Exploring Point-E for Indoor Scene Text to Point Cloud Generation

Luyang Busser Alessia Hu Oline Ranum Sina Taslimi Luc P.J. Sträter
Miranda Zhou
University of Amsterdam
{luyang.busser, alessia.hu, oline.ranum,
sina.taslimi, luc.strater, miranda.zhou}@student.uva.nl

Abstract

In this study, we aim to explore the potential of layout control in 3D pointcloud diffusion by means of explainability. Our approach builds upon two recent advancements: the pointcloud diffusion model Point-E [24] and training-free layout guidance [5] in image diffusion. Our contribution is two-fold: first, we incorporate layout control into the text-to-image diffusion process within the Point-E pipeline. Secondly, we delve into the attention maps, the key component for layout control in 2D, of pointcloud diffusion and investigate the impacts of manipulating attention through masking mechanisms during inference. We discover further capabilities of the Point-E model not discussed in the original work, as well as intriguing attention trends that could potentially guide future research on layout control in pointcloud diffusion.

1. Introduction

Recent years have witnessed significant advancements in the field of 3D-object generation [6, 17, 19, 31]. One crucial aspect of the generation process is the ability to condition the output on textual or visual prompts, known respectively as text-to-3D or image-to-3D synthesis.

While current models have exhibited impressive proficiency in producing objects in isolation [10, 24, 25, 35], full scene generation remains an open problem [2, 4, 16]. In particular, indoor scenes pose additional challenges as they exhibit complex landscapes in terms of shapes, textures and functionalities, due to possible designs and indoors activities [8, 33].

Lately, a powerful class of generative methods known as diffusion models have emerged, demonstrating impressive abilities of producing high quality images [27, 28], videos [12, 14] and pointclouds [20, 24]. However, it is often the case that these models fail to interpret components of prompts that specify instructions on the spatial configuration of the composition. In order to mitigate this, several approaches [5, 9, 39] have been proposed that offer to steer the layout of the diffusion samples.

In the paper *Point-E: A System for Generating 3D pointclouds from Complex Prompts*, Nichol et al. [24] introduce a diffusion-based model which conditions generated pointclouds on textual prompts. Their model display an impressive ability to produce high-quality, synthetic pointclouds composed of singular items. However, during our exploration of Point-E we discovered that their text-to-pointcloud pipeline is not immediately suited for multi-object generation and composition control.

While exploring the model we identified two potential bottlenecks for scene generation: we discovered that conditioning the pointclouds on an image prompt was much more likely to produce satisfactory and well-resolved pointclouds, containing multiple objects. As such, we conclude that (i) complications are likely to have arisen in the decomposition of the text prompt and its interpretation into the visual domain, as demonstrated in Figure 3. However, while conditioning directly on the image space can yield reasonable results, it is often desirable to have control over the object layout in real-world applications. Unfortunately, (ii) the localization of objects in 3D space is known to be challenging for diffusion models, making them currently impractical for many real-life scenarios.

In this study, we explore approaches for generating indoor scenes using Point-E. Furthermore, we examine the information that can be derived from attention maps extracted from intermediate diffusion samples, and their potential role in layout control for the image-to-pointcloud domain. By considering the explainability of the model, we aim to derive insights that can aid in future research on developing controllable scene generation methods. Our main contributions are

1. We incorporate layout control on text-to-image diffusion in the Point-E pipeline and highlight the limitations.

2. We explore the attention maps of the modified point-E

pipeline and the effects of manipulating attention during inference.

3. Provide advice on the most effective way to continue investigating layout control for pointcloud generation based on our findings.

We release our analysis tools and demonstration notebooks at https://github.com/Bromosama/point_e_team10.

2. Related Work

Text-to-3D diffusion. Text-to-3D diffusion is the process of synthesizing 3D models from a text prompt. 3D digital content creation has many applications in different areas, such as gaming, entertainment, architecture, and robotics simulation [18]. By using text to generate 3D models, however, one may enable methods for describing and guiding content creation during production. Recent studies have proposed text-to-3D diffusion. DreamFusion [26] does this without resorting to any 3D data for training, by considering a 3D model from a Neural Radiance Field (NeRF) [22] and optimizing a loss on different renderings of the model. Magic3D [18] extends on this by considering a low-resolution diffusion prior and accelerating the computation with a sparse 3D hash-grid structure, leading to a 2x faster and higher quality model. [30] incorporates an in-depth depth map into a pretrained text-to-2D diffusion model to overcome the problem of lack of 3D awareness in such models. On the other hand, Point-E [24] considers a dataset consisting of millions of 3D models to train their model and generates 3D pointclouds from text prompts. Hence, we use this model for our problem and give detailed explanation about Point-E in subsection 3.1.

Indoor Scene Generation. Indoor scene generation refers to the process of creating virtual representations or simulations of indoor environments. It involves generating synthetic images or 3D models that mimic the appearance, layout, and characteristics of real-world indoor spaces, such as rooms, buildings, or architectural interiors.

Indoor scene generation has applications in various fields including virtual environments and simulations [21], data augmentation and synthesis [11], architectural design and visualization [15], content creation and entertainment [13], and human-robot interaction [3].

Recent work has proposed indoor scene generation conditioned on room layout [33] or through generative adversarial network (GAN) mehtods [34], or by providing a user-friendly task specification of the required objects in the case of [38]. Wang et al. in 2021 [33] also provides indoor scene generation from text, but uses a dataset of indoor scenes to train their model. Our work differs from previous work in that we generate simple indoor scenes from text prompts while not depending on a specifically designed input prompt or a dataset for indoor scenes.

Layout Control in Image Generation. Many works have been proposed for generating images with a controllable layout. This is especially useful for specifying the exact location of some or all objects inside an image that is desired. Many of such works do not operate on text prompts for image generation, instead, they focus on coarse spatial layouts consisting of bounding boxes and object categories [32, 36, 37, 40]. The layout serves as the input, and the model generates images based on the specified objects and their locations.

A text prompt can also contain the semantics of location, for example, "A frog to the right of a cat". Although text-to-image diffusion models can leverage their text prompt itself to guide where an object can be located, it can be seen that these models have trouble generating the objects correctly and may not generate all the objects in the prompt [5]. Furthermore, it is difficult to describe the exact location of an object using text.

The work proposed by Chen et al. [5], which we use for guiding 2D images we generate for indoor objects, incorporates layout control into Stable Diffusion by interpolating its cross-attention maps with the layout positions to guide where the model should look when generating the object, based on a selected word from the text prompt. Unlike previous works, this method does not require extra training for localizing the objects. More details are available in subsection 3.3.

3. Method

In this section, we present the method of the Point-E model. Next, we expand on how we use Stable Diffusion to improve the generation for indoor scenes. Lastly, we explore layout control by means of editing the attention maps.

3.1. Point-E

In this section, we present the methodology employed by Nichol et al. [24], referred to as Point-E, for generating pointclouds conditioned on text captions. Instead of training a single generative model to directly produce pointclouds, Point-E utilizes a three-step generation process, which we further extend and enhance in our work. An overview of the model is illustrated in Figure 1.

1. The first step in the generation process is to generate a synthetic view conditioned on the provided text prompt. This synthetic view serves as an intermediary representation, bridging the gap between textual information and generation. In the original paper, the synthetic view was generated with a *frozen* GLIDE model [23], *fine-tuned* on renderings of 3D models with no background. Next, the synthetic view is compressed into a latent embedding by a pre-trained CLIP VIT-l/14 model. However, in the code base they did not provide

this first stage of the pipe-line, except for some example outputs. From this point in the paper we will refer to this step as the **text-to-image stage**.

2. In the second step, Point-E produces a coarse consisting of 1,024 points. A conditional, permutation invariant diffusion model is trained to synthesize novel 3D pointclouds. diffusion was first introduced by Zhou et al. [41] and extended by the Point-E authors to include RGB colors. The network architecture for a single denoising step of the diffusion model comprises of 12 transformer layers. This reverse process is conditioned on the embedding generated in the previous stage. To train this model the authors used a large data-set of (image, pointcloud) pairs, which they did not open-source. However, they did provide the trained image-to-pointcloud diffusion model. From this point in the paper we will refer to this as the **image-to-pointcloud stage**.

3. Finally, in the third step, Point-E employs a pointcloud upsampler to refine the coarse pointcloud and produce a fine pointcloud consisting of 4,096 points. In this paper we will not expand on this stage further.

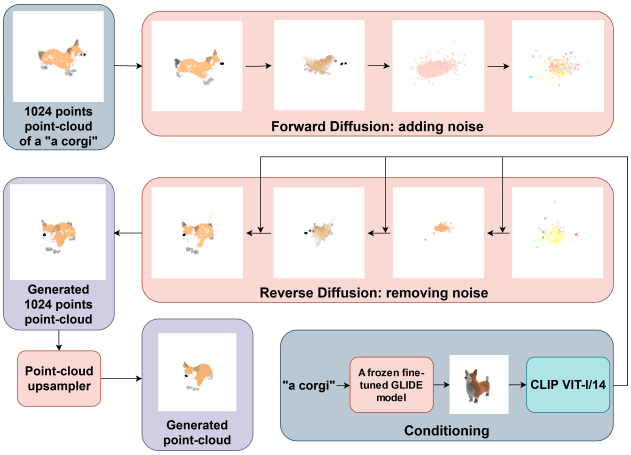


Figure 1. Point-E method.

3.2. View synthesis

As mentioned in the previous section the GLIDE model used by the authors is not publicly available. However, they do provide an open-source alternative to generate pointclouds from text. Instead of the embedding generated by the text-to-image stage, the image-to-pointcloud stage is now conditioned on an embedding generated by a CLIP text encoder as shown in Figure 2.

While this simplified method works reasonably well for simple objects, it fails to produce indoor scenes as we set

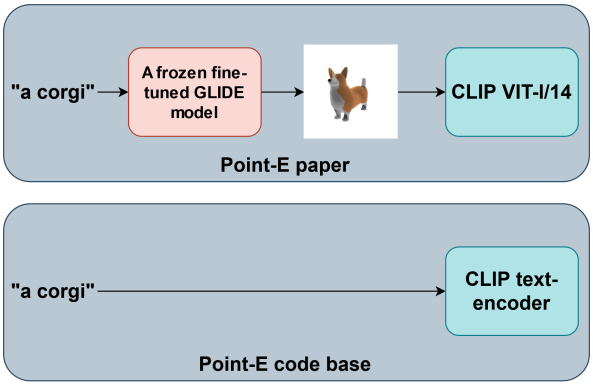


Figure 2. Difference between the conditioning module in the original paper vs the original code base.

out to do in this paper. To overcome this limitation, we employ Stable Diffusion to generate the synthetic views instead of GLIDE. In Figure 3 we show an example indoor scene. With this intervention Point-E can seemingly solve indoor scenes in a zero-shot manner, however it still struggles with what objects to model. In addition to the expected objects it also models shades, floors and walls from the generated synthetic view. Naturally, this outcome is considered undesirable. To address this concern, we use an automatic background remover, by removal.ai [1], effectively resolving these issues.

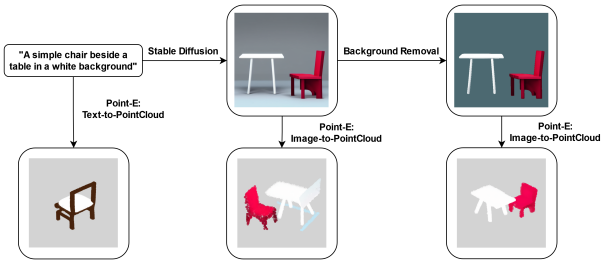


Figure 3. View synthesis with Stable Diffusion.

3.3. Layout control

While the previous section showed that Point-E can solve 3D scenes, it did so in a single object manner; it generates pointclouds with no sense of localization as it is not necessary for generating a single object. If the image-to-pointcloud stage is run on the same image multiple times, the orientation between the objects will be different every time and will not necessarily mimic the one of the image. This is highly undesirable behavior for real world applications. Thus, some form of layout control is needed.

Since retraining is unfeasible due to a lack of data,

training-free approaches are the only option. In the image domain this has recently been successfully applied by Chen et al. [5]. They achieve layout guidance by manipulating the cross-attention maps used by the model to perform text-conditioning on the image. The latter is achieved by attending to different parts of the text prompt, where each part of the image attends independently to different words through cross-attention. The image reconstruction is steered through backpropagation, following user-specified inputs, e.g. bounding box coordinates for each object. This method, called backward guidance, operates by imposing a bias such that each bounding box is associated to a token y_i. An overview of the process is shown in Figure 4. Initially, we investigated if using this approach in the text-to-image stage of our pipeline would enforce localization and orientation.

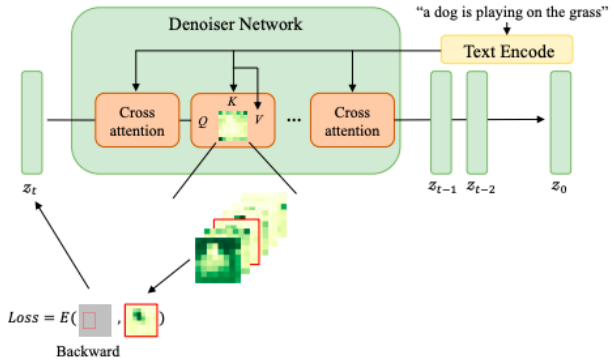


Figure 4. An overview of Layout Guidance pipeline [5].

To extend this method to guidance of 3D objects in an indoor scene, the attention maps of both the text-to-image stage and the image-to-pointcloud stage would have to be altered. Furthermore, these two would have to be linked to each other. While we do not achieve this goal in this paper, we explore how this problem could effectively be approached.

3.4. Explainability

Point-E exhibits an intriguing configuration concerning the attention mechanism. By incorporating an image as a conditioning input through concatenation, the attention mechanism encompasses both pointcloud-to-pointcloud attention and image-to-pointcloud attention within a single comprehensive matrix. This concatenation of the image implies that rather than introducing it separately at different stages of the diffusion process, it is combined with the pointcloud representation as input. Consequently, the input to the diffusion process consists of the image representation combined with the pointcloud representation. To examine the necessary modifications required for layout control, a thorough visualization and exploration of the attention maps is conducted.

3.4.1 Text-to-image layout guidance

To investigate how much of the pointcloud localization can be steered through text-to-image layout guidance, further images are generated with permutations to object positions: one where a chair is located on top of the table, and one where the chair is placed under the table.

3.4.2 Text-to-image attention maps

The Stable Diffusion model in the Layout Control paper [5] adopts the U-Net architecture [29]. The architecture comprises of three different branches: down-sampling, mid-sampling, and up-sampling. Each branch is made of several cross-attention blocks that contain three layers in the following order: ResBlock, Self-Attention, and Cross-Attention. The up- and down-sampling branches contain three cross-attention blocks, with the three layers repeated 3 and 2 times within their cross-attention blocks respectively. The mid-sampling branch has one cross-attention block, where the three respective layers appear only once. During latent updates, both the conditioned text embeddings and the unconditional text embeddings consist of 77 token vectors and are divided over 8 attention heads. Consequently, retrieving the attention maps from each branch results in a matrix of the following form:

$$\text{AttentionMap} \in \mathbb{R}^{(s,\, l,\, 2\times n,\, h\times w,\, d)}$$

where s equals the number of cross-attention blocks , l the number of layer repetitions, n the attention heads, h and w the size of the image, and d denotes the length of the token vectors. The exact dimensions of each cross-attention block are defined in the appendix Table 2. For running the inference, the used text prompt is

```
A simple chair beside a table in
a white background.
```

where the two phrases `table` and `chair` are associated to the bounding boxes with coordinates [0.1, 0.4, 0.5, 0.8] and [0.75, 0.4, 0.95, 0.8] respectively.[1] The model configurations, adopted from Zhang et al. [5] can be found in the appendix Table 1.
As mentioned in subsection 3.3, each token is associated to an attention map. Therefore, the attention maps are extracted and visualized at each time step for each token i, while being averaged over attention heads and all three layers within a single cross-attention block. As stated in Chen et al. [5], the down-sampling branch conforms to layout

[1] Bounding box values correspond to the following points: [x-min, y-min, x-max, y-max]

control minimally. Thereupon, most of the text-to-image attention maps analysis are focused on the mid- and up-sampling branches.

3.4.3 Image-to-pointcloud attention maps

Next, we go over the methodology of extracting the attention maps from the pointcloud diffusion model. For this we use the non-interactive Python debugger tool [7]. It allows the user to extract any stored variable in a particular method. In our case we hooked the forward method of the last attention block that calculates the query, key and value matrix.

After extracting the data, we proceed by plotting the attention maps on a grid and conducting an evaluation to analyze the orderings of the map in relation to the structured input embeddings. The attention maps have a dimension of 1282×1282. The input of the transformer is arranged so that index 0 corresponds to the time token, index 1:257 corresponds to the CLIP embeddings and index 257:1281 corresponds to the . The attention map as such refers to the relative attention between all input embeddings. As such, we refer to 4 distinct regions of the heatmap as image-to-pointcloud, pointcloud-to-image, image-to-image and pointcloud-to-pointcloud as demonstrated in Figure 5. Note that in this research we do not go in detail for the image-to-image domain nor the pointcloud-to-image domain. These areas of the attention map did not reveal any enough interpretable information therefore nothing conclusive can be deduced. Nevertheless, we did experiment on those parts of the matrix and some of these experiments will be discussed later in this report.

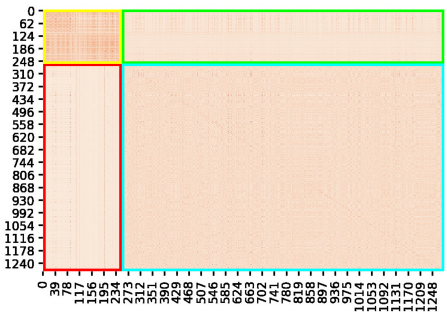


Figure 5. Naming convention for the attention map.

Our goal is to assess the structure of the attention map during the diffusion process, examining whether and how it is organized. To accomplish this, we conduct a series of experiments to qualitatively investigate the trends displayed by the map. These experiments include:

- **Attention clouds**: We overlay color-code the with the max & average attention received by each point. The that is being generated is conditioned on the same image as discussed in the image-to-pointcloud domain.
- **Point trackers**: We color-code a singular point in the pointcloud for a given index consistently across iterations, keeping all other colors diffuse.
- **Cross-Attention maps**: Trying to understand how the conditioned image is being attended to by the pointcloud diffusion process by visualizing the corresponding attention maps.

The "cross-attention" maps are derived by isolating either the red or green components of the complete attention matrix. These regions represent the attention allocated to specific portions of the conditioning image during the generation of the pointcloud. It is important to note that there are two distinct regions of interest. Through experimental investigations, we have determined that the red region contains the most significant and comprehensible information. These findings will be discussed in subsequent sections of this paper.

Lastly, we examine what happens if we manipulate the attention maps. We do this in a qualitative manner, where we edit part of the attention map and see what the results are on the generated pointcloud at multiple stages of the generative process. We change the attention values during inference with different strengths. Concretely, we mask parts of the attention map or multiply them by a factor. The manipulations that are tested are the following:

- **Uniform attention**: to learn how much is learned solely by the MLP part of the Transformer.
- **Pointcloud-to-pointcloud self-attention**: to learn how consistent the points are. In specific we explore the importance of the diagonal element of the attention matrix, i.e. how points attend to themselves.
- **Cross attention**: to learn what effect the attention mechanism has on the conditioning we investigate both image-to-pointcloud and pointcloud-to-image attention manipulation.

4. Results

The following section will present the most important results obtained from our experiments. As mentioned earlier, the pipeline consists of two parts, so the section will be divided accordingly. The first part will focus on the results of the text-to-image component, which uses Stable Diffusion. This will be followed by the image-to-pointcloud part. In this study, we build upon the work of the authors in the training-free layout-control research [5]. They modify the attention maps to constrain the diffusion process, which means that most of the figures in this paper involve

visualizations of attention and pointclouds resulting from attention manipulation. Therefore, understanding and manipulating attention maps are crucial for investigating both methodologies.

4.1. Text-to-image layout guidance

As shown in Figure 6, even though the synthetic view is enforced to place a chair on top of the table, Point-E will still generate the objects next to each other in the pointcloud. In addition, constraining the chair to be under the table resulted in just one object being generated in the pointcloud, where the chair is seemingly considered to be a part of the table.

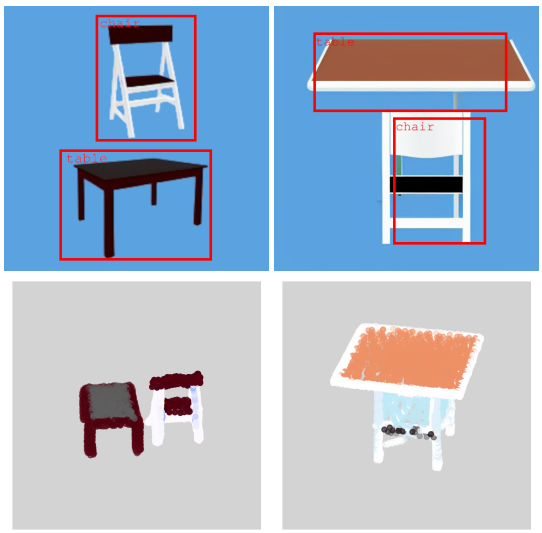


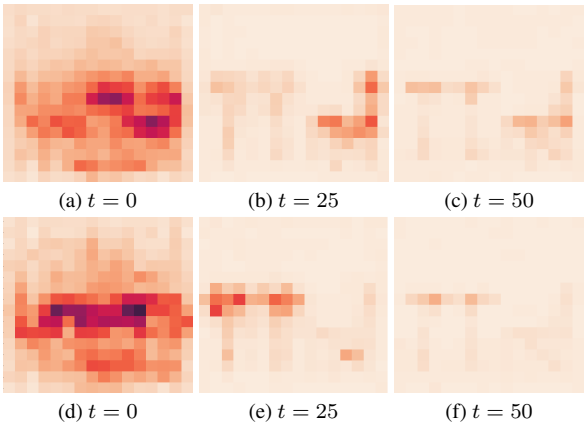
Figure 6. Generation with only 2D layout control. Left is the synthetic view generated by Stable Diffusion with layout control and right is the output by the Point-E image-to-pointcloud stage.

4.2. Text-to-image attention

The attention maps of the first block of the up-sampling branch can be seen in Figure 7. We can see that initially, the attention attends to the general localization in the image, specified by the bounding box coordinates inputted by the user. As the diffusion process further progresses, the attention focuses from outlining the coarse shapes of the objects to refining its details.

4.3. Image-to-Pointcloud attention

In this section, we present the results of our experiments conducted on the attention map of the pointcloud diffusion. We furthermore provide visualizations of what happens after modifying the attention map in the Point-E model.



(a) $t = 0$ (b) $t = 25$ (c) $t = 50$

(d) $t = 0$ (e) $t = 25$ (f) $t = 50$

Figure 7. Average attention map: the first row represents the token *"chair"*, the second represents the token *"table"*.

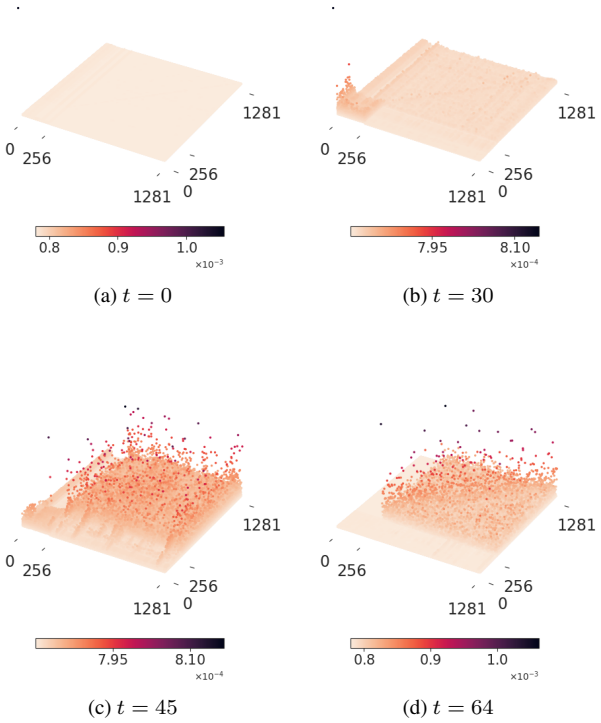
4.3.1 Attention visualization

Figure 8 showcases the 2D attention maps, where the attention weight is represented along the z-axis. The color coding is set for visualization purposes, where a darker color corresponds to a higher attention value. Our observations confirm the presence of four distinct regions that exhibit different behaviors throughout the diffusion process. We note that these regions align precisely with the borders corresponding to the structure of the input embeddings. This indicates a consistent association between the attention maps and the structured embedding spaces, which persists throughout the diffusion process.

Figure 9 showcases the average pointcloud-to-pointcloud self-attention map projected onto the pointcloud at various time steps. We observe that initially the attention values are dispersed throughout the entire pointcloud. As the diffusion process progresses, shapes gradually emerge, with the attention values and points focusing on outlining the main shape contours. In the final iteration, we notice a shift in attention toward the finer details of the target objects.

Figure 10 depicts the cross attention mechanism employed in the image-to-pointcloud process. Specifically, the figure showcases the mean attention bestowed upon each CLIP embedding by the model during pointcloud generation. These attention patterns are visually represented as maps superimposed upon the conditioning image, facilitating a more discernible comparison of their structural characteristics and facilitating subsequent discussions within this report.

The aforementioned inclusion of the two attention maps serves the purpose of establishing a connection between the attention structure exhibited by the pointcloud itself and the cross attention structure observed between the points and the conditioning image. By scrutinizing these attention maps, it becomes possible to identify and analyze the un-



(a) $t = 0$ (b) $t = 30$

(c) $t = 45$ (d) $t = 64$

Figure 8. Attention maps (Github Animation) for pointcloud diffusion. A strong color corresponds to a higher position along the z-axis, i.e. a stronger attention weight. The position along the xy-plane indicates the index in the attention map, where $x, y \in: \{0, \ldots, 255\}$ corresponds to the CLIP embedding domain, $x, y \in: \{256\}$ is the time embedding and $x, y \in: \{257, \ldots, 1281\}$ is the PC domain.

(a) $t = 0$ (b) $t = 30$ (c) $t = 65$

Figure 9. Attention cloud (Github Animation): The pointcloud color-coded with average self-attention

derlying relationship between the pointcloud's attentional behavior and its interaction with the conditioning image.

Observations reveal that in the pointcloud-to-pointcloud attention maps, there is an evident elevation in attention values allocated to corner and edge points. On the other hand, in the image-to-pointcloud attention maps, it is observed that around time step 35, the model exhibits greater attention towards the overall shapes of the components depicted

(a) $t = 15$ (b) $t = 35$ (c) $t = 55$

Figure 10. Cross attention maps (Github Animation): attention from CLIP embeddings overlaid on input image

in the image. Subsequently, in later stages, the attention shifts towards finer details, with a particular focus on the table. Interestingly, this is a similar pattern as observed in the Stable Diffusion attention maps (Figure 7). More on that in later sections. Next, visualizations of pointclouds are presented subsequent to the manipulation of specific sections of the attention map pertaining to Point-E. These outcomes hold substantial significance as they contribute to the assessment of the feasibility of executing layout guidance within Point-E.

4.3.2 Additional properties of the pointcloud

Figure 11 displays a single point tracked with a set index for multiple time steps. We observe that the tracked index has a consistent flow of motion throughout diffusion.

(a) $t = 40$ (b) $t = 45$ (c) $t = 60$

Figure 11. Point tracker (Github Animation): One index of the attention map is colored to display the correlation between a point in the and an index in the heatmap.

4.3.3 Attention manipulation

As mentioned in Section 3.4.3 we also investigate what happens to the pointcloud when we edit the attention maps. While we are mostly interested in indoor scenes, the task of editing attentions is very fragile and therefore it is hard to see trends in complicated scenes. Therefore this section is instead based on a more simple task were there are clearer visual trends, i.e. image-to-pointcloud diffusion for a stack of cubes. Figure 12 shows the base-line, where no attention manipulation happens. Note that for all manipulations a few iterations are shown here in the paper. For full GIFs of baseline, uniform, and masked image-to-pointcloud, image-to-image, pointcloud-to-image, pointcloud-to-pointcloud,

pointcloud-to-pointcloud (diagonal) attention, and amplified image-to-pointcloud, image-to-image, pointcloud-to-image, pointcloud-to-pointcloud, pointcloud-to-pointcloud (diagonal) attention we refer the reader to our github page. However, these GIFs are made with a lighter model version of Point-E as to speed up computation, and therefore, while the trends are the same, the exact frames do not match.

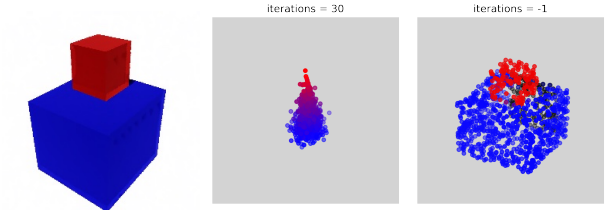


Figure 12. Base-line: no attention manipulation.

The most straight forward manipulation is to overwrite the attention to be uniform. The results of this are shown in Figure 14 in the Appendix. The most interesting thing to note is that without relying on attention the model initially can already find a cube-like shape. However, later in the model this shape collapses again. Next, we manipulated the pointcloud-to-pointcloud self-attention. Masking or amplifying these attentions out right resulted in similar results as uniform attention. Therefore, we decided to alter only the diagonal elements, which correspond to a point attending to itself. The results are shown in Figure 15 in the Appendix. Interesting things to note here are that masking the the attention of a point to itself does not seem to influence the final pointcloud much. Further, amplifying these attentions makes the model stick to the first shape it finds, in this case a single cube.

Finally, we also investigated editing the cross-attention between the image and the pointcloud. The results of image-to-pointcloud attention are shown in Figure 13. Most notably the effects of manipulating image-to-pointcloud and pointcloud-to-image attention seem to be opposite. Masking the image-to-pointcloud attention makes the model need more iterations to recognise that it has to model two objects and not one. This same trend is seen when amplifying pointcloud-to-image attention. In contrast amplifying image-to-pointcloud attention does not hurt separation of the objects in early iterations, but leads to difficulty it resolving the details. Lastly, masking pointcloud-to-image attention seems to just not be able to get any good initial shapes.

5. Discussion

From the text-to-image attention maps, it can be inferred that imposing bounding boxes successfully constraints the model to generate objects within their defined bounds. The

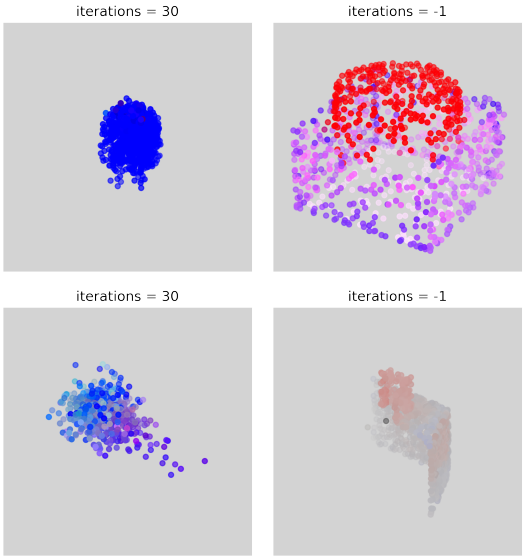


Figure 13. Image-to-pointcloud cross-attention manipulation. Top is masking the this part of the attention map and bottom is amplifying it with a factor 1.25 .

initial assumption was that localization of objects can be enforced through utilizing 2D layout control to an extent. However, as shown in subsection 4.1, this turned out to not be the case, with the objects being either placed next to each other or regarded as a single object. As the application of layout guidance in the text-to-image pipeline seems to be incapable of layout control in the pointcloud, further investigation is conducted towards the applicability of layout control in the image-to-pointcloud pipeline. To this end, the text-to-image attention maps are utilized to observe for any resemblances or differences with regards to the image-to-pointcloud pipeline.

5.1. Image to pointcloud pipeline

In order to assess the efficacy of layout guidance within the image-to-pointcloud pipeline, it is crucial to investigate the information contained in the attention maps. In the following paragraphs, we discuss what information can be concluded from the analysis on the image-to-pointcloud part of the pipeline, and how they compare to the text-to-image pipeline.

In Figure 8, we observed distinguishable behavior in the attention maps across the expected four regions with certain noteworthy global trends. We consider these results to be strong indicators that the structure of the initial embeddings in fact do have a direct correspondence to the structure of the attention map. In the earlier time steps, the most activity is displayed in the image-to-image self-attention region, before the values appear to disperse towards the image-to-

pointcloud cross-attention region. Towards the final stages of diffusion, the pointcloud-to-pointcloud domain becomes increasingly active. In particular, the region exhibits the emergence of a growing diagonal edge across the map, indicating that the points in the attend less towards their neighbors and the clip embeddings and increasingly more towards themselves.

Furthermore, we observed from Figure 11 that the pointclouds displayed consistent behavior in localization. However, it is important to note that we cannot yet conclude that each point in the pointcloud has a fixed corresponding index in the attention map. Determining this correlation is crucial to deciding whether the pointcloud-to-pointcloud self-attention domain can be used to effectively perform layout control through direct perturbation on the map.

By comparing the behavior of the attention maps between the text-to-image and image-to-pointcloud domain we observed that the attention values exhibited similar patterns. In both cases, in the earlier stages the attention mechanism appears to search more globally across the image-space. Initially, both processes tend to search globally across the image-space before attending towards the shape contours. Towards the final iterations, the attention localizes on the finer details of the target objects. This is an indication that both dimensions share certain reasoning strategies, which could be used to argue for applying similar strategies also in the cross-attention of the 3D domain.

Analysis of the pointcloud-to-pointcloud attention maps reveals that edge and corner points receive a higher average level of attention compared to points situated within homogeneous regions. This finding is both valuable and expected, as it indicates that the pointcloud-to-pointcloud attention captures the structural characteristics of shapes. On the other hand, the image-to-pointcloud attention maps exhibit a less pronounced yet still discernible structure. It is important to note that the image is embedded using CLIP, employing a vision transformer as the underlying backbone. Consequently, the embeddings contain inherent structural information.
During the intermediate stages of the diffusion process, we observe a noticeable shift in attention values within regions corresponding to objects in the image. This phenomenon coincides with the pointcloud starting to form individual components and delineating their respective locations. The implication of this observation is that the attention maps can potentially be modified, such as through masking, to encourage the diffusion process to focus more on specific parts of the conditioning image. This manipulation could serve as an initial step toward achieving coarse layout control, influencing the generation of large shapes. Subsequently, in the later stages of the diffusion process, the model transitions to a finer detail mode, prioritizing edges and corners of particular objects.

Lastly, the results of the attention manipulation indicate that for 3D layout control one would need to alter both image-to-pointcloud as well as pointcloud-to-image attention. Image-to-pointcloud is important in the early stages in resolving the initial shapes, while pointcloud-to-image attention is important in adding the details of the objects. It is important to note here that amplifying one part of the attention map automatically means scaling down the other parts of the attention map and vice versa. This is because eventually all attentions need to sum up to 1 and thus a softmax is used. Ultimately, if you amplify both image-to-pointcloud as well as pointcloud-to-image attention than the performance is almost the same as the base-line, while if you mask both you get the same as uniform attention. This is a strong clue that these are at least the attentions that need to be edited for 3D layout control.

5.2. Limitations and future work

Even though these experiments answers some questions about layout control for pointcloud generation, there are still many crucial unresolved questions regarding the complete guidance of this process. A few of them being:

- What methods can we apply to gain direct layout control during the diffusion steps of the pointcloud?
- What methods can we apply to achieve consistent orientation during the diffusion steps of the pointcloud?
- What happens to the diffusion process by combining the modification methods applied to the attention map?
- How to deal with the fact that point clouds, unlike images, are explicit scene representations?

Hence, to advance our understanding and explore potential strategies for achieving layout control in 3D environments, it is imperative to address these points. For instance, one could consider exploring weighted combinations of multiple modifications or selectively masking out specific regions within the attention matrix. By iteratively investigating these modifications, we can gain insights into their impact on the generation process and assess their potential for facilitating layout control.

Additionally, the results obtained from the image-to-pointcloud attention matrix exhibit promise. However, due to constraints on time, we were unable to thoroughly explore its potential implications in our current work. Therefore, it becomes crucial to devote sufficient resources to comprehensively investigate the image-to-pointcloud attention matrix and evaluate its effectiveness in achieving layout control. Especially because we observe similar patterns as the attention map in Stable Diffusion.

By systematically addressing these areas, we can further our understanding of the underlying mechanisms and potentially identify effective approaches for attaining layout control in 3D scenarios.

As mentioned earlier, masking or other modifications to the image-to-pointcloud attention map should alter the way the pointcloud is generated. However, for this project we lack conclusive evidence to show this in the report. Future work should investigate this more in-depth and explore the possibilities for layout control using the image-to-pointcloud attentions. Furthermore, orientation is also important factor of layout control in 3D. Think of how we can impose an orientation constraint during the diffusion process.

6. Acknowledgements

We would like to thank our teaching assistants Xiaoyan Xing, Ronny Velastegui Sandoval, and Partha Das for guiding us throughout the project.

References

[1] Removal A.I. Background remover api for developer, Aug 2021. 3

[2] Miguel Angel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, Afshin Dehghan, and Joshua Susskind. Gaudi: A neural architect for immersive 3d scene generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 25102–25116. Curran Associates, Inc., 2022. 1

[3] Niklas Bergström, Mårten Björkman, and Danica Kragic. Generating object hypotheses in natural scenes through human-robot interaction. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 827–833. IEEE, 2011. 2

[4] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks, 2022. 1

[5] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance, 2023. 1, 2, 4, 5

[6] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander Schwing, and Liangyan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation, 2023. 1

[7] Cifkao. Cifkao/nopdb: Nopdb: Non-interactive python debugger, Apr 2021. 5

[8] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W. Taylor, and Joshua M. Susskind. Unconstrained scene generation with locally conditioned radiance fields, 2021. 1

[9] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis, 2023. 1

[10] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images, 2022. 1

[11] Georgios Georgakis, Arsalan Mousavian, Alexander C Berg, and Jana Kosecka. Synthesizing training data for object detection in indoor scenes. *arXiv preprint arXiv:1702.07836*, 2017. 2

[12] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos, 2022. 1

[13] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1–22, 2013. 2

[14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. 1

[15] Satoshi Ikehata, Hang Yang, and Yasutaka Furukawa. Structured indoor modeling. In *Proceedings of the IEEE international conference on computer vision*, pages 1323–1331, 2015. 2

[16] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models, 2023. 1

[17] Yuhan Li, Yishun Dou, Xuanhong Chen, Bingbing Ni, Yilin Sun, Yutian Liu, and Fuzhen Wang. 3dqd: Generalized deep 3d shape prior via part-discretized diffusion process, 2023. 1

[18] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 2

[19] Zhengzhe Liu, Peng Dai, Ruihui Li, Xiaojuan Qi, and Chi-Wing Fu. Iss++: Image as stepping stone for text-guided 3d shape generation, 2023. 1

[20] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation, 2021. 1

[21] Pablo Martinez-Gonzalez, Sergiu Oprea, Alberto Garcia-Garcia, Alvaro Jover-Alvarez, Sergio Orts-Escolano, and Jose Garcia-Rodriguez. Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. *Virtual Reality*, 24:271–288, 2020. 2

[22] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 2

[23] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[24] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts, 2022. 1, 2

[25] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 1

[26] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[27] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 4

[30] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023. 2

[31] Zifan Shi, Sida Peng, Yinghao Xu, Yiyi Liao, and Yujun Shen. Deep generative models on 3d representations: A survey, 2022. 1

[32] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2647–2655, 2021. 2

[33] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*, pages 106–115. IEEE, 2021. 1, 2

[34] Ming-Jia Yang, Yu-Xiao Guo, Bin Zhou, and Xin Tong. Indoor scene generation from a collection of semantic-segmented depth images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15203–15212, 2021. 2

[35] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation, 2022. 1

[36] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8584–8593, 2019. 2

[37] Bo Zhao, Weidong Yin, Lili Meng, and Leonid Sigal. Layout2image: Image generation from layout. *International Journal of Computer Vision*, 128:2418–2435, 2020. 2

[38] Yizhou Zhao, Kaixiang Lin, Zhiwei Jia, Qiaozi Gao, Govind Thattai, Jesse Thomason, and Gaurav S Sukhatme. Luminous: Indoor scene generation for embodied ai challenges. *arXiv preprint arXiv:2111.05527*, 2021. 2

[39] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation, 2023. 1

[40] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22490–22499, 2023. 2

[41] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 3

7. Appendix

7.1. Workload distribution

- Luyang Busser: Extracting the attention of Point-E, visualising some attention maps (average attention and cross attention of point-e), initial exploration training-free layout control paper and make comparisons to Point-E. In report wrote big parts of results, discussion and limitation section, contribution to methods and introduction and related works. Coordinated meetings mostly, and main contact person with the TAs.
- Alessia Hu: Exploring the original code base for Point-E, such as trying different prompts and test the limitations of the model and get a better understanding of the pipeline, exploring the code base of layout-control and extracting and visualizing the attention maps for the different attention blocks and setting up a notebook, support in writing the report
- Oline Ranum: Restructuring repositories & readme, exploring codebase for layout control and Point-e, evaluating the application of layout control to point e, building demonstrations for attention map generations and result production, exploration of 3D attention maps and subsequent analysis, report writing.
- Sina Taslimi: Exploring the layout-control paper and code, visualizing attention of Stable Diffusion, exploring the Point-E paper and repository, writing the report
- Luc P.J. Sträter: Initial research into paper and code of lay-out control and Point-E. Then focused on image-to-pointcloud stage, specifically worked on the explainability part, in which I did the attention manipulation. For the paper I wrote the sections 3.1, 3.2, 3.3, 3.4 (partly), 4.1, 4.3 (partly), and 5 (partly).
- Miranda Zhou: Exploring code base of layout-control, research in U-Net architecture for understanding of the attention map structures, extracting and analyzing the text-to-image attention maps, setting up the notebook demonstrating text-to-image attention maps visualizations, support in writing the report and designing poster.

7.2. Attention manipulation figures

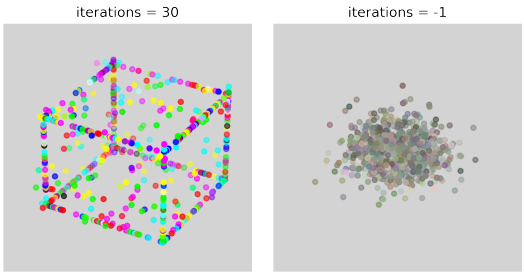


Figure 14. Uniform attention.

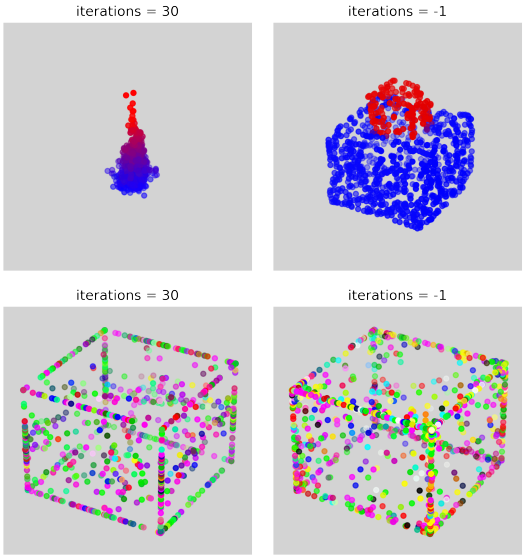


Figure 15. Diagonal of pointcloud-to-pointcloud self-attention manipulation. Top is masking the this part of the attention map and bottom is amplifying it with a factor 1.25 .

7.3. Layout Guidance configurations

Loss scale:	30
Batch size:	1
Loss threshold:	0.2
Max iterations:	5
Max index step:	10
Timesteps:	51
Guidance scale:	7.5
Random seed:	445
Noise Scheduler	
β start:	0.00085
β end:	0.012
β schedule:	scaled linear
num train timesteps:	1000

Table 1. Configurations of Stable Diffusion v1-5 with Layout-Guidance

7.4. Dimensions of U-Net branches

Branch	**Dimensions**
Up-sampling	$3 \times 16 \times 256 \times 77$
	$3 \times 16 \times 1024 \times 77$
	$3 \times 16 \times 4096 \times 77$
Mid-sampling	$1 \times 1 \times 16 \times 64 \times 77$
Down-sampling	$2 \times 16 \times 4096 \times 77$
	$2 \times 16 \times 1024 \times 77$
	$2 \times 16 \times 256 \times 77$

Table 2. Extracted dimensions for each cross-attention block