

# L-MAGIC: Language Model Assisted Generation of Images with Coherence

Zhipeng Cai\* Matthias Mueller Reiner Birkl Diana Wofk Shao-Yen Tseng  
 Junda Cheng Gabriela Ben-Melech Stan Vasudev Lai Michael Paulitsch  
 Intel Labs

## Abstract

In the current era of generative AI breakthroughs, generating panoramic scenes from a single input image remains a key challenge. Most existing methods use diffusion-based iterative or simultaneous multi-view inpainting. However, the lack of global scene layout priors leads to subpar outputs with duplicated objects (e.g., multiple beds in a bedroom) or requires time-consuming human text inputs for each view. We propose L-MAGIC, a novel method leveraging large language models for guidance while diffusing multiple coherent views of 360° panoramic scenes. L-MAGIC harnesses pre-trained diffusion and language models without fine-tuning, ensuring zero-shot performance. The output quality is further enhanced by super-resolution and multi-view fusion techniques. Extensive experiments demonstrate that the resulting panoramic scenes feature better scene layouts and perspective view rendering quality compared to related works, with >70% preference in human evaluations. Combined with conditional diffusion models, L-MAGIC can accept various input modalities, including but not limited to text, depth maps, sketches, and colored scripts. Applying depth estimation further enables 3D point cloud generation and dynamic scene exploration with fluid camera motion. Code is available at <https://github.com/ZhipengCai/L-MAGIC-code-release>.

## 1. Introduction

Diffusion models have achieved state-of-the-art performance in image generation. However, generating a 360° panoramic scene from a single perspective image remains a challenge, which is an important problem in many computer vision applications, such as architecture design, movie scene creation, and virtual reality (VR).

Training a model to directly generate panoramic images is challenging due to the lack of diverse large-scale datasets. Hence, most existing works separate panoramic scenes into multiple perspective views, and inpaint them us-

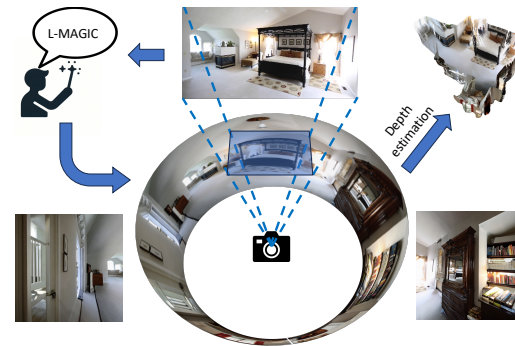


Figure 1. **Teaser.** L-MAGIC is a novel method to generate a 360° panoramic scene from a single input image. L-MAGIC utilizes large language models to control perspective diffusion models to generate multiple views with coherent 360° layout. L-MAGIC is also compatible with images synthesized by conditional generative models, making it capable of creating panoramic scenes from various input modalities. A set of perspective images rather than a single panoramic image also allows our method to leverage off-the-shelf monocular depth estimation models to enable immersive experiences, e.g., scene fly-through or 3D point cloud generation.

ing pre-trained diffusion models. To ensure generalization, the diffusion model is either frozen without any architecture change [11] or combined with extra modules trained on small datasets for integrating multi-view information [25].

A common approach to encode the scene information during multi-view inpainting is to provide a text-conditioned diffusion model with the description of the input image, which is generated by a user or an image captioning model [13]. Though effective for extending local scene content, such approaches suffer from *incoherence* in the overall scene layout. Specifically, using the same text description for diffusing different views along the 360° panorama leads to artifacts and unnecessarily repeated objects. Current inpainting methods have no mechanism to leverage global scene information in individual views.

In this work, we show that state-of-the-art (vision) language models, without fine-tuning, can be used to control multi-view diffusion and effectively address the above problem. We propose L-MAGIC (Fig. 1), a novel framework

\*Corresponding author (zhipeng.cai@intel.com)

leveraging large language models to enable the automatic generation of diverse yet coherent 360° views from a given input image. L-MAGIC relies on iterative warping-and-inpainting. Pre-trained (vision) language models are used to: (1) generate layout descriptions of different views that are used in text-conditioned inpainting, (2) automatically determine whether salient objects should be repeated or not for a specific scene, and (3) monitor the inpainting outputs to avoid challenging cases where diffusion models violate the text guidance. A key contribution is the prompt design for language and diffusion models to make L-MAGIC *fully automatic*. In addition, smooth multi-view fusion and super-resolution techniques are used to ensure high resolution and quality when producing the final panorama.

Experiments show that L-MAGIC generates 360° panoramic scenes with higher quality and more coherent layouts compared to state-of-the-art methods. Not relying on model fine-tuning makes L-MAGIC effective on in-the-wild images, and extendable, using conditional diffusion models [19, 29], to other types of inputs such as text, depth maps, sketch drawings and color scripts/segmentation masks. Applying depth estimation further enables the creation of 3D point clouds and immersive scene fly-through experiences with both camera rotation and translation.

## 2. Related Work

**Diffusion models.** Diffusion models learn to generate data by inverting the diffusion process, *i.e.*, removing the noise in the data (see [26] for a detailed survey). By separating the data generation process into multiple noise removal steps (reverse process) [10], diffusion models learn image synthesis much more effectively than GANs [9]. Recently, latent diffusion models [19] have been proposed to improve the training speed and image synthesis quality by performing the diffusion in latent space. By training the model on large-scale image-caption pairs [21], they achieve remarkable quality and robustness in text-conditioned image synthesis. Further fine-tuning of latent diffusion models using large mask strategies [24] produces robust text-conditioned inpainting models, which are used in this work.

**Panoramic scene generation.** Various approaches have been proposed for panoramic scene generation. Some of them [8, 22] treat the panorama as a single equirectangular image, and generate it in a single forward pass. However, such approaches struggle to close the loop at both ends of the generated equirectangular image, even with purposely designed spherical positional embeddings [8]. Meanwhile, the lack of large scale training data makes it impractical to train a generalizable model for image-to-panorama, and limits the robustness of the model, resulting in inconsistent outputs with the input descriptions.

Some recent methods [11, 25] create panoramas by gen-

erating multiple perspective views using robust pre-trained diffusion models trained on large-scale perspective data. Text2room [11] generates 10 rotated views of a panorama using stable diffusion v2 inpainting without layout control and focuses on indoor scenes and mesh generation. MVDiffusion [25] ensures multi-view local texture consistency by extra multi-view attention modules fine-tuned on small datasets. Though applicable to both text-to-panorama and image-to-panorama, these methods struggle to generate diverse 360° views. Specifically, there is no mechanism to encode the global scene layout into the generation or inpainting of different views. Hence, conditioning the method only on the input image or text results in salient objects (*e.g.*, beds in a bedroom) being generated repeatedly across views. In this work, we guide the multi-view diffusion process with large language models to automatically generate panoramic scenes with coherent and diverse 360° layouts.

**Language models.** Recent language model advancements have enabled many important applications (see [30] for details). Trained on large-scale data with humans in the loop, ChatGPT [4] has demonstrated super-human performance on various language-based tasks. In this work, we utilize ChatGPT to automatically generate coherent multi-view scene descriptions for the consumption of a pre-trained diffusion model that generates multiple perspective views of a panoramic scene. Leveraging multi-modal data, vision language models [28] have further enhanced language models to understand visual inputs. In this work, we utilize pre-trained VQA models [13] to automatically generate a scene description for the input image, and to avoid unnecessarily repeated objects across the generated panoramic scene.

## 3. Methodology

The goal of this work is to generate a coherent 360° panoramic scene given a single (perspective) image. Note that this setting is very general since the input image could be either captured in the real world or synthesized. For example using conditional diffusion models [19, 29], one can synthesize the input image using inputs such as text descriptions, sketches, depth images, and so on.

As shown in Fig. 2, L-MAGIC generates the panoramic scene with iterative warping-and-inpainting. The *warping* step generates an incomplete perspective view and the mask of the missing region (Sec. 3.1). The *inpainting* step completes the masked region with assistance from language models (Sec. 3.2). The final panorama is created by fusing the generated views with some post-processing to enhance the quality and resolution (Sec. 3.3).

### 3.1. Warping

At each warping step, we project all completed views onto a unit sphere representing the panoramic scene, and then ren-

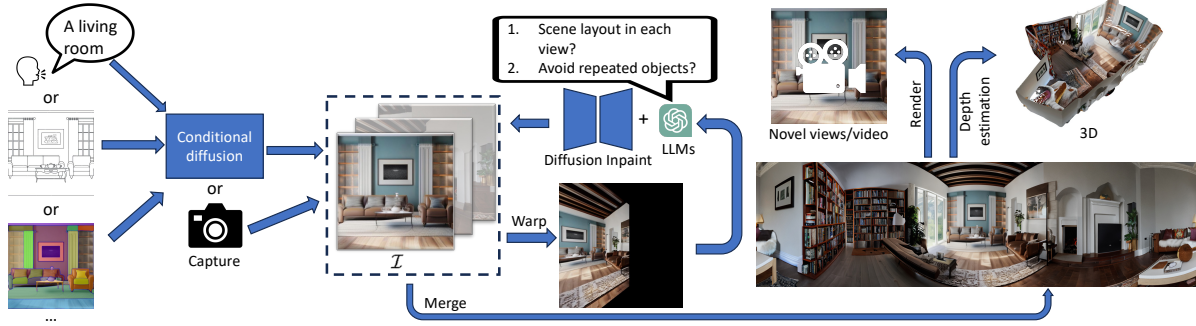


Figure 2. **L-MAGIC pipeline.** The input is an image  $\mathcal{I}$  either captured in the real-world or synthesized, *e.g.*, by conditional diffusion models. Multiple novel views to compose a 360° panoramic scene are generated by iterative warping and inpainting. Pre-trained diffusion models assisted by pre-trained language models are used to generate views with both high-quality local textures and coherent 360° layouts. Further quality enhancement techniques ensure smooth blending of multiple views into high-resolution panoramic scenes. L-MAGIC can generate panorama images, immersive videos, and 3D point clouds from various types of inputs, such as images, text, and sketch drawings.

der the next incomplete view to inpaint based on the relative camera pose. To project an image to the unit sphere, we first construct a mesh by defining the vertices  $\mathcal{V}$  on each image pixel and creating edges  $\mathcal{E}$  between adjacent pixels. Then, we project the vertices to a unit sphere by

$$\mathbf{v}_{\text{sp}} = \frac{\mathbf{K}^{-1}\mathbf{v}}{\|\mathbf{K}^{-1}\mathbf{v}\|}, \quad (1)$$

where  $\mathbf{K}$  is the intrinsic matrix,  $\mathbf{v} \in \mathcal{V}$  is the homogeneous coordinate [23] of a pixel, and  $\mathbf{v}_{\text{sp}}$  is the projected location. To warp a completed view  $A$  to a novel view  $B$ , where  $\mathbf{R}$  is the rotation matrix from  $A$  to  $B$ , we rotate each projected vertex  $\mathbf{v}_{\text{sp}}$  of  $A$  by  $\mathbf{v}_{\text{rot}} = \mathbf{R}\mathbf{v}_{\text{sp}}$ , and then perform rasterization-based rendering [18]  $(\mathcal{I}, \mathcal{M}) = \text{rasterize}(\mathcal{V}_{\text{rot}}, \mathcal{E}, \mathbf{K}')$  where  $\mathcal{V}_{\text{rot}}$  is the set of rotated vertices and  $\mathbf{K}'$  is the intrinsic matrix of image  $B$ . The output  $\mathcal{I}$  is a warped image and  $\mathcal{M}$  is a binary mask indicating whether inpainting is required for a pixel (obtained by checking for each pixel whether ray-casting hits a valid mesh face).

To ensure the local inpainting consistency of each perspective view, we use a large field of view (FoV) and adjust the rotation angles so that both known and unknown regions are reasonably large after warping. In practice, a FoV of 100 degrees with roughly 40 degrees of rotation between adjacent views works well. To further reduce the iterative error accumulation, we expand the scene alternatively from both sides of the input image, rather than expanding in a single direction. To ensure a smooth 360° loop closure, we tune the rotation angles so that the final view has a large incomplete region at the center, resulting in a sequence of views with rotation angles of  $\{0^\circ, 41^\circ, -41^\circ, 82^\circ, -82^\circ, 123^\circ, 200.5^\circ\}$  (for loop closure).

### 3.2. Inpainting with Language Model Assistance

The inpainting step completes a warped view with a consistent local style and a coherent 360° scene layout. We

utilize the Stable Diffusion v2 inpainting model [19], which can effectively extrapolate the large missing region of each warped view while maintaining local style consistency. However, naive inpainting without any prior will generate severe artifacts. One common prior explored before [11, 25] is a user-provided text description of the scene or input image. Yet, using the same description in different views may generate duplicate objects such as multiple beds in a bedroom (see Fig. 4), since perspective inpainting methods have no mechanism to split the layout into different views. Please refer to Sec. 4.3 for a detailed ablation study.

To address these problems, we use a vision language model  $\mathcal{L}_v(\cdot)$  (BLIP-2 [13]) and a language model  $\mathcal{L}(\cdot)$  (ChatGPT4 [4]) to guide the inpainting process. In the following, we describe our method (Alg. 1) in detail. The exact prompts used in Alg. 1 to interact with language and diffusion models are provided in Appendix A.

Before warping and inpainting, we first prompt  $\mathcal{L}_v(\cdot)$  to generate the description  $d_{\mathcal{I}}$  for the input image  $\mathcal{I}$  (line 2). We ask two questions so that  $d_{\mathcal{I}}$  contains both coarse and fine levels of detail. Next, we ask  $\mathcal{L}(\cdot)$  to imagine the global scene layout  $d_{360}$  (line 3) based on  $d_{\mathcal{I}}$ , where each line of  $d_{360}$  corresponds to the description of a specific view. To avoid duplicate objects, we request a compact description of individual views without mentioning objects in other views.

$d_{360}$  mostly contains objects of individual views. Using such descriptions as the inpainting prompt can lead to inconsistent style at distant views. Hence, we ask  $\mathcal{L}(\cdot)$  to remove objects from  $d_{\mathcal{I}}$  and obtain the final scene-level description  $d_{\text{scene}}$ , *e.g.*, ‘a bedroom with a wooden bed’ becomes ‘a bedroom’ (line 4).  $d_{\text{scene}}$  is later used together with  $d_{360}$  to ensure a consistent multi-view style.

Though  $d_{\text{scene}}$  ensures the multi-view style consistency, the training data bias of diffusion models may still result in objects commonly associated with a particular scene being generated, even if not explicitly mentioned in  $d_{360}$ . For ex-

---

**Algorithm 1** L-MAGIC

---

```
1: Input: Initial image  $\mathcal{I}$ , intrinsic matrix  $\mathbf{K}$  of  $\mathcal{I}$ , intrinsic matrix  $\mathbf{K}'$  and poses  $\mathcal{P} = \{\mathbf{R}_i\}_{i=1}^N$  of warped views, vision language model  $\mathcal{L}_v(\cdot)$ , language model  $\mathcal{L}(\cdot)$ , text-conditioned inpainting model  $f_{\text{inpaint}}(\cdot)$ .
2:  $d_{\mathcal{I}} \leftarrow \mathcal{L}_v(\mathcal{I}, \text{'Description of input image'})$ 
3:  $d_{360} \leftarrow \mathcal{L}(d_{\mathcal{I}}, \text{'Layout of individual views'})$ 
4:  $d_{\text{scene}} \leftarrow \mathcal{L}(d_{\mathcal{I}}, \text{'Remove object-level descriptions'})$ 
5:  $d_{\text{repeat}} \leftarrow \mathcal{L}(d_{\mathcal{I}}, \text{'Avoid duplicated objects'})$ 
6:  $\mathcal{W}_1, \mathcal{M}_1 \leftarrow \text{warp}(\mathcal{C}, \mathbf{K}, \mathbf{K}', \mathbf{R}_1)$ 
7:  $\mathcal{I}_1 \leftarrow f_{\text{inpaint}}(\mathcal{W}_1, \mathcal{M}_1, d_{\text{scene}}) \triangleright$  expand the FoV of  $\mathcal{I}$ 
8:  $\mathcal{C} \leftarrow \{\mathcal{I}_1\}$ 
9: for  $i = 2$  to  $N$  do
10:    $c \leftarrow 0$ 
11:    $\mathcal{W}_i, \mathcal{M}_i \leftarrow \text{warp}(\mathcal{C}, \mathbf{K}, \mathbf{K}', \mathbf{R}_i)$ 
12:    $d_i \leftarrow \text{generate\_prompt}(d_{360}, d_{\text{scene}}, d_{\text{repeat}}, i)$ 
13:    $\mathcal{I}_i \leftarrow f_{\text{inpaint}}(\mathcal{W}_i, \mathcal{M}_i, d_i)$ 
14:   for each object  $\mathcal{O}$  in  $d_{\text{repeat}}$  do
15:     if  $\mathcal{L}_v(\mathcal{I}_i, \text{'Is } \mathcal{O} \text{ in } \mathcal{I}_i?') = \text{'yes'} \cap c < 20$  then
16:        $c \leftarrow c + 1$ 
17:       Go to line 13.
18:     end if
19:   end for
20:    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathcal{I}_i\}$ 
21: end for
22: return  $\text{merge}(\mathcal{C})$ 
```

---

ample, a bed is often generated with the word ‘bedroom’ in the prompt, resulting in duplicate beds in multiple views. To avoid this problem, we further let  $\mathcal{L}(\cdot)$  automatically determine whether there are some objects in the scene that require repetition avoidance (line 5).

After each warping step, we use the outputs from lines 2 to 5 to automatically generate the prompt for text-conditioned inpainting (line 12). Specifically, for the warped view with  $0^\circ$  rotation ( $i = 1$ ), we use  $d_{\text{scene}}$  as the prompt ( $d_i$ ) for text-conditioned inpainting (line 7). For other views (line 13), if there is no object in  $d_{\text{repeat}}$ , *i.e.*, no repetition avoidance required, we perform inpainting with the prompt ‘*a peripheral view of  $\langle d_{\text{scene}} \rangle$  where we see  $\langle$ the corresponding description in  $d_{360} \rangle$* ’. If any object exists in  $d_{\text{repeat}}$ , we use the positive prompt of ‘*a peripheral view of  $\langle d_{\text{scene}} \rangle$  where we only see  $\langle$ the corresponding description in  $d_{360} \rangle$* ’, and the negative prompt of ‘*any type of  $\langle$ the object in  $d_{\text{repeat}} \rangle$* ’ (one sentence for each object in  $d_{\text{repeat}}$ ). The positive prompt template prevents Stable Diffusion from generating common objects of an environment (*e.g.*, the bed in a bedroom). The negative prompt template avoids duplication of objects mentioned in  $d_{\text{repeat}}$ .

Bias exists in the training data of diffusion models — an image with the caption of ‘a bedroom’ mostly contains a bed. Therefore, repeated objects can still be generated

even with constraints from the prompt. To further alleviate this problem, we use  $\mathcal{L}_v(\cdot)$  to check whether each inpainted image  $\mathcal{I}_i$  contains objects mentioned in  $d_{\text{repeat}}$  (line 15). If the answer is ‘yes’, we re-run inpainting until the answer becomes ‘no’ or the maximum number of trials  $c$  is reached.

### 3.3. Quality and Resolution Enhancement

Several techniques are also proposed to enhance the quality and resolution of the final panorama.

As shown in Appendix B, adjacent pixels at the center of an image have a larger angular distance than the ones at the side of an image. When warping a completed view to a novel view, the original central region becomes the side region, making the rendered image blurry due to interpolation. Meanwhile, the resolution of the Stable Diffusion output is  $512 \times 512$ , the panorama created by these images has a low resolution. To address both problems, we apply super-resolution [3] to the output  $\mathcal{I}_i$  of each inpainting step, increasing the resolution of  $\mathcal{I}_i$  to  $2048 \times 2048$ . Then, we warp the high-resolution image to a low-resolution novel view so that no (strong) interpolation is required. After performing all warping and inpainting steps, we simply fuse the super-resolution images to generate a high-resolution panorama.

During warping and panorama generation, multiple perspective images might have overlaps at the same region. To avoid sharp boundaries when merging them, we perform a weighted average, *i.e.*, given multiple warped pixels at the same location with colors  $\mathbf{c}_i$ , the final merged pixel color is  $\mathbf{c}_{\text{merge}} = \frac{\sum_i w_i \mathbf{c}_i}{\sum_i w_i}$ , where the weight  $w_i$  is computed as the distance to the nearest image boundary at the original view  $i$ . This strategy effectively down-weights the pixels near the warping boundaries, ensuring a smooth transition during multi-view fusion.

To create the final panorama (line 22), we first project each view to the unit sphere same as in Sec. 3.1. Then, we perform the equirectangular projection [5] to warp multiple projected views to the same equirectangular plane, and merge them into a single equirectangular image.

### 3.4. Discussion

L-MAGIC is *fully automatic* — no human interaction is required to link language models and diffusion models. This is realized by 1) careful prompt engineering, which enables language models to output texts that can be automatically converted into the inpainting prompt, and 2) handling the edge cases where language models or diffusion models do not generate outputs that satisfy the requirements in the input prompt. For example, ChatGPT sometimes still outputs the layout description  $d_{360}$  with an erroneous format, making automatic prompt extraction fail catastrophically. We automatically detect such cases, and re-run line 3 to ensure the algorithm flow, see Appendix A for more details.

L-MAGIC requires *no model fine-tuning*, which ensures the zero-shot performance and makes it capable of accepting other types of inputs leveraging conditional generative models (see Sec. 4 for results). This advantage also allows individual modules to be replaced by future methods to enhance the performance, *e.g.*, change BLIP-2+ChatGPT to GPT-4V [1], or use other inpainting models.

## 4. Experiments

We describe our experimental setup in Sec. 4.1 and compare our method with other panorama generation methods in Sec. 4.2. We then analyze the contribution of individual components of our methods in Sec. 4.3. Finally, we demonstrate several down-stream applications, such as scene fly-through and 3D scene generation in Sec. 4.4.

### 4.1. Experimental Setup

**Baselines.** We evaluate our method on both image-to-panorama and text-to-panorama. For *image-to-panorama*, we compare against:

1. Stable Diffusion v2 [19]: we use the prompt ‘360 degree panorama of <scene description>’ to inpaint the panorama image in a single diffusion process.
2. Text2room [11]: we take the panorama generation component (at steps 11-20 of the pipeline) for comparison.
3. MVDiffusion image-to-panorama model [25].

To enable text conditioning in these methods, we use BLIP-2 to obtain the description of the input image. For *text-to-panorama*, we compare against:

1. Text2light [8]: GAN-based text to panorama model.
2. Stable Diffusion v2 [19]: we use the prompt ‘360 degree panorama of <input prompt>’ to generate the panoramic image in a single diffusion process.
3. LDM3D [22] panorama model: we only use the output RGB image and the prompt ‘360 degree panorama of <input prompt>’ to generate the panoramic image in a single diffusion process.
4. Text2room panorama generation module [11].
5. MVDiffusion text-to-panorama model [25].

**Implementation Details.** L-MAGIC is implemented with PyTorch [15] and the official release of BLIP-2 [13], Stable Diffusion [19] and the ChatGPT4 API [4]. It takes 2-5 minutes to generate a 360° panorama depending on the number of repetitions in line 17 of Alg. 1. We take the official code and model for all other methods. In text-to-panorama, L-MAGIC uses Stable Diffusion v2 conditioned on the given text prompt to generate the input image.

**Data.** To evaluate the in-the-wild performance, we collect data that does not overlap with the training data of any method for both tasks. For image-to-panorama, we use 20 indoor and 20 outdoor images from tanks-and-temples [12] and RealEstate10K [31]. For text-to-panorama, we use

ChatGPT to generate 20 random scene descriptions (10 indoor and 10 outdoor, see Appendix C).

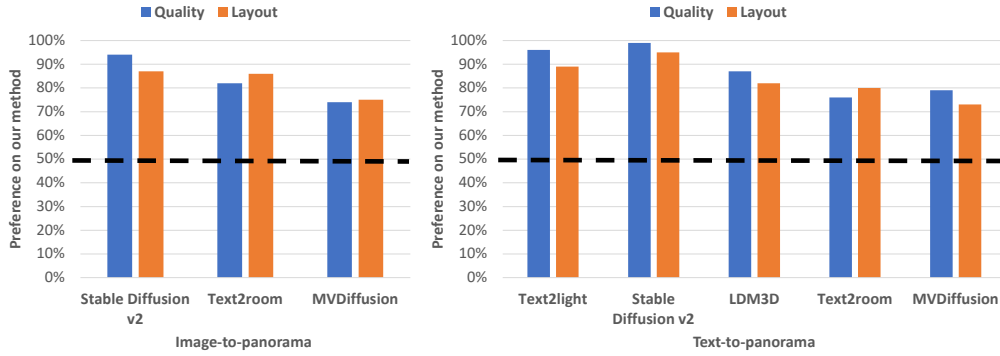
**Metrics.** To evaluate different methods with respect to quality and multi-view diversity, we compute the Inception Score (IS) [20] for the perspective views of the panorama. Since existing quantitative metrics do not capture all aspects of human perception of quality [7], we follow existing works [8, 11, 14] for a complementary human evaluation. To this end, we set up a voting web page that shows side-by-side panoramic scenes generated using the same input, one with our method and one with a baseline. We ask 15 anonymous voters to choose which panorama has higher quality and scene structure (see Appendix D for an example voting page). To minimize voting bias, we randomly shuffle the order of the side-by-side panoramas and hide the generation method names. We use the votes to compute a preference score from 0 to 1 for our method compared to the baselines. This score is simply the percentage of votes for our method with respect to quality and structure.

### 4.2. Main Results

As shown in Fig. 3, our method performs better for both image-to-panorama and text-to-panorama, even compared to task-specific methods. To further understand the performance of different methods, we provide in Fig. 4 and 5 the qualitative results for both tasks. *Stable Diffusion v2* treats a panorama as a single image. It cannot close the 360° loop since equirectangular projection [5] splits the loop-closure area to two sides of an image (moved to the middle of the rendered panorama for better visualization). In the meantime, due to the lack of large-scale panorama training data, it still generates unnecessarily repeated objects such as multiple beds in a bedroom. *Text2room* and *MVDiffusion* separate a panorama into multiple perspective views. Inpainting them using the same prompt results in unreasonably repeated objects in multiple views. Due to the limited panorama training data, *Text2light* cannot fully understand zero-shot scene descriptions generated by ChatGPT, resulting in scenes not consistent with the input prompt. Similar to Stable Diffusion v2, treating panorama as a single image makes it fail on loop closure. *LDM3D* is fine-tuned on top of a perspective latent diffusion model. Though better than Text2light, it still cannot close the loop and sometimes fails to generate scenes that are consistent with the details of the prompt (*e.g.*, generating a non-modern living room when asked for a modern one). *Our method* works robustly on various inputs, generating panoramic scenes with high perspective rendering quality and reasonable 360° scene layouts (see supplementary videos for more details).

### 4.3. Analysis

We further analyze different components of our method in this section. The analysis is conducted from 3 aspects: 1)



(a) Human evaluations.

Image-to-panorama	IS ( $\uparrow$ )
Stable Diffusion v2	1.93
Text2room	2.11
MVDiffusion	2.08
Ours	<b>2.66</b>

Text-to-panorama	IS ( $\uparrow$ )
Text2light	1.67
Stable Diffusion v2	1.58
LDM3D	1.86
Text2room	1.89
MVDiffusion	1.86
Ours	<b>2.26</b>

(b) Algorithmic evaluations

Figure 3. **Quantitative results for image-to-panorama and text-to-panorama.** (a) Human evaluations. Each baseline has two bars representing respectively the quality of rendered perspective views and the 360° layout. The value of the bar means the frequency where our method is preferred in the voting. Above 50% (dashed line) means our method is more preferred than the corresponding baseline. (b) Algorithmic evaluation by computing the Inception Score (IS). L-MAGIC consistently outperforms previous methods on both metrics.



Figure 4. **Image-to-panorama visualizations.** Stable Diffusion v2 cannot close the 360° loop (sharp boundaries at the middle). Text2room and MVDiffusion lack mechanisms to avoid duplicate objects. L-MAGIC outputs have high local view quality and coherent scene layouts.

the scene prior, 2) the inpainting method and 3) quality enhancement techniques. For each aspect, we remove a component or replace it with other methods, and perform the same evaluation as in the main experiments. We use the same data used in the image-to-panorama main experiment for analysis. The results are reported in Fig. 6. Please refer to Appendix E for the visualization comparisons.

For the *scene prior*, we remove the prior from chatGPT (no GPT) and all text guidance (no prompt) respectively. Without the global layout prior from chatGPT, the structure of the outputs becomes worse. Without any prompt guidance, both the scene layout and the perspective view ren-

dering quality degrade severely.

For the *inpainting method*, we replace the Stable Diffusion v2 model with 3 state-of-the-art text-conditioned inpainting methods, namely, Blended Latent Diffusion (BLD inpaint) [6], Deep Floyd (DF inpaint) [2] and Stable Diffusion XL (SDXL inpaint) [16]. Interestingly, though some of the methods [16] are published later than Stable Diffusion v2, their capacity to perform large mask inpainting is limited on in-the-wild images, resulting in worse performance in terms of both scene layouts and rendering quality. Note that the inception scores for some methods (e.g., BLD inpaint) are higher than ours despite a much worse



Figure 5. **Text-to-panorama visualizations.** Text2light, Stable Diffusion v2 and LDM3D cannot close the 360° loop (sharp boundaries at the middle). Text2room and MVDiffusion generate panoramas with duplicate objects. L-MAGIC effectively addresses these problems, resulting in high-quality panorama with reasonable scene layouts.

performance from human evaluation. This is caused by the adversarial samples generated by these methods (see Appendix F for examples), where the local patches of the adversarial samples are not consistent with the input image, yet leading to a high diversity in the inception score. Similar issues have also been discovered in other problems [7], which shows the importance of human evaluation.

For the *quality enhancement techniques*, we remove respectively the super-resolution (no SR) and smoothing (no smooth) techniques mentioned in Sec. 3.3. Though the scene layout does not degrade much, the perspective view rendering quality is lower due to the blur or artifacts.

#### 4.4. Applications

Combining matured computer vision techniques makes our pipeline applicable to a wide range of applications.

**Anything-to-panorama.** Conditional diffusion models [19, 29] can now generate an image from diverse types

of inputs. The strong zero-shot performance of L-MAGIC makes it possible to generate panoramic scenes from potentially any inputs compatible with conditional diffusion models. As shown in Fig. 7, we can use [29] to generate a single image from 1) a depth map, 2) a sketch drawing or 3) a colored script or segmentation image. Then, this generated image can be used in L-MAGIC to produce realistic panoramic scenes. This flexibility makes L-MAGIC beneficial to a wide range of design applications.

**3D scene generation.** Applying state-of-the-art depth estimation models, we can further generate 3D scenes from the output of L-MAGIC. Specifically, we compute the depth map for multiple perspective views, then we merge the depth maps into the equirectangular image plane by aligning all views to the initial view. Then we convert the corresponding panoramic depth map to a 3D point cloud. We use Metric3D [27] and DPT-hybrid [17] to estimate the depth for indoor and outdoor scenes respectively. The alignment

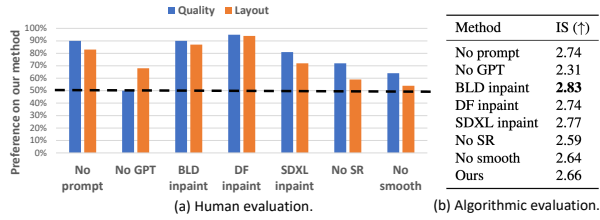


Figure 6. **Analysis.** We follow the same evaluation protocol as in the main experiment (Fig. 3). From human evaluations, we see that removing the language assistance results in severe degradation in either the scene layout (No GPT) or both the quality and the layout (No Prompt). Replacing Stable Diffusion v2 with other inpainting methods results in performance drop on both the quality and the layout. The rendering quality decreased with quality enhancement techniques removed. The Inception Score cannot accurately reflect the performance on adversarial examples (see Appendix F), resulting in inconsistency with human evaluation results.



Figure 7. **Panorama generated from other input modalities.** L-MAGIC can effectively create panoramas from various input modalities, such as an input depth map (top), a sketch drawing (middle) and a colored script or a segmentation mask (bottom). The dotted bounding box indicates the region of the initial perspective view, which is generated by conditional diffusion models.

is done by optimizing the scale and shift of each depth map to enforce the depth from multiple views to be the same at the same pixel. Fig. 8 shows sampled results. Despite some artifacts caused by the limitation of monocular depth models, L-MAGIC can generate diverse indoor and outdoor point clouds from various types of inputs.

**Immersive video.** One can also render immersive videos from our panorama. Specifically, we first generate a panorama using our pipeline, and then generate a series of camera poses for individual video frames. Then we warp the panorama to each frame view according to the camera poses. To further enable scene fly-through with camera translations, we apply depth-based warping when translation is involved in a frame, and inpaint the missing region introduced after translation. See Appendix G for implemen-



Figure 8. **3D point cloud generation.** Performing depth estimation on the generated panorama further enables the creation of 3D point clouds from diverse inputs. We can generate point clouds for both indoor and outdoor scenes, even the underwater scene with clear geometry of the fishes and coral reefs.

tation details and the supplementary videos for the results.

## 5. Conclusion

We have proposed L-MAGIC, a novel method that can generate 360° panoramic scenes from a single input image. L-MAGIC leverages large (vision) language models to guide diffusion models to smoothly extend the local scene content with a coherent 360° layout. We have also proposed techniques to enhance the quality and resolution of the generated panorama. Extensive experiments demonstrate the effectiveness of L-MAGIC, outperforming state-of-the-art methods for image-to-panorama and text-to-panorama across metrics. Combined with state-of-the-art computer vision techniques such as conditional diffusion models and depth estimation models, our method can consume various types of inputs (text, sketch drawings, depth maps etc.) and generate outputs beyond a single panoramic image (videos with camera translations, 3D point clouds, etc.). See Appendix H for discussions about limitations and future works.



## References

- [1] <https://openai.com/blog/chatgpt-can-now-see-hear-and-speak>. 5
- [2] <https://github.com/deep-floyd/if>. 6
- [3] <https://huggingface.co/stabilityai/stable-diffusion-x4-upscaler>. 4
- [4] <https://openai.com/blog/introducing-chatgpt-and-whisper-apis>. 2, 3, 5
- [5] [https://en.wikipedia.org/wiki/equirectangular\\_projection](https://en.wikipedia.org/wiki/equirectangular_projection). 4, 5
- [6] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42(4):1–11, 2023. 6
- [7] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018. 5, 7
- [8] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2light: Zero-shot text-driven hdr panorama generation. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 2, 5
- [9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2
- [10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2
- [11] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. *arXiv preprint arXiv:2303.11989*, 2023. 1, 2, 3, 5
- [12] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 5
- [13] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 2, 3, 5
- [14] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 5
- [15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5
- [16] 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. 6
- [17] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 7, 13
- [18] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 3
- [19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3, 5, 7
- [20] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 5
- [21] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2
- [22] Gabriela Ben Melech Stan, Diana Wofk, Scottie Fox, Alex Redden, Will Saxton, Jean Yu, Estelle Aflalo, Shao-Yen Tseng, Fabio Nonato, Matthias Muller, et al. Ldm3d: Latent diffusion model for 3d. *arXiv preprint arXiv:2305.10853*, 2023. 2, 5
- [23] Peter Sturm. Multi-view geometry for general camera models. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 206–212. IEEE, 2005. 3
- [24] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 2
- [25] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdifffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv preprint arXiv:2307.01097*, 2023. 1, 2, 3, 5
- [26] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 2022. 2
- [27] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9043–9053, 2023. 7
- [28] Jingyi Zhang, Jiaying Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *arXiv preprint arXiv:2304.00685*, 2023. 2
- [29] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2, 7
- [30] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, XiaoLei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie

Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. 2

- [31] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 5