

# Poetry2Image: An Iterative Correction Framework for Images Generated from Chinese Classical Poetry

Jing Jiang<sup>1\*</sup>, Yiran Ling<sup>1\*</sup>, Binzhu Li<sup>1\*</sup>, Pengxiang Li<sup>1\*</sup>, Junming Piao<sup>1</sup>, Yu Zhang<sup>1,2†</sup>

<sup>1</sup>Harbin Institute of Technology

<sup>2</sup>Research Center for Social Computing and Information Retrieval (SCIR)

{2021110679, 2021110742, 2021112888, 2021110869, 2022111726}@stu.hit.edu.cn

{zhangyu@ir.hit.edu.cn}@hit.edu.cn

## Abstract

Text-to-image generation models often struggle with key element loss or semantic confusion in tasks involving Chinese classical poetry. Addressing this issue through fine-tuning models needs considerable training costs. Additionally, manual prompts for re-diffusion adjustments need professional knowledge. To solve this problem, we propose Poetry2Image, an iterative correction framework for images generated from Chinese classical poetry. Utilizing an external poetry dataset, Poetry2Image establishes an automated feedback and correction loop, which enhances the alignment between poetry and image through image generation models and subsequent re-diffusion modifications suggested by large language models (LLM). Using a test set of 200 sentences of Chinese classical poetry, the proposed method—when integrated with five popular image generation models—achieves an average element completeness of 70.63%, representing an improvement of 25.56% over direct image generation. In tests of semantic correctness, our method attains an average semantic consistency of 80.09%. The study not only promotes the dissemination of ancient poetry culture but also offers a reference for similar non-fine-tuning methods to enhance LLM generation.

## 1 Introduction

Text-to-image generation combines natural language understanding with image generation models, which synthesize realistic images conditioned on natural language descriptions. When text-to-image generation models deal with prompts requiring professional knowledge, such as Chinese classical poetry, they are prone to losing key elements or causing semantic confusion. It is challenging to accurately describe the precise meaning of poetry as illustrated in Fig. 1.

<sup>1</sup>\*Equal contribution. <sup>2</sup>†Email corresponding.



Figure 1: Direct text-based image generation often results in losing key elements in the image. Our method addresses this issue by implementing targeted image corrections, effectively capturing the semantics and artistic essence conveyed by the poem.

Some existing works have made efforts to alleviate these problems. One solution (Avrahami et al., 2022; Hertz et al., 2022) focuses on image editing to refine the generated images, but they suffer from complicated prompts or image understanding. Researchers also use Lora lightweight fine-tuning (Hu et al., 2021), retrieval-augmented generation from external knowledge database (Gao et al., 2023), and specialized poetry models such as Jiuge (Zhipeng et al., 2019; Deng et al., 2020; Yi et al., 2020) to

construct poetry-specific models. These methods, however, result in additional training costs and limited compatibility between models.

*Can external knowledge database be incorporated to edit the generated images and alleviate inconsistency between poetry and image?*

In this work, we introduce Poetry2Image, an iterative correction framework for image generation from Chinese classical poetry. This method identifies key elements in the initial generated image and employs text-guided image editing to alleviate inconsistency between poetry and image. Distinct from the conventional open-loop generation approach, our method presents a closed-loop generation process capable of iteratively refining the initial image.

Initially, the retrieval system searches the input poetry in the poetry database and returns its translation and appreciation. Subsequently, an initial image is generated from the translation. The large language model (LLM) extractor is then employed to extract key elements. The initial image and key elements are simultaneously input to the Open Vocabulary Detector to obtain information about elements in the initial image. Through the element information, the LLM suggester provides modification suggestions presented as a box selection in the image. Image editing models apply these suggested modifications to edit the initial image. Finally, the process above will be iterated multiple times to improve the consistency between poetry and image till no more suggestions.

Notably, Poetry2Image has no constraints on text-to-image generation models utilized for initial image generation. Furthermore, iterative correction operations eliminate the need for additional training costs, while the automated image generation and feedback process significantly reduces manual annotation. **The main contributions of this study can be summarized as follows:**

1. We introduce an Iterative Correction Framework for images generated from Chinese classical poetry, alleviating the loss of key elements and semantic confusion.
2. The proposed method is not only compatible with mainstream text-to-image generation models (e.g. DALL-E) but also training-free.
3. We discuss the generalization capabilities and limitations of adopting external knowledge databases for image generation, which provides a reference for similar non-fine-tuning methods to enhance LLM generation.

## 2 Related Works

### 2.1 Text-to-Image Generation

Text-to-image generation is the task of synthesizing images conditioned on natural language prompts. Recent advancements in diffusion models (Sohl-Dickstein et al., 2015; Dhariwal and Nichol, 2021; Song et al., 2020) have significantly improved the quality of text-to-image generation, such as Dreambooth (Ruiz et al., 2023) and DALL-E 3 (Betker et al., 2023). Despite their impressive visual quality, these models struggle with complex prompts, which tend to generate images lacking core semantic elements and cause semantic confusion (Feng et al., 2022; Lian et al., 2023; Bar-Tal et al., 2023). Some recent studies (Xie et al., 2023; Yang et al., 2023; Lian et al., 2023) incorporate bounding boxes as conditional controls to the diffusion process. Several recent papers (Huang et al., 2023; Xu et al., 2024; Fang et al., 2023) leverage image understanding feedback, which builds a general-purpose reward model to refine diffusion models for text-image alignment. Despite their progress, there are two limitations in handling image generation with complex prompts: (i) open-loop generation in a single iteration cannot guarantee the alignment between generated images and prompts; (ii) these methods result in additional training costs. To address these issues, we introduce a training-free cyclic self-correction framework to enhance the alignment of images with complex prompts.

### 2.2 Text-Guided Image Editing

Text-guided image editing synthesizes images from a given image and text descriptions. Classic image editing aims at fine-grained manipulation by inpainting masked regions while keeping the remaining areas. Studies (Avrahami et al., 2022; Meng et al., 2021) show that using user-generated masks for spatial editing in image generation is a straightforward yet effective method. Another method (Balaji et al., 2022; Hertz et al., 2022) focuses on predicted masks for spatial editing, demonstrating that manipulating image-text cross-attention masks is also effective. Evolved from spatial editing, text-guided image editing (Brooks et al., 2023; Kawar et al., 2023) accepts direct commands, allowing editing without regional masks. Despite some progress, these works mainly focus on diffusion models but often suffer from complicated prompts or image understanding. Recent advancements have demonstrated the capabilities of incor-

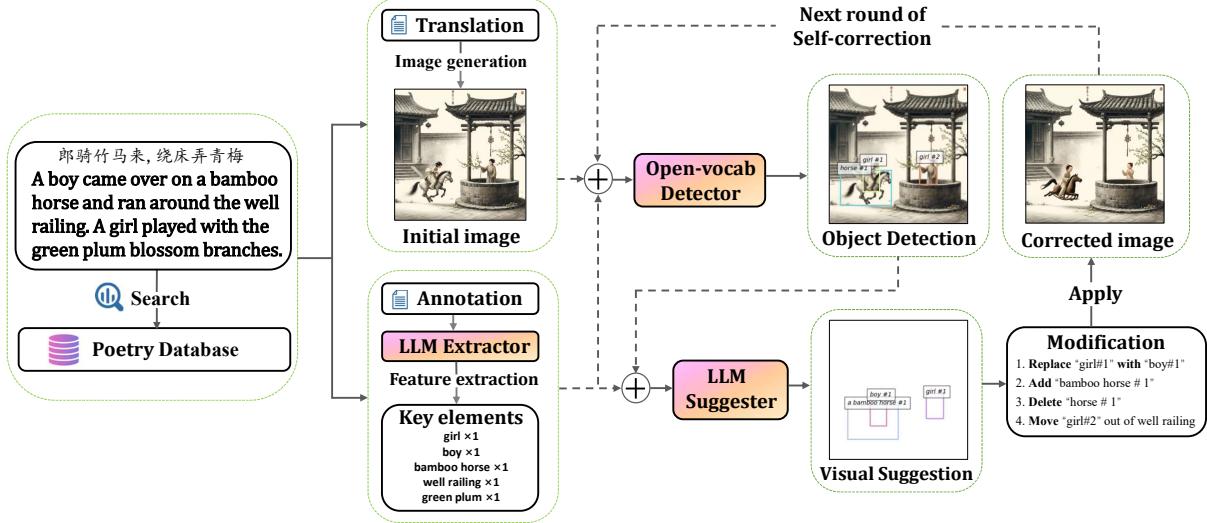


Figure 2: Automated iterative correction framework for images generated from poetry. Utilizing a pre-built poetry dataset, the process begins with the extraction of the poetry and generation of an initial image, followed by the implementation of a self-feedback image correction iteration loop. The loop functions by analyzing the semantics of the poem text and the image elements identified by Open Vocabulary Detector (OVD), utilizing LLM. It then outputs correction suggestions that guide the diffusion models for image editing, continuously providing feedback to progressively align the text semantics with the image semantics.

porating external language (Brooks et al., 2023) or vision (Kirillov et al., 2023) pre-trained models for editing. These methods, however, struggle with fine-grained manipulation according to user-provided texts when editing images in a single iteration.

### 3 Method

In this section, we introduce the iterative correction framework for poetry generation, as shown in Fig. 2. Compared to common texts, poetry is semantically implicit. Firstly, the implicit semantic elements should be extracted. Secondly, an image semantic error correction mechanism should be established to alleviate potential semantic inconsistencies in the images.

#### 3.1 Extract Implicit Semantics Based on LLM

**Dataset construction** should consider the meanings of poems, completeness of translation annotations, and cultural popularity. We consider rhetorical techniques involved in literature with semantic implicit features such as metaphor, personification, hyperbole, and allusion. Then, we allocate 200 well-known sentences with their modern Chinese translations, keyword annotations, and phrase explanations from the largest public platform of Chinese classical poetry, GuShiWen.com<sup>1</sup>.

<sup>1</sup> <https://www.gushiwen.cn/>

---

#### Algorithm 1 Key Elements Extraction

---

**Input:** Poetry  $p$ ; Poetry Database  $S$

- 1:  $d_{min} \leftarrow 0$
- 2:  $p_{find} = \emptyset$  // Query list initialization
- 3: **for**  $i = 1$  to  $N$  **do**
- 4:      $d = F_{similarity}(p, S[i])$
- 5:     **if**  $d \leq d_{min}$  **then**
- 6:          $d_{min} \leftarrow d$
- 7:          $p_{find}.append(S[i])$
- 8:     **end if**
- 9: **end for**
- 10:  $t_{find} = F_{Translation}(p_{find})$
- 11:  $n_{find} = F_{Annotation}(p_{find})$
- 12:  $E_{key} = LLM_{extract}(p_{find}, t_{find}, n_{find})$
- 13: **Initial image:**  $P_{origin} = Diff(t_{find})$

**Output:** Key elements  $E_{key}$ ; Initial image  $P_{origin}$

---

**Data extension** involves processing the dataset to match the input features required for detection and ensuring generality for prompts from different image generation models. Key elements of the poetry are extracted along with their translations, appreciations, and annotations, to facilitate monitoring of the elements completeness in the generated images. To automate the extraction process and achieve high extraction accuracy, we use GPT-4 for key element extraction, and design prompts for the LLM, as illustrated in Fig. 3.

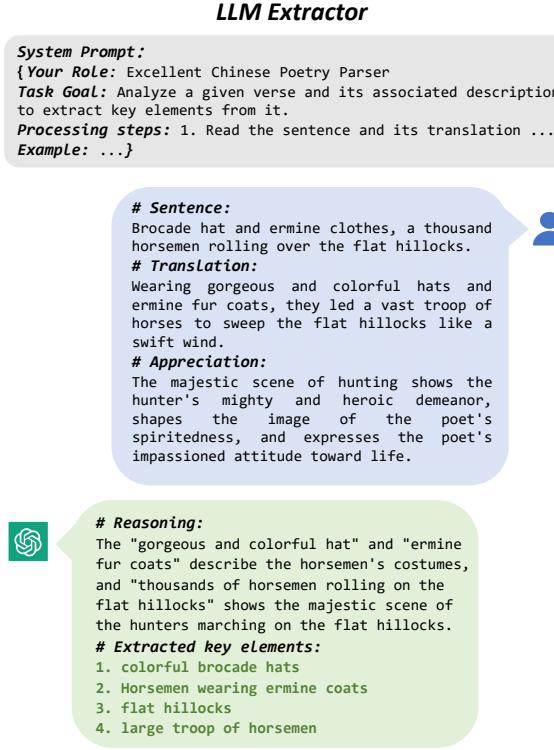


Figure 3: An illustration of the LLM Extractor, a key element extraction module. Upon retrieving the poem’s translation and critical appreciation from the poetry database, these texts along with the system prompt are fed into the LLM. Subsequently, the LLM outputs the key elements contained in the poetry.

We use the extracted key elements as supervisory text for subsequent image editing. The procedure is shown in Algorithm 1.

$P$  denotes collections of images, distinguished by subscripts.  $d$  measures semantic distance, ranging from 0 to 1.  $F_{similarity}$  calculates semantic cosine similarity.  $F_{translation}$  searches for poetry translations, denoted by  $t_{find}$ .  $F_{Annotation}$  searches for poetry annotations, denoted by  $n_{find}$ .

Additionally, we conduct manual element extraction on the poem and compare these results with those extracted by LLMs to validate the effectiveness of our methodology.

### 3.2 Automated Iterative Correction Framework

**Initial image generation** focuses on using translations of poetry as inputs for generating images instead of original poems. This approach ensures the images accurately reflect the poems’ meanings, avoiding ambiguities caused by historical linguistic changes and complex rhetorical devices such as metaphors and personifications.

**Detecting and correction** involves identifying where the key elements of the image are located. We use the Open Vocabulary Detector (OVD), an open-source recognition method based on an open corpus, to build the front part of our image correction component. The input to this part includes the initial generated images, and the recognition labels derived. After the OVD performs these extractions, feedback suggestions on the bounding boxes is generated, which will be transmitted to LLM for analysis in the form of labels and region annotations, as illustrated in Fig. 4. The LLM suggester provides modification suggestions and proposes a new box for the image elements. The extracted elements need to be compared with the labels of the elements in the bounding box to detect whether complete and correct elements are in the initial generation of the diagram. Algorithm 2 shows the procedure in detail.

---

#### Algorithm 2 Image Feedback Correction

---

**Input:** Key elements  $E_{key}$ ; Initial image  $P_{origin}$

- 1: **key elements detection:**  $E' \leftarrow \emptyset$
- 2:  $P_{feedback} \leftarrow P_{origin}$
- 3: **while**  $E' \neq E_{key}$  **do**
- 4:      $L' = OVD(P_{feedback})$
- 5:      $L'' = LLM_{suggester}(L', E_{key})$
- 6:      $L''' = LLM_{transform}(L'')$
- 7:      $P_{feedback} = Diff(L''')$
- 8:      $E' \leftarrow OVD(P_{feedback})$
- 9: **end while**
- 10: **return**  $P_{out} \leftarrow P_{feedback}$

**Output:** Final generated image  $P_{out}$

---

$L$  represents a list of intermediate calculation results.  $E$  represents a list of key elements in the semantics of poetry.

For all the bounding boxes in the image, the element labels are compared with the results from the LLM Extractor to determine whether a bounding box need to be retained or modified. This is discussed in the following scenarios:

1. Retain: Keep the bounding box unchanged if it is included in the LLM Extractor’s result, which is key elements.
2. Remove: Delete the bounding box based on the LLM analysis of the poetic imagery.
3. Add: If an key element from the LLM Extractor’s result is missing in the current generated image, the LLM selects a new rectangular area and adds the missing key element label based on the poetry translation.

### LLM Suggester

**System Prompt:**

{Your Role: An Expert in Bounding Box Adjustment for Images  
**Task Goal:** Adjust the bounding box list in a image, output the modified bounding box list.  
**Bounding Box Specifications:** 1. Image Coordinates: define ...  
**Necessary Information:** 1. Image Description: describes the ...  
**Key Guidelines:** 1. Alignment: Updated Objects should cover ...  
**Processing steps:** 1. Interpret the prompts: Carefully read ...  
**Example:** ...}

**# User Prompt:**

-**Image Description:** The Incense Burner Peak in the sunshine of the purple haze, see waterfalls like white silk hanging in front of the mountain from a distance.  
-**Elements that must be included:** ['sunshine', 'peak', 'purple haze', 'waterfall']  
**# Current Objects:**  
[('peak #1',[0.021,0.983,0.949,0.389]),('incense burner #1',[0.341,0.269,0.188,0.189]),('sunshine #1',[0.407,0.80,0.15,0.114]),('waterfall #1',[0.390,0.724,0.191,0.354]),('purple haze #1',[0.405,0.562,0.570,0.291])]



**# Reasoning:**

The Image Description doesn't depict the incense burner. The 'incense burner #1' appears due to a misunderstanding of the Incense Burner Peak, so 'incense burner #1' should be removed to correct the misunderstanding.

**# Updated Objects:**

[('incense burner peak #1',[0.021,0.983,0.949,0.389]), ('sunshine #1',[0.407,0.80,0.15,0.114]), ('waterfall #1',[0.390,0.724,0.191,0.354]), ('purple haze #1',[0.405,0.562,0.570,0.291])]

Figure 4: An example of the LLM Suggester, a module dedicated to modifying image bounding boxes. After conducting OVD-based element recognition to determine the existing bounding box, the translation, this bounding box, and the system prompt are inputted into the LLM. The LLM then adjusts the bounding box based on the semantic information in the translation, outputting the modified bounding box.

4. Move: If there is a positional conflict between bounding boxes in the current generated image, the LLM selects a new position, deletes the bounding box in the origin position, and regenerates it in the new position.

5. Replace: If a bounding box in the current image conflicts with the LLM Extractor's result, the LLM deletes the element and adds a proper element in the original position.

After prompt generation, we obtain suggested modifications, which can simplify the issue into a standard text-based editing task. We select the appropriate open-source diffusion models, input the suggested modifications, and use the bounding boxes and labels from the LLM suggester to guide SAM (Kirillov et al., 2023) for semantic segmentation, completing a round of image modifications.

**Recycle and Finish** involves determining if the image correctly contains all key elements. We first detect all elements in the initial correction images

and generate border images. These border images and element words are then input into the LLM Suggester. If the LLM Suggester provides new modification suggestions, they are applied to generate subsequent correction images, and the process repeats. If no new suggestions are provided or the loop reaches the preset limit, the image and text are deemed consistent, and the loop exits, yielding the final result.

**Evaluation** involves the elemental completeness and the semantic consistency in the poetry generated image, and we establish an image-text consistency evaluation model, as shown in Eq. 1.

$$\arg \max_{s,e} \Theta = \frac{\alpha \left( \frac{S}{s_\epsilon} \right) + \beta \left( \frac{e}{e_\epsilon} \right)}{\alpha + \beta} \quad (1)$$

$\Theta$  is the quantitative measure for assessing generated images of ancient poems, considering semantic features  $s$  and key elements  $e$ , with upper thresholds  $s_\epsilon$  and  $e_\epsilon$  respectively. Linear parameters  $\alpha$  and  $\beta$  dictate the focus:  $\beta = 0$  evaluates semantic compliance, while  $\alpha = 0$  evaluates key elemental completeness.

Model	Average Similarity	Rank
GPT-4-Turbo	0.8740	4
GLM-4	0.8763	3
<b>Claude-3</b>	<b>0.8868</b>	<b>1</b>
GPT-3.5-Turbo	0.8660	5
ERNIE-4.0	0.8834	2

Table 1: Evaluation of Element Extraction Effectiveness of Various Large Language Models. According to this evaluation, the Claude-3 model exhibits the highest effectiveness in key element extraction.

## 4 Experiment

### 4.1 Key Elements Extraction

In our image generation process, the initial stage uses LLM Extractor to semantically extract key elements from the database corpus. The accuracy of LLM Extractor is crucial to the subsequent process and needs to be evaluated in detail.

**Settings.** We select a dataset of 200 Chinese poems with implicit semantics and manually annotated them to establish a benchmark for element extraction. The poems are then processed using five LLMs: GPT-4-Turbo (Achiam et al., 2023), GPT-3.5-Turbo (Brown et al., 2020), Claude-3 (Anthropic, 2024), GLM-4 (Zeng et al., 2023), and

Method	Elemental Completeness	Semantic Consistency
DALL-E	56.33%	81.94%
<b>DALL-E+ours</b>	<b>90.20% (+33.87%)</b>	<b>84.18% (+2.24%)</b>
CogView	50.69%	77.77%
CogView+ours	67.28% (+17.59%)	78.82% (+1.05%)
Wenxin Yige	33.17%	80.95%
<b>Wenxin Yige+ours</b>	<b>64.76% (+31.58%)</b>	<b>81.77% (+0.82%)</b>
Stable Diffusion	37.71%	72.25%
Stable Diffusion+ours	63.12% (+25.41%)	73.87% ( <b>+1.62%</b> )
Midjourney	48.45%	80.06%
Midjourney+ours	67.78% (+19.33%)	81.79% ( <b>+1.73%</b> )

Table 2: Comparison with Image Generation Models. Our method shows a significant improvement in elemental completeness through image generation models. For elemental completeness, the accuracy improvement ranges from 17.59% to 33.87%, and for semantic consistency, it also achieves a certain degree of improvement..

ERNIE-4.0 (Sun et al., 2019). To assess the effectiveness of key element extraction by these LLM Extractors, the BERT-based-Chinese model (Devlin et al., 2019) was employed. The similarity between the manually annotated key elements and elements extracted by the LLMs served as a quantitative performance indicator.

**Results.** Based on the cosine similarity evaluation, effectiveness scores for element extraction were shown in Tab. 1. In terms of semantic understanding, Claude-3 exhibited the highest performance with a score of 0.8868, closely followed by ERNIE-4.0. GPT-4-Turbo and GLM-4 demonstrate comparable performance, whereas GPT-3.5-Turbo shows marginally reduced accuracy. Given its superior performance, Claude-3 is also employed as the LLM Extractor in experiments that do not involve LLM tuning. Overall, the five LLMs tested within our method achieve an element extraction accuracy exceeding 0.85, demonstrating notable consistency. Therefore, we contend that our method substantiates the use of LLMs for extracting key elements in Chinese poems, providing a robust foundation for subsequent processes.

#### 4.2 Verification of Self-Correcting Cycles Across Different Generation Models

**Settings.** We evaluate our image error correction method using elemental completeness and semantic consistency. Poetry2Image is applied to five text-to-image generation models to validate the effect: DALL-E-3 (Peebles and Xie, 2023), CogView3 (Zheng et al., 2024), Midjourney, Wenxin Yige,

and Stable Diffusion (Esser et al., 2024). Each model is assessed using a dataset of 200 Chinese poems with complex semantics, as shown in Fig. 5. Utilizing Open-vocabulary Detector OWL-ViT v2 (Minderer et al., 2024), we quantify key elements in both the initial and corrected images to determine elemental completeness. Employing BERT-Chinese (Devlin et al., 2019), we measure semantic consistency by comparing the image content with the corresponding translation.

**Result.** The full-process evaluation results are shown in Tab. 2. In the key elemental completeness test, Peotry2Image achieved an accuracy improvement ranging from 17.59% to 33.87%. Moreover, the elemental completeness of DALL-E has reached 90.20%, demonstrating good performance. In the semantic correctness test, the average semantic consistency reached 81.64%. Peotry2Image maintained stability in semantic consistency, indicating that it meets the standards for consistency between image content and poem sentiment. This stability also suggests that the improvement in image quality primarily results from enhanced element integrity.

#### 4.3 Comparison of Iteration Rounds

**Settings.** To ensure the improvement in elemental completeness and ascertain the maximum efficacy of our method, we conduct an iterative comparison experiment. Observation points are established within the automated process, sequentially recording improvements in elemental completeness as the number of image iteration rounds increased.

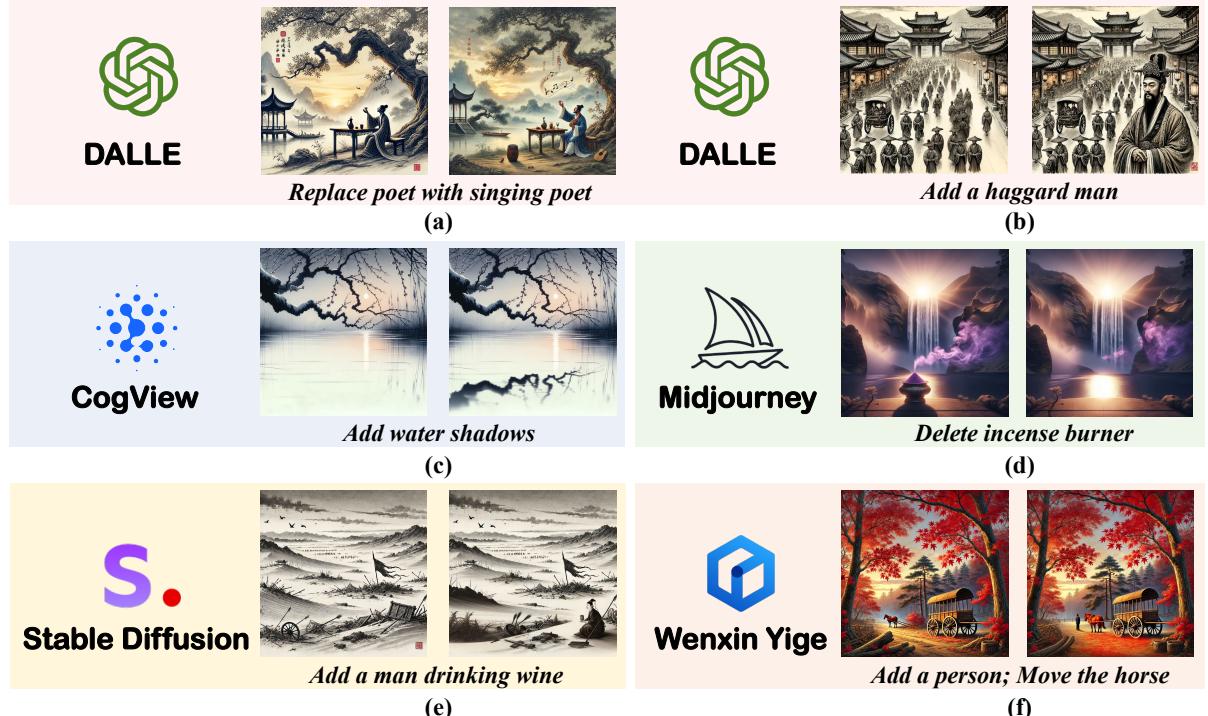


Figure 5: Image generation effect of the whole process evaluation. Peotry2Image enhances image generation quality for specialized texts like classical poetry and addresses core issues such as morpheme loss and semantic confusion. The poems corresponding to the images can be found in Appendix B.

**Result.** Based on the experimental outcomes as shown in Tab. 3, the following conclusions can be drawn:

1. With increasing image iteration rounds, elemental completeness improves, achieving a notable increase of **27.30%** by the first round.
2. The elemental completeness of the images ultimately stabilizes at approximately **90%** around 3 rounds of iterations, demonstrating our method’s effectiveness in accurately correcting and redrawing most ideal elements.

Round	Elem. Completeness	Improv.
0	56.33%	-
1	83.63%	+27.30%
2	87.50%	+3.87%
3	90.20%	+2.70%

Table 3: The relationship between the number of iterations and elemental completeness shows that as the number of image iterations increases, the completeness of elements in the images correspondingly rises, achieving a significant gain of 27.30% by the first round.

#### 4.4 Ablation Experiment

**Settings.** In order to verify the effect of the initial generation on correction, we perform an ablation experiment, removing additional information such

as translations and annotations, and directly utilize the original text of the poems for generation.

**Result.** The results of the experiment show in Tab. 4. The completeness of the initial generation remains largely unchanged after eliminating additional information, while after detection and correction the elemental completeness decreases by approximately 11%. This is because most elements are derived directly from the poem’s text, so the initial generation’s completeness is unaffected. However, images generated directly based on poems lack much of the additional semantics from the additional information, due to factors such as lack of stylistic richness, which reduces the completeness of the picture elements, and subsequent modifications can be much less effective.

Setup	Elemental Completeness	
	Initial Image	First Round
Poetry	54.61%	72.50%
Translation	56.33%	83.63%

Table 4: Ablation experiment result. After the elimination of the additional information, the completeness of the initial generation remains largely unchanged, while the completeness of the elements after the detection and correction decreases by approximately 11%.

## 5 Discussion

### 5.1 The Influence of the Number of Key Elements in Poetry

To evaluate the performance of Poetry2Image in processing Chinese classical poetry with different numbers of key elements, we design a series of experiments and use elemental completeness as the evaluation indicator.

The experimental results, as illustrated in Tab. 5, indicate that with fewer key elements, such as 3, the initial generation covers most elements, resulting in minimal improvement in overall elemental completeness. As the number of key elements increases, the initial generation’s missing rate escalates. However, Poetry2Image compensates for this by completing elements more rapidly than they are missed, resulting in a 15% to 20% improvement in elemental completeness. Specifically, for information-intensive poems containing up to six elements, the elemental completeness improvement rate reaches 23.73%. This demonstrates Poetry2Image’s efficacy in improving elemental completeness. However, when the number of elements exceeds seven, the image fails to achieve the desired elemental completeness, posing a challenge in balancing the aesthetics of the image with the improvement of elemental completeness.

Num of Elements	Improvement
3	+13.63%
4	+15.35%
5	+18.61%
<b>6</b>	<b>+23.73%</b>
7	+3.57%

Table 5: The impact of the number of key elements contained in poetry on the elemental completeness. Poetry2Image performs well when dealing with poetry with multiple key elements, ranging from 3 to 6.

### 5.2 The Influence of Poetry Language Types

To further assess the generalizability and applicability of Poetry2Image, we extended its application to multilingual poetry. We test Poetry2Image on 100 classical Japanese and English poems representing diverse linguistic and cultural backgrounds.

The results, as detailed in Appendix A, demonstrate that our method is effective not only with Chinese classical poetry but also with Japanese and English poetry. This confirms the wide applicability

of Poem2Image and provides insights into generating images of multilingual poetry.

## 6 Limitations

The limitations of Poetry2Image stem from the intrinsic characteristics of poetry, as illustrated in Fig. 6. The poems corresponding to the images can be found in Appendix B. When the genre of the poem is lyrical or didactic, the key elements in the sentence are scarce or abstract, so both the initial image and the corrected image fail to capture the key elements used for generation, leading to unsatisfactory correction results. In addition, when dealing with proper nouns such as historical personal names (e.g. ‘Zhou Yu’) in the poems, the elements cannot be recognized and understood by the OVD and diffusion models, resulting in suboptimal correction results.

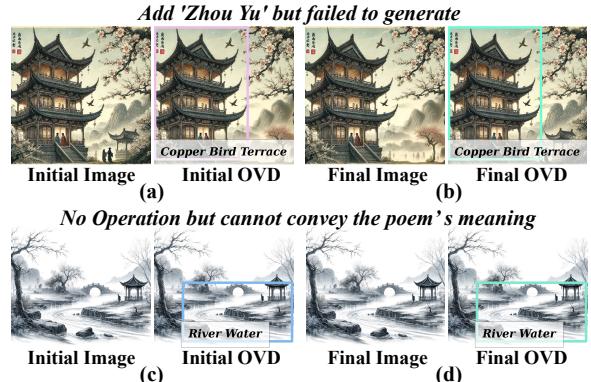


Figure 6: The diffusion model is unable to understand the key element ‘Zhou Yu’, who is a historical figure, so cannot generate it. In the second poem, all elements can be identified, but it fails to convey the sense of nostalgia for the dead hero.

## 7 Conclusion

We propose Poetry2Image, an iterative correction framework that integrates image generation, error correction and feedback. This framework enhances image generation quality for specialized texts like Chinese classical poetry and addresses core issues such as element loss and semantic confusion. Our method is adept at element-rich or multi-lingual poems and is compatible with other image generation models. Additionally, our approach provides a reference for similar non-fine-tuning methods to enhance LLM generation.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.
- Omri Avrahami, Dani Lischinski, and Ohad Fried. 2022. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218.
- Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. 2022. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*.
- Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. 2023. Multidiffusion: fusing diffusion paths for controlled image generation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML’23. JMLR.org.
- James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jian-feng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. 2023. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8.
- Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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.
- Liming Deng, Jie Wang, Hangming Liang, Hui Chen, Zhiqiang Xie, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2020. An iterative polishing framework based on quality aware masked language model for chinese poetry generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7643–7650.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.
- Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.
- Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*.
- Guian Fang, Zutao Jiang, Jianhua Han, Guansong Lu, Hang Xu, Shengcai Liao, and Xiaodan Liang. 2023. Realigndiff: Boosting text-to-image diffusion model with coarse-to-fine semantic re-alignment. *arXiv preprint arXiv:2305.19599*.
- Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2022. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv preprint arXiv:2212.05032*.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.
- Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- 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*.
- Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Hui-wen Chang, Tali Dekel, Inbar Mossneri, and Michal Irani. 2023. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017.
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.
- Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. 2023. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*.
- Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.

- Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. 2024. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36.
- William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*.
- Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.
- Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *Preprint, arXiv:1904.09223*.
- Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. 2023. 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.
- Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. 2024. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36.
- Zhengyuan Yang, Jianfeng Wang, Zhe Gan, Linjie Li, Kevin Lin, Chenfei Wu, Nan Duan, Zicheng Liu, Ce Liu, Michael Zeng, et al. 2023. Reco: Region-controlled text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14246–14255.
- Xiaoyuan Yi, Ruoyu Li, Cheng Yang, Wenhao Li, and Maosong Sun. 2020. Mixpoet: Diverse poetry generation via learning controllable mixed latent space. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9450–9457.
- Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2023. GLM-130B: an open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.
- Wendi Zheng, Jiayan Teng, Zhuoyi Yang, Weihan Wang, Jidong Chen, Xiaotao Gu, Yuxiao Dong, Ming Ding, and Jie Tang. 2024. Cogview3: Finer and faster text-to-image generation via relay diffusion. *Preprint, arXiv:2403.05121*.
- Guo Zhipeng, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. 2019. Jiuge: A human-machine collaborative chinese classical poetry generation system. In *Proceedings of the 57th annual meeting of the association for computational linguistics: system demonstrations*, pages 25–30.

## Appendix

### A Results of Poetry Image Correction in Multiple Languages

Poetry examples in different languages and test results of Poetry2Image are shown below.

1. Japanese Haiku: The moon in the water; Broken and broken again, Still it is there.
2. American English Poetry: On the beach at night alone, As the old mother sways her to and fro singing her husky song, As I watch the bright stars shining, I think a thought of the clef of the universes and of the future.
3. British English Poetry: O wild West Wind, thou breath of Autumn’s being Thou, from whose unseen presence the leaves dead Are driven, like ghosts from an enchanter fleeing, Yellow, and black, and pale, and hectic red.



Figure 7: Poetry image generation in different languages and styles. The left is generated directly from literal meaning, and the right shows corrections of our method.

Initially, for Japanese poetry, we chose the renowned haiku of Matsuo Basho for analysis. Our method accurately identified the metaphor of a ‘broken moon in the water’ and appropriately adjusted the image from a moon in the sky to reflect this. Subsequently, for English poetry, we tested poems by Whitman and Shelley. The results indicate that our method effectively interprets and corrects metaphors such as ‘old mother’ and ‘ghosts’.

### B Poetry Text for Generating Images

#### Poetry Text for Generating Fig. 5.

Poetry a: Singing loudly in front of the wine, life is short and the days pass by quickly.

Poetry b: The capital is filled with nobles in fine cars and beautiful clothes, but you are extremely talented but your face is haggard.

Poetry c: The sparse shadows of plum blossoms are reflected obliquely in the clear water, and the faint fragrance of plum blossoms is drifting in the hazy moonlight.

Poetry d: The Xianglu Peak is covered with purple haze under the sunlight, and from a distance you can see a waterfall hanging in front of the mountain like white silk.

Poetry e: I am facing a cup of sad wine, thousands of miles away from home. I have a lot of thoughts, thinking about the unrest on the border, the unfinished work, and I don’t know when I can return to my hometown.

Poetry f: I stopped the carriage just because I loved the maple forest in the evening. The frost-stained maple leaves are more beautiful than the bright flowers in February.

#### Poetry Text for Generating Fig. 6.

Poetry a: Without the help of the east wind, Jiangnan would have been a ruin; the beautiful Erqiao would have been locked up in the Tongque Tower forever.

Poetry b: The people back then are no longer around, but the Yishui River is still as cold today.

### C System Prompt Setup of Extractor and Suggester in Our Method

We use a recognition method based on open vocabulary detector to detect key elements of poems and an automatic iterative correction framework to generate images through secondary diffusion. The system prompt setup of our extractor and suggester are shown below.

---

```
1 # Your role: Excellent Chinese poetry parser
2
3 ## Task objective: Analyze the given poem and its related descriptions, and extract
   the key image elements from it.
4
5 ## Processing steps
6 1. Read the poem provided by the user and its translation and appreciation.
7 2. Identify all the key image elements mentioned in the poem or translation and
   record them. The key image elements will be used to draw an ink painting of
   this poem later.
8 3. The key image elements are listed in the form of "noun" or "adjective + noun".
9 4. Explain your reasoning and organize your results in the format of the example.
10 5. The elements must be complete, including all the key elements mentioned in the
    poem.
11 6. Abstract descriptions such as atmosphere and emotion must not appear in the key
    elements, such as "desolate atmosphere" and "sad mood".
12 7. Please ensure that the key image elements have no brackets, quotation marks or
    other special characters, and are specific nouns or adjective + noun
    combinations.
13
14 ## Example
15 - Example 1
16 Original sentence: Yellow sand and golden armor worn through a hundred battles,
   never return until Loulan is conquered.
17 Translation: The soldiers guarding the border have experienced a hundred battles,
   their armor worn through, their ambitions undying, they will not return home
   until they defeat the invading enemy.
18 Appreciation: The first sentence shows the long time of guarding the border, the
   frequent battles, the hardship of the battles, the strength of the enemy, and
   the desolation of the border. The second sentence expresses the soldiers' lofty
   ambitions to serve their country to the death and their sincere patriotic
   enthusiasm.
19 Reasoning: The yellow sand and golden armor mentioned in the description reflect
   the hardships of border defense and the tenacity of the soldiers.
20 Image elements:
21 1. Yellow desert
22 2. Soldiers wearing golden armor
23 3. Desolate border battlefield
24
25 - Example 2
26 Original sentence: Brocade hats and mink furs, thousands of cavalry roll across the
   flat hills.
27 Translation: Wearing gorgeous and bright hats, wearing mink furs, leading a mighty
   large army, like a gust of wind, sweeping across the flat hills.
28 Appreciation: The magnificent scene of hunting shows the hunter's majestic and
   heroic spirit, shapes the poet's high-spirited image, and shows the poet's
   passionate attitude towards life.
29 Reasoning: "Brocade hat" and "sable fur" describe the cavalry's clothing, and
   "thousands of cavalry rolling on the flat hill" shows the magnificent scene of
   a large army marching and hunting on a flat hill.
30 Picture elements:
31 1. Gorgeous hat
32 2. Cavalry wearing sable fur
33 3. Broad flat hill
34 4. Huge cavalry team
35
36 Your current task: Follow the above steps carefully and accurately identify the
   screen elements based on the given poem. Be sure to follow the above output
   format.
```

---

Table 6: System Prompt Setup of Extractor in Our Method.

---

```

1 # Your Role: An Expert in Bounding Box Adjustment for Images
2
3 ## Objective
4 Adjust the bounding box list in a square image according to the User Prompt
   information provided, output the modified bounding box list , and ensure that
   Updated Objects completely and correctly cover all elements in Elements that
   must be included.
5
6 ## Bounding Box Specifications and Manipulations
7 1. Image Coordinates: define a square image with corners at [0, 0] and [1, 1].
8 2. Box Format: specify the box using [top-left x, top-left y, width, height].
9 3. Operations: four operations: Add, Delete, Move, and Replace.
10 4. Object name: Attach "#n" to the object name to indicate the nth occurrence of
    the same object name.
11 5. Composition of the bounding box: it consists of the object name and the box.
12
13 ## Necessary Information
14 1. Image Description: describes all the elements that must be included in the image
   and shows the semantic relationship between all the elements in the image.
15 2. Current Objects: a list of bounding boxes in the square image, listing the
   objects currently present in the image and their corresponding bounding boxes
16 3. Reasoning: Change from Current Objects to Updated Objects, output your reasoning
   process.
17 4. Updated Objects: outputs a list of the bounding boxes you expect to see in the
   square image, listing the objects and corresponding bounding boxes that should
   be present in the desired image.
18
19 ## Key Guidelines
20 1. Alignment: Updated Objects should completely and correctly cover all elements
   that must be included; The expected images corresponding to Updated Objects
   should be highly consistent with the Image Description.
21 2. Boundary Adherence: Keep all bounding box coordinates within the [0, 1] range.
22 3. Minimize Modifications: Only modify the bounding boxes of Current Objects that
   do not completely or correctly cover the image elements.
23 4. Minimize Overlap: Minimize intersections between bounding boxes and adjust as
   needed to reduce overlap without loss of bounding box coverage.
24
25 ## Process Steps
26 1. Interpret the prompts: Carefully read and understand the information provided in
   the User Prompt: Image Description and Elements that must be included.
27 2. Implement changes: View Current Objects and make the necessary adjustments.
28 3. Explain Adjustments: Clearly explain the reason for each border modification and
   ensure that each adjustment meets the key criteria.
29 4. Output results: Provide detailed reasoning first, and then an updated list of
   bounding boxes - Updated Objects - in a structured format that demonstrates the
   changes made.
30
31 ## Examples
32 User Prompt:
33 - Image Description: "The Incense Burner Peak in the sunshine of the purple haze,
   see waterfalls like white silk hanging in front of the mountain from a
   distance."
34 - Elements that must be included: ['sunshine','peak','purple haze','waterfall']
35 Current Objects: [('peak #1', 0.021, 0.983, 0.949, 0.389), ('incense burner
   #1',[0.341, 0.269, 0.188, 0.189]), ('sunshine #1',[0.407, 0.80, 0.15, 0.114]),
   ('waterfall #1',[0.390, 0.724, 0.191, 0.354]), ('purple smoke #1',[0.405,
   0.562, 0.570, 0.291])]
36 Reasoning: The Image Description doesn't depict the incense burner. The 'incense
   burner #1' appears due to a misunderstanding of the Incense Burner Peak, so
   'incense burner #1' should be removed to correct the misunderstanding.
37 Updated Objects: [('peak #1',[0.021, 0.983, 0.949, 0.389]), 'sunshine #1',[0.407,
   0.80, 0.15, 0.114]), ('waterfall #1',[0.390, 0.724, 0.191, 0.354]), ('purple
   haze #1',[0.405, 0.562, 0.570, 0.291])]
38
39 Your Current Task: Follow the provided guidelines and steps to adjust bounding
   boxes while ensuring the completeness and accuracy of key elements. Please
   Ensure adherence to the output format specified above.

```

---

Table 7: System Prompt Setup of Suggester in Our Method