

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Shielding QR Codes: Unveiling the Real-World Illicit Promotion Behind Adversarial QR Codes

Lijie Wu and Xiaoping Zhang, *Tsinghua University;* Mingxuan Liu, *Zhongguancun Laboratory;* Yue Qin, *Central University of Finance and Economics;* Baojun Liu, *Tsinghua University;* Geng Hong, *Fudan University;* Zhenrui Zhang and Chenghui Wu, *Baidu Inc;* Hui Jiang, *Tsinghua University and Baidu Inc*

https://www.usenix.org/conference/usenixsecurity26/presentation/wu-lijie

This paper is included in the Proceedings of the 35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the 35th USENIX Security Symposium is sponsored by



جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

ARTIFACT EVALUATED
USENIX
AVAILABLE

Shielding QR Codes: Unveiling the Real-World Illicit Promotion Behind Adversarial QR Codes

Lijie Wu[‡], Xiaoping Zhang[‡], Mingxuan Liu[§*✉], Yue Qin[†✉], Baojun Liu[‡✉], Geng Hong[φ], Zhenrui Zhang[ǂ], Chenghui Wu[¶], Hui Jiang[‡¶],

[‡]*Tsinghua University,* [§]*Zhongguancun Laboratory,* [†]*Central University of Finance and Economics,* [φ]*Fudan University,* [ǂ]*Baidu Inc*

Abstract

Adversarial QR Code Images (AQRIs) represent an emerging threat vector for covert (usually illicit) online promotion. They use adversarial perturbations to evade QR detectors (e.g., OCR-based models) while retaining decodability for information delivery, facilitating malicious content dissemination, and posing risks to both platforms and users. Though adversarial attacks are well-studied, targeted techniques against structured QR codes are underexplored.

To systematically investigate the real-world AQRI abuse, we cooperated with a leading Internet service provider. With the help of our partner, grounded in empirical observations, we introduce Adato, an enhanced framework for AQRI detection, by prioritizing finder pattern regions and identifying adversarial techniques through cross-platform consistency checks. Experimental results demonstrate that Adato achieves 98.6% precision and 97.8% recall on AQRI detection, significantly outperforming existing detectors. With the collaboration of our partner, we legally obtained posts with images from five well-known international social media platforms from September, 2024 to March, 2025, e.g., Reddit, Baidu Tieba. We applied Adato to over 40 million images and identified 68,467 AQRIs, demonstrating their widespread real-world use and their ability to evade existing moderation mechanisms. Analysis of our detected AQRIs reveals that AQRIs are widely used for illicit promotion: 95.78% are linked to 2,079 malicious URLs, spanning 7 business categories. Additionally, we analyzed the information dissemination strategies employed, such as redirect chains and indirect propagation paths that exploit cross-platform inconsistencies. These results highlight Adato's effectiveness in strengthening existing moderation and recognition pipelines against AQRI abuses.[1]

✉ Corresponding authors.

[1]The full version of this paper, including appendix, is available at https://github.com/S5deded/usenix-security26-qr-codes-full-paper/blob/main/full-paper.pdf.

1 Introduction

QR codes are a popular medium for fast and convenient information dissemination, capable of encoding various types of payload, e.g., text and URLs. Commonly, QR codes are used to direct users to external websites on social media platforms, making them an efficient cross-platform redirection mechanism. However, their encoding and redirection capabilities also make QR codes a dissemination vector for malicious content [63, 117], such as phishing links [21, 28, 65, 85, 124]. To protect users from malicious content via QR codes, major social platforms implement QR code moderation and posting restrictions, including TikTok [96], Baidu Tieba [10], and Sina Weibo [88]. Specifically, QR code moderation typically involves detecting a QR region in the image, decoding the embedded payload, and then inspecting the decoded content against security and policy rules. *Despite advancements in moderation technologies, abuse of QR codes to spread malicious content has evolved in parallel.*

Adversarial QR Code Image (AQRI). AQRIs represent a new class of threats: QR codes that are visually perturbed to evade platform moderation systems while remaining decodable to end users (see Figure 3). By disrupting image-layer detection, AQRIs can bypass platform-level moderation and be successfully published on social media platforms, thereby expanding the reach of illicit or policy-violating content. Unlike traditional adversarial examples that induce model misclassification through imperceptible noise, AQRIs apply perceptible yet controlled transformations that preserve decodability. This constraint makes AQRI generation more targeted and practically impactful. Recent advances in generative models further exacerbate this threat by enabling AQRIs to blend into complex visual backgrounds, making detection harder and evasion more effective [31, 91, 108, 113]. AQRIs thus exemplify the exploitation of adversarial AI techniques in real-world malicious content dissemination campaigns.

While most social platforms have QR code moderation, they have not recognized the novel AQRI threat. Specifically, testing 4 mainstream QR detection tools and 5 platforms'

moderators on 100 in-the-wild AQRIs showed poor detection rates, exposing their inability to handle this novel threat, since AI models relying on structural pixel features fail to capture obfuscated AQRI traits. It also makes conventional adversarial defense (e.g., feature alignment) ineffective [3, 14, 30, 105, 116]. *Thus, efficient, automated AQRI detection and impact assessment are urgently needed.*

To address the gap in understanding and mitigating such novel AQRI threats, we collaborated with a leading global internet service provider, Baidu[2]. Social media's broad public reach reflects the impact of AQRIs on ordinary internet users and poses severe challenges to platform moderation. To assist moderation systems in detecting AQRIs and analyzing their underlying techniques and payloads, we scope our measurement study to five high-impact, large-user-base international platforms. Specifically, our work focuses on: *How do AQRIs affect existing QR code detection and moderation capabilities? What is the current usage and abuse status of AQRI in real-world scenarios? Which adversarial techniques and strategies are employed in crafting AQRIs?*

Our Work: Detecting AQRI with Adato. Addressing the challenges posed by AQRIs is nontrivial due to their stealthy design, black-box nature, and the limited effectiveness of existing detection methods. Motivated by empirical analysis of real-world AQRIs collected from user complaints with our partner, we develop Adato (AQRI Detection and Analysis Tool), a unified framework for detecting and restoring AQRIs to enable robust decoding, including 4 steps. First, Adato collects posts from 5 popular social media platforms (Baidu Tieba, Baijiahao, Reddit, X, and Instagram) via legal APIs and scraping. Second, it improves QR code detection by localizing and verifying finder pattern regions, which AQRIs typically preserve for decodability. Third, Adato identifies adversarial techniques by exploiting inconsistencies across multiple QR scanners and moderation engines, and attributes each AQRI to specific manipulations techniques using statistical profiling of basic visual features. Finally, for visual restoration and decoding, Adato performs technique-aware feature-level restoration to correct perturbations and improve decoding success. Evaluated on a real-world dataset, Adato achieves 98.6% precision and 97.8% recall on AQRI detection, and reaches a 99.87% decoding success rate. Moreover, Adato has a mere 46ms processing latency for QR code detection. It meets platform requirements for large-scale moderation, improving the performance of existing detectors by an average of 5.48x while enhancing end-user decoding success rates.

Analyzing AQRI in the wild. To answer research questions for AQRIs, we legally crawled 40,147,738 images from 5 social media platforms over 181 days (2024.09.13-2025.03.13), with the help of our partner. Finally, Adato identified 1,585,706 QR code images (3.94%), among which 68,467 were verified as AQRIs (4.32%). It reveals that AQRI has been widely adopted across social platforms for promotion. Furthermore, we conducted an in-depth analysis of AQRI's manipulation techniques, embedded information, abuse patterns, and impacts. Specifically, Adato identified 13 adversarial techniques in real-world AQRIs that, unlike traditional attacks requiring complex optimization [67, 68, 84, 109], rely on simple image processing such as structural distortion (e.g., geometric deformation, downscaling) and visual concealment (e.g., transparency, black masks). Though simple, their combination and fusion enable highly effective, low-cost evasion. Moreover, our analysis reveals the emerging use of generative methods, predominantly Stable Diffusion [82], in fabricating AQRIs. These tools effectively blend multiple adversarial effects while preserving decodability, posing an escalating challenge to current moderation systems. Compared to non-AI counterparts, AI-generated AQRIs exhibit significantly higher bypass rates (by 20.0%) and decoding success (by 18.0%) on average. The absence of usage restrictions and the accessibility of user-friendly interfaces further increase their abuse potential. In contrast, advanced Large Vision Models (LVMs) display functional blindness: although they achieve moderate bypass rates (73.5%), their end-to-end generation fails to preserve QR syntax, resulting in low decodability (13.3%) and rendering them currently impractical for adversarial use.

We further investigated the decoded information to confirm the actual content being propagated by AQRIs. Our analysis revealed that AQRIs were widely abused for various underground activities. Website links (99.80%) are the most common form of content dissemination. AQRI-related websites adopt diverse construction forms, from personally built sites (46.54%), abuse of short-link services (23.46%), to compromised authoritative websites (11.00%). They cover 7 malicious types, most of which are online pornography. Additionally, we found that adversaries mask malicious content with legitimate images and mimic normal posting behaviors. Furthermore, we observed a unique cross-platform strategy: leveraging varying moderation capabilities across platforms to propagate AQRIs with differing adversarial intensities.

Contributions. This paper makes the following contributions:

- *Enhanced AQRI Detection and Restoration:* We propose Adato, the first dedicated tool for detecting and restoring Adversarial QR Code Images. Leveraging finder-pattern-guided geometric validation, Adato achieves robust AQRI detection with a 97.8% success rate. Adato's restoration capability makes it a reliable pre-adaptor, boosting detection by 5.48× across five existing QR detectors, including our partner's commercial QR processing system deployed in real-world settings. To support broader adoption, we will release Adato as an open-source tool for developers and practitioners.
- *Novel Insights of AQRI Impact:* Based on Adato, we conducted the first large-scale, systematic measurement study of real-world AQRI abuse and constructed the first annotated AQRI dataset. Our analysis provides new insights into AQRIs, revealing prevalent evasion strategies, common manipulation

[2]Our collaborator provides multiple internet services, including famous social media platforms.

techniques, and their tangible impacts on mainstream QR processing platforms. We plan to share the dataset with researchers upon request to catalyze future research in AQRI analysis and defense.

2 Background

In this section, we first introduce QR codes and how they are disseminated and moderated on online platforms, then define adversarial QR code images (AQRIs), present our threat model, and clarify the scope of our study.

2.1 QR Code: Concept and Properties

QR codes are matrix barcodes with information-carrying pixel distributions, available in versions 1-40 [98]. Higher versions use more modules for larger capacity, increasing graphical complexity. Despite varying content-dependent pixel layouts, standardized design specifications require two key components: finder pattern and encoding region [98]. A QR code typically includes three finder patterns (top-left, top-right, and bottom-left), which determine its boundary and orientation. The encoding region, enclosed by these finder patterns, encodes information through a prescribed arrangement of pixels.

A functional QR code must satisfy two properties: *detectability* and *decodability*, which correspond to the two stages of QR processing. **Detectability** means a QR tool can reliably determine *whether a QR code is present and localize its region* (e.g., estimate its bounding quadrilateral and orientation). **Decodability** means the tool can correctly *recover the embedded payload*. Both properties rely on preserving key structural constraints: the finder patterns must remain sufficiently intact for localization. While detectability and decodability are fundamental properties of QR codes, their realization in practice depends on the specific detection and decoding systems in use. In standard QR processing, detection and decoding are *sequential* operations: a QR code must first be *detected* before its payload can be *decoded*. After successful detection and decoding, a QR code can reveal payloads such as URLs, plain text, or base64-encoded data, making it a widely used medium for promotion and information dissemination.

2.2 QR Code Dissemination and Moderation

QR codes are widely shared on many platforms, with social media being a primary vector (e.g., TikTok [96], Instagram [45], and Sina Weibo [88]). To mitigate malicious dissemination, external redirection, and unauthorized advertising, many platforms impose explicit or implicit restrictions on QR-code usage. By analyzing the Terms of Use (ToU) of eight major platforms, we identify three regulatory stances: (1) Baidu Tieba, Sina Weibo, Zhihu, and Xiaohongshu explicitly prohibit QR code postings to prevent external redirection [11, 89, 112, 123], making the mere presence of a QR code a policy violation, regardless of its payload; (2) TikTok bans QR codes that embed malicious or prohibited content (e.g., malware, pornography, gambling, phishing, or unauthorized commercial links) [97], and external reports suggest that videos with QR codes may be excluded from recommendations [69]; (3) Instagram, X (formerly Twitter), and Reddit do not state explicit policies, but our measurements indicate implicit restrictions. For instance, Instagram limits scanning to proprietary in-app codes [66], and some subreddits ban external links, including short URLs and QR codes [81]. Beyond platform-specific policies, QR detection is also supported by general-purpose moderation services such as Google Cloud Vision AI [36] and Baidu AI Cloud [8].

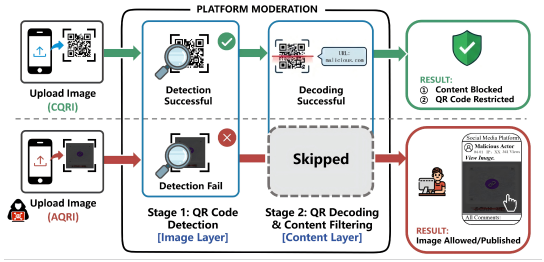


Figure 1: Overview of the Platform Moderation Pipeline and AQRI Evasion Mechanism.

Moderation Pipeline. As shown in Figure 1, the moderation process typically involves two steps.

• *Step 1: QR Code Detection (Image Layer)*: Platforms first detect the presence and location of QR codes within images using object detection techniques. Early rule-based methods (e.g., fixed pixel distributions, square edges, ISO patterns) [12, 24, 86, 90] have largely been replaced by deep learning models such as R-CNN [32, 33] and Transformers [103], which offer higher accuracy. Detected regions may undergo image enhancement (e.g., binarization, geometric correction, grid alignment) to support reliable decoding [18, 40, 56].

• *Step 2: QR Decoding & Content Filtering (Content Layer)*: Once the QR region is localized and preprocessed, the platform extracts the embedded payload using a QR decoder. This content typically includes URLs or plain text. Common techniques include blacklist matching and machine learning classifiers to detect malicious links, phishing attempts, or unauthorized promotions [50, 53, 83].

Notably, platforms implement these moderation steps differently according to their policies. Platforms that ban all QR codes may rely only on Step 1 (detection), while those targeting malicious payloads use both Step 1 and Step 2 (decoding and content filtering). Despite these variations, the underlying moderation pipeline follows the same sequential flow.

2.3 AQRI: Definition and Threat Model

Adversarial QR Code Images (AQRIs) are crafted to evade platform QR-code moderation for covert or broader dissemination reach, while remaining decodable by end users to ensure successful information delivery. Unlike traditional adversarial examples that use imperceptible perturbations to induce misclassification [34, 54, 72, 73, 92], AQRIs apply low-complexity, coarse-grained transformations (e.g., masking, distortion, occlusion) that exploit weaknesses in automated detection systems, while remaining decodable by end users regardless of human perception and visual usability.

Threat Model. We formalize the threat model of AQRIs to precisely characterize their adversarial objective. Let $x \in \mathcal{X}$ denote a conventional QR code image (CQRI), encoding a message $m \in \mathcal{M}$. Let $\mathsf{Detect}_S : \mathcal{X} \to \{0,1\}$ be the detection function employed by a QR processing system S (e.g., a platform or a QR processing tool), where $\mathsf{Detect}_S(x) = 1$ indicates that a QR code is successfully localized in the image. Once detected, the system applies a decoding function $\mathsf{Decode}_S : \mathcal{X} \to \mathcal{M} \cup \{\perp\}$ to extract the embedded message, with $\perp$ indicating decoding failure. Thus, we expect $\mathsf{Detect}_S(x) = 1$ and $\mathsf{Decode}_S(x) = m$ for a valid CQRI. Let $\delta : \mathcal{X} \to \mathcal{X}$ be a transformation function used by the adversary to generate an AQRI $x^{\text{adv}} = \delta(x)$, which is then published on platform P and satisfies the following conditions:

$$\mathsf{Detect}_P(x^{\text{adv}}) = 0,\ \mathsf{Detect}_Q(x^{\text{adv}}) = 1,\ \mathsf{Decode}_Q(x^{\text{adv}}) = m \quad (1)$$

for some tool $Q \in \mathcal{D}$, where $\mathcal{D}$ denotes the set of potential QR processing systems, including external or end-user applications. This setting reflects the **transmission model**, where the AQRI is *undetectable* on the publishing platform P, thereby skipping the decoding stage, yet remains *detectable* and *decodable* by external tools—enabling it to bypass moderation while still delivering its payload (usually illicit) to end users. This transmission model stems from differences in the capabilities of QR code processing tools.

• *Adversarial roles and objectives.* In this model, the adversary is the party who generates and publishes AQRIs, typically pursuing two objectives: 1) bypassing image-layer moderation (Step 1) to publish the QR code and expand its dissemination scope; and 2) ensuring the embedded message remains decodable by end users, thereby completing the information delivery process despite evading platform detection. The victims include both the platform, whose moderation pipeline is circumvented, and the users, who may unknowingly receive unreviewed or policy-violating content.

This threat model reflects real-world abuse scenarios in which adversaries exploit inconsistencies across platform-specific detection and decoding systems. Our preliminary measurement study (see Section 3) highlights this gap, showing how a QR code can evade moderation on the host platform but remains functional for end-user—thereby enabling covert or policy-violating dissemination.

Research Scope. This study investigates AQRIs as a distinct threat vector that targets the image-layer detection stage of platform moderation pipelines. While prior work has largely focused on bypassing content-layer checks through payload obfuscation (e.g., concealing malicious URLs), our work highlights a more fundamental evasion strategy: perturbing the visual structure of QR codes to evade initial detection altogether. Rather than introducing new attacks, our study analyzes AQRIs already used in the wild. Specifically, through formal collaboration with a major global network service provider, we curated a high-quality ground-truth dataset of user-reported, manually verified AQRIs and analyzed their impact on moderation pipelines (Section 3). We then characterize the adversarial properties of these AQRIs by comparing them to CQRIs, using a set of interpretable visual metrics to quantify key structural and perceptual differences and support threat landscape analysis (Section 4).

3 Preliminary Study of AQRI

To empirically validate the threat model of AQRIs, we conduct a comprehensive measurement study using a real-world dataset. As AQRIs are inherently difficult to collect from public sources due to their ability to evade standard QR detection, we collaborated with a major global network service provider to curate user-reported problematic QR codes via their complaint-handling infrastructure. We then examine the impact of AQRIs on three key components of QR code moderation pipelines: detection capability, decoding reliability, and overall moderation effectiveness.

AQRI Ground-truth Dataset Construction. With the assistance of our partner, we collected 5,000 QR code-related images from real-world user complaints spanning August to September 2024. To ensure data quality, these samples were initially filtered via the platform's predefined spam and malicious QR tags before undergoing our manual review. To ensure image quality, we first filtered out images with widths or heights below 300 pixels. Further, two expert researchers independently verified the adversarial characteristics: images exhibiting signs of manipulation through image processing techniques and a visibly disrupted or altered appearance were labeled as AQRIs. Any disagreements were resolved through arbitration by a third expert. As a result, we construct a ground-truth dataset of 3,000 real-world AQRIs.

To evaluate their effectiveness in bypassing moderation pipelines, we test detection performance across 4 commercial QR detectors, examine decodability using 10 widely-used decoding tools, and assess moderation responses by actively posting AQRIs to 5 major social media platforms. Due to platform usage limitations, we randomly selected 100 AQRIs from our dataset and paired them with 100 CQRIs sampled from a public dataset [20] for comparative analysis.

AQRI Impact on Detection Capability. We assess the adversarial impact of AQRIs on detection capability using 4 mainstream QR code detection services (Table 1). Specifically, we submitted each image to the corresponding service

Table 1: Detecting Success Rate of CQRI and AQRI on public QR code detection service.

Tool	Detecting Success Rate (%)	
	CQRI	AQRI
Google Vision AI [36]	95.0	11.0
Baidu AI Cloud [8]	93.0	10.0
Tencent Cloud [95]	96.0	13.0
Aliyun [4]	96.0	10.0

Table 2: Decoding success rates of various decoders on AQRI.

App-Decoder	Rate (%)	Library	Rate (%)
WeChat [93]	86.0	Qreader [26]	56.0
Alipay [5]	75.0	OpenCV-WeChat [71]	55.0
QQ [94]	48.0	Zbar [48]	41.0
iOS Scanner	67.0	Zxing [118]	38.0
Google Chrome [35]	38.0	BoofCV [76]	33.0

interface and recorded the outputs, which varied in format across systems: for instance, Baidu AI Cloud returns a confidence score, whereas Google Vision AI provides tiered classification labels (e.g., UNKNOWN to VERY LIKELY). Following the criteria used in prior work [38], we treat a QR code as successfully detected if the confidence score exceeds 0.5 or the classification label is "POSSIBLE" or "VERY LIKELY".

Table 1 shows that QR code detectors perform well on CQRIs but struggle significantly with AQRIs. All services detect no more than 13% of AQRIs, while achieving over 93% on CQRIs. These results indicate that current detection systems are ill-equipped to handle AQRIs and fall short of the reliability for real-world QR code detection.

AQRI's Decodeablity. We conducted experiments using 10 widely adopted QR code processing tools, including 5 end-user scanning applications and 5 open-source QR decoding libraries (Table 2). While end-user applications support both detection and decoding, detection is typically user-assisted, requiring manual alignment of the QR code within a predefined scanning frame. An AQRI is defined as decodable by a tool if its content is retrieved within 3 seconds. We thus evaluate *sample-level success* (decodable by at least one tool) and *tool-level success* (the percentage of sample-level successful AQRIs that a specific tool can decode, as reported in Table 2).

As shown in Table 2, although AQRIs are designed to evade automated detection, they remain decodable in practice. Every AQRI in our sample was successfully decoded by at least one tool. WeChat, Alipay, and the iOS Scanner achieved the highest success rates and accounted for most decoding successes, reflecting their widespread use and robust performance. Nonetheless, manual inspection revealed that these tools struggle with specific AQRI variants, such as those that combine overexposure with density compression, highlighting their limited resilience to diverse adversarial transformations. A detailed analysis of these transformations and their effects on detection and decoding is provided in Section 5.2.

Table 3: Posting Success Rate of CQRI and AQRI on different social platforms.

Platform	Posting Success Rate (%)	
	CQRI	AQRI
TikTok	0.0	82.0
Weibo	0.0	71.0
Xiaohongshu	0.0	75.0
Baidu Tieba	0.0	84.0
Zhihu	0.0	91.0

AQRI Impact on Platform Moderation. We further conduct active experiments by posting both AQRIs and CQRIs to social media platforms to evaluate their direct impact on platform moderation effectiveness. We targeted five platforms known to moderate QR code content (Table 3) and recorded the success rate of each post. If the QR code is detected, the post will not be successfully published. In our active experiments, the QR code samples were constructed by mimicking real AQRIs only at the image level (i.e., visual appearance). The decoded payloads were entirely benign. Specifically, each QR code contained only plain text stating the purpose of the experiment and providing our contact information, with no malicious URLs, spam content, or harmful payloads. To prevent user confusion and minimize moderation burden, all posts were removed within one minute, and the experiments were conducted under the supervision of our collaborator's legal team (Ethical Considerations).

Table 3 summarizes the posting success rates across platforms. All CQRIs were successfully blocked on every platform, resulting in a 0% posting success rate, which is consistent with stated platform policies on QR code restrictions (see Section 2.2). In contrast, AQRIs achieved a posting success rate of 80.3%, revealing a substantial gap in current moderation overall capabilities against adversarial QR codes.

Conclusion. Our experiments support the AQRI threat model by revealing the asymmetry between platform-side detection and end-user decoding. AQRIs can evade image-level detection while remaining reliably decodable by common user tools, enabling the covert dissemination of policy-violating content that circumvents platform safeguards yet remains accessible to recipients. In addition, our analysis of undetected AQRIs yields two key observations that may inform the development of more robust QR code detection mechanisms.

Key observations on AQRIs. First, to preserve their core function as information carriers ($\mathsf{Decode}_Q(x^{\mathrm{adv}}) = m$), AQRIs consistently retain essential structural features of QR codes, particularly the finder patterns, despite adversarial modifications. Second, their primary distinction from conventional QR images lies in their ability to evade detection; detection failure itself ($\mathsf{Detect}_P(x^{\mathrm{adv}}) = 0$) may thus serve as a useful signal of adversarial manipulation.

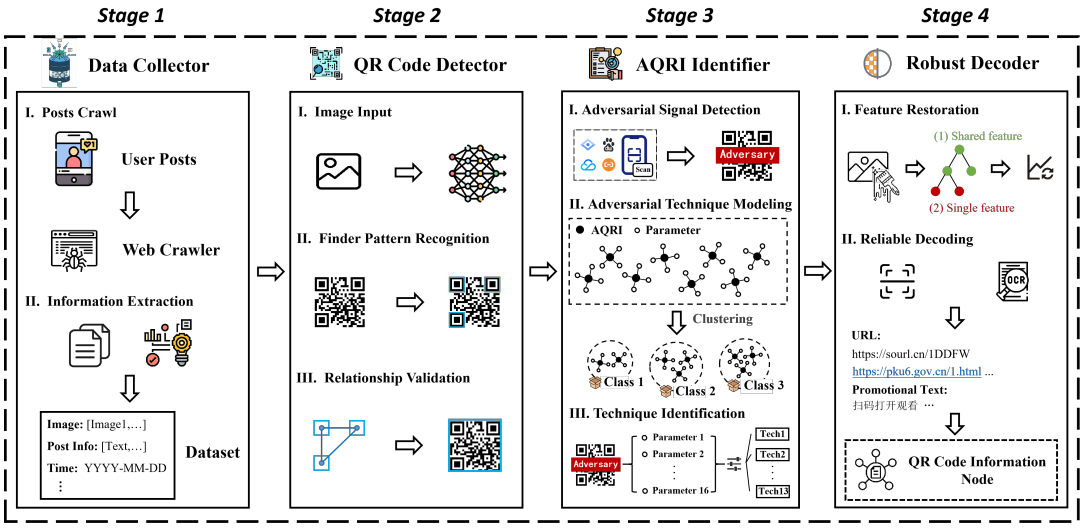


Figure 2: Workflow of Adato.

4 Detect AQRI In The Wild

To address the growing AQRI threat, we design and develop **Adato** (AQRI Detection and Analysis Tool (Figure 2), a unified framework for enhanced adversarial signal detection, and visual restoration to enable robust decoding. Motivated by two empirical insights from real-world AQRIs (Section 3), we propose a four-stage pipeline for robust AQRI detection, characterization, and decoding: (1) collect image-based posts from popular social media platforms via APIs and scraping to build an analysis database; (2) localize QR codes using resilient geometric cues, especially corner finder patterns; (3) flag adversarial cases by cross-platform scan inconsistencies and identify the underlying manipulation techniques; and (4) apply technique-specific restoration guided by normal/adversarial feature ranges to recover CQRI-like structure, then decode with standard tools for further analysis. Importantly, Adato can function as a standalone tool or as a preprocessing module that augments existing QR scanning and content moderation systems, thereby improving their capabilities against adversarial manipulation. Below, we elaborate on each step in detail.

4.1 Data Collection

To maximize coverage, we targeted high-traffic, image-focused communities and hashtags across 5 popular social media platforms. For Baidu Tieba and Reddit, we selected the top 10 communities by traffic plus image-centric groups, totaling 35 communities (maximum traffic 17.32M visits). For Baijiahao, Instagram, and Twitter[3], we combined 15 QR-related keywords from the Ground-truth dataset (Section 3) with the top 50 trending platform-specific hashtags [2, 100], yielding 65 keywords per platform. We collected posts on Reddit, Instagram, and Twitter via official APIs [46,79,110] using hashtag search and community crawling. As Baidu Tieba and Baijiahao provide no official API, we built a custom scraper with rate limiting (e.g., sleep intervals) for low-volume, long-horizon collection. From each post, we extracted image URLs and associated text to form the detection dataset. Regarding data analysis and storage, we used only publicly available data anonymized for technical analysis with no PII, and all procedures were supervised by our collaborator's legal department (see Ethical Considerations).

As shown in Table 4, we constructed a large-scale dataset by crawling image-containing posts from five widely used social media platforms: Baidu Tieba, Baidu Baijiahao, Reddit, Twitter, and Instagram. The data collection process spanned 181 consecutive days, from September 13, 2024 to March 13, 2025. During this period, we harvested a total of 24,311,446 posts, from which we extracted 40,147,738 image data points. This extensive dataset captures a broad and diverse sample of user-generated visual content across multiple platforms and regions, enabling a thorough examination of AQRI prevalence, characteristics, and evasion behaviors in the wild.

4.2 QR Code Detector: Toward Robust AQRI Detection

After QR image collection, Adato determines whether an image contains a QR code and localizes its region. As shown in Figure 2, QR decoding relies on the integrity of the three corner finder patterns, which QR standards require to remain largely intact [1]. Accordingly, adversaries typically preserve these regions to keep AQRIs decodable (Section 3). Adato exploits this invariance by locating the finder patterns and

[3]Twitter is also known as X

Table 4: Distribution of AQRI Detections across Platforms

Platform	# Post	# Image	# QR code	# AQRI	# Post with AQRI
Baijiahao [9]	10,872,307	16,308,460	704,511	30,566	28,563
Baidu Tieba [10]	7,083,573	11,758,731	432,023	19,388	18,730
Reddit [80]	2,618,265	4,922,338	174,054	8,342	7,056
Instagram [45]	2,117,391	4,563,912	138,191	5,713	5,600
Twitter [101]	1,619,910	2,594,297	136,927	4,458	4,334
All	24,311,446	40,147,738	1,585,706	68,467	64,283

validating their geometric and spatial configuration against the standard layout. We elaborate on the two steps below.

1) Finder Pattern Region Detection: To detect finder pattern regions, we use a YOLOv8-based detector [102] customized for QR structures: we replace the standard IoU loss with Wise-IoU [99], Wise-IoU = $(1-\text{IoU})\cdot w$, where the dynamic weight w emphasizes small-object localization to improve coordinate precision on compact, high-contrast finder patterns; we also deepen the backbone with additional convolutional layers for finer texture features and add spatial attention to suppress adversarial noise and background clutter. The spatial attention module applies channel-wise average and max pooling, concatenates the pooled maps, passes them through a convolution layer (e.g., a 7×7 kernel), and applies a sigmoid to produce a 2D attention map. The attention map is then multiplied element-wise with the (input or intermediate) feature maps to reweight spatial locations, emphasizing finder pattern regions while suppressing irrelevant background clutter and adversarial artifacts. We train the detector for 160 epochs on our annotated dataset and observe stable convergence; at inference time, given a single image, the model outputs a structured text file specifying the detected finder pattern regions as rectangular bounding boxes.

2) Positional and Spatial Relationship Validation: After detecting finder pattern regions, the system applies a rule-based geometric verification procedure to determine whether the detections form a valid QR code layout. Following QR standards, it requires (i) at least three detected finder patterns, (ii) three non-overlapping regions of approximately equal size, and (iii) centroids of any such triple that form an approximately right-angled triangle consistent with the canonical finder-pattern geometry. If these conditions hold, the image is classified as containing a QR code, and the system estimates the full QR bounding box by geometrically aligning the verified finder patterns, using the top-left and bottom-right coordinates as location indicators. The localized QR region then supports downstream adversarial technique analysis and precise cropping for decoding, ensuring that only structurally valid QR codes advance to subsequent restoration stages.

4.3 AQRI Identifier: Technique Discovery and Identification

Adversarial Signal Detection. To determine whether a QR image has been adversarially manipulated, we propose an adversarial-signal detection method that exploits discrepancies across multiple QR detection tools. The method is motivated by a key observation: unlike CQRIs, AQRIs may evade detection, and such failures can signal adversarial behavior. We instantiate this method using a real-world detector suite comprising four commercial QR detection services (Google Cloud Vision AI, Baidu AI Cloud, Tencent Cloud, and Aliyun). We selected these detectors based on their prevalence and empirical performance (Section 3), prioritizing those more susceptible to AQRIs to maximize coverage of realistic evasion cases. Any sample that bypasses at least one detector is treated as an adversarial candidate.

Adversarial Technique Modeling. We present a data-driven methodology to analyze and identify adversarial techniques in AQRIs based on their visual and structural distortions. In this work, an adversarial technique denotes an effect-centric distortion pattern identifiable from the final image, which we infer by clustering observed AQRI samples according to their distortion characteristics. Specifically, we construct a 16-dimensional feature space $\mathcal{V}=\mathbb{R}^{16}$ capturing visual and structural attributes commonly perturbed in AQRIs. Appendix B details the features, including metrics capturing aspects of contrast, color distribution, edge clarity, finder-pattern visibility, and image region occupancy. We map each QR code image $x\in\mathcal{X}$ to a feature vector $v(x)\in\mathcal{V}$ via a feature extraction function $v:\mathcal{X}\to\mathcal{V}$. Using these representations, we apply unsupervised clustering to a dataset of real-world AQRIs (Section 4.5), obtaining 23 clusters that group images with similar distortion patterns. We then manually inspect each cluster's centroid and representative samples to identify recurring manipulation patterns, which yields 13 distinct adversarial techniques (Section 5.2). For clarity, we name each technique after the closest interpretable image operation that captures the corresponding pattern.

Feature-based Adversarial Technique Identification. Unsupervised clustering assigns each image to a single cluster, which prevents comprehensive technique attribution when an AQRI exhibits multiple adversarial techniques. To address this limitation, we propose a feature-based identification method that defines technique-specific adversarial ranges over the feature dimensions and enables multi-label attribution by comparing an image's features against these ranges.

• *Modeling Adversarial Feature Ranges:* We model each feature's statistical distribution to estimate both its normal range (from CQRIs) and its technique-specific adversarial ranges. For each feature $i\in[1,16]$, we define the universe of valid values as $\mathcal{U}_i$. From a collection of CQRIs, we model the empirical distribution of feature i's values as p_i^{norm}. To derive the *normal value range* $\mathcal{R}_i^{\text{norm}}\subseteq\mathcal{U}_i$, we define it as the interval containing

$1-\alpha$ of the cumulative probability mass of the distribution $\mathcal{R}_i^{\text{norm}} = \left[\mu_i^{\text{norm}} - z_{\alpha/2}\sigma_i^{\text{norm}},\ \mu_i^{\text{norm}} + z_{\alpha/2}\sigma_i^{\text{norm}}\right]$, where μ_i^{norm} and σ_i^{norm} denote the mean and standard deviation of feature i over CQRIs, and $z_{\alpha/2}$ is the standard normal quantile for confidence level $1-\alpha$. This technique corresponds to the standard confidence interval construction for unimodal distributions and is widely adopted in statistical literature [15].

Similarly, for each adversarial technique $\mathcal{T}_j$, we model the distribution of feature i across QR codes identified to exhibit this technique, denoted $p_{i,j}^{\text{adv}}$, and derive its adversarial value range using the same high-confidence interval estimation: $\mathcal{R}_{i,j}^{\text{adv}} = \left[\mu_{i,j}^{\text{adv}} - z_{\alpha/2}\sigma_{i,j}^{\text{adv}},\ \mu_{i,j}^{\text{adv}} + z_{\alpha/2}\sigma_{i,j}^{\text{adv}}\right]$. The *anti-adversarial range* excludes values statistically associated with adversarial perturbations. We define it as the set complement of the adversarial range over the universe $\mathcal{R}_{i,j}^{\text{anti}} = \mathcal{U}_i \setminus \mathcal{R}_{i,j}^{\text{adv}}$.

To determine whether feature i is significantly affected by adversarial technique $\mathcal{T}_j$, we compute their correlation $C_{i,\mathcal{T}_j}$ as *Bhattacharyya distance* between the normal and adversarial feature distirbutions $C_{i,\mathcal{T}_j} = \text{BD}(p_i^{\text{norm}}, p_{i,j}^{\text{adv}}) = \frac{1}{4}\ln\left(\frac{1}{4}\left(\frac{\sigma_i^{\text{norm}}}{\sigma_{i,j}^{\text{adv}}} + \frac{\sigma_{i,j}^{\text{adv}}}{\sigma_i^{\text{norm}}} + 2\right)\right) + \frac{1}{4}\left(\frac{(\mu_i^{\text{norm}} - \mu_{i,j}^{\text{adv}})^2}{\sigma_i^{\text{norm}} + \sigma_{i,j}^{\text{adv}}}\right)$, assuming both distributions are approximately Gaussian. The Bhattacharyya distance is widely used for quantifying the separability of two probability distributions [49]. A large Bhattacharyya distance (e.g., exceeding a threshold τ) indicates that feature i is strongly influenced by $\mathcal{T}_j$.

• *Adversarial Technique Attribution:* Given the modeled ranges, we define binary identification functions for each technique $\mathcal{T}_j$. For an input image x with feature vector $\mathbf{v}(x)$, we say that feature i indicates technique $\mathcal{T}_j$ if $M_{i,j}(\mathbf{v}(x)) = \mathbb{I}\left[v_i(x) \in \mathcal{R}_{i,j}^{\text{adv}}\right]$, where $\mathbb{I}[\cdot]$ is the indicator function. Then, x is identified as exhibiting technique $\mathcal{T}_j$ if any of the features correlated with $\mathcal{T}_j$ falls into its adversarial range $\mathcal{T}_j(x) = \bigvee_{C(i,\mathcal{T}_j)>\tau} M_{i,j}(\mathbf{v}(x))$. To obtain the complete set of adversarial techniques associated with an image x, we evaluate each technique-specific identification function $\mathcal{T}_j(x)$ across all defined techniques $\mathcal{T}_1, \ldots, \mathcal{T}_K$. Specifically, we define $\mathcal{T}(x) = \left\{\mathcal{T}_j \,\middle|\, \mathcal{T}_j(x) = 1,\ j = 1, 2, \ldots, K\right\}$. Here, $\mathcal{T}(x) \subseteq \{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$ denotes the set of adversarial techniques identified in image x, based on whether any correlated feature falls into the corresponding adversarial range.

This formulation supports multi-label detection, reflecting the fact that a single AQRI often exhibits multiple adversarial techniques simultaneously. The resulting set $\mathcal{T}(x)$ provides the foundation for subsequent targeted AQRI restoration.

4.4 Robust Decoder: Restoration and Reliable Decoding

Technique-Specific Feature Restoration. To restore an AQRI x, we aim to adjust its feature values so that none fall within the adversarial ranges of the identified techniques, including two phases.

• *Shared Feature Adjustment.* For each feature i that is correlated with multiple detected techniques $\text{Tech}(i) = \{\mathcal{T}_j \mid C(i, \mathcal{T}_j) > \tau\}$, we compute the restoration target range $\mathcal{R}_i^{\text{restore}} = \mathcal{R}_i^{\text{norm}} \cap \bigcap_{\mathcal{T}_j \in \text{Tech}(i)} \mathcal{R}_{i,j}^{\text{anti}}$. If the intersection is non-empty, Adato adjusts the image using tailored image processing operations (e.g., histogram equalization, deblurring) to move $v_i(x)$ into $\mathcal{R}_i^{\text{restore}}$. Otherwise, the system raises an alert indicating that the feature distribution of the image exhibits abnormal characteristics inconsistent with any known anti-adversarial range, suggesting a potentially novel or heavily obfuscated adversarial technique.

• *Single-Technique Feature Adjustment.* For features correlated with only one technique, we adjust $v_i(x)$ to fall within $\mathcal{R}_i^{\text{norm}} \cap \mathcal{R}_{i,j}^{\text{anti}}$ for the corresponding $\mathcal{T}_j$. After both phases, the resulting image x' satisfies $\forall j, \forall i$ correlated with $\mathcal{T}_j$, $\quad v_i(x') \notin \mathcal{R}_{i,j}^{\text{adv}}$, indicating successful mitigation of adversarial characteristics.

Reliable Decoding. Following the restoration processing of AQRIs, we applied the robust decoding tool of WeChat [93] to parse all restored AQRIs. Since Adato had already analyzed, identified, and mitigated known adversarial techniques to improve image scannability, this step aimed to verify the effectiveness of the restoration process through direct decoding. In cases where decoding still failed, we attributed the failure either to the presence of unknown adversarial techniques beyond our current taxonomy or to incomplete restoration. For such instances, we adopted a brute-force strategy that iteratively adjusted image parameters until a successful scan was achieved. The procedure of Adato for technique-specific restoration and decoding is detailed in Appendix A.

4.5 Evaluation of Adato

Implementation. Adato is deployed on a Linux server with 56 CPU cores, 377GB RAM, and 4 V100 GPUs. For QR code detector, we built a CNN YOLOv8 model using TensorFlow [23]. Next, for the AQRI identifier, we strictly followed official developer documentation for each tool, implemented the code to invoke them, and automatically obtained processing results from existing tools. We applied DBSCAN [27] for unsupervised clustering, and set the key hyperparameter ε as 31, by setting k as the number of features and $\varepsilon = 2k - 1$, based on our experiment. To derive the feature distribution in CQRIs, we set $\alpha = 0.05$, corresponding to a 95% confidence interval. To determine whether a feature is significantly affected by an adversarial technique, we set the threshold value of Bhattacharyya distance as τ =2.6. Finally, in the restoration phase, we used threshold adjustment to restore pixels and decode them via WeChat's QR code processing tool.

Datasets. To train, validate, and benchmark Adato, we prepare a comprehensive dataset suite spanning both controlled and real-world settings. This suite consists of two main parts: (1) a manually curated ground-truth dataset, and (2) a carefully annotated subset of real-world data for robust evaluation. Each dataset plays a distinct role in supporting various stages of

Table 5: Adato's detection performance and restoration improvement for QR code detection tools on `RealEval-QRC`.

Tool	Original Recall* (%)	After-restored Recall* (%)	Improved by
Adato	97.8	–	–
Google Vision AI [36]	13.2	77.8	5.89×
Tencent Cloud [95]	15.1	80.6	5.34×
Baidu AI Cloud [8]	11.5	71.5	6.22×
Aliyun [4]	8.6	73.2	8.51×
Partner's detector	62.1	90.3	1.45×

*: Recall means the success rate of AQRI detection.

system development and assessment.

• *Groundtruth Dataset*: To support Adato's initial QR code verification, we constructed a labeled ground-truth dataset with three image categories: AQRIs, CQRIs, and non-QR code images. Specifically, 3,000 AQRIs were collected from real-world user complaints (Section 3). In parallel, we randomly sampled 2,000 CQRIs from a publicly available dataset [20], manually verifying that they followed standard QR code encoding practices and exhibited no signs of adversarial manipulation. To provide a contrasting set of non-QR content, we randomly selected 2,000 non-QR images from the raw collected dataset in Section 4.1. All images were independently labeled and cross-validated by two researchers. The resulting 7,000-image dataset was split into training, validation, and test sets using an 8:1:1 ratio and was used for training and evaluating the QR Code Detector.

• *Real-World AQRI Evaluation Set* `(RealEval-QRC)`. To assess Adato's detection and restoration capabilities in real-world scenarios, we constructed a manually labeled evaluation dataset from our large-scale image collection. Specifically, we randomly sampled 1,000 images from raw collected dataset in Section 4.1. We iteratively annotated sampled images until we obtained a target composition of 500 AQRIs, 250 CQRIs, and 250 non-QR code images, forming a manually curated dataset denoted as `(RealEval-QRC)`. While this dataset adopts a balanced class distribution to enable systematic evaluation, it is drawn entirely from in-the-wild sources and thus retains the natural variability and noise of real-world data.

Evaluation of Adato. We evaluate Adato using `RealEval-QRC` and *Groundtruth Dataset*, first presenting step-wise results, then overall performance.

• *QR Code Detector.* On *Groundtruth Dataset*, it achieves 96.0% precision on the test set and 95.5% on the validation set. We further evaluated its performance on `RealEval-QRC`, where it demonstrates 98.6% precision, 97.8% recall, and an overall accuracy of 97.40% for AQRI detection. Furthermore, to benchmark its effectiveness, we compared Adato with several QR Code detectors. As reported in Table 5, Adato achieves an overall detection recall of 97.8%, significantly outperforming other models in detecting AQRIs. Notably, Google Vision AI achieved only 13.2% recall on AQRIs, highlighting Adato's superiority in this challenging task. Expanding our analysis to out-of-distribution scenarios, we test Adato's robustness in detecting AQRIs with new forms of visual perturbations under N-M settings (M=1,...,6). By excluding M techniques from training (full results in Table 6), we observe that under the N-1 setting (95.7% of the training data), Adato maintains high performance, achieving 97.2% precision and 91.4% recall. A sharp drop (both metrics < 65%) occurs only in the N-6 setting (75.0% data). This confirms Adato's strong generalization to out-of-distribution samples.

Table 6: Robustness of Adato under N-M Out-of-Distribution Settings.

Setting	Training Data (%)	Precision (%)	Recall (%)
N-1	95.7	97.2	91.4
N-2	93.2	93.9	86.5
N-3	81.0	79.5	74.1
N-4	78.6	71.8	68.2
N-5	77.2	67.4	65.3
N-6	75.0	< 65.0	< 65.0

• *AQRI Identifier.* We evaluate the performance of the Adversarial Signal Detection component within Adato's AQRI Identifier on `RealEval-QRC`. Adato achieves 93.3% precision, 91.2% recall, and 89.7% accuracy in identifying the presence of adversarial manipulations in QR code images. Furthermore, to evaluate Adato's capability in identifying specific adversarial manipulations, we conduct a fine-grained performance breakdown across all 13 individual techniques. As detailed in Appendix C, Adato achieves consistently high precision, recall, and accuracy across diverse techniques, demonstrating its robustness in isolating distinct adversarial attacks.

• *Robust Decoder.* Adato achieves a reliable decoding success rate of 99.87% on `RealEval-QRC` after the restoration process. Upon manual inspection, the decoding failures came from two primary sources: non-functional QR codes (e.g., icon-based or SVG types) that lacked both decodability and valid information, and images with severe damage that were thus inherently undecodable.

• *Effectiveness as a pre-adaptor.* Based on the calculated adversarial feature range, we utilize the final two stages of Adato as a pre-adaptor to improve the effectiveness of various third-party QR code detectors. As shown in Table 5, detectors such as Google Vision AI, Baidu AI Cloud, and Tencent Cloud experienced recall improvements of 5 to 8.5 times over their baselines. Furthermore, by deploying the pre-adaptor within our partner's proprietary detector, we achieved a state-of-the-art recall of 90.3%. These Results demonstrate the generality and effectiveness of Adato's visual restoration design.

We further explored the efficiency of Adato and pre-adaptor (see Table 7): for Adato, it takes 52.66 minutes for 10,000 images, demonstrating efficiency in handling massive image

Table 7: Processing Efficiency of Adato.

Stage	Runtime (min)	Images per minute
Data Crawling	5.21	1919
QR Code Detector	7.67	1304
AQRI Identifier	26.28	381
Robust Decoder	13.50	741
Overall	52.66	190

data from social media platforms. As the pre-adaptor involves only the final two stages, with no need for range calculation and decoding, its runtime is reduced to 19.68 minutes.

5 Real-World Impact of AQRI

In this section, based on Adato's detection results on the large-scale real-world dataset, we conducted a systematic measurement study on AQRI to reveal the current status of real-world usage and abuse.

5.1 Overall Dissemination of AQRI

By legally crawling image-containing posts published on 5 social media platforms, We obtained a total of 40,147,738 images contained in 24,311,446 posts. First, we identified 1,585,706 QR code-bearing images, with an average of 1.06 per post. Among these, 68,467 (4.32%) images were identified as AQRI, across 64,283 posts. After the restoration and decoding of AQRIs, we successfully parsed 68,379 valid embedded information, including 68,331 URLs and 48 text messages. Additionally, 4,018 accounts were found to post AQRIs. Of these accounts, 51.29% released at least five posts, and the maximum post count for a single account was 180.

Finding 1: Real-world results indicate that AQRIs are already widely used as dissemination vectors on social platforms, with active posting activities.

5.2 Adversarial Techniques in AQRI: Image Processing & Generative AI

Understanding the adversarial techniques employed in AQRIs is essential for uncovering how they exploit vulnerabilities in current moderation pipelines and for explaining the empirical disparities observed in detection and decoding. Based on the clustering analysis in Section 4.3, we identify 13 distinct categories of adversarial techniques. Each category is named after its closest image processing operation, forming a structured AQRI taxonomy (Figure 3) that captures how AQRIs disrupt QR code processing across four key dimensions: visual concealment, contrast degradation, structural distortion, and sharpness reduction. Further details on each technique are provided in Appendix D.

Table 8: Top 10 adversarial technique combinations.

Operation Category	Count
Color Overlay + Edge Erosion + Noise Addition + Inversion	11,736
Blurring + Haze Overlay + Geometric Deformation + Overexposure + Contrast Blending	7,694
Blurring + Haze Overlay + Overexposure + Contrast Blending	4,241
Black Overlay + Color Overlay + Edge Erosion + Noise Addition + Inversion	3,354
Color Overlay + Edge Erosion + Noise Addition + Inversion + Contrast Blending	2,819
Color Overlay + Noise Addition + Inversion	2,124
Blurring + Color Overlay + Edge Erosion + Inversion	1,591
Blurring + Haze Overlay + Overexposure + Contrast Blending + Inversion	1,504
Color Overlay + Edge Erosion + Inversion	1,317
Blurring + Color Overlay + Inversion	1,146

5.2.1 Image Processing for AQRI

Based on the identification of adversarial techniques, we analyzed the real-world usage, shown in Figure 4. First, we find that the most prevalent adversarial techniques in AQRI can be implemented via basic image processing operations, like Inversion (55.44%) and Color Overlay (48.65%). Unlike traditional adversarial attacks that rely on complex optimization [67, 68, 84, 109], AQRIs also use QR-specific obfuscation strategies, such as Geometric Deformation (20.43%) and Density Compression (1.25%). This preference likely reflects the need to maintain decodability.

Furthermore, we found that using multiple techniques is more common in AQRIs. Specifically, as shown in Figure 5, only 13.60% used a single adversarial technique. The maximum number of combined techniques reached 8 (covering 411 AQRIs). Notably, 4 techniques formed the most common combination, covering 22,457 AQRIs (32.80%). We further analyzed the most common combination strategies, as shown in Table 8. We find common combinations typically employ techniques across multiple key dimensions, as illustrated in Figure 6. This combination strategy, we argue, significantly increases adversarial effect complexity, thereby improving bypass success rates.

Finding 3: To maintain decodability, adversarial techniques in AQRIs are tailored to QR-specific scenarios rather than general complex adversarial techniques, most implemented via basic image processing operations. Though individual techniques are basic, combining multiple ones yields complex adversarial effects, ensuring a high evasion success rate at minimal cost.

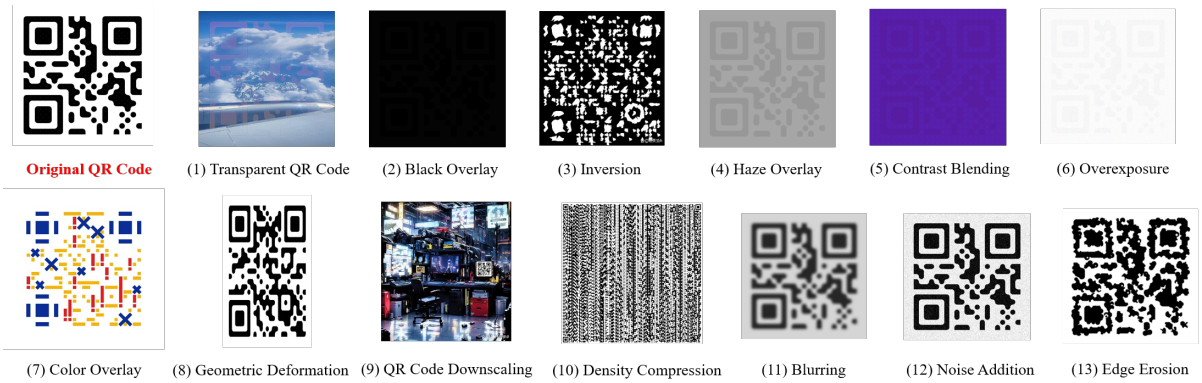


Figure 3: Examples of each adversarial technique in AQRIs.

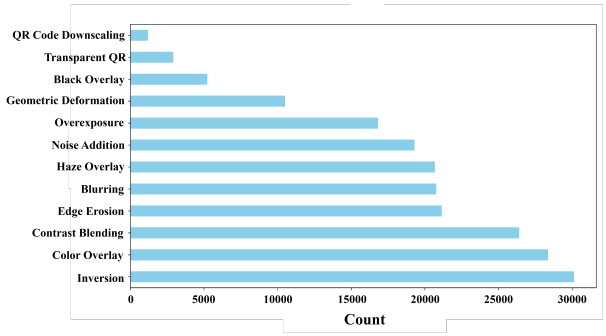


Figure 4: Adversarial techniques used in AQRIs.

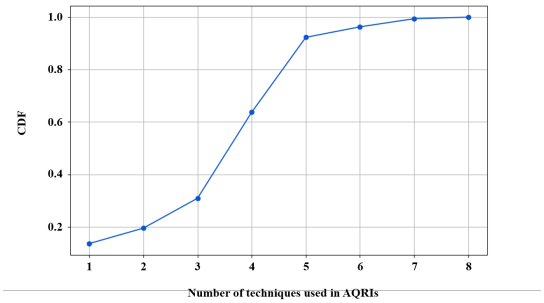


Figure 5: CDF of Adversarial Technique Usage Count in AQRIs.

5.2.2 Generative AI in AQRI

The creation of AQRIs is no longer limited to manual adversarial transformations, as adversarial image generation has become increasingly scalable and sophisticated with the rise of large-scale generative models [58]. Our analysis of in-the-wild AQRIs from Adato reveals that adversaries have begun incorporating AI-generated imagery (AIG) to construct what we term `Generative AQRIs` (as shown in Figure 6), which often embed multiple adversarial effects within a single image, enabling more complex and evasive attacks. To identify such content and assess its impact, we use Hive's AI Detector [42], which detects AI-generated images and estimates their likely source models, offering insights into the underlying generative techniques. This tool demonstrated 99% accuracy on a validation set of 100 Generative and 100 non-AI AQRIs during our small-scale verification experiment.



(a) AQRI 1 (b) AQRI 2 (c) AQRI 3

Figure 6: Examples of AQRI: (a) combining multiple adversarial techniques, (b) utilizing Generative AQRI, and (c) integrating adversarial techniques with Generative AQRI. Part of each image is obscured to prevent the dissemination of malicious content.

Prevalence of Generative AQRI. Among the 68,467 in-the-wild AQRIs identified in Section 5.1, 5,377 (7.85%) were constructed using generative AI, with 98.89% attributed to Stable Diffusion [82], making it the predominant tool for generating AI-driven AQRIs. The remaining cases involved models such as Wan [104] and Kling [52]. Notably, we find no evidence of advanced large vision models (LVMs) such as Sora [70], Nano Banana [37], or Gork [111] being employed in this dataset, indicating that adversaries currently rely on general-purpose generative models, with limited adoption of more specialized vision architectures. Over our six-month collection period, the monthly volume of Generative AQRIs surged by 350% (from 282 cases in September 2024 to 1,269 cases in March 2025), indicating a rapid escalation in their

adoption.

Bypass Performance. Generative methods naturally blend multiple adversarial effects in AQRIs, with common technique combinations (Table 8) frequently observed. For example, 4,836 (89.94%) Generative AQRIs exhibit the most prevalent combination: Color Overlay, Edge Erosion, Noise Addition, and Inversion. To assess evasion, we randomly sampled 100 Generative and 100 non-AI AQRIs and tested them against the detectors in Table 1. Generative AQRIs achieved a significantly higher average bypass rate of 89.0%, compared to 69.0% for non-AI AQRIs. Surprisingly, advanced detectors based on large vision models (LVMs), including Baidu AI Cloud [8] and Aliyun [4], performed worse on Generative AQRIs, yielding bypass rates of 87.0% and 69.0%, versus 72.0% and 31.0% for non-AI counterparts. Manual inspection shows that Stable Diffusion often generates photorealistic backgrounds that distract model attention from the QR code, causing misclassification as artistic content. Generative AI thus produces more sophisticated AQRIs, making QR codes harder to detect and moderation more difficult.

Abuse Potential: Accessibility, Restriction, and Decodability. To evaluate the feasibility of fabricating Generative AQRIs, we examined publicly available generative services based on Stable Diffusion. We collected the 20 most-starred Stable Diffusion-based QR code generators on GitHub, 16 of which offered web interfaces for direct use. First, manual review of README files and service policies showed that none of the 20 repositories or 16 web interfaces imposed restrictions or content checks on the QR payload, allowing unrestricted use and potential abuse for generating policy-violating AQRIs. Second, several generators, such as QRBTF [55], offered extensive prompts, visual customization options, and AI tutorials, significantly lowering the barrier to use. Additionally, generators like QRCode-antfu [6] include decodability verification to optimize readability, supporting the threat model by enabling reliable information delivery under evasion. Our decodability evaluation on 100 Generative and 100 non-AI AQRIs across multiple decoders showed that Generative AQRIs consistently achieved higher decoding success, with an average margin of 18%, likely due to the greater flexibility of generative methods in refining QR structures. Given their superior bypass performance, improved decodability, and ease of fabrication, Generative AQRIs constitute a compelling upgrade for adversaries. While still a minority of all AQRIs, the maturity of the technology and accessibility of supporting services make them an emerging vector for abuse.

Evaluating LVMs for Generating AQRI. Although we found no evidence of LVM use in our dataset, their ability to generate highly realistic images raises concerns about potential abuse for AQRI generation. To assess this, we selected 3 leading LVMs (Nano Banana, Sora, and Grok) and created 20 prompts based on the styles of 20 in-the-wild Generative AQRIs. Each model generated 20 AQRIs from a controlled CQRI, yielding 60 images in total. Manual inspection showed that Nano Banana and Grok preserved QR structures, while 40% of Sora's outputs failed and were discarded. Bypass evaluation achieved a 73.5% average success rate—higher than non-AI AQRIs but lower than Stable Diffusion. Decodability remained a major limitation: only 8 of 60 images (13.3%) were readable. This illustrates what we call "Functional Blindness" in LVMs: they replicate the visual appearance of AQRIs sufficiently to evade detection but fail to preserve the syntactic structure needed for decoding. Unlike Stable Diffusion, which refines existing CQRIs, LVMs generate content from scratch and often ignore decoding constraints, making them currently ineffective for producing functional AQRIs.

Finding 4: Generative AQRIs, primarily created via Stable Diffusion, are highly evasive, reliably decodable, and easily accessible given unrestricted usage policies and low barriers to use. Their ability to blend multiple adversarial effects while preserving functionality poses a growing threat to existing moderation systems. In contrast, large vision models remain ineffective for functional AQRI generation due to poor decodability.

5.3 Information Behind AQRI

To uncover their adversarial intent, we decoded 68,467 AQRIs via Adato and extracted valid content from 68,379 of them. Of these, 99.80% (68,331) embedded URLs, indicating AQRIs are primarily used to redirect to promotional websites. Analyzing the remaining 48 text-encoded AQRIs, we found that 15 of these entries were fraudulent, including spammer contacts and fraudulent promotional content.

URL Category. URLs (99.80%) are the primary information embedded in AQRIs, so we analyzed them further. Deduplication yielded 2,136 distinct URLs, including 861 fully qualified domain names (FQDNs) and 639 second-level domain names (SLDs). Manual inspection of these domains and their webpage content identified four domain types associated with AQRI activities. An initial double-coding of 300 randomly selected URLs (14%) achieved 100% agreement, allowing the remainder to be confidently annotated by a single researcher. First, 501 short URLs (23.46%) involve 13,642 AQRIs, mainly from 6 services (e.g., Twitter, Weibo, Google, ibit.com). Short links enable auto-redirection and are widely used to hide malicious links [121], increasing the moderation difficulty of AQRI and their embedded info. Of these, 407 URLs (10,281 AQRIs) provide porn app downloads (covert channels), and 83 (2,011 AQRIs) offer illegal LLM-based services (e.g., porn chatbots, drug sales). Second, 406 URLs (19.01%) belong to 5 major social media platforms (e.g., WeChat, Instagram), linking to 8,019 AQRIs. These URLs typically redirect to platform-specific accounts or posts. In-depth analysis shows linked accounts engage in illegal services: fraud (e.g., fake investments, pig-butchering scams) and porn promotion. Three confirmed Instagram accounts

promote pornography/gambling sites, with 293k followers and 5,896 AQRI posts. Notably, 235 URLs (11.00%) from government/educational domains involve 6,667 AQRIs. Adversaries exploit these sites' vulnerabilities to compromise; they deploy AQRIs to leverage authoritative credibility for covert promotion [87, 120]. Finally, personally built sites[4] are the most common, accounting for 994 URLs (46.54%). 99.95% of these sites host malicious payloads spanning the seven categories detailed below.

URL Content Businiess. To further reveal AQRI-propagated content, we identified 8 scenarios from related URLs (by domain, post, and webpage content). Legitimate promotion is rare (2.67% of URLs, e.g., personal websites). The remaining 2,079 URLs (97.33%) link to 65,578 AQRIs (95.78%) promoting malicious activities, covering 7 types: Pornography (65.12%), gambling (29.63%), account trading (2.45%), drug sales (1.14%), proxy services (0.76%), crypto sales (0.60%), and malware delivery (0.30%). Payload distributions exhibit clear cross-platform divergence. Pornographic content concentrates heavily on Baijiahao and Tieba, comprising 77.6% of the malicious AQRIs detected on these platforms. In contrast, gambling and drug-related promotions prevail on Twitter, Instagram, and Reddit, accounting for 80.2%, 52.3%, and 69.8% of their respective AQRI volumes. Despite active malicious AQRI propagation, professional link detectors largely overlook them. For example, VirusTotal [41] flagged only 143 URLs as malicious. This stems from AQRI's diverse URL types (e.g., social platforms, educational sites), inherent stealth, and low attention.

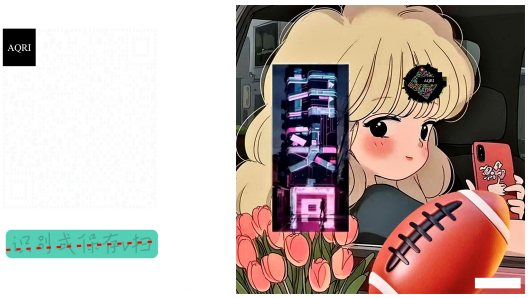
Finding 5: Analysis of embedded info shows AQRI is widely abused in illegal businesses. Online pornography is most common; notably, government/educational domains are misused to spread AQRIs.

5.4 Information Dissemination Strategies

To unveil the strategy of AQRI dissemination, we analyze AQRI posts to identify traffic-driving and user-attraction techniques, and uncover methods to obfuscate illicit operations, optimize evasion, and circumvent content-layer censorship.

Strategy 1: Additional Adversarial Signal. Promoters employ multi-modal evasion techniques. First, we utilized Google Vision AI [36] to classify image content and identify specific threats (e.g., porn). Manual verification confirmed that attackers frequently mask illicit intent behind benign imagery, such as cartoons redirecting to adult content. Besides, adversaries also embed and obfuscate extra text in QR images. For instance, "微信扫描 (WeChat scan)" may be altered to "V-扫 (V-Scan)" by replacing "微" with "V" (see Figure 7). Finally, adversaries forge trust indicators to simulate authenticity. Some QRs embed counterfeit icons from trusted platforms (e.g., fake WeChat QRs), misleading users into believing they are legitimate while bypassing actual WeChat services.

Strategy 2: Mimicking Normal Behavior. To evade moderation during dissemination, adversaries mimic legitimate user behavior. For example, 54.96% of posts contain only two images, imitating casual sharing to evade "marketing account" detection, and 69.34% are posted after 21:00, likely exploiting perceived weaker nighttime moderation. Second, adversaries exploit cross-platform differences in QR moderation to construct indirect propagation paths. Specifically, AQRI-based malicious content is posted on low-moderation platforms (e.g., Baidu Tieba) and users are guided to scan it using high-decoding platforms (e.g., WeChat), enabling redirection and dissemination of malicious resources. This gap turns weaker platforms into "entry intermediaries" in the attack chain (see Figure 8).



(a) AQRI 1 contains adversarial text (b) AQRI 2 contains adversarial text

Figure 7: Examples of adversarial text usage in AQRI.

Strategy 3: Enhancing Redirection Challenge. When directing users to malicious resources, adversaries adopt multi-dimensional evasion strategies. First, most platforms inspect only initial QR redirects, leaving multi-level chains unchecked. Adversaries exploit this gap by creating long redirect paths. We observed heavily tampered QRs leading to intermediate pages with slightly modified QRs, which further redirected to a WeChat account that automatically forwarded users to illegal gambling sites. Access restrictions are also employed: 136 URLs used cloaking to hide malicious content from moderation crawlers. Besides that, they exploit or forge

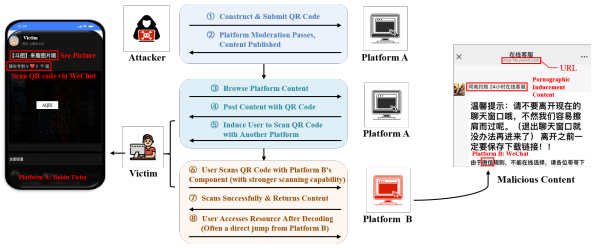


Figure 8: Workflow of Indirect Propagation Path

legitimate info to mislead detectors. They illegally exploited

[4] We define personally built sites as domains independently registered and hosted by individual developers.

legitimate domains (e.g., government/educational or serverless platforms [62]) to host AQRIs; e.g., 3,529 AQRIs were found, uploaded to such sites with porn content. Additionally, promoters hide malicious links behind normal images via short links, extending redirection chains to complicate auditing.

Finding 6: AQRI dissemination activities show diverse adversarial promotion strategies to further increasing moderation difficulty.

6 Discussion

Method Limitations. Despite efforts in method design and implementation, several limitations remain. First, Adato identifies QR codes using finder patterns according to specifications. This can produce false positives when images contain similar patterns, such as grid-like windows. Manual analysis of 2,000 randomly sampled images from the QR code detector on the In-the-Wild Dataset found only 11 false positives, indicating a negligible impact due to the very low false positive rate. Moreover, these misidentified images lack decodability and are filtered out in step 3 (Robust Decoder), leaving results unaffected. Second, we cluster AQRIs using 16-dimensional features to identify adversarial techniques. Although this may introduce some errors, a precision of 93.3% ensures their impact on the analysis is minimal.

Responsibily Abuse Threat Disclosure. Based on our measurements, we disclosed AQRI-related risks to affected social media platforms by reporting detected AQRIs and post links, and summarizing potential moderation gaps revealed by differences in adversarial techniques. Several vendors responded positively, and we shared our enhanced tool, Adato, as an adaptor to support their moderation mechanisms.

Lessons Learned and Mitigation. Our work is the first to systematically reveal real-world AQRIs, which are severely abused in illegal promotions-showing current moderation mechanisms fail to address emerging AQRI risks. Based on our findings, we propose two mitigation recommendations.

• *Enhance moderation capabilities against AQRIs.* Active testing and risk disclosure show that current moderation mechanisms remain largely unaware of AQRI risks. Our proposed tool, Adato, can serve as a pre-enhancer for existing QR code tools, improving moderation by detecting and correcting adversarial perturbations. We will open-source Adato along with the AQRI ground-truth dataset for research use.

• *Enhance user prompts.* QR code moderation must consider both the image and its embedded resources. Some platforms (e.g., Zhihu [122]) lack security prompts for QR-induced redirections. Unlike direct links, QR codes prevent users from previewing embedded content, allowing adversaries to use AQRIs to bypass moderation and deliver malicious content. Therefore, platforms should enhance QR code detection and provide effective user prompts before scanning.

7 Related Work

QR Code Detection. QR codes are increasingly abused in malicious activities, e.g., malicious promotion [117] and covert attacks [7, 39, 59, 107, 114]. For effective defense, existing work focuses on enhancing QR code detection. Liao et al. [60] proposed a training-free QR code generator based on Stable Diffusion. Vinay et al. [25] designed a fast detection scheme combining progressive discrimination with MobileNet to boost practical accuracy. *However, current techniques severely lack AQRI detection capabilities, enabling AQRIs to bypass existing mechanisms effectively.*

Image Content Moderation. Image content moderation is key to securing social media security, with ML-based image processing models as primary technical means. Platzer et al. [77] proposed ML-based skin detection for porn images, while Yuan et al. [117] introduced R-CNN-based methods for illicit promotion image detection. For videos, Wehrmann et al. [106] classified adult content via CNN-LSTM fusion, and Perez et al. [75] improved accuracy by combining static and motion information. *However, these methods neglect the impact of adversarial techniques applied to QR Codes.*

Adversarial Attack and Defense. Adversarial attacks add subtle perturbations to images, misleading AI models into misclassification while preserving human recognition. Their generation methods evolved from gradient-based approaches [22, 34, 47, 54, 64, 68, 84, 115], to boundary exploration methods for black-box scenarios [13, 16, 17, 19, 43, 44, 61, 67, 73, 74]. Wu et al. [109] further explored ways to enhance adversarial example transferability. Given the attack capability of adversarial examples on AI models, their defense is a research focus. Qin et al. [78] used class-conditional reconstruction for detection, noting CapsNet outperforms CNNs in perceptual alignment to aid this. Li et al. [57] leveraged contextual inconsistency of adversarial patterns in images for external detection. Yin et al. [116] introduced generative adversarial training to learn a detector, enhancing robustness against adaptive attacks via asymmetric training. *However, existing research on adversarial examples targets general scenarios. The unique structure of QR codes makes existing generation and defense methods hard to apply directly to AQRIs.*

8 Conclusion

Adversarial QR Code Images (AQRI) emerge to bypass current social media platforms' QR code moderations by adding perturbations to conventional QR code images (CQRIs). Our work is the first to systematically measure this emerging threat. First, based on ground-truth dataset analysis, we designed a finder-pattern-based enhanced AQRI identification method, Adato, followed by restoring added adversarial techniques to ensure successful decoding. Using Adato on 40,147,738 images collected from 5 social media platforms, we detected 68,467 AQRIs. We conducted the first real-world measure-

ment analysis on these AQRIs, covering their adversarial techniques, embedded promotional information, campaign strategies, and promotional gains. Combined with responsible disclosure, we propose several mitigation recommendations, and will open-source our identification and restoration tool, Adato, to help enhance AQRI moderation capabilities.

Acknowledgments

We sincerely thank all anonymous reviewers and our shepherd for their valuable and constructive comments on improving the paper. This work is supported by Zhongguancun Laboratory.

Ethical Considerations

First of all, through a formal collaboration with our partner, all research activities were conducted under the supervision of the partner's legal department, partially compensating for the absence of an Institutional Review Board (IRB) at our own institution. We follow the ethical guidelines in the Belmont Report [29] and the Menlo Report [51], and proactively address three ethical considerations relevant to this study: (1) the use of user complaint data in Section 3; (2) active posting experiments in Section 3; (3) post and image data collection from five social media platforms in Section 4 and (4) researcher protection during the manual analysis of URLs behind AQRIs in Section 5.

First, to collect real-world AQRIs as our ground-truth dataset, we leverage large-scale user complaint data with the assistance of our partner. Although the complaint interface includes an explicit privacy protection statement, such data may still contain personally identifiable information (PII). To mitigate these ethical concerns, our partner performs manual review and anonymization (i.e., salted hashing) of all sensitive information before our data access. As a result, researchers never interact with any user PII and only analyze images and image-related post content from reported complaints.

Second, to accurately evaluate AQRI's impact on moderation, we implemented active posting experiments in Section 3. However, on platforms that restrict QR codes, this could potentially conflict with their Terms of Service (ToS). Following legal advice, we conducted a stakeholder-aware risk assessment that explicitly accounted for potential harms to both platforms and users, and we therefore adopted the following strict mitigation measures: 1) we deleted all posts within one minute of submission, thereby testing only immediate moderation feedback; 2) each post included a disclaimer stating the experimental purpose and a contact email, and we committed to immediately terminating the experiment upon any user complaint; 3) if a post remained unbanned after one minute, we reported it before deletion to alert the platform; 4) we constructed the QR codes to mimic real AQRIs exclusively at the image level (i.e., visual appearance and distortion effects), ensuring all decoded payloads were entirely benign plain text devoid of any malicious URLs or harmful content. Crucially, as a simulated evaluation, these experiments exposed real-world vulnerabilities and can support the development of defenses against emerging risks; in this sense, the anticipated benefits plausibly outweighed the potential harms. We received no negative user feedback. Given the one-minute exposure window, the maximum view count was 6, with 87.9% of successful posts had only a single view (by the researcher), which further minimized potential impact.

Third, to measure the real-world prevalence and impact of AQRIs at scale, we analyzed public posts containing images (including QR images) from five major social media platforms. For platforms with official data APIs (Reddit, Instagram, and Twitter), we legally purchased API access with the assistance of our partner and strictly complied with the relevant platform policies. For platforms without public APIs (Baidu Tieba and Baijiahao), we developed custom crawlers that adhere to robots.txt and platform-specific crawling policies. We further enforced rate limiting (e.g., minimum request intervals) to reduce server load and avoid violations of the platforms' ToS, consistent with prior work [119]. Moreover, during data collection, we did not collect any sensitive user account information (e.g., usernames). Although public posts rarely contain explicit personally identifiable information (PII), we further mitigated privacy risks by anonymizing any potential PII that appeared in AQRI-related posts. We used all data solely for technical analysis and did not involve human-subject interaction. Our partner securely stored the data on physically isolated servers, with access restricted to authorized administrators. We have already disclosed all detected AQRIs and posts to related social media platforms to help them timely identify and defend against the emerging AQRI risk, and we will permanently delete all raw data upon project completion. Overall, our partner's legal team supervised all data collection, processing, and storage procedures to minimize potential ethical risks.

Finally, to protect researchers when accessing URLs behind AQRIs, we implemented strict technical and psychological safeguards: 1) all accesses were conducted in an isolated virtual machine using a sandboxed browser (no persistent storage, script execution disabled) to prevent exposure to potential malware; 2) all researchers received specialized training prior to access. Upon encountering malicious content, we logged only categorical tags without storing any page resources. Recognizing the risk of psychological distress, researchers conducted the review at a self-managed pace with regular well-being check-ins and could pause the experiment at any time or access institutional counseling services if they experienced discomfort.

Responsible Disclosure. Beyond the above measures, we proactively disclosed all our experimental findings, including results from active probing experiments and large-scale mea-

surements, via their reporting channels and Security Response Centers (SRCs) of the five analyzed social media platforms. This disclosure aimed to raise awareness of AQRI-related risks and to provide platforms with concrete post-level evidence to facilitate timely mitigation. To date, we have received positive responses from two platforms. Furthermore, we reported the detection flaws to the evaluated vision API providers, documenting our contact dates and their responses.

Open Science

Our research adheres to open science principles to foster reproducibility and collaboration.

System components. We open-source the vast majority of code in Adato, except for the Data Collector which contains sensitive API credentials. All other components are published on https://doi.org/10.5281/zenodo.20322613. The Feature Restoration part can serve as an adapter for the preprocessing stage of existing recognition and scanning tools, significantly improving the performance of QR code detectors against AQRIs. To ensure third-party users can utilize our work, we will launch a webpage to demonstrate our results. This webpage will feature an online interface for users to submit and detect AQRIs, along with our contact information.

Ground-truth dataset. To help the community understand and defend against AQRIs, we release the AQRI Ground-truth Dataset to enable threat analysis and improve detector training. However, due to the malicious nature of the pictures, the dataset is hosted on https://doi.org/10.5281/zenodo.20325847 under Restricted Access.

Vulnerability disclosure. We are responsibly disclosing our findings to affected platforms and assisting them in deploying our enhancer to comprehensively mitigate this security risk. Therefore, the detection results are withheld because our vulnerability disclosure and platform remediation are still ongoing. Interested researchers may request the data via the contact information that will be provided on our webpage.

References

[1] ISO/IEC 18004. Information technology automatic identification and data capture techniques bar code symbology qr code, 2000.

[2] Adsupervisor. Top 100 IG Hashtags. https://adsupervisorhk.com/blog/instagram-hashtag-tutorial. (Access in February, 2026).

[3] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Proc. of ACCV*, pages 622–637. Springer, 2018.

[4] Alibaba. Aliyun. https://www.aliyun.com/. (Access in February, 2026).

[5] Alibaba. Alipay. https://www.alipay.com/. (Access in February, 2026).

[6] Anthony Fu. Anthony's QR Toolkit. https://qrcode.antfu.me/. (Access in February, 2026).

[7] Xiaolong Bai, Zhe Zhou, XiaoFeng Wang, et al. Picking up my tab: Understanding and mitigating synchronized token lifting and spending in mobile payment. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 593–608, 2017.

[8] Baidu. Baidu AI Cloud. https://cloud.baidu.com/product/imagecensoring. (Access in February, 2026).

[9] Baidu. Baidu Baijiahao. https://baijiahao.baidu.com/. (Access in February, 2026).

[10] Baidu. Baidu Tieba. https://tieba.baidu.com/. (Access in February, 2026).

[11] Baidu. Baidu Tieba Community Guidelines. https://tieba.baidu.com/tb/pc/common-pc.html?key=b337c655543620a37c372836f277b76115. (Access in February, 2026).

[12] Luiz Belussi and Nina Hirata. Fast qr code detection in arbitrarily acquired images. In *Proc. of SIBGRAPI*, pages 281–288. IEEE, 2011.

[13] Thomas Brunner et al. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *Proc. of ICCV*, pages 4958–4966, 2019.

[14] Fabio Carrara et al. Adversarial image detection in deep neural networks. *Multimedia Tools and Applications*, 78(3), 2019.

[15] George Casella and Roger L Berger. *Statistical inference*. Duxbury Press, 2002.

[16] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *Proc. of IEEE S&P*. IEEE, 2020.

[17] Pin-Yu Chen et al. Zoo: Zeroth order optimization-based black-box attacks to deep neural networks without training substitute models. In *Proc. of ACM AISec*, pages 15–26, 2017.

[18] Rongjun Chen et al. Rapid detection of multi-qr codes based on multistage stepwise discrimination and a compressed mobilenet. *IEEE IOTJ*, 10(18):15966–15979, 2023.

[19] Minhao Cheng et al. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018.

[20] Coledie. Qr codes. https://www.kaggle.com/datasets/coledie/qr-codes, 2022. Accessed: 2024-07-31.

[21] Adrian Dabrowski et al. Qr inception: Barcode-in-barcode attacks. In *Proc. of SPSM*, pages 3–10, 2014.

[22] Yingpeng Deng and Lina J Karam. Universal adversarial attack via enhanced projected gradient descent. In *Proc. of ICIP*, pages 1241–1245. IEEE, 2020.

[23] TensorFlow Developers. Tensorflow. *Zenodo*, 2022.

[24] Markéta Dubská, Adam Herout, and Jiří Havel. Real-time precise detection of regular grids and matrix codes. *JRTIP*, 11(1):193–200, 2016.

[25] Vinay Edula, Kalyan Ammisetty, Aakash Kotha, et al. A novel framework for qr code detection and decoding from obscure images using yolo object detection and real-esrgan image enhancement technique. In *Proc. of ICCCNT*, pages 1–6. IEEE, 2023.

[26] Eric-Canas, jandom, scito, MichaelCurrie. Qreader. https://github.com/Eric-Canas/QReader. (Access in February, 2026).

[27] Martin Ester et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[28] Riccardo Focardi, Flaminia L Luccio, and Heider AM Wahsheh. Usable security for qr code. *JISA*, 48:102369, 2019.

[29] United States. National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. *The Belmont report: ethical principles and guidelines for the protection of human subjects of research*. Department of Health, Education and Welfare, 1979.

[30] Scott Freitas et al. Unmask: Adversarial detection and defense through robust feature alignment. In *IEEE BigData*, pages 1081–1088. IEEE, 2020.

[31] Gonzalo J Garateguy et al. Qr images: optimized image embedding in qr codes. *IEEE TIP*, 23(7):2842–2853, 2014.

[32] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[33] Ross Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of CVPR*, pages 580–587, 2014.

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

[35] Google. Google. https://www.google.com/. (Access in February, 2026).

[36] Google. Google Cloud Vision AI. https://cloud.google.com/vision/. (Access in February, 2026).

[37] Google DeepMind. Nano Banana (Gemini Image Generation). https://gemini.google/overview/image-generation. (Access in February, 2026).

[38] Keyan Guo et al. Moderating illicit online image promotion for unsafe user generated content games using large {Vision-Language} models. In *USENIX Security 24*, pages 5787–5804, 2024.

[39] Xing Han et al. Medusa attack: Exploring security hazards of {in-app}{QR} code scanning. In *USENIX Security 23*, pages 4607–4624, 2023.

[40] Yu He and Yang Yang. An improved sauvola approach on qr code image binarization. In *ICAIT*, pages 6–10. IEEE, 2019.

[41] Hispasec Sistemas Company. Virus Total. https://www.virustotal.com/gui/home/search. (Access in February, 2026).

[42] Hive. Hive AI Detector. https://hivemoderation.com/ai-generated-content-detection. (Access in February, 2026).

[43] Andrew Ilyas, Logan Engstrom, Anish Athalye, et al. Black-box adversarial attacks with limited queries and information. In *ICML*, pages 2137–2146. PMLR, 2018.

[44] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018.

[45] Instagram. Instagram. https://www.instagram.com/. (Access in February, 2026).

[46] Instagram. Instagram api. https://developers.facebook.com/docs/instagram-api. (Access in February, 2026).

[47] Uyeong Jang, Xi Wu, and Somesh Jha. Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In *ACSAC*, pages 262–277, 2017.

[48] Jeff Brown. Zbar. http://zbar.hg.sourceforge.net:8000/hgroot/zbar/zbar. (Access in February, 2026).

[49] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE TCOM*, 15(1):52–60, 1967.

[50] Abdul Karim et al. Phishing detection system through hybrid machine learning based on url. *IEEE Access*, 11:36805–36822, 2023.

[51] Erin Kenneally and David Dittrich. The menlo report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102*, 2012.

[52] Kuaishou's Large Model Algorithm Team. Kling AI. https://app.klingai.com/global/. (Access in February, 2026).

[53] Jitendra Kumar et al. Phishing website classification and detection using machine learning. In *ICCCI*, pages 1–6. IEEE, 2020.

[54] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[55] Latent Cat. QRBTF (QR Code Beautifier). https://qrbtf.com/. (Access in February, 2026).

[56] Kejing Li, Fanwu Meng, Zhipeng Huang, and Qi Wang. A correction algorithm of qr code on cylindrical surface. In *Journal of Physics: Conference Series*, volume 1237, page 022006. IOP Publishing, 2019.

[57] Shasha Li et al. Connecting the dots: Detecting adversarial perturbations using context inconsistency. In *ECCV*, pages 396–413. Springer, 2020.

[58] Xinfeng Li, Yuchen Yang, Jiangyi Deng, et al. Safegen: Mitigating sexually explicit content generation in text-to-image models. In *Proc. of ACM CCS*, pages 4807–4821. ACM, 2024.

[59] Yijie Li, Yi-Chao Chen, Xiaoyu Ji, et al. Screenid: Enhancing qrcode security by fingerprinting screens. In *IEEE INFOCOM*, pages 1–10. IEEE, 2021.

[60] Jia-Wei Liao, Winston Wang, Tzu-Sian Wang, et al. Diffqrcoder: Diffusion-based aesthetic qr code generation with scanning robustness guided iterative refinement. In *IEEE/CVF WACV*, pages 5916–5925. IEEE, 2025.

[61] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[62] Yijing Liu, Mingxuan Liu, Yiming Zhang, Baojun Liu, Jia Zhang, Geng Hong, Haixin Duan, and Min Yang. Dive into the cloud: Unveiling the (ab) usage of serverless cloud function in the wild. In *Proceedings of the 2025 ACM Internet Measurement Conference*, pages 63–77, 2025.

[63] M. DeCarlo. "AVG: QR Code-based Malware Attacks to Rise in 2012."Techspot News. http://www.techspot.com/news/47189-avg-qr-code-based-malware-attacks-to-rise-in-2012.html. (Access in February, 2026).

[64] Aleksander Madry et al. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[65] Vasileios Mavroeidis and Mathew Nicho. Quick response code secure: A cryptographically secure anti-phishing tool for qr code attacks. In *MMM-ACNS*, pages 313–324. Springer, 2017.

[66] Meta. Instagram Community Guidelines. https://transparency.meta.com/en-gb/policies/community-standards/. (Access in February, 2026).

[67] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, pages 2574–2582, 2016.

[68] Maria-Irina Nicolae et al. Adversarial robustness toolbox v1. 0.0. *arXiv preprint arXiv:1807.01069*, 2018.

[69] nvm. Unoriginal, low-quality and QR code content in almost every video - Is there a working method to avoid this? https://www.blackhatworld.com/seo/unoriginal-low-quality-and-qr-code-content-in-almost-every-video-is-there-a-working-method-to-avoid-this.1657624/. (Access in February, 2026).

[70] OpenAI. Sora: Creating image from text. https://sora.chatgpt.com/. (Access in February, 2026).

[71] OpenCV. WeChatQRCode Class Reference. https://github.com/opencv/opencv_contrib/blob/master/modules/wechat_qrcode/src/wechat_qrcode.cpp#L156. (Access in February, 2026).

[72] Nicolas Papernot et al. The limitations of deep learning in adversarial settings. In *EuroS&P*, pages 372–387. IEEE, 2016.

[73] Nicolas Papernot et al. Practical black-box attacks against machine learning. In *Proc. of Asia CCS*, pages 506–519, 2017.

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

[75] Mauricio Perez et al. Video pornography detection through deep learning techniques and motion information. *Neurocomputing*, 230:279–293, 2017.

[76] Peter Abeles. BoofCV. https://boofcv.org/. (Access in February, 2026).

[77] Christian Platzer, Martin Stuetz, and Martina Lindorfer. Skin sheriff: a machine learning solution for detecting explicit images. In *SFCS*, pages 45–56, 2014.

[78] Yao Qin et al. Detecting and diagnosing adversarial images with class-conditional capsule reconstructions. *arXiv preprint arXiv:1907.02957*, 2019.

[79] Reddit. Praw: The python reddit api wrapper. https://praw.readthedocs.io/en/stable/. (Access in February, 2026).

[80] Reddit. Reddit. https://www.reddit.com/. (Access in February, 2026).

[81] reddit\r\videos. reddit\r\videos. https://www.reddit.com/r/videos/. (Access in February, 2026).

[82] Robin Rombach et al. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[83] Doyen Sahoo, Chenghao Liu, and Steven CH Hoi. Malicious url detection using machine learning: A survey. *arXiv preprint arXiv:1701.07179*, 2017.

[84] Jaydip Sen and Subhasis Dasgupta. Adversarial attacks on image classification models: Fgsm and patch attacks and their impact. *arXiv preprint arXiv:2307.02055*, 2023.

[85] Filipo Sharevski, Amy Devine, Emma Pieroni, and Peter Jachim. Phishing with malicious qr codes. In *EuroUSEC*, pages 160–171, 2022.

[86] Jen-Yu Shieh, Jia-Long Zhang, Yu-Ching Liao, and Chih-Ming Lin. Enhancing the recognition rate of two-dimensional barcodes image and applications. In *CISP*, volume 3, pages 1567–1571. IEEE, 2011.

[87] Ravindu De Silva et al. Compromised or attacker-owned: A large scale classification and study of hosting domains of malicious urls. In *USENIX Security*, pages 3721–3738, 2021.

[88] Sina Weibo. Sina Weibo. https://weibo.com/. (Access in February, 2026).

[89] Sina Weibo. Weibo Community Guidelines. https://service.account.weibo.com/h5/roles/gongyue. (Access in February, 2026).

[90] Gábor Sörös and Christian Flörkemeier. Blur-resistant joint 1d and 2d barcode localization for smartphones. In *MUM*, pages 1–8, 2013.

[91] Hao Su et al. Artcoder: an end-to-end method for generating scanning-robust stylized qr codes. In *CVPR*, pages 2277–2286, 2021.

[92] Christian Szegedy et al. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[93] Tencent. Wechat. https://www.wechat.com/. (Access in February, 2026).

[94] Tencent. QQ. https://im.qq.com/. (Access in February, 2026).

[95] Tencent. Tencent Cloud. https://www.tencentcloud.com/. (Access in February, 2026).

[96] TikTok. TikTok. https://www.tiktok.com/. (Access in February, 2026).

[97] TikTok. TikTok Advertesment Policy. https://ads.tiktok.com/help/article/tiktok-ads-policy-ad-format-and-functionality. (Access in February, 2026).

[98] Sumit Tiwari. An introduction to qr code technology. In *ICIT*, pages 39–44. IEEE, 2016.

[99] Zanjia Tong, Yuhang Chen, Zewei Xu, and Rong Yu. Wise-iou: bounding box regression loss with dynamic focusing mechanism. *arXiv preprint arXiv:2301.10051*, 2023.

[100] Twiiter Trends 24. Twitter Trends Worldwide | Top Twitter Trending Hashtags & Topic Right Now. https://www.twittertrends24.com/. (Access in February, 2026).

[101] Twitter. Twitter. https://x.com/. (Access in February, 2026).

[102] Ultralytics. YOLOv8. https://github.com/ultralytics/ultralytics. (Access in February, 2026).

[103] Ashish Vaswani et al. Attention is all you need. *NeurIPS*, 30, 2017.

[104] wan ai Alibaba. Wan: Open and Advanced Large-Scale Video Generative Models. https://wanai.pro/. (Access in February, 2026).

[105] Jingyi Wang et al. Adversarial sample detection for deep neural network through model mutation testing. In *ICSE*, pages 1245–1256. IEEE, 2019.

[106] Jônatas Wehrmann, Gabriel S Simões, Rodrigo C Barros, and Victor F Cavalcante. Adult content detection in videos with convolutional and recurrent neural networks. *Neurocomputing*, 272:432–438, 2018.

[107] Chuxiong Wu and Qiang Zeng. Do you see how i pose? using poses as an implicit authentication factor for {QR} code payment. In *USENIX Security*, pages 4571–4588, 2024.

[108] Guangyang Wu et al. Text2qr: Harmonizing aesthetic customization and scanning robustness for text-guided qr code generation. In *CVPR*, pages 8456–8465, 2024.

[109] Weibin Wu et al. Boosting the transferability of adversarial samples via attention. In *CVPR*, pages 1161–1170, 2020.

[110] X. x api. https://developer.x.com/en/products/x-api/enterprise. (Access in February, 2026).

[111] xAI. Gork (Grok AI). https://grok.com. (Access in February, 2026).

[112] Xiaohongshu. Xiaohongshu Community Guidelines. https://agree.xiaohongshu.com/h5/terms/ZXXY20221213003/-1. (Access in February, 2026).

[113] Mingliang Xu et al. Stylized aesthetic qr code. *IEEE TMM*, 21(8):1960–1970, 2019.

[114] Guangtao Xue et al. Screenid: Enhancing qrcode security by utilizing screen dimming feature. *IEEE/ACM ToN*, 31(2):862–876, 2022.

[115] Chin-Yuan Yeh et al. Attack as the best defense: Nullifying image-to-image translation gans via limit-aware adversarial attack. In *CVPR*, pages 16188–16197, 2021.

[116] Xuwang Yin, Soheil Kolouri, and Gustavo K Rohde. Gat: Generative adversarial training for adversarial example detection and robust classification. *arXiv preprint arXiv:1905.11475*, 2019.

[117] Kan Yuan et al. Stealthy porn: Understanding real-world adversarial images for illicit online promotion. In *IEEE S&P*, pages 952–966. IEEE, 2019.

[118] Z. CROSSING. Multi-format 1D/2D barcode image processing library implemented in Java, with ports to other languages. https://github.com/zxing/zxing. (Access in February, 2026).

[119] Mingming Zha et al. Understanding cross-platform referral traffic for illicit drug promotion. In *ACM CCS*, pages 2132–2146, 2024.

[120] Jialong Zhang, Chao Yang, Zhaoyan Xu, and Guofei Gu. Poisonamplifier: A guided approach of discovering compromised websites through reversing search poisoning attacks. In *RAID*, volume 7462 of *Lecture Notes in Computer Science*, pages 230–253, 2012.

[121] Zhibo Zhang et al. Misdirection of trust: Demystifying the abuse of dedicated URL shortening service. In *NDSS*, 2025.

[122] Zhihu. Zhihu. https://www.zhihu.com/. (Access in February, 2026).

[123] Zhihu. Zhihu Community Guidelines. https://zhuanlan.zhihu.com/p/506696688. (Access in February, 2026).

[124] Anfu Zhou, Guangyuan Su, Shilin Zhu, and HuaDong Ma. Invisible qr code hijacking using smart led. *IMWUT*, 3(3):1–23, 2019.