
ChatCam: Empowering Camera Control through Conversational AI

Xinhang Liu¹ Yu-Wing Tai² Chi-Keung Tang¹
¹HKUST ²Dartmouth College

Abstract

Cinematographers adeptly capture the essence of the world, crafting compelling visual narratives through intricate camera movements. Witnessing the strides made by large language models in perceiving and interacting with the 3D world, this study explores their capability to control cameras with human language guidance. We introduce *ChatCam*, a system that navigates camera movements through conversations with users, mimicking a professional cinematographer’s workflow. To achieve this, we propose *CineGPT*, a GPT-based autoregressive model for text-conditioned camera trajectory generation. We also develop an *Anchor Determinator* to ensure precise camera trajectory placement. *ChatCam* understands user requests and employs our proposed tools to generate trajectories, which can be used to render high-quality video footage on radiance field representations. Our experiments, including comparisons to state-of-the-art approaches and user studies, demonstrate our approach’s ability to interpret and execute complex instructions for camera operation, showing promising applications in real-world production settings. We will release the codebase upon paper acceptance.

1 Introduction

Cinematographers skillfully capture the essence of the 3D world by maneuvering their cameras, creating an array of compelling visual narratives [8]. Achieving aesthetically pleasing results requires not only a deep understanding of scene elements and their interplay but also meticulous execution of techniques.

Recent progress of large language models (LLMs) [1] has marked a significant milestone in AI development, demonstrating their capability to understand and act within the 3D world [29, 30, 87]. Witnessing this evolution, our work explores the feasibility of empowering camera control through conversational AI, thus enhancing the video production process across diverse domains such as documentary filmmaking, live event broadcasting, and virtual reality experiences.

Although the community has devoted considerable effort to controlling the trajectories of objects and cameras in video generation approaches for practical usage [4, 82, 75, 28], or predicting similar sequences through autoregressive decoding processes [35, 64], generating camera trajectories has yet to be explored. This task involves multiple elements such as language, images, 3D assets, and, beyond mere accuracy, necessitates visually pleasing rendered videos as the ultimate goal.

We propose *ChatCam*, a system that allows users to control camera operations through natural language interaction. As illustrated in Figure 1, leveraging an LLM agent to orchestrate camera operations, our method assists users in generating desired camera trajectories, which can be used to render videos on radiance field representations such as NeRF [52] or 3DGS [36].

At the core of our approach, we introduce *CineGPT*, a GPT-based autoregressive model that integrates language understanding with camera trajectory generation. We train this model using a paired text-trajectory dataset to equip it with the ability for text-conditioned trajectory generation. We also

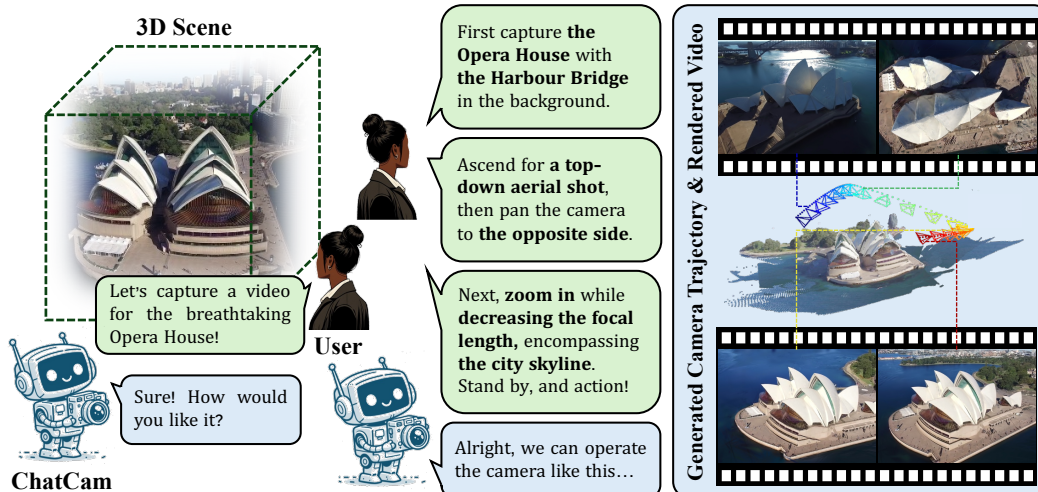


Figure 1: **Empowering camera control through conversational AI.** Our proposed *ChatCam* assists users in generating desired camera trajectories through natural language interactions. The generated trajectories can be used to render videos on radiance field representations such as NeRF [52] or 3DGS [36].

propose an *Anchor Determinator*, a module that identifies relevant objects within the 3D scene to serve as anchors, ensuring correct trajectory placement based on user specifications. Our LLM agent parses compositional natural language queries into semantic concepts. With these parsed sub-queries as inputs, the agent then calls our proposed *CineGPT* and *Anchor Determinator*. It composes the final trajectory with the outputs from these tools, which can ultimately be used to render a video that fulfills the user’s request.

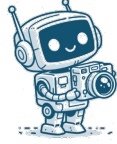
With comprehensive evaluations and comparisons to other state-of-the-art methods, our method exhibits a pronounced ability to interpret and execute complex instructions for camera operation. Our user studies further demonstrate its promising application prospects in actual production settings. In summary, this paper’s contributions are as follows:

- We introduce *ChatCam*, a system that, for the first time, enables users to operate cameras through natural language interactions. It simplifies sophisticated camera movements and reduces technical hurdles for creators.
- We develop *CineGPT* for text-conditioned camera trajectory generation and an *Anchor Determinator* for precise camera trajectory placement. Our LLM agent understands users’ requests and leverages our proposed tools to complete the task.
- Extensive experiments demonstrate the effectiveness of our method, showing how AI can effectively collaborate with humans on complex tasks involving multiple elements such as language, images, 3D assets, and camera trajectories.

2 Related Work

Multimodal Language Models. Large-scale language models (LLMs) [9, 19, 17, 1, 70] enabled by extensive datasets and model size, have demonstrated surprising emerging abilities. The emergence of multimodal models [43, 42, 33] is captivating as they can process text alongside other modalities such as images [25], audio [27], and videos [78]. Some unified models can perceive inputs and generate outputs in various combinations of text, images, videos, and audio [51, 68, 77, 86]. LLMs hold the potential to act as agents [76, 80, 62], allowing them to be driven by goals, reason about their objectives, devise plans, utilize tools, and interact with and gather feedback from the environment. Our proposed method involves multiple modalities including language, images, 3D fields, and camera trajectories, and utilizes LLMs as agents to assist users in operating cameras.

Radiance Field Representations. Utilizing continuous 3D fields modeled by MLPs and volumetric rendering, Neural Radiance Fields (NeRFs) [52] achieved breakthrough for novel view synthesis.



From outside the window, enter the bedroom and explore its layout following an S-shaped path.



Observation: The user is requesting a camera trajectory, starting from a location outside the window and then covering the bedroom’s layout following an S-shaped path.

Reasoning: I need to find a starting point outside the window, then move through the window into the room and capture the room using an S-shaped path. Additionally, I should use a smooth panning speed to better showcase the room’s layout.

Plan:

1. Call **Anchor Determinator** with “capture the bedroom from outside the window”.
2. Call **CineGPT** with “pan straight forward, smooth panning speed”.
3. Call **CineGPT** with “an S-shaped path, smooth panning speed”.
4. Connect output trajectories from steps 2 and 3, ensuring they pass through the anchor point determined in step 1.
5. Ask the user for further clarification if needed.
6. Render a video using the generated camera trajectory.

Figure 2: **Overview of the ChatCam pipeline.** Given a camera operation instruction, ChatCam reasons the user’s request and devises a plan to generate a trajectory using our proposed CineGPT and Anchor Determinator. The agent then utilizes the outputs from these tools to compose the complete trajectory and render a video.

Subsequent research has emerged to improve NeRFs and broaden their applications [69], including enhancing rendering quality [5, 6, 7?, 47], modeling dynamic scenes [84, 55, 58, 71, 74, 10, 22, 2?], improving computational efficiency [83, 23], and facilitating 3D scene editing [45, 84, 73, 34, 46, 40]. Replacing the deep MLPs with a feature voxel grid network has demonstrated enhancement in both training and inference speed [13, 66, 24, 54]. More recently, 3D Gaussian Splatting [36] has further advanced visual quality and rendering efficiency. Compared to traditional 3D representations, radiance field representations offer superior photorealistic rendering quality, therefore, this study focuses on camera manipulation upon mainstream radiance field representations such as NeRF or 3DGS.

3D Scene Understanding. Early methods for 3D semantic understanding [32, 67, 79, 15] primarily focused on the closed-set segmentation of point clouds or voxels. NeRF’s capability to integrate information from multiple viewpoints has spurred its application in 3D semantic segmentation [88, 20, 46, 53, 65, 26, 60, 31, 48, 49, 21]. Among these, [40, 37, 12] combine image embeddings from effective 2D image feature extractors [41, 11, 59, 39] to achieve language-guided object localization, segmentation, and editing. [21] proposes semantic anisotropic Gaussians to simultaneously estimate geometry, appearance, and semantics in a single feed-forward pass. Another line of research integrates 3D with language models for tasks such as 3D question answering [3], localization [14, 57, 81], and captioning [16]. Additionally, [29, 30, 87] propose 3D foundation models to handle various perception, reasoning, and action tasks in 3D environments. However, the AI-assisted operation of cameras within 3D scenes remains an unexplored area.

Trajectory Control and Prediction. Controlling the trajectories of objects and cameras is crucial to advance current video generation approaches for practical usage. TC4D [4] incorporates trajectory control for 4D scene generation with multiple dynamic objects. Direct-a-Video [82], MotionCtrl [75],

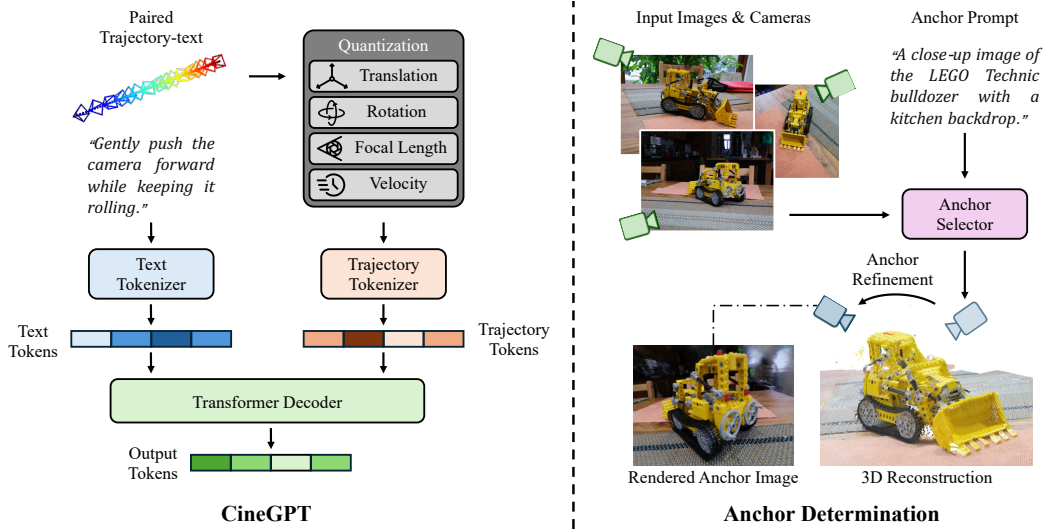


Figure 3: **(a) CineGPT.** We quantize camera trajectories to sequences of tokens and adopt a GPT-based architecture to generate the tokens autoregressively. Learning trajectory and language jointly, CineGPT is capable of text-conditioned trajectory generation. **(b) Anchor Determination.** Given a prompt describing the image rendered from an anchor point, the anchor selector chooses the best matching input image. An anchor refinement procedure further fine-tunes the anchor position.

and CameraCtrl [28] manage camera pose during video generation; however, they are either limited to basic types or necessitate fine-tuning of the video diffusion model. Moreover, these approaches require user-provided trajectories, whereas we, for the first time, generate camera trajectories conditioned on text.

3 Method

Figure 2 provides an overview of our method’s pipeline. ChatCam analyzes the user’s camera operation instruction and devises a plan to generate a trajectory using our proposed CineGPT and Anchor Determinator. Finally, an AI agent utilizes the outputs from these tools to compose the complete trajectory.

3.1 Text-Conditioned Trajectory Generation

To enable text-conditioned trajectory generation, we collect a text-trajectory dataset and introduce CineGPT, a GPT-based autoregressive model integrating language and camera trajectories. Illustrated in Figure 3 (a), our method quantizes camera trajectories into a sequence of trajectory tokens using a trajectory tokenizer. Subsequently, a multi-modal transformer decoder is employed to convert input tokens into output tokens. Upon training, our model adeptly generates token sequences based on user-provided text prompts. These sequences are then de-quantized to reconstruct the camera trajectory.

Camera Trajectory Parameterization. For each single frame, our camera parameters include rotation $\mathbf{R} \in \mathbb{R}^{3 \times 3}$, translation $\mathbf{t} \in \mathbb{R}^3$, and intrinsic parameters $\mathbf{K} \in \mathbb{R}^{3 \times 3}$. We further convert the rotation matrix \mathbf{R} into the $\mathbb{S}^2 \times \mathbb{S}^2$ space [89] to facilitate computational efficiency and simplify the optimization process. The total M -frame camera trajectory is formulated as:

$$c_{1:M} = \{c_i\}_{i=1}^M = \{(\mathbf{R}_i, \mathbf{t}_i, \mathbf{K}_i)\}_{i=1}^M. \quad (1)$$

To additionally model the velocity of camera movement, we introduce a global parameter t representing the total duration. Consequently, the instantaneous velocity of each frame can be approximated by the relative translation and rotation to the previous frame over unit time.

Text-Trajectory Dataset. Given the scarcity of readily available data on camera operations, we manually constructed approximately 1000 camera trajectories using Blender [18]. These trajectories encompass a diverse range of movements, including various combinations of translations, rotations,

focal lengths, and velocities. Each trajectory is accompanied by a human language description detailing the corresponding movements. This dataset spans various scenarios, capturing both simple pan-tilt-zoom motions and more complex trajectories mimicking real-world scenarios.

Trajectory Tokenizer. We leverage a trajectory tokenizer based on the Vector Quantized Variational Autoencoders (VQ-VAE) architecture [72] to represent camera trajectories as discrete tokens. Our trajectory tokenizer consists of an encoder \mathcal{E} and a decoder \mathcal{D} . Given an M -frame camera trajectory $c_{1:M} = \{c_i\}_{i=1}^M$, the encoder \mathcal{E} encodes it into L trajectory tokens $z_{1:L} = \{z_i\}_{i=1}^L$, where $L = M/l$ and l is the temporal downsampling rate. The decoder \mathcal{D} then decodes $z_{1:L}$ back into the trajectory $\hat{c}_{1:M} = \{\hat{c}_i\}_{i=1}^M$. Specifically, the encoder \mathcal{E} first encodes frame-wise camera parameters $c_{1:M}$ into a latent vector $\hat{z}_{1:L} = \mathcal{E}(c_{1:M})$, by performing 1D convolutions along the time dimension. We then transform $\hat{z}_{1:L}$ into a collection of codebook entries z through discrete quantization. The learnable codebook $Z = \{z_i\}_{i=1}^K$ consists of K latent embedding vectors, each with dimension d . The quantization process $Q(\cdot)$ replaces each row vector with its nearest codebook entry, as follows:

$$z_i = Q(\hat{z}_i) = \arg \min_{z_k \in Z} \|\hat{z}_i - z_k\|_2^2, \quad (2)$$

where $\|\cdot\|_2$ denotes the Euclidean distance. After quantization, the decoder projects $z_{1:L}$ back to the trajectory space as the reconstructed trajectory $\hat{c}_{1:M} = \mathcal{D}(z_{1:L})$. In addition to the reconstruction loss, we adopt embedding loss and commitment loss similar to those proposed in [85] to train our trajectory tokenizer. With a trained trajectory tokenizer, a camera trajectory $c_{1:M}$ can be mapped to a sequence of trajectory tokens $z_{1:L}$, facilitating the joint representation of camera trajectory and natural language for text-conditioned trajectory generation.

Cross-Modal Transformer. We utilize a cross-modal transformer decoder to generate output tokens from input tokens, which may consist of text tokens, trajectory tokens, or a combination of both. These output tokens are subsequently converted into the target space. To train our decoder-only transformer, we denote our source tokens as $X_s = \{x_s^i\}_{i=1}^{N_s}$ and target tokens as $X_t = \{x_t^i\}_{i=1}^{N_t}$. We feed source tokens into it to predict the probability distribution of the next potential token at each step $p_\theta(x_t|x_s) = \prod_i p_\theta(x_t^i|x_t^{<i}, x_s)$. The objective function is formulated as:

$$\mathcal{L}_{\text{LM}} = - \sum_{i=1}^{N_t} \log p_\theta(x_t^i|x_t^{<i}, x_s). \quad (3)$$

By optimizing this objective, we aim to equip CineGPT with the ability to capture intricate patterns and relationships within the data distribution. We then fine-tune CineGPT on supervised trajectory-language translation leveraging our paired text-trajectory dataset, where the input for this stage can either be a camera trajectory or a text description, while the target is the opposite modality. During inference, CineGPT can generate camera trajectories solely from textual descriptions as inputs.

3.2 Object-Centric Trajectory Placement with Anchors

While CineGPT enables text-conditioned trajectory generation, its generation process solely focuses on determining the camera’s movements, without contextual connection to specific scenes. Consequently, CineGPT alone cannot effectively handle user prompts that involve object-centric descriptions, such as directives like “directly above the Sydney Opera House”. In this light, we bridge trajectory generation with each underlying scene with “anchors” serving as reference points within the scene to achieve more accurate placement of trajectories, as illustrated in Figure 3 (b).

Our anchor determination procedure takes natural language descriptions of an image as input. This procedure identifies a set of camera parameters that can render an image that best matches the given description. Current 3D visual grounding approaches [57, 81] typically entail learning a 3D feature field [40, 37] and localizing objects within the scene, which often results in high computational costs. In contrast, our anchor determinator adopts a different strategy. Initially, it selects the input image that best matches the given text description as an initial anchor. Subsequently, an anchor refinement process is employed to iteratively improve upon this initial anchor, ultimately yielding the final anchor. This approach offers a more efficient alternative to traditional methods, reducing computational overhead while still achieving accurate scene anchoring.

Initial Anchor Selector. Since our method leverages radiance field representations to render videos, we naturally have access to the input images for training the 3D scene representations. We utilize

an initial anchor selector based on CLIP [59] to choose the image from these input images that best matches the text prompt. To be specific, for i -th input image I_i , we extract their CLIP image features and convert the text prompt T into a CLIP text feature. Next, we compute the cosine similarity between the CLIP text feature vector and each of the CLIP image feature vectors. We select the best matching image with the highest cosine similarity score as the initial anchor. This can be formulated as:

$$i_{\text{anchor}} = \arg \max_i \frac{f_{\text{image}}(I_i) \cdot f_{\text{text}}(T)}{\|f_{\text{image}}(I_i)\| \|f_{\text{text}}(T)\|}, \quad (4)$$

where $f_{\text{image}}(\cdot)$ and $f_{\text{text}}(\cdot)$ represent the image and text feature extractor, respectively.

Anchor Refinement. Using the camera parameters c_{anchor} associated with the selected image as initialization, we further minimize the following objective to obtain the final anchor camera parameters:

$$\min_c \mathcal{L}_{\text{anchor}}(c) = - \frac{f_{\text{image}}(R(c)) \cdot f_{\text{text}}(T)}{\|f_{\text{image}}(R(c))\| \|f_{\text{text}}(T)\|}, \quad (5)$$

where $R(\cdot)$ is the rendering function and c is initialized with c_{anchor} . The optimization of c is performed using gradient descent, with the update rule given by:

$$c_{t+1} = c_t - \eta \nabla_c \mathcal{L}_{\text{anchor}}(c_t), \quad (6)$$

where η is the learning rate. The optimization typically achieves convergence within 100 to 1000 steps. This refinement process ensures that the camera parameters are adjusted to better match the text prompts, handling cases where the initial input images do not align well with the prompts.

3.3 Trajectory Generation through User-Friendly Interaction

With our proposed CineGPT and anchor determination, a large language model acts as an agent to interpret the user’s requests, generates a plan to use various tools, and composes a final camera trajectory. We adopt GPT-4 [1] to interpret users’ natural language inputs and subsequently produce trajectory prompts. Specifically, we use a carefully designed prompt to instruct the LLM agent to reason about the user’s requirements and devise a plan consisting of the following steps: 1) Break down the complex text query into sub-tasks that CineGPT and the Anchor Determinator can effectively handle. 2) Use these tools to generate atomic trajectories and determine anchor points. 3) Compose the final trajectory by concatenating atomic trajectories and ensuring they pass through the anchors.

Observing, Reasoning, and Planning. Research indicates that LLMs can be prompted to decompose complex goals into sub-tasks, essentially thinking step-by-step [76]. As illustrated in Figure 2, we begin by instructing the agent to describe its observations, providing a summary of the current situation. The agent then uses this summary to reason and develop a mental scratchpad for high-level planning. Finally, it outlines specific steps to achieve the overarching goal of generating the user-required camera trajectory.

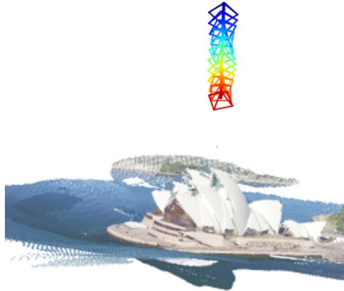
Utilization of Proposed Tools. We inform our agent of the expected input and output format, i.e., the APIs, of our proposed CineGPT and Anchor Determinator, and instruct the agent to interact with them following the given format. In its outlined specific steps to generate the user-required camera trajectory, it first calls CineGPT and Anchor Determinator to obtain atomic trajectories and anchor points, respectively. Note that both tools can be called multiple times, and multiple atomic trajectories can later be concatenated into final trajectories that pass through all anchor points correctly.

Final Trajectory Composition. Here we explain how to combine atomic trajectories from CineGPT with anchor points to form the final trajectory. The agent first decides the role of the anchors in the ultimate trajectory, either as a starting point or an ending point of some atomic trajectory. Then affine transformations are applied to the respective atomic trajectories to ensure that their starting or ending points align with the anchor points. For the remaining atomic trajectories not controlled by anchor points, affine transformations are applied to make the endpoint of the previous trajectory align with the starting point of the subsequent trajectory.

4 Experiments

We assess the performance of our proposed ChatCam for human language-guided camera operation across a series of challenging scenarios. Through ablation studies, we provide empirical evidence of

Zoom in from directly above the Sydney Opera House and roll the camera.



Do a dolly zoom focusing on the chandelier.



Pan the camera from left to right along the piano, then turn to look at the bicycle on the right.

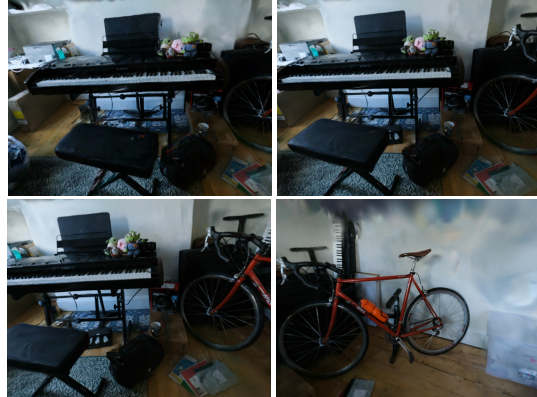


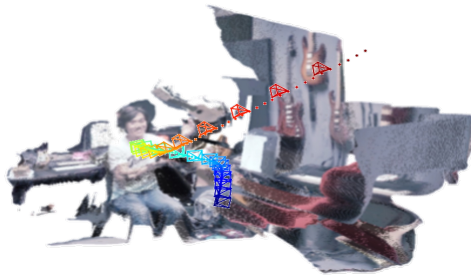
Figure 4: **Qualitative results on indoor and outdoor scenes.** Visualizations of our generated trajectories from input text descriptions and the frames in the final rendered video. Our method is capable of understanding and executing instructions and providing correct translations, rotations, and camera focal lengths. Additionally, our method can comprehend more specialized terms such as “dolly zoom”.

the effectiveness of its fundamental components. We kindly refer the reader to our supplementary material for additional experimental results, including rendered **videos**.

4.1 Experimental Setup

Implementation Details. We implement our approach using PyTorch [56] and conduct all the training and inference on a single NVIDIA RTX 4090 GPU with 24 GB RAM. The trajectory tokenizer has a codebook with $K = 256$ latent embedding vectors, each with dimension $d = 256$. The temporal downsampling rate of the trajectory encoder is $l = 4$. Our cross-modal transformer decoder consists of 24 layers, with attention mechanisms employing an inner dimensionality of 64.

Sweep across the boy in black, the keyboard, and the boy in white, then zoom out to frame the boys and the white guitar together.



Starting from the man playing cards, pan the camera along the corridor, then back up to the end and look to the right.



Figure 5: **Qualitative results on human-centric scenes.** Visualizations of our generated trajectories from input text descriptions and the frames in the final rendered video. Our method performs effectively in scenes with multiple humans.

The remaining sub-layers and embeddings have a dimensionality of 256. We train CineGPT using the Adam optimizer [38] with an initial learning rate of 0.0001. It takes approximately 30 hours to converge. Our anchor determination utilizes CLIP [59] with a ViT-B/32 Transformer architecture. The learning rate of anchor refinement is 0.002. By default, we use GPT-4 [1] as our LLM agent, and its prompt will be released with our codebase. We render final videos using 3DGS [36] as the 3D representation.

Tested Scenes. We tested our method on scenes from a series of datasets suitable for 3D reconstruction with radiance field representations, including: (i) *mip-NeRF 360* [6], a real dataset with indoor and outdoor scenes. (ii) *OMMO* [50], a real dataset with large-scale outdoor scenes. (iii) *Hypersim* [61], a synthetic dataset for indoor scenes. (iv) *MannequinChallenge* [44], a real dataset for human-centric scenes. If camera poses associated with images were not provided, we used COLMAP [63] for camera pose estimation. For each scene, we reconstructed using all available images without train-test splitting.

Baselines. As the first method to enable human language-guided camera operation, there is no established direct baseline for comparison. Therefore, we adopt 3D understanding approaches based on radiance field representations to let the LLM agent attempt to select a series of images corresponding to the input text from input images and interpolate their camera poses to construct camera trajectories. These methods include LERF [37], utilizing CLIP embeddings, and SA3D [12], utilizing SAM embeddings.

Evaluation Metrics. To evaluate the accuracy of the generated trajectories, we manually construct ground truth trajectories and compute the mean squared errors (MSEs) of translations and rotations relative to them. Additionally, we conduct a user study to evaluate the rendered videos using generated camera trajectories, where users are asked to select the video with the best **visual quality** and best **alignment** with the input text.

Facing the piano, pull the camera back, then glance over at the TV on the left, and back to the piano.



Figure 6: **Qualitative comparisons.** Our approach avoids moving the camera to unreasonable positions such as inside objects, obtaining videos with better visual effects, and aligning best with input texts.

Table 1: **Quantitative comparisons and evaluations.** Our full model performs better than baselines and variants in terms of trajectory accuracy, visual quality, and alignment with input text.

Method	LLM Agent	Anchor Determination	Translation MSE (\downarrow)	Rotation MSE (\downarrow)	Visual Quality (\uparrow)	Alignment (\uparrow)
SA3D [12]	<i>GPT-4</i>	-	19.5	6.3	5.7	3.8
LERF [37]	<i>GPT-4</i>	-	17.7	4.9	9.4	28.3
ChatCam (Ours)	<i>LLaMA-2</i>	✓	6.4	3.6	-	-
ChatCam (Ours)	<i>GPT-3.5</i>	✓	7.3	3.5	-	-
ChatCam (Ours)	<i>GPT-4</i>	✗	16.2	8.5	-	-
ChatCam (Ours)	<i>GPT-4</i>	✓	5.3	2.9	84.9	67.9

4.2 Results

As shown in Figure 4, our method demonstrates the ability to understand and execute camera operation instructions on a range of complex indoor and outdoor scenes, giving appropriate translation, rotation, and focal length. Our method also understands more technical terms such as dolly zoom, which creates a special visual effect by zooming the camera out while adjusting the focus. In Figure 5 we further showcase the qualitative results of our method in human-centric scenes. Our method can correctly handle user instructions about specific people and create correct and vivid visual effects.

Comparisons. In Figure 6 we qualitatively compare our method with LLM agents utilizing SA3D or LERF to locate target objects. The baselines do simple interpolation of keyframes because they have no knowledge about camera trajectories and tend to move the camera to unreasonable spots (such as entering an object). Therefore, the video rendered by baselines contains artifacts and is not correctly consistent with the input text. However, our method achieves better visual quality and alignment with input texts. Quantitative comparisons in Table 1 further prove that our method has better performance and is preferred by users.

Ablation Study. We present our ablation study in Table 1. We evaluate the performance of our method using different LLMs as agents. Our approach achieved the best accuracy using GPT-4 [1] as the agent, better than GPT-3 [9] and LLaMA-2 [70]. Without our proposed anchor determination, our method cannot correctly place trajectories within 3D scenes, thereby being less accurate than our full model.

5 Conclusion

This paper presents ChatCam, a system designed for camera operation through natural language interactions. By introducing CineGPT, we bridge the gap between human language guidance and camera control, achieving text-conditioned trajectory generation. Our proposed anchor determination procedure further ensures precise camera trajectory placement. Our LLM agent comprehends users’ requests and effectively utilizes our proposed tools to compose the final trajectory. Through extensive experiments, we demonstrate the effectiveness of ChatCam, showcasing its ability to collaborate with humans on complex tasks involving language, images, 3D assets, and camera trajectories. ChatCam has the potential to simplify camera movements and reduce technical barriers for creators.

Acknowledgements

This work was supported in part by Dartmouth College A&S Startup fund and by the Research Grant Council of the Hong Kong SAR under Theme-based Research Scheme, grant no. T22-606/23R.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O’Toole, and Changil Kim. HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [3] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139, 2022.
- [4] Sherwin Bahmani, Xian Liu, Yifan Wang, Ivan Skorokhodov, Victor Rong, Ziwei Liu, Xihui Liu, Jeong Joon Park, Sergey Tulyakov, Gordon Wetzstein, et al. Tc4d: Trajectory-conditioned text-to-4d generation. *arXiv preprint arXiv:2403.17920*, 2024.
- [5] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021.
- [6] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022.
- [7] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19697–19705, 2023.
- [8] Blain Brown. *Cinematography: theory and practice: image making for cinematographers and directors*. Routledge, 2016.
- [9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.
- [10] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 130–141, 2023.
- [11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021.
- [12] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs, 2023.
- [13] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, pages 333–350. Springer, 2022.
- [14] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision (ECCV)*, pages 202–221. Springer, 2020.
- [15] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [16] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3193–3203, 2021.

- [17] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.
- [18] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.
- [19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [20] Zhiwen Fan, Peihao Wang, Xinyu Gong, Yifan Jiang, Dejia Xu, and Zhangyang Wang. Nerf-sos: Any-view self-supervised object segmentation from complex real-world scenes. *International Conference on Learning Representations (ICLR)*, pages arXiv–2209, 2023.
- [21] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large spatial model: End-to-end unposed images to semantic 3d. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [22] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12479–12488, 2023.
- [23] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5501–5510, 2022.
- [24] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5501–5510, 2022.
- [25] Rohit Girdhar, Alaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190, 2023.
- [26] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and P.J. Narayanan. Interactive Segmentation of Radiance Fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [27] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 976–980. IEEE, 2022.
- [28] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.
- [29] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *arXiv*, 2023.
- [30] Yining Hong, Zishuo Zheng, Peihao Chen, Yian Wang, Junyan Li, and Chuang Gan. Multiply: A multisensory object-centric embodied large language model in 3d world, 2024.
- [31] Benran Hu, Junkai Huang, Yichen Liu, Yu-Wing Tai, and Chi-Keung Tang. Nerf-rpn: A general framework for object detection in nerfs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [32] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3d segmentation on point clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [33] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.
- [34] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12949–12958, 2021.
- [35] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

- [36] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4), 2023.
- [37] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerp: Language embedded radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
- [39] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [40] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:23311–23330, 2022.
- [41] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations (ICLR)*, 2022.
- [42] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, pages 19730–19742. PMLR, 2023.
- [43] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning (ICML)*, pages 12888–12900. PMLR, 2022.
- [44] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4521–4530, 2019.
- [45] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
- [46] Xinhang Liu, Jiaben Chen, Huai Yu, Yu-Wing Tai, and Chi-Keung Tang. Unsupervised multi-view object segmentation using radiance field propagation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:17730–17743, 2022.
- [47] Xinhang Liu, Shiu-hong Kao, Jiaben Chen, Yu-Wing Tai, and Chi-Keung Tang. Deceptive-nerf: Enhancing nerf reconstruction using pseudo-observations from diffusion models. *arXiv preprint arXiv:2305.15171*, 2023.
- [48] Yichen Liu, Benran Hu, Junkai Huang, Yu-Wing Tai, and Chi-Keung Tang. Instance neural radiance field. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [49] Yichen Liu, Benran Hu, Chi-Keung Tang, and Yu-Wing Tai. Sanerf-hq: Segment anything for nerf in high quality. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [50] Chongshan Lu, Fukun Yin, Xin Chen, Wen Liu, Tao Chen, Gang Yu, and Jiayuan Fan. A large-scale outdoor multi-modal dataset and benchmark for novel view synthesis and implicit scene reconstruction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7557–7567, 2023.
- [51] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *International Conference on Learning Representations (ICLR)*, 2022.
- [52] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.
- [53] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshstein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20669–20679, 2023.
- [54] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022.
- [55] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5865–5874, 2021.

- [56] 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 (NeurIPS)*, 32, 2019.
- [57] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–824, 2023.
- [58] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10318–10327, 2021.
- [59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of Machine Learning Research (PMLR)*, volume 139, pages 8748–8763, 2021.
- [60] Zhongzheng Ren, Aseem Agarwala, Bryan Russell, Alexander G. Schwing, and Oliver Wang. Neural volumetric object selection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [61] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10912–10922, 2021.
- [62] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.
- [63] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.
- [64] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8579–8590, 2023.
- [65] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kotschieder. Panoptic lifting for 3d scene understanding with neural fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9043–9052, June 2023.
- [66] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5459–5469, 2022.
- [67] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. Point scene understanding via disentangled instance mesh reconstruction. *European Conference on Computer Vision (ECCV)*, 2022.
- [68] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- [69] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, volume 41, pages 703–735. Wiley Online Library, 2022.
- [70] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [71] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12959–12970, 2021.
- [72] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

- [73] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3835–3844, 2022.
- [74] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13524–13534, 2022.
- [75] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. *arXiv preprint arXiv:2312.03641*, 2023.
- [76] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [77] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *CoRR*, abs/2309.05519, 2023.
- [78] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, 2021.
- [79] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds, 2019.
- [80] Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*, 2023.
- [81] Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F. Fouhey, and Joyce Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent, 2023.
- [82] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. *arXiv preprint arXiv:2402.03162*, 2024.
- [83] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5752–5761, 2021.
- [84] Jiakai Zhang, Xinhang Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. Editable free-viewpoint video using a layered neural representation. *ACM Transactions on Graphics (TOG)*, 40(4):1–18, 2021.
- [85] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. *arXiv preprint arXiv:2301.06052*, 2023.
- [86] Juntao Zhang, Yuehuai Liu, Yu-Wing Tai, and Chi-Keung Tang. C3net: Compound conditioned controlnet for multimodal content generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [87] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *International Conference on Machine Learning (ICML)*, 2024.
- [88] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15838–15847, 2021.
- [89] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Supplementary Materials for *ChatCam: Empowering Camera Control through Conversational AI*

This supplementary document presents additional qualitative results and discusses the limitations and societal impacts of our proposed approach.

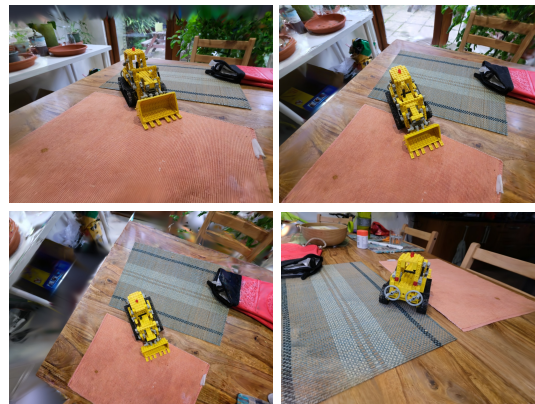
A Video

For better visualization of our reconstruction results, we create a set of video visualizations. We highly recommend to watch *supplementary_video.mp4* for more results.

Move the camera over the arm of the man with his arm raised, then pan to the guy and lady taking a selfie together.



Leap from the front of bulldozer to the back.

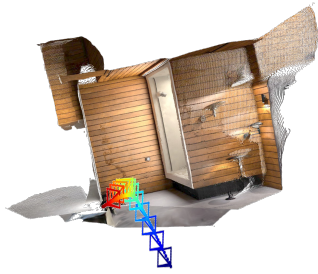


Quickly move back and forth in front of the plants and bikes.



Figure A: Additional qualitative results. (1)

Can you try sitting on one of the little stools?



Start from capturing everybody, then move closer to the man wearing the red shirt.



Facing the piano, pull the camera back, then glance over at the TV on the left, and back to the piano.



Figure B: Additional qualitative results. (2)

B Additional Results

We present additional qualitative results in Figure A and Figure B.

C Limitations

As the first AI-assisted system for language-guided camera operation, our method relies on LLMs as agents, and therefore its efficiency depends on LLMs. With the rapid development of the community, this limitation may be alleviated.

Our current results are limited to static scenes due to the limited availability of high-quality 4D Dynamic NeRF/3DGS data. Extending our approach to a dynamic scene would be straightforward by introducing a timestamp in the Anchor Determinator. We leave this as one of our future work.

D Societal Impacts

Our approach has great potential to help creators in industries such as television, movies, games, etc. reduce their burden by simplifying the learning costs of utilizing 3D assets like radiance field reconstructions. This allows content creators to focus on their creations. We must also admit that as existing 3D assets become more and more abundant, it is inevitable that there will be harmful content in them, and our method may contribute to the creation of harmful content. We encourage the community to play wisely with ChatCam.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist".**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction state the contributions made in the paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitations in appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We fully describe our proposed pipeline and core building components.

Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release the data and code upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We include implementation details for our experiments, to a level of detail that is necessary to appreciate the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We describe in detail how we obtain quantitative metrics in our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We indicate the type and number of GPUs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: We conform with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss both potential positive societal impacts and negative societal impacts of the work performed in appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: We credit all such assets by appropriate citations and statements.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[NA\]](#)

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[Yes\]](#)

Justification: Our paper includes details about its user study.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not pose such risks.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.