Improving Explicit Spatial Relationships in Text-to-Image Generation through an Automatically Derived Dataset

Anonymous ACL submission

Abstract

Existing work has observed that current text-to-image systems do not accurately reflect explicit spatial relations between objects such as *left of* or *below*. We hypothesize that this is because explicit spatial relations rarely appear in the image captions used to train these models. We propose an automatic method that, given existing images, generates synthetic captions that contain 14 explicit spatial relations. We introduce the Spatial Relation for Generation (SR4G) dataset, which contains 9.9 millions image-caption pairs for training, and more than 60 thousand captions for evaluation. In order to test generalization we also provide an *unseen* split, where the set of objects in the train and test captions are disjoint. SR4G is the first dataset that can be used to spatially fine-tune text-to-image systems. We show that fine-tuning two different Stable Diffusion models (denoted as SD_{SR4G}) yields up to 9 points improvements in the VISOR metric. The improvement holds in the *unseen* split, showing that SD_{SR4G} is able to generalize to unseen objects. SD_{SR4G} improves the state-of-the-art with fewer parameters, and avoids complex architectures. Our analysis shows that improvement is consistent for all relations. The dataset and the code are publicly available.[1]

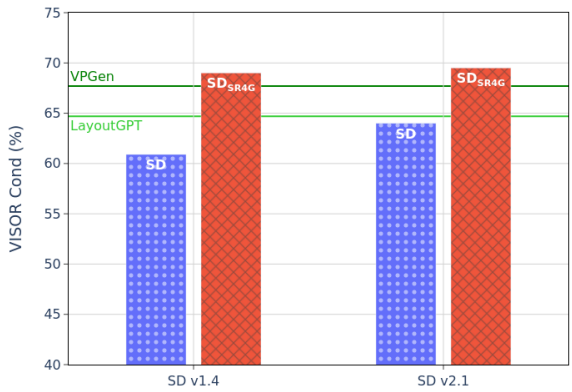


Figure 1: Fine-tuning Stable Diffusion on our SR4G dataset improves results significantly (two versions of SD shown), surpassing the state of the art in spatial-aware systems (see Section 4).

1 Introduction

Text-to-image generators such as Midjourney, Stable Diffusion (Rombach et al., 2022) and Dalle-3 (Betker et al., 2023) have recently made rapid advances and generated a lot of interest. However, those systems are still far from being perfect and show some important weaknesses. For instance, as observed by (Gokhale et al., 2023) and (Cho et al., 2023b) among others, current text-to-image generators do not represent well explicit spatial relations like *left of* or *below*, which limits their capabilities for important applications like text-based image editing (Kawar et al., 2023).

We hypothesize that the poor performance for explicit spatial relations is due to the lack of such relations in the datasets used to train those models. To support our hypothesis we analysed the LAION-2B dataset (Schuhmann et al., 2022), which has been used to train the state-of-the-art open source model Stable Diffusion. LAION-2B takes the captions from alt-text fields of images on the web. We automatically searched for explicit spatial relations (*left*, *right*, *below* and so on) and found that only 0.72% of cations contain the target words. Furthermore, 64.1% of these relations are *left* and *right*, which cannot be captured by image generators, as random horizontal flips are applied to images during training.

Motivated by the lack of captions with spatial relations, we focus on the training data to improve current end-to-end diffusion models; this is complementary to proposed architectural modifications on the system itself (Cho et al., 2023b; Feng et al., 2023). More concretely, we propose an approach

[1] Url will be announced upon acceptance.

to automatically generate synthetic captions which contain explicit spatial relations with paired real images. Leveraging the object annotations in COCO (Lin et al., 2014) and heuristic rules to infer the spatial relation between two bounding boxes, we build a dataset of real images paired with synthetic captions, called Spatial Relations for Generation (SR4G).

We use SR4G to fine-tune two Stable Diffusion models, assuming that exposure to image-caption pairs with explicit spatial relations will enhance the capabilities of the models to represent those relations. To evaluate our fine-tuned models and compare to the unmodified base models, we use the recently proposed VISOR metric (Gokhale et al., 2023), which we extend to support more spatial relations.

The contributions of this paper are the following: (1) We release SR4G, the first benchmark that allows to fine-tune, develop and evaluate the spatial understanding capabilities of text-to-image models for 14 explicit relations; (2) Our experiments show that fine-tuning Stable Diffusion on SR4G improves the understanding of spatial relations and provides more accurate images; (3) The improvement holds even when tested on unseen objects, showing that the models are able to learn the relations, generalizing to unseen objects; (4) The results exceed the state-of-the-art in spatial understanding for image generation (Cho et al., 2023b; Feng et al., 2023) with fewer parameters and avoiding complex architectures or Large Language Models.

2 Related Work

Many text-to-image systems have been proposed in the last few years. In general, we can distinguish between those based on auto-regressive transformer architectures, such as the original Dall-E (Ramesh et al., 2021), the multi-task system OFA (Wang et al., 2022) or CogView2 (Ding et al., 2022); and those based on diffusion models, pioneered by GLIDE (Nichol et al., 2022), which evolved into current latent diffusion models such as Stable Diffusion (Rombach et al., 2022) and Attend-and-Excite (Chefer et al., 2023).

Although the results of text-to-image systems keep improving, recent work has shown that their performance for explicit spatial relations is low (Gokhale et al., 2023; Cho et al., 2023b); the models struggle to correctly draw textual descriptions like *a cat on top of a table*. To overcome these limitations, VPGen (Cho et al., 2023b) and LayoutGPT (Feng et al., 2023) propose pipeline systems, combining Large Language Models to generate layouts from textual prompts and layout-to-image generators such as GLIGEN (Li et al., 2023). The difference between both systems is that VPGen fine-tunes Vicuna-13B (Chiang et al., 2023) to generate layouts from textual descriptions, whereas LayoutGPT relies on Llama-2-7B (Touvron et al., 2023) and in-context learning for the same purpose.[2]

To avoid the use of complex and large pipeline systems, (Yang et al., 2023) propose ReCo, an end-to-end system which uses layout descriptions in the input. In this paper, we also focus on end-to-end systems, but we avoid inserting layout information into the input, as this imposes a substantial burden on users compared to simple text inputs.

To evaluate the performance of text-to-image generators for explicit spatial relations, dedicated datasets have been created, since commonly used datasets like COCO (Lin et al., 2014), CC12M (Changpinyo et al., 2021) or LAION (Schuhmann et al., 2022), contain very few examples of explicit spatial relations. For example, (Gokhale et al., 2023) propose the SR_{2D} dataset, composed of synthetic captions created combining two objects in the COCO object vocabulary and four explicit spatial relations. SR_{2D} only contains captions and it is thus not amenable for training. Similarly (Feng et al., 2023) published the Numerical and Spatial Reasoning dataset (NSR-1K) which does include caption-image pairs. The spatial part contains only 1021 image-caption pairs (738 for train and 283 for test, no development) for 4 relations, insufficient for accurate evaluation and too small for training.

Our paper proposes a new dataset with synthetic captions **and paired images** which can be used to train and evaluate spatial understanding of text-to-image generation systems, containing 14 different spatial relations and including 9.9 million image/caption pairs (Section 3). Finally, for evaluating the generated images, we follow (Gokhale et al., 2023; Feng et al., 2023; Cho et al., 2023b) and use an off-the-shelf object detector to extract bounding boxes and compute the spatial relation between detected objects.

[2] Originally they use LLMs from the OpenAI GPT family, but they have released a publicly available Llama-2 based variant of LayoutGPT, which we use in this work.

3 SR4G: A new synthetic dataset for explicit spatial relation generation

Given the shortcomings of previous datasets, we propose to generate meaningful synthetic captions for real images, and use them to build the SR4G dataset (Spatial Relations for Generation). We increase the number of spatial relations used in previous work (Gokhale et al., 2023; Cho et al., 2023b; Feng et al., 2023) including not only projective or scale relations, but also topological ones. The full list of unambiguous spatial relations we used is as follows:

Projective: *left of*, *right of*, *above* and *below*.
Topological: *overlapping*, *separated*, *surrounding* and *inside*.
Scale: *taller*, *shorter*, *wider*, *narrower*, *larger* and *smaller*.

Our objective is to build a dataset for training, development and evaluation. For training, we need image-caption pairs, but for evaluation, captions with spatial relations are enough, since, following previous work (Gokhale et al., 2023; Cho et al., 2023b), the outputs of the image generator are not evaluated against real images. The evaluation method is described in Section 3.4.

3.1 Captions for evaluation

We first generate a set of spatial triplets of the form ⟨*subject, relation, object*⟩. We build our initial set of triplets using all pairwise combinations of the 80 objects in the vocabulary of COCO (Lin et al., 2014), yielding $3,160$ object pairs, and combining each pair with all of our 14 spatial relations, resulting in 88,480 spatial triplets.

However, some spatial triplets in the initial set are not *natural*. For example, it is very difficult to find natural images for triplets like ⟨*skis, above, toothbrush*⟩ or ⟨*truck, inside, cat*⟩. We want to remove those *unnatural* triplets from our dataset to focus on triplets that appear in natural images. Therefore, we identify all triplets that appear at least once in the training split of the COCO dataset and use that subset to generate our evaluation captions, which consists of 68.8% of the entire set of triplets (60,836 triplets).

Using hand-designed templates to be as simple as possible (Appendix A.1), we generate the final evaluation captions from the set of spatial triplets (Figure 4 shows some examples). Those captions reflect only the spatial relations between two objects, avoiding to include any other textual details.

3.2 Image-caption pairs for training

For training, we need captions with explicit spatial relations and real images in which those relations are depicted. We use the COCO 2017 training split to collect real images with object annotations and define a methodology to generate first spatial triplets from those images, and then textual captions derived from those triplets.

Given an image I and a list of n objects $O_I = \{o_1, o_2, \ldots, o_n\}$ belonging to I, the goal is to generate a triplet with a valid spatial relation r between two objects in O_I: o_s and o_o, where $s, o \in \{1, \ldots, n\}$. For each object o_i, we know its respective label l_i and bounding box (*bbox*) $bb_i = \{x_i^0, y_i^0, x_i^1, y_i^1\}$, that is, four coordinates that define the position and size of o_i in the image.

Therefore, $t_j = \langle l_s, r, l_o \rangle$ is a triplet defined in SR4G that is represented in I. We call this set of valid triplets $T_I = \{t_1, \ldots, t_m\}$, where m is the number of valid spatial relations in the given image I. This implies that each relation r has to be linked to a heuristic rule f_r where, given the *bboxes* of two objects, it determines whether a given triplet is instantiated or not (see Eq. 1). We follow (Johnson et al., 2018) and define f_r functions, which represent unambiguous spatial relations between two object bounding boxes (see Appendix A.2).

$$t_j = \langle l_s, r, l_o \rangle \in T_I \longleftrightarrow f_r(bb_s, bb_o) \quad (1)$$

We apply data augmentation strategies (random crops and horizontal flips) to the original COCO images in order to obtain an image I and its object list O_I. Then, we randomly select two objects as o_s and o_o, compute the list of valid relations using our predefined f_r functions, and randomly select one of these relations, building the j-th valid relation of I without computing the entire T_I set: $t_j = (l_s, r, l_o)$. Finally, we verbalize the obtained triplet t_j using the same hand-designed templates as for the evaluation captions (Section 3.1).

3.3 Dataset splits

We build two different splits of SR4G, namely the *main* and the *unseen* splits. The *main* split consists of all the spatial triplets/captions of the SR4G test set (see Section 3.1). The training instances are generated on-the-fly without any restrictions on the triplets, which means that the same triplet can be in train, validation and test splits. For the *unseen* split, we randomly divide the COCO dataset's 80 objects into training, validation and test sets of

Splits	Images	Unique Captions			I/C Pairs
		Train	Val	Test	
Main	103.4k	60.8k	2.5k	60.8k	9.9M
Unseen	83.6k	46.9k	2.5k	8.0k	4.8M

Table 1: SR4G dataset's statistics. *Images* column refers to the number of images used during training, *Unique triplets* column represents the amount of unique triplets, and *I/C pairs* refers to the number of unique image/caption pairs that can be generated.

$|O_{\text{train}}| = 45$, $|O_{\text{val}}| = 5$ and $|O_{\text{test}}| = 30$ objects, respectively. More specifically, during training we just take objects from O_{train} into account when randomly selecting *bboxes* to dynamically build spatial captions. For validation, as there are few combinations that can be built with O_{val}, we select triplets that contain one of these 5 objects at least once and do not contain any object that is set aside for the test split. For testing purposes we use triplets built by only using objects from O_{test}. Table 1 shows the relevant numbers of our splits (more details in Appendix A.3).

3.4 Evaluation metrics

To evaluate the performance of text-to-image systems for spatial relations, we use three evaluation metrics proposed by (Gokhale et al., 2023):

Object Accuracy: Given a generated image I' and two object labels l_a and l_b, object accuracy measures whether both objects appear in I'. We obtain a list of objects for I', i.e., $L_{I'} = \{l_1, \ldots, l_n\}$, by using an off-the-shelf open-vocabulary object detector, OWL-ViT (Minderer et al., 2022). This metric is useful for analyzing the object generation capabilities of an image generator, as it does not take the relation r into account.

$$\text{OA}(I, l_a, l_b) = \begin{cases} 1 & \text{if } l_a, l_b \in O_{I'} \\ 0 & \text{else} \end{cases} \quad (2)$$

VISOR: Given a generated image I' and a spatial triplet $t = (l_a, r, l_b)$, VISOR measures whether both objects appear and if the spatial relation r is valid between them. Function f_r takes the bounding boxes of both objects (bb_a and bb_b) and compares them to check if the triplet is valid. Bounding boxes are provided by the object detector. VISOR increases both when the model generates the requested objects and when the ratio of correctly generated relations increases, showing the ability of the model in visualising spatial triplets.

$$\text{VISOR}(I, t) = \begin{cases} 1 & \text{if } l_a, l_b \in L_{I'} \wedge \\ & f_r(bb_a, bb_b) \\ 0 & \text{else} \end{cases} \quad (3)$$

VISOR$_{\text{Cond}}$: This is the proportion of correctly generated spatial triplets, taking into account only images in which both objects are generated.

Given that our contribution focuses on spatial understanding, we focus on VISOR$_{\text{Cond}}$, as it quantifies the ability of the model to represent spatial relations correctly without considering its object generation capability. It is the most informative measure, specially when comparing between systems which might have different object generation abilities, as it isolates the understanding of spatial relations. We thus use it as our main performance metric in the experiments, although we also report the other two metrics, while extending the number of spatial relations from 4 to 14,

4 Experiments and Results

In this section we show that end-to-end models improve their capability of depicting spatial relations when they are fine-tuned with synthetic training examples. Furthermore, we find that our fine-tuned models SD$_{SR4G}$ generalize to unseen objects during fine-tuning.

4.1 Experimental set-up

Models. We use Stable Diffusion (SD) as the base model, as it shows the best performance on spatial relation generation among publicly available end-to-end models (Gokhale et al., 2023). We use two different versions of Stable Diffusion: SD v1.4 and SD v2.1, which generate images of 512x512 and 768x768 pixels, respectively.

Training. To fine-tune SD models on SR4G, we use the original loss function proposed by (Rombach et al., 2022), i.e., the mean square error over latent noise representations. We fine-tune SD models for 100k training steps with an effective batch-size of 64 instances, evaluating on the validation split every 5k steps. After training is complete, we select the checkpoint with the highest VISOR$_{\text{Cond}}$ value on the validation split. Following (Gokhale et al., 2023), we generate four images per spatial relation in all of our evaluations for consistency. More details can be found in Appendixes B and C.

Model	VISOR$_{\text{Cond}}$ ↑	VISOR ↑	OA ↑
Main split			
SD v1.4	60.9	17.6	29.0
SD v2.1	64.0	27.4	42.8
SD$_{SR4G}$ v1.4	69.0	26.8	38.9
SD$_{SR4G}$ v2.1	**69.5**	31.7	45.6
Unseen split			
SD v1.4	60.1	17.3	28.7
SD v2.1	64.0	28.4	44.4
SD$_{SR4G}$ v1.4	68.9	23.7	34.4
SD$_{SR4G}$ v2.1	**69.4**	29.4	42.4

Table 2: Results obtained for the *main* and *unseen* splits of SR4G. Base models SD v1.4 and v2.1 are shown alongside with fine-tuned SD$_{SR4G}$ models.

Model	Par.	VISOR$_{\text{Cond}}$ ↑	VISOR ↑	OA ↑
Main split				
LayoutGPT	8.1B	64.7	24.7	38.1
VPGen	14.1B	67.7	34.5	51.0
SD v2.1	1.3B	64.0	27.4	42.8
SD$_{SR4G}$ v2.1	1.3B	**69.5**	31.7	45.6
Unseen split				
LayoutGPT	8.1B	64.7	24.7	38.1
VPGen †	14.1B	68.4	37.0	54.1
SD v2.1	1.3B	64.0	28.4	44.4
SD$_{SR4G}$ v2.1	1.3B	**69.4**	29.4	42.4

Table 3: Comparison to the state of the art, including model size for both splits. † VPGen is contaminated, as it was trained on layouts containing spatial triplets that appear in our test split.

4.2 Main results

Table 2 shows the results for our base and fine-tuned models for both SR4G splits, with the best results according to the main comparison metric in bold.

Main split: We observe that the SD$_{SR4G}$ models improve all metrics respect to the base SD models, increasing both object and spatial relation generation capabilities considerably. These results are in line with our initial hypothesis, proving that the exposure to image-caption pairs with explicit spatial relations improves spatial relation generation. Our results show that SD$_{SR4G}$ v1.4 and v2.1 have almost the same spatial capabilities, but v2.1 excels for object rendering. Notice that the differences of the base SD models are much bigger.

Unseen split: To analyse whether the improvements of SD$_{SR4G}$ on the *main* split come from learning specific correlations between pairs of objects, or between objects and spatial relations, we check the results on the *unseen* split. The *unseen* split uses different objects in train and test, and it is thus designed to decouple objects from spatial relations, allowing us to focus on the performance for spatial relations in isolation. In Table 2, we see that both versions of SD$_{SR4G}$ consistently improve the VISOR$_{\text{Cond}}$ and VISOR metrics over the base SD systems, also for the *unseen* split. It is specially interesting that VISOR$_{\text{Cond}}$, which is not influenced by object accuracy, is almost the same as for the *main* split. That means that our models are generalizing to unseen objects during the fine-tuning step. The behaviour of both versions is very similar to the *main* split.

Image quality: As we are using synthetic captions to train, we make sure that the image generation capabilities of these models do not worsen over training. Therefore, we monitor the Fréchet Inception Distance (FID) (Heusel et al., 2017) between the model's generated images from human annotated captions (retrieved from the COCO 2017 validation split) and their respective real images. During all of our experiments FID values have been constant and have not worsen after training. A random set of examples can be seen in Figure 4.

4.3 Comparison with the state of the art

We also compare against two recent state-of-the-art pipeline models: LayoutGPT and VPGen. The backbone Large Language Model (LLM) of VPGen has already been fine-tuned for layout generation,[3] so we use VPGen with no further adaptation. Note that the layout generation module of VPGen has been trained on COCO, and thus contains the objects underlaying our test sets. In the case of LayoutGPT, adaptation is performed with in-context learning. We thus define a set of instances that will be used as in-context examples to condition the 7B parameter Llama-2 LLM. For this purpose, we randomly extract 400 caption-layout pairs per different relation from our SR4G dataset, and build a set of 5.6k instances of caption-layout pairs. For inference, $k = 8$ examples are chosen by computing the CLIP-based similarity (Radford et al., 2021) between the input caption and the set of in-

[3] They use three different datasets to obtain caption-layout pairs to fine-tune the LLM: Flickr30K entities (Plummer et al., 2015), COCO instances 2014 (Lin et al., 2014), and PaintSkills (Cho et al., 2023a).

Type	Relation	Main Split	Unseen Split
Projective	*Left of*	70.3 (+7.0)	69.8 (+8.8)
	Right of	72.4 (+8.0)	67.9 (+3.9)
	Above	72.0 (+4.5)	70.4 (+2.2)
	Below	71.4 (+4.5)	70.3 (+2.8)
Topological	*Overlapping*	86.9 (-4.9)	84.0 (-5.2)
	Separated	79.5 (+17.0)	84.8 (+18.5)
	Surrounding	29.8 (+2.3)	21.7 (-2.1)
	Inside	43.4 (-7.4)	39.2 (-6.4)
Scale	*Taller*	71.2 (+1.6)	75.6 (+5.0)
	Shorter	67.5 (+8.5)	69.0 (+11.9)
	Wider	71.6 (+4.3)	73.0 (+6.9)
	Narrower	69.3 (+9.3)	67.1 (+5.0)
	Larger	71.5 (+0.5)	74.7 (+1.9)
	Smaller	65.2 (+12.7)	63.3 (+13.5)

Table 4: $\text{VISOR}_{\text{Cond}}$ values per relation obtained by SD_{SR4G} v2.1. The difference in $\text{VISOR}_{\text{Cond}}$ between SD v2.1 and fine-tuned SD_{SR4G} is given in brackets.

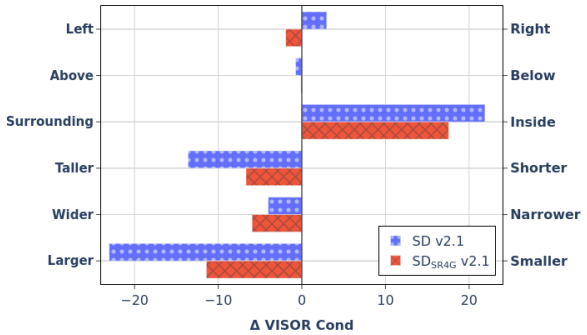


Figure 2: The horizontal axis depicts the difference of $\text{VISOR}_{\text{Cond}}$ values between relation pairs with opposing meanings defined on each side of the vertical axis. Results for SD and SD_{SR4G} v2.1 on the *unseen* split.

context examples, retrieving the top-k most similar examples and using them to condition the model to generate the proper layout.

Table 3 shows the obtained results for both SR4G splits. The same trend is observed, i.e. SD_{SR4G} v2.1 clearly outperforms both state-of-the-art pipeline systems in terms of $\text{VISOR}_{\text{Cond}}$, which measures the correctness of the spatial relation when both objects are generated. The improvement is especially important considering that both pipeline systems are significantly larger in terms of parameters, with a more complex architecture involving LLMs, and that both are specifically designed to generate scene layouts.

The table also shows the two auxiliary metrics, with VPGen obtaining the best results for object accuracy and VISOR. That is expected, since VPGen has been trained specifically for object generation, and VISOR is calculated over all the recognised objects. In fact, the better VISOR results are only due to better object accuracy, as our method produces better spatial configurations after factoring out object accuracy from VISOR ($\text{VISOR}_{\text{Cond}}$). Also note the contamination issue for the *unseen* split, as the text-to-layout step of VPGen has been fine-tuned on COCO. This implies that VPGen has seen text-layout pairs using the entire set of objects, having been trained on all the objects in our test set.

5 Analysis

We show an extensive analysis of the consequences of fine-tuning on SR4G, covering performance per relation, biases for opposite relations, performance by frequency of triplets and qualitative examples.

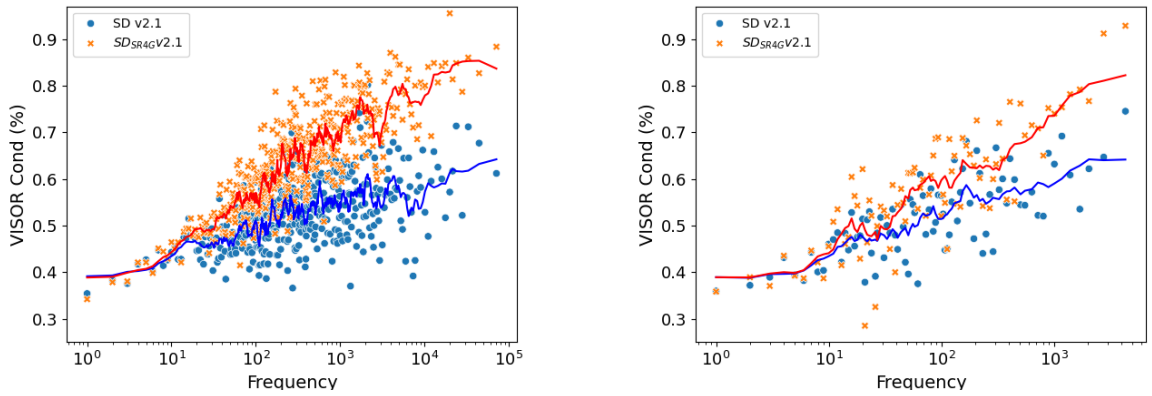
5.1 Analysing performance per relation

In Table 4 we show $\text{VISOR}_{\text{Cond}}$ values per spatial relation for SD_{SR4G} v2.1 (our best model), both in the *main* and *unseen* splits.

First, we observe that all projective relations significantly improve for both splits. The improvement is bigger for *left of* and *right of*. That might be due to random horizontal flips applied only to the images during the training of SD models, which are expected to damage the model's ability to correctly learn those relations.

Topological relations show a more variable behaviour. In the case of *separated*, our unique topological relation that does not involve generating overlapping objects, SD_{SR4G} is capable of improving its performance by up to 18.5 points $\text{VISOR}_{\text{Cond}}$. However, for *overlapping*, fine-tuning is not helpful. SD v2.1 already knows how to generate images with the *overlapping* relation, achieving $\text{VISOR}_{\text{Cond}}$ values of 91.8 and 89.2 in both test splits. On the other hand, *surrounding* and *inside* seem to be especially hard. The $\text{VISOR}_{\text{Cond}}$ values are low for the SD model and fine-tuning even makes them worse (especially for *inside*). This is a limitation of our current approach, and different training strategies must be explored to tackle this issue.

Finally, SD_{SR4G} improves for all scale relations. It is curious to observe that *taller*, *wider* and *larger* perform better than their opposites, even though the improvements over the base SD model are more modest. That suggests that the base SD model might have a bias towards those spatial relations.



(a) Results using *main* splits.

(b) Results using *unseen* splits.

Figure 3: Correlation between the frequency of SR4G triplets in COCO training instances (shown in the logarithmic horizontal axis) and their respective VISOR$_{\text{Cond}}$ results for SD v2.1 and SD$_{SR4G}$ v2.1. Triplets are grouped by frequency for visibility.

5.2 Analysing biases for opposite relations

Most of our relations have an opposite relation, i.e., *right of* is the opposite of *left of*. There are a total of six pairs of opposites in our relation set, which are listed in Figure 2 along with the difference in performance for these pairs before and after fine-tuning using the *unseen* split.

We want to see whether performance biases between opposites are reduced by fine-tuning. Figure 2 shows strong preferences of our base model SD v2.1 (in Appendix D, we show that those differences are correlated with the rate of appearance of each relation in the pretraining dataset of the SD models). We can also observe that SD$_{SR4G}$ v2.1 significantly reduces the difference in VISOR$_{\text{Cond}}$ between all relation pairs (except for *wider* and *narrower*), showing that fine-tuning reduces the inherent biases of the base model.

5.3 Performance by frequency of triplets

As SR4G is derived from natural images, some triplets are more frequent than others. To measure how the frequency of training triplets affects the results of our fine-tuned models, in Figure 3, we depict the VISOR$_{\text{Cond}}$ values of SD v2.1 and SD$_{SR4G}$ v2.1 depending on the frequency of each triplet in the COCO training set.

Figure 3a shows the results for the *main* split. In this case, the image generator has seen test triplets during training and, as expected, the more frequent these triplets, the greater the improvement after the fine-tuning. We can also observe that, even though SD models have not seen COCO images before, its performance is correlated with our computed frequencies.

On the other hand, Figure 3b shows a similar plot when training and evaluating on the *unseen* split. We observe similar correlations as in Figure 3a with both models. However, now we are evaluating on images generated from unseen triplets composed by objects that have not been seen during fine-tuning. Therefore, these results show that it is easier to transfer what is learnt to the most common triplets, even though we have not trained on them.

5.4 Qualitative Analysis

In order to visualize and qualitatively evaluate the generated images, we take SD v2.1 and SD$_{SR4G}$ v2.1 fine-tuned on the *main* split. We discard the most common and uncommon spatial triplets. The rationale is that the most common triplets often contain easy-to-generate relations (e.g., ⟨*truck, larger, dog*⟩) as generating both objects is enough to instantiate the relation itself, whereas the least frequent ones do not seem natural and would not be used in a prompt (e.g., ⟨*bus, shorter, traffic light*⟩). Therefore, we randomly pick triplets that occur between 100 and 1,000 times in COCO annotations (we obtain that range from the frequency analysis in Figure 3). We start generating images using random captions. We keep the first nine image pairs where both objects are generated correctly. Those nine pairs can be found in Figure 4, where we also indicate whether the spatial relation in the caption is depicted correctly or not.

Some of the captions of Figure 4 describe *easy* spatial relations, such as number 2, 3, 6, 7 and 9,

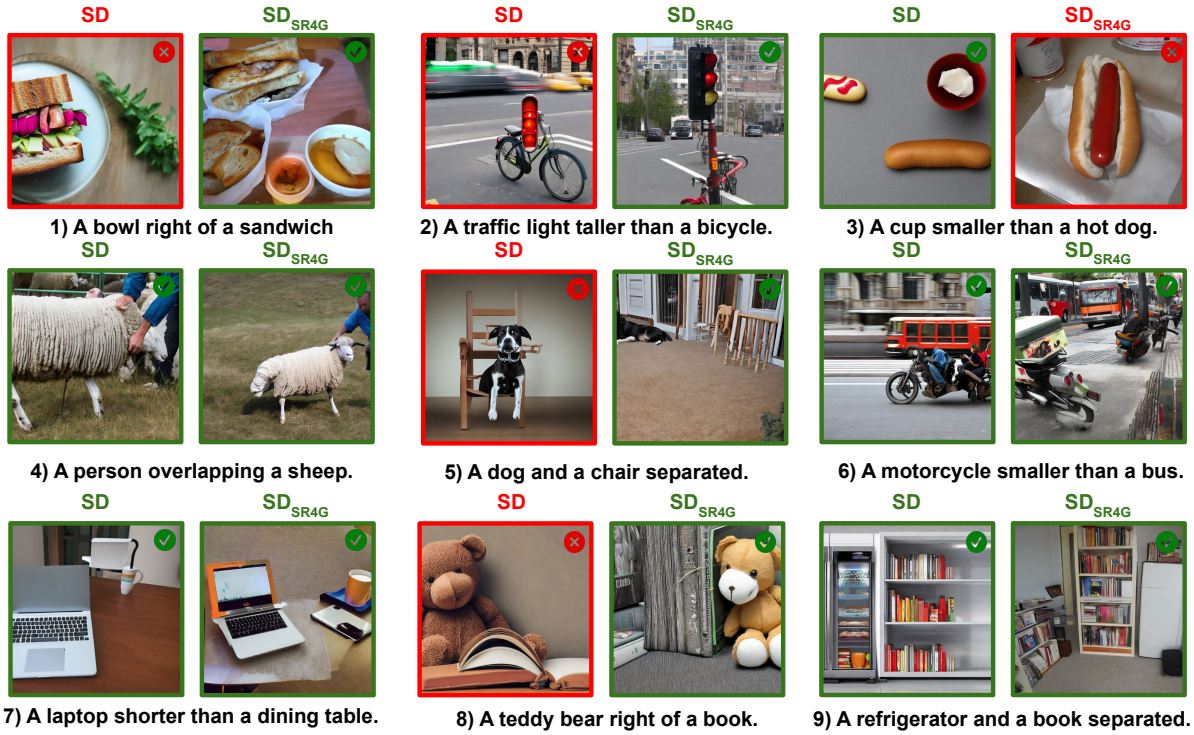


Figure 4: Image generation examples by SD v2.1 and SD$_{SR4G}$ v2.1 fine-tuned on the *main* split. Following our relation-specific heuristics, if the relation in the caption is correctly depicted, we indicate this with a green tick. Otherwise, there is a red cross in the top-right corner of the image.

where usually, if the correct objects are generated, the relation is also correct. SD$_{SR4G}$ generates those relations correctly, except for 3, which we denoted as a failure because the cup is not fully visible (the decision is arguable). SD fails for 2, rendering the traffic light very oddly. Captions 1, 4, 5 and 8 are more demanding: SD$_{SR4G}$ correctly depicts all the relations (*right of* twice, *overlapping* and *separated*), but SD fails for 1, 5 and 8. The failures are interesting: for 1 and 8, the spatial relations of the captions might not be the most typical ones in natural images, and SD struggles. However, for 5 it should be very common to see dogs and chairs separated, but SD does not follow the caption, which suggests that the relation *separated* is not known to SD.

6 Conclusions

In this work we define a dataset generation pipeline to build synthetic captions containing explicit spatial relations from COCO images and annotations. Fine-tuning diffusion models with these image-caption pairs outperforms the original diffusion models and also surpasses state-of-the-art pipeline models for spatial relation generation. We find that SD$_{SR4G}$ generalizes to unseen objects during fine-tuning. Further analysis shows that SD$_{SR4G}$ learns to better depict projective and scale relations, reduces the bias that the original model has for opposite relations, and generalizes better to spatial triplets that are more frequent in real images.

As future work, we plan to expand our relation set to include depth information with relations such as *in front of* and *behind*. We would also like to explore new ways to collect and annotate natural captions with spatial relations and evaluate state-of-the-art models with them.

7 Limitations

SR4G only contains captions in English, which limits its usage for non-English languages. To make it multi-lingual, caption generation scripts should be modified. On the other hand, SR4G is focused on unambiguous spatial relations defined over bounding box information, since they can be generated and evaluated automatically using off-the-shelf object detectors and heuristic rules. In that sense, orientation relations are discarded, even though their analysis is very interesting. Finally, we focus on 2D spatial relations. To introduce 3D relations should also be possible, using off-the-shelf depth estimation systems for images.

References

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng 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.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568.

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

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Jaemin Cho, Abhay Zala, and Mohit Bansal. 2023a. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3043–3054.

Jaemin Cho, Abhay Zala, and Mohit Bansal. 2023b. Visual programming for text-to-image generation and evaluation. *arXiv preprint arXiv:2305.15328*.

Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. 2022. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *Advances in Neural Information Processing Systems*, 35:16890–16902.

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

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

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, 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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. 2022. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. 2022. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and

Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.

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

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR.

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.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. 2022. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113.

A Details on SR4G Dataset

In this appendix, we give more details about our *main* and *unseen* splits, as well as defining our hand designed templates and heuristics used to determine whether an image contains a given spatial relation between two objects.

A.1 Hand designed templates

The templates we use to generate captions from spatial triplets are shown in Table 5. As can be seen, those templates are designed to be as simple as possible, omitting attributes and verbs and focusing only on the objects and their spatial relation. This is very important to analyse spatial understanding in isolation.

Type	Relation	Template
Projective	*Left of*	⟨A⟩ to the left of ⟨B⟩.
	Right of	⟨A⟩ to the right of ⟨B⟩.
	Above	⟨A⟩ above ⟨B⟩.
	Below	⟨A⟩ below ⟨B⟩.
Topological	*Overlapping*	⟨A⟩ overlapping ⟨B⟩.
	Separated	⟨A⟩ and ⟨B⟩ separated.
	Surrounding	⟨A⟩ surrounding ⟨B⟩.
	Inside	⟨A⟩ inside of ⟨B⟩.
Scale	*Taller*	⟨A⟩ taller than ⟨B⟩.
	Shorter	⟨A⟩ shorter than ⟨B⟩.
	Wider	⟨A⟩ wider than ⟨B⟩.
	Narrower	⟨A⟩ narrower than ⟨B⟩.
	Larger	⟨A⟩ larger than ⟨B⟩.
	Smaller	⟨A⟩ smaller than ⟨B⟩.

Table 5: Templates used to generate synthetic captions.

A.2 Heuristic rules

We use heuristic rules to both build the dataset and evaluate the generated images. Assuming the spatial triplet $\langle l_s, r, l_o \rangle$ and the bounding boxes of its objects bb_s and bb_o that appear in an image, we define the heuristic rule f_r of relation r to determine whether the triplet is fulfilled in the image or not. We set $bb_i = \{x_i^0, y_i^0, x_i^1, y_i^1\}$ by defining the top-left $\{x_i^0, y_i^0\}$ and bottom-right coordinates $\{x_i^1, y_i^1\}$ of the bounding-box (*bbox*).

For *left of*, *right of*, *above* and *below*, we follow the heuristic rules defined in (Gokhale et al., 2023), by computing the centroid of each *bbox* $c_i = \{x_i^c, y_i^c\}$ and comparing their corresponding coordinates.

As we expand to 10 more relations, we follow the rules described in (Johnson et al., 2018). In our scale relations we compare either the height (*taller* and *shorter*), width (*wider* and *narrower*) or area (*larger*, *smaller*) difference between both *bboxes*. In the cases of *surrounding* and *inside*, we check whether bb_o is contained in bb_s or vice versa. Finally, using the Intersection over Union (IoU) of both *bboxes*, we say that both objects are *separated* if their IoU is 0, and *overlapping* if their IoU is positive.

O_{Train}
person, *car*, *motorcycle*, *airplane*, *train*, *boat*, *fire hydrant*, *bench*, *bird*, *elephant*, *bear*, *giraffe*, *handbag*, *tie*, *snowboard*, *baseball bat*, *baseball glove*, *surfboard*, *cup*, *knife*, *spoon*, *apple*, *sandwich*, *orange*, *broccoli*, *carrot*, *pizza*, *donut*, *chair*, *couch*, *potted plant*, *bed*, *dining table*, *toilet*, *laptop*, *mouse*, *remote*, *keyboard*, *oven*, *sink*, *book*, *clock*, *teddy bear*, *hair drier*, *toothbrush*
O_{Val}
umbrella, *cake*, *tv*, *refrigerator*, *vase*
O_{Test}
bicycle, *bus*, *truck*, *traffic light*, *stop sign*, *parking meter*, *cat*, *dog*, *horse*, *sheep*, *cow*, *zebra*, *backpack*, *suitcase*, *frisbee*, *skis*, *sports ball*, *kite*, *skateboard*, *tennis racket*, *bottle*, *wine glass*, *fork*, *bowl*, *banana*, *hot dog*, *cell phone*, *microwave*, *toaster*, *scissors*

Table 6: Objects used in train, val and test sets of our *Unseen split*.

A.3 Main and Unseen Splits

Table 6 shows the sets of objects used for training, validation and test in the *unseen* split, which we refer to as O_{train}, O_{val} and O_{test}, respectively.

There are few combinations that can be built with O_{val} for validation in the *unseen* split, so we select triplets that contain one object from O_{val} at least once and do not contain any object that is set aside for the test split. In other words, there are up to $(2 \cdot |O_{\text{train}}| \cdot |O_{\text{val}}| + \binom{|O_{\text{val}}|}{2}) \cdot 14 = 6,580$ triplets that fulfil this rule (around 5,326 that naturally occur in the COCO dataset).

Validation is computationally costly in both splits, as several images have to be generated to compute the evaluation metrics defined in Section 3.4. Preliminary experiments showed that generating just 10k images is enough to get consistent results. Thus, we randomly selected 2.5k spatial captions for the validation splits for both *main* and *unseen* splits (as we generate 4 images per caption).

B Training settings

Hyperparameters: In Table 7 we define the hyperparameters used for training. Learning rate and optimizer parameters are the ones used during the pretraining of SD models, the other listed hyperparameters have been adapted to our available infrastructure. We also take advantage of Exponential Moving Average (Kingma and Ba, 2015) to update the parameters of the models with an AdamW optimizer (Loshchilov and Hutter, 2019) and we do not use any learning-rate scheduler. We do validation runs every 5k steps and do not set any early-stopping mechanism.

Hyperparameter	Value
Training steps	100k
Batch size	64
Learning Rate	10^{-5}
Optimizer	AdamW
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	10^{-8}
Weight decay	0.01
Mixed-precision	bf16

Table 7: Fine-tuning hyperparameters of the diffusion models.

GPU usage: Due to different memory needs, we use 2 and 4 NVIDIA A100 GPUs to fine-tune SD v1.4 and SD v2.1 models, respectively. In both cases we use an effective batch size of 64 by changing the amount of instances assigned to each GPU. Each of our fine-tunings need 3 days to be completed.

Data augmentation: During training we apply random horizontal flips and random crops to our images as a data augmentation strategy (resulting in I^* and O^j). Note that, random horizontal flips are common during the training of text-to-image models. This implies that spatial relations, such as *left of* and *right of*, can not be learnt correctly (as captions are not transformed according to those flips). Nevertheless, in our case we apply the same transformations to *bboxes*, which are used to generate captions synthetically, keeping this data augmentation strategy while maintaining the generated caption's spatial correctness.

Random crops might reduce the number of objects in O_{I^*}. If there are less than two objects after a given crop, we redo it up to max_iter times until there are at least two objects in the image.

We also define the hyperparameter k as the number of captions that can be concatenated to build the image-caption pairs built during training. Table 8 shows the results obtained by concatenating $k \in \{1, \ldots, 5\}$ captions. We observe that $k = 2$ obtains the best results, and we use this value of k

Nº Captions	$\text{VISOR}_{\text{Cond}}$ ↑	VISOR ↑	OA ↑
1	68.1	26.5	38.9
2	**69.4**	27.4	39.5
3	67.7	27.1	40.0
4	63.7	21.9	34.3
5	63.0	22.9	36.3

Table 8: We fine-tune SD v1.4 in the *main* split concatenating different amounts of captions in the input. These results correspond to the validation set of our *main* split.

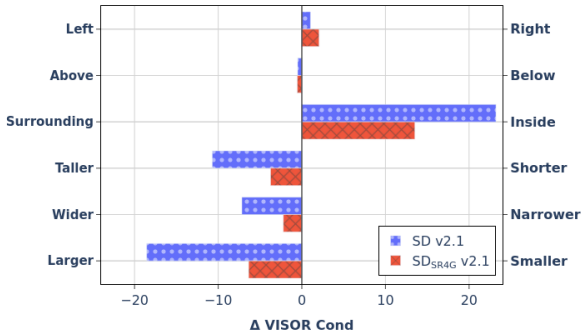


Figure 5: The horizontal axis depicts the difference of $\text{VISOR}_{\text{Cond}}$ values between relation pairs with opposing meanings defined on each side of the vertical axis. These results correspond to SD and SpaD v2.1 trained and evaluated using *main* splits.

Preferred Rel.	Opposite Rel.	Ratio of Appearance
Right (5M)	Left (5.6M)	0.91
Above (1.6M)	Below (0.7M)	2.47
Inside (2M)	Surrounding (0.3M)	7.61
Taller (49.3K)	Shorter (29.4K)	1.86
Wider (54.6K)	Narrower (5.7K)	9.62
Larger (0.8M)	Smaller (0.2M)	3.17

Table 9: Ratio in which the first relation appears more than the other. The relation in the first column is the preferred one by SD.

during our entire work.

C Evaluation settings

The evaluation metrics used in this paper use an object detector to determine whether objects are generated correctly and where are located in the image. Following (Gokhale et al., 2023), we use OWL-ViT, an open-vocabulary object detector that uses a CLIP (Radford et al., 2021) backbone with a ViT-B/32 transformer architecture (Zhai et al., 2022). We also set 0.1 as the confidence threshold of OWL-ViT, which determines how sure the model must be for a given region of the image to contain a specific object.

As an open-vocabulary object detector, OWL-ViT takes as input the objects we want to detect and, in order to do so, we use their recommended template ("a photo of a ⟨OBJ⟩.") instead of the object label alone.

Due to the variability of images generated by Stable Diffusion, we generate 4 images per evaluation caption. Therefore, we generate 10k images per validation and a total of 243.3k and 32.1k images to test each model in the *main* and *unseen* splits, respectively.

D LAION Dataset and Spatial Relations

Figure 2 shows that Stable Diffusion models have a strong bias towards some spatial relations, preferring *taller* to *shorter*, for instance. To complete those results, we also show the same graphic but in the *main* split, which exhibits a very similar behaviour (Figure 5). To understand the origin of those biases, we checked the frequency of each spatial relation in the LAION-2B dataset (English subset), used to train SD models. Table 9 shows the appearances of 12 relations, divided in 6 relation pairs with opposite meanings. Every relation has its number of appearances in LAION into brackets. For each opposite relation pair, the first column contains the relation that best works with SD. The third column shows the ratio of appearance between the preferred relation and its opposite (>1 indicates that the preferred relation appears more times in LAION than its opposite relation). The results indicate that there is a clear correlation between the ratio of appearance of a relation and the bias of SD models. The only exception is the *right* and *left* pair, but both appear similar times and the bias towards *right* is very small.