Cross-Attention Masking for Generative Spatial Control

Kevin Blin
Department of Computer Science
ETH Zürich
kblin@student.ethz.ch

Vandit Sharma
Department of Computer Science
ETH Zürich
sharmav@student.ethz.ch

Steven Wang
Department of Computer Science
ETH Zürich
stewang@student.ethz.ch

Han Xi
Department of Computer Science
ETH Zürich
hxi@student.ethz.ch

Abstract

Stable Diffusion lacks an explicit mechanism for spatial control, often leading to false placement of objects and window artifacts. In this work, we propose a method utilizing cross-attention control in the latent space of stable diffusion models to generate more spatially coherent images. Our top-performing approach attains aa VISOR score of 13.4% on images necessitating binary object placement, doubling the performance of the baseline Stable Diffusion model. We also explore image prompts describing multiple objects and multiple spatial constraints by using GPT-4 prompting to generate suitable attention masks.

1 Introduction

Fueled by the recent interest in *Generative AI*, diffusion models like Stable Diffusion are now commonly used for tasks such as text-to-image generation and inpainting. These models, however, lack an explicit spatial control mechanism, assuming that spatial information is preserved in the prompt embedding generated by text encoder models like CLIP. Yet, recent studies show that these models often fail in accurate spatial positioning of generated objects [2], leading to issues like false positioning, partition windows, and object omission. This work explores several approaches to address these limitations.

The main contributions of our work are as follows: 1) We propose a novel method to achieve better generative spatial control using stable diffusion without the use of external grounding information from the user. 2) We develop an end-to-end pipeline that can generate coherent images with multiple objects mapped correctly to their corresponding location in the image given a single prompt. 3) We evaluate our approaches both quantitatively and qualitatively, and provide interesting insights for future work.

2 Related Work

Text-to-image models were first introduced in 2015 with alignDRAW [9], which first implemented the concept of text sequence conditioning. Reed, Akata, Yan et al. applied generative adversarial networks [3] to this task in 2016 [15]. The field saw further developments with OpenAI's CLIP [12], which used jointly learned text-image embeddings through contrastive learning, and DALL-E [13], a generative model that employed Variational AutoEncoder and transformer architectures [19] to generate images using CLIP embeddings.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

In parallel, diffusion-based generative models were proposed by Sohl-Dickstein et al [17], Yang and Song [18], and Ho et al. [6]. The intersection of these two areas led to the creation of Diffusion Models [11]. Further advancements were made with latent diffusion models [16] which are used in Stable Diffusion and DALL-E 2 [14].

Spatial control in these models has been addressed by several methods. ControlNet [22] introduced a method to train a wrapper network around Stable Diffusion, incorporating additional grounding information such as pose data and bounding boxes. GLIGEN [7] extended this concept by training a self-attention layer that fused image pixels with grounding information. The Prompt-to-Prompt approach [4] used cross-attention layers to control the relationship between the spatial layout and each token in the prompt. These methods have provided some degree of spatial control, but they require additional grounding information.

3 Methodology

Our core idea involves using cross-attention control in the latent space to condition different parts of the generated image with different prompts. In this process, we aim to extract the image regions to be conditioned on each object directly from the prompt. Our approach is simpler than existing methods of spatial control such as GLIGEN [7] or ControlNet [21] which rely on explicit grounding information from the user.

3.1 Cross-Attention Control Experiments

Here, we elaborate on the two main experiments that we perform. Complete details of all other experiments have been provided in Appendix A.

3.1.1 Baseline Experiment

To test whether conditioning different parts of the image with different prompts can lead to spatial control, we first conduct a simple baseline experiment. We consider the scenario of placing two objects to the left and right half of the image, respectively. To do this, we build upon the implementation of Hertz et al. [4]. We overwrite the cross-attention module of the UNet in Stable Diffusion with our hard-coded implementation in such a way that it reads two prompts and conditions only one-half of the image on each prompt. Figure 1 explains the idea behind our baseline experiment.

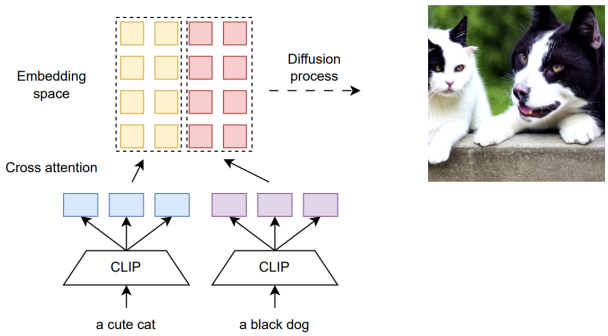

Figure 1: Our baseline experiment.

3.1.2 Divider+Negative Prompt Experiment

In this experiment, we introduce a divider between the two cross-attention masks, with the intention of achieving a better separation between the two objects. The divider can be imagined as a strip running across the middle of the image. By default, we use a divider width of 0.2 times the image size. For conditioning the divider region, we use a negative prompt, which is defined by the concatenation of the left and right prompts. Examples for each experiment are reported in Appendix A

3.2 Evaluation

To evaluate the image quality and object spatial control achieved by our different methods, we use the Object Accuracy(OA) and VISOR metrics proposed by Gokhale et al. [2]. OA indicates the extent to which all objects described in prompts appear in the generated images. OA can work with prompts describing multiple objects. On the other hand, VISOR can only handle prompts describing two objects. The goal of VISOR to benchmark if binary spatial constraints are met by text-to-image models. Given a text prompt with objects A and B related by relationship R, the two metrics can formally be defined as follows:

$$OA = \begin{cases} 1 & \text{if A and B in image} \\ 0 & \text{otherwise} \end{cases} \qquad VISOR = \begin{cases} 1 & \text{if A and B in image, R satisfied} \\ 0 & \text{otherwise} \end{cases}$$

The pipeline for computing overall OA and VISOR scores for a given method looks as follows: 1) We first generate 1000 prompts containing a random combination of two objects from the MS COCO [8] dataset related by a left/right or above/below relationship in the image. 2) We generate 4 images for each prompt to arrive at a total of 1000 images per method. 3) We compute the OA and VISOR scores for each image using the VISOR pipeline, which first employs the OWL-ViT [10] object detector to detect the presence of prompted objects, and then uses the metric logic to compute OA and VISOR scores. 4) Finally, we average the OA and VISOR scores across all randomly generated images to arrive at the final OA and VISOR score for a method.

3.3 Auto-grid

In the final phase of our research, we expanded our spatial control method to accommodate prompts with an arbitrary number of spatial constraints, as illustrated in Figure 8. Utilizing GPT-4, we identified the segments of the prompt corresponding to distinct objects. This process facilitated the generation of a two-dimensional array of prompt substrings, where each cell in the array is occupied by at most one object. The comprehensive prompt can be found in Appendix D. Following this, we utilized the grid representation produced by GPT-4 to generate images, applying cross-attention masks corresponding to each object in the prompt. Given the complexity of these prompts, the VISOR metric, which is designed to apply to simple spatial relationships between two objects, is not applicable. Consequently, we opted to explore qualitative results.

4 Results

4.1 Cross-Attention Control Experiments

We perform ablation experiments to explore which combinations of our cross-attention control technique, negative prompting, and adding a divider between cross-attention masks result in the best VISOR and Object Accuracy scores.

Q1: Does cross-attention masking improve VISOR and OA scores? Our cross-attention masking method improves the VISOR score over the direct prompting baseline from 0.067 to 0.134. Since the masking forces each of the two objects in the VISOR prompt to appear in the correct location of the generate, VISOR scores are approximately equal to OA for all of our experiments.

Q2: What is the effect of negative prompting and dividers on VISOR and OA scores? Contrary to expectations, negative prompting and introducing a divider between cross-attention masks decrease VISOR and OA scores. Initially, we anticipated these techniques would prevent "hybrid" object creation, such as half-dog, half-cat animals, thereby improving scores (see figure 2). However, as Table 1 shows, enlarging the mask divider from 0.0 to 0.3 of the image progressively worsens image quality. In all experiments, OA approximates VISOR, as objects, if generated, appear in correct locations. We hypothesize that divider incorporation reduces OA by shrinking each object's mask, making Stable Diffusion less likely to generate the object. Despite this, dividers seem to aid in object separation, maintaining a consistent background with minor artifacts.

Method	n_prompts	Neg prompting	Divider size	VISOR	OA
baseline	1000	n/a	n/a	0.067	0.1319
ours	1000	yes	0	**0.134**	**0.136**
ours	500	yes	0.1	0.111	0.114
ours	1000	yes	0.2	0.107	0.107
ours	500	yes	0.3	0.064	0.064

Table 1: A comparison of VISOR and OA scores for the baseline direct prompting method and our cross-attention masking method with different divider sizes between the two object masks. `n_prompts` is the number of random VISOR prompts sampled for each result (from a total of $25,280$ possible prompts). For each prompt, we generate 4 images.

Method	n_samples	Neg prompting	Divider size	VISOR	OA
baseline	1000	n/a	n/a	0.067	0.1319
ours	1000	yes	0.2	0.107	0.107
ours	1000	no	0.2	**0.121**	**0.122**

Table 2: A comparison of VISOR and OA scores in cross-attention masking method with and without negative prompting in the mask divider. Introducing the negative prompting reduced the VISOR and OA scores.

4.2 Auto-grid Results

We present randomly generated images from our auto-grid method next to directly generated images from the same seed and model. As shown in Figure 8, our method performs much better than the baseline on a vegetable display image prompt with three objects and specific spatial constraints. Our method consistently generates blue corn, potatoes, and tomatoes in roughly the right locations, whereas the direct method neither positions the objects in the correct locations nor generates all the vegetables in the prompt.

Figure 9 shows an instance where our method performs worse than the direct prompting baseline. This time we use a simple prompt with a common constraint, “A castle on a hill". The baseline method correctly renders this image all four times, but our method succeeds only once, sometimes neither generating a castle nor a hill.

5 Conclusion

In this work, we have proposed a novel method for achieving improved generative spatial control in stable diffusion models without the need for external grounding information. Our approach, which utilizes cross-attention control in the latent space, has demonstrated promising results in generating more spatially coherent images without the need for additional spatial information than the text prompt. Our best-performing approach has outperformed the baseline stable diffusion model in terms of Object Accuracy (OA) and VISOR score on images requiring binary object placement.

Our work has a few limitations. Firstly, we use Stable Diffusion 1.5, not the latest 2.1 version, as we build on Hertz et al.’s [4] cross-attention control code. This could account for our lower object accuracy scores compared to Gokhale et al. [2]. Secondly, our grid approach’s evaluation is limited as existing metrics like VISOR [2] only score images with two objects, necessitating a multi-object relationship metric. Thirdly, our grid approach sometimes fails to generate all prompted objects, especially when many objects are involved or their grid size is small. Incorporating approaches from recent works like Chefer et al. [1] could enhance object accuracy. Currently, our grid approach generates one object per grid cell. Future research could investigate generating overlapping objects within the same grid location, utilize Frechet inception distance [5] to assess if negative prompting and dividers yield more natural-looking images as anticipated, and explore strategies to preserve object accuracy as the cross-attention mask area diminishes.

References

[1] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models, 2023.

[2] Tejas Gokhale, Hamid Palangi, Besmira Nushi, Vibhav Vineet, Eric Horvitz, Ece Kamar, Chitta Baral, and Yezhou Yang. Benchmarking spatial relationships in text-to-image generation, 2023.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[4] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022.

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[7] 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, 2023.

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[9] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention, 2016.

[10] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers, 2022.

[11] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021.

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[13] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[14] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[15] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis, 2016.

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[17] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[18] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[21] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[22] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

A Additional Experiments

A.1 Hybrids

Results of the first experiment where the two prompts are fed into the right and left parts of the conditional noise, respectively, without any separation.

Figure 2: Prompts are *elephant* and *horse* for the leftmost image, and *cat* and *dog* for the other two.

A.2 Divider Experiment

The baseline experiments provide decent outcomes, but a notable issue arises in the form of hybrid objects being generated, as given in A.1. Our hypothesis suggests that this problem stems from the absence of a separation mechanism within the self-attention layers, which causes these layers to treat objects as a single entity. To overcome this limitation, we introduce a divider between the cross-attention masks, as explained in section 3.1.2. For all experiments (including main result and 1), we confine the cross-attention control to 80% of the left and right halves of the latent space and implemented zeroed cross-attention in the strip region. However, this approach leads to artifacts, as depicted in the leftmost image in figure 3. This could be attributed to the model lacking contextual information about the region with zeroed attention. To confirm this, we conducted an additional experiment where we used zeroed attention in the right half of the latent space. The resulting images are displayed as the three images on the right side in figure 3. Observing these images, it becomes evident that zeroed attention indeed induces an undefined region. This means we need to have more control in the strip region.

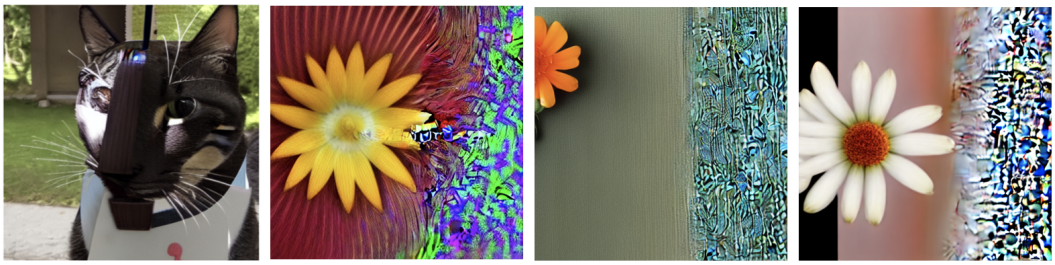

Figure 3: Leftmost image: Using zeroed cross attention in the middle strip to force separation (left prompt: “a cat", right prompt: “a dog"). All remaining: zeroed cross attention in the right-half region of the image (left prompt: “a flower")

Taking a step further, we replaced the zeroed attention with the negative embedding of the concatenation of the left and right prompt:

$$-\text{embedding}([left\_prompt, right\_prompt])$$

With this modification, the model demonstrates a tendency to generate separate objects instead of hybrids. However, a new issue arises where the middle strips exhibit artifacts similar to those observed in Figure 3. Furthermore, the generated images tend to be blurry.

Figure 4: Negative attention in the middle region of the image (left prompt: "a cat", right prompt: "a dog"). As before, the cross-attention control takes up 80% of both halves of the latent image.

Apart from negative embedding, we have also tried to replace zeroed attention with the embedding of the empty prompt. The results are given in figure 5. This approach can generate images with a coherent background, free from random noise artifacts. However, both sides of the image appear as if they were stitched together through the middle strip. Furthermore, the model could still produce objects in separate backgrounds, similar to what is observed in vanilla Stable Diffusion 1.5.

Figure 5: Injection of the embedding of the empty string in the middle region of the image

A.3 Divider+Background Experiment

In this particular experiment, we incorporated additional context while keeping the separation. To introduce context, we employed a background prompt. Unlike the previous experiment where we divided the image, we utilized 100% cross attention weights within the strip region and adjusted the percentage of the cross attention weights in the other regions. Through this approach, we achieved partial success. Specifically, by assigning appropriate weightage to the background, we successfully separated the objects while maintaining a coherent background. This outcome is clearly illustrated in Figure 6. However, there are certain limitations to be acknowledged. First, this method requires an additional background prompt to guide the generation process. Second, finding the optimal weightage for both low and high values is challenging, as neither extreme is ideal for achieving good results consistently. Furthermore, the weightage required for generating high-quality results is not fixed and may vary depending on the specific scenario. It is worth noting that the generation quality using this approach is poorer compared to other methods we have explored and experimented with.

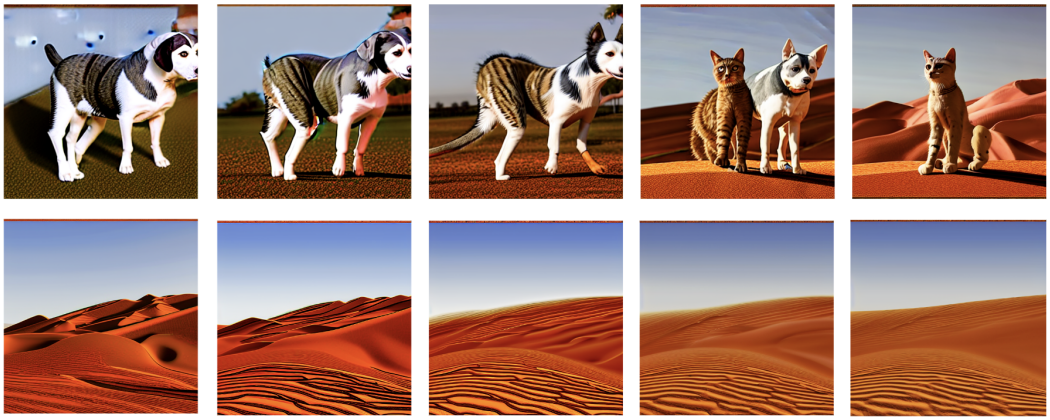

Figure 6: Introducing context though a background prompt. Left prompt: “A cat", right prompt: “a dog", background prompt: “desert"". The background weightage varies from 0 to 1 in increments of 0.11. The resulting images are arranged in a left-to-right and top-to-bottom order, starting top left.

A.4 Main result

The following are the results of our top-performing experiment. In this setup, the central area of the unconditional noise is filled with the negative prompt, while the surrounding area is filled with the unconditional prompt. The two prompts are fed into the right and left parts of the conditional noise, respectively, and are separated by the unconditional prompt

Figure 7: Prompts are *potatoes* and *corn* for the leftmost image, and *cat* and *dog* for the other two.

B Full Experimental Results

Table 3 contains results for all of our experiments and includes individual object accuracy metrics. The VISOR paper observes OA1 was several percentage points higher than OA2 most benchmarked models, indicating the direct prompting is more likely to generate the first object in the prompt than the second object in the prompt. Our `baseline` experiment corroborates this observation.

Our prompting technique only applies cross-attention of a single object's CLIP tokens to any one pixel. Furthermore, the first and second objects are uniformly placed on the top, bottom, left, and right strips of the image with equal probability. Therefore OA1 should be the same as OA2 in our experiments (all lines in Table 3 except the `baseline` experiment). The fact that OA1 and OA2 are so different in our experiments (for example in experiment `xattn01`, the gap between OA1 and OA2 is on the order of the gap between these metrics in `baseline`!) suggests that results at this level of random sampling have very high variance, and we need to use more compute and sample many more prompts and images to get representative results.)

Experiments `xattn02` and `xattn04` are equivalent because introducing a negative prompt to the divisor has no effect if the divisor size is zero.

n_prompts	Neg prompting	Divider size	Experiment	VISOR	OA	OA1	OA2
1000	n/a	n/a	baseline	0.067	0.1319	0.459	0.399
1000	TRUE	0.2	xattn01	0.107	0.107	0.37	0.371
1000	TRUE	0	xattn02	0.134	0.136	0.407	0.419
1000	FALSE	0.2	xattn03	0.1205	0.1217	0.4283	0.37575
1000	FALSE	0	xattn04	0.1255	0.1283	0.4002	0.42025
500	TRUE	0.1	xattn05	0.111	0.114	0.4115	0.3845
500	TRUE	0.3	xattn06	0.0635	0.064	0.2845	0.2945

Table 3: Full results table for all of our experiments. n_prompts is the number of VISOR prompts sampled for experiment. (For every prompt we generated 4 images.) Divider size indicates the proportion of the image that is allocated to the unconditional prompt as a divider between the left and right (or top and bottom) cross-attention masks. OA and VISOR scores are the Object Accuracy and VISOR scores as defined in Section 3. OA1 and OA2 refer to the single-object object accuracy scores of the first object in the prompt and the second object in the prompt respectively.

C Example auto-grid generations

Figures 8 and 9 compare our auto-grid image generations against baseline direction prompting on the same random seeds. Further discussion of these images can be found in Section 4.

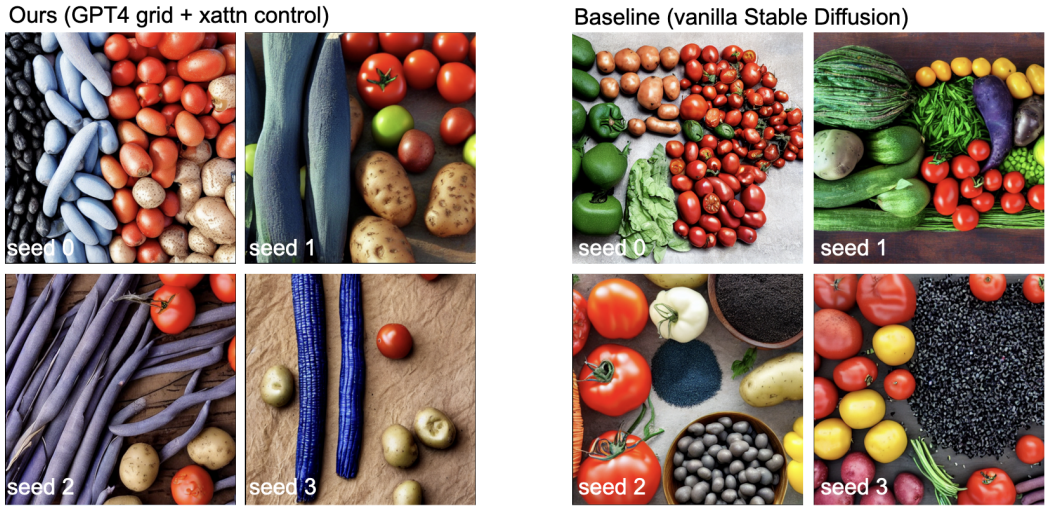

Figure 8: A side-by-side comparison of four random images generated our auto-grid method and images generated by direct prompting baseline on Stable Diffusion v1.4. The image prompt is “A vegetable display which consists of blue corn in the left half of the image, potatoes in the bottom left, and tomatoes in the top right." GPT-4 identified three objects in this prompt, “Blue Corn", “Potatoes", and “Tomatoes". For the auto-grid images we used a grid of dimensions 6×6.

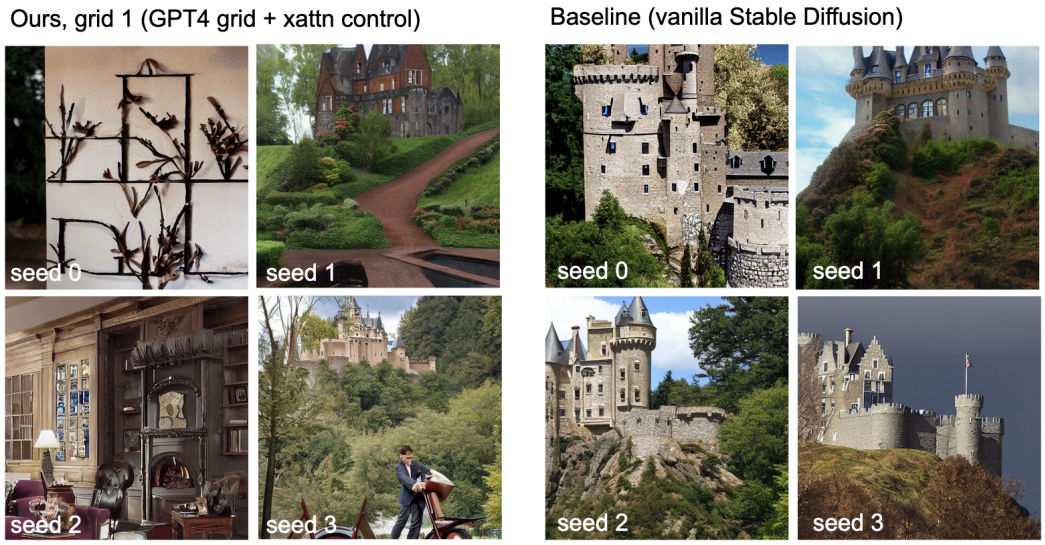

Figure 9: A side-by-side comparison of four random images generated our auto-grid method and images generated by direct prompting baseline on Stable Diffusion v1.4. The image prompt is “A castle on a hill." GPT-4 identified two objects in this prompt, “Castle" and “Hill".

D Full prompt for auto-grid experiments

We use the following prompt template to query attention masks from GPT-4. For easy parsing, we ask GPT to output its response in a JSON format. We enforce chain-of-thought reasoning [20] with the first two fields of JSON, where the language model is asked to list the different objects in the image and list the different spatial constraints between the objects.

Our auto-grid prompt template can creating attention masks from image captions with an arbitrary number of spatial constraints and an arbitrary number of objects. `{description}` and `{N}` are replaced with the image prompt (caption) and the side length of the grid respectively.

```
I need to map different objects in a description of an image to
    a NxN grid. Each cell of the grid can be assigned to an
   object or Empty.
I will give you the image description and you should respond
   with
(1) first a JSON object containing an array representation of
   the NxN grid and your reasoning.
(2) second a text representation of the NxN grid.

Example Input 1:
```
 Description: "A cat on the left and a dog on the right"
 N=4
```

Example Output 1:
```
{{
"reasoning_objects": "There is a Cat and there is a Dog",
"reasoning_horizontal": "From left to right, we have the Cat
   and then the Dog.",
"grid_text": "| Empty | Empty | Dog | Dog |\n| Cat | Cat | Dog
   | Dog |\n| Cat | Cat | Dog | Dog |\n| Cat | Cat | Dog | Dog
    |"
"grid": [["cat", "cat", "dog", "dog"], ["cat", "cat", "dog", "
   dog"], ["cat", "cat", "dog", "dog"], ["cat", "cat", "dog",
   "dog"]],
"prompt": "A cat on the left and a dog on the right",
"N": 4
}}

Empty	Empty	Dog	Dog
Cat	Cat	Dog	Dog
Cat	Cat	Dog	Dog
Cat	Cat	Dog	Dog
```

Example Input 2:
```
Description: "A cat on top of a book on top of a large table"
N=5
```

Example Output 2:
```
{{
"reasoning_objects": "There is a Cat, a Book, and a Table",
"reasoning_horizontal": "From top to bottom, we have the Cat,
   the Book, and the Table.",
```
```

```
"grid_text": "| Cat | Cat | Empty | Empty| Empty |\n| Cat | Cat
    | Empty | Empty | Empty |\n| Book | Book | Empty | Empty |
    Empty |\n| Table | Table | Table | Table | Table |\n|
   Table | Table | Table | Table | Table |"
"grid": [["cat", "cat", "empty", "empty", "empty"], ["cat", "
   cat", "empty", "empty", "empty"], ["book", "book", "empty",
    "empty", "empty"], ["table", "table", "table", "table", "
   table"], ["table", "table", "table", "table", "table"]],
"prompt": "A cat on top of a book on top of a large table",
"N": 5
}}
```

‘‘‘

The following examples will show the grid visualization for brevity, but for your responses, please remember include the JSON output.

Example Input 3: "4 cats in each corner starting at the dog in the middle"
Output:
Cat	Empty	Empty	Cat
Empty	Dog	Dog	Empty
Empty	Dog	Dog	Empty
Cat	Empty	Empty	Cat

Example Input 4: "A flock of birds flying above the mountains, a river in the foreground"
Output:
Birds	Birds	Birds	Birds
Mountains	Mountains	Mountains	Mountains
River	River	River	River
Empty	Empty	Empty	Empty

Example Input 5: "A single bird flying above the mountains, a river in the foreground"
Output:
Bird	Mountains	Mountains	Empty
Empty	Mountains	Mountains	Empty
River	River	River	River
Empty	Empty	Empty	Empty

Example Input 6: "Two birds flying above the mountains, a river in the foreground"
Output: | Bird | Bird | Mountains | Mountains |
Empty	Empty	Mountains	Mountains
River	River	River	River
Empty	Empty	Empty	Empty

Example Input 7: "Three birds flying above the mountains, a river in the foreground"
Output: | Bird | Bird | Bird | Mountains |
| Empty | Empty | Empty | Mountains |
| River | River | River | River |

```
| Empty | Empty | Empty | Empty |


Example Input 8: "Five birds flying above the mountains, a
    river in the foreground"
Output: | Bird | Bird | Bird | Bird |
| Bird | Mountains | Mountains | Mountains |
| Empty | River | River | River |
| Empty | Empty | Empty | Empty |



Input:
‘‘‘
Description: {description}
N={N}
‘‘‘
```