Repairing Catastrophic-Neglect in Text-to-Image Diffusion Models via Attention-Guided Feature Enhancement

Zhiyuan Chang[1,2,3] **Mingyang Li**[1,2,3*] **Junjie Wang**[1,2,3]
Yi Liu[4] **Qing Wang**[1,2,3*] **Yang Liu**[4]
[1]State Key Laboratory of Intelligent Game, Beijing, China
[2]Science and Technology on Integrated Information System Laboratory,
Institute of Software Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences [4]Nanyang Technological University
{zhiyuan2019, mingyang2017, junjie, wq}@iscas.ac.cn, yi009@e.ntu.edu.sg, yangliu@ntu.edu.sg

Abstract

Text-to-Image Diffusion Models (T2I DMs) have garnered significant attention for their ability to generate high-quality images from textual descriptions. However, these models often produce images that do not fully align with the input prompts, resulting in semantic inconsistencies. The most prominent issue among these semantic inconsistencies is catastrophic-neglect, where the images generated by T2I DMs miss key objects mentioned in the prompt. We first conduct an empirical study on this issue, exploring the prevalence of catastrophic-neglect, potential mitigation strategies with feature enhancement, and the insights gained. Guided by the empirical findings, we propose an automated repair approach named *Patcher* to address catastrophic-neglect in T2I DMs. Specifically, *Patcher* first determines whether there are any neglected objects in the prompt, and then applies attention-guided feature enhancement to these neglected objects, resulting in a repaired prompt. Experimental results on three versions of Stable Diffusion demonstrate that *Patcher* effectively repairs the issue of catastrophic-neglect, achieving 10.1%-16.3% higher Correct Rate in image generation compared to baselines.

1 Introduction

Text-to-Image Diffusion Models (T2I DMs) (Rombach et al., 2022; Saharia et al., 2022; Ramesh et al., 2022a) have gained widespread attention in recent years due to their remarkable ability to generate images from textual descriptions (i.e. prompt). However, it has been demonstrated that the image generated by T2I DMs may not strictly adhere to the description of the input prompt, leading to inconsistencies in the semantics.

To this end, many approaches have been proposed to enhance the generation quality through inference process optimization (Liu et al., 2022; Feng et al., 2023; Chefer et al., 2023) and handcrafted prompt writing guidelines (Liu and Chilton, 2022; Oppenlaender, 2022). The former requires modifications to the model structure or parameters, which is difficult for users to perform. Although the latter is relatively easier to implement, it requires a significant amount of manual effort and suffers poor scalability. Recently, Hao et al. (2023) also proposed a method to enhance the quality of generated images by automating the refinement of user-inputted prompts.

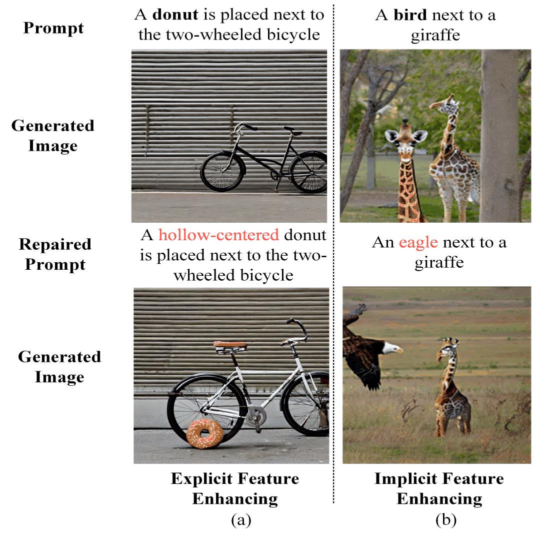

Prompt
A **donut** is placed next to the two-wheeled bicycle
A **bird** next to a giraffe
Generated Image
Repaired Prompt
A hollow-centered donut is placed next to the two-wheeled bicycle
An eagle next to a giraffe
Generated Image
Explicit Feature Enhancing
(a)
Implicit Feature Enhancing
(b)

Figure 1: Examples of catastrophic neglect in the generated images by T2I DMs, and the enhancement of explicit and implicit features.

According to previous study (Chefer et al., 2023), one of the most prominent issues in semantic consistency is the **catastrophic-neglect**, i.e., the images generated by T2I DMs often miss some of the key objects mentioned in the textual prompts. This issue is particularly prevalent when a prompt involves multiple objects. Figure 1 demonstrates two illustrative cases where one of the two objects is neglected by T2I DMs. In Figure 1 (a), we notice that the object "bicycle" in prompt is described with the explicit feature "two-wheeled" while "donut"

[*] Corresponding authors

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 11379–11390
November 12-16, 2024 ©2024 Association for Computational Linguistics

is not. We try to craft prompts to repair the issue, and results reveal that by adding a specific explicit feature to the "donut" (e.g., "hollow-centered"), the catastrophic-neglect issue can be resolved. Furthermore, as the feature is added, the attention difference between the two mentioned objects (i.e., "bicycle" and "donut") is reduced according to the explainable tool (Tang et al., 2023). It seems that reduction in attention difference can potentially indicate the T2I DMs put more balanced attention towards the two involved objects, resulting both of them can be successfully generated. In Figure 1 (b), we notice that the object "bird" in the prompt is a more general concept with fewer implicit features compared with the concept "giraffe", according to the hierarchical structure in WordNet (Miller, 1995). Taken in this sense, we can successfully repair the issue through using more imageable concept (such as "eagle") to replace "bird" in the prompt, and the attention difference between two mentioned objects (i.e., "eagle" and "giraffe") is also reduced. Motivational study in Section 2 provides more details.

Motivated by the above analysis, we assume the attention difference can guide the mitigation of catastrophic-neglect issue, and this can be achieved through enhancing objects with specific features (i.e., explicit features) or using more imageable concepts (i.e., implicit features) to balance the attention among involved objects in the prompt.

Therefore, this paper proposes an automatic repair approach named *Patcher* to address catastrophic-neglect in T2I DMs, guided by the attention difference among objects of input prompt. Specifically, *Patcher* first parses the original prompt and identifies the objects neglected by the T2I DMs. Then, guided by the difference of attention scores, *Patcher* produces the repaired prompt via enhancing explicit feature (achieved by asking LLMs for suitable modifiers) and implicit features (realized by hyponym substitution using WordNet), and redetermined whether there are still neglected objects in the generated image.

Experimental results demonstrate that *Patcher* effectively repairs the issue of catastrophic-neglect in T2I DMs, achieving 10.1%-16.3% higher Correct Rate in image generation compared to baselines, as tested on Stable-Diffusion V1.4, V1.5, and V2.1 models. Additionally, ablation study shows that both explicit and implicit feature enhancing in *Patcher* contribute to resolving the catastrophic-neglect issue in T2I DMs. We provide the public reproduction package[1].

2 Motivation

To better understand catastrophic-neglect and guide the design of the automated repair approach, we conduct the empirical analysis from three aspects, i.e., their prevalence across prompts with different number of objects, potential mitigation strategies based on feature enhancement, and corresponding insights into the effectiveness of feature enhancement.

2.1 Issue Prevalence

On the one side, we investigate the error rate of T2I DMs in handling prompts involving different numbers of objects through manual evaluation. On the other side, we explore the proportion of catastrophic-neglect among all errors. First, we construct three datasets containing single-object, double-object and triple-object prompts respectively. For the single-object prompts, we reuse the 80 object descriptions from different semantic categories in MSCOCO dataset (Lin et al., 2014) Based on these single-object prompts, we synthesize new prompts containing two or three objects using GPT-3.5 by adding essential conjunctions, adverbs or interactions , aiming to generate inputs for T2I DMs that conform to human expressions[2].

We then input the single-object prompts and multi-object prompts into Stable Diffusion V2.1, a state-of-the-art T2I DM, and manually evaluate the proportion of incorrectly generated images that are not consistent with the prompt (i.e. Error Rate).

The evaluation results show that the Error Rate significantly increases (2.5%->50.4%->86.0%) with the numbers of objects in the prompt. Furthermore, for the prompts with single, double and triple object, catastrophic-neglect issue accounts for 100%, 93.4%, and 94.0% of all the incorrectly generated images. The remaining incorrectly images are those where the features of multiple objects are blended into a single object. In general, when faced with multi-object prompts, the T2I DM is prone to generating incorrect images, with catastrophic-neglect being the most severe issue in such scenario.

[1] https://github.com/lsplx/patcher

[2] The size of the dataset is described in Sections 4.1.

2.2 Issue Mitigation via Feature Enhancement

Section 1 has illustratively demonstrated that the imbalance of explicit/implicit features carried by objects in the prompts may lead to the catastrophic-neglect. This section tries to craft the prompts with the idea of adding explicit or implicit features to those neglected objects to investigate whether the issue could be mitigated statistically. Specifically, we apply feature enhancement to double-object and triple-object datasets (Constructed in Section 2.1 with 4041 prompts). First, we manually add explicit features to the neglected objects. These features enhance the physical appearance of the original objects without altering the semantic meaning of the original prompts. As shown in Figure 1, the neglected object "donut" was enhanced with the feature "hollow-centered".

Second, we enhance the prompts using implicit features. We manually replace the description of the neglected object with its hyponym with help of WordNet, which denotes a specific concept compared to the original object (Miller, 1995). As shown in Figure 1, we replaced "bird" with "eagle" to obtain the repaired prompt. The evaluation results show that, compared to the Error Rate before feature enhancement, manually constructed explicit and implicit features reduce Stable Diffusion's Error Rate by 26.9% and 24.6%, respectively.

2.3 Explanation for Feature Enhancement

To explore the reasons behind feature enhancement, we use the attention explainability tool (DAAM) (Tang et al., 2023) to investigate whether the attention differences between multiple objects change before and after feature enhancement. Given a specific token from the input prompt, DAAM aggregates the T2I DM's cross-attention values across layers to obtain its **attention score**. The attention score of each token represents the token's importance in the image generation process. The **attention difference** indicates the disparity in the T2I DM's attention score to different object tokens. We assume that reducing the attention difference between multiple objects can help the T2I DM more evenly focus on the features of each object and generate them correctly. For double-object prompts, we compute the absolute difference in attention scores between the two objects. For prompts with triple object, we first calculate the pairwise differences in attention scores and then average them. We use the prompts that generates incorrectly images from multi-objects prompts constructed in Section 2.1 and the repaired prompts manually constructed in Section 2.2.

Table 1: The attention difference between multiple objects before and after using explicit and implicit features. 'Correct' and 'Wrong' respectively indicates the results of the newly generated images after adding the features.

Strategy	**Correct**		**Wrong**	
	Before	***After***	***Before***	***After***
Explicit Feature	658	232	808	887
Implicit Feature	934	437	713	1442

The result is shown in Table 1. The attention difference between multiple objects significantly decreases for prompts that correctly generate images after enhancing explicit or implicit features. Besides, this reduction in attention difference accounts for 80.9% of the correctly generated images. In contrast, for prompts that still generate incorrect images, the attention difference increases. Moreover, the reduction in attention difference accounts for 29.0% of these incorrect generated images. This indicates that features reducing the attention difference between objects are more effective in repairing the catastrophic-neglect in the T2I DM.

3 Methodology

Figure 2 shows the overview of *Patcher*. *Patcher* consists of two stages: (1) **Neglected Objects Identification** would determine whether the T2I DM neglect any objects in the input prompt; (2) **Feature Enhancement for Neglected Objects** would enhance explicit and implicit features for neglected objects and construct the repaired prompt.

3.1 Neglected Objects Identification

To identify the neglected objects, *Patcher* first extracts the objects from the input prompt. Specifically, *Patcher* first parses the textual descriptions into a dependency tree using a transformer-based (Vaswani et al., 2017) language model[3]. It then extracts noun phrases from this tree as the object entities. In the meanwhile, *Patcher* employs DAAM to obtain the attention scores and produces the token-attention pairs (TAP) for each token in the prompt description, which will be utilized in Section 3.2 to guide the feature enhancement.

After that, *Patcher* calculates the similarities of each extracted object entities and generated images

[3] https://huggingface.co/spacy/en_core_web_trf

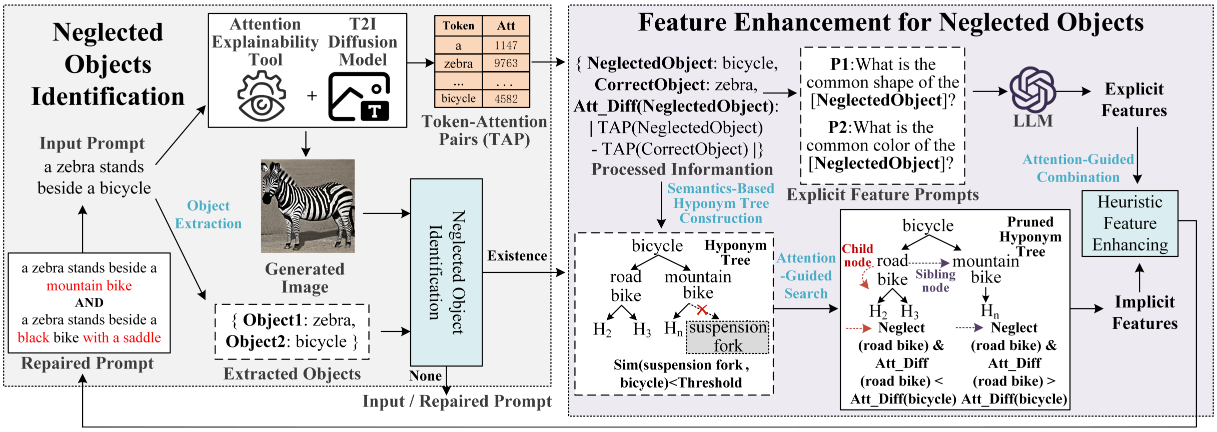

Neglected Objects Identification
Attention Explainability Tool
T2I Diffusion Model
+
Token | Att
a | 1147
zebra | 9763
... | ...
bicycle | 4582
Token-Attention Pairs (TAP)
Input Prompt
a zebra stands beside a bicycle
Object Extraction
Generated Image
{ Object1: zebra, Object2: bicycle }
Extracted Objects
Neglected Object Identification
Existence
None
Input / Repaired Prompt
a zebra stands beside a mountain bike
AND
a zebra stands beside a black bike with a saddle
Repaired Prompt
Feature Enhancement for Neglected Objects
{ NeglectedObject: bicycle, CorrectObject: zebra, Att_Diff(NeglectedObject): | TAP(NeglectedObject) - TAP(CorrectObject) |}
Processed Informantion
P1:What is the common shape of the [NeglectedObject]?
P2:What is the common color of the [NeglectedObject]?
Explicit Feature Prompts
LLM
Explicit Features
Attention-Guided Combination
Heuristic Feature Enhancing
Semantics-Based Hyponym Tree Construction
bicycle
Hyponym Tree
road bike
mountain bike
H2
H3
Hn
suspension fork
Sim(suspension fork , bicycle)<Threshold
Attention -Guided Search
bicycle
Pruned Hyponym Tree
Child node
road bike
mountain bike
Sibling node
H2
H3
Hn
Neglect (road bike) & Att_Diff (road bike) < Att_Diff(bicycle)
Neglect (road bike) & Att_Diff (road bike) > Att_Diff(bicycle)
Implicit Features

Figure 2: The overview of *Patcher*. The procedure in the dashed box is executed only the first time.

by Clipscore (Radford et al., 2021). Due to the presence of corresponding visual features when the object is in the image and their absence when it is not, there is a significant difference in similarity between the two scenarios, *Patcher* sets a threshold based on the empirical study to determine whether an object is neglected in the image. If the similarity between the object and the image is below the threshold, we consider the object to be neglected by the T2I model. Conversely, we consider the object to be correctly generated by the T2I model. If there are no neglected objects in the prompt, output the current prompts; otherwise, *Patcher* sends the prompt into the following stage for repair.

3.2 Feature Enhancement for Neglected Objects

After the first stage, *Patcher* derives a set of neglected objects and a set of correctly identified objects. Recall that it also obtains the attention scores for each token in the first stage. Typically, an object contains a single token; if it contains multiple tokens, *Patcher* calculates the average of the attention scores for these tokens. In this way, we obtain the attention score for each neglected object and correct object. We then calculate the differences in attention scores between neglected objects and correct objects.

Specifically, it first calculates the pairwise differences between attention scores of objects from the neglected and correct object sets, and then averages these differences. This provides a comprehensive measure of how uniformly the T2I DM's attention is distributed between two set of objects. The calculation process is shown in Equation 1, where O_i denotes the attention score corresponding to the i-th object from the neglected object set, and O_j denotes the attention score corresponding to the j-th object from the correct object set.

$$\text{Att\_Diff} = \frac{1}{|N||C|} \sum_{O_i \in N} \sum_{O_j \in C} |O_i - O_j| \qquad (1)$$

Next, *Patcher* employs two repair strategies: 1) *Explicit Feature Enhancing*, which is used to obtain the physical features of the neglected objects; 2) *Implicit Feature Enhancing*, which is used to obtain hyponyms of the neglected objects guided by the attention difference. With the two strategies, *Patcher* simultaneously generates explicit and implicit features, each forming a repaired prompt, which together constitute two repaired prompts to determine whether there are neglected objects in them. Following introduces the prompt repair process with the two strategies respectively.

3.2.1 Explicit Feature Enhancement

From the explicit perspective, objects' features are enhanced from two aspects, i.e., shape and color, leveraging the LLM's powerful understanding of the general knowledge (Chang et al., 2024) with the carefully designed prompt (See Appendix A for specific details).

The prompt consists of three parts: 1) the specific question, which directly asks the LLM about the core objective, 2) the output guidelines, which constrain the format of the model's output and guide it to produce diverse responses, and 3) the example, which helps the LLM understand the question and produce the response expected by the users. As shown in Figure 2, *Patcher* inputs the explicit feature prompts into the LLM[4], which in return provides a variety set of explicit features. For each explicit feature, *Patcher* replaces the description of the

[4] The LLM is GPT-3.5

neglected object in the original prompt with an enhanced description containing the object and its explicit feature, generating a candidate prompt. *Patcher* iteratively queries the T2I models with the candidate prompts until no neglected objects (determined with the strategy in Section 3.1) or reaching the maximum iteration number (set as 4 in our study). If all color and shape explicit features fail to make the neglected object visible in the image, *Patcher* selects the feature with the smallest attention difference from both candidate sets of color and shape, then combines them to generate the final repaired prompt.

3.2.2 Implicit Feature Enhancement

To obtain the hyponyms of a neglected object, *Patcher* uses Natural Language Processing tool (Bird et al., 2009) to search all hyponyms of the object, i.e., including the direct hyponyms and those indirect hyponyms, recursively, until no further hyponyms are found. As shown in the hyponym tree in Figure 2, *Patcher* constructs a hyponym tree for "bicycle", where the child node "mountain bike" is a direct hyponyms of "bicycle". For nodes at the same hierarchical level, such as "mountain bike" and "road bike", their conceptual levels are similar, making them sibling nodes. Besides, the child nodes of "mountain bike" are indirect hyponyms of "bicycle". Among these, some indirect hyponyms such as "Suspension Fork" have already deviated from the original semantic concept of the root node "bicycle", which could not help the T2I DM generate correct original object. To mitigate this issue, *Patcher* performs semantic-based pruning for the hyponym tree. Specifically, by traversing each child node of the hyponym tree using breadth-first search, *Patcher* maps the textual representation of the current node object and neglected object into a vector space using a language model (Brown et al., 2020), then computes the cosine similarity between them. If the similarity is below a certain threshold, *Patcher* prunes the current node and its children.

After that, *Patcher* performs an attention-guided search on the pruned hyponym tree, as detailed in Algorithm 1. For each node, *Patcher* first replaces the neglected object in the original prompt with the hyponym represented by that node (Line 2-4). Then, input the generated repaired prompt into the Neglected Objects Identification Stage to judge whether the neglected object still exists. If there are no neglected objects, output the repaired prompt (Line 5-8); otherwise, proceed with the attention-guided search (Line 9-15). Specifically, *Patcher* calculates the attention difference between the replaced hyponym and the correct objects in the repaired prompt, then compares it with the original attention difference between neglected object and correct objects. Considering that child nodes contains more implicit features compared to sibling nodes, if the attention difference is reduced, *Patcher* continues the search with the child nodes of the current node; otherwise, search its sibling nodes.

Algorithm 1: Attention-Guided Search

```
Input: Hyponym tree $T$ with root node (Neglected
       Object) $NO$, original prompt $P$, and
       Att_Diff(NO)
1  Initialize a queue $Q$ with tree $T$ and enqueue the root
     node $NO$;
2  while $Q$ is not empty do
3      $node$ = $Q$.dequeue();
4      repaired prompt = replace($P, NO, node$);
5      $judgment$ = Judge(repaired prompt);
6      if $judgment$ is None then
7          output repaired prompt;
8          return;
9      else
10         if $Att\_Diff(node) < Att\_Diff(NO)$
             then
11             for each child $c$ of $node$ do
12                 $Q$.enqueue($c$);
13         else
14             for each sibling $s$ of $node$ do
15                 $Q$.enqueue($s$);
16 output "No correct node found";
```

During the process of explicit and implicit feature enhancement, if a correct image is generated, the corresponding repaired prompt is returned. Otherwise, the prompt that achieves the minimum attention difference is returned.

4 Experimental Setup

4.1 Datasets

For experimental evaluation, we first introduce the popularly used datasets constructed by HILA et al. (Chefer et al., 2023) for T2I task. Given that publicly available datasets only involve prompts with double objects combined by an "and" relationship, we further based on some of the 80 single objects in MSCOCO with the help of LLM (same as the datasets in Section 2.1). Followings introduces the details of the datasets.

- Template-Based Pairs (TBP): It is the public dataset constructed by HILA et al. (Chefer et al., 2023) used for T2I task. All the prompts in the dataset contain two objects that are constructed by three templates, i.e., "a [animalA] and a [animalB]", "a [animal] and a [color][object]", and "a [colorA][objectA] and a [colorB][objectB]". The placeholders in the templates are filled with 12 types of animals, 12 objects and 11 colors.

- Two/Three-Object Prompts (TwOP/ThreeOP): The detailed construction of our two datasets can be found in Section 2.1. The constructed datasets contain 3,160 prompts with two objects and the same number of prompts with three objects.

4.2 Subject Models

To investigate the performance of *Patcher* in repairing catastrophic-neglect issue. we introduce three T2I DMs (Stable Diffusion V1.4 (SD V1.4), Stable Diffusion V1.5 (SD V1.5), and Stable Diffusion V2.1 (SD V2.1)) for their wide adoption in community. All models are run on a 3090 GPU with 24GB of VRAM.

4.3 Evaluation Metric and Measurement Method

We adopt two evaluation metrics.

- *CLIPScore*: it measures the similarity between the input prompt and generated image, and is used in many previous studies (Hao et al., 2023; Chefer et al., 2023). However, it serves as a weaker indication of image-text similarity in T2I task, as correctness of generated images cannot be absolutely determined directly based on the magnitude of the value.

- *Correct Rate (CR)*: the percentage of correctly generated images out of all generated images. Compared to CLIPScore, CR is a direct measurement indicating whether a generated image is correct. For an image generated by T2I models, we manually judge whether it is correct by a annotation team consisting of one senior researcher and two Ph.D students. If more than half of the members perceive the generated image to be semantically consistent with the input prompt, we consider it as a correctly generated one.

4.4 Baselines

Our baselines include approaches based on prompt optimization (Promptist) and inference process optimization (AE). Besides above two baselines specific for T2I DMs, we have also specifically established a baseline that iteratively refines the output results through iterative queries (LR), which is a commonly-used strategy for performance improvement in the LLM context (Chao et al., 2023; Mehrotra et al., 2023).

Promptist (Hao et al., 2023) is the state-of-the-art approach to improve the generation quality of T2I DMs via prompt optimization. It first performs supervised fine-tuning with a pretrained language model on a small collection of manually engineered prompts. Then it defines a reward function that encourages the T2I DM to generate more aesthetically pleasing images while preserving the original prompt intentions. After that, it uses reinforcement learning with the reward function to further boosts performance of the fine-tuned model.

Attend-and-Excite (AE) (Chefer et al., 2023) is the state-of-the-art approach specific for catastrophic-neglect in T2I DMs via inference process optimization. Specifically, it adds an attention guidance mechanism during the model's inference stage to enhance the cross-attention units. This mechanism ensures that the model attends to all object tokens in the text prompts and boosts their activations, thereby encouraging the model to generate all objects described in the text prompts. However, AE requires prior knowledge of the positions of object tokens in the original prompts. For the input prompts, we use the object extraction method in *Patcher* to identify and return the positions of the objects within the prompts. Finally, the prompts, along with the positional information of the objects, are fed into the T2I DM enhanced by AE to generate optimized images.

LLM-Repair (LR) improves the quality of the generated images by the iterative query strategy that is commonly employed in practice to improve the outputs in the LLM context. Specifically, with the original prompt, LR first identifies the neglected objects in the generated employing the first stage in *Patcher*. After that, LR leverages GPT-3.5 to produce the new prompt describing the details of the neglected objects and asking for the T2I model to mitigate the catastrophic-neglect as much as possible in the next iteration (the prompt templates in shown in Appendix B). Then, LR iteratively query the T2I models until no object is identified as neglected one or reaching the maximum iteration number (set as 8 in our study).

5 Results

We designed two sets of experiments to explore the performance of *Patcher* in repairing catastrophic-neglect: the effectiveness of *Patcher* and the ablation study within *Patcher*.

5.1 Effectiveness of *Patcher*

Table 2 shows the effectiveness of *Patcher* and baselines in CR and CLIPScore. The column "Original" represents the quality of the images generated by different T2I DMs with the original prompts in the three datasets. The last four columns show the quality of the generated images after repair for three baselines and *Patcher* respectively. From the perspective of CR, *Patcher* achieves the best performance across all T2I models under testing and datasets, surpassing the baselines of 10.1%-16.3%. Especially on the last two datasets, TwOP and ThreeOP with more complex inter-object relationships or a greater number of objects, *Patcher* shows a more substantial improvement (31.8% higher than the original prompts and 12.4%-21.9% higher than the three baselines).

Table 2: The Correct Rate (CR) and the ClIPScore of the original prompts, *Patcher* and baselines.

Dataset	Model	Metric	Original	LR	Promptist	AE	*Patcher*
TBP	SD V1.4	*CR*	61.4%	75.0%	78.9%	83.6%	**89.8%**
		CLIPScore	32.0%	32.2%	32.2%	32.6%	**32.7%**
	SD V1.5	*CR*	55.1%	76.1%	78.2%	79.3%	**88.0%**
		CLIPScore	31.8%	32.0%	32.1%	**32.3%**	**32.3%**
	SD V2.1	*CR*	72.4%	84.4%	81.1%	85.4%	**96.0%**
		CLIPScore	32.8%	33.0%	32.7%	33.2%	**33.4%**
TwOP	SD V1.4	*CR*	45.6%	63.6%	53.8%	63.2%	**77.8%**
		CLIPScore	30.3%	30.5%	29.5%	30.6%	**30.7%**
	SD V1.5	*CR*	45.8%	67.9%	56.2%	68.2%	**78.0%**
		CLIPScore	30.1%	30.6%	29.4%	**30.7%**	**30.7%**
	SD V2.1	*CR*	49.6%	69.1%	63.8%	69.4%	**80.2%**
		CLIPScore	30.5%	30.7%	30.1%	**30.9%**	**30.9%**
ThreeOP	SD V1.4	*CR*	12.4%	28.6%	32.2%	29.0%	**41.0%**
		CLIPScore	31.2%	31.3%	29.7%	31.6%	**31.7%**
	SD V1.5	*CR*	13.4%	28.9%	32.2%	32.6%	**46.4%**
		CLIPScore	31.2%	31.3%	30.2%	**31.6%**	**31.6%**
	SD V2.1	*CR*	14.0%	30.1%	33.6%	34.3%	**48.2%**
		CLIPScore	31.3%	31.4%	30.4%	31.7%	**31.8%**

Compared to Promptist, *Patcher* achieves an CR improvement of 16.3%. Promptist automates the addition of modifiers at the end of the input prompts, such as "highly detailed", "masterpiece", or "sharp focus", to enhance the quality of the generated images. Adding such modifiers could help the T2I DM focus more on depicting the overall semantics of the prompt. However, in cases where there are significant feature differences between multiple objects, enhancing the T2I DM's focus on the entire sentence of the prompt could not effectively narrow the attention difference between different objects. It still requires the addition of appropriate modifiers to objects with weaker features. As for AE, it optimizes the inference process within T2I DMs rather than the prompts, which is supposed to be effective in principle but more difficult for end users to perform. However, *Patcher* still achieves superior performance compared to AE, with a CR improvement of 10.1%. As for LR, similar to *Patcher*, multiple attempts are needed to repair the prompts. Statistical analysis shows that LR requires an average of 5.7 attempts to correctly repair an image, whereas *Patcher* requires only 2.3 attempts. Additionally, *Patcher*'s CR exceeds LR by 14.2%, demonstrating the effectiveness of feature enhancement. The result also implies that if lacking guidance on feature enhancement, relying solely on the intrinsic capabilities of T2I MDs makes it difficult to effectively improve the accuracy of generated images.

For the CLIPScore, the results shows that the improvements are subtle. The reason is that for prompts containing multiple objects, the presence of some objects from the prompt in the generated image can still result in a high CLIPScore. Therefore, the similarity difference between correctly generated images and incorrectly generated images with respect to the original prompts is subtle. Furthermore, as we illustrated in Section 4.3, CLIPScore is a weak indicator with which we can not directly infer whether a generated image is correct or not. By comparison, CR together with the manual judgement is more suitable and direct to evaluate whether the catastrophic-neglect issue is mitigated or not. In general, the results demonstrate that *Patcher* significantly improves CR while maintaining CLIPScore compared to original dataset, demonstrating it's effectiveness.

5.2 Ablation Study

To investigate the effectiveness of the core component in *Patcher*, we conducted ablation experiments to explore the Correct Rate (CR) after removing Explicit Feature Enhancement (EFE) and Implicit Feature Enhancement (IFE) individually. The results, as shown in Table 3, show that both EFE and IFE significantly improve CR. Specifically, for datasets containing prompts with two objects, EFE

Table 3: The Correct Rate of the original prompts, Explicit Feature Enhancing (EFE), Implicit Feature Enhancing (IFE) and *Patcher*.

Dataset	Model	Original	EFE	IFE	*Patcher*
TBP	SD V1.4	61.4%	82.6%	79.3%	**89.8%**
	SD V1.5	55.1%	81.0%	74.2%	**88.0%**
	SD V2.1	72.4%	90.1%	83.7%	**96.0%**
TwOP	SD V1.4	45.6%	73.2%	59.0%	**77.8%**
	SD V1.5	45.8%	70.4%	61.6%	**78.0%**
	SD V2.1	49.6%	75.6%	66.4%	**80.2%**
ThreeOP	SD V1.4	12.4%	34.0%	24.6%	**41.0%**
	SD V1.5	13.4%	40.2%	27.2%	**46.4%**
	SD V2.1	14.0%	41.8%	29.5%	**48.2%**

and IFE achieve CRs of 78.8% and 70.7%, respectively, which are 23.9% and 15.8% higher than the CR of the original dataset. For datasets containing prompts with three objects, EFE and IFE achieve CR improvements of 25.4% and 13.8%, respectively, compared to the original dataset. This demonstrates the effectiveness of each component of *Patcher*. Moreover, combining EFE and IFE achieves a higher CR, indicating that the two components complement each other and that their combination can address a broader scope of catastrophic-neglect issue.

6 Related Work

6.1 Text-to-Image Diffusion Models

In recent years, the diffusion model has emerged as a more advanced and popular framework for text-to-image (T2I) generation compared to traditional non-diffusion methods like Variational Autoencoders (VAEs) (Yan et al., 2016; Mansimov et al., 2016) and Generative Adversarial Networks (GANs) (Zhu et al., 2019; Ye et al., 2021). Compared to GANs and VAEs, diffusion models achieve better results due to their stability during training and ability to progressively refine images, leading to higher quality and more detailed outputs (Ho et al., 2020; Nichol and Dhariwal, 2021) . To control the generation of diffusion models, Dhariwal and Nichol (2021) firstly propose a conditional image synthesis method utilizing classifier guidance, achieving great success in text-to-image generation. Following that, some representative studies (Bao et al., 2022; Ramesh et al., 2022b; Rombach et al., 2022; Saharia et al., 2022) of text-to-image diffusion models have been proposed, based on the conditioning mechanism. Our experiments are based on Stable Diffusion (Rombach et al., 2022) considering its wide applications.

6.2 Different Issue Types in T2I DM

With the rapid development of T2I DMs, researchers have primarily focused on two main aspects: safety issue and fundamental performance issue Zhai et al. (2023, 2024a,b); Liu et al. (2024); Borji (2023).

As for the issue in fundamental performance, Borji (2023) systematically discusses all existing issues in image generation but does not analyze the causes of the catastrophic-neglect issue in T2I models when prompts contain multiple object descriptions. According to the motivation, we discovered that catastrophic-neglect is the most prevalent issue (accounts for 94.0% in Error Rate) when prompts include multiple object descriptions. Liu et al. (2023) mentions the issue of object omission, assuming that specific action descriptions cause some objects to be missing in the image. Our proposal addresses object omission caused by inconsistent features among multiple objects, highlighting a different insight. Samuel et al. (2024) addresses the issue of text-to-image models generating incorrect objects for rare concepts. It focuses more on single objects, which is not consistent with the issue our approach aims to solve. Aithal et al. (2024) discusses the hallucination issue, where text-to-image generated images contain samples that have never existed in the training set. This type of hallucination issue is not related to the catastrophic-neglect issue we are addressing.

6.3 Optimizations For T2I DM

Some research efforts have focused on optimizing the inference process of T2I DMs. For instance, Liu et al. (2022); Feng et al. (2023); Chefer et al. (2023) have worked on improving the guidance mechanism through cross-attention, enabling T2I DMs to better focus on each object and attribute within the prompts, which helps in generating more accurate images. Additionally, there are works focusing on hand-crafted guidelines for prompt optimization. These studies involve selecting and composing prompts to generate images that achieve a distinct visual style and high quality (Liu and Chilton, 2022; Oppenlaender, 2022). Such approaches often rely on manual intervention and expert knowledge. To automate the construction of optimized prompts, Hao et al. (2023) propose an approach that combines supervised learning and reinforcement learning to train a prompt optimization model. The optimized prompts generated by this model are able to produce more aesthetically pleasing images and better adhere to the semantic content of the prompts. Just as large language models exhibit biases in their understanding of different words Li et al. (2024), T2I DMs face similar issues. This leads to the issue of unbalanced object characteristics when describing multiple objects. In this study, we focus on repairing catastrophic-neglect in T2I DMs by optimizing at the prompt level.

7 Conclusion

This paper proposes an approach (*Patcher*) to repair catastrophic-neglect in Text-to-Image Diffusion Models by attention-guided features enhancement of neglected objects in the generated images. *Patcher* first inputs the prompt into a T2I DM and an attention explainability tool to obtain the generated image and the attention scores for each token. It then checks whether all objects in the prompt appear in the generated image based on the text-image similarity. If any objects are neglected, *Patcher* iteratively searches for suitable explicit and implicit features to enhance the neglected objects based on the attention differences between the objects. Experimental results demonstrate that *Patcher* effectively addresses the issue of catastrophic-neglect in T2I DMs, achieving a 10.1%-16.3% higher Correct Rate based on manual annotation compared to baselines, as tested on Stable-Diffusion V1.4, V1.5, and V2.1 models. Additionally, ablation experiments show that both explicit feature enhancing and implicit feature enhancing in *Patcher* contribute to resolving the issue of catastrophic-neglect in T2I DMs.

Limitations

There are two limitations to the current study. Firstly, *Patcher* primarily repair the issue of catastrophic-neglect for objects and does not consider errors related to the attributes of the objects. Considering that attribute repairing requires the accurate generation of objects as a foundation and the catastrophic-neglect is a prevalent issue in T2I DMs, this work first attempts to repair object neglect. We will explore the repair of attribute neglect in future work.

Secondly, *Patcher* requires multiple iterations to identify suitable features for generating correct repaired prompts, which increases the time cost of repair. To mitigate this, we utilize attention differences to guide the search for the optimal features, and results show that, on average, only 2.3 iterations are needed to find the enhanced features that can correctly repair the image.

Acknowledgments

This work was supported by the National Natural Science Foundation of China Grant No.62232016, No.62072442, No.62272445 and No.62402484, Youth Innovation Promotion Association Chinese Academy of Sciences, Basic Research Program of ISCAS Grant No. ISCAS-JCZD-202304, and Major Program of ISCAS Grant No. ISCAS-ZD-202302.

References

Sumukh K Aithal, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. 2024. Understanding hallucinations in diffusion models through mode interpolation. *arXiv preprint arXiv:2406.09358*.

Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. 2022. All are worth words: a vit backbone for score-based diffusion models. *CoRR*, abs/2209.12152.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Ali Borji. 2023. Qualitative failures of image generation models and their application in detecting deepfakes. *Image and Vision Computing*, 137:104771.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *CoRR*, abs/2310.08419.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. 2023. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Trans. Graph.*, 42(4):148:1–148:10.

Prafulla Dhariwal and Alexander Quinn Nichol. 2021. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8780–8794.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun R. Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2023. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. 2023. Optimizing prompts for text-to-image generation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yuxi Li, Yi Liu, Gelei Deng, Ying Zhang, Wenjia Song, Ling Shi, Kailong Wang, Yuekang Li, Yang Liu, and Haoyu Wang. 2024. Glitch tokens in large language models: categorization taxonomy and effective detection. *Proceedings of the ACM on Software Engineering*, 1(FSE):2075–2097.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. 2022. Compositional visual generation with composable diffusion models. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XVII*, volume 13677 of *Lecture Notes in Computer Science*, pages 423–439. Springer.

Qihao Liu, Adam Kortylewski, Yutong Bai, Song Bai, and Alan Yuille. 2023. Discovering failure modes of text-guided diffusion models via adversarial search.

Vivian Liu and Lydia B. Chilton. 2022. Design guidelines for prompt engineering text-to-image generative models. In *CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022 - 5 May 2022*, pages 384:1–384:23. ACM.

Yi Liu, Guowei Yang, Gelei Deng, Feiyue Chen, Yuqi Chen, Ling Shi, Tianwei Zhang, and Yang Liu. 2024. Groot: Adversarial testing for generative text-to-image models with tree-based semantic transformation. *arXiv preprint arXiv:2402.12100*.

Elman Mansimov, Emilio Parisotto, Lei Jimmy Ba, and Ruslan Salakhutdinov. 2016. Generating images from captions with attention. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum S. Anderson, Yaron Singer, and Amin Karbasi. 2023. Tree of attacks: Jailbreaking black-box llms automatically. *CoRR*, abs/2312.02119.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR.

Jonas Oppenlaender. 2022. A taxonomy of prompt modifiers for text-to-image generation. arxiv. *arXiv preprint arXiv:2204.13988*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022a. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022b. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. 2024. Generating images of rare concepts using pre-trained diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4695–4703.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. 2023. What the DAAM: interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5644–5659. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. 2016. Attribute2image: Conditional image generation from visual attributes. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pages 776–791. Springer.

Hui Ye, Xiulong Yang, Martin Takác, Rajshekhar Sunderraman, and Shihao Ji. 2021. Improving text-to-image synthesis using contrastive learning. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, page 154. BMVA Press.

Shengfang Zhai, Huanran Chen, Yinpeng Dong, Jiajun Li, Qingni Shen, Yansong Gao, Hang Su, and Yang Liu. 2024a. Membership inference on text-to-image diffusion models via conditional likelihood discrepancy. *arXiv preprint arXiv:2405.14800*.

Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. 2023. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1577–1587.

Shengfang Zhai, Weilong Wang, Jiajun Li, Yinpeng Dong, Hang Su, and Qingni Shen. 2024b. Discovering universal semantic triggers for text-to-image synthesis. *arXiv preprint arXiv:2402.07562*.

Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. 2019. DM-GAN: dynamic memory generative adversarial networks for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5802–5810. Computer Vision Foundation / IEEE.

A Details of Explicit Feature Prompts

The details of the explicit feature prompts are illustrated below. In *Patcher*, we replace the placeholders in the following prompts with the neglected objects.

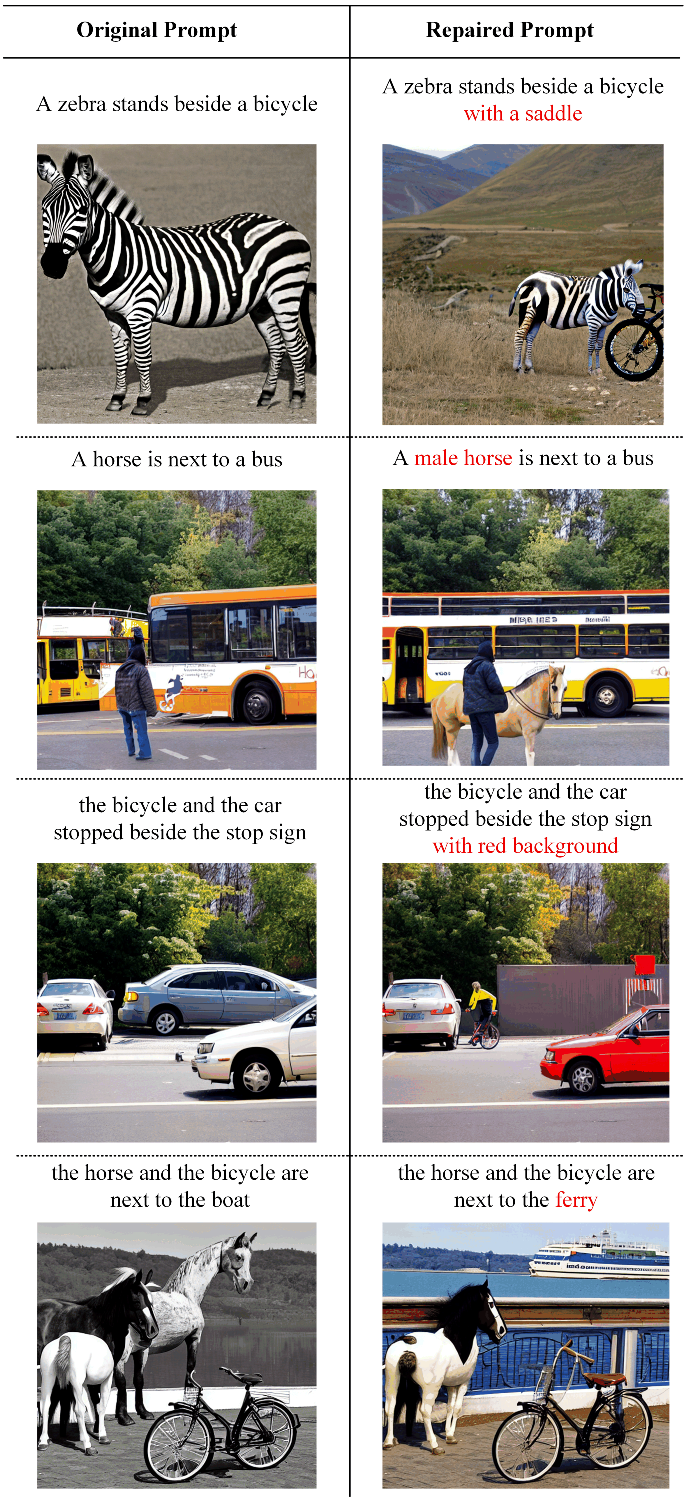
Shape Feature Prompt:
What are the common shapes of the ***[Neglected Object]***?
Please output the answer without explanation. There are two guidelines: 1) The output should add shapes to the neglected object to construct a fluent phrase, separating each phrase with a semicolon; 2) Each shape should originate from a distinct perspective.
Example:
Question: What are the common shapes of bicycle?
Output: two-wheeled bicycle; bicycle with pedals; bicycle with chain and gears

Color Feature Prompt:
What are the most common color of the ***[Neglected Object]***?
Please output the answer without explanation. There are two guidelines: 1) The output should add colors to the neglected object to construct a fluent phrase, separating each phrase with a semicolon; 2) Each color should originate from a distinct perspective.
Example:
Question: What are the most common colors of apple?
Output: red apple; green apple

B Details of The Prompt in LLM-Repair

The details of the prompt in LLM-Repair is illustrated below. In Patcher, we replace the placeholders in the following prompts with the input prompt and the neglected object.

LLM-Repair Prompt:
Input Prompt: ***[Input Prompt]***
The input prompt is fed into the Text-to-Image model. However, the ***[Neglected Object]*** is not shown on the generated image. Please repair the input prompt and output eight repaired prompt and and separating each prompt with a semicolon without explanation.



Original Prompt	Repaired Prompt
A zebra stands beside a bicycle	A zebra stands beside a bicycle with a saddle
A horse is next to a bus	A male horse is next to a bus
the bicycle and the car stopped beside the stop sign	the bicycle and the car stopped beside the stop sign with red background
the horse and the bicycle are next to the boat	the horse and the bicycle are next to the ferry

Figure 3: Images generated by original prompts and repaired prompts.

C Examples of Images Generated by Original Prompts and Repaired Prompts Derived from *Patcher*

Examples of the images generated from the original prompt and the repaired prompt generated by *Patcher* are shown in the figure 3.