Self-correcting LLM-controlled Diffusion Models

Tsung-Han Wu* Long Lian* Joseph E. Gonzalez Boyi Li† Trevor Darrell†
UC Berkeley

Figure 1. Existing diffusion-based text-to-image generators (e.g., DALL-E 3 [16]) generally struggle to precisely generate images that correctly align with complex input prompts, especially for the ones that require numeracy and spatial relationships. Our Self-correcting LLM-controlled Diffusion (SLD) framework empowers these diffusion models to automatically and iteratively rectify inaccuracies through applying a set of latent space operations (addition, deletion, repositioning, *etc*.), resulting in enhanced text-to-image alignment.

Abstract

Text-to-image generation has witnessed significant progress with the advent of diffusion models. Despite the ability to generate photorealistic images, current text-to-image diffusion models still often struggle to accurately interpret and follow complex input text prompts. In contrast to existing models that aim to generate images only with their best effort, we introduce Self-correcting LLM-controlled Diffusion (SLD). SLD is a framework that generates an image from the input prompt, assesses its alignment with the prompt, and performs self-corrections on the inaccuracies in the generated image. Steered by an LLM controller, SLD turns text-to-image generation into an iterative closed-loop process, ensuring correctness in the resulting image. SLD is not only training-free but can also be seamlessly integrated with diffusion models behind API access, such as DALL-E 3, to further boost the performance of state-of-the-art diffusion models. Experimental results show that our approach can rectify a majority of incorrect generations, particularly in generative numeracy, attribute binding, and spatial relationships. Furthermore, by simply adjusting the instructions to the LLM, SLD can perform image editing tasks, bridging the gap between text-to-image generation and image editing pipelines. Our code is available at: https://self-correcting-llm-diffusion.github.io.

*Equal contribution. †Equal advising.

1. Introduction

Text-to-image generation has made remarkable advancements, especially with the advent of diffusion models. However, these models often struggle with interpreting complex input text prompts, particularly those that require skills such as understanding the concept of numeracy, spatial relationships, and attribute binding with multiple objects. Despite the astonishing scaling of model sizes and training data, these challenges, as illustrated in Fig. 1, are still present in state-of-the-art open-source and proprietary diffusion models.

Several research and engineering efforts aim to overcome these limitations. For instance, methods such as DALL-E 3 [16] focus on the diffusion training process and incorporate high-quality captions into the training data at a massive scale. However, this approach not only incurs substantial costs but also frequently falls short in generating accurate images from complicated user prompts, as shown in Fig. 1. Other work harnesses the power of external models for a better understanding of the prompt in the inference process before the actual image generation. For example, [6, 11] leverages Large Language Models (LLMs) to pre-

process textual prompts into structured image layouts and thus ensures the preliminary design aligns with the user's directives. However, such integration does not resolve the inaccuracies produced by the downstream diffusion models, particularly in images with complex scenarios like multiple objects, cluttered arrangements, or detailed attributes.

Drawing inspiration from the process of a human painting and a diffusion model in generating images, we observe a key distinction in their approach to creation. Consider a human artist tasked with painting a scene featuring two cats. Throughout the painting process, the artist remains cognizant of this requirement, ensuring that two cats are indeed present before considering the work complete. Should the artist find only one cat depicted, an additional one would be added to meet the prompt's criteria. This contrasts sharply with current text-to-image diffusion models, which operate on an *open-loop* basis. These models generate images through a predetermined number of diffusion steps and present the output to the user, regardless of its alignment with the initial user prompt. Such a process, irrespective of scaling training data or LLM pre-generation conditioning, lacks a robust mechanism to ensure the final image aligns with the user's expectations.

In light of this, we propose our method **S**elf-correcting **LLM**-controlled **D**iffusion (**SLD**) that performs *self-checks* to confidently offer users guarantees of the alignment between the prompt and the generated images. Departing from conventional single-round generation methods, SLD is a novel *closed-loop* approach that equips diffusion models with the ability to iteratively identify and rectify errors. Our SLD framework, illustrated in Fig. 2, contains two main components: LLM-driven object detection as well as LLM-controlled assessment and correction.

The SLD pipeline follows a standard text-to-image generation setting. Given a textual prompt that outlines the desired image, SLD begins with calling an image generation module (*e.g.*, the aforementioned open-loop text-to-image diffusion models) to generate an image in a best-effort fashion. Given that these open-loop generators do not guarantee an output that aligns perfectly with the prompt, SLD then conducts a thorough evaluation of the produced image against the prompt, with an LLM parsing key phrases for an open-vocabulary detector to check. Subsequently, an LLM controller takes the detected bounding boxes and the initial prompt as input and checks for potential mismatches between the detection results and the prompt requirements, suggesting appropriate self-correction operations, such as adding, moving, and removing objects. Finally, utilizing a chosen base diffusion model (*e.g.*, Stable Diffusion [20]), SLD employs latent space composition to implement these adjustments, thereby ensuring that the final image accurately reflects the user's initial text prompt.

Notably, SLD is agnostic to the initial generation process, which allows us to use DALL-E 3 APIs as a subroutine to generate initial images and then employ the open-source Stable Diffusion model family to conduct latent operations to attain self-correction. Furthermore, none of these operations require additional training on our base diffusion model [20], which easily allows our method to be applied to various diffusion models without external human annotation or training.

We demonstrate that our SLD framework can achieve significant improvement over current diffusion-based methods on complex prompts with the LMD benchmark [11]. The results show that our method can surpass LMD+, which is a strong baseline that already leverages LLM in the image process generation, by 9.0%. More importantly, with DALL-E 3 for initial generation, the generated images from our method achieve 26.5% performance gains compared to ones before self-correction.

Finally, since the SLD pipeline is agnostic to the initially generated image, it can easily be transformed into an image editing pipeline by simply changing the prompts to the LLM. While text-to-image generation and image editing are often treated as distinct tasks by the generative modeling community, our SLD can perform these two tasks with a unified pipeline. **We list our key contributions below:**

1. SLD is the first to integrate a detector and an LLM to *self-correct* generative models, ensuring accurate generation without extra training or external data.
2. SLD offers a unified solution for both image generation and editing, enabling enhanced text-to-image alignment for any image generator (*e.g.*, DALL-E 3) and object-level editing on any images.
3. Experimental results show that SLD can correct a majority of incorrect generations, particularly in aspects of numeracy, attribute binding, and spatial relationships.

2. Related Work

2.1. Text-to-Image Diffusion Models

Diffusion-based text-to-image generation has advanced significantly. Initial studies [19–21] showed diffusion models' ability to create high-quality images, but they struggle with complex prompts. Subsequent research [3, 10, 18, 24–26] has incorporated additional inputs such as keypoints and bounding boxes to control the diffusion generation process.

Recent advancements have incorporated LLMs to control the generation of diffusion models, bypassing the need for additional complementary information as inputs [6, 7, 11–13, 27]. In these approaches, LLMs play a central role in directly interpreting user textual prompts and managing the initial layout configuration. Despite some progress, these models often operate in an open-loop fashion, producing images in one iteration that cannot guarantee the generated images align with user prompts.

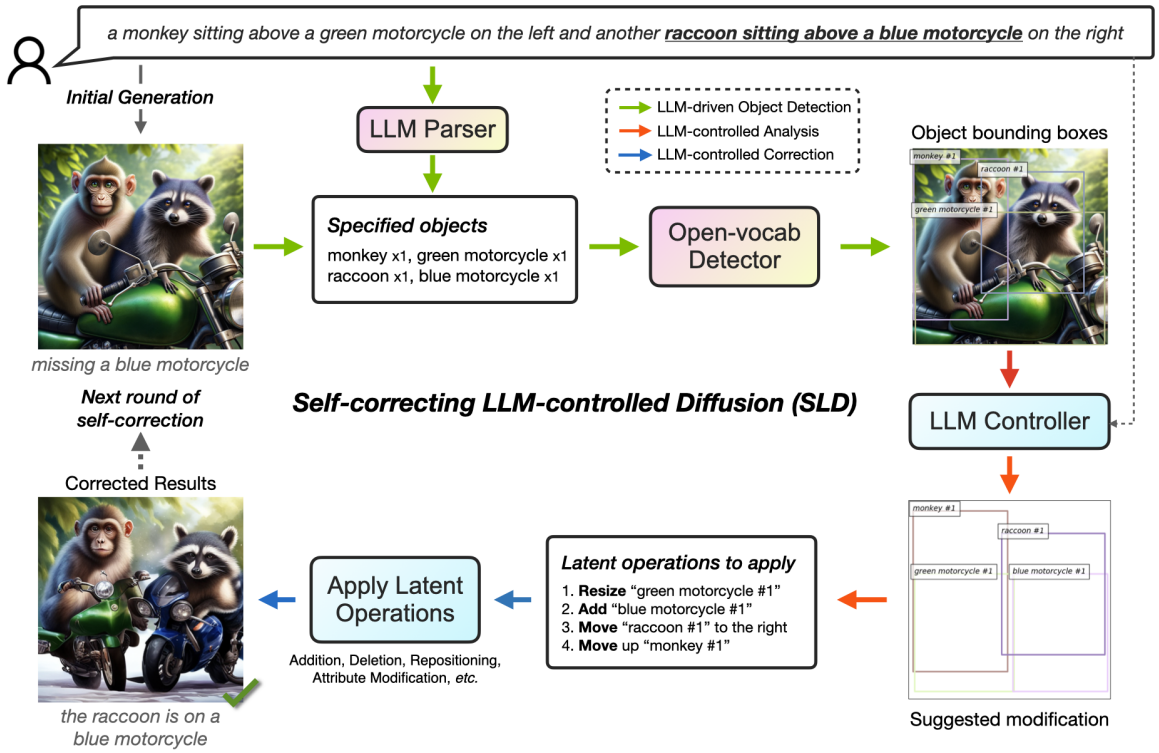

Figure 2. Our proposed Self-correcting LLM-controlled Diffusion (SLD) enhances text-to-image alignment through an iterative self-correction process. It begins with LLM-driven object detection (Sec. 3.1), and subsequently performs LLM-controlled analysis and correction (Sec. 3.2). The entire pipeline is outlined in Algorithm 1.

Unlike prior work, SLD is the first closed-loop diffusion-based generation method. Integrating advanced object detectors and LLMs, SLD performs iterative self-checking and correction, significantly enhancing text-to-image alignment. This improvement spans numeracy, attribute binding to multiple objects, and spatial reasoning, applicable to various models, including those like DALL-E 3. Also, SLD extends beyond existing NLP self-correction techniques [17] by incorporating bounding box spatial representation and novel latent editing mechanisms tailored for text-to-image tasks. These designs are essential for ensuring high-quality, precise self-correction in image generation tasks.

2.2. Diffusion-based Image Editing

Recent advancements in text-to-image diffusion models have significantly expanded their applications in image editing, encompassing both global and local editing tasks. Techniques like Prompt-2-prompt [8] and InstructPix2Pix [2] specialize in global edits, such as style transformations. Conversely, methods like SDEdit [15], DiffEdit [4], and Plug-and-Play [22] focus on local edits, targeting specific areas within images. Despite their progress, these methods often struggle with precise object-level manipulation and tasks that require spatial reasoning, such as resizing or repositioning objects. While recent approaches like Self-Guidance [5] offer fine-grained operations, they still necessitate user inputs for specific coordinates when moving or repositioning objects.

Unlike these methods only focus on diffusion models, SLD introduces the combination of detectors and LLMs in the loop of editing, enabling fine-grained editing with only user prompts. Also, SLD excels in various object-level editing tasks, including adding, replacing, moving, and modifying attributes, swapping, and so on, demonstrating a notable improvement in both ease of use and editing capabilities.

3. Self-correcting LLM-controlled Diffusion

In this section, we introduce our **S**elf-correcting **LLM**-controlled **D**iffusion (**SLD**) framework. SLD consists of two main components: LLM-driven object detection (Sec. 3.1) as well as LLM-controlled assessment and correction (Sec. 3.2). Moreover, with a simple change of the LLM instructions, we show that SLD is applicable to image editing, unifying text-to-image generation, and editing as discussed in Sec. 3.3. The complete pipeline is shown in Algorithm 1.

3.1. LLM-driven Object Detection

Our SLD framework starts with LLM-driven object detection, which extracts information necessary for downstream assessment and correction. As shown with green arrows in

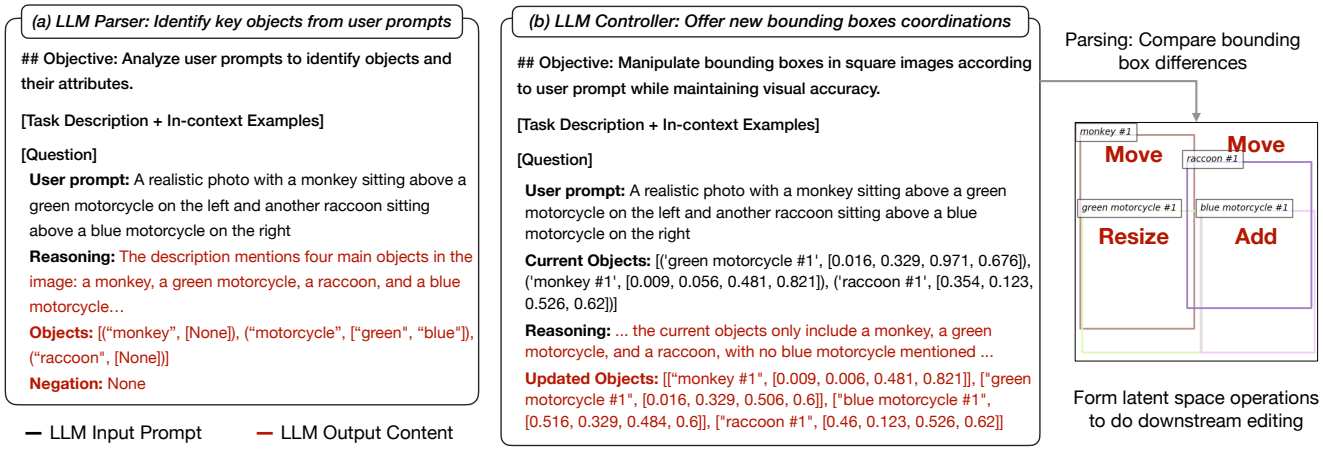

Figure 3. Our self-correction pipeline is driven by two distinct LLMs: **(a)** The LLM parser analyzes user prompts P to extract a list of key object information S, which is then passed to the open-vocabulary detector. **(b)** The LLM controller, taking both the user prompt P and currently detected bounding boxes B_{curr} as input, outputs suggested new bounding boxes B_{next}. These are subsequently transformed into a set of latent space operations Ops for image manipulation.

Fig. 2, the LLM-driven object detection includes two steps: **1)** We leverage an LLM as a parser that parses the user prompt and outputs key phrases that are potentially relevant to image assessment. **2)** The phrases are then passed into an open-vocabulary object detector. The detected boxes are supposed to contain information that supports the assessment of whether the image is aligned with the specifications in the user prompt.

In the initial step, an LLM parser is directed to extract a list of key object details, denoted as S, from the user-provided text prompt P. This parser, aided by text instructions and in-context examples, can easily accomplish this as shown in Fig. 3 (a). For a user prompt that includes phrases like "a green motorcycle" and "a blue motorcycle," the LLM is expected to identify and output "green" and "blue" as attributes associated with "motorcycle." When the prompt references objects without specific quantities or attributes, such as "a monkey" and "a raccoon," these descriptors are appropriately left blank. Importantly, the LLM's role is not limited to merely identifying object nouns; it also entails identifying any associated quantities or attributes.

In the second step, an open-vocabulary detector processes the list of key object information, S, parsed in the first step, to detect and localize objects within the image. We prompt the open-vocabulary object detector with queries formatted as `image of a/an [attribute] [object name]`, where the "attribute" and "object name" are sourced from the parser's output. The resulting bounding boxes, B_{curr}, are then organized into a list format like `[("[attribute] [object name] [#object ID]", [x, y, w, h])]` for further processing. A special case is when the prompt poses constraints on the object quantity. For cases where attributed objects (e.g., "blue dog") fall short compared to the required quantities, a supplementary count of non-attributed objects (e.g., "dog") is provided to provide context for the subsequent LLM controller deciding whether to add more "blue dogs" or simply alter the color of existing dogs to blue. We will explain these operations, including object addition and attribute modification, in greater detail in Sec. 3.2.1.

3.2. LLM-controlled Analysis and Correction

We use an LLM controller for image analysis and the subsequent correction. The controller, given the user prompt P and detected boxes B_{curr}, is asked to analyze whether the image, represented by objects bounding boxes, aligns with the description of the user prompt and offers a list of corrected bounding boxes B_{next}, as shown in Fig. 3 (b).

SLD then programmatically analyzes the inconsistencies between the refined and original bounding boxes to output a set of editing operations Ops, which includes addition, deletion, repositioning, and attribute modification. However, a simple set-of-boxes representation does not carry correspondence information, which does not allow an easy way to compare the input and the output layout of the LLM controller when multiple boxes share the same object name. For example, when there are two cat boxes in both the model input and the model output, whether one cat box corresponds to which cat box in the output layout is unclear. Rather than introducing another algorithm to guess the correspondence, we propose to let the LLM output correspondence with a very simple edit: we give an object ID to each bounding box, with the number increasing within each object type, as a suffix added after the object name. In the in-context examples, we demonstrate to the LLM that the object should have the same name and object ID before and after the proposed correction.

Algorithm 1 Self-correction for Image Generation.

Input: User prompt P, Initial generated image I, Maximum number of self-correction round K.

for $k \leftarrow 1$ to K **do**
  $S \leftarrow \texttt{LLM-Parser}(P)$
  $B_{curr} \leftarrow \texttt{Detector}(S)$
  $B_{next} \leftarrow \texttt{LLM-Analysis}(P, B_{curr})$
  $Ops \leftarrow \texttt{Diff}(B_{curr}, B_{next})$
  if $Ops \neq \emptyset$ **then** (i.e., $B_{next} \neq B_{curr}$)
    $I = \texttt{Correction}(Ops, B_{next}, B_{curr})$
  else
    break
  end if
end for

Output: Image I.

3.2.1 Latent Operations for Training-Free Image Correction

The LLM controller outputs a list of correction operations to apply. For each operation, we first transform the original image into latent features. Our approach then executes a series of operations Ops, such as addition, deletion, repositioning, and attribute modification, applied to these latent layers. It is worth noting that although there are only four operations, they are sufficient to handle a majority of misalignments naturally. We explain how each operation is performed below.

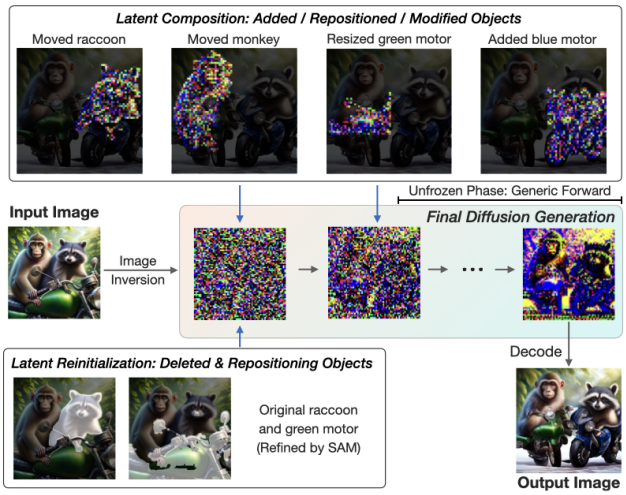

Figure 4. Our latent operations can be summarized into two key concepts: (1) latent in removed regions are re-initialized to Gaussian noise, and latent of newly added or modified objects are composited onto the canvas. (2) Latent composition is confined to the initial steps, followed by "unfrozen" steps for a standard forward diffusion process, enhancing visual quality and avoiding artificial copy-and-paste effects.

Addition. Inspired by [11], the addition process entails two phases: pre-generating an object and integrating its latent representation into the original image's latent space. Initially, we use a diffusion model to create an object within a designated bounding box, followed by precise segmentation using models (*e.g.*, SAM [9]). This object is then processed through a backward diffusion sequence with our base diffusion model, yielding masked latent layers corresponding to the object, which are later merged with the original canvas.

Deletion operation begins with SAM refining the boundary of the object within its bounding box. The latent layers associated with these specified regions are then removed and reset with Gaussian noise. This necessitates a complete regeneration of these areas in the final denoising process.

Repositioning involves modifying the original image to align objects with new bounding boxes, taking care to preserve their original aspect ratios. The initial steps include shifting and resizing the bounding box in the image space. Following this, SAM refines the object boundary, succeeded by a backward diffusion process to generate its relevant latent layers, similar to the approach in the addition operation. Latent layers corresponding to the excised parts are replaced with Gaussian noise, while the newly added sections are integrated into the final image composition. An important consideration in repositioning is conducting object resizing in the image space rather than the latent space to maintain high-quality results.

Attribute modification starts with SAM refining the object boundary within the bounding box, followed by applying attribute modifications such as DiffEdit [4]. The base diffusion model then reverses the image, producing a series of masked latent layers ready for final composition.

After editing operations on each object, we proceed to the recomposition phase as shown in Fig. 4. In this phase, while latents for removed or repositioned regions are reinitialized with Gaussian noise, the latents for added or modified latents are updated accordingly. For regions with multiple overlapping objects, we place the larger masks first to ensure the visibility of the smaller objects.

The stack of modified latent then undergoes a final forward diffusion process, which begins with steps in which regions not reinitialized with Gaussian noise are frozen (*i.e.*, forced to align with the unmodified latent at the same step). This is crucial for the accurate formation of updated objects while maintaining background consistency at the same time. The procedure finishes with several steps where everything is allowed to change, resulting in a visually coherent and correct image.

3.2.2 Termination of the Self-Correction Process

While we observe that one round of generation is often enough for a majority of the cases, subsequent rounds could still benefit the performance in terms of correctness further,

Method	Accuracy				
	Negation	Numeracy	Attribute	Spatial	Average
MultiDiffusion [1]	29%	28%	26%	39%	30.5%
Backward Guidance [3]	22%	37%	26%	67%	38.0%
BoxDiff [23]	22%	30%	37%	71%	40.0%
LayoutGPT + GLIGEN [6, 10]	36%	65%	26%	78%	51.3%
DALL-E 3 [16]	25%	38%	74%	71%	52.0%
+ 1-round SLD (Ours)	90%	61%	**80%**	83%	78.5% **(+ 26.5)**
LMD+ [11]	**100%**	82%	49%	86%	79.3%
+ 1-round SLD (Ours)	**100%**	**98%**	63%	**92%**	**88.3%** **(+ 9.0)**

Table 1. Our method can be applied to various image generation methods and improves the generation accuracy by a large margin.

making our self-correction an iterative process. Thus, determining the optimal number of self-correction rounds is critical for balancing efficiency and accuracy. As outlined in Algorithm 1, our method sets a maximum number of attempts on the correction rounds to ensure the process finishes within a reasonable amount of time.

The process completes when the LLM outputs the same layout as the input (*i.e.*, if the bounding boxes suggested by the LLM controller (B_{next}) align with the current detected bounding boxes (B_{curr})), or when the maximum rounds of generation are reached, indicating that the method is unable to make a correct generation for the prompt. This iterative process provides guarantees on the correctness of the image, up to the accuracy of the detector and the LLM controller, ensuring it aligns closely with the initial text prompt. We explore the efficacy of multi-round corrections in Sec. 4.3.

3.3. Unified text-to-image generation and editing

In addition to self-correcting image generation models, our SLD framework is readily adaptable for image editing applications, requiring only minimal modifications. A key distinction is in the format of user input prompts. Unlike image generation, where users provide scene descriptions, image editing requires users to detail both the original image and the desired changes. For instance, to edit an image with two apples and a banana by replacing the banana with an orange, the prompt could be: "Replace the banana with an orange, while keeping the two apples unchanged."

The editing process is similar to our self-correction mechanism. The LLM parser extracts key objects from the user's prompt. These objects are then identified by the open-vocabulary detector, establishing a list of current bounding boxes. The editing-focused LLM controller, equipped with specific task objectives, guidelines, and in-context examples, analyzes these inputs. It proposes updated bounding boxes and corresponding latent space operations for precise image manipulation.

SLD's ability to perform detailed, object-level editing distinguishes it from existing diffusion-based methods like InstructPix2Pix [2] and prompt2prompt [8], which mainly address global image style changes. Also, SLD outperforms tools like DiffEdit [4] and SDEdit [15], which are restricted to object replacement or attribute adjustment, by enabling comprehensive object repositioning, addition, and deletion with exact control. Our comparative analysis in Sec. 4.2 will further highlight SLD's superior editing capabilities over existing methods.

4. Experiments

4.1. Comparison with Image Generation Methods

Setup. We evaluate the performance of the SLD framework with the LMD benchmark [11], which is specifically designed to evaluate generation methods on complex tasks such as handling negation, numeracy, accurate attribute binding to multiple objects, and spatial reasoning. For each task, 100 programmatically generated prompts are fed into various text-to-image generation methods to produce corresponding images. We evaluate the images generated by our method and the baselines with open-vocabulary detector OWL-ViT v2 [14] for a robust quantitative evaluation of the alignment between the input prompts and the generated images. We compared SLD with several leading text-to-image diffusion methods, such as Multidiffusion [1], BoxDiff [23], LayoutGPT [6], LMD+ [12], and DALL-E 3 [16]. To ensure fair comparisons, all models incorporating LLMs used the same GPT-4 model. In our SLD implementation, we utilized LMD+ as the base model for latent space operations and OWL-ViT v2 for the open-vocabulary object detector.

Results. As shown in Tab. 1, applying the SLD method to both open-source (LMD+) and proprietary models (DALL-E 3) significantly enhances their performance in terms of generation correctness. **For negation tasks,** as LMD+ converts user prompts containing "without" information into negative prompts, which already achieves a remarkable 100% accuracy without SLD integration. In contrast, even

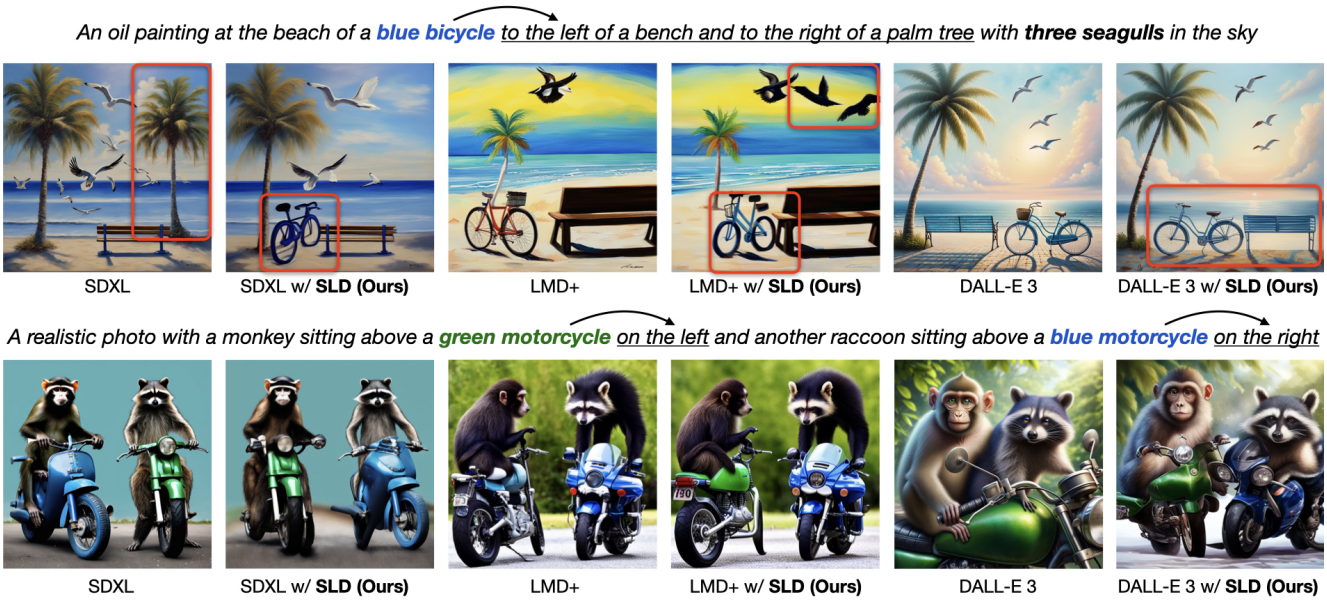

Figure 5. SLD enhances text-to-image alignment across diverse diffusion-based generative models such as SDXL, LMD+, and DALL-E 3. Notably, as highlighted by the red boxes in the first row, SLD precisely positions a blue bicycle between a bench and a palm tree, while maintaining the accurate count of palm trees and seagulls. The second row further demonstrates SLD's robustness in complex, cluttered scenes, effectively managing object collision through our training-free latent operations.

though DALL-E 3 also uses an LLM to rewrite the prompt, it still fails for some negation cases, likely because the LLM simply puts the negation keyword (*e.g.*, "without") into the rewritten prompt. In this case, our SLD method can automatically rectify most of these errors. **For numeracy tasks,** integrating SLD with LMD+ results in a significant improvement, with up to 98% accuracy. We noted that DALL-E 3 often struggles to generate an image with the correct number of objects. However, this issue is substantially mitigated by SLD, which enhances performance by over 20%. **For attribute binding tasks,** SLD improves the performance of both DALL-E 3 and LMD+ by 6% and 14%, respectively. Notably, DALL-E 3 initially outperforms LMD+ in this task, likely due to its training on high-quality image caption datasets. Finally, **for spatial reasoning tasks,** the integration of SLD with both LMD+ and DALL-E 3 demonstrates enhanced performance by 12% and 6%, respectively.

4.2. Application to Image Editing

As discussed in Sec. 3.3, SLD excels in fine-grained image editing over existing methods. As demonstrated in Fig. 6, our integration of an open-language detector with LLMs enables precise modifications within localized latent space regions. SLD adeptly performs specific edits, like seamlessly replacing an apple with a pumpkin, while preserving the integrity of surrounding objects. In contrast, methods like InstructPix2Pix [2] are confined to global transformations, and DiffEdit [4] often fails to accurately locate objects for modification, leading to undesired results.

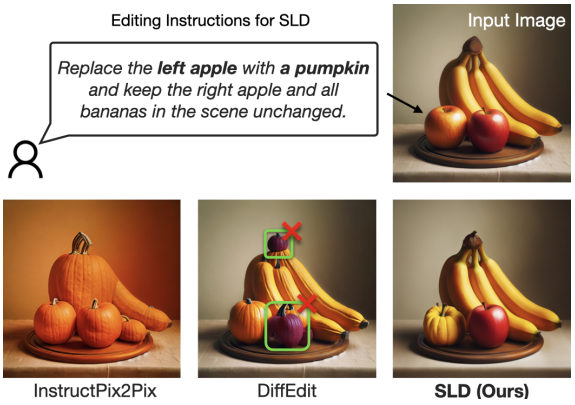

Figure 6. When instructed to perform object-level image editing. InstructPix2Pix [2] completely fails to accomplish the task, and DiffEdit [4] falls short, as highlighted in the green box of the image. Conversely, our method demonstrates a significantly better performance in executing these object-level edits.

Furthermore, as exemplified in Fig. 7, SLD supports a wide array of editing instructions, including counting control (such as adding, deleting, or replacing objects), attribute modification (like altering colors or materials), and intricate location control (encompassing object swapping, resizing, and moving). A standout example is featured in the "Object Resize" column of the first row, where SLD precisely enlarges the cup on the table by an exact factor of 1.25×. We encourage readers to verify this with a ruler for a clear demonstration of our method's precision. This level of pre-

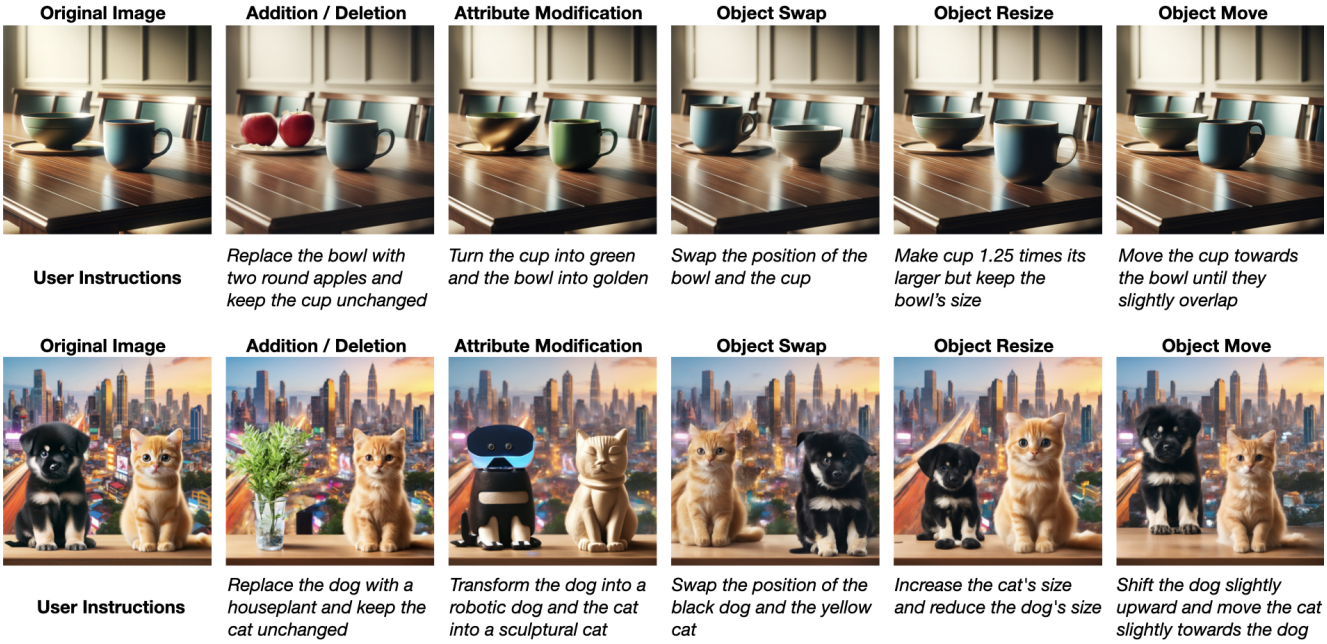

Figure 7. SLD can handle a diverse array of image editing tasks guided by natural, human-like instructions. Its capabilities span from adjusting object counts to altering object attributes, positions, and sizes.

Figure 8. SLD struggles with objects of complex shapes, as the SAM module may unintentionally segment adjacent parts during the process.

cision stems from the detector's exact object localization coupled with the LLMs' ability in reasoning and suggestions for new placements. Such detailed control over spatial adjustments is unmatched by any previous method, highlighting SLD's contributions to fine-grained image editing.

4.3. Discussion

Multi-round self-correction. Our analysis in Tab. 2 highlights the benefits of multi-round self-correction and the fact that the first round correction is always the most effective one and has a marginal effect. The first round of corrections substantially mitigates issues inherent in Stable Diffusion [20]. Then, a second round of correction still yields significant improvements across all four tasks.

Limitations and future work. A limitation of our method is illustrated in Fig. 8, where SLD fails to accurately remove a person's hair. In this instance, despite the successful identification and localization of the hair, the complex nature of its shape poses a challenge to the SAM module used for region selection, resulting in the unintended removal of the person's face in addition to the hair. However, since the person's cloth is not removed, the base diffusion model fails to generate a natural composition. This suggests that a better region selection method is needed for further improvements in the generation and editing quality.

Method	Accuracy					
	Negation	Numeracy	Attribute	Spatial	Average	
SD [20]	19%	38%	24%	33%	28.5%	
+ 1-round SLD	69%	55%	25%	69%	54.5%	(+ 26.0)
+ 2-round SLD	**73%**	**61%**	**31%**	**75%**	**60.0%**	(+ 31.5)

Table 2. While the majority of errors are typically rectified in the first round, multi-round correction consistently outperforms a single-round approach.

5. Conclusion

We introduce the Self-correcting Language-Driven (SLD) framework, a pioneering self-correction system using detectors and LLMs to significantly enhance text-to-image alignment. This method not only sets a new SOTA in the image generation benchmark but is also compatible with various generative models, including DALL-E 3. Also, SLD extends its utility to image editing applications, offering fine-grained object-level manipulation that surpasses existing methods.

References

[1] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023. 6

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 3, 6, 7

[3] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. *arXiv preprint arXiv:2304.03373*, 2023. 2, 6

[4] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *The Eleventh International Conference on Learning Representations*, 2023. 3, 5, 6, 7

[5] Dave Epstein, Allan Jabri, Ben Poole, Alexei A. Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *arXiv preprint arXiv:2306.00986*, 2023. 3

[6] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *arXiv preprint arXiv:2305.15393*, 2023. 1, 2, 6

[7] Hanan Gani, Shariq Farooq Bhat, Muzammal Naseer, Salman Khan, and Peter Wonka. LLM blueprint: Enabling text-to-image generation with complex and detailed prompts. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[8] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 3, 6

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 5

[10] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023. 2, 6

[11] Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*, 2023. 1, 2, 5, 6

[12] Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. Llm-grounded video diffusion models. *arXiv preprint arXiv:2309.17444*, 2023. 6

[13] Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. *arXiv preprint arXiv:2309.15091*, 2023. 2

[14] Neil Houlsby Matthias Minderer, Alexey Gritsenko. Scaling open-vocabulary object detection. *NeurIPS*, 2023. 6

[15] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 3, 6

[16] OpenAI. Dall·e 3 system card, 2023. 1, 6

[17] Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*, 2023. 3

[18] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[19] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 8

[21] 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. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[22] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023. 3

[23] Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7452–7461, 2023. 6

[24] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 2

[25] Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, et al. Reco: Region-controlled text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14246–14255, 2023.

[26] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2

[27] Tianjun Zhang, Yi Zhang, Vibhav Vineet, Neel Joshi, and Xin Wang. Controllable text-to-image generation with gpt-4. *arXiv preprint arXiv:2305.18583*, 2023. 2