Automated Testing Method for Text-to-Image Software

Siqi Gu
Nanjing University
China

ABSTRACT

Recently, creative generative artificial intelligence software has emerged as a pivotal assistant, enabling users to generate content and seek inspiration rapidly. Text-to-image (T2I) software, being one of the most widely used among them, is used to synthesize images with simple text input by engaging in a cross-modal process. However, T2I software often encounters defects and erroneous, including omitting focal entities, low image realism, and mismatched text-image information. The cross-modal nature of T2I software makes it challenging for testing methods to detect defects. Lacking test oracles further increases the complexity of testing.

To address this deficiency, we propose **ACTesting**, an **A**utomated **C**ross-modal **Testing** Method of Text-to-Image software, the first testing method designed specifically for T2I software. We construct test samples based on entities and relationship triples following the fundamental principle of maintaining consistency in the semantic information to overcome the cross-modal matching challenges. To address the issue of testing oracle scarcity, we first design the metamorphic relation for T2I software and implement three types of mutation operators guided by adaptability density. In the experiment, we conduct ACTesting on four widely-used T2I software. The results show that ACTesting can generate error-revealing tests, reducing the text-image consistency by up to 20% compared with the baseline. We also conduct the ablation study that effectively showcases the efficacy of each mutation operator based on the proposed metamorphic relation.

CCS CONCEPTS

• **Software and its engineering → Software testing and debugging**.

KEYWORDS

software testing, cross-modal, text-to-image

ACM Reference Format:
Siqi Gu. 2024. Automated Testing Method for Text-to-Image Software. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Deep learning has undergone significant evolution. The rise in popularity of transformer [12], generative adversarial network (GAN) [61] and diffusion model [40] empower Artificial Intelligence Generative Content (AIGC) to yield surprising results in fields such as image, text, and audio generation within simple inputs. The emergence of large-scale models precipitates a rise in creative generative tasks. Text-to-image (T2I) task, being one of the most popular among them, aims to automatically synthesize a creative image based on the simple input text as shown in Figure 1. Recently, several T2I models [19, 24, 27, 28, 36, 40, 41, 59] and software [1, 2, 11, 32, 34, 44] from top IT companies show great performance on fidelity and creativity of the output image. The high-quality content and the simple interaction flow make increasing use of T2I software in our daily lives. Representative applications include generating the image for cross-modal data augmentation [50] (Computer Science), artistic creation and designing [53] (Art) and multilingual communication assistant for deaf [25] (Medical). The ability to handle cross-modal information and understand rich scenarios greatly enhances T2I software's potential for development.

Figure 1: Sample inputs and outputs of T2I software

However, the outputs from T2I software are not entirely reliable or capable of meeting the expected requirements. Despite diverse techniques that have been researched and adapted to improve the inner engine of T2I software, the generative results could be abnormal or incorrect [22, 52]. This is due to the huge and complex neural network structure, labeling errors in training datasets and uneven feature distribution. Generating unpleasant, offensive or inappropriate image content may result in significant repercussions [51]. Objectionable or discriminatory content can lead to brand damage

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

in a business setting and even trigger public protests or legal action. Consequently, it is significant to test the generation quality and robustness of T2I software.

Effective testing methods can swiftly detect the defects in T2I software, helping measure its robustness. However, developing testing methods for T2I software presents persistent challenges [5, 16, 48]. Initially, in contrast to conventional software with delineated internal logic, a significant segment of commercial T2I software offers only API interfaces to end-users, obscuring their internal engine and foundational models. This makes white-box testing methods no longer suitable for this task. Secondly, the most intricate issue is cross-modal alignment, necessitating aligning two modes with completely different information representations during the T2I software's testing phase. Traditional robustness testing methods, including boundary value analysis, adversarial attack, and anomaly detection, are also not suitable for the cross-modal generation task because of the lack of testing oracles. In addition, classic test methods based on text augmentation such as swapping the order of the words or introducing spelling errors are possible to cause errors in semantic information or affect the quality of the output images. Lastly, the tasks of quantifying the software's generation robustness and defining anomalies in the testing process are both crucial and demanding. To the best of our knowledge, there are no testing methods specifically designed for T2I software available.

To tackle the challenges mentioned above, we propose ACTesting, an automated and black-box cross-modal testing method for T2I software. ACTesting is designed based on the metamorphic testing theory to address the issue of oracle scarcity and cross-modal semantic matching. Specifically, it designs the metamorphic relation based on the fundamental principle of maintaining consistency in the semantic information across different modalities. To keep the semantic preservation, we apply the entity-relationship (ER) triple to represent the focal semantic information in both modalities. Then, we implement three types of mutation operators based on the ER triple to detect the defects of tested T2I software. This method aims to test the generation robustness of such software for consistently high quality and cross-modal alignment output.

We conduct experiments on four widely-used T2I software to evaluate the effectiveness of ACTesting. Tested software generates 113,736 synthetic images based on the MS-COCO validation dataset referring [13]. We employ I-FID, I-IS and RP ([6, 13, 42, 54]) as the basic evaluation metrics for image realism and text-image relevance. We also design the error-detecting ($Error_e$ and $Error_r$) and miss-detection ($Miss_e$ and $Miss_r$) rates as the metrics, based on the designed metamorphic relation [56] and scene graph generation model [47]. The experimental results demonstrate that ACTesting can reduce image quality by 2.9% to 15% and decrease text-image match consistency by 7.5% to 21.1%. The average error rate of the three operators is around 60%, which is 1.75 times higher than that of the baseline text mutation operator. Moreover, we conduct the ablation study to further elucidate the effectiveness of each operator. Not only can the operators be flexibly combined, but the results also show that these combined operators effectively increase the miss-detection rate beyond that of the basic operators.

The main contributions of this paper are as follows:

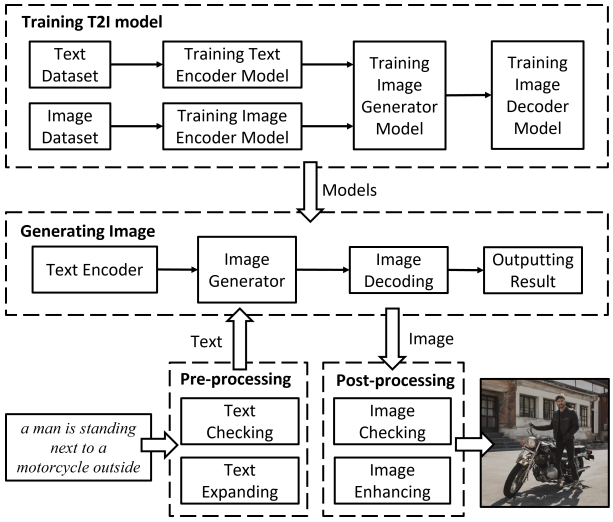

Figure 2: The workflow of modern T2I software

- **Method**. We introduce the *first* automated cross-modal testing method for T2I software, termed ACTesting. This method keeps the core idea of cross-modal semantic preservation and employs the entity-relationship triple to design the metamorphic relation. We design three kinds of mutation operators to detect the erroneous and test the generation robustness of T2I software under the guidance of adaptability density.
- **Tool**. We integrate the aforementioned method, ACTesting, into a Python tool. This represents the first black-box automated testing tool for T2I. We make the code for the entire process available on GitHub[1].
- **Study**. We conduct a comprehensive experiment to evaluate the performance of four industrial T2I software utilizing our ACTesting. The test findings further illustrate that our testing procedure can adeptly produce error-revealing test cases, leveraging our adaptability density-guided operators. The ablation study further shows that the proposed mutation operator can be flexibly combined for enhanced effectiveness.

2 PRELIMINARIES

In this section, we begin by outlining the workflow of modern Text-to-Image (T2I) software. We then detail the techniques employed by current T2I software, introducing how they effectively convert text into images. Finally, we present the motivation behind our proposal. This introduction sets the stage for a deeper exploration of our innovative contributions to testing T2I software.

2.1 T2I Software Workflow

Figure 2 delineates the general pipeline of modern T2I software. Leveraging expansive training image-text datasets, text encoder and image encoder models undergo pre-training to extract the cross-modal feature and establish mappings within a latent space. After that, the image generator and decoder models are trained end-to-end for image information reconstitution. T2I software then

[1]https://anonymous.4open.science/r/ACTesting-9478/

deploys these trained models (the text encoder, image generator, and image decoder) for the image synthesis procedure. For more flexible adjustment of input and output, T2I software designs pre-processing and post-processing stages to verify data quality. The pre-processing stage not only ensures text input conforms to predefined standards but also augments more details to the input text to fit the trained models. The initial image is generated based on the processed text input. In the terminal phase, post-processing assesses the initial image for both fidelity and ethical considerations, further employing image enhancement algorithms to mitigate noise and fulfill details. The core part of the whole workflow is the trained T2I model, thus we briefly introduce some modern engines in the next section.

2.2 Modern T2I Engine

With the rapid development of this technique, we introduce two main series of popular T2I models: Autoregressive and Diffusion-based models.

Autoregressive Model. Autoregressive methods can exploit large-scale datasets through time-series models, predicated on the dependence of the historical time-series of forecasting targets in different periods. Representative method DALL-E [39] from OpenAI, which regards both text and image tokens as sequential data, performing autoregression via Transformer architecture. Another model Parti [57] from Google encodes images as sequences of discrete tokens and reconstructs these sequences into high-quality images. However autoregressive methods nature need substantial computation resources.

Diffusion-Based Model. The most predominant approach is diffusion-based methods, which are also the new state-of-the-art models in T2I generation. Diffusion models (DMs) aim to reserve a process of perturbing the data with noise for sample generation. As a milestone work, Stable diffusion [40] trains the diffusion models within latent space, incorporating the text tokens during the denoising phase to fuse the cross-model features. DALL-E2 [38] applies CLIP [37] to learn and match the multimodal representations, evidencing unparalleled efficacy. Predominantly, contemporary T2I software engines are grounded in diffusion models.

2.3 Motivation

As depicted in Figure 3, we present examples generated by prevalent T2I software. In the left part (Figure a), we employ four variations of the sentence as inputs. Although the input text conveys nearly identical meanings, the output images exhibit variations in missing components. Subfigures 1 and 3 omit the "man", while subfigure 2 lacks the "bird". Only subfigure 4 accurately represents the correct entities ("man", "bird", "stone") and relationships ("watching", "standing on"). Moreover, in the right part (Figure b), we note that with an increase in the number of entities and relationships, the realism of the images diminishes. There are further omissions evident in both entities (subfigure 5) and relationships (subfigure 6).

Therefore, we consider that T2I software exhibits high sensitivity to the focal ERs, where even simple alterations in sentence structure could potentially lead to a reduction in text-image consistency. This

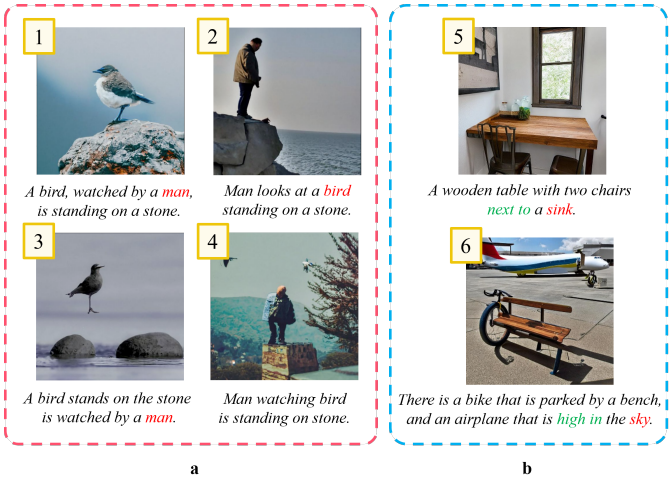

Figure 3: Examples of the input-output pairs of text-to-image software. All the images are generated with OpenAI.DALL-E

implies that the design of metamorphic relations based on cross-modal semantic preservation should also be approached from the perspectives of entities and relationships.

3 APPROACH

This section introduces ACTesting, which is proposed to test T2I software automatically and evaluate the generation robustness of the software in a black-box scenario. Figure 4 presents the overview of ACTesting. ACTesting can generate new input text based on the seed text by applying transformation operators. Rather than utilizing common text augmentation methods, ACTesting implements adaptability density constraint to design the operators of replacing, removing, and augmenting the ER triple. ACTesting designs metamorphic relation to testing the generation robustness based on the image entity and space relationship detection method.

3.1 Adaptability Density-guided Mutation Operators

We note that not all words in the input text have an impact on the quality of the results generated by T2I software. Therefore, traditional text mutation operators are not effective in thoroughly testing the defects in T2I software. As we mentioned in Sec 2.3, mutation operators should be highly relevant to focal ERs across two modalities. The problem we need to address is how to guide and design these mutation operators effectively and then generate the testing samples.

In the T2I software testing phase, items can be characterized by a set of exercises M. Testers use a certain measurement R to test each exercise $\mu \in M$. The performance of software π under an exercise μ can be denoted as $R(\pi, \mu)$. Because the measurements often need to be made multiple times, it is usual to apply the expected value of the performance of π as $E[R(\pi, \mu)]$. As the T2I software structure is complicated, M may contain more than one problem. The metrics Φ most commonly used to describe aggregated results are:

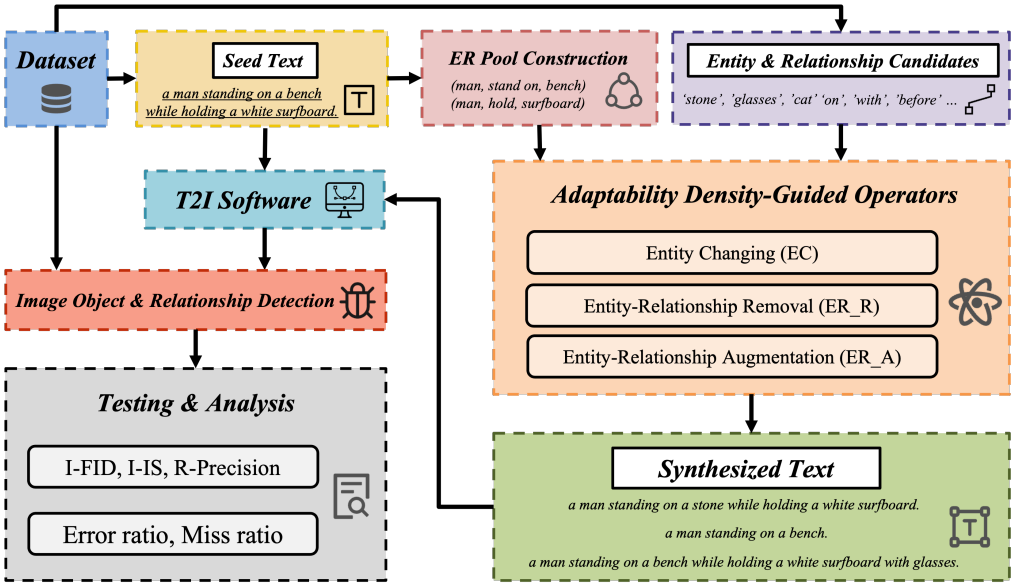

Figure 4: The Overview of ACTesting

$$\Phi(\pi, M, p) = \sum_{\mu \in M} \Phi(\pi, \mu, p) = \sum_{\mu \in M} p(\mu) \bullet \mathbb{E}[R(\pi, \mu)] \quad (1)$$

where $p(\mu)$ is the probability of μ. T2I software π is large and complicated enough that we clearly cannot test π on the whole set M. Random sampling with p is a reasonable choice. If M and p define the benchmark, is probability-proportional sampling on p the best way to test software? The answer is no, in general. Because the set M can include very easy and very challenging problems simultaneously (e.g. text summarization, text understanding, question answering, sentiment analysis, relation extraction, image generation, image super-resolution, and image composition). R can also be nondeterministic and/or subject to measurement error. Therefore, the probability-proportional sampling can be inefficient and costly. The better idea to approximate Eq.1 is to sample in a more purposeful way.

Our testing objective is to test the generation robustness of the T2I software. Robustness denotes the ability of software to consistently operate across diverse situations and the performance of the software to abnormal inputs and conditions. Essentially, we seek to understand the software's adaptability density with respect to varied test samples. In the example of Figure 3, the fourth image illustrates the highest adaptability density compared with others because it successfully synthesizes all the entities ("man", "bird", "stone") and relationships ("watching", "standing on").

Therefore, to better optimize the sampling method in the testing phase, we need to define the adaptability density constraint $d : M \rightarrow \mathbb{R}^+$. In the most ideal state, for each π being tested, $\Phi(\pi, M_1, p) > \Phi(\pi, M_2, p)$ if $d(\mu_1) < d(\mu_2)$. That is, the higher the adaptability density is, the better the evaluation performance is. As a result, we formulate the adaptability density-driven sampling method as follows:

$$\Phi(\pi, M, h) = \sum_{\mu \in M} \Phi(\pi, M, h) = \sum_{\mu \in M} h(\mu) \bullet p(\mu | h(\mu)) \bullet \mathbb{E}[R(\pi, \mu)] \quad (2)$$

where $h(\mu)$ is the probability of μ. For better instantiation and a more flexible condition, we discretize h. The adaptability density constraint is finally improved to that for each π being tested and two adaptability density levels a and b ($a \leq b$), we can get that $\Phi(\pi, M_a, p) > \Phi(\pi, M_b, p)$ (where $M_a = \mu | \mu \in M, d(\mu) = a$).

Typically, π remains a black box to end-users, obscuring the exact techniques it employs. However, it is unequivocally necessary for this task to align text and images, implying a requirement for consistency in entities and their interrelationships, albeit represented differently. Given these considerations, we define the adaptability density constraint corresponding to the matching degree of semantic information in two modalities.

However, it is difficult to directly calculate the adaptability density of T2I software based on the semantic information. The contrastive language-image pre-training (CLIP) [37] model is not suitable for this purpose, as it is highly vulnerable to attacks and lacks sufficient interpretability to provide further explanations to testers for anomalies. Furthermore, using CLIP as a step in the testing process is unfair because some T2I software (e.g. DALLE) utilizes this model as a pre-trained model. Therefore, to better represent semantic information in cross-modal data, and given that the input text for T2I software is mostly scene descriptions, which is highly analogous to scene understanding in the field of computer vision, we adopt the ER model to represent semantic information.

As shown in Figure 1, we deliver four output images corresponding to two input texts. The salient entities in each image are consistent with the input texts, marked in red (e.g. "man", "motorcycle", "cat"). If the input text describes the relationship between focal entities, marked in green (e.g. "standing next to", "and", "sleep on"),

the generated image should describe as well. Each entity and relationship has its *class*. Based on the entity and relation extraction technique in nature language processing, each text can be constructed to an entity-relation pool (ER pool), containing several entity-relation triples (e.g. ("dog", "with", "cat"), ("dog", "on", "bed"), ("cat", "on", "bed")). All elements in the ER pool will be converted to the class they belong to (e.g. "and" convert to "with", "sleep on" convert to "on", "man" convert to "person") referring to [9, 58]. The focus of the paper does not include the analysis of the singular and plural nouns.

The absence of test oracles in the T2I software makes it necessary to apply the metamorphic testing method. Metamorphic testing is introduced as a solution to the test oracle problem in test case generation [8]. It enables the generation of transformed tests from successful ones and mitigates potential issues arising from the lack of test oracles by defining the concept of Metamorphic Relations (MR). MR represents an essential characteristic of the target function or algorithm concerning various inputs and their corresponding expected outputs. Hence, the breach of MR serves as an indicator of potential software defects [43]. Establishing a suitable MR forms the foundation for conducting metamorphic testing.

Formally, we denote the T2I software as f_{T2I}, the input text in the seed set $\mathbb{S} = \{s_i\}$. Due to the unique nature of generative tasks, we do not use the existing label. We define the testing method in ACTesting as f_{er}, which is used to measure the degree of cross-modal semantic information matching. The results can be formalized as follows:

$$r_{s_i} = f_{er}(f_{T2I}(s_i)) \tag{3}$$

Then we define the transformation operators f_m to generate a set of new text $\mathbb{A} = \{a_j\}$. The mutation results (generated image) can be formalized as follows:

$$r_{a_j} = f_{er}(f_{T2I}(f_m(s_i))) \tag{4}$$

The testing domain is defined for each generation result as follows:

$$\mathbb{M}_{s_i} = \{m_j \| d(n_{s_i}, m_j) < \epsilon\} \tag{5}$$

where ϵ is a parameter for measuring strategy. Finally, the MR to test the T2I software with the newly generated sample can be formalized as follows:

$$\forall s_i \in \mathbb{S} \wedge \forall a_j \in \mathbb{A}, r_{a_j} \in \mathbb{M}_{s_i} \tag{6}$$

Different from the traditional text mutation operators, ACTesting is aimed at performing robustness testing for T2I software. As mentioned in Sec 2.3, we observe that when the same sentence undergoes a change in a single entity, there are differences in the generated quality exhibited by the T2I software. Additionally, as the number of entities and relationships in the sentence increases, the software's processing capabilities also show some difference. Therefore, based on the adaptability density constraint we introduced, three kinds of transformation operators are designed. We do not directly mutate the input text; instead, we first construct each text into an ER pool. Figure 5 presents an example of implementation for mutation operators of ACTesting.

Entity Changing (EC): The EC operator substitutes one entity in the ER pool, thus constructing a pair of highly similar input text samples. The output image should ideally remain consistent, aside

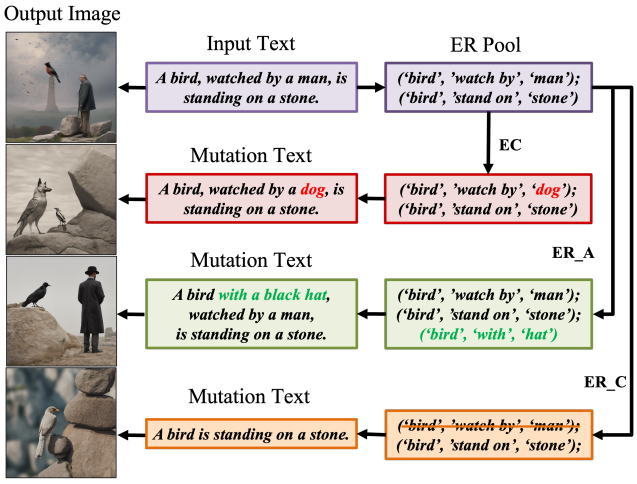

Figure 5: The mutation operators of ACTesting

from the substituted entity after generation. EC operator is primarily employed to ascertain whether T2I software demonstrates variability in generating content for diverse entities. In essence, the EC operator assesses the adaptability of T2I software, examining whether it excessively relies on pre-existing ER in the training dataset, consequently resulting in constrained generation capabilities. EC operator focuses on the issue of generation stability within the context of robustness testing.

Entity-Relationship Removal (ER_R): The ER_R operator randomly removes an ER triple. Removing one ER triple does not lead to unreadable sentences or semantic errors, which sets it apart from traditional text mutation strategies. The generation of images should ensure that the retained entities and relationships remain unaffected and maintain consistent image quality. ER_R operator aims at constructing a pair of text samples with differing levels of complexity, used to investigate whether the T2I models exhibit significant disparities in generating these different samples, which is also the reflection of the aspect of generating robustness.

Entity-Relationship Augmentation (ER_A): The ER_A operator randomly adds an entity and its corresponding relationship from candidate entities and relationships to the original ER pool. We construct the candidate ERs from the dataset. After obtaining the new ER pool, we use it as a basis to rebuild the input text. By increasing the variation of the original text input, ACTesting can explore the search space more extensively to uncover potential issues. ER_A operator aims to increase the complexity of the original samples, expanding the input space to test the robustness of T2I software.

In summary, the design of the three mutation operators is aimed at detecting defects in T2I software, based on mutation relationships that maintain consistency with cross-modal semantic information.

3.2 ACTesting Workflow

To ensure the adaptability of density-guided mutation operators can test the robustness of T2I software, ACTesting also encompasses the construction of a complete testing process. As shown in

Figure.4, ACTesting includes several main processing components: ER pool construction, adaptability density-guided operators, text synthesizing, image entity and relationship detection, and testing and evaluation results calculation. We select representative seed texts from the dataset that contain diverse ERs at the beginning. Then, ACTesting constructs an ER pool for each seed text based on the principle of semantic information preservation.

ER Pool Construction. The input text in the T2I software is different from other natural language processing tasks like text Classification or text abstract. The text often contains the scene or the objects of the generated image, which renders traditional text mutation operators no longer applicable to inputs for this task. Such as swapping the order of the words or introducing spelling errors in the text. Considering the different ways in which various modalities express information, we extract each input text as an ER pool, utilizing consistent cross-modal semantic information features. Subsequently, all mutation operations are performed based on the ER pool, ensuring that the semantic features of the input text remain intact. We adopt the relationship extraction, entity recognition and structured representation technique to construct the ER pool referring [35]. The ER candidates are also construed by this method based on the open-source dataset.

Mutation Operator Implementation As introduced in Sec 3.1, three kinds of mutation operators are aimed at testing the generation robustness of T2I software through cross-modal alignment. ACTesting adopts the EC and ER_A operators to all the inputs and adopts the ER_R operator to the input having more than one ER. The transformation operators of ACTesting are shown in Fig.5.

Text Synthesizing. After completing the mutation operations based on the adaptability density-guided operators, ACTesting constructs the synthesized text based on the mutated ER pools. To ensure the fluency of synthesized sentences and the integrity of semantic information, we define templates for generating sentence structures and utilize BERT [12] to enhance the details of the sentences. At this point, we have obtained paired text inputs.

Image object and relationship detection. We send the paired text inputs to the tested T2I software, respectively, and generate the output images. To address the challenge of cross-modal semantic matching, we employ the object detection model to identify entities in images and use the scene understanding model to detect relationships between entities in images. In the end, we forward the detected results to the next component.

Testing and Analysis. We consider that the degree of matching between images and text is fundamentally based on the matching degree of entities and relationships. Based on the selection of previous mutation operators, different MRs are used to compute test results. The formalized MR is introduced in the Sec 3.1. Specifically, we design MRs for each mutation operator as follows:

$$K_e = Set_e(r_{s_i})) \cap Set_e(r_{a_i})) \tag{7}$$

$$K_r = Set_r(r_{s_i})) \cap Set_r(r_{a_i})) \tag{8}$$

where $Set_e(r_{s_i}))$ and $Set_e(r_{a_i}))$ denote the set of entity class in r_{s_i} and r_{a_i}, $Set_r(r_{s_i}))$ and $Set_r(r_{a_i}))$ denote the set of relationship class in r_{s_i} and r_{a_i}.

MR1 is designed for the EC operator:

$$K_e == Set_e(r_{s_i})) - Set_e(e_1) == Set_e(r_{a_i})) - Set_e(e_2) \tag{9}$$

$$K_r == Set_r(r_{s_i})) == Set_r(r_{a_i})) \tag{10}$$

MR2 is designed for the ER_R operator:

$$K_e == Set_e(r_{a_i})) \tag{11}$$

$$K_r == Set_r(r_{a_i})) \tag{12}$$

MR3 is designed for the ER_A operator:

$$K_e == Set_e(r_{s_i})) \tag{13}$$

$$K_r == Set_r(r_{s_i})) \tag{14}$$

If any of the K_e or K_r in each MR are violated under a certain operator, the generated results will be separately reported as either entity errors (p_e) or relationship errors (p_r).

In summary, we detail the specific implementation of metamorphic relationships tailored for testing T2I software. We will delve into the specific evaluation metrics related to that in the following section. ACTesting encompasses a comprehensive process of test case generation and test result computation, where each module plays a crucial role in the overall effectiveness of the testing.

4 EXPERIMENT DESIGN

Our evaluation was designed to answer the three main research questions in the experiments:

RQ1: Can ACTesting generate error-revealing tests for T2I software under the adaptability density-guided?

RQ2: Can three kinds of transformation operators effectively detect the defects of T2I software?

RQ3: How does the combination of different transformation operators impact the robustness of the tested T2I software?

4.1 Datasets and T2I Software

In this section, we introduce the dataset and software we implement in our experiments.

Dataset. We mainly use the MS-COCO [7] as the seed sets, which is a widely recognized benchmark in the field of computer vision. It is introduced to address the need for rich annotations including labels of object class, object instances, object locations and the relationships to other objects. Importantly, the dataset stands out for its image captioning component, where each image is accompanied by at least five different captions provided by human annotators. This feature makes MS-COCO highly valuable at the intersection of vision and natural language processing.

T2I Software. To elucidate the effectiveness of the testing methods we propose, we implement four Text-to-Image software in the experiment. We employ the method of calling APIs to alleviate the potential bias and ensure the black-box nature of the experimental process. We briefly introduce this software as follows:

OpenJourney [2] As one of the most popular text-to-image software, OpenJourney is painting waves in the world of ai-generated art. It is often quoted as a free alternative to MidJourney as it is a Stable Diffusion model trained using thousands of Midjourney images from its v4 update.

[2]https://www.openjourney.art/

Wan Xiang [3] from Alibaba Cloud platform, is an evolving AI painting model that can create corresponding images or artworks from textual descriptions using machine learning and natural language processing technologies. This model is based on Alibaba's proprietary combinatorial generative model, Composer [23].

Stable Diffusion XL (SD XL) [4] from Stability.ai platform, help users create descriptive images with shorter prompts and generate words within images. The model is a significant advancement in image generation capabilities, offering enhanced image composition and face generation that results in stunning visuals and realistic aesthetics.

DeepAI Image Generator [5] from DeepAI platform, creates an image from scratch from a text description. It can be used to generate AI art or for general silliness. It provides the functions of AI image generation and AI image edition API call services for developers.

4.2 Experimental Setup

Implementing steps. We conduct testing experiments on 4062 seeds of text-image pairs from the MS-COCO validation dataset. We define 150 categories of entities and 50 kinds of relationships based on the distribution of the dataset. After that, we implement three adaptability density-guided operators (EC, ER_R, ER_A) to generate three mutation testing sets. To demonstrate the effectiveness of our testing methodology, we have selected a commonly used text mutation operator, random synonym substitution (SS), as our baseline. Furthermore, to validate the effectiveness of each of the three operators, we integrated EC with the other two operators (EC+ER_R, EC+ER_A), resulting in two types of compounded mutant test inputs. Lastly, we generate the synthetic images based on T2I software for all the test sets. In summary, we get 4 software * 4062 captions * 7 test sets = 113,736 synthetic images. All images are resized to 512×512. We calculate evaluation metrics based on all inputs and outputs and subsequently obtain the results.

Running environment. All experiments are performed on a Ubuntu 20.04.6 LTS server with RTX 3090 GPU. We implement ACTesting on Python 3.7. We experiment with our methods on four widely-used T2I software based on the public dataset MS-COCO [7], and the targets of the testing process are black box software. We use the NLP Package Stanza [6] powered by Stanford NLP Group and Bert [12] to analyze the input text based on the Pytorch Framework.

4.3 Evaluation Metrics

Firstly, we follow the convention of the T2I model research community and employ several metrics. We adopt the Improved-Fréchet Inception Distance (I-FID), Improved-Inception Score (I-IS), and R-Precision (RP) based on [6], [42], [54] and [13] for evaluating the image realism and text relevance.

To compute the image realism metrics (I-FID and I-IS), we initially extract features using a pre-trained Inception-v3 network [45].To solve the overfitting problem, we adapt to calibrate the confidence score of the classifier (Inception-v3), to which we opt to apply the popular network calibration method of temperature scaling [20].

[3]https://wanxiang.aliyun.com/creation
[4]https://stablediffusionweb.com
[5]https://deepai.org/machine-learning-model/texT2Img
[6]https://stanfordnlp.github.io/stanza/

The formula of I-FID is defined below.

$$\mathrm{I-FID} = ||\mu_r - \mu_g||^2 + \mathrm{trace}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}\right) \quad (15)$$

where $X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$ and $X_g \sim \mathcal{N}(\mu_g, \Sigma_g)$ are the features of real images and generated images extracted by a pre-trained Inception-v3 model. Then, these two feature sets are represented as two multivariate Gaussian distributions. A lower I-FID value indicates superior image realism. The formula of I-IS is defined below.

$$\mathrm{I-IS} = \exp(\mathbb{E}_x D_{KL}(p(y_c|x) \parallel p(y_c))), \quad (16)$$

where x is the generated image and y_c is the class label. We consider that $p(y_c|x)$ must have low entropy and $p(y_c)$ must have high entropy. Therefore the KL-divergence between $p(y_c)|x)$ and $p(y_c|x)$ should be large. Higher IS value means better image quality and diversity.

The RP metric is widely utilized to assess the consistency between text and image. The principle behind RP involves re-querying a synthesized image using the initial input caption. Specifically, an image is generated based on a true textual description, amidst 99 other randomly chosen mismatched captions. This generated image is then used to search for the input description among 100 potential captions. The retrieval is considered successful if the image's matching score with the original caption ranks the highest. For this multi-modal encoding process, we chose CLIP following [13], a robust encoder for both text and images, which has been trained on an extensive dataset containing 400 million text-image pairs. A higher RP value indicates a better text-image matching degree.

Secondly, we define the error-detection ratio of entity ($Error_e$) and error ratio of relationship ($Error_r$) based on the MRs defined in Sec 3.2.

$$Error_e = \frac{\sum(p_e)}{N}, \quad (17)$$

$$Error_r = \frac{\sum(p_r)}{N}, \quad (18)$$

where p_e and p_r represent the error reports for entities and relationships according to the certain MR. N denotes the sum of the test samples. A higher error report rate indicates the detection of more defects.

Thirdly, we apply the accuracy referring to object detection and relationship retrieval for entities and relationships in generated images, detected by the scene graph generation model [47] (The model possesses a detection accuracy rate of 90%) to evaluate the text-image relevance. To more vividly demonstrate our testing method, we use 1 minus the accuracy rate to represent the miss-detection ratio. The two metrics are formulated as follows:

$$Miss_e = 1 - \frac{\sum D_e(r)}{\sum Set_e(s)} \quad (19)$$

$$Miss_r = 1 - \frac{\sum D_r(r)}{\sum Set_r(s)} \quad (20)$$

where s and r represent the input text and generated image. $Set_r(s)$ and $Set_e(s)$ donate the relationships and entities contained in the input text. $D_e(r)$ and $D_r(r)$ represent the detection results of entities and relationships by model D_e and D_r. A higher miss-detection ratio means that more focal entities and relationships are lost in the generated images.

5 RESULT ANALYSIS AND DISCUSSION

5.1 Answer to RQ1

We conduct our ACTesting on each software to explicitly demonstrate the effectiveness of our method. We also display the testing results of the metrics on real images, illustrating the differences between generative images and real images across various operators.

Results. Table 1 presents the I-FID, I-IS, and RP of all tested software on the seed sets and mutation testing sets. From the third column in Table 1, it's evident that all four mutation operators effectively increased the I-FID values, thereby reducing the quality of the generated images. Among them, the ER_A operator showed an average increase of about 5%, ranking it first in effectiveness. Additionally, the I-IS values for all the software under test decrease on the four mutation test sets. The decreased range of EC, ER_A, and ER_R operators is 2.9% to 9.6%, 8% to 15%, and 1.7% to 8.8%. In the RP column, the three operators included in ACTesting significantly reduced the precision values of the seed test set, with the EC operator consistently decreasing the RP value by 16.4% to 21.1%. The ER_R and ER_A operators, on average, decreased by 7.5% and 8.4% respectively. In comparison, the performance of the SS operator was relatively mediocre.

Discussion. We discover that the degree of reduction in image quality and text-image relevance is closely related to transformation. The test sets mutated by the ER_R and ER_A operators achieve the higher I-FID values and the lower I-IS values. Additionally, the test set mutated by the EC operator leads to a consistent and substantial decrease in RP values. This is attributed to the EC operator disrupting the more common ER triples, compelling the software to generate more innovative results. As a consequence, this reduces the consistency between the text and the generated images. The reason for the lesser decline in the I-FID and I-IS metrics compared to RP is that our operators are primarily focused on detecting defects in text-image consistency. The three operators included in ACTesting all outperform the baseline SS, which serves as the baseline. The results indicate that ACTesting can generate error-revealing tests for T2I software.

Table 1: The I-FID, I-IS and RP of different software on the tests generated by four guidance approaches and original data

Software	Oper	I-FID	I-IS	RP
SD XL	Orig	24.27	45.83	94.09%
	EC	25.60	41.46	**77.18%**
	ER_R	25.54	41.80	86.95%
	ER_A	**25.67**	**38.65**	87.05%
	SS	25.63	44.09	92.07%
DeepAI	Orig	26.41	44.38	93.01%
	EC	28.03	43.15	**77.20%**
	ER_R	28.36	43.62	86.39%
	ER_A	**28.60**	**40.49**	86.66%
	SS	27.68	43.47	90.35%
Wan Xiang	Orig	26.16	48.38	94.02%
	EC	27.99	44.16	**78.56%**
	ER_R	32.15	45.14	88.06%
	ER_A	**33.11**	**41.94**	86.24%
	SS	28.53	48.78	92.79%
OpenJourney	Orig	30.36	43.80	91.68%
	EC	30.76	41.14	**72.33%**
	ER_R	31.29	42.52	82.99%
	ER_A	**31.49**	**40.29**	81.91%
	SS	30.97	42.64	87.47%
MS COCO	Real-Image	2.62	46.00	83.54%

5.2 Answer to RQ2

For each operator, ACTesting designs a related MR to detect the defect. As described in Section 3.2, we consider the degree of matching between images and text based on the ERs. Therefore, the corresponding MR for each operator will be categorized into two types: entities and relationships. These will be separately addressed in error reporting, and donating error rate of entities and relationships ($Error_e$ and $Error_r$). The MR for the baseline SS operator is set such that both entities and relationship sets remain unchanged.

Results. Figure.6 presents the $Error_e$ and $Error_r$ of different software on the tests generated by four operators (three proposed operators and one baseline). Subfigure *a* displays the error rate associated with the MR of entities. The error rate for the SS operator remains around 0.40, while the ER_R operator consistently stays around 0.60. The EC operator averages an error rate of 0.60, and the ER_A operator often ranks first, reaching its highest error rate of 0.69 in the OpenJourney software. Subfigure *b* illustrates the error rates for MR of relationships. In this case, the SS operator maintains error rates all below 0.4, while both the ER_R and EC operators exhibit error rates all above 0.6 but below 0.7. The ER_A operator ranks first, maintaining an error rate of 0.7 stably.

Discussion. Although the SS operator can also detect the defect within a certain range, our proposed operators can detect more erroneous behaviors of the tested software. The testing efficiency of the ER_A operator is around 1.75 that of the SS operator. From the experimental results of $Error_e$, ER_A is effective for most of the software both on entities and relationships. However, ER_R and EC show better performance when testing Wan Xiang. The results of $Error_r$ demonstrate that both of the three operators can detect the defects effectively and stably. Among the four software, Wan Xiang is more robust to the transformation. In conclusion, three kinds of transformation operators can effectively detect the defects of T2I software based on the related MR.

5.3 Answer to RQ3

Table 2 shows the ablation experiment results of several mutation operators. To better demonstrate the effectiveness of three kinds of operators, we conduct the ablation experiments on the three operators, with the evaluation criteria being the miss detection rate for object detection and relationship retrieval in the generated images. We use Stanza to extract the ER from the input text as the ground truth. The metrics $Miss_e$ and $Miss_r$ we use are explained in Eq.19 and Eq.20.

Results. Table 2 presents the ablation experiment results for six mutation test sets. The $Miss_e$ column values display the rates

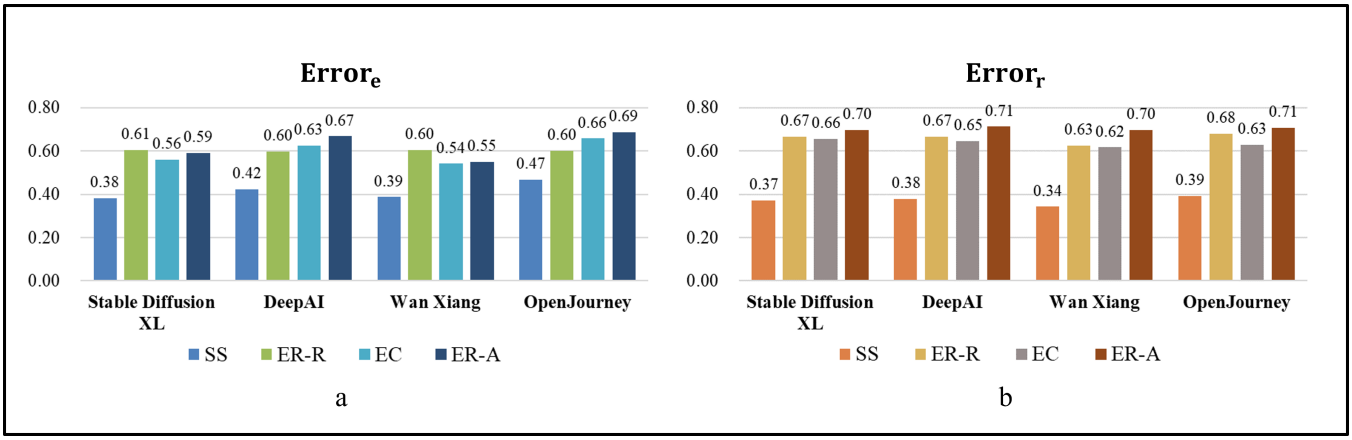

Figure 6: The error rates of different software on the tests generated by four mutation operators

Table 2: The ablation experiment results of different mutation operators.

	$Miss_e$						$Miss_r$					
Software	Orig	ER_R	ER_A	EC	EC+ER_R	EC+ER_A	Orig	ER_R	ER_A	EC	EC+ER_R	EC+ER_A
SD XL	0.1834	0.3122	0.3325	0.3528	0.3702	**0.4338**	0.3227	0.2986	0.3332	0.3329	0.3443	**0.3528**
DeepAI	0.2076	0.3086	0.3732	0.3675	0.3904	**0.4755**	0.3206	0.3107	0.3261	0.3430	0.3435	**0.3580**
Wan Xiang	0.1844	0.2733	0.3271	0.3449	0.3598	**0.4293**	0.3360	0.3164	0.3334	0.3433	0.3506	**0.3541**
OpenJourney	0.2364	0.2876	0.4099	0.4011	0.4273	**0.5210**	0.3395	0.3301	0.3463	0.3464	0.3520	**0.3699**

of missed detection for the four software under various mutation conditions. The table clearly shows that the tests conducted with the combined EC plus ER_R, and EC plus ER_A operators result in the highest rates of missed detection, surpassing those of the three individual operators. The value of $Miss_e$ for the EC+ER_A operator is, on average, 26% higher than that of the EC operator, and 29.1% higher than that of the ER_A operator. The average value of $Miss_e$ for the EC+ER_R operator exceeds that of the EC operator by 5.5% and surpasses the ER_R operator by a significant margin of 31.3%. The $Miss_r$ column reflects the rate of missed detections for relationships. It shows a trend consistent with that of the $Miss_e$ column. Specifically, the average $Miss_r$ value for the EC+ER_A operator exceeds those of the EC and ER_A operators by 5.1% and 7.2%, respectively. Although the EC+ER_R operator leads the EC operator on average $Miss_r$ by only 1.8%, it surpasses the ER_R operator by a notable margin of 10.8%, still demonstrating significant competitiveness.

Discussion. The ablation study demonstrates that three kinds of operators are effective and flexible to use. Firstly, the value of $Miss_e$ and $Miss_r$ for each mutation operator is substantially higher than that of the seed test set, where EC+ER_A shows the best performance. This aligns with the trends observed in the common metrics presented in Table 1, further validating the effectiveness of this testing method. Secondly, we observe that the missed detection rate for entity recognition is higher than that for relationship recognition. This may be attributed to the higher precision of the object detection model and indicates that the accuracy of relationship recognition in image scene understanding still requires improvement. Additionally, the reason for the relatively lower competitiveness of the ER_R and EC+ER_R operators is that removing certain ER somewhat facilitates the T2I software in processing the remaining content. However, due to the removal of connective semantic properties, this operator still manages to detect defects in the software. This experiment also demonstrates that both Wan Xiang and Stable Diffusion XL exhibit commendable robustness.

5.4 Analysis

When we abstract the text-image pairs into triples, for text, it is not necessary to describe all relationships between entities, while for images, it is necessary to present all of them when constructing a visual scene. This difference in presentation formats can lead generative software to interpret the text as triples in different paradigms, leading to the omission of certain entities or relationships during the generation process.

Complex nested ERs often pose significant challenges for building models to generate images. In each input with nested ERs, these can be understood as multiple nodes and multiple edges. This means that the model needs to handle the joint probability distribution of each node and edge. Since nested relationships are typically highly correlated, the joint probability distribution between child nodes and parent nodes can influence each other, leading to a decrease in model performance and an inability to accurately predict node states.

Regarding the most commonly used diffusion models, which are based on modeling noise probability distributions, their loss function constrains the predicted noise to be close to the true noise at each step. When there are more or unfamiliar entity relationships and the joint probability distribution becomes more complex, the

denoising process becomes more challenging. Their mechanisms tend to involve attention mechanisms, often selecting the parts they consider important for construction while discarding the parts that are difficult to build or that they consider unimportant.

5.5 Threats to Validity

Test subject. The selection of T2I software is one of the vital threats to validity. With the rapid development of T2I software, a diverse array of such software is continually emerging, each with varying engines and mechanisms. To alleviate this threat, we employ four commonly used and popular T2I software powered by different core engines.

Dataset Selection The selection of the source dataset is another threat that comes from. Due to the fact that not all training and validation datasets used by various software are publicly available, there is a potential issue of data domain unfairness. To alleviate this threat, we select the most commonly used and classic open-source dataset MS-COCO as our seed collection to ensure a level of fairness in the experiments.

Data transformation implementation. The last threat is related to the data transformation implementation. Although the mutation operators are well-designed, there still exists a margin of error in identifying entities and relationships within the input text. To overcome this issue, we use both the Stanza and the Bert models to enhance the precision of entity and relationship identification.

6 RELATED WORK

This section covers the synthesis method employed by T2I software for text-to-image conversion and discusses the evaluation metrics commonly used in this domain. Following this, we introduce various AI testing methods that are relevant to our study.

6.1 Text-to-Image Synthesis and Evaluation

With the advancement of deep learning, particularly the widespread application of large models in recent times, the performance of T2I software has seen a significant leap forward. Over an extended period, the majority of methods are based on Generative Adversarial Network (GAN), including stacked architecture-based [4, 59, 60], attention-based [24, 46, 54], siamese architecture-based [14, 21, 55] and cycle consistency-based [28, 33, 36]. As diffusion models make significant breakthroughs in generative tasks, researchers increasingly focus their attention on diffusion structure to construct T2I engine [19, 27, 40, 41]. In addition, autoregressive methods utilize time-series models on large datasets, relying on historical data dependencies across various periods show good performance. A prime example is OpenAI's DALL-E [39]and Google's Parti model [57]. However, although more complex model structures enhance the generative effects, T2I software still has inevitable issues.

Therefore, researchers have proposed a series of evaluation metrics to reasonably assess the performance of T2I software, which are usually based on image quality and text-image alignment. Inception score (IS) [42] and Fréchet Inception Distance (FIS) [6] are two common indicators to leverage the image quality and image diversity based on pre-trained Inception-v3 network [45]. R-precision (RP) [54] is usually used to assess the consistency between text and images. In addition to these general metrics, they also have some new metrics. Tan et al. [13] propose several improved metrics based on IS, FID, SOA, and RP. Cho et al. [10] recently introduce two novel interpretable/explainable visual programming frameworks for T2I generation.

However, aside from the issue of overfitting inherent in evaluation metrics themselves, T2I software lacks systematic testing methods to assess its robustness. Unlike the previous methods which focus on evaluating the generation quality and text-image alignment, ACTesting aims to utilize the entity-relationship triple to construct the cross-model testing framework for the generation robustness.

6.2 AI Testing

Due to the scarcity of testing methods for the T2I task, we introduce some relevant testing methods. After achieving success in traditional testing tasks with mutation testing, researchers propose testing methods to test the deep learning systems. Lei et al. [31] apply the mutation testing method and specialize it for deep learning systems to assess the quality of test data. David et al. [3] conduct a rich empirical study identifying the impact of mutation operators and coverage criteria on the distribution of the generated deep learning test cases. In addition, Simos et al. [18] propose a systematic testing methodology, Importance-Driven(IDC), to assess the semantic diversity of a test set. Currently, one of the most prominent areas of interest in AI testing is the testing methods aimed at autonomous driving [15, 17, 29, 49]. AI testing methods also cover other new areas, including speech recognition systems [26], question-answering systems [30], and image captioning systems [56].

However, up to this point, no testing methods have been specifically proposed for T2I software. Therefore, building upon cross-modal consistency and mutation testing, we propose ACTesting, a black-box testing method specifically designed for T2I software. This approach is applicable to all T2I software and is effective in detecting defects and issues within them.

7 CONCLUSION

In this paper, we introduce ACTesting, an automated, black-box approach grounded in metamorphic testing theory to address the challenges in cross-modal testing for Text-to-Image (T2I) software. ACTesting emphasizes maintaining semantic consistency across different modalities and utilizes the entity-relationship triple to encapsulate key semantic information. We develop three mutation operators based on this triple to identify defects in T2I software, focusing on testing the generation robustness of T2I software. Our evaluation of ACTesting involves four T2I software, generating 113,736 synthetic images, using the MS-COCO validation dataset. We employed metrics I-FID, I-IS, and RP for assessing image realism and text-image relevance. Additionally, we introduced error-detection ($Error_e$ and $Error_r$) and miss-detection ($Miss_e$ and $Miss_r$) rates. The findings revealed that ACTesting could degrade image quality by 2.9% to 15% and diminish text-image match consistency by 7.5% to 21.1%. Furthermore, our ablation study showcased the individual effectiveness of each operator, highlighting their capability for flexible combination and superior performance in increasing miss-detection rates compared to basic operators.

REFERENCES

[1] Alibaba. 2023. Wan Xiang. Website. https://wanxiang.aliyun.com/creation.

[2] Baidu. 2023. ERNIE-ViLG. Website. https://ai.baidu.com/tech/creativity/ernie_Vilg.

[3] David Berend, Xiaofei Xie, Lei Ma, Lingjun Zhou, Yang Liu, Chi Xu, and Jianjun Zhao. 2020. Cats are not fish: Deep learning testing calls for out-of-distribution awareness. (2020), 1041–1052.

[4] Navaneeth Bodla, Gang Hua, and Rama Chellappa. 2018. Semi-supervised FusedGAN for conditional image generation. In *Proceedings of the European conference on computer vision*. 669–683.

[5] Ali Borji. 2019. Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding* 179 (2019), 41–65. https://doi.org/10.1016/j.cviu.2018.10.009

[6] Naresh Babu Bynagari. 2019. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Asian Journal of Applied Science and Engineering* 8 (2019), 25–34.

[7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1209–1218.

[8] Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, TH Tse, and Zhi Quan Zhou. 2018. Metamorphic testing: A review of challenges and opportunities. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 1–27.

[9] Meng-Jiun Chiou, Roger Zimmermann, and Jiashi Feng. 2021. Visual Relationship Detection With Visual-Linguistic Knowledge From Multimodal Representations. *IEEE Access* 9 (2021), 50441–50451.

[10] Jaemin Cho, Abhay Zala, and Mohit Bansal. 2023. Visual Programming for Text-to-Image Generation and Evaluation. *arXiv preprint arXiv:2305.15328* (2023), 1–22.

[11] DeepAI. 2023. DeepAI Image Generator. Website. https://deepai.org/machine-learning-model/text2img.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018), 1–16.

[13] Tan M Dinh, Rang Nguyen, and Binh-Son Hua. 2022. TISE: Bag of metrics for text-to-image synthesis evaluation. In *European Conference on Computer Vision*. 594–609.

[14] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629* (2016), 1–26.

[15] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. 2020. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems* 22, 3 (2020), 1341–1360.

[16] Stanislav Frolov, Tobias Hinz, Federico Raue, Jörn Hees, and Andreas Dengel. 2021. Adversarial text-to-image synthesis: A review. *Neural Networks* 144 (2021), 187–209.

[17] Xinyu Gao, Zhijie Wang, Yang Feng, Lei Ma, Zhenyu Chen, and Baowen Xu. 2023. Benchmarking Robustness of AI-enabled Multi-sensor Fusion Systems: Challenges and Opportunities. *arXiv preprint arXiv:2306.03454* (2023), 1–12.

[18] Simos Gerasimou, Hasan Ferit Eniser, Alper Sen, and Alper Cakan. 2020. Importance-driven deep learning system testing. (2020), 702–713.

[19] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2022. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10696–10706.

[20] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. 1321–1330.

[21] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition*. 1735–1742.

[22] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. 2020. Semantic object accuracy for generative text-to-image synthesis. *IEEE transactions on pattern analysis and machine intelligence* 44, 3 (2020), 1552–1565.

[23] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. 2023. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778* (2023), 1–16.

[24] Wanming Huang, Richard Yi Da Xu, and Ian Oppermann. 2019. Realistic image generation using region-phrase attention. In *Asian Conference on Machine Learning*. 284–299.

[25] Tuan-Luc Huynh, Khoi-Nguyen Nguyen-Ngoc, Chi-Bien Chu, Minh-Triet Tran, and Trung-Nghia Le. 2022. Multilingual Communication System with Deaf Individuals Utilizing Natural and Visual Languages. In *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*. 683–688. https://doi.org/10.1109/RIVF55975.2022.10013851

[26] Pin Ji, Yang Feng, Jia Liu, Zhihong Zhao, and Zhenyu Chen. 2022. ASRTest: automated testing for deep-neural-network-driven speech recognition systems. (2022), 189–201.

[27] Taehoon Kim, Gwangmo Song, Sihaeng Lee, Sangyun Kim, Yewon Seo, Soonyoung Lee, Seung Hwan Kim, Honglak Lee, and Kyunghoon Bae. 2022. L-verse: Bidirectional generation between image and text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16526–16536.

[28] Qicheng Lao, Mohammad Havaei, Ahmad Pesaranghader, Francis Dutil, Lisa Di Jorio, and Thomas Fevens. 2019. Dual adversarial inference for text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7567–7576.

[29] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. 2022. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. (2022), 17182–17191.

[30] Zixi Liu, Yang Feng, Yining Yin, Jingyu Sun, Zhenyu Chen, and Baowen Xu. 2022. QATest: A Uniform Fuzzing Framework for Question Answering Systems. (2022), 1–12.

[31] Lei Ma, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Felix Juefei-Xu, Chao Xie, Li Li, Yang Liu, Jianjun Zhao, et al. 2018. Deepmutation: Mutation testing of deep learning systems. (2018), 100–111.

[32] Midjourney. 2023. Midjourney. Website. https://www.midjourney.com.

[33] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4467–4477.

[34] OpenAI. 2023. DALL·E 2. Website. https://openai.com/research/dall-e.

[35] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082* (2020), 1–8.

[36] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. 2019. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1505–1514.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[38] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *URL https://arxiv. org/abs/2204.06125* 7 (2022), 1–5.

[39] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*. PMLR, 8821–8831.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[41] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* 35 (2022), 36479–36494.

[42] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems* 29 (2016), 1–10.

[43] Sergio Segura, Gordon Fraser, Ana B Sanchez, and Antonio Ruiz-Cortés. 2016. A survey on metamorphic testing. *IEEE Transactions on software engineering* 42, 9 (2016), 805–824.

[44] Stability.ai. 2023. Stable Diffusion. Website. https://stablediffusionweb.com.

[45] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[46] Hongchen Tan, Xiuping Liu, Xin Li, Yi Zhang, and Baocai Yin. 2019. Semantics-enhanced adversarial nets for text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10501–10510.

[47] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. Unbiased Scene Graph Generation from Biased Training. In *Conference on Computer Vision and Pattern Recognition*. 1–16.

[48] Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2015. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844* (2015), 1–10.

[49] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. (2018), 303–314.

[50] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. 2023. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944* (2023), 1–33.

[51] Luke Tredinnick and Claire Laybats. 2023. The dangers of generative artificial intelligence. *Business Information Review* 40 (2023), 02663821231183756.

[52] Fuxiang Wu, Liu Liu, Fusheng Hao, Fengxiang He, and Jun Cheng. 2022. Text-to-Image Synthesis based on Object-Guided Joint-Decoding Transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 18092–18101. https://doi.org/10.1109/CVPR52688.2022.01758

[53] Xianchao Wu. 2022. Creative painting with latent diffusion models. *arXiv preprint arXiv:2209.14697* (2022), 1–17.

[54] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1316–1324.

[55] Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. 2019. Semantics disentangling for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2327–2336.

[56] Boxi Yu, Zhiqing Zhong, Xinran Qin, Jiayi Yao, Yuancheng Wang, and Pinjia He. 2022. Automated testing of image captioning systems. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*. 467–479.

[57] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. 2022. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789* 2 (2022), 1–5.

[58] Qifan Yu, Juncheng Li, Yu Wu, Siliang Tang, Wei Ji, and Yueting Zhuang. 2023. Visually-Prompted Language Model for Fine-Grained Scene Graph Generation in an Open World. *arXiv preprint arXiv:2303.13233* 1 (2023), 1–18.

[59] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 5907–5915.

[60] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2018. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence* 41, 8 (2018), 1947–1962.

[61] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.