MF-GGNN: Crack Visual Reasoning CAPTCHA Holistically Using a Novel Multi-Feature Fusion-based Graph Gated Neural Network

Botao Xu, Haizhou Wang, *Member, IEEE,* Wenxian Wang, and Xingshu Chen

***Abstract*—With the advancement of computer vision and deep learning algorithms, many traditional text-based and image-based CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) have faced the significant risk of being automatically recognized and cracked. To enhance security, some leading CAPTCHA service providers have begun to explore interactive CAPTCHAs that integrate visual reasoning tasks, which greatly increase machine recognition difficulty. Compared to traditional CAPTCHAs, these CAPTCHAs leverage the unique advantage of the human brain in comprehensively analyzing and judging real-world information. Existing visual reasoning CAPTCHA research employs non-end-to-end model. However, it is unable to fully learn the rich features from CAPTCHAs through holistically modeling the logical relations. Additionally, the modular approach requires training modules separately and assembling them at the end, leading to substantial manual effort. To address the issues mentioned above, this paper presents an end-to-end learning framework called MF-GGNN (Multi-Feature Fusion-based Graph Gated Neural Network). The framework utilizes GGNN to simulate human association and reasoning over visual content. Specifically, we first locate objects in the image using an object detection model. We then extract multiple features and generate question attention distribution. Finally, we integrate the above outputs using a graph neural gated network for multi-step reasoning. Our model can capture the deep intrinsic connections between objects and their spatial information to achieve end-to-end graph reasoning. MF-GGNN achieves an average attack success rate of 92.2% across multiple CAPTCHAs. The results of various experiments demonstrate that our method can excellently accomplish visual reasoning tasks in CAPTCHAs. In addition, we construct a large-scale, open-source dataset, namely ViRC (Visual Reasoning CAPTCHAs), including over 50,000 CAPTCHAs and corresponding questions. Our research findings will provide references and insights for evaluating and designing more secure and robust next-generation CAPTCHAs.**

***Index Terms*—CAPTCHA, visual reasoning, multi-feature, question attention, graph gated neural network.**

I. Introduction

THE evolution of websites and mobile applications in the Information Age has been fueled by technological advancements, which provide convenience for users but are facing automated online services abuse such as creating fake accounts, spamming, and conducting denial-of-service attacks. Consequently, in 2004, Von Ahn *et al.* designed a program called CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) [1] to prevent bot programs. In the following years, due to its simple generation, easy deployment, and effective defense, CAPTCHA has been rapidly and widely adopted in the Internet. CAPTCHA typically employs tasks that are challenging for machine programs but are easily solved by humans, such as identifying distorted characters, matching text and images, or moving sliders. Only successful completion of these challenges would grant access and operations. By leveraging this approach, CAPTCHAs effectively block a significant number of malicious bot attacks, safeguarding user data and website systems to a certain extent.

Based on the concept of distinguishing humans from machines using CAPTCHA, various types of CAPTCHAs emerge. They employ specific cognitive tasks for verification. Common types of CAPTCHAs include text-based CAPTCHA [1], [2], [3], [4], [5], image-based CAPTCHA [6], [7], [8], [9], audio-based CAPTCHA [10], [11], and slider-based CAPTCHA [8], [12]. Human's ability of reasoning and judgment can easily handle the challenges involving numbers, letters, images, audio and video recognition in CAPTCHA, while recognition capabilities of early machine algorithms drop behind [4].

In the early stage, the CAPTCHAs mentioned above played a positive role in defending against automated malicious attacks. Text-based CAPTCHA gained popularity as one of the most widely adopted types due to its simple implementation and easy deployment, favored by most online websites [4]. To enhance the security of text-based CAPTCHA, developers subsequently employed various security mechanisms, such as background noise, character rotation, overlap, and distortion [4], [5]. Image-based CAPTCHA also found widespread application due to simple user interaction and rich content [9], which can be categorized into selection-based, slider-based, and click-based types. Breaking image-based CAPTCHA requires developing programs to recognize image content, including object categories, locations, shapes, and so on. In recent years, the rise and development of deep learning technology pose significant challenges to traditional text-based and image-based CAPTCHA. Since each CAPTCHA can be associated with an AI (Artificial Intelligence) problem, they are susceptible to being cracked by attackers using machine learning [2], [3], [13], [14] and deep learning [9], [15], [16],

(Corresponding author: Haizhou Wang.)

Botao Xu, Haizhou Wang, Wenxian Wang, and Xingshu Chen are with the School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China (e-mail: z860578251@126.com; whzh.nc@scu.edu.cn; catean@scu.edu.cn; chenxsh@scu.edu.cn).

This work is partially supported by the Key Research and Development Program of Science and Technology Department of Sichuan Province under grant No.2023YFG0145 and the National Key Research and Development Program of China under grant No. 2022YFC3303101.

[17] techniques. Furthermore, the anti-recognition mechanisms employed by CAPTCHA developers to enhance security drastically degraded the user experience, yet failed to achieve expected security outcomes [18].

To address the aforementioned issues, Tencent[1], one of the most well-known CAPTCHA service providers from China, proposed the first visual reasoning CAPTCHA called VTT (Visual Turing Test) [19]. The task of this CAPTCHA presents an image containing multiple objects and asks users to select the correct object according to the question. Only by clicking on the specific region of the image can users pass the human-machine verification. Soon afterwards, other well-known CAPTCHA service providers, such as Geetest[2], NetEase[3], Shumei[4], Xiaodun[5], and Dingxiang[6], also introduced similar visual reasoning CAPTCHAs to combat bot programs, as shown in Fig. 1. The challenges in these CAPTCHAs typically involve examining the intrinsic properties of objects, such as shape, size, and color, while some questions also involve complex visual and spatial logic relationships, such as relative position, relative size, similar colors and shapes. This type of CAPTCHA is based on both image recognition and natural language processing, and it leverages logical reasoning to defend against attacks, achieving a good defensive effect [19]. The designers of VTT conducted attack experiments using a relational network to evaluate its security, achieving only a 4.7% success rate. As visual reasoning CAPTCHAs have emerged only in recent years, existing research on their security analysis is limited.

The task of visual reasoning CAPTCHA is essentially an instance of REC (Referring Expression Comprehension) [20], which requires identifying the target visual instance based on a given language expression. In REC, most works [21], [22], [23], [24] opt for a multi-step detection approach, first detecting salient regions from the image, and then selecting the most matching region through multimodal interaction. Among these, models like MAttNet [21] and UNINEXT [22] mainly rely on the visual backbone detectors such as Mask R-CNN [25] and DETR [26] within the models to complete the inference process. However, the language expressions in visual reasoning CAPTCHAs are more complex, manifested in the incorporation of intricate logical reasoning tasks within the CAPTCHA question. This adds complexity to existing REC models since they are inadequate in achieving satisfactory results on such CAPTCHA datasets.

Existing research on visual reasoning CAPTCHAs [27] has employed modular hierarchical network that integrates knowledge and rules to accomplish CAPTCHA cracking. However, the model's ability to learn object attributes and relative relationships remains inadequate. Furthermore, since the modular approach is not an end-to-end logical reasoning process, the performance of subsequent modules is greatly influenced by the results of previous modules. Simultaneously, training and coordinating multiple independent sub-module networks also lead to issues of increased complexity and computational overhead.

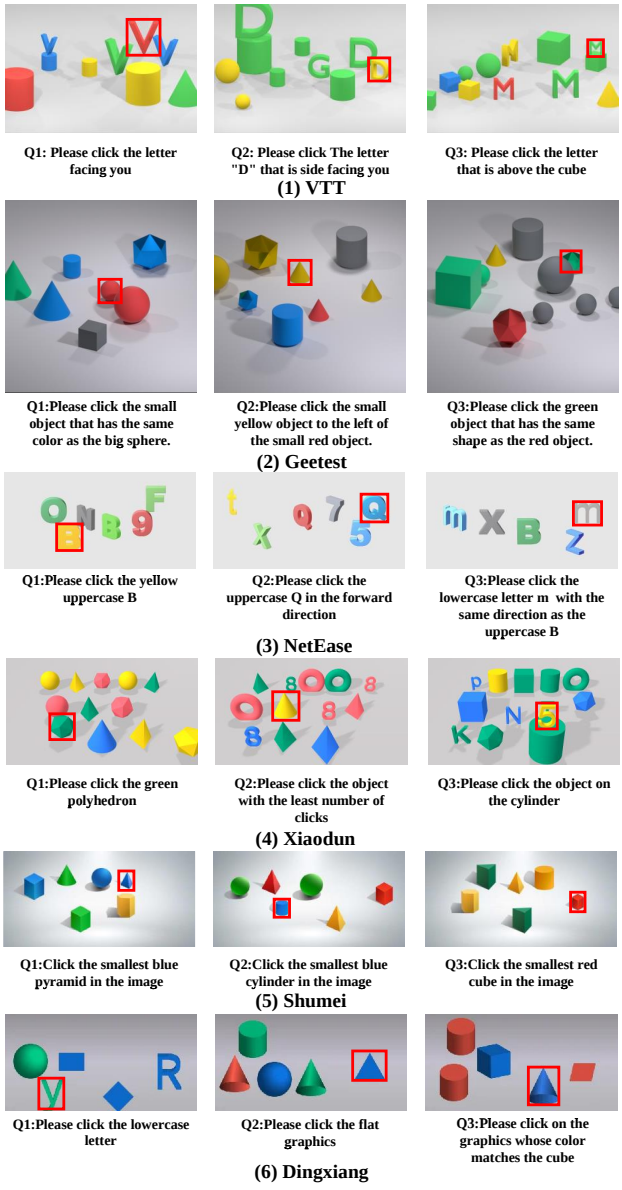


Fig. 1. Examples of visual reasoning CAPTCHAs.

Therefore, this paper proposes MF-GGNN (Multi-Feature Fusion-based Graph Gated Neural Network), a general, end-to-end reasoning framework based on GGNN that can effectively crack visual reasoning CAPTCHAs. The framework consists of an object detection module, a multi-feature extraction module, a question encoder based on attention network, and a graph reasoning model. First, we use a Mask R-CNN [25] based object detection method to identify all objects in the image and extract the detected object bounding boxes and labels. We then extract multi-features of the objects and fuse them with the question attention representation. These features are organized into six distinct categories: visual feature, relative visual feature, absolute position feature, relative position feature, label feature, and correlation feature. They collectively empower the model to thoroughly grasp and understand the

[1]https://cloud.tencent.com/product/captcha
[2]https://www.geetest.com/show
[3]https://dun.163.com/trial/space-inference
[4]https://www.ishumei.com/trial/captcha.html
[5]https://xiaodun.com/index.html
[6]https://www.dingxiang-inc.com/business/captcha

complex connections presented in the CAPTCHA cracking tasks. Finally, we construct a multi-step reasoning network based on graph nodes and edges, treating each detected object and its attributes as nodes, and relative relationships as edges in the graph neural network, to obtain predictions. Unlike existing method that relies on modular approach, our end-to-end framework architecture streamlines the learning process and enhances the ability of our model to capture complex dependencies and relationships within the objects in CAPTCHAs. However, current visual reasoning CAPTCHA attack research faces two fundamental bottlenecks: Firstly, the construction of visual reasoning CAPTCHA datasets faces challenges due to anti-crawling measures implemented by CAPTCHA service providers and the significant human resources required for annotation of high-quality. Secondly, the only existing research [27] on visual reasoning CAPTCHAs attack didn't release their constructed dataset, thereby greatly restricting further research in this field. Therefore, to validate the effectiveness of the framework, we construct and publish the first large-scale and comprehensive Visual Reasoning CAPTCHA dataset (ViRC) comprising CAPTCHAs from six distinct CAPTCHA providers, including VTT, Geetest, Xiaodun, NetEase, Shumei, and Dingxiang. The dataset contains more than 50,000 CAPTCHA images, with question lengths varying between 5 and 25 Chinese characters. It includes over 200 different types of objects, such as 3D and 2D shapes of uppercase and lowercase letters, numbers, geometric figures, and similar elements. We conduct experiments on CAPTCHAs including VTT, Geetest, Xiaodun, NetEase, Shumei, and Dingxiang that we collected, achieving ASR (Attack Success Rates) of 90.3%, 91.9%, 93.1%, 78.6%, 100.0%, and 99.2%, respectively. Additionally, we perform comparative experiments with our framework against advanced models [21], [27], [28], [29], [30], [48], [58], [59]. The results demonstrate that our method outperforms prior research in terms of ASR on VTT, Geetest, Xiaodun, Shumei, and Dingxiang, achieving SOTA (State-Of-The-Art) performance. On NetEase dataset, we also attained a relatively high success rate, with considerable improvement over our original framework. In summary, our main contributions can be divided into three aspects:

- **We construct and release the first large-scale labeled visual reasoning CAPTCHA dataset (ViRC).** ViRC comprises six different visual reasoning CAPTCHA sources, containing over 50,000 CAPTCHA images along with their corresponding questions. Our dataset is characterized not only by its broad coverage but also by its inclusion of a rich variety of object categories and logical reasoning challenges. This dataset will significantly contribute to the advancement of research in the field of visual reasoning CAPTCHA.
- **To the best of our knowledge, we are the first to propose a novel end-to-end multiple feature fusion-based framework for cracking visual reasoning CAPTCHA.** The framework employs a graph gated neural network model to address the visual reasoning task, and relies on the updating mechanism of GRU to propagate visual, semantic, and other information within the network. Compared to existing modular-based attack, our end-to-end framework offers significant advantages, which simplifies the training process by eliminating the need for separate module optimization, making the overall workflow more efficient. Additionally, the unified architecture enables faster cracking speeds, as the model processes CAPTCHA challenges in a single, streamlined step. These improvements collectively enhance the effectiveness of our attack model on multiple visual reasoning CAPTCHA schemes, achieving an average attack success rate of 92.2%. Multiple experimental results indicate the effectiveness and reasoning logicality of our model.
- **In order to improve the logical reasoning capability for multiple objects, we extract visual, relative visual, absolute position, relative position, question correlation, and label features specially for visual reasoning CAPTCHA.** By conducting a series of feature ablation experiments, we observed that the removal of any individual feature resulted in a significant decrease in the model's overall performance, with an average reduction of 2.73%, 3.47%, 2.8%, 3.57%, 3.23%, and 4.67%, respectively. Among them, the label feature contributes the most, while the visual feature contributes the least. By learning the visual and spatial position relationships of objects, the framework accomplishes the relevant reasoning tasks, and also visualizes the reasoning process.

We would also like to mention that a shorter conference version of this paper has been published in the *Proceedings of the 19th EAI International Conference on Security and Privacy in Communication Networks (SecureComm 2023)*. Our previous visual reasoning CAPTCHA dataset was collected from VTT, NetEase, Geetest, Shumei, and Xiaodun. In this journal version, we have expanded the dataset by including CAPTCHAs from Dingxiang, one of the largest CAPTCHA providers in China, to make our dataset more comprehensive. Additionally, we have further proposed feature extraction methods incorporating CAPTCHA question features that were not considered in the conference version, such as label feature and correlation feature. These new features enable the model to better understand the semantic relationships and logical dependencies embedded in the CAPTCHA questions. We further enhanced the graph reasoning model by augmenting the graph topology, where each node now includes representations of label features and correlation features. Moreover, we designed additional experiments, including CAPTCHA feature analysis, feature ablation study, and adversarial sample attack experiment, to further demonstrate the effectiveness of our proposed framework.

II. Related Work

Existing research on CAPTCHA security analysis can be categorized into many types, including text-based CAPTCHAs [1], [2], [3], [4], [5], [31], [32], [33], [34], image-based CAPTCHAs [6], [7], [8], [9], audio/video-based CAPTCHAs [10], [11], slider-based CAPTCHAs [8], [12], and visual

reasoning CAPTCHAs [19], [27]. We will introduce related defense and cracking works surrounding the three of the most popular CAPTCHA types: text-based, image-based, and visual reasoning CAPTCHAs.

A. Text-based CAPTCHAs

Text-based CAPTCHAs, being the first type of CAPTCHA proposed [1], are widely used due to their simple deployment and low generation cost. Users are required to provide the correct character sequence according to the text images in order. Text-based CAPTCHAs use text recognition as the fundamental task, but they have been evolving along with the progress of text recognition methods. Existing text-based CAPTCHAs employ various anti-recognition mechanisms to prevent malicious machine attacks: Amazon [17] uses rotated characters, Google's reCAPTCHA [31] uses distorted characters, and Baidu [35] uses multiple font styles. These CAPTCHAs increase the difficulty of cracking by altering character shapes. Apple [32] and Microsoft [3] respectively apply overlapping characters and two-layer text, using complex character structures to prevent machine segmentation and recognition. Platforms like Sina [2], Scihub [5], and Douban [33] adopt different approaches by not changing the character shapes or structures, but rather adding noise like dots, lines, and shadows to the background to interfere with recognition and reduce machine recognition accuracy. It is noteworthy that modern text-based CAPTCHAs no longer rely on a single defense mechanism, but instead employ multiple defensive measures simultaneously to increase the cracking difficulty.

While the defense mechanisms of text-based CAPTCHAs continue to evolve, this has not prevented text recognition techniques from cracking them. Substantial research has proposed various methods and models for cracking text-based CAPTCHAs: Gao *et al.* [3] used a color-filling segmentation algorithm to fill in hollow characters, then removed noise components and extraneous contour lines. Finally, they used segmentation and recognition algorithms to successfully crack multiple hollow CAPTCHAs. Directing at noise in text-based CAPTCHAs, Chen *et al.* [5] proposed multiple denoising methods based on spatial domain filters, Gibbs transforms, Hough transforms, and morphological operations. [14] systematically summarized works using machine learning and deep learning to break text-based CAPTCHAs, pointing out that text-based CAPTCHAs are no longer secure. Gao *et al.* [13] constructed a character recognition method based on Log-Gabor filters, achieving good success rates on multiple text-based CAPTCHA schemes. Their attack is simple, effective, and generalized.

In summary, attacks on text-based CAPTCHAs have become increasingly effective over time. There is currently very limited scope for further enhancing the defenses of text-based CAPTCHAs against these attacks. And their security is no longer sufficient enough to withstand existing attack methods [16], [36].

B. Image-based CAPTCHAs

Existing image-based CAPTCHAs are divided into click-based CAPTCHAs, slider-based CAPTCHAs, and selection-based CAPTCHAs [6], [7], [8], [9]. Compared to text-based CAPTCHAs that require manual user input of character strings, image-based CAPTCHAs provide a better user experience. Additionally, images contain richer content and information, hence they also have widespread applications.

Slider-based CAPTCHAs [8], [9], [12] require users to drag a slider to a specified position, and differentiate between humans and machines based on the mouse trajectory and accuracy of the slider placement. The process of cracking slider-based CAPTCHA primarily involves two steps: first, obtaining the slider and its target sliding position, and then generating a corresponding sliding trajectory based on the offset distance to the target position, using scripts to simulate mouse trajectory dragging. This type of CAPTCHA can be cracked by employing conventional target matching algorithms and generating sliding trajectories, lacking sufficient security. However, due to their low cost and high user-friendliness, slider-based CAPTCHAs still see widespread deployment.

Click-based CAPTCHAs [8], [9] present a prompt and a background image containing characters, requiring users to click on the characters in the image sequentially according to the prompt. They can be viewed as a simplified variation of text-based CAPTCHAs. The underlying AI task for click-based CAPTCHAs is similar to text-based CAPTCHAs, incorporating analogous anti-recognition mechanisms. However, with the advancement of character recognition techniques, the security of click-based CAPTCHAs has also been diminishing.

Selection-based CAPTCHAs require users to select images of specified object categories from a set of images based on prompts. The Asirra CAPTCHA [7] asks users to select images containing cats from 12 images. Selection-based CAPTCHAs like Google's reCAPTCHA[7] [8] and hCAPTCHA[8] are used on websites . Works such as [9], [37] achieved attack success rates of 79% and 83.25% respectively on the more difficult reCAPTCHA v2 using image detection and classification models. Given the powerful image object classification and recognition capabilities of existing deep learning models, it is easy to classify image objects, and selection-based CAPTCHAs have been proven to be no longer secure.

C. Visual Reasoning CAPTCHAs

Visual reasoning CAPTCHA is a new type of CAPTCHA that combines the richness of content found in image CAPTCHAs with the examination of semantic and logical information. The only existing work targeting visual reasoning CAPTCHA is by Gao's team [27], who proposed a model consisting of multiple modules including semantic parsing, detection, classification, and fusion module. The core process involves using image localization algorithms to identify key objects in the image, and extracting the visual attributes and features of these objects. Finally, the fusion module filters out irrelevant object attributes using the reasoning program, leaving the remaining object as the predicted answer. Ultimately, the model achieved attack success rates of 88.0%, 90.8%,

[7]https://developers.google.com/recaptcha/docs/display
[8]https://accounts.hcaptcha.com/demo

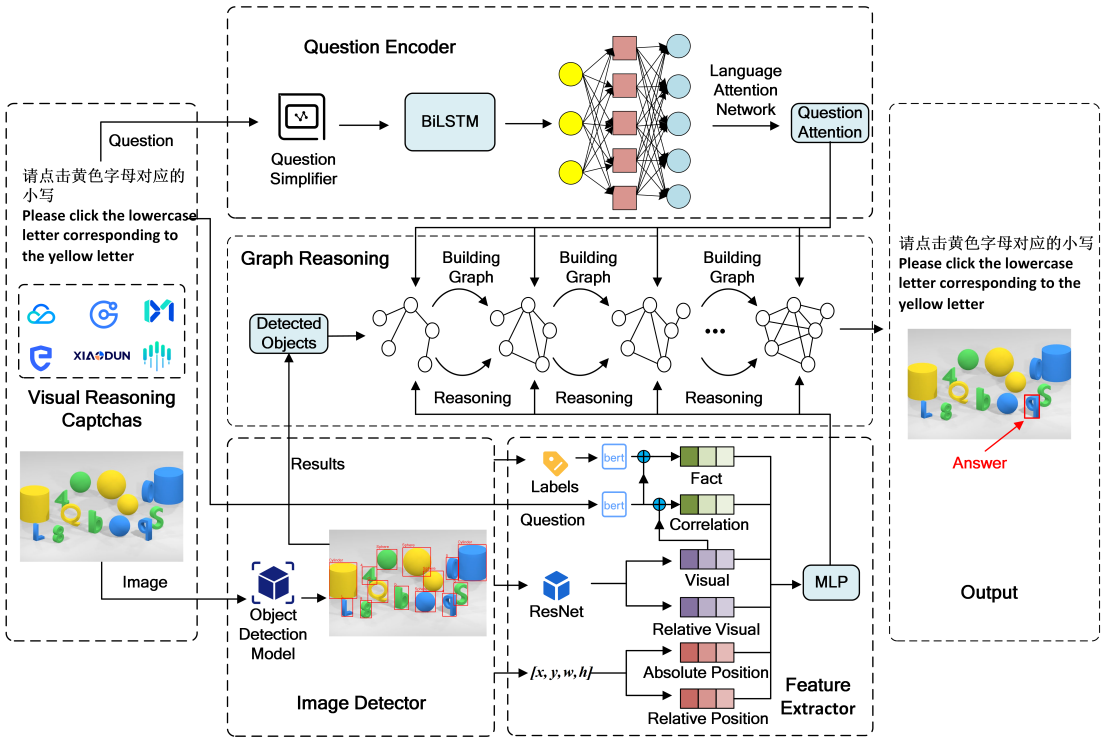


Fig. 2. Cracking framework for visual reasoning CAPTCHAs.

79.2%, 86.2%, 95.9%, and 98.6% on VTT, Geetest, Xiaodun, NetEase, Shumei, and Dingxiang respectively.

While the modular approach has achieved relatively high attack success rates, it still has some shortcomings. As visual reasoning CAPTCHAs continually increase and upgrade, the new measures employed lead to an increasing number of objects and more complex logical relationships within the CAPTCHAs. In the modular approach, the detection module may fail to identify some partially occluded objects, and their exclusion from the reasoning program leads to failures in logic reasoning. Additionally, due to the large number and variety of objects, the detection and classification modules do not sufficiently learn the image features extracted by the network. Consequently, the reasoning program employed by the model may filter out objects relevant to the question, hindering the learning of latent logical relationships. Finally, the modular approach requires separate training of modules. Assembly of these modules also imposes a heavy burden of manual labor.

To address these limitations, this paper proposes an end-to-end framework for cracking visual reasoning CAPTCHAs, employing GGNN to solve the visual reasoning task in CAPTCHAs for the first time. We design and extract features of CAPTCHA objects and question attention distributions to better learn the underlying logical relationships. The model propagates and fuses multi-features within the network through the updating mechanism of GRU, implementing successful attacks on multiple visual reasoning CAPTCHA schemes.

III. Method

In this paper, we view the visual reasoning CAPTCHA task as a retrieval problem: Given a CAPTCHA image I and a question Q, our goal is to obtain the object region R_{answer} specified by Q.

In this section, we propose a new attack framework for visual reasoning CAPTCHAs and detail the methods employed, as illustrated in Fig. 2. The attack framework is based on graph gated neural network and implements an end-to-end functionality, effectively cracking six most popular types of visual reasoning CAPTCHAs. The framework consists of four main components: an object detection module, multi-feature extraction, an attention network based question encoder, and a graph reasoning model. The object detection module is responsible for identifying all objects in the image and extracting object bounding boxes and labels. Multi-features, combined with the question attention distribution, are input into the graph reasoning model, constructing a multi-step reasoning network based on graph nodes and edges to obtain the final predicted answer. The detail of our proposed framework is introduced as follows.

A. Dataset Construction

1) Data Collection: Current research on visual reasoning CAPTCHAs is scarce, with only Gao *et al.* conducting relevant attack research [27]. However, they did not make their dataset public, which greatly limits further research in this field. To collect experimental data, we previously developed web crawlers targeting visual reasoning CAPTCHAs across multiple CAPTCHA platforms and collected a large number of visual reasoning CAPTCHA image-question pairs. Subsequently, we constructed the dataset and performed data annotation based on this collection.

To prevent large-scale malicious collection of CAPTCHAs, visual reasoning CAPTCHA service providers typically em-

TABLE I
COMPARISON WITH OTHER DATASET

Dimensions	Scale	Collection time	Diversity	Open-source availability
Gao *et al.*	38,500	2021	6	×
ViRC	54,000	2023-2024	6	✓

ploy strict anti-crawling measures, which impose certain limitations on CAPTCHA data collection. We implemented simulated browser dynamic clicking to circumvent these anti-crawling mechanisms and collect sufficient visual reasoning CAPTCHAs. Through web crawlers, we gathered a large number of visual reasoning CAPTCHA images and their corresponding question texts, storing them for subsequent use. Additionally, we implemented filtering algorithms to remove duplicate CAPTCHAs, ensuring the reasonability of the dataset.

Our previous visual reasoning CAPTCHA dataset was collected from VTT, NetEase, Geetest, Shumei, and Xiaodun. In this paper, we publish the ViRC dataset which additionally include Dingxiang CAPTCHAs. The sizes of VTT, Geetest, NetEase, Xiaodun, Shumei, and Dingxiang datasets are 10,000, 10,000, 10,000, 10,000, 300 and 10,000, respectively. Please note that the CAPTCHA data was collected between February 26, 2023 and November 17, 2023, and the dataset is open-sourced at [9].

2) Object Detection: Constructing a visual reasoning CAPTCHA dataset requires obtaining the target region corresponding to the question of the CAPTCHA. To reduce the training burden on the reasoning model and decrease the number of model's parameters, we did not adopt the typical approach of REC works [21], [22], [23], [24], which involves integrating the target region detection training process into the reasoning training. In reality, the task humans need to solve for CAPTCHAs is to identify all objects in the CAPTCHA image and find the unique target object among them. Therefore, we first train corresponding object detection models for different visual reasoning CAPTCHAs, and then use the detected object regions and labels as input for the subsequent reasoning model. The paper employs the Mask R-CNN model [25], a deep learning-based model capable of simultaneously recognizing and segmenting objects in images. Compared to other object detection models, Mask R-CNN offers more precise localization and segmentation, and is based on the computationally efficient Faster R-CNN [38]. Our Mask R-CNN model uses the pre-trained ResNet-50 network parameters on ImageNet as initial parameters for training, and generates separate model parameters for each CAPTCHA type. Furthermore, to ensure that we detect as many objects as possible in the CAPTCHA, we set a threshold of 0.7 based on our pilot experiments. Using this model, we detected various object categories in the visual reasoning CAPTCHAs, such as uppercase and lowercase letters, digits, 2D and 3D geometric objects. Finally, the object regions obtained from the CAPTCHAs are defined as R, and the labels are defined as L.

3) Data Annotation: For the data annotation task, we invited professional annotators to label the answers for the visual reasoning CAPTCHAs. We developed an online annotation website to facilitate the annotation process. This website displays one visual reasoning CAPTCHA image and its corresponding question on each page. The professional annotators click on the answer region within the CAPTCHA using a mouse, and the backend of the website records the mouse click coordinates relative to the entire image and stores them in a database. Finally, we exported the data and used scripts to determine which object in the CAPTCHA the coordinates belonged to. The object is then regarded as the answer for that particular CAPTCHA, thereby completing the annotation process. In total, we annotated over 50,000 CAPTCHAs from six most popular CAPTCHA service provider platforms.

4) Dataset Comparison: To demonstrate the advantages of our newly proposed dataset, we have conducted an extensive comparative analysis with Gao *et al.* [27]'s dataset across five critical dimensions: scale, collection time, diversity and open-source availability. The comparison results are presented in TABLE I. Our dataset exhibits two key superiorities: a) Larger scale with updated temporal coverage (2023-2024 vs. 2021); b) First publicly released collection containing both visual reasoning CAPTCHA annotations and source images. By making the dataset openly available upon request, we aim to foster transparency, collaboration, and reproducibility within the research community.

B. Multi-Feature Extraction

In this section, we propose a multiple feature fusion approach. The approach analyzes and extracts features related to the visual reasoning CAPTCHA images and questions. These features are categorized into six types: absolute position feature, relative position feature, visual feature, relative visual feature, and additionally, in contrast to [30], we introduce label feature and correlation feature.

Visual Feature f_{vi}: Existing works [39], [40] on the visual reasoning dataset CLEVR have shown that vectors extracted from the conv4 layer of the pre-trained ResNet101 network can effectively represent the visual feature of objects. A notable characteristic of visual reasoning CAPTCHA is the presence of numerous regions irrelevant to the question, i.e., redundant backgrounds. Therefore, unlike the approach of above works that extracts features from the entire image, our model crops the CAPTCHA image based on the object bounding boxes. The bounding boxes are returned by the object detection model from the previous section. After obtaining multiple object images, we extract the image feature f_{vi} for each object. For example, "请点击在红色物体左侧的大号黄色方块。 *(Please click on the large yellow square to the left of the red object.)*", the model must accurately identify and differentiate objects based on their visual properties (e.g., red object and large yellow square) to solve the task. By focusing on cropped object regions, our approach ensures that the extracted visual features are highly relevant to the question, minimizing the interference from irrelevant background information.

[9]https://github.com/yiyepianzhounc/ViRC

Position Feature f_{po}: The object detection model outputs object bounding boxes $r_n = [x, y, w, h]$, where x, y represent the top-left coordinates of the object bounding box, and w, h represent its width and height. Previous works [41], [42] have shown that an object's position and relative position features can effectively represent the relationships between objects, which aligns with the logical reasoning requirements for positional relationships in visual reasoning CAPTCHAs. Therefore, we define $f_{po} = [\frac{x}{W}, \frac{y}{H}, \frac{x+w}{W}, \frac{y+h}{H}, \frac{w \times h}{W \times H}]$, where W, H denote the width and height of the CAPTCHA. Our experimental results indicate that the first four features can learn the position of a target object within the CAPTCHA. The last feature $\frac{w \times h}{W \times H}$ represents the size of an object compared to the entire CAPTCHA image. This feature is particularly useful for solving questions that require size-based reasoning, such as identifying the smallest or largest object in the image. For example, in question "点击图中最小的绿色六棱柱 *(Click on the smallest green hexagonal prism)*" the model must determine which green hexagonal prism occupies the smallest proportion of the image area. By incorporating these position and size features, our approach ensures that the model can effectively handle a wide range of spatial and size-related reasoning tasks in visual reasoning CAPTCHAs.

Relative Position Feature f_{rp}: Some visual reasoning CAPTCHAs require users to accurately answer the relative position relationships between objects, e.g., "请点击在大号灰色方块前面的灰色正方体。 *(Please click on the gray cube in front of the large gray square.)*" from Geetest and "请点击圆柱体上的物体 *(Please click on the object on the cylinder)*" from Xiaodun. Therefore, we further extract the pairwise positional relationships between objects within each CAPTCHA as edge features f_{edge}, as shown in Eq. 3, based on the existing absolute position features. We adopt the polar coordinate representation proposed in [28] to describe the relative position features, as polar coordinates can represent the relationship of object positions more compactly using a combination of angles in Eq. 1 and distances in Eq. 2. Here, c_x, c_y denotes the center point of an object, θ represents the angular relationship, and ρ represents the distance relationship to characterize two objects i, j:

$$\theta = \frac{\arctan \frac{c_{y_j} - c_{y_i}}{c_{x_j} - c_{x_i}}}{\frac{\pi}{2}}, \tag{1}$$

$$\rho = \frac{\sqrt{(c_{x_j} - c_{x_i})^2 + (c_{y_j} - c_{y_i})^2}}{\sqrt{W^2 + H^2}}, \tag{2}$$

$$f_{edge} = [\theta, \rho]. \tag{3}$$

To fully represent the relative position feature, we define $f_{rp} = [f_{vi_i}, f_{edge}, f_{po_j}]$, which integrates the visual feature of object i, the edge feature between objects i and object j, and the absolute position feature of object j. This comprehensive representation enables the model to effectively reason about complex spatial relationships, such as "in front o", "on top of", or "next to" which are critical for solving visual reasoning CAPTCHAs.

Relative Visual Feature f_{rv}: Relative visual feature is used to process questions inquiring about objects with the same color, shape, or size, such as "请点击与倾斜的物体形状相同的物体 *(Please click on the object with the same shape as the tilted object)*" and "请点击数字*1*颜色一样的大写*O (Please click on the uppercase O with the same color as the digit 1)*". We define $f_{obj_i} = [f_{vi_i}, f_{po_i}, l_i]$ as the complete visual feature of object i, where f_{vi_i} captures the object's visual attributes (e.g., color and size), f_{po_i} encodes its positional information, and l_i represents its object shape. This will assist the model in more comprehensively utilizing relative visual features for reasoning. We define $f_{rv} = [f_{obj_i}, f_{obj_j}]$ to represent the relative visual features between object i and object j. By concatenating the complete visual features of both objects, the model can effectively analyze and compare their shared or distinct attributes. For instance, in the case of identifying objects with the same shape or color, f_{rv} allows the model to evaluate the similarity between the visual properties of the two objects.

Label Feature f_{la}: Through further analysis of the CAPTCHA, we find that the previous VRC-GraphNet [30] model lacks utilization of text features from the questions. The object detection model outputs predicted labels for the objects, and these label texts often directly correspond to key terms mentioned in the CAPTCHA questions. We categorize these textual correspondences as label feature of the CAPTCHA. Specifically, for a question like "点击图中最小的绿色长方体 *(Click on the smallest green cuboid)*", the object detection model will detect the cuboid objects in the image and output their labels, which precisely correspond to the label of "cuboid" required by the question. This alignment highlights the importance of incorporating textual information into the reasoning process. Therefore, in order to extract text features, we employ the state-of-the-art Chinese Bert-based feature extraction model to obtain the embedding vector v_{ques} for the entire question and the embedding vectors v_{label} for all predicted labels. For each object i, we define its label feature as $f_{la} = [v_{label_i}, v_{ques}]$, which combines the textual representation of the object's label with the semantic context of the question.

Correlation Feature f_{co}: After extracting the question features, we ponder whether there exists a feature that can simultaneously learn the visual features of the CAPTCHA and the question features, enabling a certain degree of correlation reasoning. Through subsequent experiments, we found that defining $f_{co} = [f_{vi_n}, v_{ques}]$, which associates the visual features and question features, can improve the reasoning accuracy of the model, where n represents the nth object. The reason is that the model can implicitly learn the visual attributes of an object while associating these attributes with the question during the reasoning process. For example, for the question "请点击与倾斜的物体形状相同的物体 *(Please click on the object with the same shape as the tilted object)*", the reasoning model discovers the tilted object based on visual features and associates this attribute with the question's requirement. This feature not only improves the ASR of model but also enhances its ability to handle complex CAPTCHA challenges that require simultaneous understanding of both visual and textual information.

C. Attention Network of Question Encoder

When reading questions, humans typically focus on the key information in the questions. Neural networks can effectively learn from this mechanism [43], which is why we introduce question attention in our neural network model. For visual reasoning CAPTCHA problems, we first preprocess the questions using a simplification algorithm, since most CAPTCHA questions contain redundant and irrelevant information such as "请点击 *(please click)*", "你 *(you)*", punctuation marks, etc. By preprocessing the input questions to the network, we reduce the number of words in a question, thereby reducing the complexity of the subsequent reasoning network. Meanwhile, some words in Chinese have the same semantics, so we unify them into the same word, such as "正方体", "立方体" and "方块", which are unified as "正方体 *(cube)*"; "左侧" and "左边", "左方" are unified as "左侧 *(left)*", etc.

After processing by the simplification algorithm, we establish a vocabulary table for the questions in each type of CAPTCHA. We use Jieba [10], a Chinese word segmentation library, to segment the Chinese questions, thereby establishing a mapping table from Chinese characters to vectors. We then map the question vectors $< w_1, w_2, ..., w_t >$ to numeric vectors $< n_1, n_2, ..., n_t >$ based on the vocabulary table. After processing through the word embedding layer and multilayer perceptron, we obtain word embeddings $embedding_t$, which are input into a BiLSTM network [44] for encoding. We extract the output of this network as the semantic information representation:

$$output_t, embedding_t = BiLSTM(embedding_t). \quad (4)$$

We perform a linear transformation on the semantic information representation of the embedding layer vector $embedding_t$, with an output dimension of 6, corresponding to the number of multimodal features we extract. The reasoning logic for CAPTCHAs on different platforms differs in emphasis. For example, VTT, Xiaodun, Geetest not only focus on visual features but also introduce positional features, while NetEase and Shumei do not consider positional relationships. Therefore, we then use the $Softmax$ function to map the features to probabilities. In this way, we obtain the probability weights corresponding to each feature, which can represent the contribution degree of different features to the reasoning result:

$$weights = Softmax(FC(embedding), 1). \quad (5)$$

Secondly, our model also benefit from the language attention network proposed in [45]. We introduce a trainable attention network that uses different vectors V_{feat}, where $\{vi, rv, po, rp, la, co\} \in feat$, to compute the relevance between the question word embeddings and each type of feature, i.e., the attention as shown in Eq. 6. While we also compute the attention distribution weights of the embedding layer in Eq. 7:

$$a_{feat,t} = \frac{exp(V_{feat} \cdot output_t)}{\sum_{k=1}^{T} exp(V_{feat} \cdot output_k)}, \quad (6)$$

[10]https://pypi.org/project/jieba

$$weight\_atten_{feat} = \sum_{t=1}^{T} a_{feat,t} \cdot embedding_t. \quad (7)$$

D. Graph Reasoning Model

We build a graph reasoning model based on the features of visual reasoning CAPTCHAs. The inputs to this model come from three parts: (a) The output of the object detection model; (b) The six types of features extracted from the multimodal feature extraction part; (c) The question attention distribution weights and the contribution weights of the features from the question attention network. We constructed a corresponding graph neural network to represent the relationships between objects in the CAPTCHA. After multiple reasoning steps, the network finally outputs the predicted object ID and finds its corresponding object detection bounding box as the final answer.

1) Graph Topology Construction: Previous works in image-text retrieval [50], [51] have shown that aligning image and text is helpful for capturing fine-grained relationships between visual and textual elements, which is crucial for tasks requiring cross-modal understanding. Therefore, we adopt graph neural network in the model. The graph topology structure aligns visual image objects with textual question tokens, enabling the model to jointly reason over visual and textual modalities. In our graph reasoning model, the nodes $V = \{v_i\}_{i=1}^{N}$ represent all the objects detected in the visual reasoning CAPTCHA by the object detection module, and the edges $E = \{e_{i,j}\}_{i=1,j=1}^{N}$ represent the relationships between pairs of objects, forming a directed graph neural network $G = \{V, E\}$. Each node v_i corresponds to an object o_i, and $e_{i,j}$ represents the relation between objects i and j. The node attributes include the extracted position feature f_{po}, visual feature f_{vi}, label feature f_{la}, and correlation feature f_{co}, while the edge attributes include the extracted relative position feature f_{rp} and relative visual feature f_{rv}.

2) Multi-Step Reasoning: Upon completing the construction of graph neural network, our initial step involves utilizing a multi-layer perceptron to meticulously process and refine the features extracted from the CAPTCHA. Inspired by the GGNN designed in [46], we incorporate a GRU-like update mechanism into the neural network. In this way, we facilitate a dynamic, multi-step reasoning process that enhances the network's ability to capture complex patterns and relationships:

$$f_{mm} = W_{mm}(W_{ques} weight\_atten_{feat} + W_{feat} f_{feat}), \quad (8)$$

$$A_i^{rel,n} = \tanh(f_{rel} h_i^{rel,n-1}), \quad (9)$$

$$h_i^{rel,n} = GRU(A_i^{rel}, h_i^{rel,n-1}), \quad (10)$$

$$h_i^{abs,n} = Softmax(W_{abs} \tanh(f_{abs})) h_i^{abs,n-1}, \quad (11)$$

where, rel contains attributes corresponding to relationship like relative position feature and relative visual feature; abs contains attributes corresponding to single object like visual feature, position feature and label feature and correlation feature; $W_{ques}, W_{feat}, W_{mm}$ are the fully-connected layer parameters for processing the question attention, features, and multimodal fusing features respectively. The fused multimodal

TABLE II
PARAMETER SELECTION FOR MF-GGNN.

Parameters			VTT	Geetest	NetEase	Dingxiang	Shumei	Xiaodun
Question Encoder	Word embedding size		128					
	BiLSTM	Number of layers	2					
		Hidden state size	256	256	256	128	128	256
		Dropout	0.5	0.1	0.2	0.1	0.1	0.2
Graph Reasoning	Node dim		1024					
	Edge dim		5					
	MLP	Number of layers	2					
		Hidden state size	1024	512	512	512	512	1024
		Activation function	ReLU					
Train	Batch size		50	50	50	50	6	50
	Learning rate		0.0005	0.0005	0.0005	0.0005	0.0001	0.0005
	Epoch		50					

features are represented by Eq. 8. Eq. 10 and Eq. 11 are the hidden state of object i after the nth reasoning step. The graph reasoning process is iterative, with each step refining the attention distribution and feature representations. This mimics human-like reasoning, where the focus shifts progressively from coarse to fine-grained details. After completing the multi-step reasoning, we use the final layer of the model hidden states to obtain the final predicted score for the answer object, as shown in the equation (12), note that $\{abs, rel\} \in feat$:

$$score = \sum \tanh(atten) \cdot tanh(h_i^{feat,n}). \tag{12}$$

Finally, we treat the ultimate question as a multi-class classification task, where we select the object with the highest probability out of all N objects in a CAPTCHA image. The probability of object i is represented as $p_i = \frac{\exp(score_i)}{\sum_{j=1}^{N} \exp(score_j)}$, and we use cross-entropy as the model's loss function in Eq. 13.

$$L = -\sum_{i=1}^{N} c_i \cdot \log(p_i). \tag{13}$$

Specifically, if object i is the answer to the visual reasoning CAPTCHA, $c_i = 1$, otherwise $c_i = 0$.

IV. EXPERIMENTS

A. Experiment Environment Settings and Parameters Selection

In our work, the experiments are conducted on an Intel(R) Xeon(R) E5-2680 v4 CPU, a 12GB TITAN X GPU, 15GB of memory, and a 12th Gen Intel(R) Core(TM) i7-12700H CPU, an 8GB NVIDIA GeForce RTX 3060 GPU, and 16GB of memory. All deep learning models are implemented using the Pytorch library.

For model parameter selection, due to the varying complexity of images and question logic across different types of CAPTCHAs, we trained separate models with tailored parameters for each type. The parameter selection process involved a combination of empirical testing and validation on a held-out dataset to ensure optimal performance. The final parameter settings for each CAPTCHA type are summarized in TABLE II. For the BiLSTM in question encoder, we adjusted the hidden state size and number of layers to effectively process the textual information in the questions, considering factors such as sentence length and vocabulary diversity. For the MLP in the Graph Reasoning process, we selected the number of hidden layers and hidden state size based on the complexity of the visual reasoning tasks, ensuring sufficient capacity to capture the relationships between objects. For example, the visual reasoning tasks in VTT and Xiaodun are more complex than those in other CAPCTHAs. Therefore, their hidden state sizes are larger than others.

B. Analysis of CAPTCHA Features

In this section, we explain in detail the characteristics of existing visual reasoning CAPTCHAs, conducting statistical analyses on the object categories, question lengths, and vocabulary related to question logic in the CAPTCHAs. We find that these visual reasoning CAPTCHAs have a large number of object categories. And the question lengths are generally longer than the expression lengths of datasets used for regular REC tasks. Furthermore, since the purpose of CAPTCHAs is to prevent recognition by machine programs, the logical reasoning involved is more complex, encompassing attributes such as relative position, distance, size, and others.

1) Category Analysis of CAPTCHA Objects: Visual reasoning CAPTCHAs contain many types of objects, which poses certain challenges for model reasoning. We separately count the number of object classes in CAPTCHAs dataset from the six platforms, as shown in Fig. 3. The legend shows different object classes, with the number in parentheses after each class indicating the number of subclasses it contains. The statistics reveal that the objects in CAPTCHAs can be mainly categorized into three types: geometric shapes, numbers, and letters. Among them, VTT, NetEase, Xiaodun, and Dingxiang contain a relatively large variety, with 53, 57, 46, and 45 classes, respectively. In contrast, Geetest and Shumei only contain geometric shapes, with a relatively smaller number of classes. Upon further subdivision, letters in CAPTCHAs include both uppercase and lowercase, resulting in more classes and posing a challenge for the model in terms of case recognition. In Dingxiang, we found that geometric objects can be further divided into 3D geometric shapes (e.g., spheres, cylinders, cubes) and 2D geometric shapes (e.g., rectangles, triangles, parallelograms), with 3D objects accounting for 52% and 2D objects accounting for 48%.

2) Length Analysis of CAPTCHA Questions: To analyze the impact of question length in visual reasoning CAPTCHAs

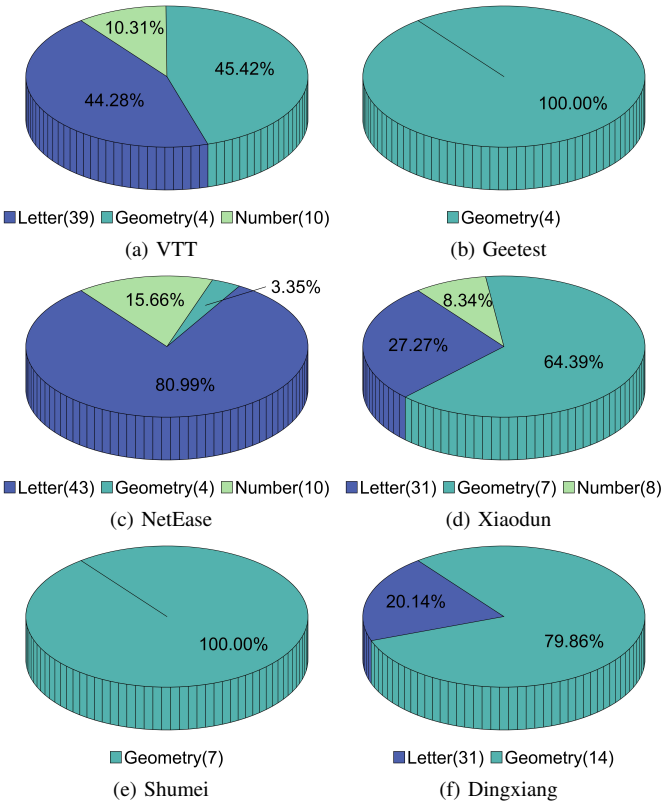


Fig. 3. Percentage of object catgories in visual reasoning CAPTCHAs.

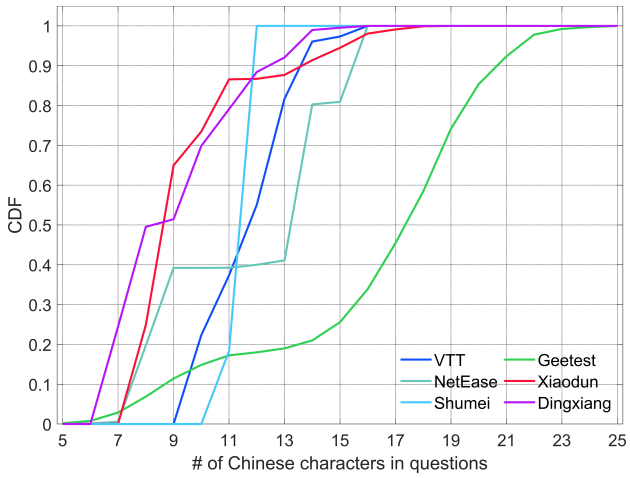


Fig. 4. Number of Chinese characters in questions of visual reasoning CAPTCHAs

on subsequent experiments, we count the number of Chinese characters in questions for the six platforms and draw a figure of CDF (Cumulative Distribution Function). From Fig. 4, we can see that except for Shumei where question lengths are concentrated around 11 and 12, the other CAPTCHAs contain questions of varying lengths. Specifically, for VTT and Dingxiang, question lengths are mainly between 10 and 14 characters, while for Xiaodun they are concentrated between 8 and 11 characters. The lengths for Geetest are generally longer than the other CAPTCHAs, whose curve shows an upward trend across all lengths and primarily ranging from 16 to 21 characters, with the longest reaching up to 25 characters.

Fig. 5. Word cloud of visual reasoning CAPTCHAs.

3) Word Frequency Analysis of CAPTCHA Questions: In order to better illustrate and showcase the features of the CAPTCHA data, we generate word clouds based on the part-of-speech tags for all the questions in the datasets. As shown in Fig. 5, the left image (a) represents word clouds for object nouns, where the most frequent word is "字母 *(letter)*", followed by geometric shapes and letter, number. The right image (b) represents word clouds for object attributes such as letter case, color, orientation, shape, size, etc., where the most frequent word is "相同 *(same)*", which corresponds to the frequent occurrence of '字母 *(letter)*" in the left image. From the analysis, we can learn that attacking visual reasoning CAPTCHAs requires model to have the ability of extracting key information from questions.

C. Comparisons with Other Approaches

In this section, we conducted experiments on six datasets and compared our method with previous works. The models adopted in our benchmark experiments can be categorized into two classes: a) One class comprises models similar to the tasks in visual reasoning CAPTCHA. Such as the one [28] achieving good performance on the VQA 2.0 dataset, LCGN [29] which exhibits outstanding performance on the CLEVR-ref dataset [47], and MAttNet [21], GroundingDINO [48] obtaining excellent results on real-scenario datasets like RefCOCO [49]. Given the recent rapid development of large language multimodal models, we have also expanded our comparisons to include cutting-edge approaches such as NExT-Chat [58] and GroundingGPT [59]. b) The other class includes models specifically designed for visual reasoning CAPTCHA, such as [27], [30] and our proposed model. The training datasets for all baseline methods, except for modular attack [27], are aligned with those used in our method. As for the work of [27], the code and dataset are not publicly available, making it difficult to reproduce their results. Therefore, we directly quoted their experimental results for comparison. The specific results are shown in TABLE III.

In our experiments, we split the datasets into training, validation, and test sets with a ratio of 8:1:1. Note that in line with the widely-used experimental design schemes [4], [27], we train separate models for each CAPTCHA platform due to the significant differences in image content and question logic across different CAPTCHA types. For instance, the visual reasoning tasks in VTT CAPTCHAs differ substantially from those in Geetest CAPTCHAs, making it necessary to tailor the training process to each platform's unique characteristics. This

TABLE III
ASR COMPARISON WITH OTHER APPROACHES

Model	VTT		Geetest		Xiaodun		NetEase		Shumei		Dingxiang	
	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test
LCGN [29]	10.8%	10.9%	11.5%	11.6%	16.8%	16.6%	16.9%	16.7%	13.3%	10.0%	15.6%	16.0%
VQA [28]	19.5%	19.1%	18.4%	18.6%	20.2%	20.3%	25.4%	25.0%	73.3%	70.0%	26.1%	26.0%
Modular [27]	N/A	88.0%	N/A	90.8%	N/A	79.2%	N/A	**86.2%**	N/A	95.9%	N/A	98.6%
MAttNet [21]	85.8%	88.0%	78.5%	76.9%	88.7%	88.4%	68.3%	65.5%	**100.0%**	**100.0%**	90.5%	90.2%
GroundingDINO [48]	N/A	24.3%	N/A	46.7%	N/A	24.0%	N/A	58.9%	N/A	66.7%	N/A	56.5%
NExT-Chat [58]	N/A	29.3%	N/A	42.7%	N/A	38.4%	N/A	24.5%	N/A	60.0%	N/A	67.1%
GroundingGPT [59]	N/A	42.1%	N/A	70.8%	N/A	50.4%	N/A	41.1%	N/A	76.7%	N/A	64.1%
VRC-GraphNet [30]*	88.8%	89.5%	77.9%	76.8%	89.4%	89.7%	71.9%	72.0%	96.7%	**100.0%**	92.6%	92.9%
Ours	**89.2%**	**90.3%**	**91.6%**	**91.9%**	**91.5%**	**93.1%**	**77.2%**	78.6%	**100.0%**	**100.0%**	**98.6%**	**99.2%**

* Noted that VRC-GraphNet is the model from our conference version of paper.

TABLE IV
EFFICIENCY AND COMPUTATIONAL ANALYSIS OF VARIOUS MODELS.
(TRT = TRAINING TIME, ACT = AVERAGE CRACKING TIME)

Model	VTT		Geetest		Xiaodun		NetEase		Shumei		Dingxiang		Parameter Scale
	TrT(s)	ACT(s)	TrT(s)	ACT(s)	TrT(s)	ACT(s)	TrT(s)	ACT(s)	TrT(s)	ACT(s)	TrT(s)	ACT(s)	
LCGN [29]	58.40	0.4783	81.80	0.3583	68.30	0.3223	96.30	0.4051	52.50	0.3891	95.10	0.3173	12M
VQA [28]	114.60	0.4791	142.50	0.3600	144.80	0.3224	95.90	0.4059	43.40	0.3896	54.90	0.3177	86M
Modular [27]	N/A	0.9600	N/A	0.8700	N/A	0.9700	N/A	0.8300	N/A	0.7900	N/A	0.7600	N/A
MAttNet [21]	1124.70	0.4990	1260.20	0.3760	1925.00	0.3550	1188.30	0.4570	412.10	0.4890	1463.30	0.3420	92M
GroundingDINO [48]	N/A	11.1000	N/A	9.4000	N/A	18.6000	N/A	13.8000	N/A	4.0000	N/A	8.4000	172M
NExT-Chat [58]	N/A	19.4000	N/A	14.9000	N/A	17.3000	N/A	12.8000	N/A	7.7000	N/A	15.7000	7B
GroundingGPT [59]	N/A	19.9000	N/A	15.9000	N/A	12.8000	N/A	10.2000	N/A	3.5000	N/A	6.0000	7B
Human [27]	N/A	9.1000	N/A	9.7000	N/A	10.2000	N/A	4.4000	N/A	9.8000	N/A	5.4000	N/A
VRC-GraphNet [30]	2462.40	0.4890	4183.85	0.3650	2015.73	0.3350	1125.87	0.4080	24.25	0.4160	1425.54	0.3220	13∼38M
Ours	2269.44	0.5050	6005.30	0.3770	3389.08	0.3340	999.60	0.4160	23.63	0.4220	1110.59	0.3200	16∼48M

design ensures that the addition of the Dingxiang CAPTCHA dataset in our journal version does not influence the training or evaluation of models for other CAPTCHAs, as each model is trained and tested exclusively on its corresponding dataset.

The results show that our model achieves higher attack success rates than other methods on VTT, Geetest, Xiaodun, Shumei, and Dingxiang, with significant improvements over our original model [30]. The performance improvements are solely the result of methodological advancements in our framework, such as the introduction of label features and correlation features, rather than changes in the datasets. Although we haven't obtained the optimal result on NetEase, we find that this may be related to the intrinsic characteristics of the NetEase CAPTCHA, as its performance is also relatively low on holistic models like MAttNet. CAPTCHA feature analysis in Subsection IV.B reveals that the NetEase dataset presents unique challenges due to its higher visual complexity and longer average question length. The NetEase dataset not only contains a wider variety of objects, which increases the difficulty of object detection and relationship extraction, but also features longer and more complex questions on average, requiring the model to process and reason over significantly more information. Ultimately, our attack demonstrates the significant security risks faced by visual reasoning CAPTCHA, as our model can achieve high attack success rates despite the use of complex AI tasks.

We also evaluated the computational efficiency of our model by measuring the average time required to crack each CAPTCHA. Our experiments show that the proposed framework achieves competitive efficiency, as shown in TABLE IV, with an average cracking time of 0.5050, 0.3770, 0.3340, 0.4160, 0.4220 and 0.3200 seconds per CAPTCHA for VTT, Geetest, Xiaodun, NetEase, Shumei and Dingxiang, respectively. Besides, the average cracking time of our model is significantly lower than the human response time [27] on visual reasoning CAPTCHAs, and outperform other visual reasoning research in [27]. Although some baseline models achieve faster training and cracking speeds, our model attains the highest ASR, which is the most critical metric for evaluating CAPTCHA-cracking performance. To further address the computational intensity, we conducted experiments to measure the parameter count of our model and compare it with baseline models. The results in TABLE IV indicate that our model is more lightweight than most baseline models, with a significantly lower parameter count. Generally, the average cracking time per CAPTCHA and computational resource size of our model remain within a reasonable range, fully meeting the requirements of real-world scenarios. This demonstrates that our framework not only maintains high accuracy but also meets the real-time requirements for practical applications.

TABLE V
FEATURE ABLATION STUDY ON ASR OF VARIOUS VISUAL REASONING CAPTCHAS
(VI = VISUAL FEATURE, RV = RELATIVE VISUAL FEATURE, PO = POSITION FEATURE, RP = RELATIVE POSITION FEATURE, LA = LABEL FEATURE, CO = CORRELATION FEATURE)

Method	VTT	Geetest	Xiaodun	NetEase	Shumei	Dingxiang
w/o VI	89.5% (↓0.8%)	91.6% (↓0.3%)	91.1% (↓2.0%)	73.5% (↓3.3%)	93.3% (↓6.7%)	95.9% (↓3.3%)
w/o RV	86.3% (↓4.0%)	91.7% (↓0.2%)	90.0% (↓3.1%)	67.1% (↓9.7%)	100.0% (↓0.0%)	95.4% (↓3.8%)
w/o PO	89.2% (↓1.1%)	91.5% (↓0.4%)	90.4% (↓2.7%)	68.6% (↓8.2%)	100.0% (↓0.0%)	94.8% (↓4.4%)
w/o RP	89.4% (↓0.9%)	83.6% (↓7.6%)	90.3% (↓2.8%)	70.0% (↓6.8%)	100.0% (↓0.0%)	95.9% (↓3.3%)
w/o LA	90.1% (↓0.2%)	87.7% (↓4.2%)	89.2% (↓3.9%)	68.2% (↓8.6%)	96.7% (↓3.3%)	91.4% (↓7.8%)
w/o CO	89.3% (↓1.0%)	91.4% (↓0.5%)	91.4% (↓1.7%)	67.2% (↓9.6%)	100.0% (↓0.0%)	92.6% (↓6.6%)
Full Model	**90.3%**	**91.9%**	**93.1%**	**78.6%**	**100.0%**	**99.2%**

D. Ablation Study

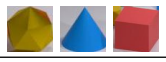
We then conduct ablation experiments to verify the contribution of each type of feature to the reasoning results. These features are proposed in the multi-feature extraction module and the experiment is measured by the CAPTCHA attack success rate. We sequentially remove visual feature, relative visual feature, position feature, relative position feature, label feature, and correlation feature, and compare them with the complete model. As shown in the TABLE V, after removing each of the six multimodal features, the attack success rate of the model decreases, indicating that all six features contribute to cracking visual reasoning CAPTCHAs. Through these ablation experiments, we observe that the removal of any individual feature resulted in a substantial decrease in the model's overall performance, with average reductions of 2.73%, 3.47%, 2.8%, 3.57%, 4.67%, and 3.23% for the visual feature, relative visual feature, position feature, relative position feature, label feature, and correlation feature, respectively. This clearly highlights the critical role each feature plays in enabling the model to learn the underlying logic of CAPTCHAs and perform accurate reasoning. Among these features, the label feature contributes the most, with the highest performance reduction of 4.67% when removed, while the visual feature contributes the least, with the smallest performance reduction of 2.73%.

E. ASR of Different Categories

In this experiment, we calculate the attack success rate of different categories in six kinds of visual reasoning CAPTCHAs, as shown in TABLE VI. We find that our model achieves the highest attack success rate on geometry among all visual reasoning CAPTCHA types, which all excceed 90%. The performance on letter is the worst, which may have something to do with the fact that categories of letter have more variations like uppercase and lowercase. Shumei achieves attack success rate of 100% since Shumei has only geometry objects and its reasoning logic is relatively simple. Although there are many categories with lots of attributes in VTT and Xiaodun such as color, size and direction, our model still achieves good performance, which demonstrates its ability to learn multi-modal and the logical reasoning behind the corresponding questions.

The table shows that for VTT and NetEase, the relative visual feature contributes the most, followed by position feature and correlation feature. This is related to the extensive visual logic assessment in their questions, such as color and orientation, where the model needs to associate the questions with the images. For Geetest and Xiaodun, relative position feature contributes more significantly, as their questions involve a considerable amount of relative position reasoning. Shumei is a special case, because its question logic is relatively simple, involving only visual assessment. Therefore, visual feature and label feature contribute more.

TABLE VI
PROPORTION AND ATTACK SUCCESS RATES OF DIFFERENT CATEGORIES IN VISUAL REASONING CAPTCHAS.

Platform	**Category**	**Examples**	**Proportion**	**ASR**
VTT	Letter		72.0%	87.80%
	Number		12.7%	93.70%
	Geometry		15.3%	99.34%
Geetest	Geometry		100.0 %	91.9%
NetEase	Letter		81.8%	77.02%
	Number		16.0%	85.00%
	Geometry		2.2%	90.91%
Xiaodun	Letter		27.7%	89.89%
	Number		8.3%	93.98%
	Geometry		64.0%	94.38%
Shumei	Geometry		100.0%	100.00%
Dingxiang	Letter		50.1%	98.80%
	Geometry		49.9%	99.60%

F. Influence of Train Dataset Size

In our work, the train dataset sizes for VTT, Geetest, Xiaodun, and Dingxiang visual reasoning CAPTCHAs are 8,000 CAPTCHA-question pairs each, while for NetEase and Shumei, they are 10,000 and 240, respectively. For the latter two, we increase the dataset size for NetEase due to its overall low attack success rate, while for Shumei, we could only obtain a total of 300 CAPTCHAs due to website anti-crawling restrictions.

For the purpose of investigating our model's performance with smaller train datasets, we train the model using different

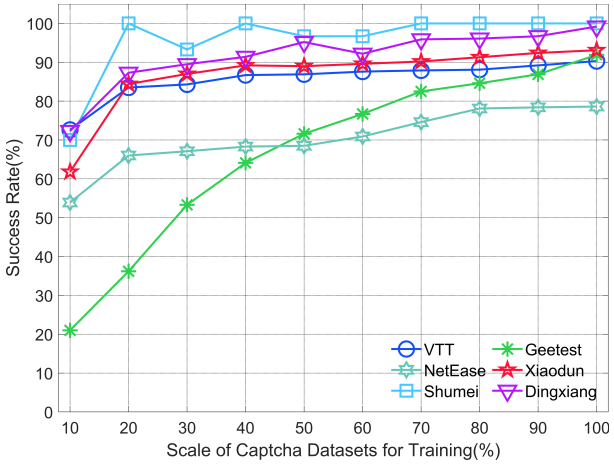


Fig. 6. ASR of model training with different dataset sizes.

fractions of the CAPTCHAs datasets. The attack success rates of the trained models are shown in Fig. 6. We can see that even when the train dataset size is only 30% of the original, the attack success rates for all CAPTCHAs exceed 50%. For Geetest, although the attack success rate is lower with a smaller train dataset, it increases rapidly with the dataset size and eventually exceeds 90%. The remaining CAPTCHAs also exhibit a general upward trend in accuracy as the train dataset size increases. Among them, VTT, Xiaodun, Dingxiang, and Shumei achieve over 80% attack success rates even when the dataset size is only 20% of the original, demonstrating that our model's reasoning ability can still achieve good results with limited training data.

G. Visualization of Model Reasoning

Furthermore, to better demonstrate the effectiveness of our model's reasoning on visual reasoning CAPTCHAs, we visualize the question attention distribution during the multi-step reasoning process, as shown in Fig. 7. Given each token in the question, by visualizing the model's output, we can see that the model not only localized the objects but also learned the objects' intrinsic attributes and relative relationships between objects, such as position, color, size, etc. Taking VTT as an example, in the first image, the model focuses on the region associated with the object "正方体 *(cube)*". In the second image, building upon the previous attention distribution, the model finds the region most strongly associated with the attribute "最小 *(smallest)*", which is the green cube. In the third image, the attention shifts to the region related to the attribute "颜色相同 *(same color)*", and finally, it outputs and localizes the object with the highest probability as the answer. The highlighted image regions change continuously during the multi-step reasoning process and are related to the question, indicating that our model can complete the logical reasoning required in the CAPTCHAs. Graph-based reasoning is particularly suitable for visual reasoning CAPTCHAs because it naturally models the relationships between objects in the image and the logical dependencies in the question. The theoretical foundation of our approach lies in the following principles: a) **Image-Question Alignment**: The attention

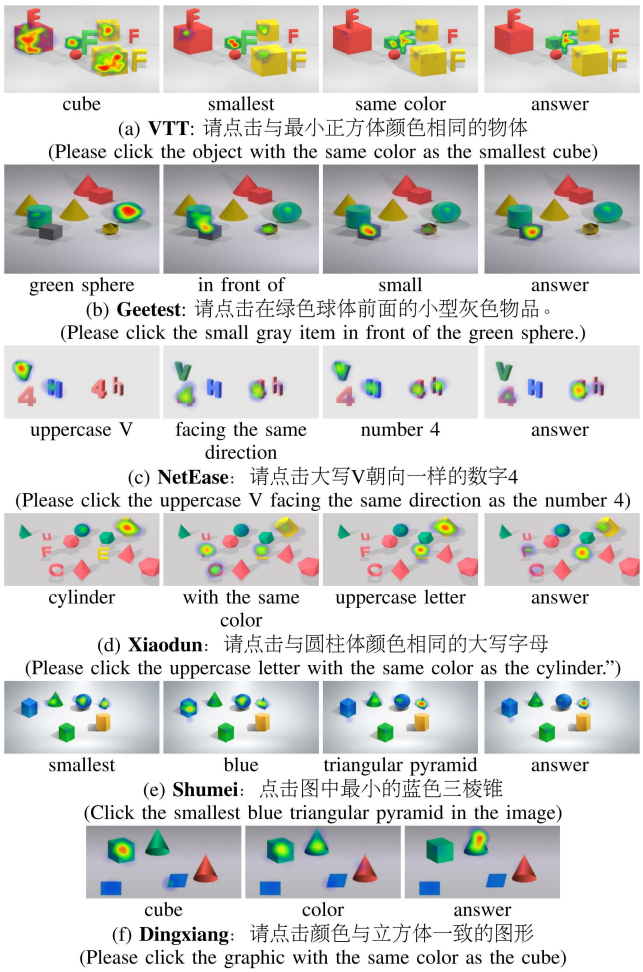
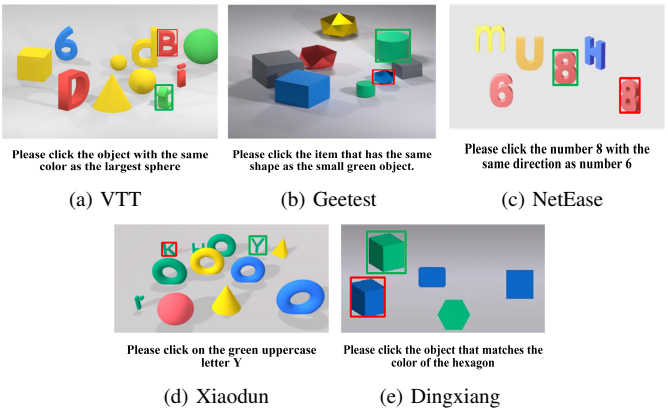


Fig. 7. Visualization of model reasoning ability.

mechanism dynamically aligns image regions with question tokens, ensuring the model associates visual objects with their described attributes. b) **Relational Reasoning**: The graph topology captures both absolute (e.g., position, color) and relative (e.g., "smallest", "same color") relationships, enabling multi-step reasoning through iterative updates of node and edge features. c) **Iterative Refinement**: The GRU-like update mechanism allows the model to refine its understanding over multiple steps, ensuring accurate and logical reasoning.

H. Limitation Analysis

To illustrate the availability of our method, we carry out the limitation analysis experiment on visual reasoning CAPTCHAs. Some unsuccessful attacks are shown in Fig. 8. Since we achieve 100% attack success rate on Shumei CAPTCHAs, there's no unsuccessful examples for Shumei. For VTT, Geetest, NetEase and Xiaodun, our model can precisely give the right answer even when the questions are relatively hard, involving colors, size, position, etc. This demonstrates that our model has comprehensive ability to understand questions for visual reasoning tasks. Although there is only geometry objects in Geetest, the questions involve many relative position and visual attributes which make the logical reasoning more comlex. The wrong prediction given by our model in NetEase is close to the answer. But the orientation of the number may confuse the model since the



(a) VTT (b) Geetest (c) NetEase

(d) Xiaodun (e) Dingxiang

Fig. 8. Our model's error predictions of visual reasoning CAPTCHAs.

difference between the two number "8" is little. In Xiaodun, however, the model fails on Q3 because of the similar shapes of two letters "K" and "Y". The model achieves best result on Shumei, which has something to do with the fact that geometry objects in Shumei are very simple and there are no occlusions.

I. Adversarial Sample Attack Analysis

To evaluate the robustness of our framework against adversarial perturbations, we conduct a series of adversarial attack experiments. Similar to methods adopted in [52], [53], our adversarial samples are generated by introducing controlled noise perturbations to the CAPTCHA images using primary Gaussian noise and salt-and-pepper noise. These methods simulate potential real-world scenarios where CAPTCHA images may be distorted or corrupted, challenging the model's ability to maintain high performance under such conditions. The experimental results in TABLE VII demonstrate that our framework maintains an overall high attack success rate even in the presence of adversarial noise. Although our model achieves an average ASR drop of 7.7% under adversarial sample attack, this reduction is still comparable to or better than several baseline models. Besides, we still maintain the highest ASR over five types of CAPTCHAs. This proves our model's robustness against adversarial samples, which can be attributed to the effective feature extraction and reasoning mechanisms in our framework.

V. Discussion and Future Work

First, despite the strong performance of our framework across various CAPTCHA platforms, we recognize that certain limitations remain, particularly in handling highly complex and adversarial scenarios such as those encountered in the NetEase dataset. The observed performance drop on this dataset can be attributed to its increased object diversity, longer and more complex questions, and the potential presence of inherent adversarial perturbations. Future work will focus on incorporating adversarial training to enhance resilience against noise and leveraging advanced feature extraction techniques such as multi-scale feature fusion and attention mechanisms to better capture subtle visual difference.

Second, our experiments reveal that the generalization ability of our framework across various CAPTCHA platforms is limited due to the significant differences in task requirements and characteristics among various CAPTCHA schemes. For instance, VTT CAPTCHAs typically involve fewer objects and simpler spatial relationships, while Xiaodun CAPTCHAs often contain more objects and require more intricate logical reasoning. Cross-platform transfer experiments, where a model trained on one platform (VTT) was tested on another platform (Xiaodun), demonstrated poor performance, with ASR of 8.4%, 9.4%, 7.3%, 14.3% and 15.7% for LCGN, VQA, MAttNet, VRC-GraphNet, and our MF-GGNN model, respectively. These results underscore the necessity of training separate models for each platform, as the visual and logical patterns learned from one platform do not generalize well to others. Future work will explore transfer learning techniques to improve cross-platform generalization and develop a more unified framework capable of adapting to multiple CAPTCHA schemes without requiring platform-specific training.

Third, we acknowledge the computational intensity of the proposed method and its potential impact on scalability in resource-constrained environments. While our current framework achieves competitive efficiency and maintains a lightweight parameter count compared to baseline models, we plan to further explore model lightweighting techniques in future work. This includes optimizing the graph reasoning module, investigating efficient neural architectures, and exploring pruning or quantization strategies to reduce computational overhead without compromising performance. These efforts will ensure that our framework remains scalable and practical for real-world deployment in diverse environments.

VI. Defense Suggestions

Based on previous experiments on attacking visual reasoning CAPTCHAs, we propose three suggestions for designing more secure visual reasoning CAPTCHAs.

- **Incorporating adversarial images**: Previous studies [34], [54], [55] have proposed adversarial attacks on

TABLE VII
VARIOUS MODELS' ASR OF CAPTCHAS UNDER ADVERSARIAL ATTACK.

Model	VTT	Geetest	Xiaodun	NetEase	Shumei	Dingxiang
LCGN [29]	7.8% (↓3.1%)	10.9% (↓0.7%)	15.2% (↓1.4%)	14.8% (↓1.9%)	10.0% (↓0%)	13.8% (↓2.2%)
VQA [28]	16.4% (↓2.7%)	17.7% (↓2.5%)	16.4% (↓3.9%)	21.4% (↓3.6%)	66.7% (↓3.3%)	25.7% (↓0.3%)
MAttNet [21]	68.3% (↓19.7%)	75.2% (↓1.7%)	77.5% (↓10.9%)	**73.4%** (↓12.8%)	93.3% (↓6.7%)	75.7% (↓14.5%)
GroundingDINO [48]	9.5% (↓14.8%)	40.5% (↓6.2%)	20.4% (↓3.6%)	40.2% (↓18.7%)	60.0% (↓6.7%)	48.4% (↓8.1%)
NExT-Chat [58]	22.3% (↓7.0%)	28.9% (↓13.8%)	23.7% (↓14.7%)	6.7% (↓17.8%)	46.7% (↓13.3%)	55.2% (↓11.9%)
GroundingGPT [59]	26.4% (↓15.7%)	55.3% (↓15.5%)	76.6% (↓13.1%)	23.2% (↓17.9%)	73.3% (↓3.4%)	63.2% (↓0.9%)
VRC-GraphNet [30]	**83.9%** (↓5.6%)	62.4% (↓14.4%)	76.2% (↓13.5%)	63.4% (↓8.6%)	93.3% (↓6.7%)	75.8% (↓17.1%)
Ours	83.1% (↓7.2%)	**79.9%** (↓12.0%)	**85.5%** (↓7.6%)	71.4% (↓7.2%)	**100%** (↓0.0%)	**86.4%** (↓12.2%)

images that can trick deep learning models into misclassification. Experiment on adversarial attack also indicates that adding certain perturbation can lead to ASR drop in CAPTCHAs. Therefore, it is a good idea to use adversarial techniques to introduce imperceptible changes to pixel values of CAPTCHA. These perturbations are designed to be invisible to humans but can mislead AI models. Designers can also add some background noises to disturb the recognition of objects in visual reasoning CAPTCHA images.

- **Using larger category set**: Experiment on the ASR of different categories has revealed that ASR of letter is generally lower than ASR of geometry. This indicates that a larger category set is helpful for preventing bots from developing specialized recognition capabilities for a limited set of CAPTCHA elements. To makes it harder for bots to bypass the CAPTCHA, designers can expand categories into Chinese characters or real objects such as cup, football, clothing and so on.
- **Adding more logical attributes**: From previous limitation analysis, we can see that VTT, Geetest and Xiaodun have more complicated logic reasoning demands compared to the rest of CAPTCHAs. As a consequence, the attack success rates of these types of CAPTCHAs are relatively lower. We suggest that adding more attributes such as orientation and relative position to visual reasoning CAPTCHAs, this can help prevent more deep learning model attacks.

VII. Conclusions

This paper introduces a groundbreaking end-to-end graph reasoning framework MF-GGNN that leverages the power of Graph Gated Neural Networks to tackle the challenge of visual reasoning CAPTCHAs. This paper begins by outlining the specific demands of visual reasoning CAPTCHA tasks, which often require complex understanding and inference beyond simple pattern recognition. and then adopt methods such as multi-feature fusion, question attention network, and graph gated neural network in the framework. Specifically, we use an object detection module to extract object bounding boxes and labels, and calculate the attention distribution and feature contribution weights output from the question encoding module based on six types of multi-features in the CAPTCHA image: visual, relative visual, absolute position, relative position, label, and correlation feature. We then move on constructing GGNN with object attributes as nodes and relationship attributes as edges, incorporating a GRU-like update mechanism to realize multi-step reasoning, and finally providing the object bounding box in the visual reasoning CAPTCHA as the answer. Our framework can achieve SOTA results on multiple visual reasoning CAPTCHAs from our dataset ViRC. Through detailed experiments, we not only validate the effectiveness of the MF-GGNN framework but also showcase its potential for generalization across multiple visual reasoning CAPTCHA tasks. Moreover, the framework's efficiency is highlighted by its ability to achieve robust performance even when trained on a relatively small dataset.

However, there still exist some limitations in our method. The performance of MF-GGNN on NetEase still has room to improve. In addition, ASR on large category set like letters is still unsatisfactory. In the future, we will consider building more effective network to adapt to larger category set. We will also explore adversarial training to improve the robustness of our model against adversarial attacks.

References

[1] L. von Ahn, M. Blum, N. Hopper, et al., "CAPTCHA: Using Hard AI Problems for Security". *Proceedings of the 22nd International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt)*, Warsaw, Poland, May 4-8, 2003, pp. 294-311.

[2] H. Gao, X. Wang, F. Cao, et al., "Robustness of Text-based Completely Automated Public Turing Test to Tell Computers and Humans Apart". *IET Information Security*, 2016, 10(1): 45-52.

[3] H. Gao, M. Tang, Y. Liu, et al., "Research on the Security of Microsoft's Two-layer CAPTCHA". *IEEE Transactions on Information Forensics and Security*, 2017, 12(7): 1671-1685.

[4] P. Wang, H. Gao, X. Guo, et al., "An Experimental Investigation of Text-based CAPTCHA Attacks and Their Robustness". *ACM Computing Surveys*, 2023, 55(9): 1-38.

[5] X. Xu, L. Liu, B. Li, "A Survey of CAPTCHA Technologies to Distinguish Between Human and Computer". *Neurocomputing*, 2020, 408: 292-307.

[6] R. Gossweiler, M. Kamvar, and S. Baluja, "What's up CAPTCHA?: a CAPTCHA Based on Image Orientation". *Proceedings of the 18th International Conference on World Wide Web (WWW)*, Madrid, Spain, April 20-24, 2009, pp. 841-850.

[7] J. Elson, J. R Douceur, J. Howell, et al., "Asirra: a CAPTCHA That Exploits Interest-aligned Manual Image Categorization". *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, Alexandria, Virginia, USA, October 28-31, 2007, pp. 366-374.

[8] Y. Zhang, H. Gao, G. Pei, et al., "A Survey of Research on CAPTCHA Designing and Breaking Techniques". *Proceedings of the 18th IEEE International Conference on Trust, Security and Privacy In Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, Rotorua, New Zealand, August 5-8, 2019, pp. 75-84.

[9] B. Zhao, H. Weng, S. Ji, et al., "Towards Evaluating the Security of Real-world Deployed Image CAPTCHAs". *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security (AISec)*, Toronto, Canada, October 15-19, 2018, pp. 85-96.

[10] H. Meutzner, S. Gupta, and D. Kolossa, "Constructing Secure Audio CAPTCHAs by Exploiting Differences Between Humans and Machines". *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI)*, Seoul, Korea, April 18-23, 2015, pp. 2335-2338.

[11] Y. Soupionis, and D. Gritzalis, "Audio CAPTCHA: Existing solutions assessment and a new implementation for VoIP telephony". *Computers & Security*, 2010, 29(5): 603-618.

[12] G. Chang, H. Gao, and Ge Pei, "The Robustness of Behavior-verification-based Slider CAPTCHAs". *Journal of Information Security and Applications*, 2024, 81: 103711.

[13] H. Gao, J. Yan, F. Cao, et al., "A Simple Generic Attack on Text CAPTCHAs". *Proceedings of the 23rd Network and Distributed System Security Symposium (NDSS)*, San Diego, California, USA, February 21-24, 2016.

[14] A. Dionysiou, and E. Athanasopoulos, "SoK: Machine vs. machine - A Systematic Classification of Automated Machine Learning-based CAPTCHA Solvers". *Computers & Security*, 2020, 97: 101947

[15] Y. Zi, H. Gao, Z. Cheng, et al., "An End-to-end Attack on Text CAPTCHAs". *IEEE Transactions on Information Forensics and Security*, 2019, 15: 753-766.

[16] M. Tang, H. Gao, Y. Zhang, et al., "Research on Deep Learning Techniques in Breaking Text-based CAPTCHAs and Designing Image-based CAPTCHA". *IEEE Transactions on Information Forensics and Security*, 2018, 13(10): 2522-2537.

[17] C. Li, X. Chen, H. Wang, et al., "End-to-End Attack on Text-based CAPTCHAs Based on Cycle-Consistent Generative Adversarial Network". *Neurocomputing*, 2021, 443: 223-236.

[18] E. Bursztein, S. Bethard, C. Fabry, et al., "How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation". *Proceedings of the 31st IEEE Symposium on Security and Privacy (S&P)*, Oakland, California, USA, May 16-19, 2010, pp. 399-413.

[19] H. Wang, F. Zheng, Z. Chen, et al., "A CAPTCHA Design Based on Visual Reasoning". *Proceedings of the 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, 2018, pp. 1967-1971.

[20] Y. Qiao, C. Deng, and Q. Wu, "Referring Expression Comprehension: A Survey of Methods and Datasets". *IEEE Transactions on Multimedia*, 2021, 23: 4426-4440.

[21] L. Yu, Z. Lin, X. Shen, et al., "MAttNet: Modular Attention Network for Referring Expression Comprehension". *Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, USA, June 18-22, 2018, pp. 1307-1315.

[22] B. Yan, Y. Jiang, J. Wu, et al., "Universal Instance Perception as Object Discovery and Retrieval". *Proceedings of the 33rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, Canada, June 20-22, 2023, pp. 15325-15336.

[23] Y. Zhou, R. Ji, G. Luo, et al., "A Real-Time Global Inference Network for One-Stage Referring Expression Comprehension". *IEEE Transactions on Neural Networks and Learning Systems*, 2023, 34(1): 134-143.

[24] G. Luo, Y. Zhou, X. Sun, et al., "Multi-Task Collaborative Network for Joint Referring Expression Comprehension and Segmentation". *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June 14-19, 2020, pp. 10031-10040.

[25] K. He, G. Gkioxari, P. Dollar, et al., "Mask R-CNN". *Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 22-29, 2017, pp. 2980-2988.

[26] N. Carion, F. Massa, G. Synnaeve, et al., "End-to-End Object Detection with Transformers". *Proceedings of the 16th European Conference on Computer Vision (ECCV)*, Glasgow, UK, August 23-28, 2020, 17, pp. 213-229.

[27] P. Wang, H. Gao, C. Xiao, et al., "Extended Research on the Security of Visual Reasoning CAPTCHA". *IEEE Transactions on Dependable and Secure Computing*, 2023, 20(6): 4976-4992.

[28] W. Norcliffe-Brown, S. Vafeias, and S. Parisot, "Learning Conditioned Graph Structures for Interpretable Visual Question Answering". *Advances in Neural Information Processing Systems*, 2018, 31.

[29] R. Hu, A. Rohrbach, T. Darrell, et al., "Language-conditioned Graph Networks for Relational Reasoning". *Proceedings of the 17th IEEE International Conference on Computer Vision (ICCV)*, Seoul, South Korea, October 27- November 2, 2019, pp. 10294-10303.

[30] B. Xu, and H. Wang, "VRC-GraphNet: A Graph Neural Network-based Reasoning Framework for Attacking Visual Reasoning CAPTCHAs". *Proceedings of the 19th EAI International Conference on Security and Privacy in Communication Networks (SecureComm)*, Hong Kong, China, October 19-21, 2023: 165-184.

[31] P. Baecher, N. Büscher, M. Fischlin. "Breaking reCAPTCHA: A Holistic Approach Via Shape Recognition". *Proceedings of the 38th International Conference on ICT Systems Security and Privacy Protection (SEC)*, Lucerne, Switzerland, June 7-9, 2011, pp. 56-67.

[32] P. Wang, H. Gao, X. Guo, et al., "A deep learning based attack on text CAPTCHAs by using object detection techniques". *IET Information Security*, 2022, 16(2): 97-110.

[33] P. Wang, H. Gao, Q. Rao, et al., "A Security Analysis of CAPTCHAs With Large Character Sets". *IEEE Transactions on Dependable and Secure Computing*, 2021, 18(6): 2953-2968.

[34] J. Zhang, J. Sang, K. Xu, et al, "Robust CAPTCHAs Towards Malicious OCR". *IEEE Transactions on Multimedia*, 2020, 23: 2575-2587.

[35] G. Ye, Z. Tang, D. Fang, et al., "Yet Another Text CAPTCHA Solver: A Generative Adversarial Network Based Approach". *Proceedings of the 25th ACM Conference on Computer and Communications Security (CCS)*, Toronto, Canada, October 15-19, 2018, 17, pp. 332-348.

[36] C. Shi, S. Ji, Q. Liu, et al., "Text CAPTCHA Is Dead? A Large Scale Deployment and Empirical Study". *Proceedings of the 27th ACM Conference on Computer and Communications Security (CCS)*, USA, November 9-13, 2020, 16, pp. 1391-1406.

[37] M. Hossen, Y. Tu, M. Rabby, et al., "An Object Detection Based Solver for Google's Image reCAPTCHA v2". *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, San Sebastian, Spain, October 14-16, 2020, pp. 269-284.

[38] S. Ren, K. He, R. Girshick, et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(6): 1137-1149.

[39] E. Perez, F. Strub, H. Vries, et al., "Film: Visual Reasoning with a General Conditioning Layer". *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana, USA, February 2-7, 2018, 32(1).

[40] R. Saqur, and K. Narasimhan, "Multimodal Graph Networks for Compositional Generalization in Visual Question Answering". *Advances in Neural Information Processing Systems*, 2020, 33: 3070-3081.

[41] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-aware Graph Attention Network for Visual Question Answering". *Proceedings of the 17th IEEE International Conference on Computer Vision (ICCV)*, Seoul, South Korea, October 27- November 2, 2019, pp. 10313-10322.

[42] H. Zhang, Z. Kyaw, S. Chang, et al., "Visual Translation Embedding Network for Visual Relation Detection". *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA, July 21-26, 2017, pp. 5532-5540.

[43] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention is All you Need". *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, Long Beach, California, USA, December 4-9, 2017, pp. 30.

[44] M. Schuster, and K. K. Paliwal, "Bidirectional Recurrent Neural Networks". *IEEE Transactions on Signal Processing*, 1997, 45(11): 2673-2681.

[45] K. Cho, B. Merriënboer, C. Gulcehre, et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October 25-29, 2014, pp. 1724-1734.

[46] D. Beck, G. Haffari, and T. Cohn, "Graph-to-Sequence Learning Using Gated Graph Neural Networks". *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia, July 15-20, 2018, pp. 273-283.

[47] R. Liu, C. Liu, Y. Bai, et al., "CLEVR-Ref+: Diagnosing Visual Reasoning With Referring Expressions". *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, USA, June 16-20, 2019, pp. 4185-4194.

[48] S. Liu, Z. Zeng, T. Ren, et al., "Grounding DINO: Marrying DINO with Grounded Pre-training for Open-set Object Detection". *Proceedings of the 18th European Conference on Computer Vision (ECCV)*, Milan, Italy, September 29- October 4, 2024, pp. 38-55.

[49] L. Yu, P. Poirson, S. Yang, et al., "Modeling Context in Referring Expressions". *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, October 11-14, 2016 pp. 69-85.

[50] J. Guo, M. Wang, Y. Zhou, et al., "HGAN: Hierarchical Graph Alignment Network for Image-Text Retrieval", *IEEE Transactions on Multimedia*, 2023, 25: 9189-9202.

[51] Y. Xu, Y. Bin, J. Wei, et al., "Align and Retrieve: Composition and Decomposition Learning in Image Retrieval With Text Feedback", *IEEE Transactions on Multimedia*, 2024, 26: 9936-9948.

[52] C. Shi, X. Xu, S. Ji, et al., "Adversarial CAPTCHAs". *IEEE Transactions on Cybernetics*, 2022, 52(7): 6095-6108.

[53] R. Shao, Z. Shi, J. Yi, et al., "Robust Text CAPTCHAs Using Adversarial Examples". *Proceedings of the 10th IEEE International Conference on Big Data (BigData)*, Washington DC, USA, December 15-18, 2022, pp. 1495-1504.

[54] H. Yuan, Q. Chu, F. Zhu, et al., "AutoMA: Towards Automatic Model Augmentation for Transferable Adversarial Attacks". *IEEE Transactions on Multimedia*, 2023, 25: 203-213.

[55] L. Gao, Z. Huang, J. Song, et al., "Push & Pull: Transferable Adversarial Examples With Attentive Attack". *IEEE Transactions on Multimedia*, 2022, 24: 2329-2338.

[56] J. Johnson, B. Hariharan, and V. Maaten, et al., "Inferring and Executing Programs for Visual Reasoning". *Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 22-29, 2017, pp. 2989-2998.

[57] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks". *Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, USA, June 18-22, 2018, pp. 7132-7141.

[58] A. Zhang, Y. Yao, W. Ji, et al., "NExT-Chat: An LMM for Chat, Detection and Segmentation". *ArXiv preprint*, arXiv:2311.04498, 2023.

[59] Z. Li, Q. Xu, D. Zhang, et al., "GroundingGPT: Language Enhanced Multi-modal Grounding Model". *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Bangkok, Thailand, December 15-18, 2024, pp. 6657-6678.