3D-SPATIAL MULTIMODAL MEMORY

Xueyan Zou[1◇], **Yuchen Song**[1◇], **Ri-Zhao Qiu**[1◇], **Xuanbin Peng**[1◇]
Jianglong Ye[1], **Sifei Liu**[2], **Xiaolong Wang**[1,2]
[◇]Core Contribution
[1]UC San Diego [2]NVIDIA

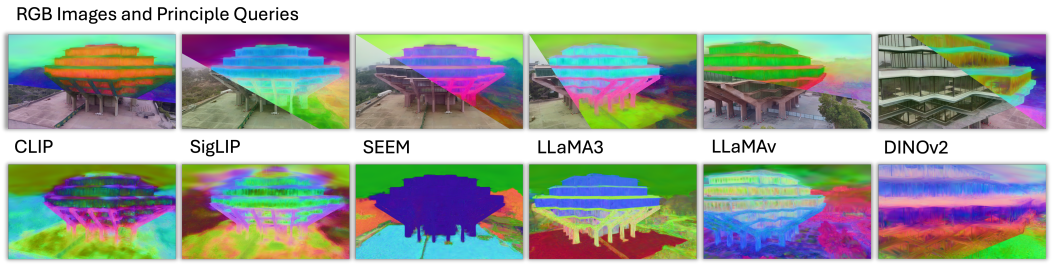

Figure 1: Our proposed MultiModal Memory integrates Gaussian splatting with foundation models to efficiently store multimodal memory in a Gaussian structure. The feature maps rendered by our approach exhibit high fidelity, preserving the strong expressive capabilities of the foundation models.

ABSTRACT

We present 3D Spatial MultiModal Memory (**M3**), a multimodal memory system designed to retain information about medium-sized static scenes through video sources for visual perception. By integrating 3D Gaussian Splatting techniques with foundation models, **M3** builds a multimodal memory capable of rendering feature representations across granularities, encompassing a wide range of knowledge. In our exploration, we identify two key challenges in previous works on feature splatting: (1) computational constraints in storing high-dimensional features for each Gaussian primitive, and (2) misalignment or information loss between distilled features and foundation model features. To address these challenges, we propose **M3** with key components of principal scene components and Gaussian memory attention, enabling efficient training and inference. To validate **M3**, we conduct comprehensive quantitative evaluations of feature similarity and downstream tasks, as well as qualitative visualizations to highlight the pixel trace of Gaussian memory attention. Our approach encompasses a diverse range of foundation models, including vision-language models (VLMs), perception models, and large multimodal and language models (LMMs/LLMs). Furthermore, to demonstrate real-world applicability, we deploy **M3**'s feature field in indoor scenes on a quadruped robot. Notably, we claim that **M3** is the first work to address the core compression challenges in 3D feature distillation.

1 INTRODUCTION

Human perception encompasses the world across spatial dimensions. When encountering static elements in the visual environment, individuals tend to organize and store knowledge progressively, starting from a coarse overview and refining it into finer details. For instance, we can recall our daily surroundings with varying levels of detail, ranging from a high-level layout to specific part-level features. However, for larger-scale environments, our understanding tends to remain more coarse and generalized. Previous works, such as NeRF [23] and 3DGS [16], have demonstrated the ability to store scene-level information at the pixel level for intermediate-scale scenes. However, these models lack the capability to retain the semantic understanding of the scene like humans.

In this study, we aim to develop a spatial memory system for static scenes, capable of processing static scene video clips spanning spatial horizons. The primary objective is to store all human-

processable information in a format that is precise, efficient, and amenable to future interactive queries. Our approach leverages 3D Gaussian splatting techniques and incorporates features extracted from foundation models to construct scenes imbued with semantic knowledge. The selection of Gaussian splatting as our structural format was motivated by two key considerations: First, the need to address video redundancy through efficient compression, and second, the requirement for multi-granular information representation. Gaussian splatting inherently provides a framework for representing the smallest units of information as Gaussian primitives as well as naturally eliminating the spatial redundancy, aligning well with the motivations.

Previous feature splatting works such as F-3DGS [51] and F-Splat [26] directly distill 2D feature maps obtained from foundation models into 3D Gaussians via differentiable rendering. We observe two key issues: First, due to the computational limitations, the feature vector dimensions in Gaussian primitives are significantly reduced compared to the original 2D feature maps (typically 16-64 versus 1024), potentially causing an information bottleneck. Second, while the original feature maps may not be inherently 3D-consistent, enforcing 3D consistency in the Gaussians can cause misalignment between the original and distilled features. Consequently, the distilled feature may not accurately capture the knowledge embedded in the foundation model.

To address these issues, we present MultiModal Memory (**M3**), a better integration of Gaussian splatting and multimodal foundation models that efficiently store expressive multimodal memory in a Gaussian structure, facilitating spatial queries. Specifically, we propose to store the original high-dimensional 2D feature maps in a memory bank called principal scene components and use the low-dimensional principal queries from 3D Gaussians as indices. Instead of directly distilling the 2D features into 3D embeddings, we apply Gaussian memory attention between the principal scene components and principal queries to render the foundation model embeddings in a 3D scene.

In this way, we combine the best of both foundation models and Gaussian splatting: preserving the high expressive ability of the original foundation model feature maps while maintaining a 3D-consistent, low-dimensional Gaussian structure of the scene. Furthermore, we also design a heuristic algorithm to minimize redundancy in the memory bank by reducing the raw features from the video stream. These reduced features are referred to as Principal Scene Components. Example feature maps rendered by **M3** are visualized in Fig. 1.

To evaluate **M3**, we employ a diverse set of foundation models, including vision-language models, LMM/LLMs, and perception models. We adopt both low-level metrics (e.g. PSNR) to assess the model's feature memorization capability and high-level metrics (e.g. mIoU, IR, TR) to assess its performance on downstream perception tasks. Extensive experiments demonstrate that **M3** outperforms previous works in both memorization and downstream tasks while maintaining low computational costs. Lastly, we deploy **M3** on a quadruped robot platform for grasping, showcasing its potential for real-world generalization from single-scene, multi-scene, and long-horizon tasks.

2 Related Work

Foundation Models. The field of multimodal learning has seen remarkable progress, leading to the development of diverse foundation models. In the vision-language domain, models such as CLIP [28], Florence [40; 45], and the recent SigLIP [47] employ ViT-style [5] transformer architectures to align visual and linguistic representations. For vision-specific tasks, SAM [17; 29] excels in part-level clustering, while DINO [4; 24] advances self-supervised representation learning. In document understanding, LayoutLM [42; 41; 11] combines OCR and text classification for comprehensive document analysis. The language domain has seen significant advancements in reasoning capabilities, exemplified by the LLaMA [36; 37; 6] and Mistral [14; 15] series. While these works have pushed language reasoning to new heights, recent studies like [31; 35; 7] explore mixture-of-experts approaches to enhance visual representation learning in foundation models. These developments, along with advanced models such as ChatGPT [1] and Claude [2], form the backbone of modern Multimodal Large Language Models (MLLMs), paving the way for more sophisticated AI systems.

3D Gaussians and Feature Field. NeRF [23] revolutionized 3D scene representation, but its implicit nature caused slow rendering and training. 3D Gaussian Splatting [16] (3DGS) emerged as a faster, more explicit alternative, enabling rapid training and real-time rendering. Since then, 3DGS has been enhanced: H3DGS [44] improved large-scale rendering, Mip-Splatting tackled

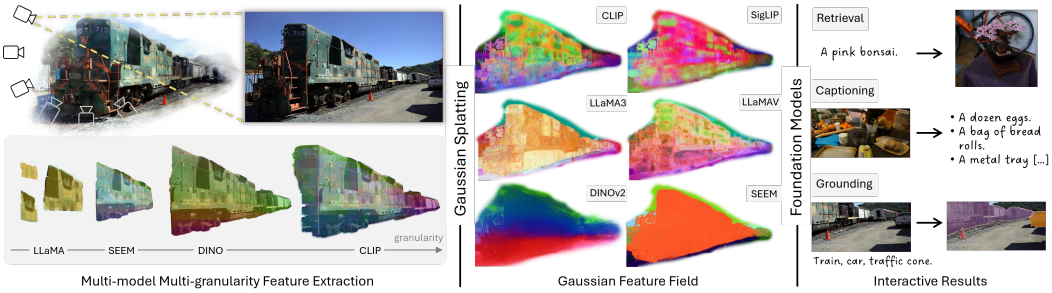

Figure 2: A scene (**V**) is composed of both structure (**S**) and knowledge (**I**). To model these, we leverage multiple foundation models to extract multi-granularity scene knowledge, and employ 3D Gaussian splatting to represent the spatial structure. By combining these techniques, we construct a spatial multimodal memory (**M3**), which enables downstream applications such as retrieval, captioning and grounding.

anti-aliasing for high detail, and WildGaussians [19] addressed occlusion and appearance changes. Grendel-GS [50] enabled multi-GPU training for efficiency with larger datasets. Researchers began incorporating 3D feature fields into neural rendering pipelines, moving from NeRF-based models like F3RM [32] to 3DGS-based ones. Feature 3DGS [51] added feature representations to 3DGS, leading to advancements like Feature Splatting [26; 13] for language-driven scene synthesis and LEGaussians [33] for open-vocabulary scene understanding. LiveScene [27] introduced interactive radiance fields, while recent work [46] focuses on improving 2D features with 3D-aware fine-tuning for better 2D-3D integration.

Scene Graph and Video Memory. Long-horizon scene understanding encompasses both spatial and temporal dimensions. For spatial modeling, scene graphs have been prominent: ConceptFusion [12] introduced open-set multimodal 3D mapping, ConceptGraphs [8] extended this to open-vocabulary 3D scene graphs, and Hierarchical Open-Vocabulary 3D Scene Graphs [39] applied these concepts to language-grounded navigation. Beyond Bare Queries [21] and Open Scene Graphs [22] further demonstrated their utility in object retrieval and navigation. However, these approaches often rely on heuristic edge/node construction and lack direct LMM integration via embeddings. For temporal aspects, previous works have focused on using memory bank embeddings to store information across frames. For instance, MA-LMM [9], MovieChat [34], and Hierarchical Memory [38] introduced various memory augmentation techniques for video understanding. Flash-VStream [48] and Streaming Long Video Understanding [25] concentrated on real-time processing of long video streams. While these temporal methods integrate better with LMMs, they face challenges such as image over-compression (representing an entire frame with a single embedding), frame redundancy (adjacent frames containing overlapping spatial information), and lack of explicit spatial information. Our 3D Gaussian approach bridges this gap, combining spatial precision with temporal flexibility and LMM compatibility.

3 METHOD

3.1 3D-SPATIAL MULTIMODAL MEMORY (M3) PIPELINE.

A real-world visual perception scene (**V**) consists of both structure (**S**) and knowledge (**I**). The structure of Visual Granularity ($\mathcal{VG}$) can range from the fine details such as leaf shapes, to large-scale elements, such as city layouts. Concurrently, the Knowledge Space ($\mathcal{KS}$) spans scales from specific information, such as leaf species (e.g. a red maple leaf) to a comprehensive interpretation (e.g. The space needle in Seattle...) of a view ($\mathbf{V}_*$). Gaussian splatting serves as a framework for constructing scene structure with finest granularity, represented as gaussian primitives, while foundation models provide vast world knowledge spanning various scales for scene knowledge. The organic integration of Gaussian splatting and Foundation Models infuses scene structure with multi-granularity knowledge, enabling the construction of a full-stack Multimodal Memory of the scene with precise spatial information. To maintain efficiency while preserving the global representation of foundation model features, we compress the extracted features from foundation models into principal scene components (**PSC**) for each scene and learn to probe the scene via Gaussian Splatting

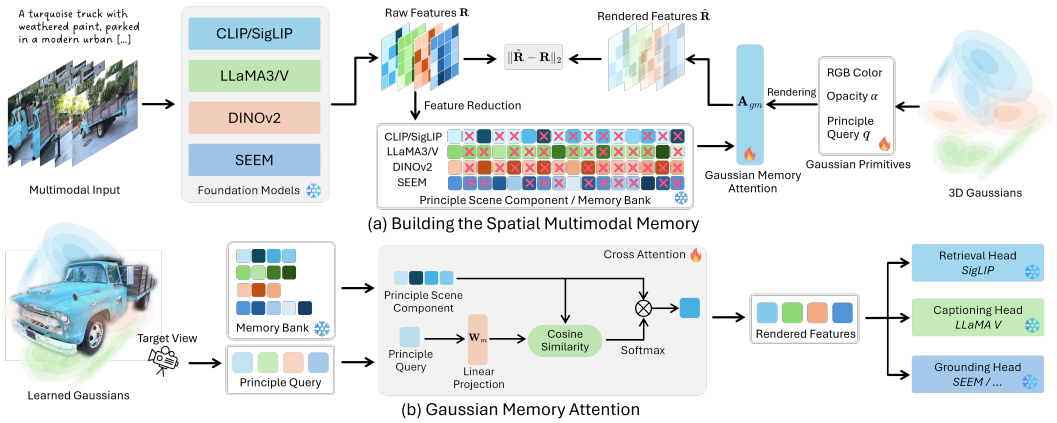

Figure 3: Given a video sequence, we utilize foundation models ($\mathbf{F}$) to extract raw features ($\mathbf{R}$). These features are reduced using Algorithm 1, producing principal scene components ($\mathbf{PSC}$), which are stored in a memory bank. We introduce optimizable attribute queries (q) to Gaussian primitives, and apply a Gaussian Memory Attention ($\mathbf{A}_{gm}$) mechanism to produce the final rendered features ($\hat{\mathbf{R}}$), which can be linked back to various heads of the foundation models.

parameters, denoted as principle query ($\mathbf{Q}_p$). Ultimately, leveraging the rendering capabilities of Gaussian Splatting, we can dynamically populate the GS structure with multi-granularity information spanning the entire view of the scene. Our pipeline is illustrated in Fig. 2.

3.2 M3 Preliminaries.

Visual Granularity ($\mathcal{VG}$). Visual granularity ($\mathcal{VG}$) typically represents the clustering pixel scope of an image, a concept introduced in Semantic-SAM [20]. Given a view $\mathbf{V}_* \in \mathbb{R}^{h \times w, 3}$ (h, w denote the pixel dimensions) in the scene $\mathbf{V}$, it is composed of multi-granularity segments ranging from individual pixels to the full view (as illustrated in the left part of Fig. 2), represented by $\mathbf{V}_* = \{V_*^1, V_*^2, ..., V_*^m\}$, where $V_*^i \in \mathbb{R}^{p,3}$ is the ith granularity of the view $\mathbf{V}_*$, p is the number of pixels, and m denotes the total number of granularities. This multi-granularity approach is introduced because humans naturally possess multi-granularity recognition of the world for various utilities.

Knowledge Space ($\mathcal{KS}$). Different foundation models ($\mathbf{F}$) focus on various aspects of knowledge. For instance, CLIP [28] and SigLIP [47] concentrate on image-level perception, while Semantic-SAM [20] emphasizes part-level visual grouping. In contrast, LLaMA3/v [6] incorporates both local and global attention mechanisms. The features generated by these models occupy different knowledge spaces $\mathbf{F}(\mathbf{V}_*) \in \{\mathcal{KS}^1, \mathcal{KS}^2, ..., \mathcal{KS}^c\}$ where c is the total number of knowledge spaces here, emphasizing diverse aspects such as visual alignment ($\mathcal{KS}^1$), semantics ($\mathcal{KS}^2$), reasoning ($\mathcal{KS}^3$), and etc.

Principle Scene Components ($\mathbf{PSC}$) and Principle Query ($\mathbf{Q}_p$). We extract foundation model features for each view, denoted as $\mathbf{F}_*(\mathbf{V}) = \{\mathbf{E}_1^*, \mathbf{E}_2^*, ...\mathbf{E}_n^*\}$ for each foundation model ($\mathbf{F}_*$) and scene ($\mathbf{V}$), where n is the number of views. These foundation model features are represented as $\mathbf{E}_i^* \in \mathbb{R}^{[h \times w, d]}$, where h, w denote the feature pixel dimensions. However, different views often contain redundant and similar features. We define the key features that construct the scene as Principle Scene Components ($\mathbf{PSC}$), drawing inspiration from the terminology of Principal Component Analysis. The attribute within Gaussian representation responsible for indexing $\mathbf{PSC}$ is denoted as Principle Query ($\mathbf{Q}_p$), which is learnable parameters in each Gaussian primitive.

3.3 Spatial Multimodal Memory

Build Scene Structure via 3D Gaussians. We formally define the input of **M3** as a video sequence with frames, where each frame corresponds to a view $\mathbf{V}_*$. 3D Gaussian splatting [16] is employed to fit the scene, with each view rendered by the Gaussian rasterizer. For each Gaussian primitive, the optimizable attributes include the centroid ($x \in \mathbb{R}^3$), rotation quaternion ($r \in \mathbb{R}^4$) storing the rotation and scaling matrix, opacity value ($\alpha \in \mathbb{R}^3$), and spherical harmonics ($sh \in \mathbb{R}^3$). To model the Principle Scene Components ($\mathbf{PSC}$), we introduce an additional optimizable attribute: principle

Algorithm 1 Raw Feature ($\mathbf{R}$) Similarity Reduction Algorithm

Input : $\mathbf{R} \in \mathbb{R}^{[n \times h \times w, d]}$ (Raw Features), $\theta \in (0, 1]$ (threshold), $c \in \mathbb{N}$ (chunk size)
Output: $\mathbf{PSC} \subseteq \mathbf{R}$ (Principle Scene Components)

```
SimilarityReduction(R, θ, c)  n ← |R| # Number of raw features
R̂ ← { e_i / ||e_i||_2 : e_i ∈ R } # Normalize raw features
I ← ∅ # Set of filtered indices
U ← {0}^n # Usage mask, initially all false
for k ← 0 to ⌊n/c⌋ − 1:
    C_k ← { ê_i : i ∈ [kc, (k+1)c) ∩ ℕ } # Current chunk
    S_k ← C_k · R̂^⊤ # Similarity matrix for chunk
    for j ← 0 to |C_k| − 1:
        if U_{kc+j} = 0:
            J ← { i : S_{k,j,i} ≥ θ } # Similar indices
            if ∀i ∈ J : U_i = 0:
                I ← I ∪ {kc + j} # Select Principle Components
                ∀i ∈ J : U_i ← 1
PSC ← { e_i : i ∈ I }  return PSC
```

queries ($q \in \mathbb{R}^l$) with flexible dimensionality to accommodate various foundation models. Each foundation model utilizes s degrees from the $\mathbf{Q}_p \in \mathbb{R}^l$. These degrees are rendered alongside Gaussian parameters to produce view-based principle queries $\mathbf{Q}_p^{\mathbf{V}_*}$ with shape $[H, W, l]$. Following [51], the colors and principle queries are rendered as:

$$C = \sum_{i \in N} c_i \alpha_i T_i, \quad \mathbf{Q}_p = \sum_{i \in N} q_i \alpha_i T_i, \quad \text{where } T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{1}$$

Here, N represents the set of sorted Gaussians overlapping with the given pixel, and T_i denotes the transmittance, defined as the product of opacity values of previous Gaussians overlapping the same pixel.

Extract Multi-Granularity Scene Knowledge. Upon preparing the attributes in the Gaussian primitives, we extract multi-granularity scene knowledge via foundation models. Different foundation models focus on different aspects of knowledge projection and granularity, as illustrated in Fig. 4. In this paper, we employ a set of foundation models $\mathbf{F} = \{\texttt{CLIP}, \texttt{SigLIP}, \texttt{DINOv2}, \texttt{LLaMA3}, \texttt{LLaMAv}, \texttt{SEEM}\}$, where LLaMAv is the vision-instruct version of LLaMA3. For each view, we extract foundation model embeddings, formally expressed as $\mathbf{F}(\mathbf{V}_*) = \mathbf{E} \in \mathbb{R}^{[h \times w, d]}$.

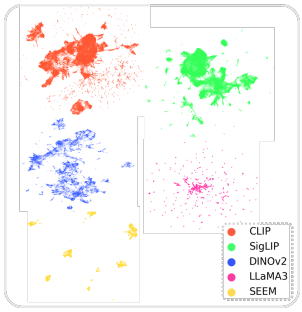

Figure 4: The UMAP visualization of model embedding manifolds reveals distinct shapes, reflecting different focus.

We implement specific algorithms for projecting LLaMA3 language embeddings and SEEM [52] visual prompts into pixel-level features. For LLaMA3, we first use SoM [43] and Semantic-SAM to extract language descriptions for each region. The language prompt of each region is represented as $\mathbf{T} \in \mathbb{R}^{[l_1, d]}$, where l_1 is the number of regions extracted by Semantic-SAM. For SEEM, we utilize visual prompts corresponding to each region, with visual prompts for each image represented as $\mathbf{O} \in \mathbb{R}^{[l_2, d]}$, where l_2 is the number of regions segmented by SEEM. We then splat the features to the pixel level, duplicating the prompts within each mask region, resulting in $\mathbf{T}$ and $\mathbf{O}$ being indexed to the dimension of $\mathbb{R}^{[h \times w, d]}$.

After feature extraction, we obtain raw features ($\mathbf{R} \in \mathbb{R}^{[n, h \times w, d]}$) for the full Scene with n views within each foundation model. These raw features span various granularities and knowledge spaces, providing a comprehensive multimodal (vision and language) understanding of the scene. In correlation to 3D Gaussian Splatting, the smallest granularity component is at the pixel level, with the most low-level knowledge projection being the RGB color value.

Compress Scene Knowledge to Memory. While the scene knowledge is extracted from foundation models $\mathbf{F}$ into raw feature space $\mathbf{R} \in \mathbb{R}^{[n, h \times w, d]}$, the dimensionality is too high for storage and rendering in each scene. Previous works such as F-3DGS [51] and F-Splat [26] address this issue through feature distillation. However, we observe two major problems with feature distillation: (a) The distilled feature experiences information loss compared to the original feature due to feature

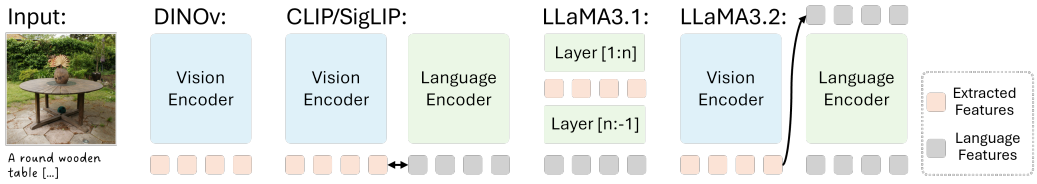

Figure 5: Illustration of patch-level visual embedding extraction their applications.

compression (usually 16 to 64 dims sampling linear projection to the original dimension d that is usually 1000 dims or more). (b) The upsampled feature may have misalignments with the original knowledge space of the foundation model, making it difficult to be decoded by the original $\mathbf{F}$. To resolve these issues, we first flatten the raw features ($\mathbf{R}$) into $\mathbb{R}^{[n\times h\times w,d]}$, and then perform similarity reduction on the first dimension using Algorithm 1. The reduced raw feature represents the principal scene component ($\mathbf{PSC}$), also named as memory bank, serving as the essential representation of the scene. The memory bank or $\mathbf{PSC}$ has dimensionality $\mathbb{R}^{[t,d]}$, where t depends on the similarity threshold we set. The reduction is effective due to the presence of many duplicated features in neighboring spatial pixels within a view, or duplicated regions across views. We visualize the memory bank building process in Fig.3 a.

Gaussian Memory Attention. Given view-based principle queries $\mathbf{Q}_p^{\mathbf{V}_*} \in \mathbb{R}^{[H,W,n]}$ that is rasterized by gaussian primitives, and principle scene components $\mathbf{PSC} \in \mathbb{R}^{[l,d]}$, we perform Gaussian Memory Attention ($\mathbf{A}_{gm}$) to obtain the rendered feature in alignment with foundation models. With learnable random initialized memory projection $\mathbf{W}_m \in \mathbb{R}^{[n,d]}$, we formally define the Gaussian Memory Attention as follows:

$$\hat{\mathbf{R}} = \mathbf{A}_{gm}(\mathbf{Q}_p^{\mathbf{V}_*}) = \texttt{Softmax}(\mathbf{Q}_p^{\mathbf{V}_*} \times \mathbf{W}_m \times \mathbf{PSC}^T) \times \mathbf{PSC}. \tag{2}$$

This Gaussian memory attention links the $\mathbf{Q}_p$ with $\mathbf{PSC}$ and projects it into the corresponding foundation model knowledge space. The attention process is depicted in Fig. 3 b.

Scene Rendering and Deployment. Given the rendered features $\hat{\mathbf{R}}$ for each foundation model aligning with the corresponding foundation model space, we can link back to the powerful functions of foundation models. We expect that for models like `CLIP, SigLIP, SEEM`, the rendered feature can be directly used for vision language-based tasks such as retrieval and grounding. For generative-based models like `LLaMA3, LLaMAv`, we anticipate that the feature can be directly used in captioning or simple visual question answering tasks. Formally, we express this as $\mathcal{X} = \mathbf{F}_{\texttt{dec}}(\hat{\mathbf{R}})$.

4 Experiments

4.1 Experimental Setup

Datasets. To support extensive quantitative and qualitative evaluation, we perform experiments using several existing scene datasets [3; 18; 10] and collected a custom robot dataset (**M3**-Robot) using a quadruped robot and a drone. Specifically, we use *Garden* (an outdoor scene) from Mip-NeRF360 [3], *Train* from the Tank & Temples dataset [18], and *PlayRoom* as well as *DrJohnson* from the Deepblending dataset [10]. For the **M3**-Robot dataset, we collect images using two mobile robots. The *Table-Top* sequence is collected from a RealSense 405D camera mounted on the end effector of a Unitree Z1 robot arm on a Unitree B1 quadruped robot, where a human operator tele-operates the robot with a remote to obtain centripetal views of tabletop objects. Images in the *Geisel* sequence are collected by a tele-operated DJI Mini4-Pro drone. The collected images are processed by COLMAP [30] to obtain camera parameters and initialization.

Memory across multiple Foundation Models. The multi-modal memory mechanism allows **M3** to retain knowledge from many models, which differs from existing distillation-based methods that only distill a few (2-3) models. Specifically, as provided in Sec. 3.3, we employ 6 foundation models to resemble human memory of different aspects. Each model has different granularity and focus of different semantics: image-level vision-language understanding via `CLIP` [28] as well as `SigLIP` [47]; pixel-level semantic understanding via `SEEM` [52]; self-supervised structural feature via `DINOv2` [24]; and LLaMA3.1/3.2v [6] for multi-modal understanding and reasoning.

Dataset	Method	Time	DINOv2		CLIP		SigLIP		SEEM		LLaMA3		LLaMAv	
		Min ↓	Cosine↓	L2↓	Cosine↓	L2↓	Cosine↓	L2↓	Cosine↓	L2↓	Cosine↓	L2↓	Cosine↓	L2↓
Train	F-3DGS [51]	∼480	0.3753	0.9989	0.3310	0.1532	0.3828	0.0003	0.1035	0.0979	0.5080	0.0145	0.6287	46.37
	F-Splat [26]	∼60	0.6779	1.975	0.5937	0.5060	0.6179	0.0867	0.4269	11.57	0.7041	1.5637	0.7030	56.09
	M3	∼60	0.5321	1.681	0.3140	0.2800	0.2811	0.0004	0.1389	0.2251	0.4401	0.0253	0.7069	53.43
Garden	F-3DGS [51]	∼600	0.3270	0.8386	0.2737	0.1209	0.3967	0.0003	0.1203	0.1570	0.4607	0.0156	0.5715	47.5502
	F-Splat [26]	∼60	0.7347	1.9616	0.7002	1.6942	0.7147	0.1573	0.4224	9.4654	0.6390	2.2054	0.7511	60.8692
	M3	∼60	0.5701	1.7279	0.3168	0.2876	0.2927	0.0004	0.1839	0.3469	0.3387	0.0217	0.7235	58.0492
Drjohnson	F-3DGS [51]	∼1200	0.1738	0.2108	0.3894	0.1726	0.5505	0.0004	0.1738	0.2108	-	-	0.7065	48.39
	F-Splat [26]	∼60	0.8069	2.0250	0.6673	0.7537	0.6774	0.1436	0.4650	10.4111	0.6387	1.9263	0.8184	54.82
	M3	∼60	0.5878	1.7553	0.3435	0.2924	0.2975	0.0004	0.2456	0.4179	0.3175	0.0226	0.7224	52.68
Playroom	F-3DGS [51]	∼480	0.5728	1.4222	0.4407	0.1969	0.6684	0.0004	0.2063	0.2620	0.4521	0.0170	0.7222	53.6363
	F-Splat [26]	∼60	0.7956	1.9640	0.6458	0.7808	0.6737	0.1266	0.4745	10.8735	0.4538	0.1764	0.8185	59.4252
	M3	∼60	0.7290	2.0391	0.4046	0.3307	0.3912	0.0006	0.4602	0.8904	0.4798	0.0293	0.7471	54.44

Table 1: Feature Distance in comparison with distillation methods that use similar or higher budgets across datasets and foundation models.

Dataset	Method	#Param	CLIP				SigLIP					
			mIoU	cIoU	AP50	AP60	I2T@1	I2T@5	I2T10	T2I@1	T2I@5	T2I10
Train	**Ground Truth**	-	25.3	26.3	14.7	3.3	81.5	97.3	100.0	71.0	89.4	92.1
	F-3DGS [51]	61M	24.2	24.3	16.3	7.1	2.6	13.2	28.9	0.0	2.6	18.4
	M3	**35M**	**25.4**	**26.5**	**19.6**	**12.5**	**55.2**	**84.2**	**92.1**	**52.6**	**84.2**	**92.1**
Playroom	**Ground Truth**	-	25.6	24.2	9.6	3.0	96.5	100.0	100.0	62.0	96.5	100.0
	F-3DGS [51]	61M	23.8	21.4	11.9	3.0	79.3	96.6	96.6	31.0	79.3	89.7
	M3	**35M**	23.1	**23.1**	**11.9**	**5.9**	72.4	**96.6**	**100.0**	**41.3**	65.5	68.9
Geisel	**Ground Truth**	-	19.5	21.4	5.3	0.0	100.0	100.0	100.0	60.0	85.7	91.4
	F-3DGS [51]	61M	19.0	20.4	14.1	1.2	45.7	94.3	100.0	0.0	20.0	34.3
	M3	**35M**	**21.8**	**23.5**	**16.5**	**11.8**	**100.0**	**100.0**	**100.0**	**71.4**	**85.7**	**94.2**

Table 2: Feature/RGB metrics for all foundation models and scene.

In Fig. 5, we provide a comprehensive illustration of how we extract features from foundation models. The extracted features are marked in orange in alignment with language representations optionally or continued to be the input of the language Encoder.

Loss Computation. For each input image, we extract patch-level embeddings from the aforementioned models. Previous methods [26; 51] compute the patch-wise distance loss on the rendered features, this not only has a high volume of GPU memory consumption that hinders parallel training for all the foundation models but also creates artifacts when downsampling the feature. In compensate, we use point-based loss, where we sample 2000 points ranging from both predict and ground truth features for distance loss computation. This largely reduces the computation overhead for training, as shown in Table. 1.

Low-level Evaluation Metrics. To systematically evaluate multi-modal memory, we use evaluation metrics ranging from low/pixel-level to high-level downstream tasks. In particular, the low-level evaluation metrics evaluate pixel-level image quality. For rendered image quality on evaluation views (views not provided in training), we use common metrics (PSNR, SSIM, and LPIPS [49]) as Kerbl et al. [16]. For feature quality, we report cosine and L2 distance.

High-level Evaluation Metrics. High-level evaluation metrics, different from low-level ones, focus on evaluating downstream tasks of features. For discriminative models [28; 47; 52], we will report commonly used metrics such as mIoU (mean Intersection over Union), cIoU (complete Intersection over Union), and AP (Average Precision). For retrieval, we will use IR@1 (Information Retrieval at rank 1) and TR@1 (Text Retrieval at rank 1).

4.2 Quantitative Results

Baseline Implementation For quantitative experiments, we compare **M3** with two recent distillation-based feature GS methods [26; 51]. For fair comparisons, we train all the methods in approximately 30,000 iterations (29,993 iterations for **M3** due to last-batch data loader roundoffs). The reference training features are identical for different methods. For distillation-based methods, we follow F-Splat [26] to render a latent feature and then decode the latent features to the embedding space of reference features with a multi-head MLP. For all methods, the optimization of both latent features/memory and decoders is trained from scratch for each scene.

Low-Level Results. We report the main quantitative results in Tab. 1, where the average training time and the auxiliary low-level metrics are reported. Our method, **M3**, outperforms F-Splat while reducing significantly compute than F-3DGS. `SEEM` and `LLaMA3` features extraction failed on F-

Dataset	Method	RGB	Time	CLIP		SigLIP		DINOv2		SEEM		LLaMA3		LLaMAv	
		PSNR↑	min.	Cosine↓	L2↓	Cosine↓	L2↓	Cosine↓	L2↓	Cosine↓	L2↓	Cosine↓	L2↓	Cosine↓	L2↓
Tabletop	+CLIP	21.91	∼6	0.3100	0.2956	-	-	-	-	-	-	-	-	-	-
	+SigLIP	21.84	∼10	0.3100	0.2956	0.3122	0.0005	-	-	-	-	-	-	-	
	+DINOv2	21.79	∼15	0.3101	0.2956	0.3123	0.0005	0.5161	1.6057	-	-	-	-	-	-
	+SEEM	21.93	∼20	0.3101	0.2956	0.3123	0.0005	0.5156	1.6048	0.0472	0.1013	-	-	-	-
	+LLaMA3	21.97	∼30	0.3101	0.2956	0.3122	0.0005	0.5160	1.6056	0.0472	0.1012	0.3628	0.0246	-	-
	+LLaMAv (All)	21.96	∼45	0.3100	0.2956	0.3122	0.0005	0.5157	1.6049	0.0472	0.1013	0.3628	0.0246	0.7262	59.92

Table 3: Ablation on the number of foundation models in **M3**.

Degree	# Params	Iteration	CLIP				SigLIP				DINOv2		SEEM		LLaMA3	
			Cosine↓	L2↓	mIoU	AP50	Cosine↓	L2↓	mIoU	AP50	Cosine↓	L2↓	Cosine↓	L2↓	Cosine↓	L2↓
8x6=48	14.8M	30k	0.3256	0.2880	25.4	19.6	0.2913	0.5239	19.4	2.1	0.5755	1.7664	0.1672	0.2749	0.4504	0.0264
		7k	0.3290	0.2900	25.3	14.6	0.2938	0.5277	21.8	4.8	0.5845	1.7835	0.2058	0.3463	0.4517	0.0265
16x6=96	21.5M	30k	0.3140	0.2800	25.7	19.0	0.2866	0.5172	24.3	10.3	0.5535	1.7239	0.1388	0.2247	0.4480	0.0261
		7k	0.3206	0.2842	25.3	20.6	0.2903	0.5227	23.2	8.1	0.5677	1.7513	0.1828	0.3056	0.4504	0.0263
32x6=192	34.8M	30k	0.3043	0.2735	26.7	22.8	0.2814	0.5094	25.7	11.9	0.5318	1.6807	0.0972	0.1553	0.4401	0.0253
		7k	0.3132	0.2792	26.2	21.1	0.2866	0.5172	25.5	11.4	0.5515	1.7198	0.1269	0.2139	0.4436	0.0256
64x6=384	61.4M	30k	0.2917	0.2650	28.4	23.9	0.2721	0.4957	28.5	13.5	0.5099	1.6358	0.0855	0.1321	0.4278	0.0241
		7k	0.3049	0.2734	28.1	23.9	0.2802	0.5079	27.8	13.5	0.5350	1.6870	0.1012	0.1676	0.4348	0.0248

Table 4: Ablation on the dimensions and distillation for each foundation model.

Splat, which we assume was mainly due to the ground truth feature extraction procedure, where duplication was performed to each segmentation to get pixel-level features.

Downstream Results. The downstream evaluation results of grounding and retrieval are shown in Table. 2. The ground truth grounding dataset of Train, Playroom, and Geisel is generated by SoM [] with semantic-SAM for mask labels and GPT4-o [] to generate the caption. Example data are shown in the Appendix. We evaluate all the images in the validation sets of the three datasets. The grounding results clearly show that **M3** is better than F-3DGS with half of the parameters, the gap is non-trivial especially when looking at the AP50/AP60 columns. In addition to grounding results, we also evaluate **M3** on image text retrieval, similar to grounding we use GPT4-o to generate ground truth data for three datasets. The example data are also shown in the appendix. Compared to grounding performance, **M3** performs much better than F-3DGS on retrieval results. For image text retrieval, the positive example is the evaluation image, and the negative pairs are generated by COCO datasets. We believe the large gap in the retrieval results is taken from Gaussian memory attention, where the rendered features are aligned with the original foundation model much better. When we find the correct embeddings in the dataset, this benefit gets enlarged.

Ablation Results. Table. 3 shows the ablation of the number of foundation models involved in **M3**. We gradually added foundation models from simpler single-modal models to more advanced multi-modal models. While maintaining a very efficient training time, our method has independent results from different foundation models. Our implementation is based on Grendel-GS, where the training procedure is efficiently paralleled. In Table. 4, we ablate the computation budget on training **M3** in the balance of memory footprint, training iterations, and performance. The table clearly shows that increasing the number degree will generally improve the performance on all metrics. While having 16 degrees for each foundation model is enough to obtain a reasonable performance. That is what the number is reported in the paper. In addition, increasing training iteration will generally increase the performance, while 1/4 of the training budget (7k) would usually get a reasonable performance.

4.3 Qualitative Results.

M3 consistently demonstrates superior performance across diverse datasets as shown in Fig. 6, by effectively preserving fine-grained details and ensuring smooth, coherent feature representations. The method excels at retaining intricate details such as the textures of chairs and the fine features of books, highlighting its ability to capture micro-level information. This clear layering contributes to rich semantic understanding within the scenes.

Furthermore, **M3** handles overlapping objects exceptionally well, as evident in the *Playroom* dataset, where complex arrangements are rendered with accurate structural information. The outputs from various foundation models are consistently high-quality, each retaining spatial structures and semantic information at different granularities. This demonstrates **M3**'s capability to capture both low-level spatial details and high-level semantic concepts, making it highly effective for tasks that require comprehensive scene understanding.

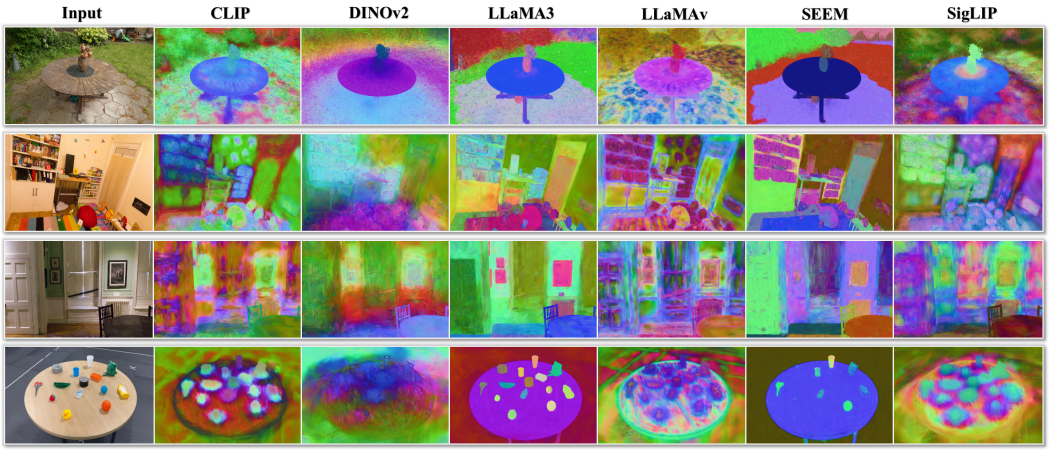

Figure 6: Qualitative results across datasets using **M3**. The figure showcases the consistent performance of the **M3** across various datasets (*Garden*, *Playroom*, *Drjohnson*, *Table-top*).

4.4 Demonstration Results.

We also deployed **M3** on a quadruped robot platform to demonstrate the potential real world applications of our model. In this experiment, we first tele-operate the robot to scan the table by taking a centripetal video with onboard camera. After memorizing the scene with **M3**, the robot is able to locate and grasp any object with text query on decoded CLIP feature. With known robot pose through LiDAR, we are able to render any camera pose ${}_{c_0}T_{c_t}$ with:

$$ {}_{c_0}T_{c_t} = {}_{c_0}T_{e_0} \times {}_{e_0}T_{b_0} \times {}_{b_0}T_{l_0} \times {}_{l_0}T_{w} \times {}_{w}T_{l_t} \times {}_{l_t}T_{b_t} \times {}_{b_t}T_{e_t} \times {}_{e_t}T_{c_t} = {}_{l_0}T_{w} \times {}_{w}T_{l_t}, \quad (3) $$

where c, e, b, l and w refer to `camera`, `end effector`, `arm base`, `lidar`, and `world` respectively. Note that to align with COLMAP coordinates, the camera pose needs modifying with ${}_{COLMAP_0}T_{c_0} \times {}_{c_0}T_{c_t}$.

We tested with the query "yellow bath duck" on the decoded CLIP feature, and as shown in Fig. 7, the rubber duck is highlighted in red. The robot can then locate the 3D position of the targeted object with depth information from its depth camera and perform a grasping task.

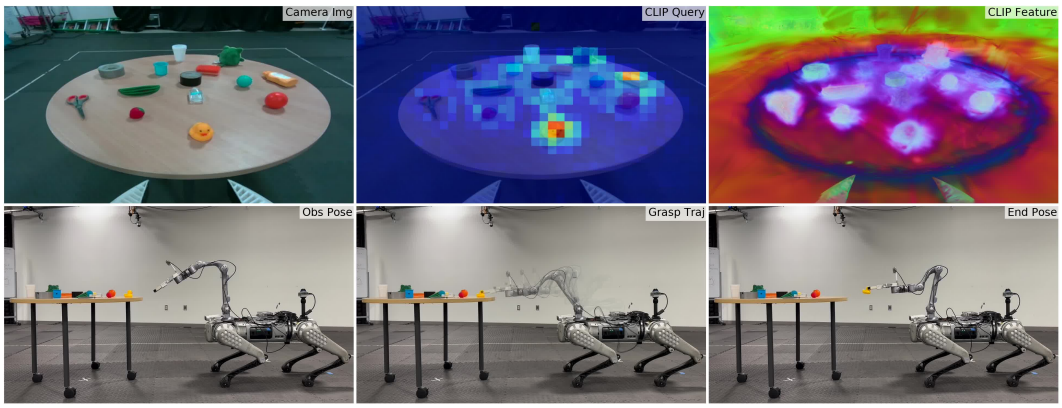

Figure 7: Real robot deployment.

Conclusion. This paper introduces **M3**, a novel approach combining foundation models with Gaussian Splatting to create a spatial multimodal memory resembling human memory. **M3** demonstrates superior downstream task accuracy with reduced training costs and shows practical utility when deployed on a real robot. One interesting future direction is to design a reasoning module that is capable of directly operating on the optimized memory bank, which we leave to future study.

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] Anthropic. The claude 3 model family: Opus, sonnet, haiku. Technical report, Anthropic, 2023. URL `https://www.anthropic.com`. Accessed: 2023-09-19.

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5470–5479, 2022.

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

[5] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[7] Xiaoran Fan, Tao Ji, Changhao Jiang, Shuo Li, Senjie Jin, Sirui Song, Junke Wang, Boyang Hong, Lu Chen, Guodong Zheng, et al. Mousi: Poly-visual-expert vision-language models. *arXiv preprint arXiv:2401.17221*, 2024.

[8] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024.

[9] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13504–13514, 2024.

[10] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. 2018.

[11] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 4083–4091, 2022.

[12] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.

[13] Mazeyu Ji, Ri-Zhao Qiu, Xueyan Zou, and Xiaolong Wang. Graspsplats: Efficient manipulation with 3d feature splatting. *arXiv preprint arXiv:2409.02084*, 2024.

[14] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[15] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[17] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

[18] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017.

[19] Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. *arXiv preprint arXiv:2407.08447*, 2024.

[20] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023.

[21] Sergey Linok, Tatiana Zemskova, Svetlana Ladanova, Roman Titkov, and Dmitry Yudin. Beyond bare queries: Open-vocabulary object retrieval with 3d scene graph. *arXiv preprint arXiv:2406.07113*, 2024.

[22] Joel Loo, Zhanxin Wu, and David Hsu. Open scene graphs for open world object-goal navigation. *arXiv preprint arXiv:2407.02473*, 2024.

[23] 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. *Communications of the ACM*, 65(1):99–106, 2021.

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[25] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *arXiv preprint arXiv:2405.16009*, 2024.

[26] Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Feature splatting: Language-driven physics-based scene synthesis and editing. *arXiv preprint arXiv:2404.01223*, 2024.

[27] Delin Qu, Qizhi Chen, Pingrui Zhang, Xianqiang Gao, Bin Zhao, Dong Wang, and Xuelong Li. Livescene: Language embedding interactive radiance fields for physical scene rendering and control. *arXiv preprint arXiv:2406.16038*, 2024.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

[29] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[30] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2016.

[31] Jinghuan Shang, Karl Schmeckpeper, Brandon B May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. *arXiv preprint arXiv:2407.20179*, 2024.

[32] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023.

[33] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5333–5343, 2024.

[34] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18221–18232, 2024.

[35] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.

[36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[37] 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.

[38] Yiqin Wang, Haoji Zhang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Hierarchical memory for long video qa. *arXiv preprint arXiv:2407.00603*, 2024.

[39] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

[40] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4818–4829, 2024.

[41] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.

[42] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1192–1200, 2020.

[43] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

[44] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19447–19456, 2024.

[45] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[46] Yuanwen Yue, Anurag Das, Francis Engelmann, Siyu Tang, and Jan Eric Lenssen. Improving 2d feature representations by 3d-aware fine-tuning. *arXiv preprint arXiv:2407.20229*, 2024.

[47] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

[48] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*, 2024.

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[50] Hexu Zhao, Haoyang Weng, Daohan Lu, Ang Li, Jinyang Li, Aurojit Panda, and Saining Xie. On scaling up 3d gaussian splatting training. *arXiv preprint arXiv:2406.18533*, 2024.

[51] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21676–21685, 2024.

[52] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36, 2024.