Supplementary Materials for RoomTex: Texturing Compositional Indoor Scenes via Iterative Inpainting

Qi Wang[1,3⋆], Ruijie Lu[2,3⋆], Xudong Xu[3⋆⋆], Jingbo Wang[3], Michael Yu Wang[1], Bo Dai[3], Gang Zeng[2], and Dan Xu[1]

[1] The Hong Kong University of Science and Technology
[2] National Key Laboratory of General AI, School of IST, Peking University
[3] Shanghai AI Laboratory

A More Implementation Details

We provide additional implementation details in the following subsections. All of our experiments are conducted on 4 NVIDIA A100 GPUs, and it takes about 90 minutes to generate a scene.

A.1 Text Prompt

Our method takes a room text prompt along with a compositional mesh based on a given room layout as input and aims to synthesize a complete 3D room texture enabling free novel view rendering inside. Each text prompt is composed of two parts, the style and the description of all the objects in the scene. During the experiments, we utilize the 'Emauromin style' as our default style and also use other styles including 'Misc Kawaii', 'Anime', 'Game Pokemon', 'Artstyle Impressionist', and so on, which can be found in this website[4]. For better stylization results, we also use the corresponding negative prompt for each style. For example, here is one of the text prompts used to generate a bedroom:
Prompt*: Emauromin style, a bedroom with oil paintings on the wall, a single-size bed, brown cotton pillows, a wooden bedside table, a wooden wardrobe, an empty bookshelf, a white desk, a chair, a square and flat ceiling lamp hanging on the ceiling. finely detailed, purism, computer rendering, minimalism, minimal product design.*
Negative prompt*: blurry, blur, text, watermark, render, 3D, NSFW, nude, CGl, monochrome, B&W. cartoon, painting, smooth, plasticblurry, low-resolution, deep-fried, oversaturated.*
All generated texture of 3D rooms presented in this paper and their corresponding text prompts with one specific style are shown in Fig. 9 and Fig. 14. As for the style prompt and the corresponding negative prompt, please refer to the

[⋆] Equal contribution, work done during the internship at Shanghai AI Laboratory.
[⋆⋆] Corresponding author: xuxudong@pjlab.org.cn
[4] https://stable-diffusion-art.com/sdxl-styles/

aforementioned website for details. During the iterative object texturing, the text prompt of every single object is its text description as well as the style instead of the whole text prompt as shown above.

A.2 Room Geometry Generation

Though the core of our method lies in generating the texture of a 3D room, it is quite straightforward to combine our method with some 3D shape generators instead of merely utilizing datasets like 3D-FRONT [4] designed by professional artists for better convenience and flexibility. Despite the fact that the geometry generated by these methods is not perfect, they can still provide the texturing process with strong geometry priors. Generally, we break down the room geometry generation into two parts: object generation and empty room generation.

3D shape generation. As for the furniture items in the scene, we generate most of our object meshes by leveraging an off-the-shelf text-to-3D object generative model, Shap-E [5]. We cut out object descriptions like *'a wooden bedside table'* or *'a white desk'* from the text prompt above, and then send them to the 3D object generative models to generate the corresponding 3D shapes. Delicate decorations like ceiling lamps and chandeliers are borrowed from Objaverse [3] since we found that current object generators are still incapable of generating such fine-grained decorations. The scarcity of such data in 3D object datasets like ShapeNet [1] makes it hard for a 3D generative model to learn. However, we believe that the quality gap between experts and 3D generators, especially for fine-grained models, will be closed with the rapid development of large-scale 3D generative models.

Empty room generation. A procedural generation process is applied to get an empty room mesh. Based on our observations of indoor scene datasets like 3D-FRONT [4], we provide users with various options for diverse room meshes. Specifically, they can decide whether to include *baseboards*, where to position *doors* and *windows*, which *ceiling style* to choose, and the size of the room. Under the guidance of these choices, an empty room mesh can be generated automatically. For example, the available ceiling styles are illustrated in Fig. 2. Moreover, the generated 3D shapes can also be included in the room according to the provided room layout, and thus a complete room mesh is obtained.

A.3 Panorama Generation Details

Initial panorama generation. We leverage the SDXL 1.0 base and refiner models [6] for image generation, where the sampler is selected as 'Euler a'. The sampling step is 50, and we switch from the base model to the refiner at fraction 0.8, *i.e.*, 40 steps. To generate a panoramic image with better visual fidelity and less distortion, we additionally add '720 degrees panorama photo view of' to the beginning of text prompt. To employ the depth guidance, a depth-based ControlNet model [12] is also applied, and the control weight is 1.5. Besides, we also use SDXL VAE, and the CFG scale is set to 6.5.

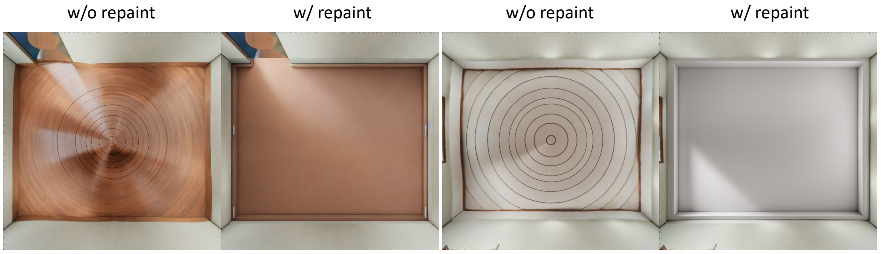


Fig. 1: Ablation study on distortion elimination. Repainting results of floors and ceilings are significantly better than those without repaint.

Empty room refinement. When refining the ceilings and floors of an empty room, we will select an upward view and an overhead view to capture the corresponding areas. The virtual camera is put at the center of the room and towards the center of the ceiling or floor. The focal length and the mask will be adjusted according to the width and height of the room. As shown in Fig. 2, ceilings with star-like or diamond-like decorations and some other styles are all supported in our method.

Panorama distortion elimination. It is noteworthy that except for the object distortion brought by the equirectangular projection, the texture of the ceilings, floor, and baseboards may also suffer from distortion as shown in Fig. 1. This kind of texture is unacceptable and may further influence the iterative object inpainting process since the room context (background texture) is unrealistic. After choosing an overhead and an upward view to repaint floors and ceilings, we can get a textured empty room with less distortion.

Super-resolution and its limitation. Due to the limitation of memory and inference speed, the generated image from the SDXL model has a resolution of $2,048 \times 1,024$. To enrich the texture details, we leverage an off-the-shelf super-resolution method [9] to upscale these panorama images to $4,096 \times 2,048$. Unfortunately, some weird artifacts may appear after using the super-resolution method as illustrated in Fig. 3.

A.4 Iterative Object Texturing Details

Settings of initial perspective view. When re-projecting $\mathbf{I}_p$ to the initial perspective view $\mathbf{v}_0$, the default focal length of the virtual camera is set to 500. However, if the default setting leads to a bad situation where the object occupies less than half of the image or extends beyond the image boundary, we will adjust the camera's focal length accordingly. To be specific, we would gradually increase the focal length until an object occupies half the width of the image while simultaneously guaranteeing it does not exceed the image. The resolution of perspective images is $1,024 \times 1,024$, and these images will also be upscaled to $4,096 \times 4,096$ via super-resolution modules. The setting of SDXL is the same as that for initial panorama generation.

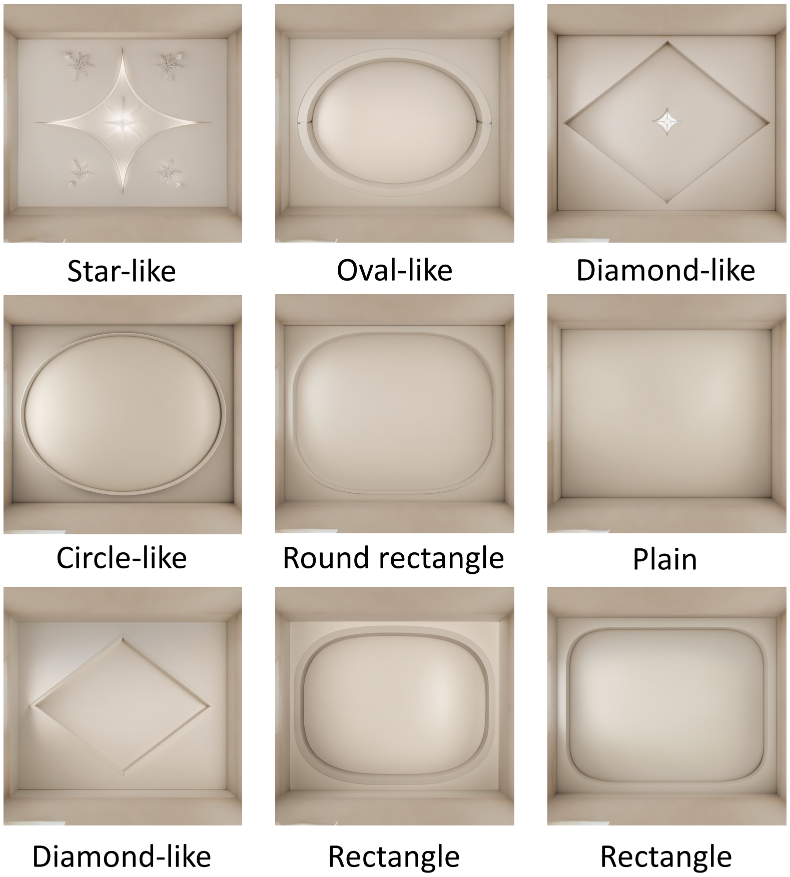


Fig. 2: Different ceiling styles. We show different designs of ceiling styles with an upward view.

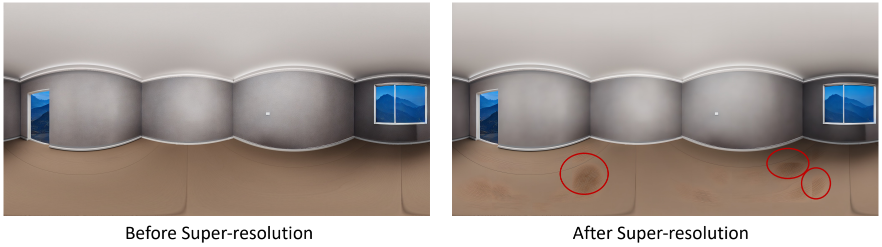


Fig. 3: Limitation of super-resolution modules. Some weird artifacts as circled out may appear as shown in the image at the bottom.

View selection. We divide the views used for iterative object texturing into two groups: basic views and additional views. First, eight basic views are selected

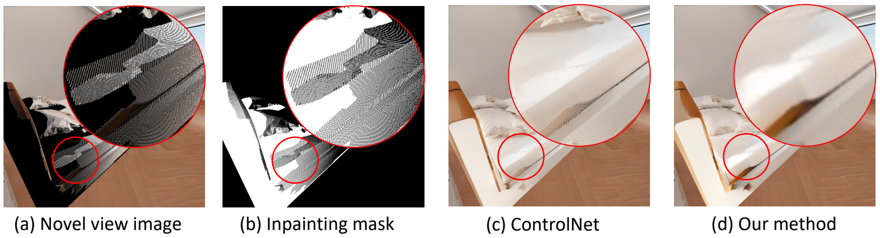


Fig. 4: Ablation study on inpainting methods. We show comparison results of inpainting using ControlNet only and our method. (a) is the rendering image from a novel view and (b) is the inpainting mask (white area). (c) shows the inpainting result using ControlNet only and (d) shows the inpainting result from our method. We can observe clear messy areas in (c) since the diffusion-based inpainting model is insensitive to sparse masks.

and all of them target at the center of the object. These cameras are roughly located in eight corners of the bounding box covering the whole 3D object. For the additional views, different strategies will be applied according to the length-width ratio of the object. If this ratio is less than 1.5, eight additional cameras will be used and still target the center of the object. Their positions are located on a sphere centered around the object with a radius of 0.7 times the diagonal length of the object bounding box, the elevation angle is set to be a random value between $\pi/6$ and $\pi/3$, and the azimuth angles are set to 0, $\pi/2$, π and $3\pi/2$, respectively. Besides, if the aspect ratio is larger than 1.5, we will select 2 groups of eight additional cameras, *i.e.*, 16 cameras in total. In particular, each group of cameras will be also located on a sphere but centered at one-third of the length of this 3D object, ensuring all the objects can be completely viewed with such 16 cameras. It's noteworthy that virtual cameras will be strictly placed within the room boundary, and cameras that are too close to the object will be deleted too.

Mask of untextured area. As we warp our images to a novel view, those originally occluded parts may be observed due to the sparsity of point clouds. For example, the front-side texture of a wardrobe may appear when we inpaint its back side. To eliminate such unreasonable pixels, we identify these areas where the depth is larger than the ground-truth depth and then remove these pixels thereby.

Inpainting strategy in sparse mask area. We use an interpolation-based method to inpaint areas with relatively sparse masks. Specifically, the interpolation-based method means Telea's inpaint algorithm in OpenCV. A comparison with using the diffusion model to inpaint these kinds of areas is shown in Fig. 4. It can be seen that ControlNet does not perform well in sparse areas while our method can generate consistent and natural results.

Selecting satisfying images. It is known that images generated by diffusion models exhibit a high degree of diversity, which makes it necessary to select one satisfying image from multiple generation candidates. While selecting the initial perspective view, we already have the text prompt $\mathbf{T}$ and the warped image $\mathbf{I}_{\text{ref}}$ from $\mathbf{I}_p$. To make sure the generated image aligns with $\mathbf{T}$ well and is similar to $\mathbf{I}_{\text{ref}}$, we compute SSIM Score [11] and CLIP Score [7] among the candidate images and select the one with the highest score:

$$\mathbf{I}_{\text{obj}} = \arg\max_j(\mathbf{S}(\mathbf{I}^j_{\text{obj}}, \mathbf{I}_{\text{ref}}) + \mathbf{C}(\mathbf{I}^j_{\text{obj}}, \mathbf{T})) \tag{1}$$

where $\mathbf{S}(\cdot)$ is the function to calculate SSIM Score, $\mathbf{C}(\cdot)$ stands for the function to calculate CLIP Score and $\left\{\mathbf{I}^j_{\text{obj}}\right\}^5_{j=0}$ represent 5 candidate images used here.

During the iterative object texturing, we notice that the inpainted areas can't strictly align with other regions on the perspective images, leading to inconsistent styles and weird patterns. Hence, we will dilate the inpainting mask and leverage the dilation areas to judge the style consistency. Specifically, we additionally compute a PSNR score in the dilated area to encourage good alignment following:

$$\mathbf{I}_{\text{obj, i}} = \arg\max_j(\mathbf{P}(\mathbf{I}^j_{\text{obj, i}}, \mathbf{I}_{\text{ref, i}}) + \mathbf{C}(\mathbf{I}^j_{\text{obj, i}}, \mathbf{T})) \tag{2}$$

where $\mathbf{P}(\cdot)$ represents the function to calculate the PSNR score, $\mathbf{I}_{\text{ref, i}}$ is the image to be inpainted under view $\mathbf{v}_i$, and the CLIP score is also used to select the most suitable inpainting results from 5 candidates.

A.5 Fine-grained Texture Control

Since our method aims to generate harmonious texture across the whole scene, it is natural for our method to ignore some semantics in the object-level text if they break the overall consistency significantly(like a blue stool among a bunch of brown furniture in the last example of Fig. 9). This misalignment is mainly due to the SDXL model, which is trained on real-world scene images with globally consistent textures. However, it is easy for our method to align with all the object-level prompts by sacrificing some extent of harmoniousness. It is up to the users themselves to decide how they would like the room texture. This fine-grained control can be achieved by simply ignoring the reference textures of these objects in the panorama during the object texturing process. Such a compromised result is shown in the teaser image as well as in Fig. 5. Apart from aligning object textures perfectly with text prompts, other fine-grained texture controls including controlling the texture of floors, walls, ceilings, and objects using scribbles are also integrated into one scene in the demo video. Moreover, as for curvy surfaces, additional conditions like normal maps can be added to further improve the texture quality. A comparison on whether adding such additional controls or not is shown in Fig. 6.

B User Study

We leverage a flask-based web application for the user study to compare our method with baselines from the human perspective. Fig. 10 shows the interface

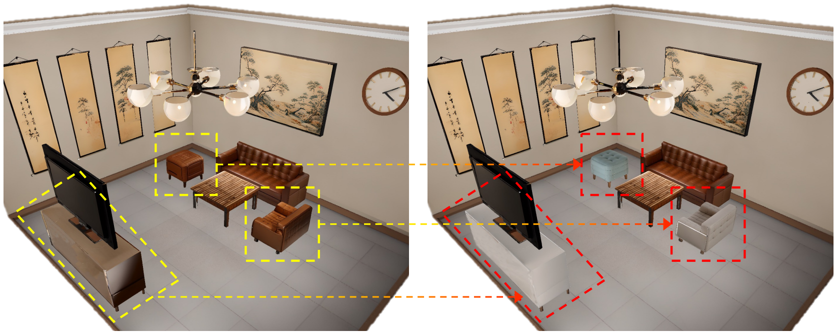


Fig. 5: Object-level alignment results. We show an object-level alignment result in the living room by simply ignoring the reference texture from the initial panorama. Every object in this scene aligns with its corresponding text prompt.

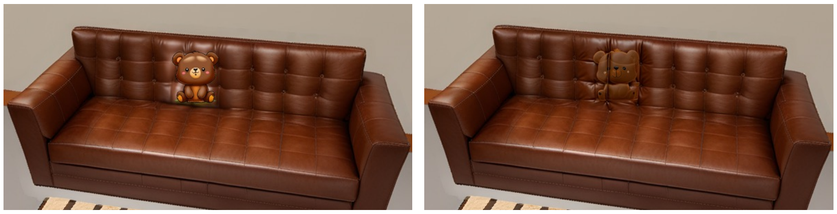


Fig. 6: Additional conditions on curvy surfaces. On the left we show a repainted result without the normal map condition and the bear seems floating. On the right we additionally add the normal map condition to improve the visual quality.

of our questionnaire, where the text description is put on the top, the room overview(a top view of the room) and two random perspective images are in the middle, and a video showing free roaming in the room is also provided below. In the questionnaires, we have 6 groups of scenes in total, where 3 results from baselines and 1 from ours are included in each group. We invite 61 volunteers to conduct the user study and each participant will be randomly shown 2 groups of scenes, *i.e.*, 8 generated scenes, and be asked to judge each presented scene from three different dimensions, 3D consistency(3DC), texture quality(TQ), and perceptual quality(PQ). Specifically, they have to give a score ranging from 1 to 5 for such three aspects. The higher, the better. In the end, we gather 488 responses from the 61 participants and calculate the overall preferences as shown in Tab. 1.

C Detailed Comparison with SceneTex

As the most closely related and concurrent work as ours, SceneTex [2] formulates the whole texturing process as an optimization problem by using a multi-resolution texture field and the VSD objective [10]. Though being able to generate compelling textures for a given room geometry, the optimization-based

Method	3DC(↑)	TQ(↑)	PQ(↑)
TEXTure-C [8]	3.06(±0.85)	2.75(±0.80)	2.88(±0.77)
TEXTure-H [8]	2.83(±0.86)	2.63(±0.84)	2.64(±0.83)
SceneTex [2]	3.98(±0.86)	3.73(±1.02)	3.56(±0.95)
Ours	**4.51**(±0.71)	**4.29**(±0.81)	**4.26**(±0.76)

Table 1: Quantitative comparison of the user study. Mean opinion scores are in the range of 1 ∼ 5. Our method outperforms TEXTure-C and TEXTure-H by a large margin.

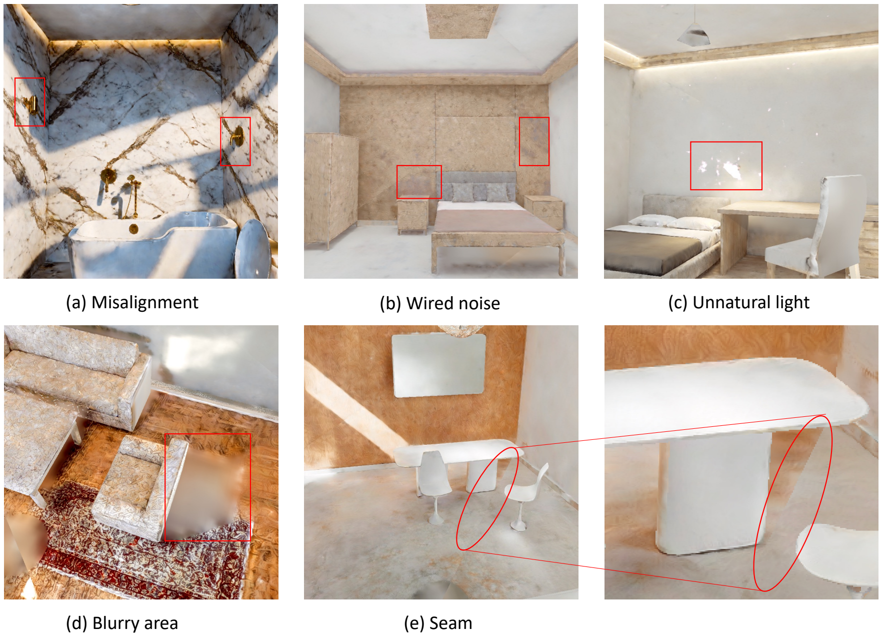


Fig. 7: Limitations of optimization-based framework. We show several obvious artifacts of the optimization-based framework including misalignment between geometry and texture, the existence of unnatural light and noise, and evident blurry areas and seams.

framework may suffer some underlying problems. First of all, the texture generated via optimization may not align well with the given geometry as the texture prior distilled from text-to-image diffusion models tends to make images look as realistic as possible from certain viewpoints. For example, as shown in Fig. 7 (a), there should not exist handle-like objects on the walls of the bathroom since there are no handles at all in the given meshes. Similarly, SceneTex is prone to generate indoor textures containing unnatural lights and weird noises as shown in Fig. 7 (b) and (c). Moreover, the choice of viewpoints leads to some blurry areas due to the severe occlusion problem in the indoor scene as shown in Fig. 7

(d). However, we believe the blurry problem may be mitigated via a more delicate viewpoint selection strategy. Some clear seams can be observed in Fig. 7 (e) due to the usage of UV map. On the other hand, textures of objects with complex topological structures may easily be affected by accumulative errors under an explicit inpainting-based framework, even though we have designed a module to detect the misalignment between depth space and rgb space. Optimization-based methods naturally possess some extent of continuity and will not be significantly impacted by a particular viewpoint. But objects with complex topological structures like multi-layer lamps still pose a challenge for both approaches due to the severe self-occlusion. In the future, we believe a well-designed strategy could marry the merits of the inpainting-based method and the optimization-based method for more harmonious and consistent texture generation.

D Additional Results

More qualitative results including a kitchen, a bedroom, and a living-dining room compared with baseline methods are shown in Fig. 11 and more stylized room results are shown in Fig. 12. Though it is more flexible to assemble a room using 3D shape generators along with our provided empty room generator by users themselves, our method is also capable of texturing a room from professional datasets like 3D-FRONT [4]. We choose five rooms including a bedroom, three living rooms, and a living-dining room from the dataset, and the results of the overhead view and several perspective views from inside are shown in Fig. 13. The overview images of these rooms as well as their corresponding text prompts are shown in Fig. 14. Moreover, as a generative approach, the results of our method are of high diversity and an example of the diverse panoramic images is shown in Fig. 8. We render some room tour videos of different scenes with different styles, which are integrated into a unified video put in the supplementary. Besides, we also present a demo video to demonstrate the effectiveness of using our misalignment detection technique. Another demo video shows how our method supports interactive fine-grained texture controls as well as a room tour video in the new room after applying these controls.

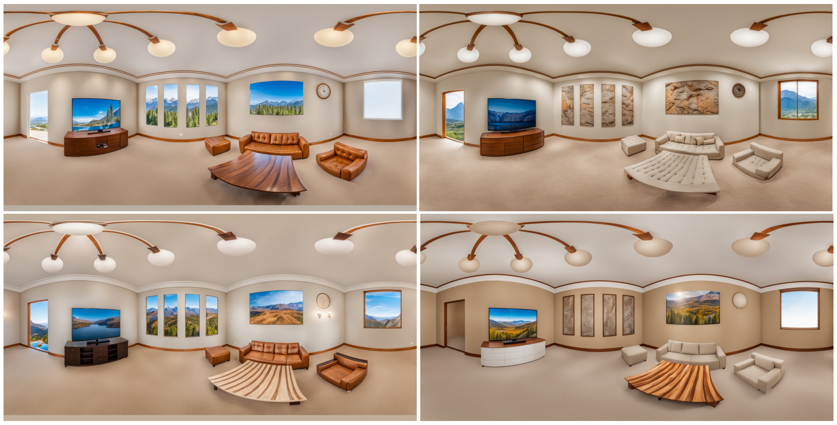
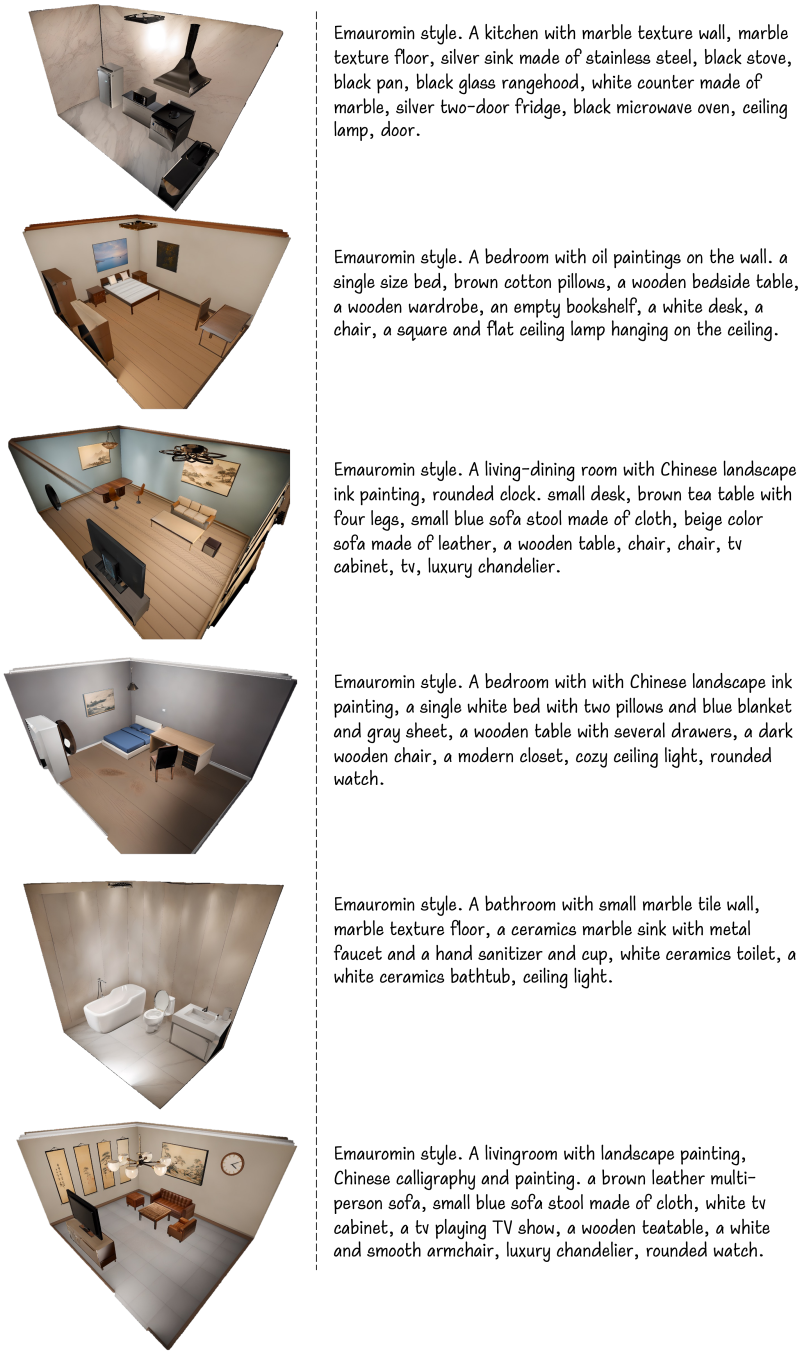


Fig. 8: High diversity of panoramic images. We use the same text prompt as the living-room example in the main paper.



Emauromin style. A kitchen with marble texture wall, marble texture floor, silver sink made of stainless steel, black stove, black pan, black glass rangehood, white counter made of marble, silver two-door fridge, black microwave oven, ceiling lamp, door.

Emauromin style. A bedroom with oil paintings on the wall. a single size bed, brown cotton pillows, a wooden bedside table, a wooden wardrobe, an empty bookshelf, a white desk, a chair, a square and flat ceiling lamp hanging on the ceiling.

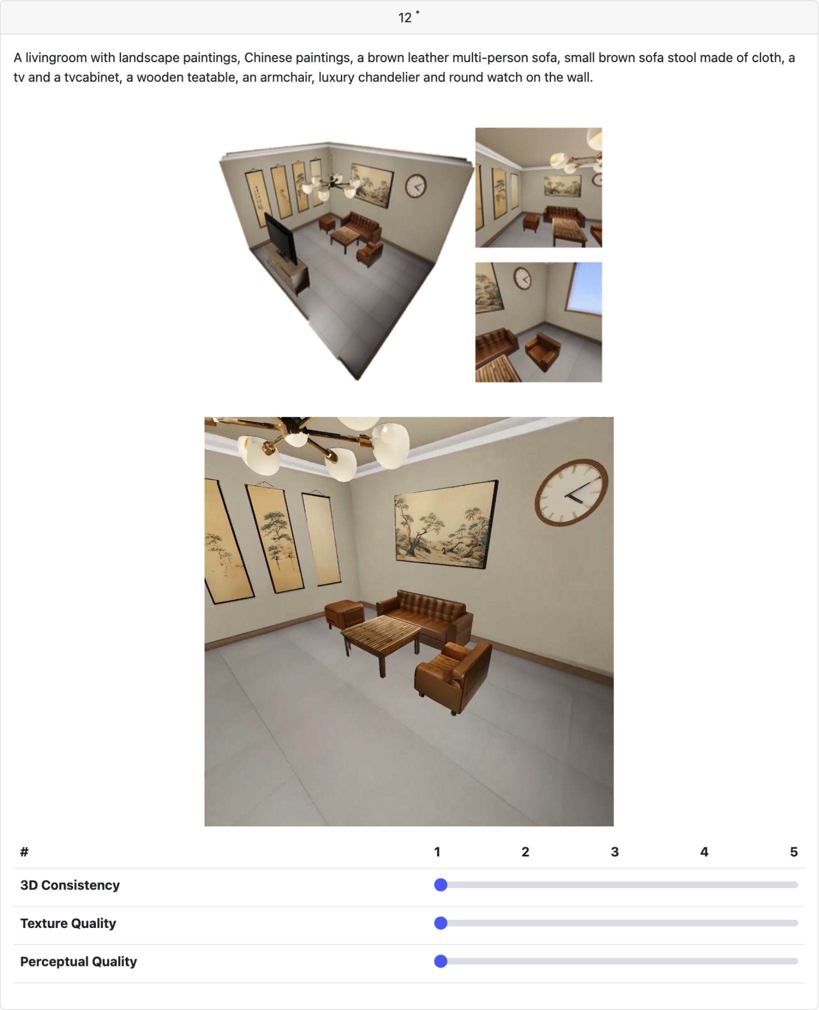
Emauromin style. A living-dining room with Chinese landscape ink painting, rounded clock. small desk, brown tea table with four legs, small blue sofa stool made of cloth, beige color sofa made of leather, a wooden table, chair, chair, tv cabinet, tv, luxury chandelier.

Emauromin style. A bedroom with with Chinese landscape ink painting, a single white bed with two pillows and blue blanket and gray sheet, a wooden table with several drawers, a dark wooden chair, a modern closet, cozy ceiling light, rounded watch.

Emauromin style. A bathroom with small marble tile wall, marble texture floor, a ceramics marble sink with metal faucet and a hand sanitizer and cup, white ceramics toilet, a white ceramics bathtub, ceiling light.

Emauromin style. A livingroom with landscape painting, Chinese calligraphy and painting. a brown leather multi-person sofa, small blue sofa stool made of cloth, white tv cabinet, a tv playing TV show, a wooden teatable, a white and smooth armchair, luxury chandelier, rounded watch.

Fig. 9: Generated rooms and their corresponding text prompts. 6 compositional rooms with default style are shown with an overview image on the left and their corresponding text prompts on the right.



12 *

A livingroom with landscape paintings, Chinese paintings, a brown leather multi-person sofa, small brown sofa stool made of cloth, a tv and a tvcabinet, a wooden teatable, an armchair, luxury chandelier and round watch on the wall.

#	1	2	3	4	5
3D Consistency	●				
Texture Quality	●				
Perceptual Quality	●				

Fig. 10: The interface of our questionnaire used in the user study. The text prompt is shown on the top, the room overview and two randomly selected perspective images are in the middle, and a room tour video is put at the bottom.

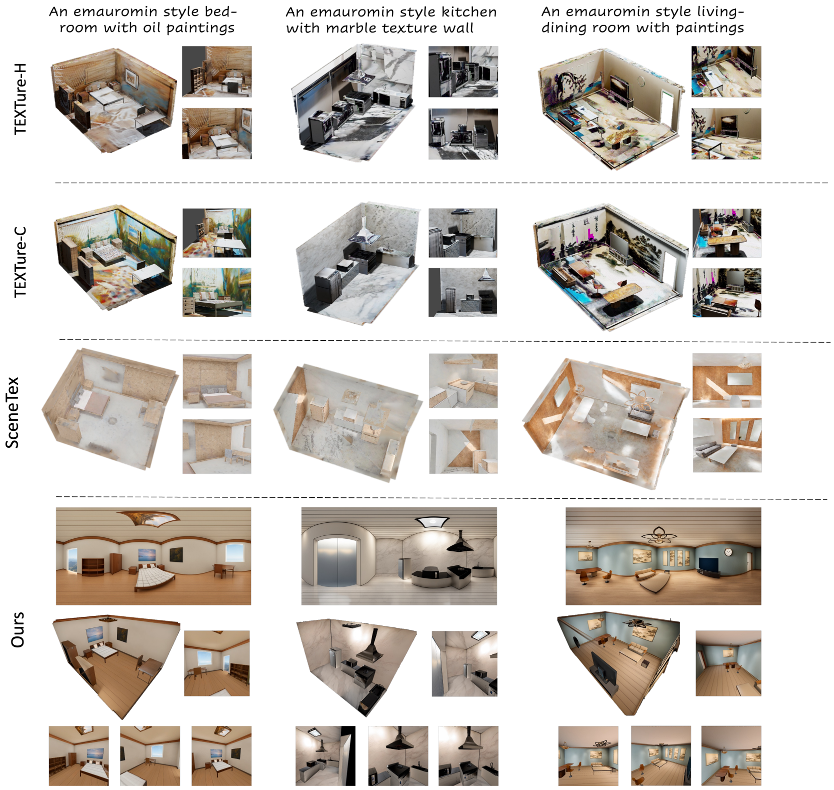
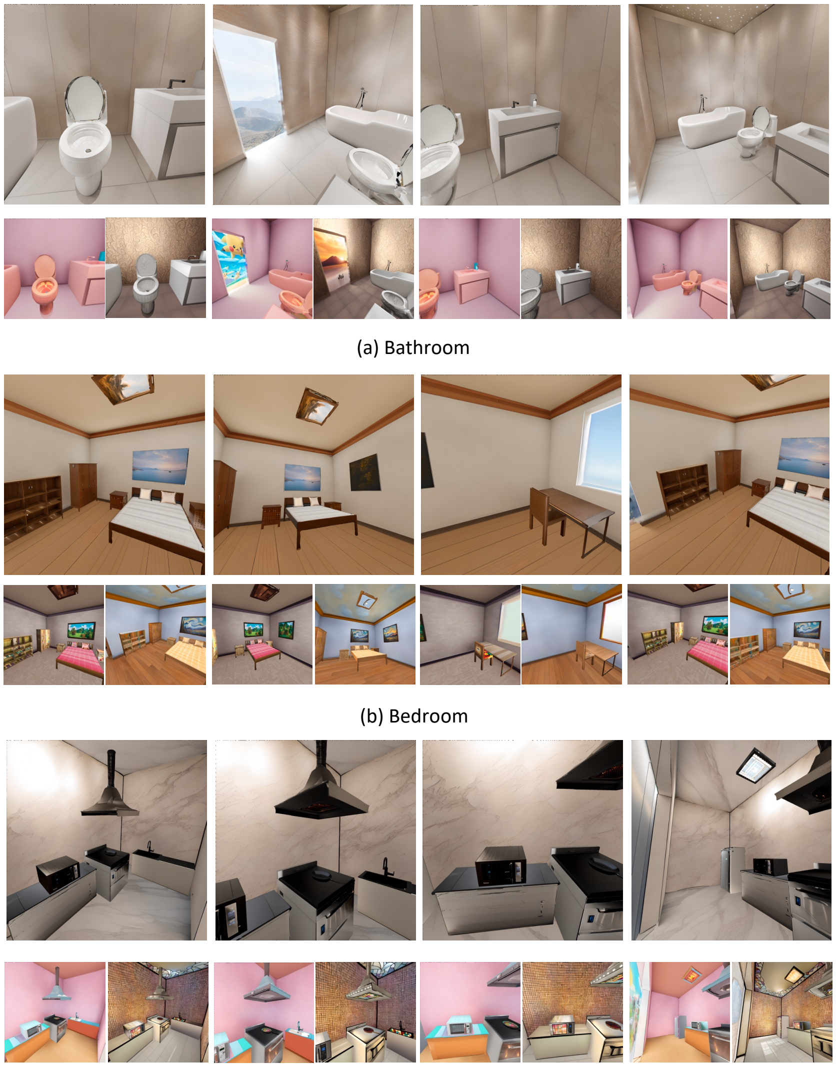


Fig. 11: More qualitative comparison. We show more qualitative results compared with baselines.



(a) Bathroom

(b) Bedroom

(c) Kitchen

Fig. 12: Results of stylized rooms. We show some stylized rooms with several rendered perspective images from several perspective views.

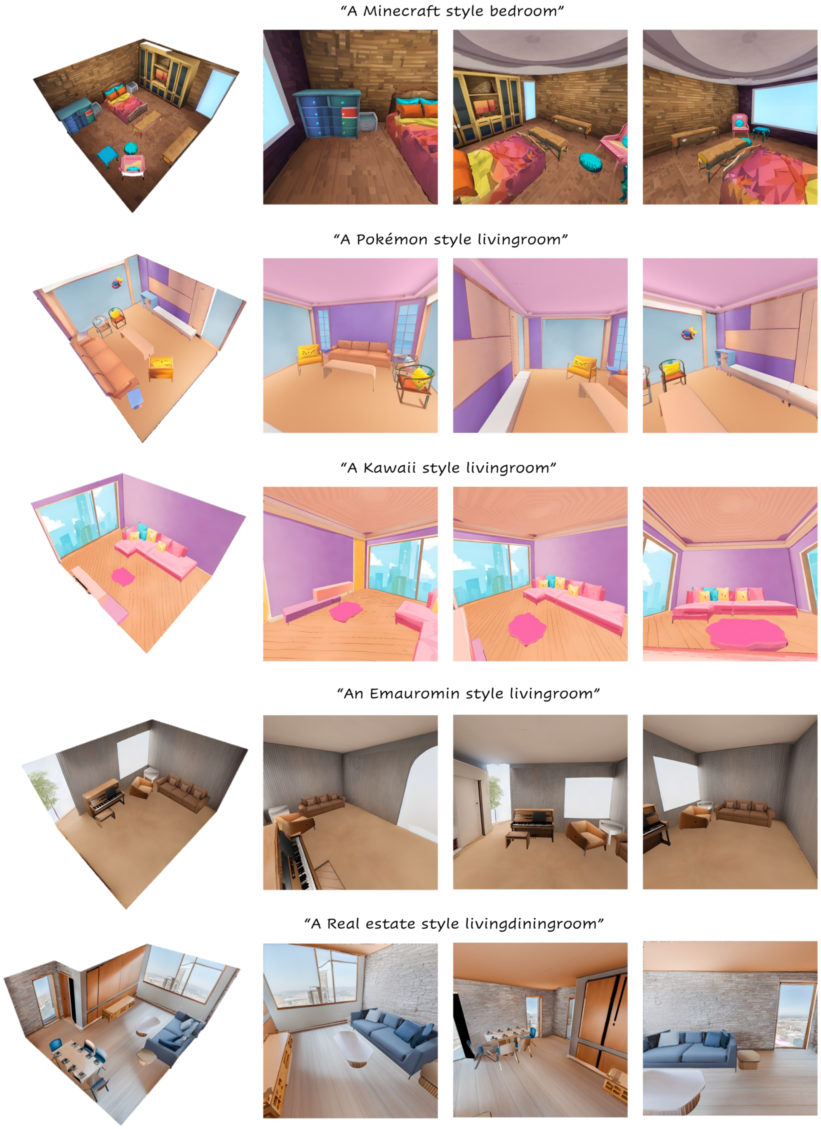

"A Minecraft style bedroom"
"A Pokémon style livingroom"
"A Kawaii style livingroom"
"An Emauromin style livingroom"
"A Real estate style livingdiningroom"

Fig. 13: Our scene texturing results on 3D-FRONT dataset We show our texturing results on the 3D-FRONT dataset with an overhead view on the left and three perspective views from inside on the right. The text prompt here is concise, please refer to Fig. 14 for details.

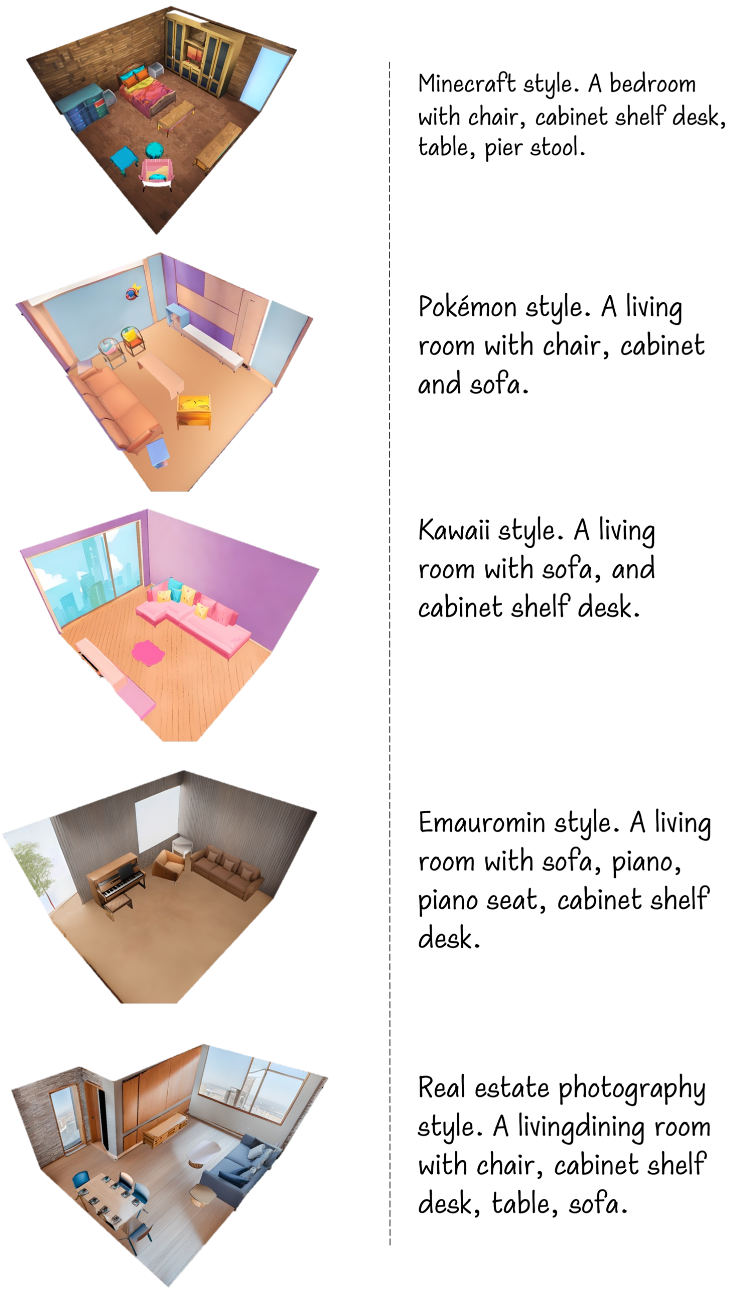

Minecraft style. A bedroom with chair, cabinet shelf desk, table, pier stool.
Pokémon style. A living room with chair, cabinet and sofa.
Kawaii style. A living room with sofa, and cabinet shelf desk.
Emauromin style. A living room with sofa, piano, piano seat, cabinet shelf desk.
Real estate photography style. A livingdining room with chair, cabinet shelf desk, table, sofa.

Fig. 14: Our scene texturing results and corresponding text prompts on the 3D-FRONT dataset. 5 3D-FRONT rooms with different styles are shown with an overview image on the left, and their corresponding text prompts on the right.

References

1. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
2. Chen, D.Z., Li, H., Lee, H.Y., Tulyakov, S., Nießner, M.: Scenetex: High-quality texture synthesis for indoor scenes via diffusion priors. In: CVPR. pp. 21081–21091 (2024)
3. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153 (2023)
4. 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. pp. 10933–10942 (2021)
5. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463 (2023)
6. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
7. 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. pp. 8748–8763. PMLR (2021)
8. Richardson, E., Metzer, G., Alaluf, Y., Giryes, R., Cohen-Or, D.: Texture: Text-guided texturing of 3d shapes. In: ACM SIGGRAPH 2023 conference proceedings. pp. 1–11 (2023)
9. Wang, X., Xie, L., Dong, C., Shan, Y.: Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1905–1914 (2021)
10. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. Advances in Neural Information Processing Systems **36** (2024)
11. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
12. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)