Compositional Text-to-Image Generation Via Region-aware Bimodal Direct Preference Optimization

Supplementary Material

In this supplementary, we provide additional details and results as follows:

- In Section 1, we provide the full results of the ablation study on SDXL based models.
- In Section 2, we provide more details about our data construction process, including the composition of collected captions from various sources and the prompts used in various stages.
- In Section 3, we provide more visualization results of our BICOMP dataset and our BIDPO method.

1. Ablation Study Details.

The full results of the ablation study on SDXL based models are shown in Table 1, Table 2 and Table 3.

2. Data Construction Details

Caption Collection. Table 4 shows the number of captions collected from each source.

LLM Prompt for Dimension Parsing. Our dimension parsing prompt is shown in Listing 1. We use DeepSeek-V3 [2] as our LLM to parse the captions.

Listing 1. prompt for dimension parsing

```
# Task Description
Given a sentence, analyze its content and
    ↪ determine which of the following
    ↪ dimensions it primarily describes:
- color: describes colors (e.g., "red", "
    ↪ yellow", "dark blue")
- shape: describes geometric forms or
    ↪ outlines (e.g., "round", "triangular
    ↪ ", "curved")
- texture: describes textures (e.g., "
    ↪ smooth", "rough") or materials (e.g.,
    ↪  "a plastic chair", "a glass window")
- spatial: describes spatial relationships
    ↪ or positions (e.g., "on the table", "
    ↪ next to", "inside", "beneath")
- non-spatial: describes actions/events
    ↪ without spatial focus (e.g., "chasing
    ↪ ", "biting")
- numeracy: describes quantities or numbers
    ↪  (e.g., "three apples", "four", "two
    ↪ ")
- others: when none of the above categories
    ↪  apply

# Priority Rules
If multiple dimensions are present, select
    ↪ according to this priority:
1. spatial and non-spatial have highest
    ↪ priority (equal)
2. numeracy comes next
3. color, shape and texture have equal
    ↪ priority (lower than above)
4. others is always lowest priority

# Output Format
Provide your analysis in exact JSON format
    ↪ as shown below. Only include the JSON
    ↪  object in your response.

{{
    "dimension": "selected_dimension"
}}

# Examples
Input: "The cube is on the shelf"
Output: {{ "dimension": "spatial" }}

Input: "Five rough textured stones"
Output: {{ "dimension": "numeracy" }}

Input: "The soft yellow pillow"
Output: {{ "dimension": "color" }}

# Input
The input sentence is: {positive_caption}

# Output
For this sentence, the dimension is:
```

LLM Prompt for Object List Parsing. We use the prompt shown in Listing 2 to parse the object list from captions. We use DeepSeek-R1 [3] as our LLM to parse the captions.

Listing 2. prompt for object list parsing

```
You are an expert in parsing textual
    ↪ sentences. Given a text that
    ↪ describing an image, you task is to
    ↪ identify and extract the main
    ↪ entities in the image.

# Requirements
- You should only put the main entities
    ↪ that are visually visible in the
    ↪ image.
- Make sure the entities you identify are
    ↪ concrete objects, not abstract
```

Table 1. **Ablation Study on T2I-CompBench [8].**

Model	Attribute Binding			Object Relationship		Numeracy
	Color	Shape	Texture	Spatial	Non-Spatial	
SDXL	58.90	46.90	53.13	21.23	31.20	50.08
SDXL-SFT	58.67	46.65	52.21	21.13	31.28	50.08
SDXL-ImageDPO	67.39	53.12	59.42	23.4	30.82	39.34
SDXL-TextDPO	23.32	16.22	14.62	0.26	20.3	6.13
SDXL-BIDPO w/o region-level guidance	77.04	57.43	68.89	23.19	32.19	59.83
SDXL-BIDPO w/ region-level guidance	79.35	60.47	71.36	23.41	32.29	59.33

Table 2. **Ablation Study on GenEval [5].**

Model	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	**Overall**
SDXL	0.95	0.68	0.42	0.85	0.11	0.19	0.53
SDXL-SFT	0.95	0.68	0.37	0.85	0.09	0.20	0.52
SDXL-ImageDPO	0.99	0.78	0.15	0.89	0.14	0.23	0.53
SDXL-TextDPO	0.13	0.01	0.01	0.11	0.01	0.02	0.04
SDXL-BIDPO w/o region-level guidance	1.00	0.83	0.52	0.90	0.16	0.23	0.61
SDXL-BIDPO w/ region-level guidance	1.00	0.87	0.56	0.90	0.17	0.23	0.62

```
    ↪ concepts; objects like 'living room'
    ↪ or 'wind' should not be identified.
- Make sure these entity objects can be
    ↪ detected by an object detector.
- Only output the entities themselves,
    ↪ without their adjectives or
    ↪ descriptions; for example, output '
    ↪ dog' instead of 'white dog'.

# Output format
Orgainize the identified main objects in
    ↪ the scene into a json dict like this:
{{
    "object_list": ["object 1", "object 2",
        ↪ ...]
}}
# Input
For the sentence: {caption}, please
    ↪ identify the main visible objects.
```

Image Describing Details. Before we prompt the VLM to do the describing tasks, we restrict the image to follow the following rules: 1) with dimension "color", "shape", or "texture", the image should contain one or two objects 2) with dimension "spatial" or "non-spatial", the image should contain exactly two objects. 3) no repeated classes in the image; each object must belong to a unique class. We use specific prompts for different dimensions. Examples of "color" and "spatial" dimension are shown in Listing 3 and Listing 4. The prompts for "shape", "texture", and "non-spatial" dimensions are similar to the "color" and "spatial" ones, respectively. We use Qwen2.5-VL-72B-Instruct [1] as our VLM to describe the images.

Listing 3. prompt for VLM describing, with dimension "color"

```
# Task explanation
Given an image with clearly marked regions-
    ↪ of-interest (each region is indicated
    ↪  by a numerical ID and contour lines)
    ↪ , please:
1. Identify all visible regions-of-interest
    ↪  by their numerical IDs
2. For each region, determine the
    ↪ predominant color of the object
    ↪ contained within it
3. Describe colors using standard web color
    ↪  names (e.g., "red", "forestgreen", "
    ↪ royalblue")
4. Handle uncertainty cases appropriately

# Output Requirements:
- Strict JSON format
- For unclear cases: use "unknown" as color
    ↪  value
- Sort results by region ID in ascending
    ↪ order

Output Example:
{
  "color_predictions": [
    {
      "region_id": 1,
      "color": "red"
```

Table 3. **Ablation Study on DPG-Bench [7].**

Model	Global	Entity	Attribute	Relation	Other	Overall
SDXL	82.44	81.87	81.17	80.54	79.77	73.38
SDXL-SFT	82.68	81.94	79.52	81.02	81.03	73.23
SDXL-ImageDPO	79.13	83.19	82.76	83.39	82.49	75.70
SDXL-TextDPO	40.07	40.44	43.14	43.92	44.82	23.98
SDXL-BIDPO w/o region-level guidance	85.46	84.22	84.45	85.28	84.18	77.53
SDXL-BIDPO w/ region-level guidance	83.92	85.28	85.13	85.03	84.55	78.84

Dataset	Number of Captions
CONPAIR [6]	13,432
ReasonGen-R1 [12]	23,470
T2I-R1 [9]	7,223
T2I-CompBench Training Set [8]	5,600
Total	**49,725**

Table 4. Number of composition-related captions collected from various sources.

```
    },
    {
      "region_id": 2,
      "color": "unknown"
    }
  ]
}

# Special Instructions:
- Ignore background colors outside marked
    ↪ regions
- Focus on the dominant colors
- IDs and contour lines are only for
    ↪ reference. DO NOT use them for color
    ↪ analysis
```

Listing 4. prompt for VLM describing, with dimension "spatial"

```
# Task explanation
Given an image with two clearly marked
    ↪ regions-of-interest (each region is
    ↪ indicated by a numerical ID and
    ↪ contour lines), please:
1. Identify the two regions-of-interest by
    ↪ their numerical IDs
2. Determine the precise spatial
    ↪ relationship between the two objects
    ↪ contained within the two regions-of-
    ↪ interest, where:
   - The reference object should be the
       ↪ visually more salient/dominant
       ↪ object (typically larger, more
       ↪ central, or more prominent in the
       ↪ scene)
   - The target object's position is
       ↪ described relative to the
       ↪ reference object
   - Use specific spatial descriptors (e.g
       ↪ ., "on the right of", "above", "
       ↪ behind")
3. Handle uncertainty cases appropriately
    ↪ when spatial relationships cannot be
    ↪ clearly determined

# Output Requirements:
- Strict JSON format
- For unknown spatial relationship: use "
    ↪ unknown"
- Always describe the target object's
    ↪ position relative to the reference
    ↪ object

Output Example:
{
  "reference_object_id": 1,
  "target_object_id": 2,
  "spatial_prediction": "in front of",
  "notes": "object 2 is in front of object
      ↪ 1"
}

For unknown cases:
{
  "spatial_prediction": "unknown"
}

# Special Instructions:
- Do not use unclear descriptions like "
    ↪ next to", "beside", "near", "close to
    ↪ ", etc
- IDs and contour lines are only for
    ↪ reference. DO NOT use them for
    ↪ spatial relationship analysis
- If neither object is clearly more salient
    ↪ , default to using the lower ID as
    ↪ reference
```

VLM prompts for Region Information Differing. We

use specific prompt for each dimension to generate distinct region information. Examples of "color" and "spatial" dimension are shown in Listing 5 and Listing 6. The prompts for "shape", "texture", and "non-spatial" dimensions are similar to the "color" and "spatial" ones, respectively. We use Qwen2.5-VL-72B-Instruct [1] as our VLM to generate the distinct region information.

Listing 5. prompt for VLM differentiation, with dimension "color"

```
# Task Explanation
Here is an image with outlined regions (
    ↪ each region is indicated by a
    ↪ numerical ID and contour lines).
And here are the list of regions with their
    ↪  dominant colors for reference (
    ↪ format: {{"region_id": N, "color": "
    ↪ color_name"}}):
{obj}

Now please propose a visually distinct
    ↪ color for each region that
    ↪ significantly differs from ALL
    ↪ provided dominant colors in the image
    ↪ .

# Requirements:
1. For each region, suggest ONE color that
    ↪ contrasts distinctly with ALL
    ↪ dominant colors in the image
2. Consider human perceptual difference (
    ↪ avoid suggesting similar hues/
    ↪ brightness)
3. Prefer standard color names (e.g., "red
    ↪ ", "green")
4. Never suggest the same as any input
    ↪ dominant color
5. When multiple options exist, choose the
    ↪ highest-contrast alternative

# Output Format (strict JSON):
{{
  "output": [
    {{"region_id": N, "different_color": "
        ↪ color_name"}},
    ...(other regions)
  ]
}}

# Examples:
Input Colors: [{{"region_id": 1, "
    ↪ dominant_color": "red"}}, {{"
    ↪ region_id": 2, "dominant_color": "
    ↪ blue"}}]
Output: {{
  "output": [
    {{"region_id": 1, "different_color": "
        ↪ yellow"}},
    {{"region_id": 2, "different_color": "
        ↪ black"}}
  ]
}}

Input Colors: [{{"region_id": 1, "
    ↪ dominant_color": "green"}}, {{"
    ↪ region_id": 2, "dominant_color": "
    ↪ yellow"}}]
Output: {{
  "output": [
    {{"region_id": 1, "different_color": "
        ↪ magenta"}},
    {{"region_id": 2, "different_color": "
        ↪ navy_blue"}}
  ]
}}
```

Listing 6. prompt for VLM differentiation, with dimension "spatial"

```
# Task Explanation
Here is an image with TWO outlined regions
    ↪ (each region is indicated by a
    ↪ numerical ID and contour lines).
And here is the spatial relationship
    ↪ between the two objects contained in
    ↪ the two regions:
object {object_id_1} is {spatial_relation}
    ↪ object {object_id_2}

Now please propose a geometrically distinct
    ↪  spatial relationship that
    ↪ significantly differs from the given
    ↪ relationship.

# Requirements:
1. Suggest ONE primary spatial relationship
    ↪  that contrasts maximally with the
    ↪ input relationship
2. Consider these transformation axes for
    ↪ differentiation:
   a) Vertical inversion (above/below
       ↪ swap)
   b) Horizontal inversion (left/right
       ↪ swap)
   c) Dimensional shift (adjacent
       ↪ separated)
   d) Topological change (inside
       ↪ outside)
3. Use standard spatial terms from this
    ↪ vocabulary:
   [above, below, on the left of, on the
       ↪ right of, in front of, behind,
       ↪ ...]
4. The new relationship must be:
   a) Physically plausible for the objects'
       ↪  shapes/sizes
```

```
   b) Perceptually distinct from original
   c) Expressed as "object X [RELATION]
      ↪ object Y"
5. Include brief reasoning in "notes"

# Output Format (strict JSON):
{{
  "output": {{
    "different_spatial_relation": "
        ↪ relation_term",
    "notes": "object [object_id_X] [
        ↪ RELATION] object [object_id_Y]"
  }}
}}

# Examples:
Input: "object A is above object B"
Output: {{
  "output": {{
    "different_spatial_relation": "below",
    "notes": "object A is below object B (
        ↪ vertical inversion)"
  }}
}}

Input: "object X is inside object Y"
Output: {{
  "output": {{
    "different_spatial_relation": "outside
        ↪ ",
    "notes": "object X is outside object Y
        ↪ (topological complement)"
  }}
}}

Input: "object 1 is adjacent to object 2"
Output: {{
  "output": {{
    "different_spatial_relation": "
        ↪ separated",
    "notes": "object 1 is separated from
        ↪ object 2 (proximity reversal)"
  }}
}}
```

VLM prompt for VQA-based filtering. We use the prompt shown in Listing 7 to filter out low-quality samples. We use Qwen2.5-VL-72B-Instruct [1] as our VLM to perform the filtering.

Listing 7. prompt for VLM VQA-based filtering

```
You are given an image with several regions
    ↪  of interest (ROIs). Each ROI is
    ↪ highlighted in the image with contour
    ↪  lines and labeled with a unique
    ↪ numerical ID.

You are also given a list of questions.
    ↪ Each question refers to one or more
    ↪ ROIs. Here are the questions:
{questions}

Your task:

1. For each question, evaluate whether the
    ↪ statement is correct with respect to
    ↪ the corresponding region(s).
2. Provide a confidence score between 0 and
    ↪  1 (`answer`) indicating how strongly
    ↪  you agree with the statement (1 =
    ↪ completely true, 0 = completely false
    ↪ ).
3. Provide a short explanation (`reason`)
    ↪ describing why you assigned this
    ↪ score.

The output format must strictly follow this
    ↪  JSON structure:

```json
[
  {{
    "question_id": <int>,
    "answer": <float between 0 and 1>,
    "reason": "<string explanation>"
  }},
  ...
]
```

Example:
Input image: contains region 1 (a yellow
    ↪ lemon) and region 2 (a red apple).
Questions:

```json
[
  {{"question_id": 0, "question": "Does
      ↪ region 1 mark a yellow lemon?"}},
  {{"question_id": 1, "question": "Does
      ↪ region 2 mark a blue apple?"}}
]
```

Expected output:

```json
[
  {{"question_id": 0, "answer": 0.99, "
      ↪ reason": "Region 1 does mark a
      ↪ yellow lemon."}},
  {{"question_id": 1, "answer": 0.01, "
      ↪ reason": "The apple in region 2 is
      ↪ actually red."}}
]
```
```

3. More Visualization Results.

We provide more visualization results of our BICOMP dataset and our BIDPO method in Figure 1, Figure 2, Figure 3, Figure 4 and Figure 5.

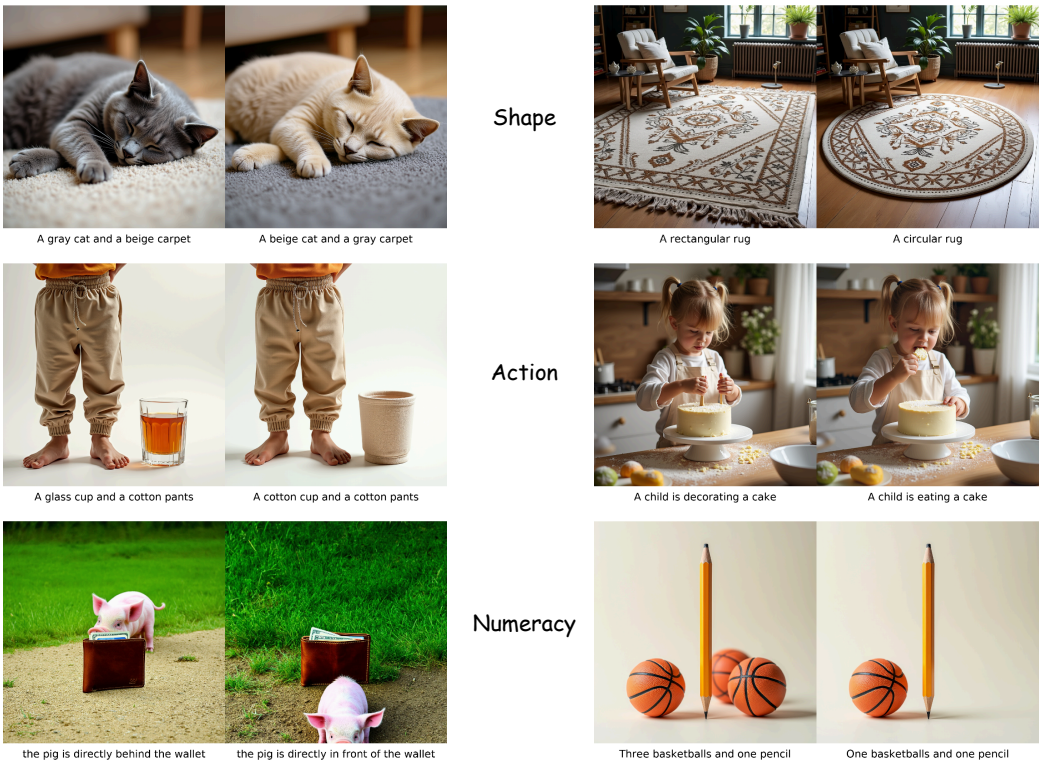

Figure 1. **Samples of each dimension in our BICOMP dataset.** For each group, the left image is generated from the original caption, and the right image is generated from the edited caption.

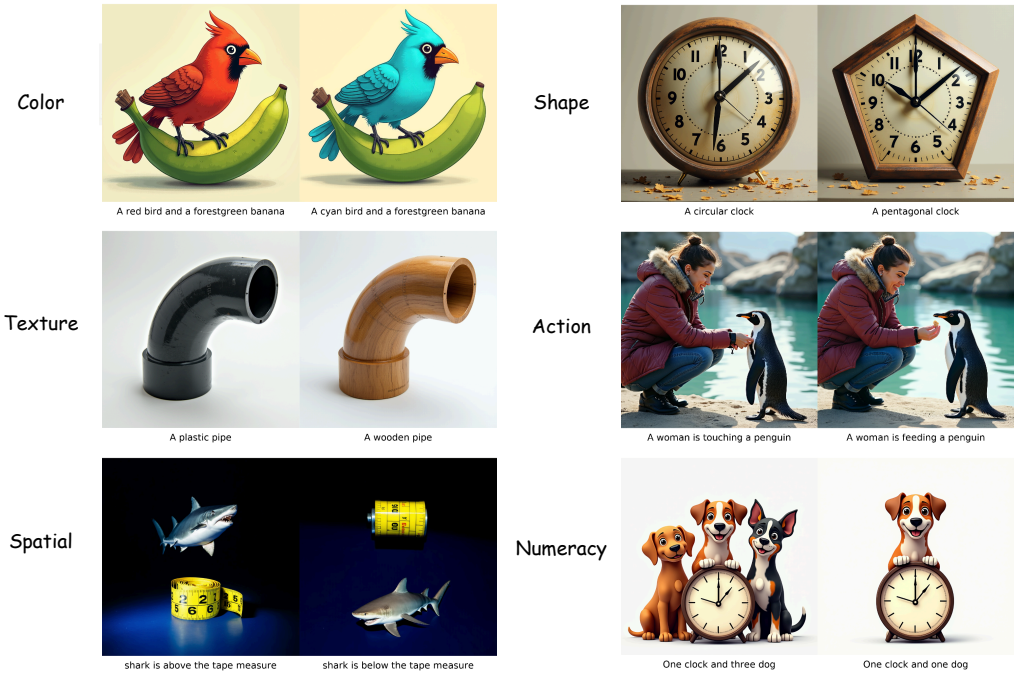

Figure 2. **Samples of each dimension in our BICOMP dataset**. For each group, the left image is generated from the original caption, and the right image is generated from the edited caption.

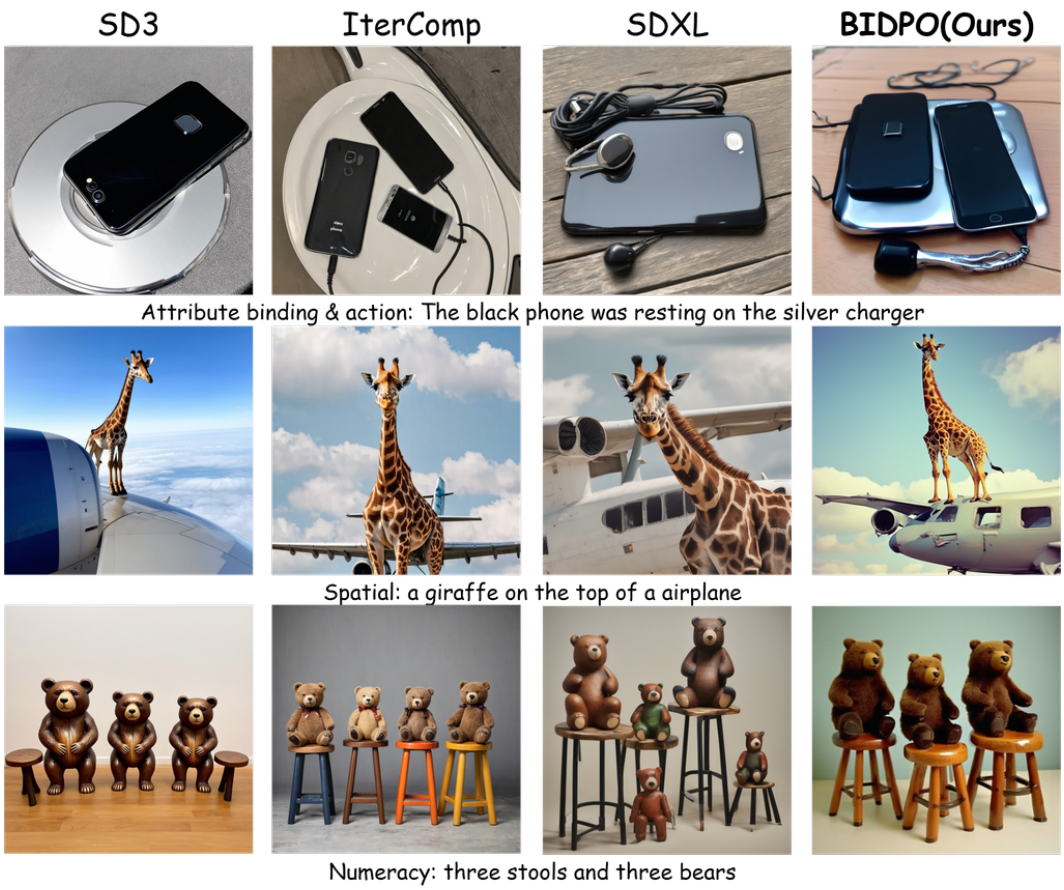

Figure 3. **Visualization of text-to-image generation results**. From left to right are Stable Diffusion 3 [4], IterComp [11], Stable Diffusion XL [10], and Stable Diffusion XL finetuned with our proposed BIDPO.

Figure 4. **Visualization of text-to-image generation results of Stable Diffusion 1.5**. We compare Stable Diffusion 1.5 finetuned with our proposed BIDPO with the original Stable Diffusion 1.5.

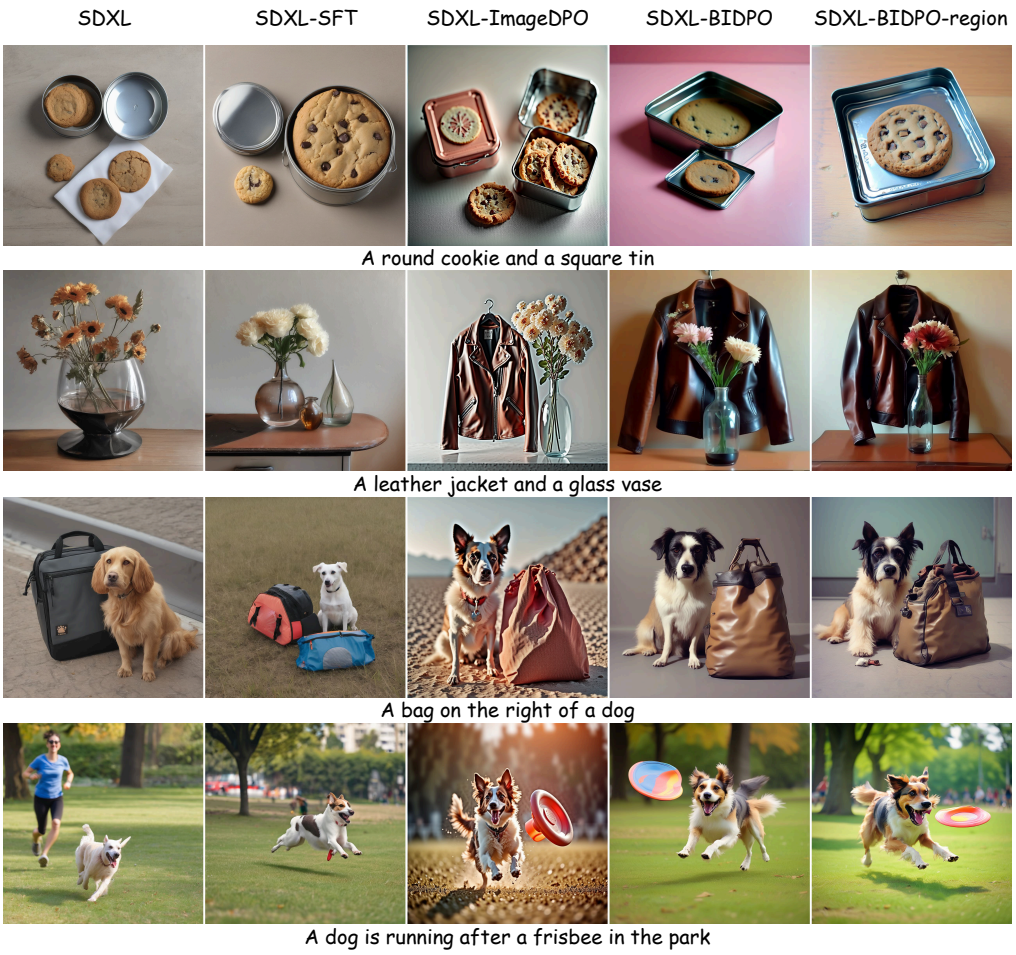

Figure 5. **Visualization of text-to-image generation results of Stable Diffusion XL**.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923, 2025. 2, 4, 5

[2] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bing-Li Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Jun-Mei Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, Ruiqi Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shao-Ping Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, Wangding Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xuan Yu, Wentao Zhang, X. Q. Li, Xiangyu Jin, Xianzu Wang, Xiaoling Bi, Xiaodong Liu, Xiaohan Wang, Xi-Cheng Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yao Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yi-Bing Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxiang Ma, Yuting Yan, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-An Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report. *ArXiv*, abs/2412.19437, 2024. 1

[3] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bing-Li Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Jiong Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, M. Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, Ruiqi Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shao-Kang Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xia Yu, Wentao Zhang, Wangding Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyu Jin, Xi-Cheng Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yu-Jing Zou, Yujia He, Yunfan Xiong, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yao Li, Yi Zheng, Yuchen Zhu, Yunxiang Ma, Ying Tang, Yukun Zha, Yuting Yan, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-An Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948, 2025. 1

[4] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 8

[5] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, 2023. 2

[6] Xu Han, Linghao Jin, Xiaofeng Liu, and Paul Pu Liang. Contrafusion: Contrastively improving compositional understanding in diffusion models via fine-grained negative images. In *ICLR*, 2025. 3

[7] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment, 2024. 3

[8] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-

world compositional text-to-image generation. In *NeurIPS*, 2023. 2, 3

[9] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *ArXiv*, abs/2505.00703, 2025. 3

[10] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *ArXiv*, abs/2307.01952, 2023. 8

[11] Xinchen Zhang, Ling Yang, Guohao Li, Yaqi Cai, Jiake Xie, Yong Tang, Yujiu Yang, Mengdi Wang, and Bin Cui. Itercomp: Iterative composition-aware feedback learning from model gallery for text-to-image generation. In *ICLR*, 2025. 8

[12] Yu Zhang, Yunqi Li, Yifan Yang, Rui Wang, Yuqing Yang, Dai Qi, Jianmin Bao, Dongdong Chen, Chong Luo, and Lili Qiu. Reasongen-r1: Cot for autoregressive image generation models through sft and rl. *ArXiv*, abs/2505.24875, 2025. 3