ArtiScene: Language-Driven Artistic 3D Scene Generation Through Image Intermediary

Supplementary Material

1. Ablation

1.1. Target Labels for Detection

In Sec. 4 we mentioned we repeat the step of object detection twice: the first time after detecting furniture and small objects, we inpaint away the small objects, and detect for furniture and remaining small objects again. Thus, the detected furniture from the two runs stays the same amount, yet the detected small objects of the second round do not intersect with the detected set of first round, which have been removed by inpainting. We show the necessity of running twice in Table 1: counting the number of small objects in 184 generated scenes, more than 15% of them was detected in the second run.

First Run (Ratio %)	Second Run(Ratio %)
3118 (84.68)	564 (15.31)

Table 1. **# Detected Small Objects.** Out of the two runs of object detection, 15.31% of all decor or small objects were detected in the second run.

1.2. Post-Processing

Our de-occlusion post-processing is very effective in removing overlaps and placing the objects in a nearby reasonable position. As shown in Table 2, without our de-occlusion post-processing, the object overlapping rate increases. However, it will still be notably lower than the previous state-of-the-art LayoutGPT [2].

Scene	Ours (Full)	Ours (De-occlusion)	LayoutGPT
B	6.48	27.30	37.26
L	2.19	18.95	27.77

Table 2. **OOR Comparison.** B and L refer to bedroom and living room, respectively. Without our de-occlusion post-processing, the object overlapping rate increases (middle column), yet still notably lower than the previous state-of-the-art (right column). We do not consider margin when calculating overlaps.

(a) Original Segmentation (b) SDXL (c) DALL-E3 (d) Pix2Gestalt

Figure 1. **Different Inpainting Methods.** We found Pix2Gestalt has strong geometry awareness. This object is the television shelf in Fig. 4.

1.3. Pix2Gestalt Inpainting and ChatGPT Selection

After object detection, GROUNDING-DINO [3] performs segmentation within the detection 2D bounding box to generate more accurate masks. Such masks may contain holes that are due to occlusions by small objects, are lose a significant part of the object due to larger occlusions. We find Remove Anything [8] no longer works for the latter, as the prior of that model seems to be stitching up the areas with smooth textures that could blend with its surroundings. Here we need such textures, as well as preservation of the general contour implied by the occluded version of the object. Inputting the inverse of the segmentation mask as the inpainting mask, We found Dall-E3 [1] often fill up the background too aggressively that the geometry of the foreground object is changed. The inpainted content of the Stable Diffusion family depend heavily on the shape of the inpainting mask: for example, if the region for inpainting has a square shape, SDXL [6] tends to synthesize a square object instead of recovering the texture of our target shelf. After trying multiple combinations of different models and inpainting mask formulation, we found Pix2Gestalt [5] is the best at recovering the original geometry and texture of the target foreground object.

One caveat is that the output of Pix2Gestalt varies with seeds. As shown in Fig. 2, some results are more reasonable than others. To automate the selection of the best result, we prompt ChatGPT to select based on 3 aspects: (1) Realistic Object: how realistic it looks like the furniture category; (2) Complete Appearance: how complete the geometry and appearance is. If an image still holes or large occlusions on the object, it should have a lower score. And (3) Consistent Texture: how consistent the texture is. If the texture of one region is unrealistically inconsistent with its neighboring textures, such as a black spot, that is probably resulted from a failed inpainting, and such that image should have a lower score. Fig. 2 shows ChatGPT succeeds at picking one of the best results.

1.4. 3D Generation Conditioning

We show the importance of generating 3D assest conditioning on both text and segmentation image in Fig. 3.

Figure 2. **Pix2Gestalt and ChatGPT Selection.** We use Pix2Gestalt to inpaint an incomplete audio. From seed 0 to 5, the results with seed 0 and 4 are more complete, ChatGPT successfully chose seed 0 as an satisfactory input for our following image conditioned 3D asset generation.

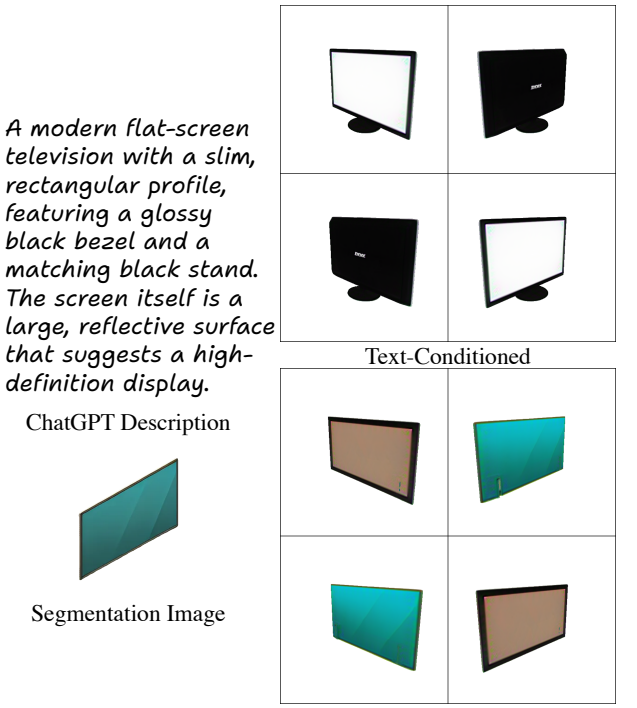


Figure 3. **Conditioning on Text and Image vs. Text Only.** This object is the television in Fig. 4. We offer four perspectives of the 3D television model generated with text only (top right) and text plus the segmentation from the image intermediary (bottom right). With only text as inputs, the asset clearly adheres less to the original appearance in the image intermediary.

1.5. Pose Estimation

For pose estimation, we tried a state-of-the-art method focusing on real life robotic tasks, FoundationPose [4]. These methods often assume the 3D model of the target object is available, and could be rendered at different poses with known camera parameters. These rendered images would then be compared with the object seen in real life, and the pose of the closest image becomes the estimated pose output. Such approaches have two gaps from our use case: first, our image intermediaries are not rendered by cameras with

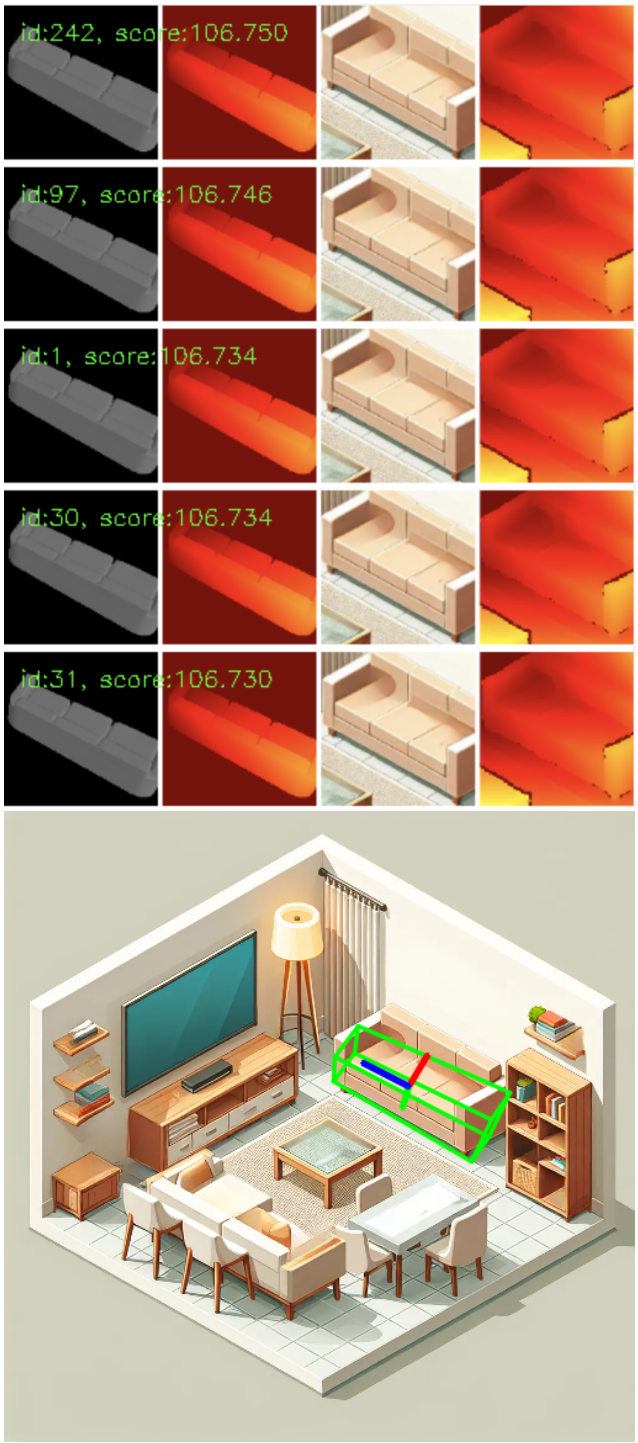
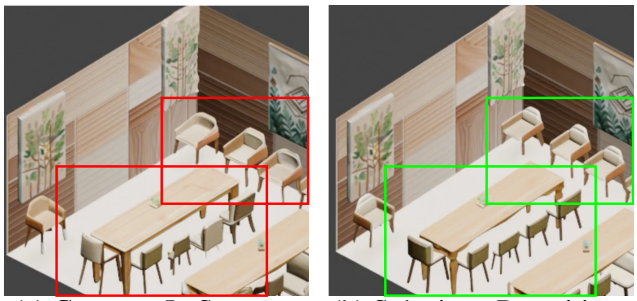


Figure 4. **Top 5 Pose Candidates by FoundationPose [4].** The scores in green texts indicate the model confidence in each candidate pose. As we do not have ground truth camera parameters, and our 3D model is not exactly the same as the 2D counterpart in the image intermediary, even the top candidates are unsatisfactory. The green box with axes in the bottom figure is a visualization of the estimated pose with the highest score.

common intrinsics, instead, they are isometric. Second, the 3D model generated at hand is not the same one as in the intermediary. As seen in Fig. 4, these errors accumulate and influence the final judgment of the model.

2. Repetition Detection



(a) Generate In Separate (b) Substitute Repetitions

Figure 5. **Effects of Repetition Detection.** We replace the different chairs and armchairs in (a) with only one chair model and one armchair model, and keep the other parameters (pose, dimensions, positions) the same to create (b). The changed parts are highlighted by the rectangles. After applying the same 3D model for all similar objects, the result looks more uniform and realistic.

To speed up the generation and make the final scene more consistent, we devise an optional module that automatically detects repetitions of objects. For public scenes such as classrooms and meeting rooms, it is common to have several pieces of furniture of the same model (e.g. chairs and desks). However, as they may be placed in different poses, we find the textual features more reliable than geometric features in determining if two pieces are of the same model. For a pair of objects o_i, o_j We calculate the cosine similarities of the features of T_i and T_j extracted by SBERT [7]. They are determined to be repetitions if the score is above 0.95. This default value is on the conservative side as we prefer less, correct substitutions over more but wrong ones. However, users could easily adjust it to achieve different level of uniformity. For example for Fig. 5, we threshold at 0.89 to substitute more aggressively. We ask ChatGPT to pick the segmented image with the highest quality, and generate one 3D asset from it. Then for all repetitions we use this same asset for pose estimation and the final placement.

3. More Discussion on Limitation

Our automated pipeline generates 3D scenes in batches, yet scene-specific manual adjustments could further improve the results. For a very small subset of examples shown in this paper, we manually excluded 3D assets from the final scene if its quality is very low, and have manually selected certain Pix2Gestalt inpainted results over the GPT suggested ones. Choosing the hyperparameters more automatically and tailored to individual scenes would further improve the results, which we will leave as a future work.

Our method is slower than retrieval-based methods as we generate each object on the fly. We believe the speed could be improved as better component models emerge. After estimating layout and appearance information for each object from the image intermediary, we could also follow the practice of previous works to do 3D asset retrieval based on these features, instead of generation. We leave this interesting extension as another future work.

References

[1] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf, 2023. 1

[2] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. NeurIPS, 36, 2024. 1

[3] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499, 2023. 1

[4] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation with foundation features. In ECCV, pages 163–182. Springer, 2025. 2

[5] Ege Ozguroglu, Ruoshi Liu, Dídac Surís, Dian Chen, Achal Dave, Pavel Tokmakov, and Carl Vondrick. pix2gestalt: Amodal segmentation by synthesizing wholes. In CVPR, pages 3931–3940. IEEE Computer Society, 2024. 1

[6] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952, 2023. 1

[7] N Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084, 2019. 3

[8] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting. arXiv preprint arXiv:2304.06790, 2023. 1