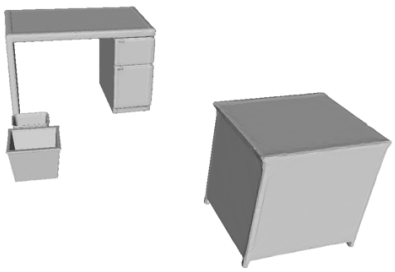
Text-DiffScene: Text-driven 3D Scene Synthesis with Permutation Equivariant Graph Diffusion

Ananth Kalyanasundaram, Yinyu Nie, Matthias Nießner
Technical University of Munich
{ananth.kalyanasundaram, yinyu.nie, niessner}@tum.de

Abstract

We address the task of scene synthesis by generating the object location, orientation and size of 3D objects of a scene in one go, conditioned on natural languages describing each object. To achieve this, we leverage diffusion models and explore their properties. We develop permutation equivariant diffusion models capable of processing scenes as a whole in one forward pass. While conventional scene generation approaches depend on the 2D or 3D representation of scenes alongside the location of objects, and assume the potential associations between them, our method does not utilize any visual information. Instead, we implicitly acquire object relationships through the attention layers used in our diffusion model. Later, we also use scene graphs and develop permutation equivariant graph diffusion models to generate scene graphs. This is the first effort that utilizes permutation equivariant diffusion models in order to generate the properties of 3D objects in a scene from language prompts.

1 Introduction



Text description:

- the recycling bin is blue color. a radiator is present behind this recycling bin.
- the gray trash can is beside the blue recycle bin, and near the brown desk in the corner of the room. the trashcan is empty.
- this is a dresser. its brown in color and is under the mattress.
- the wooden desk. it has three draws.

Figure 1: We address the task of 3D scene synthesis conditioned on natural languages.

Realistic 3D indoor scenes are highly valuable in various real-world applications related to 3D content creation. For instance, companies specializing in real estate and interior furnishing can visualize furnished rooms and their contents quickly without the need to rearrange any physical objects. These virtual rooms can be showcased through augmented or virtual reality platforms, such as a headset, providing users with the ability to walk through and interactively modify their future homes.

We address the task of scene synthesis conditioned on natural languages by generating objects and their arrangement. This is done by generating the objects' location, orientation and size. Consequently,

Preprint. Under review.

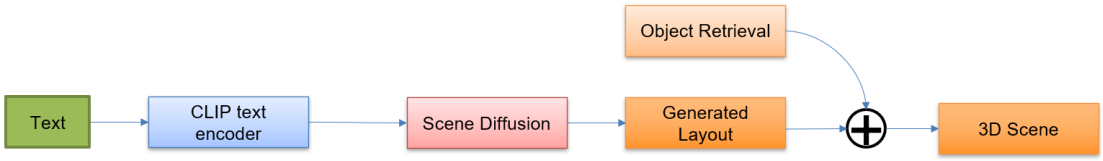


Figure 2: **Pipeline of our method**. As input, we take sentences describing each object in the scene and process it into 512 dimensional embedding with a pretrained CLIP text encoder. Conditioned on this embedding, we generate the location, orientation and size of each object in the scene. Then we use nearest neighbour retrieval from the CAD dataset to select objects for each bounding box.

the most relevant CAD model for each object is retrieved from a database and placed in the scene at the predicted location.

The conventional approach to modeling and generating scenes involves framing it as an optimization problem, where scene prior constraints are predefined based on design rules for the layout, object categories[1; 2; 3; 4], affordance maps[5; 6], or scene arrangement[7; 6]. In this line of work, the initial scene is sampled, and the scene configurations are iteratively optimized. However, defining these rules is a time-consuming and laborious process that requires the expertise of skilled artists. The scene optimization stage is also often tedious and computationally inefficient. Additionally, the pre-defined design rules can only represent basic and straightforward scene compositions, failing to capture all possible scenes.

For synthesizing complex scenes, deep generative models were used to learn scene priors [8; 9; 10] from large datasets. Methods based on GANs [11] aid in generating high quality results but they are of low diversity and often face mode collapse issues. We have also seen autoregressive models [12; 13] perform scene synthesis by predicting objects' information in a sequence, conditioned on the previous object's information. However, sequential predictions do not capture the relationships between objects effectively.

In the last few years, we have seen the emergence and success of denoising probabilistic diffusion models [14] towards the task of image synthesis [14; 15; 16] and shape generation[17; 18]. Diffusion models are known to produce high quality results with a higher diversity, while having an easier training regime compared to other generative models. We have seen graph based diffusion models being applied for the task of scene synthesis[19]. However this does not consider the set properties like order of the objects in a scene. Our method generates realistic scenes given text prompts of objects in any order.

Works on scene synthesis [13; 12; 19] until now, have just focused on synthetic datasets like 3D-FRONT [20] and SUNCG [21]. Our method is the first work, to the best of our knowledge that addresses the task of scene synthesis to a real life dataset and also takes into account the properties of scenes while using diffusion models.

Our work can be summarized as follows:

- we introduce permutation equivariant graph diffusion models which learn to produce diverse realistic scenes for indoor scene synthesis while respecting the properties of scenes.
- apply the problem of indoor scene synthesis to a real life dataset.

2 Related Work

Traditional Methods: Typically, the conventional approach involves transforming the problem into a data-driven optimization task. Many methods [3; 4] utilize a graph-based representation of objects in a scene to extract spatial relationships between them. Prior knowledge of reasonable scenes is necessary to synthesize plausible 3D scene graphs, with traditional scene priors inferred from interior design guidelines [1; 2], object frequency distributions [3; 4], affordance maps [5; 6], and scene arrangement examples [7; 6]. Using the above graph formulation and various optimization methods,

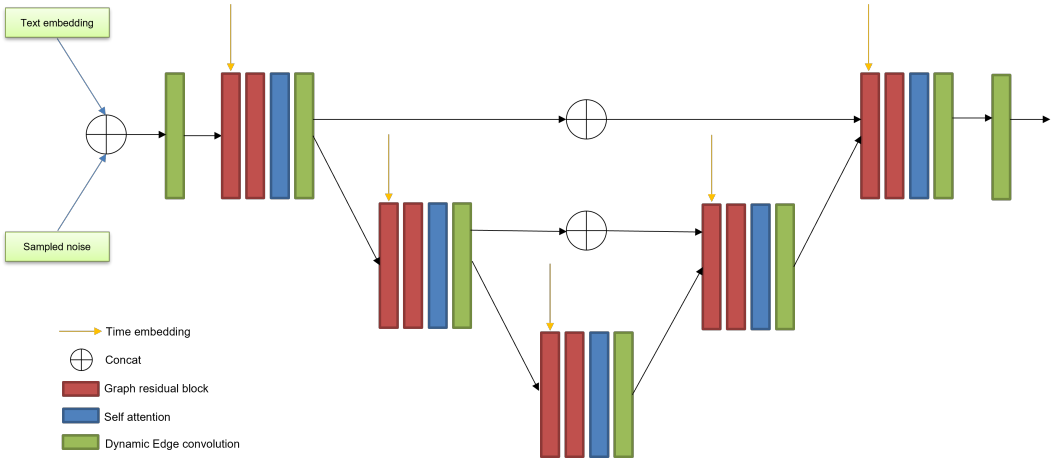


Figure 3: **Denoising UNet Architecture.** The denoising network takes in noisy object attributes and denoises them using edge convolutions with skip connections and attention blocks.

such as iterative or non-linear optimization or manual interaction, a new scene can be sampled while constrained by scene priors. In contrast to these methods, we adopt a scene graph representation to learn complex scene composition patterns from datasets, avoiding human-defined constraints and iterative optimization processes.

Deep Generative Synthesis: With the advent of deep neural networks, we have seen models capable of learning the distribution of large scale datasets. For the task of scene synthesis, a variety of generative models have been used from GANs[11], VAEs [10], autoregressive models [12; 13] and diffusion models [19]. Although GAN methods are known for their fast sampling and production of high-quality results, they tend to have limited diversity and suffer from poor mode coverage. On the other hand, VAEs[10] offer better mode coverage but face difficulties in generating accurate samples. Autoregressive models [12; 13] predict object information in a sequential manner conditioned on the previous object predictions. However this would limit the capability of the model to learn inter-object spatial relationships. Works on graph diffusion models such as [19] give out crisp realistic scenes, however these take text prompts of objects in a certain order, leading to noisy results when the order is permuted.

Diffusion models: Since it's inception, diffusion models [14] have been used in a variety of generative tasks. The most prominent of them are image synthesis [14; 15; 16], text-to-image synthesis [22; 23] and image inpainting [15]. We have also seen works in the 3D domain focusing on generating individual objects. However, unlike generating single objects, synthesizing 3D scenes involves a greater level of complexity in terms of semantics, geometry, and spatial extent. We have seen DiffuScene[19] which leverages diffusion models to generate object location, category and orientation. However their method does not consider the order of the text prompts describing the objects and produces noisy results when the order is changed. Our method produces realistic results regardless of the order of prompts.

3 Text-DiffScene

We introduce Text-DiffScene, a permutation equivariant scene graph denoising diffusion probablistic model which aims at learning the distribution of object location, size and orientation.

We consider scenes to be unordered sets of objects. Given a scene *S* containing atmost *N* objects $\{o_i\}_{i=1}^{N}$. Each node contains object information such as location, orientation and size. Due to varying number of objects in different scenes, we pad values of zero denoting empty objects to the scene. We add an objectness value to each object data denoting if it is actually an object or an 'empty' object, in order to make the model robust to zero paddings. To summarize each object $\{o_i\}_{i=1}^{N} = \{l_i, s_i, \theta_i, k_i\}$ where l, s, θ, k are the location, size, orientation and objectness of the object respectively.

3.1 Properties of Scene Graph

We define properties of scene graphs as unordered sets that need to be obeyed by our diffusion model.

- **Permutation Equivariance:** A function f is said to be equivariant to the action of a group G if $T_g(f(x)) = f(T_g(x))$ for all $g \in G$, where T_g is linear permutation related to the group element g according to []. For example, given a function f and a 3 element set $\{x_1, x_2, x_3\}$:

$$f(x_1, x_2, x_3) = \{y_1, y_2, y_3\} \implies f(x_2, x_3, x_1) = \{y_2, y_3, y_1\} \tag{1}$$

  This is particularly of importance for text conditioned synthesis since the prompts can be in any order. Hence the model should be able to produce realistic scenes given any order of the prompts.

- **Invariance to change in object taken as the origin:** Since the input text prompts only describe the location of an object with respect to another object, a relationship between absolute coordinates and the CLIP processed text embeddings is non-existent. To deal with this issue, we set the location of the first object in the scene graph as the origin and calculate the relative position of the other objects with respect to the first object. However changing the choice of the object taken as the origin results in a completely new data which the network will not recognize. Hence we require our model to be invariant to origin object choices.

$$\begin{bmatrix} 0 & 0 & 0 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{bmatrix} \neq \begin{bmatrix} 0 & 0 & 0 \\ x_1 - x_2 & y_1 - y_2 & z_1 - z_2 \\ x_3 - x_2 & y_3 - y_2 & z_3 - z_2 \end{bmatrix} \tag{2}$$

  where the first matrix is obtained by setting

Using this formulation we define our denoising diffusion probabilistic model.

3.2 Scene Graph Diffusion

We design a denoising probabilistic diffusion model where a series of Gaussian noise corruptions and denoising on graph nodes perform the transitions between the noisy and the clean scene graph distributions.

Forward Process Given a fixed 2D tensor $\mathbb{R}^{N \times D}$ where N objects of the scene, each object's location, size, orientation and objectness is concatenated to form a vector of dimension D. Starting with a clean scene graph x_0 from the underlying distribution $q(x_0)$, we gradually introduce Gaussian noise to x_0. This results in a sequence of intermediate scene graph variables $x_1, x_2..., x_T$, which have the same dimensionality as x_0, according to a pre-defined schedule of linearly increasing noise variance $\beta_1, ..., \beta_T$ (where $\beta_1 < ... < \beta_T$). The joint distribution $q(x_{1:T}|x_0)$ of the diffusion process can be expressed as:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{3}$$

where the diffusion step at time t is defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t|\sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{4}$$

One advantageous characteristic of diffusion processes is that we are able to sample x_t directly from x_0 by means of the conditional distribution:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t|\sqrt{\tilde{\alpha}_t}x_{t-1}, (1-\tilde{\alpha}_t I) \tag{5}$$

where $x_t = \sqrt{\tilde{\alpha}_t}x_0 + \sqrt{1-\tilde{\alpha}_t}\epsilon$ and $\alpha_t := 1 - \beta_t$ and $\tilde{\alpha}_t := \prod_{s=1}^{t} \alpha_s$

Generative Denoising Process The generative process is a reverse Markov chain that learns to reverse the Gaussian noise addition. Given a noisy input from a standard multivariate Gaussian distribution $x_T \in N(0, I)$ as the initial state, it corrects x_t to obtain a less noisier version x_{t-1} at each step using a learning Gaussian transition $p_\phi(x_t|x_{t-1})$ parameterized by a learnable network ϕ. Through repetition of this reverse process until the maximum number of steps T, we can reach the final state x_0, the clean scene graph we aim to obtain. Specifically, the joint distribution of the generative process $p_\phi(x_{0:T})$ is formulated as:

$$p_\phi(x_{0:T}) = p(X_t) \prod_{t=1}^{T} p_\phi(x_t|x_{t-1}) \tag{6}$$

$$p_\phi(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}|\mu_\phi(x_t, t), \Sigma_\phi(x_t, t)) \tag{7}$$

where the predicted mean and variance of the Gaussian x_{t-1}, got by feeding x_t into the denoising network ϕ, are $\mu_\phi(x_t, t)$ and $\Sigma_\phi(x_t, t)$.

Using the DDPM paper's findings [14], we estimate the mean and variance as by learning the noise.

$$\mu_\phi(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\tilde{\alpha}_t}}\epsilon_\phi(x_t, t)) \tag{8}$$

Denoising Network The denoising network is a UNet [24] that consists of MLPs with skip connections and attention layers [25], as seen in Fig.3. The attention layers extract the relations between objects in the scene. We discuss later about the choice of MLPs.

Loss function The loss can be formulated as the KL Divergence between the predicted distribution and the true distribution.

$$L_{diff} = KL(p(x_{0:T})|q(x_{1:T}|x_0)) \tag{9}$$

Using the derivations from the DDPM paper, we get :

$$L_{diff} = \mathbb{E}[||\epsilon - \epsilon_\phi(\sqrt{\tilde{\alpha}_t}x_0 + \sqrt{1-\tilde{\alpha}_t}\epsilon, t)||^2] \tag{10}$$

Text conditioned scene synthesis For text conditioning, we concatenate the text features obtained by processing the sentences using a pretrained CLIP [26] text encoder to the noise before being input into the denoising network.

3.3 Dataset

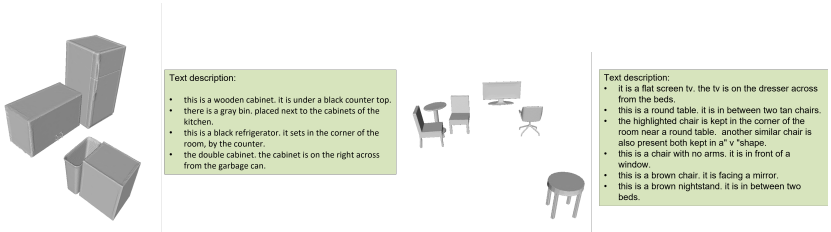


Figure 4: Examples from our dataset after linking Scan2CAD and ScanRefer datasets,

For our experiments, we use a real life dataset derived from ScanRefer [27] and Scan2CAD [28]. ScanRefer is a dataset derived from the ScanNet [29] which provides 51,583 descriptions for 800 ScanNet scenes. For our purpose, we use 5 descriptions for each object in 800 scenes. Scan2CAD is a dataset where the raw scans from ScanNet are replaced with ShapeNet [30] models aligned after optimization for 1506 scenes. From the Scan2CAD dataset, we get the object location, size, orientation and the corresponding mesh. Using instance ID from ScanNet as the link, we merge these two datasets to get a text-to-mesh-to-box mapping. In total we get 684 scenes each having 5 descriptions for every object. Considering a single set of descriptions for each object in every

scene, we get 3420 text-scene data points. We randomly shuffle and use 3290 scenes for training and 130 scenes for validation. We find that every scene from the dataset has atmost 40 objects and hence set N to 40 and considering object location, size, orientation and objectness, we set D to 8. To demonstrate the effect of permutation equivariance, we evaluate our models on randomly shuffling the prompt order of each scene in the validation set.

3.4 Baseline

For our baseline model, we use dynamic edge convolutions [31] as the MLP in the denoising network. We compare the baseline with diffusion models that use regular 1D convolutions along with attention and also spatial transformer network(STN) derived from PointNet [32]. The spatial transformer network learns a 3×3 align transformation matrix that transforms the object location into canonical space. We attach this STN to the end of the denoising network.

3.5 Permutation Equivariance

For creating a permutation equivariant diffusion model, we use 1D convolutions with kernel size of 1 as our MLPs. This also reduces the effect of overfitting due to the decrease in parameters. To achieve permutation equivariance while using edge convolutions we process the nearest neighbour dimension using 1D convolutions with a kernel size of 1, instead of using max pooling for aggregation as used in the DGCNN[31] paper.

3.6 Training

We train our diffusion models on the scenes from the training split. They are trained with a batch size of 16 on a single RTX 3090 GPU for 100,000 epochs. For our model to be invariant to changes in the choice of object origin discussed above, we randomly permute the objects and the corresponding text embeddings before it is input into the model. This ensure that our model is not pursuing some form of neighbour retrieval, and removes overfitting since it keeps learning a new data point at every iteration. Since, the average number of objects in a scene is 9, this leads to 362880 (9!) permutations atleast, thereby expanding the dataset drastically. We use a learning rate of $1e^{-4}$. For the diffusion processes, we use the default settings from the denoising diffusion probabilistic models (DDPM), where the noise intensity is linearly increased from 0.0001 to 0.02 with 1,000 time steps. At test time, we sample the scene graph from the predicted distribution, following the strategy provided by the DDPM [14] authors. We retrieve the closest CAD models from the database using the predicted size of the objects.

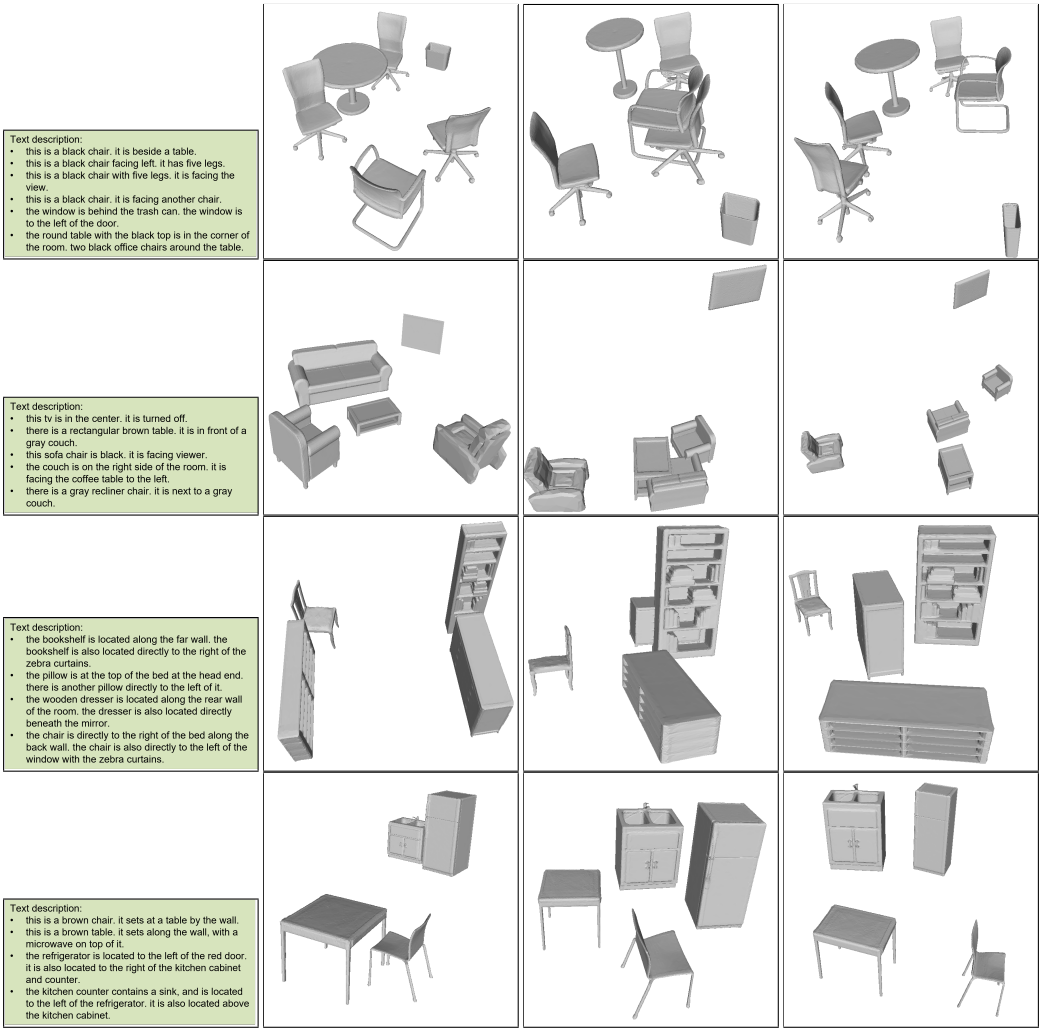
3.7 Evaluation Metrics

We use the KL divergence scores derived from the diffusion loss between the predicted and the ground truth distributions. Previous works use Frechet inception distance (FID) [33], Kernel inception distance [34] (KID $\times$ 0.001) between the top view rendered images of the predicted and the ground truth scenes. However these works utilize the 3D-FRONT and 3D-FUTURE [20] datasets which provide textures for objects and room layouts. Since we do not have textures available, our rendered images become equivalent to binary segmentation masks and hence yield very low FID and KID scores, which would not be fair to compare. As a result, we use CLIP [26; 35] score which has been used to evaluate many popular text-to-image diffusion models. CLIP-score measures the cosine similarity between the text and the image embeddings got from processing the rendered images of the scenes generated by the diffusion model.

4 Results

From Fig.5, we can see that our diffusion model with MLP and attention layers produces results with much less collisions than the model with STN in addition. We see that the scenes generated are diverse, although the semantic logic from the text prompts are mostly respected.

From Table.1, we see that our equivariant diffusion models significantly outperform the normal MLP + Attention diffusion model. This is because the normal diffusion model is not invariant to origin object changes and treats such data as an Out Of Distribution (OOD) data points. This leads to noisy



(a) Input (b) Reference (c) Ours + STN (d) Ours: MLP + Attention

Figure 5: Qualitative comparison of the methods. Generated samples from the permuted validation set

Method	KL Divergence	CLIP-Score[35]
MLP + Attention	0.3582	0.2178
Ours-Equivariant Edge Conv Graph Diffusion	0.1061	0.3615
Ours-Equivariant MLP + Attention	**0.0527**	**0.4046**
Ours-Equivariant MLP + Attention + STN	0.0552	0.4044

Table 1: Quantitative Results of the text conditioned scene synthesis.

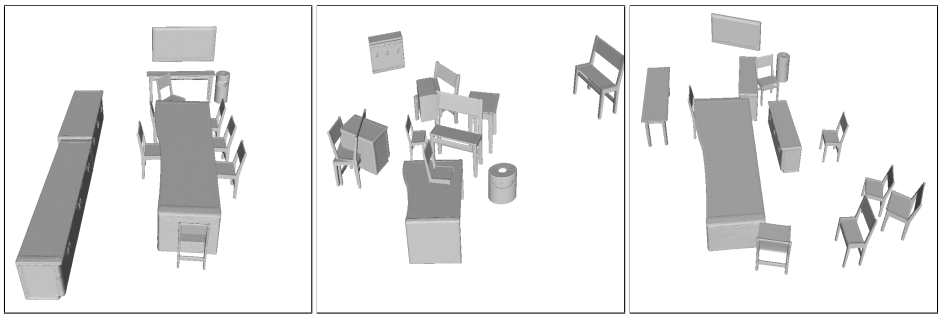
and meaningless results. Our permutation equivariant model which uses regular 1D convolutions as MLP along with attention layers achieves the best KL divergence and CLIP scores.

4.1 Ablation Study

Edge Convolutions: Using the equivariant model that leverages edge convolutions as MLP produces better results than the non-equivariant MLP models, but performs way worse and takes longer to converge than the equivariant model that uses regular 1D convolutions.

STN: Using STN in the model is observed to have a negative impact on the generated results. This results in a lot more collisions between objects.

Equivariant models: Equivariant models converge 10 times faster compared to the non equivariant counterparts. Overfitting is drastically reduced due to the smaller number of parameters. Invariance to change in origin objects, however leads to a longer time to converge. From Fig.6, we can observe that our permutation equivariant models produce realistic scenes even if the order of prompts and the origin objects are changed, as compared to the noisy results produced by the regular diffusion model.



(a) Reference (b) Normal diffusion model (c) Ours - Equivariant

Figure 6: Visual comparison of the generated results between a regular diffusion model and a permutation equivariant and an origin object invariant diffusion model.

5 Limitations and Future Work

We observe that even for our best models, object collisions occur frequently. To reduce object collisons, one could train diffusion models with deeper denoising networks. One could also postprocess the scene by minimizing the IoU between the bounding boxes of colliding objects. Since the strategy to randomly permute the order of the objects while training leads to a long duration to converge, one could look at alternative coordinate systems which could lead to origin object invariance, without the need to randomly permute.

6 Conclusion

In this work, we introduced Text-DiffScene, a novel method for generating 3D scenes conditioned on natural languages using permutation equivariant denoising diffusion models on scene graphs that learns the joint distribution of object location, size, orientation and objectness. Having estabilished a baseline, we compared it's performance to other variants and inferred that our equivariant models were vastly superior to regular diffusion models in terms of robustness to change in order of prompts and the origin object. We believe more advanced diffusion models could yield higher quality of results and less object collisions and we leave it to future work.

References

[1] Merrell, P., Schkufza, E., Li, Z., Agrawala, M., and Koltun, V., "Interactive furniture layout using interior design guidelines," *SIGGRAPH 2011* (Aug. 2011). To appear.

[2] Yeh, Y.-T., Yang, L., Watson, M., Goodman, N. D., and Hanrahan, P., "Synthesizing open worlds with constraints using locally annealed reversible jump mcmc," *ACM Transactions on Graphics (TOG)* **31**(4), 1–11 (2012).

[3] Chang, A., Savva, M., and Manning, C. D., "Learning spatial knowledge for text to 3d scene generation," in [*Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*], 2028–2038 (2014).

[4] Chang, A. X., Eric, M., Savva, M., and Manning, C. D., "Sceneseer: 3d scene design with natural language," *arXiv preprint arXiv:1703.00050* (2017).

[5] Fisher, M., Savva, M., Li, Y., Hanrahan, P., and Nießner, M., "Activity-centric scene synthesis for functional 3d scene modeling," *ACM Transactions on Graphics (TOG)* **34**(6), 1–13 (2015).

[6] Fu, Q., Chen, X., Wang, X., Wen, S., Zhou, B., and Fu, H., "Adaptive synthesis of indoor scenes via activity-associated object relation graphs," *ACM Transactions on Graphics (TOG)* **36**(6), 1–13 (2017).

[7] Fisher, M., Ritchie, D., Savva, M., Funkhouser, T., and Hanrahan, P., "Example-based synthesis of 3d object arrangements," *ACM Transactions on Graphics (TOG)* **31**(6), 1–11 (2012).

[8] Wang, K., Savva, M., Chang, A. X., and Ritchie, D., "Deep convolutional priors for indoor scene synthesis," *ACM Transactions on Graphics (TOG)* **37**(4), 1–14 (2018).

[9] Li, M., Patil, A. G., Xu, K., Chaudhuri, S., Khan, O., Shamir, A., Tu, C., Chen, B., Cohen-Or, D., and Zhang, H., "Grains: Generative recursive autoencoders for indoor scenes," *ACM Transactions on Graphics (TOG)* **38**(2), 1–16 (2019).

[10] Purkait, P., Zach, C., and Reid, I., "Sg-vae: Scene grammar variational autoencoder to generate new indoor scenes," in [*Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*], 155–171, Springer (2020).

[11] Yang, M.-J., Guo, Y.-X., Zhou, B., and Tong, X., "Indoor scene generation from a collection of semantic-segmented depth images," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 15203–15212 (2021).

[12] Wang, X., Yeshwanth, C., and Nießner, M., "Sceneformer: Indoor scene generation with transformers," in [*2021 International Conference on 3D Vision (3DV)*], 106–115, IEEE (2021).

[13] Paschalidou, D., Kar, A., Shugrina, M., Kreis, K., Geiger, A., and Fidler, S., "Atiss: Autoregressive transformers for indoor scene synthesis," *Advances in Neural Information Processing Systems* **34**, 12013–12026 (2021).

[14] Ho, J., Jain, A., and Abbeel, P., "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems* **33**, 6840–6851 (2020).

[15] Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S., "Sdedit: Guided image synthesis and editing with stochastic differential equations," in [*International Conference on Learning Representations*], (2021).

[16] Kim, G., Kwon, T., and Ye, J. C., "Diffusionclip: Text-guided diffusion models for robust image manipulation," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 2426–2435 (2022).

[17] Luo, S. and Hu, W., "Diffusion probabilistic models for 3d point cloud generation," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 2837–2845 (2021).

[18] Zhou, L., Du, Y., and Wu, J., "3d shape generation and completion through point-voxel diffusion," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 5826–5835 (2021).

[19] Tang, J., Nie, Y., Markhasin, L., Dai, A., Thies, J., and Nießner, M., "Diffuscene: Scene graph denoising diffusion probabilistic model for generative indoor scene synthesis," *arXiv preprint arXiv:2303.14207* (2023).

[20] Fu, H., Cai, B., Gao, L., Zhang, L.-X., Wang, J., Li, C., Zeng, Q., Sun, C., Jia, R., Zhao, B., et al., "3d-front: 3d furnished rooms with layouts and semantics," in [*Proceedings of the IEEE/CVF International Conference on Computer Vision*], 10933–10942 (2021).

[21] Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T., "Semantic scene completion from a single depth image," *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[22] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al., "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems* **35**, 36479–36494 (2022).

[23] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B., "High-resolution image synthesis with latent diffusion models," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 10684–10695 (2022).

[24] Ronneberger, O., Fischer, P., and Brox, T., "U-net: Convolutional networks for biomedical image segmentation," in [*Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*], 234–241, Springer (2015).

[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I., "Attention is all you need," *Advances in neural information processing systems* **30** (2017).

[26] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., "Learning transferable visual models from natural language supervision," in [*International conference on machine learning*], 8748–8763, PMLR (2021).

[27] Chen, D. Z., Chang, A. X., and Nießner, M., "Scanrefer: 3d object localization in rgb-d scans using natural language," in [*Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX*], 202–221, Springer (2020).

[28] Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A. X., and Niessner, M., "Scan2cad: Learning cad model alignment in rgb-d scans," in [*The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (June 2019).

[29] Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M., "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in [*Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*], (2017).

[30] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al., "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012* (2015).

[31] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M., "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)* **38**(5), 1–12 (2019).

[32] Qi, C., Su, H., Mo, K., and Guibas, L., "Pointnet: deep learning on point sets for 3d classification and segmentation. cvpr (2017)," *arXiv preprint arXiv:1612.00593* (2016).

[33] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S., "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems* **30** (2017).

[34] Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A., "Demystifying mmd gans," *arXiv preprint arXiv:1801.01401* (2018).

[35] Hessel, J., Holtzman, A., Forbes, M., Bras, R. L., and Choi, Y., "Clipscore: A reference-free evaluation metric for image captioning," *arXiv preprint arXiv:2104.08718* (2021).