not sure), 5 (very probably legitimate), and 6 (definitely legitimate). This is to avoid a neural determination. The other question (Q2) is "*What specific components/indicators on the webpage have influenced your choice?*"—a short answer is provided using an open text box. **(III)** Finally, participants answer demographic questions about their age group, gender, and education level.

Selection of Phishing Webpages. The main difference between these two user studies is the type of phishing webpages used. The *adversarial* study uses adversarial webpages generated by (i) replacing the logos with the adversarial logos produced by `LogoMorph` while ensuring that (ii) the resulting webpage bypasses PhishIntention's end-to-end detection. For this user study, we choose high-quality adversarial logos ($0.8 \leq \text{Sim} < 0.87$), which is justified by our "ancillary" study, discussed in Appendix B, revealing that such logos can better resemble the targeted brand (and real-world attackers would opt for these). For the *baseline* study, we use the "*unmodified*" phishing webpages that do not carry adversarial logos. The ordering of the brands in the questionnaire is randomized to mitigate biases from the order effect (e.g., participants may get tired/more familiar with the task towards the end).

Recruitment and Ethics. Our study was reviewed and approved by our IRB. We recruit participants from Prolific. We chose Prolific over other platforms such as MTurk for its higher quality of the participants' answers [44]. During the study, we did not collect any personally identifiable information (PII); all participants were anonymous and voluntary and could withdraw their data at any time during and after the study. Each participant is compensated with $2.2 for completing the survey. On average, each participant spends 17.6 minutes on the survey. In total, we recruited n=150 participants including 100 for the primary adversarial attack study

Study	Accuracy	TPR	TNR
Adversarial	0.69	0.59	0.79
Baseline	0.60	0.45	0.75

Table 9: **Users Study Results**—The adversarial study uses phishing webpages with our adversarial logos. The baseline study uses original phishing pages. We report the overall accuracy, true positive rate (TPR), and true negative rate (TNR).

and 50 for the secondary baseline study. The demographics of the participants are shown in supplementary materials [5].

5.2 User Study Results

Detection (Q1). The results are summarized in Table 9. For each study, we report the overall phishing detection accuracy, as well as the true positive rate (TPR) and true negative rate (TNR). First, we observe that users have a slightly better performance in detecting adversarial phishing pages (TPR=0.59) than in detecting unperturbed phishing pages (TPR=0.45). We run a Chi-squared statistic test to confirm the difference is statically significant (χ^2=23.3, p<0.001). However, the adversarial phishing pages are still difficult to detect by users: a TPR of 0.59 is only slightly better than random guessing. Recall that this performance is obtained under a high-prompting condition where users are prepared for (and focused on) the detection tasks. In practice, the attack is likely to be more effective when users are unprepared [18]. The benefit of adversarial webpages (w.r.t. the original ones) is that they can bypass automated detection. In addition, we find that, across both studies, participants have a higher TNR (i.e. recognizing legitimate pages) than the TPR, i.e., they can better recognize legitimate webpages than phishing ones. This observation is consistent with prior studies [25].

Reasoning (Q2). Upon examining participants' answers to the second question (indicators that influenced their decisions), we find a surprising occurrence: *logos are not the only reason* for detecting adversarial phishing pages. More specifically, in the adversarial study, only 23% of the participants who correctly identified a webpage to be phishing mentioned "logo" in their responses. In comparison, a much greater portion (36%) mentioned other factors (e.g., color scheme, overall layout, typos)—all of which were *already present in the original phishing webpage*. We again run a Chi-squared test and confirm a significantly higher number of adversarial phishing pages are detected by other factors than those detected by adversarial logos ($\chi^2 = 37.8$, $p < 0.001$).

Takeaway. Despite users recognizing adversarial phishing webpages slightly better than the original ones, it remains difficult for users to recognize adversarial phishing pages accurately (TPR=0.59). Also, most of the provided explanations are not related to our `LogoMorph` attack.

Brand	Std. AdvTrain		w/ Ref Augment	
	# Succ (Test)	Rate	# Succ (Test)	Rate
Instagram	40 (40)	1.00	1.00 (40)	0.03
Netflix	16 (16)	1.00	0 (16)	0.00
CIBC	25 (26)	0.96	21 (26)	0.81
eBay	35 (37)	0.95	5 (37)	0.14
AT&T	17 (18)	0.94	17 (18)	0.94
DHL	37 (41)	0.90	28 (41)	0.68
Google	22 (25)	0.88	1 (25)	0.04
Yahoo	7 (8)	0.88	4 (8)	0.50
Comcast	24 (29)	0.83	0 (29)	0.00
Spotify	33 (40)	0.83	0 (40)	0.00
DocuSign	29 (36)	0.81	6 (36)	0.17
Dropbox	29 (40)	0.73	1 (40)	0.03
LinkedIn	28 (40)	0.70	0 (40)	0.00
PayPal	28 (40)	0.70	0 (40)	0.00
Amazon	25 (40)	0.63	0 (40)	0.00
BOA	25 (40)	0.63	0 (40)	0.00
Outlook	25 (40)	0.63	0 (40)	0.00
Chase	18 (40)	0.45	1 (40)	0.03

Table 10: **Adversarial Retraining**—# of adversarial phishing webpages that bypass PhishIntention after adversarial retraining (standard and w/ reference augmentation). "Succ" stands for the number of successful logos; "Test" for the number of tested logos.

6 Countermeasures

Finally, we explore potential countermeasures to our proposed `LogoMorph` attack. We first apply the basic adversarial retraining method to enhance the model's ability to detect adversarial phishing webpages. After showing the ineffectiveness of this basic method, we propose a *new* method by combining adversarial retraining with the augmentation of the reference list (which is unique to reference-based phishing detectors). This new method shows much-improved performance. Finally, we experiment with *gradient masking*, which aims to make it difficult to compute adversarial logos.

Basic Adversarial Retraining. The idea of adversarial retraining [10, 35] is to retrain the detector using a portion of adversarial logos. We first perform standard adversarial retraining for PhishIntention by removing the weights of the last classification layer and training it with our successful adversarial logos. For this experiment, we constructed a dataset with all three types of logos. For image-only logos, we use the successful logos from Table 4. For text-only logos, we use the successful logos from the top 200 fonts from Table 3. For image-text logos, given we have thousands of successful ones, we randomly sample 200 for each brand. 80% of this dataset is used as the training set and the remaining 20% is used as the testing set. We train the model until it converges and we get a 98.50% accuracy on the top-1 logo brand matches. For this standard version, the adversarial logos are only used for re-training but are *not included in the reference list* for phishing detection. In other words, given a testing input logo, it is compared against the original logos.

The testing result (end-to-end) is reported in Table 10 (left). While adversarial retraining helps to reduce the bypass rate, the model is still insufficient to detect adversarial phishing pages (the bypass rate ranges from 0.45 to 1.0). Intriguingly, standard adversarial training was found to be quite effective against PhishGAP [29] (albeit it was not tested end-to-end). We also evaluate the false positive rate for standard adversarial retraining using a large benign webpage dataset from Phishpedia [30]. It has 25K benign pages, and we use 21,974 of them that have URLs to verify their brands and legitimacy. The test returns 21 false positives (0.096%). This is comparable to the 17 false positives of the original detector (0.077%).

Adv. Retraining + Reference Augmentation. Given the findings above, we introduce a new adversarial retraining method where we additionally inject the adversarial logos (from the training set) to the reference list. In other words, given a testing input, it is compared against both real logos and the added adversarial logos of a given brand to detect mimicry. The result is reported in Table 10 (right). We show that the augmented reference list has significantly improved the detection of adversarial logos. 11 out of 18 brands have a success rate lower than 10%. 14 out of 18 brands have a success rate lower than 50%. This approach only slightly increases the number of false positives (from 21 to 23, compared with standard adversarial retraining), with a false positive rate of 0.1%. The result suggests that fuzzing the reference list is more important to robustify the detector.

Gradient-Masking. Another defense method proposed by prior work [30, 32] is gradient masking. The goal is to modify the phishing detection model such that it becomes more difficult for optimization-based algorithms to generate adversarial examples. The detailed experiments are presented in supplementary materials [5]. We find that gradient masking is not effective in defending against `LogoMorph`. Gradient masking is designed to defend against classification-gradient-based adversarial examples (that introduce small noises), which is not well suited to defend against our attack.

Takeaway. Standard adversarial retraining and gradient-masking are ineffective against `LogoMorph`. We propose a new effective countermeasure that combines adversarial retraining with the augmentation of the reference list.

7 Discussion and Conclusion

In this paper, we show that the adversarial logos generated by `LogoMorph` is effective against a range of state-of-the-art phishing detectors [6, 30, 32]. The result confirms (and aggravates) the initial concern raised by [29]—compared with [29] (which only focuses on logos and logo discriminators), we further show the end-to-end effectiveness of the attack (webpage-level) against both the systems and users.

Security Considerations. We derive key lessons learned from our experiments on phishing detection systems (in §4). *First*, our *white-box evaluation* (in §4.2 to §4.4) assumes an attacker owns an exact copy of the targeted detector (i.e., PhishIntention [32]) to identify suitable adversarial logos. The webpage-level results show that, for most brands (e.g., 7 out of 11 for image-only logos, see Table 5; and 9 out of 14 for text-only logos, see Table 3), over 80% of the adversarial pages crafted via `LogoMorph` bypass PhishIntention end-to-end. *Second*, the *black-box setting*, in which the "surrogate" detector used by the attacker is different from the targeted one, makes it harder to find suitable logos. This is reflected by lower overall effectiveness in our experiments—e.g., only 6 out of 15 (or 4 out of 18) brands have more than 80% of our adversarial webpages bypass VisualPhishNet [6] (or PhishIntention [32]) end-to-end, as shown in Table 11 (Table 8). *Third*, the decrease in effectiveness underscores the importance of *keeping deep learning models* confidential: despite reliant on "security-by-obscurity", preventing unauthorized access[13] to such models is a pragmatic defense. *Finally*, in either case, however, the attacker can simply disregard brands that are "difficult to attack", and use `LogoMorph` *only for those brands that exhibit a near-perfect bypass rate* across all experiments. Indeed, our considered brands are highly popular, and real-world attackers would favor such brands in practice.

Cost of Attack. We discuss the *cost of the attack* (embracing the recommendations of [7]). The major contributors to such cost are: obtaining the surrogate detector[14], which can be done by, e.g., using the publicly available detectors from research papers; and the computational runtime to generate the logos with `LogoMorph`. For the latter, we provide the details from our experiments: generating an (adversarial) text-only logo requires on average 3.5s of CPU-time (1 thread); whereas tuning the diffusion model on one brand (of 200–500 images) requires 24–48h of training, after which only ∼0.5s are required to generate an (adversarial) image-only logo (on a single GPU). Our experiments are carried out on an 8-threaded CPU (Intel Xeon Silver 4214 2.20GHz) and NVIDIA RTX 5000 GPU: parallelization can further decrease the time required to craft our adversarial logos.

Why the Attack Works; How It is Different. The key insight is obtained from challenging the assumption that the logo is the invariant of a website's identity (and users would pay close attention to it). The intuition is that adversaries can introduce large perturbations to the logos, as long as the perturbations do not significantly change the semantics. Such perturbations can help to evade the phishing detectors while

[13] N.b.: we always assume an attacker cannot query the targeted detector (see §2.2). With query access, the "white-box" scenario could turn into a "black-box" setting wherein the attacker can query the targeted detector (potentially subject to budget constraints) to find suitable logos. Such a scenario can be explored by future work (we release our code [4]).

[14] In a "perfect knowledge" scenario, the attacker must also invest the resources to determine that the targeted detector matches the surrogate.

the adversarial phishing pages can remain effective on users. Unlike traditional gradient-based attacks [12] that compute imperceivable noises (which are easier to defend against [32]), our idea is to search for large (semantic-preserving) perturbations. For text logos, this is realized by searching for alternative fonts. For image logos, this is realized with a diffusion model to learn the logo styles. Both of these have room for further improvement. For instance, instead of searching among existing fonts, adversaries may use ML models to generate "novel" fonts [13, 45]. Moreover, our current image-logo attack fine-tunes an adversarial diffusion model for each brand. Future work may further optimize this by training a single model for all brands using *conditional* diffusion models [41].

Enhancing Automated Defense. Section 6 shows that adversarial training helps to improve the phishing detector, not necessarily because the deep learning model (for logo embedding) has been improved; instead, it is the reference augmentation that contributes more to the improved defense. In other words, for reference-based phishing detectors, the defender not only needs to include all the existing logos of a brand to the reference list but should also add *potential variants* of such logos. The diffusion model can be a generic approach to "fuzz" the reference list. This paper is focused on attacks, and thus the diffusion model is primarily tuned for "fidelity." Future work may tune the diffusion models to control and encourage "diversity" in the diffusion process to optimize the model for better defense (via reference fuzzing).

8 Acknowledgements

This work was supported in part by the National Science Foundation (NSF) under grants 2229876, 2055233, and 2030521, IBM-ILLINOIS Discovery Accelerator Institute (IIDAI), C3.ai, the European Commission under the Horizon Europe Programme, as part of the project LAZARUS (Grant Agreement no. 101070303), project SERICS (PE00000014) under the NRRP MUR program funded by the EU-NGEU, and the project Privacy Aware Anti Malware (PAAM) funded by PRIN 2022 PNRR. This research was also partially supported by Hilti.

References

[1] Google fonts. https://github.com/google/fonts, 2023.

[2] Google safe browsing. https://developers.google.com/safe-browsing/, 2023.

[3] Openphish. https://openphish.com/, 2023.

[4] Our repository. https://github.com/gyNancy/Visualphish_public, 2024.

[5] Supplementary materials. https://github.com/gyNancy/Visualphish_public/blob/main/Supplementary_Materials.pdf, 2024.

[6] Abdelnabi, S., Krombholz, K., and Fritz, M. Visualphishnet: Zero-day phishing website detection by visual similarity. In *Proc. of CCS* (2020).

[7] Apruzzese, G., Anderson, H. S., Dambra, S., Freeman, D., Pierazzi, F., and Roundy, K. "real attackers don't compute gradients": Bridging the gap between adversarial ml research and practice. In *Proc. of SaTML* (2023).

[8] Apruzzese, G., Conti, M., and Yuan, Y. Spacephish: The evasion-space of adversarial attacks against phishing website detectors using machine learning. In *Proc. of ACSAC* (2022).

[9] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., and Rieck, K. Dos and don'ts of machine learning in computer security. In *Proc. of USENIX Security* (2022).

[10] Bai, T., Luo, J., Zhao, J., Wen, B., and Wang, Q. Recent advances in adversarial training for adversarial robustness. In *Proc. of IJCAI* (2021).

[11] Bell, S., and Komisarczuk, P. An analysis of phishing blacklists: Google safe browsing, openphish, and phishtank. In *Proc. of ACSW* (2020).

[12] Carlini, N., and Wagner, D. A. Towards evaluating the robustness of neural networks. In *Proc. of IEEE SP* (2017).

[13] Cha, J., Chun, S., Lee, G., Lee, B., Kim, S., and Lee, H. Few-shot compositional font generation with dual memory. In *Proc. of ECCV* (2020).

[14] Chen, J., Huang, Y., Lv, T., Cui, L., Chen, Q., and Wei, F. Textdiffuser: Diffusion models as text painters. *arXiv preprint arXiv:2305.10855* (2023).

[15] Chiew, K. L., Tan, C. L., Wong, K., Yong, K. S., and Tiong, W. K. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences 484* (2019), 153–166.

[16] Croce, F., and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proc. of ICML* (2020).

[17] Divakaran, D. M., and Oest, A. Phishing detection leveraging machine learning and deep learning: A review. *IEEE Security & Privacy* (2022).

[18] Draganovic, A., Dambra, S., Iuit, J. A., Roundy, K., and Apruzzese, G. "do users fall for real adversarial phishing?" investigating the human response to evasive webpages. In *APWG 2023 eCrime Symposium* (2023).

[19] FBI. Internet crime report. https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf, 2022.

[20] Frauenstein, E. D., and Flowerday, S. Susceptibility to phishing on social network sites: A personality information processing model. *Computers & security* (2020).

[21] Fu, A. Y., Wenyin, L., and Deng, X. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd). *IEEE Transactions on Dependable and Secure Computing 3* (2006), 301–311.

[22] Hannousse, A., and Yahiouche, S. Towards benchmark datasets for machine learning based website phishing detection: An experimental study. *Engineering Applications of Artificial Intelligence 104* (2021), 104347.

[23] Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. AugMix: A simple data processing method to improve robustness and uncertainty.

[24] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Proc. of NeurIPS* (2020).

[25] Hu, H., Jan, S. T., Wang, Y., and Wang, G. Assessing browser-level defense against IDN-based phishing. In *Proc. of USENIX Security* (2021).

[26] IBM. Cost of a data breach. https://www.ibm.com/reports/data-breach, 2022.

[27] JI, J., ZHANG, G., WANG, Z., HOU, B., ZHANG, Z., PRICE, B., AND CHANG, S. Improving diffusion models for scene text editing with dual encoders. *arXiv preprint arXiv:2304.05568* (2023).

[28] LEE, J., TANG, F., YE, P., ABBASI, F., HAY, P., AND DIVAKARAN, D. M. D-fence: A flexible, efficient, and comprehensive phishing email detection system. In *Proc. of IEEE EuroS&P* (2021).

[29] LEE, J., XIN, Z., SEE, M., SABHARWAL, K., APRUZZESE, G., AND DIVAKARAN, D. M. Attacking logo-based phishing website detectors with adversarial perturbations. In *Proc. of ESORICS* (2023).

[30] LIN, Y., LIU, R., DIVAKARAN, D. M., NG, J. Y., CHAN, Q. Z., LU, Y., SI, Y., ZHANG, F., AND DONG, J. S. Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages. In *Proc. of USENIX Security* (2021).

[31] LIU, D., WANG, W., WANG, Y., AND TAN, Y. Phishledger: a decentralized phishing data sharing mechanism. In *Proc. of IECC* (2019).

[32] LIU, R., LIN, Y., YANG, X., NG, S. H., DIVAKARAN, D. M., AND DONG, J. S. Inferring phishing intention via webpage appearance and dynamics: A deep vision based approach. In *Proc. of USENIX Security* (2022).

[33] LIU, R., LIN, Y., ZHANG, Y., LEE, P. H., AND DONG, J. S. Knowledge expansion and counterfactual interaction for reference-based phishing detection. In *Proc. of USENIX Security* (2023).

[34] LUO, C. Understanding diffusion models: A unified perspective. In *arXiv preprint arXiv:2208.11970* (2022).

[35] MADRY, A., MAKELOV, A., SCHMIDT, L., TSIPRAS, D., AND VLADU, A. Towards deep learning models resistant to adversarial attacks. In *Proc. of ICLR* (2018).

[36] MOHAMMAD, R. M., THABTAH, F., AND MCCLUSKEY, L. Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications 25* (2014), 443–458.

[37] MONTARULI, B., DEMETRIO, L., PINTOR, M., COMPAGNA, L., BALZAROTTI, D., AND BIGGIO, B. Raze to the ground: Query-efficient adversarial html attacks on machine-learning phishing webpage detectors. In *Proc. of AISec* (2023).

[38] MORARU, A., AND DONAHUE, P. R. Top 50 most impersonated brands in phishing attacks and new tools you can use to protect your employees from them. https://blog.cloudflare.com/50-most-impersonated-brands-protect-phishing, 2023. Cloudflare.

[39] NAGARAJ, K., BHATTACHARJEE, B., SRIDHAR, A., AND GS, S. Detection of phishing websites using a novel twofold ensemble model. *Journal of Systems and Information Technology 20* (2018), 321–357.

[40] NICHOL, A. Q., AND DHARIWAL, P. Improved denoising diffusion probabilistic models. In *Proc. of ICML* (2021).

[41] NICHOL, A. Q., DHARIWAL, P., RAMESH, A., SHYAM, P., MISHKIN, P., MCGREW, B., SUTSKEVER, I., AND CHEN, M. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proc. of ICML* (2022).

[42] OEST, A., SAFAEI, Y., DOUPÉ, A., AHN, G.-J., WARDMAN, B., AND TYERS, K. Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists. In *Proc. of IEEE SP* (2019).

[43] OEST, A., SAFAEI, Y., ZHANG, P., WARDMAN, B., TYERS, K., SHOSHITAISHVILI, Y., AND DOUPÉ, A. {PhishTime}: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists. In *Proc. of USENIX Security* (2020).

[44] PALAN, S., AND SCHITTER, C. Prolific. ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance 17* (2018), 22–27.

[45] PARK, S., CHUN, S., CHA, J., LEE, B., AND SHIM, H. Multiple heads are better than one: Few-shot font generation with multiple localized experts. In *Proc. of ICCV* (2021).

[46] RAO, R. S., AND PAIS, A. R. Detection of phishing websites using an efficient feature-based machine learning framework. *Neural Computing and applications 31* (2019), 3851–3873.

[47] SAHINGOZ, O. K., BUBER, E., DEMIR, O., AND DIRI, B. Machine learning based phishing detection from urls. *Expert Systems with Applications 117* (2019), 345–357.

[48] SHENG, S., WARDMAN, B., WARNER, G., CRANOR, L., HONG, J., AND ZHANG, C. An empirical analysis of phishing blacklists.

[49] SHI, B., YANG, M., WANG, X., LYU, P., YAO, C., AND BAI, X. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern analysis and Machine Intelligence 41* (2018), 2035–2048.

[50] SHIRAZI, H., BEZAWADA, B., RAY, I., AND ANDERSON, C. Adversarial sampling attacks against phishing detection. In *Proc. of DBSec* (2019).

[51] SONG, F., LEI, Y., CHEN, S., FAN, L., AND LIU, Y. Advanced evasion attacks and mitigations on practical ml-based phishing website classifiers. *International Journal of Intelligent Systems 36*, 9 (2021), 5210–5240.

[52] TIAN, K., JAN, S. T., HU, H., YAO, D., AND WANG, G. Needle in a haystack: Tracking down elite phishing domains in the wild. In *Proc. of IMC* (2018).

[53] VERIZON. 2023 data breach investigations report. https://www.verizon.com/business/resources/reports/dbir/, 2023.

[54] WEI, W., KE, Q., NOWAK, J., KORYTKOWSKI, M., SCHERER, R., AND WOŹNIAK, M. Accurate and fast url phishing detector: a convolutional neural network approach. *Computer Networks 178* (2020), 107275.

[55] ZANGOOEI, A., DERHAMI, V., AND JAMSHIDI, F. A novel architecture for detecting phishing webpages using cost-based feature selection. *Journal of AI and Data Mining 7* (2019), 607–616.

[56] ZHU, Y., LI, Z., WANG, T., HE, M., AND YAO, C. Conditional text image generation with diffusion models. In *Proc. of CVPR* (2023).

Brand	# Bypass VisualPhishNet	Rate
AT&T	81 (81)	1.00
Instagram	199 (199)	1.00
LinkedIn	6229 (6,249)	1.00
Yahoo	35 (39)	0.90
eBay	156 (183)	0.85
CIBC	102 (121)	0.84
DHL	108 (194)	0.56
Amazon	10,401 (37,970)	0.27
Dropbox	6,787 (29,773)	0.23
Chase	4,157 (18,601)	0.22
BOA	1,593 (13,479)	0.12
Google	14 (121)	0.12
Outlook	107 (11,387)	0.01
PayPal	72 (6,383)	0.01
Netflix	0 (80)	0.00

Table 11: **Transferability to VisualPhishNet (All Logos)**—Number of adversarial phishing pages (generated with a local PhishIntention model) that successfully transfer to bypass another phishing detector VisualPhishNet. Three brands from Table 1 (Comcast, Spotify, DocuSign) are omitted because they are not included in [6].

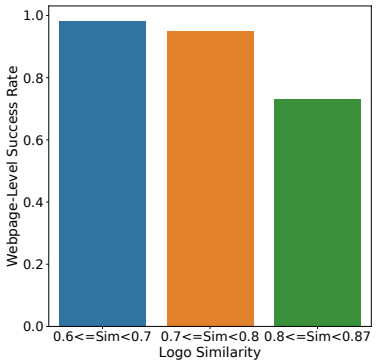

Figure 7: **Evasion Rate vs Logo Similarity**—We group the adversarial logos into 3 buckets based on their similarity, and report the aggregated webpage-level evasion rate against PhishIntention.

A Evasion Rate vs. Similarity Threshold

In this section, we further explore the trade-off between the similarity threshold and evasion success rate. For this analysis, we still focus on logos having Sim$\geq$0.6 (to make sure the logos are of reasonable visual quality). More specifically, we collect all the candidate image- and text-logos (from §4.2 and §4.3 in Table 3 and Table 5)[15], sort them based on their similarity, and then group them into three buckets: 0.6$\leq$Sim$<$0.7, 0.7$\leq$Sim$<$0.8, and 0.8$\leq$Sim$<$0.87. Then for each bucket, we add the adversarial logos to their corresponding webpages and test the webpages against PhishIntention end-to-end pipeline. The success rate of these adversarial logos is shown in Figure 7. Overall, the result confirms the expected trade-off: choosing logos with a lower similarity improves the evasion success rate. However, in the ancillary user study (Appendix B), we show that the cost of using lower-similarity logos is on users: from users' perspectives, such logos tend to also have a lower resemblance to the target brands. In practice, attackers should prioritize higher-quality logos (high similarity) as long as they can bypass the detector. However, given that attackers may not have perfect knowledge of the target system (and hence do not know the exact threshold), attackers may consider using logos that are slightly below the "anticipated" threshold (we use this intuition for choosing some of the logos of our blackbox experiment in §4.6).

B Ancillary User Study (Logo-Level)

In our main paper, we carried out a user study (§5) whose goal was assessing if users were able to identify phishing webpages that included an adversarial logo crafted via `LogoMorph`. Here, we carry out an orthogonal user study, whose goal is to determine the extent to which the logos crafted via `LogoMorph` can retain the semantics of the targeted brand. Specifically, we show various logos crafted via `LogoMorph` which achieve a certain similarity (according to PhishIntention), and ask users to rate how much such logos "resemble" the targeted brand.

Questionnaire Design. This user study follows a similar setup (in terms of platform and structure) to the one in our main paper (§5). However, here, we only display the logo in isolation—whereas in our main paper we showed the entire webpage. **(I)** We begin by informing users that the study involves "understanding the human's ability to identify website brands".[16] We then explain the participants' rights and the instructions of our survey. **(II)** Each participant is then shown a total of 52 adversarial logos. All logos used in the study have bypassed the detection threshold of PhishIntention (i.e., Sim<0.87). Specifically, we show 3 logos for each of the 18 brands listed in Table 1. These 3 logos pertain to three categories:

- *Very Similar*, i.e., adversarial logos crafted via `LogoMorph` with 0.8$\leq$Sim$<$0.87 according to PhishIntention;
- *Somewhat Similar*, i.e., adversarial logos crafted via `LogoMorph` having 0.7$\leq$Sim$<$0.8;
- *Barely Similar*, i.e., adversarial logos crafted via `LogoMorph` having 0.6$\leq$Sim$<$0.7[17]

We randomize the order of the logos for each participant to reduce bias. **(III)** Each logo (and corresponding questions) is shown in a dedicated section of our questionnaire. Upon reaching any new section, the user is first asked "Do you know the brand *b*", where *b* is the brand targeted by the logo, and the answers are "Yes, I know and visit it often" / "Yes, I know it but I do not use it often" / "No, I have never heard of it." Then, we show the logo, and ask "Does this logo resemble the brand?" and the answer is picked from a five-point Likert scale: 1=not resemble; 2=probably not resemble; 3=undecided; 4=probably resemble; 5=resemble. Also, we specifically write "Please do not look up the logo" to prevent users from checking the Internet and biasing their responses. **(IV)** At the end of the questionnaire, we ask an attention check question, showing the image of a social network and inquiring whether it represents a bank. Our study does not collect any personally identifiable information, and our participants' identities are anonymous and participation was voluntary and could withdraw their data at any time (during and after the study). On average, each participant spends $\approx$7 minutes.

Results. We collected 5,200 valid responses from 100 participants. To provide an accurate representation, we only consider the responses of users who "know" the brand (i.e., they answered "Yes" to the brand-knowledge question). These re-

[15] We did not combine the text and image parts at the logo level; as discussed in §4.4, they are directly combined at the webpage level.

[16] We note that, differently from what we did in the user study in our main paper, we are not priming the users here: we *never* mention the term "phishing" in our questionnaire. This is to ensure that the users are not biased to think that a given logo may be related to a malicious purpose, thereby increasing the quality of user responses (i.e., this study is *not* about phishing).

[17] We do not have any logo for PayPal and LinkedIn within this range, which is why we have 52 logos (given by: $3 \times 18 - 2$).

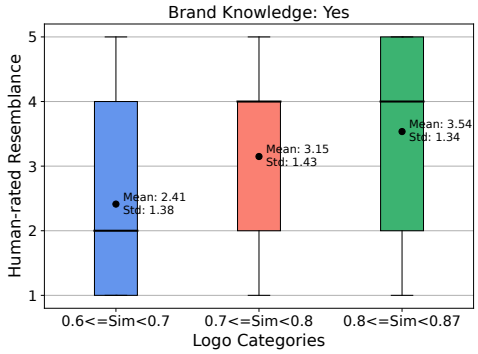

Figure 8: **Main Results of our Logo-level User Study**—Distribution of the human-rated resemblance (higher is better) across adversarial logos crafted with LogoMorph.

sponses count for about 90% of all valid responses. Figure 8 shows the aggregated "human-rated resemblance" (y-axis) across all logos within a given category (boxplots). We can see that the human-rated resemblance is highest for logos within the *very similar* category: the average rating is 3.54, and the standard deviation is 1.34, and a t-test confirms that such a rating is statistically significantly ($p < 0.001$) superior than the middle-point of 3. Hence, we can claim that *logos crafted with* LogoMorph *having 0.8≤Sim<0.87 tend to preserve the characteristics of the targeted brand* (at least according to our participants, and w.r.t. our considered 18 brands). This suggests that, from an attacker's viewpoint, it is wise to use LogoMorph by picking adversarial logos having 0.8≤Sim<0.87 to embed in the web pages. (Indeed, this is what we have done in the user study discussed in §5.) For the *somewhat similar* logos (0.7≤Sim<0.8), the average rating is 3.15 (i.e., also above the middle-point 3), indicating a positive resemblance of the target brand. Interestingly, however, for some users, even logos having a lower similarity can resemble (to some extent) the targeted brand, as shown by the wide whiskers in the blue (0.6≤Sim<0.7) boxplot.

C Extended Logo Experiments

Insofar, our evaluation revolved around a subset of 18 brands included in PhishIntention's dataset. Here, we expand the our assessment to cover more brands, demonstrating that LogoMorph is broadly applicable to a variety of brands and their logos. We regard the 18 brands used in §4 as the "*Main Set*", and then introduce an "*Expanded Set*" of 92 brands (110 brands in total). For the Expanded Set, we first include the top 50 most impersonated phishing attack brands [38] (excluding those that are not in PhishIntention's protected brand list). Then we fill out the rest of the slots based on PhishIntention's protected brand list. Among the 92 brands, 17 have image-only logos[18], and 75 have text-only logos.

[18]For brands like Capital One Bank, differentiating the text and image parts of the logo is challenging, and thus we regard the logo as image-only.

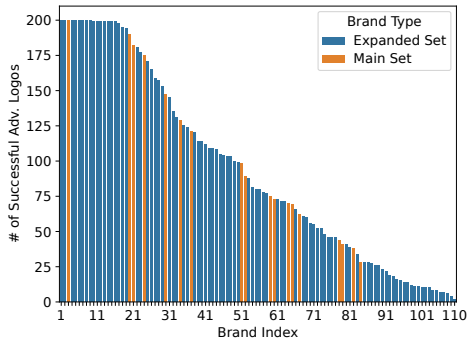

Figure 9: **Successful Adv. Logos Per Brand (110 Brands)**—We sorted the 110 brands on the x-axis based on the number of successful adversarial logos identified by LogoMorph (out of 200 candidate logos tested against PhishIntention).

Brand	# Success Logos	Rate
PayPal	165	0.83
Amazon	128	0.64
LinkedIn	128	0.64
BOA	93	0.47
DHL	81	0.41
ATT	79	0.40
Chase	78	0.39
Spotify	72	0.36
Dropbox	69	0.35
CICB	63	0.32
Outlook	30	0.15

Table 12: **Logo-Level Results for Image Logos**—Logos that bypass the *local surrogate* (out of 200 tested images per brand). We choose these logos to run a black-box attack against PhishIntention.

We replicate the white-box experiment in §4.2 and §4.3 to cover our 110 brands. For image logos, we use LogoMorph to generate 200 adversarial logos for each brand. For text logos, we follow the selection procedure (§3.1) by first generating 2,556 candidate fonts for each brand, and then selecting the top 200 fonts that bypass the similarity threshold to generate logos. The results are presented in Figure 9. The 110 brands are sorted based on the number of successful adversarial logos identified by LogoMorph against PhishIntention's logo discriminator (logo-level). The orange color represents the Man Set and the blue color represents the Expanded Set. The result shows that the 18 brands used in the main paper have sufficient diversity, covering both easy-to-attack and hard-to-attack brands. For 50 of these brands, LogoMorph identified more than 100 successful logos. For all 110 brands (100%), LogoMorph identified at least one viable attack logo for the attacker (from 200 candidate logos). In a white-box setting, the attacker would hence always find a suitable logo for each of these 110 brands.