LEG4D: LANGUAGE EMBEDDED GENERATIVE 4D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstructing complete and semantically meaningful 4D scenes from monocular videos is inherently challenging due to limited observations of moving objects. Existing dynamic reconstruction methods mainly recover visible geometry and appearance, while generative approaches often require costly score-distillation optimization and lack dense semantic representations. We present LEG4D, a Language-Embedded Generative 4D Gaussian Splatting framework for complete, semantic, and editable dynamic scene reconstruction. Given a monocular video and object masks, LEG4D employs a pretrained 3D object generation model to complete partially observed objects as object-level Gaussians. The generated Gaussians are integrated into the observed scene through pose- and geometry-aware alignment, multi-view appearance adaptation, and object replacement. A motion-aware dynamic Gaussian module subsequently optimizes the fused scene while maintaining Gaussian correspondences and trajectories across time. We further distill language-aligned features from observed frames into the dynamic Gaussians, producing a complete, temporally consistent, and language-embedded 4D representation. Experiments demonstrate that LEG4D improves both dynamic scene completeness and language-grounding accuracy over prior approaches. The resulting representation additionally supports object-level querying, localization, removal, and localized editing in dynamic scenes.

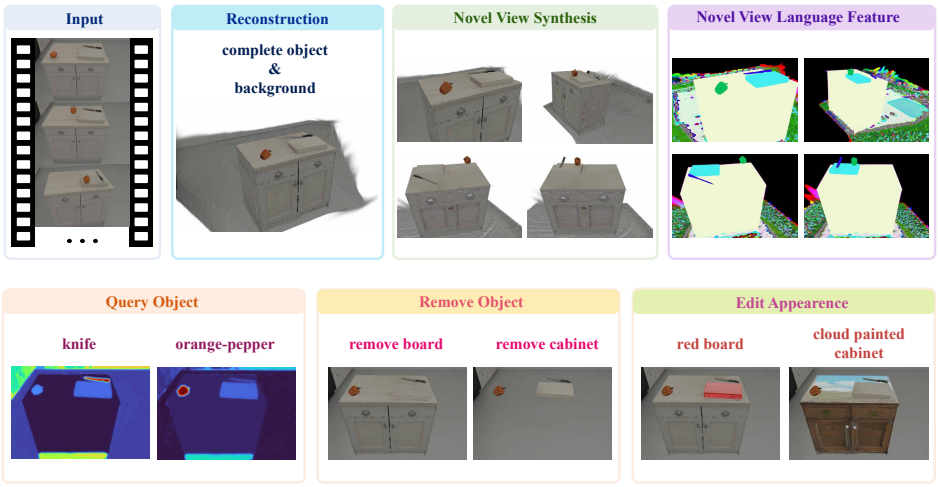


Figure 1: Overview of LEG4D. Given observations of a dynamic scene, LEG4D constructs a language-embedded generative 4D Gaussian representation that supports photorealistic novel-view synthesis and language-aligned semantic rendering. The learned representation further enables language-based object querying and object-level manipulation, including object removal and appearance editing.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Reconstructing complete semantic-aware 4D scenes is a fundamental step toward immersive spatial intelligence (Wang et al., 2026). Instead of merely recovering the appearance of visible surfaces through view-dependent optimization, the goal is to reconstruct geometrically complete 4D scenes with dense semantic embeddings, enabling holistic geometric and semantic understanding beyond observed regions. Such representations have broad applications in embodied AI (Duan et al., 2022), robotic manipulation (Li et al., 2024), and scene editing. However, the problem is inherently ill-posed, as self-occlusions, inter-object occlusions, and object motions leave large portions of scenes unobserved. Recovering plausible unseen geometry while propagating consistent semantic representations to unseen regions remains highly challenging.

Existing dynamic reconstruction methods based on neural fields (Pumarola et al., 2021; Park et al., 2021a;b; Fridovich-Keil et al., 2023) and Gaussian representations (Luiten et al., 2024; Yang et al., 2024; Wu et al., 2024a; Wang et al., 2025) primarily optimize scene representations to fit the input observations, making complete reconstruction with monocular videos inherently difficult. To overcome this limitation, recent generative 3D and 4D reconstruction methods (Chu et al., 2024; 2025) incorporate pretrained generative priors (Liu et al., 2023; Shi et al., 2024b) to complete unseen geometry and appearance. However, these methods predominantly rely on score distillation sampling (SDS) (Poole et al., 2022), resulting in slow optimization. Moreover, they primarily focus on visual supervision without explicitly embedding dense semantic representations, limiting downstream semantic understanding.

Beyond visual reconstruction, useful 4D scene representations should also support semantic understanding. Therefore, recent 4D Gaussian methods (Li et al., 2025; Fiebelman et al., 2025; Labe et al., 2024; Ji et al., 2024; Li et al., 2026; Shi et al., 2025) have begun to embed semantic, or language-aligned features into dynamic Gaussians. Typically, these methods render semantic or language feature maps from the Gaussian representation and supervise them with 2D segmentation masks, or vision-language features, enabling open-vocabulary localization, and spatio-temporal grounding. However, since such supervision is mainly obtained from input views, the learned semantic fields are largely constrained to visible regions, limiting complete object-level understanding and interaction in dynamic 4D Gaussian scenes.

To address these challenges, we propose LEG4D, a Language-Embedded Generative 4D Gaussian Splatting framework that aims to reconstruct a complete and semantically meaningful 4D Gaussian field from monocular dynamic videos. Instead of only recovering the visible scene geometry and appearance, our goal is to complete partially observed objects while embedding language-aligned semantic features into the dynamic Gaussian representation. As illustrated in Figure 1, the representation enables object-level understanding and interaction, including language-based localization, object access, and localized editing in dynamic scenes.

LEG4D achieves this goal through a generative completion and semantic embedding pipeline. Given a dynamic scene with object masks, we first use an off-the-shelf 3D object generation model (Chen et al., 2026) to produce complete object-level Gaussians for partially observed objects. These generated Gaussians are then aligned with the observed scene via pose- and geometry-aware refinement, and their appearance is adapted to the input observations using a multi-view appearance adapter (Huang et al., 2025). After replacing the original incomplete object Gaussians with the refined complete ones, we optimize the fused scene using a motion-aware dynamic Gaussian reconstruction module that maintains Gaussian correspondences and trajectories across time. Finally, we distill language-aligned features extracted from observed frames into the optimized dynamic Gaussians, resulting in a complete, temporally consistent, and language-embedded 4D Gaussian field.

Our contributions are summarized as follows:

- We propose LEG4D, a novel language-embedded generative 4D Gaussian Splatting framework for reconstructing complete, semantic, and editable dynamic scene representations from monocular videos.
- We develop a language-guided object completion and fusion pipeline that identifies partially observed dynamic objects, generates complete object-level Gaussians, and integrates them into the scene through geometric alignment, appearance adaptation, and object replacement, ensuring object completeness and cross-view consistency.

- Extensive experiments demonstrate that our method achieves more complete dynamic scene reconstruction and more accurate language grounding than prior approaches. The learned language-embedded 4D Gaussian representation further supports object-level querying, localization, removal, and localized editing in complete dynamic scenes.

2 Related Work

2.1 Neural Scene Representations

Neural Radiance Fields (NeRFs) (Mildenhall et al., 2021) represent scenes as continuous neural fields and synthesize novel views through volumetric rendering, but often incur high optimization and rendering costs. 3D Gaussian Splatting (Kerbl et al., 2023) represents a scene using explicit Gaussian primitives and enables high-quality novel-view synthesis with efficient differentiable rendering. Building upon this representation, dynamic Gaussian methods extend static Gaussians to model time-varying geometry and appearance. A common paradigm is to construct a canonical Gaussian representation (Wang et al., 2025) and learn a time-conditioned deformation or motion field that maps canonical Gaussians to their states at different timestamps (Yang et al., 2024; Wu et al., 2024a; Luiten et al., 2024; Wan et al., 2024). These approaches provide efficient representations for dynamic scenes, but mainly reconstruct geometry and appearance supported by the input observations. Consequently, objects under persistent occlusion or limited viewpoints often remain incomplete.

2.2 Generative Gaussian Splatting

Generative priors have been introduced into Gaussian representations to recover geometry beyond direct observations. In static settings, diffusion-based optimization enables text- or image-conditioned Gaussian generation (Poole et al., 2022; Tang et al., 2024; Chen et al., 2024; Yi et al., 2024). Large-scale models such as SAM 3D (Chen et al., 2026) further demonstrate strong object completion from partial real-world observations. LabelAny3D (Yao et al., 2026) leverages SAM 3D and aligns the generative object with the observed scene for automatic 3D annotation. Generative Gaussian representations have also been extended to dynamic 4D content (Ren et al., 2023; Chu et al., 2024). DreamScene4D (Chu et al., 2024) decomposes monocular videos into foreground objects and backgrounds and factorizes object motion for multi-object 4D scene generation. GenMOJO (Chu et al., 2025) follows this object-centric paradigm and further improves deformable Gaussian modeling under object interactions and mutual occlusions. However, these methods typically rely on costly per-scene optimization, which is time-consuming and often unstable. In contrast, we employ SAM 3D as an object-level generative prior to reconstruct complete object Gaussians, which are subsequently incorporated into the observed scene through geometric alignment, appearance adaptation, and object replacement.

2.3 Semantic Gaussian Splatting

Semantic scene representations augment geometry and appearance with visual-language features, enabling open-vocabulary querying, segmentation, and editing. Early language-field methods extract dense features from pretrained vision-language models and distill them into neural radiance fields through multi-view rendering supervision (Kerr et al., 2023). Subsequent approaches associate each Gaussian primitive with a learnable semantic embedding and optimize these embeddings to reproduce the 2D language features extracted from input images (Qin et al., 2024; Shi et al., 2024a; Ye et al., 2024; Wu et al., 2024b). Semantic Gaussian representations have also been extended to dynamic scenes, where language features are propagated or jointly optimized with time-varying Gaussian geometry to maintain semantic consistency across frames (Fiebelman et al., 2025; Labe et al., 2024; Li et al., 2025). However, these methods derive semantic supervision primarily from visible image observations, leaving persistently occluded or geometrically missing regions without reliable semantic embeddings. In contrast, we learn a unified language embedding field over both the observed scene Gaussians and the completed object-level Gaussians, preserving consistent object semantics across visible and reconstructed regions.

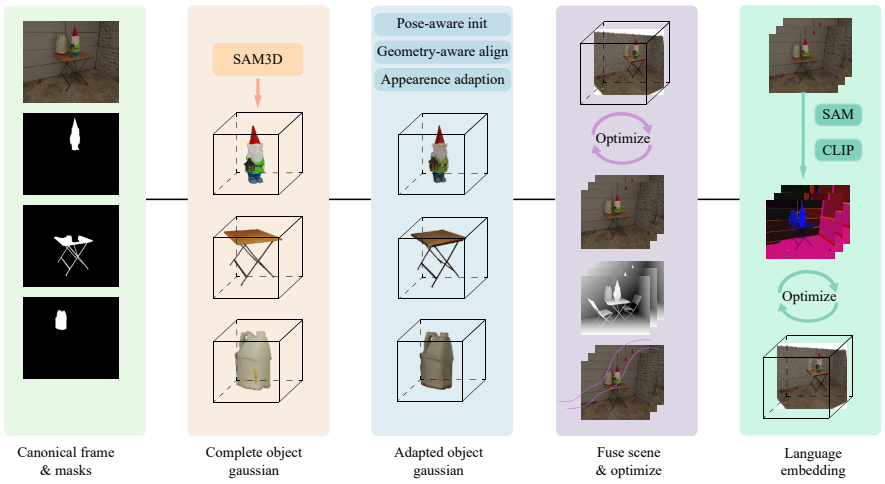


Figure 2: Given a dynamic video, LEG4D generates complete object-level Gaussians from object observations using SAM3D(Chen et al., 2026) The generated Gaussians are adapted and fused into the dynamic scene, followed by optimization with RGB, tracking, depth, and language features to obtain a complete, temporally coherent, and language-embedded 4D Gaussian scene.

3 METHOD

Given an input video, our goal is to construct a complete and semantically queryable 4D Gaussian scene. We first build a canonical Gaussian representation and learn its temporal motion from tracked video observations. We then use generative object reconstruction to complete occluded scene content and align the generated objects with the canonical scene in geometry and appearance. Finally, we attach learnable semantic embeddings to the Gaussians and optimize them using 2D semantic supervision, enabling open-vocabulary scene understanding and object-level interaction.

3.1 PRELIMINARIES

Dynamic Gaussian Splatting Dynamic Gaussian Splatting represents a scene using Gaussians defined in a canonical space, $G_i^c = \{\mathbf{x}_i^c, \mathbf{R}_i^c, \mathbf{s}_i^c, \alpha_i^c, \mathbf{c}_i^c\}$. A time-dependent transformation maps each canonical Gaussian to its state at time t:

$$G_i^t = \mathcal{W}_\theta(G_i^c, t). \tag{1}$$

For object-level rigid motion, the Gaussian center and orientation are transformed as

$$\mathbf{x}_i^t = \mathbf{R}_o^t \mathbf{x}_i^c + \mathbf{t}_o^t, \qquad \mathbf{R}_i^t = \mathbf{R}_o^t \mathbf{R}_i^c, \tag{2}$$

where $\mathbf{R}_o^t$ and $\mathbf{t}_o^t$ denote the object pose at time t. This formulation separates persistent scene content from temporal motion and enables Gaussian correspondences to be maintained across frames.

Semantic Gaussian Splatting. Semantic Gaussian Splatting augments each Gaussian with a semantic embedding $\mathbf{f}_i$. Similar to color rendering, the semantic feature at pixel p is obtained through depth-aware alpha compositing:

$$\mathbf{F}(p) = \sum_{i \in N(p)} T_i(p)\alpha_i(p)\mathbf{f}_i, \qquad T_i(p) = \prod_{j<i} (1 - \alpha_j(p)). \tag{3}$$

The rendered features are typically supervised using pixel-aligned features extracted from pretrained vision-language or segmentation models (Radford et al., 2021; Kirillov et al., 2023), thereby lifting 2D semantics into the Gaussian representation. In dynamic scenes, semantic embeddings can be associated with canonical Gaussians and propagated together with their geometry across time, providing a temporally consistent semantic field for open-vocabulary querying, segmentation, and object-level interaction.

3.2 4D Scene and Motion Representation

We first establish a canonical representation of the 4D scene. Given an input video, we use a video tracking model (Doersch et al., 2024) to extract point trajectories for the foreground objects throughout the sequence. We select the frame in which the foreground objects are collectively most visible and designate its timestamp t_c as the canonical time. The tracked observations are then canonicalized according to their temporal correspondences: for each trajectory, its observations across the video are associated with and mapped to the corresponding point in the frame at t_c. These points are back-projected using the depth and camera parameters of the canonical frame to initialize the canonical Gaussian geometry. To recover object regions that remain occluded or unobserved, we further incorporate complete object-level Gaussian representations produced by the generative construction. The generated Gaussians are aligned with their observed counterparts in pose and scale and merged into the same canonical scene space.

With the completed canonical scene established, we model its temporal evolution using a learned motion field $\mathcal{W}_\theta$. Following the Dynamic Gaussian Splatting formulation, we represent the motion field as a combination of K learnable time-dependent rigid motion components. The k-th component at time t is denoted by $T_k^t = (\mathbf{R}_k^t, \mathbf{t}_k^t)$, where $\mathbf{R}_k^t$ and $\mathbf{t}_k^t$ represent its rotation and translation, respectively. For each dynamic Gaussian G_i, we assign a set of motion coefficients $\mathbf{a}_i = \{a_{ik}\}_{k=1}^K$ and obtain its transformation at time t by blending these motion components:

$$T_i^t = \sum_{k=1}^{K} a_{ik} T_k^t. \tag{4}$$

In practice, the blending is performed separately on the translation vectors and continuous rotation parameters, after which the blended rotation is converted into a valid rotation matrix. Applying T_i^t to the canonical Gaussian produces its state at time t.

To preserve object-level motion coherence, all Gaussians belonging to the same completed object share a common set of motion coefficients. For a completed object $\mathcal{G}_o$, we impose

$$\mathbf{a}_i = \mathbf{a}_o, \qquad \forall G_i \in \mathcal{G}_o^{\text{obj}}, \tag{5}$$

where $\mathbf{a}_o$ denotes the object-level motion coefficients. Consequently, the completed object is driven by a consistent transformation over time. Background Gaussians and static completed objects are assigned identity motion.

Finally, we jointly optimize the canonical Gaussian attributes, the time-dependent motion components, and the object-level motion coefficients through differentiable rendering under the supervision of the observed video frames:

$$\mathcal{L}_{\text{4D}} = \lambda_{\text{render}}\mathcal{L}_{\text{render}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}} + \lambda_{\text{temp}}\mathcal{L}_{\text{temp}}. \tag{6}$$

Here, $\mathcal{L}_{\text{render}}$ supervises frame-level appearance reconstruction, $\mathcal{L}_{\text{geo}}$ preserves the geometry and relative scale of the composed scene, and $\mathcal{L}_{\text{temp}}$ encourages temporally coherent motion.

3.3 Generative Scene Construction

Given a target frame I_t and its object mask M_t, SAM3D produces a complete object Gaussian representation $\mathcal{G}_o^{\text{obj}} = \{G_i^{\text{obj}}\}_{i=1}^{N_o}$ in an object-local coordinate system, together with a corresponding mesh $\mathcal{M}_o$. Since the generated object is not yet aligned with the reconstructed 4D scene, we subsequently refine its pose and scale using the target depth, camera intrinsics, and camera pose. The same geometric transformations are applied to both the Gaussian representation and the mesh, ensuring their geometric consistency. The transformed mesh is later used to provide geometry conditions for MV-Adapter (Huang et al., 2025).

Starting from the initial object-to-camera pose provided by SAM3D, we render the generated object Gaussians into the target view. We coarsely refine the pose by aligning the rendered alpha map $\hat{A}_t$ with the target mask M_t, using their image-space centroid, foreground extent, and boundary consistency. To further reduce the ambiguity of silhouette-only alignment, we derive target surface normals from the depth map D_t and enforce consistency between the target and rendered normals over their valid overlap. This stage yields a coarse object-to-camera transformation T_{coarse}, which serves as the initialization for the subsequent refinement.

We further refine the object alignment using RoMaV2 correspondences (Edstedt et al., 2025). Given the coarse object-to-camera transformation T_{coarse}, we render the object image $\hat{I}_t$ and depth map $\hat{D}_t$, and establish correspondences between $\hat{I}_t$ and the masked target image $I_t \odot M_t$. Each correspondence is denoted by $(\hat{\mathbf{p}}_k, \mathbf{p}_k, w_k)$, where $\hat{\mathbf{p}}_k$ and $\mathbf{p}_k$ are the matched points in the rendered and target images, respectively, and w_k is the matching confidence. We back-project $\hat{\mathbf{p}}_k$ using $\hat{D}_t$ and K_t and transform the resulting camera-space point into the object-local coordinate system using T_{coarse}^{-1}, obtaining $\hat{\mathbf{X}}_k^o$. We define its reprojection residual as $\mathbf{r}_k = \pi\Big(K_t[\mathbf{R}(\mathbf{s} \odot \hat{\mathbf{X}}_k^o) + \mathbf{t}]\Big) - \mathbf{p}_k$ and jointly refine the object-to-camera pose $T_o^* = (\mathbf{R}, \mathbf{t})$ and anisotropic object scale $\mathbf{s}$ by solving

$$\min_{\substack{\mathbf{R},\mathbf{t} \\ \mathbf{s}\in[s_{\min},s_{\max}]^3}} \sum_k w_k \rho\left(\|\mathbf{r}_k\|_2\right) + \lambda_s \mathcal{R}_s(\mathbf{s}). \tag{7}$$

Here, $\rho(\cdot)$ is the Huber loss, and $\mathcal{R}_s(\mathbf{s})$ keeps the scale close to one while discouraging excessive anisotropy. The optimized scaling is applied consistently to the Gaussian centers and covariance matrices, from which the Gaussian rotations and scales are updated. The mesh vertices undergo the same transformation to maintain geometric consistency.

Although the correspondence-based refinement aligns the object with its image observation, a global scale–depth ambiguity may remain. We therefore render the refined object into the target view and estimate a uniform depth scale over the valid overlap Ω_t between the target mask and the rendered alpha map:

$$s_d = \frac{\sum_{p\in\Omega_t} \hat{D}_t(p) D_t(p)}{\sum_{p\in\Omega_t} \hat{D}_t(p)^2}. \tag{8}$$

We apply s_d uniformly in the target camera coordinate system, scaling both the object position and its physical size. Accordingly, the Gaussian log-scales are updated by adding $\log s_d$, and the mesh undergoes the same scaling. This corrects the global depth while preserving the object structure and image-space projection.

After the depth correction, we map the object into the 4D scene coordinate system as $T_{o\to\text{scene}} = T_{coarse}^{-1} T_o^*$, where T_o^* is the refined object-to-camera transformation and T_t maps scene-space points to the target camera coordinate system. The resulting transformation is applied consistently to the Gaussian centers and rotations and to the corresponding mesh. Finally, we adapt the object appearance to the target video. We render the aligned mesh $\mathcal{M}_o$ from six predefined views to obtain position and normal maps $\{(\mathbf{P}_v, \mathbf{N}_v)\}_{v=1}^6$. Conditioned on the target object image, its mask, and these geometry maps, MV-Adapter (Huang et al., 2025) generates six appearance-adapted views:

$$\{\tilde{I}_v\}_{v=1}^6 = \Phi_{\text{MVA}}\left(I_t \odot M_t, \{(\mathbf{P}_v, \mathbf{N}_v)\}_{v=1}^6\right). \tag{9}$$

We use these generated views as multi-view supervision to optimize the appearance attributes of the object Gaussians while preserving their aligned geometry. The resulting representation has complete geometry, scene-consistent pose and scale, and appearance compatible with the target video.

3.4 Semantic Field Construction and Optimization

To make the completed 4D Gaussian scene semantically queryable, we augment each Gaussian with a learnable semantic embedding. Given a training view I_v, we use SAM to produce multi-granularity masks and extract a CLIP (Radford et al., 2021) descriptor for each masked region. To reduce the cost of storing high-dimensional CLIP features, we train a lightweight scene-specific autoencoder and use its latent code as the compact semantic descriptor. The compressed region descriptors are assigned to the pixels covered by their corresponding masks, yielding a pixel-aligned semantic target map $\mathbf{S}_v$ for each view. We attach a semantic feature $\mathbf{f}_i \in \mathbf{R}^d$ to each Gaussian and define

$$\mathcal{G}_i = \boldsymbol{\mu}_i, \mathbf{R}_i, \mathbf{s}_i, \alpha_i, \mathbf{c}_i, \mathbf{f}_i, \tag{10}$$

where $\boldsymbol{\mu}_i$, $\mathbf{R}_i$, $\mathbf{s}_i$, α_i, and $\mathbf{c}_i$ denote position, rotation, scale, opacity, and color, respectively. During semantic optimization, we reuse the differentiable splatting operator in equation 3 to render a semantic feature map $\hat{\mathbf{S}} * v$ from the Gaussian embeddings. The embeddings are optimized by matching the rendered features to the 2D semantic targets:

$$\mathcal{L}_{\text{sem}} = \sum_v \sum_p \left|\hat{\mathbf{S}}_v(p) - \mathbf{S}_v(p)\right|_1. \tag{11}$$

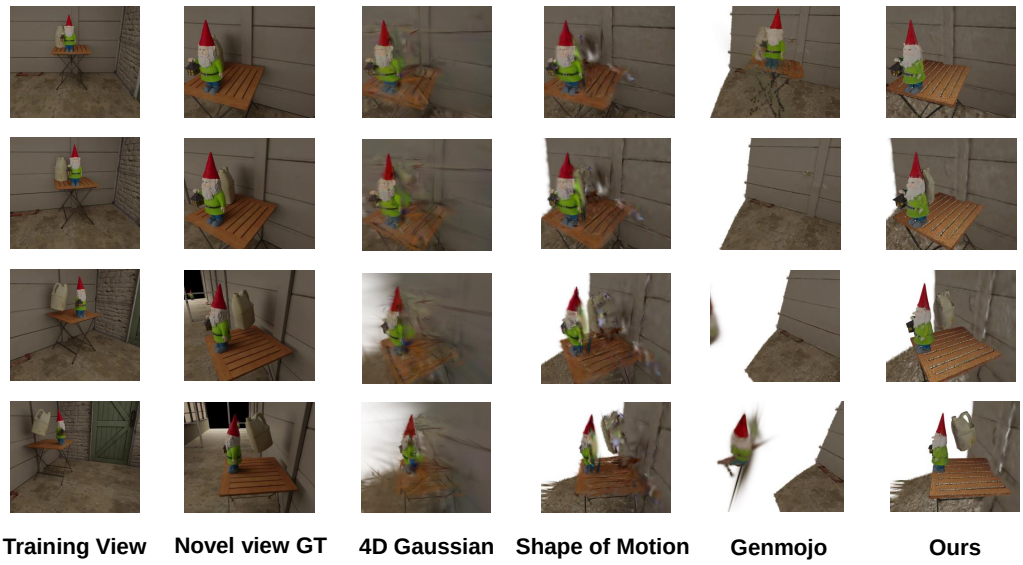


Figure 3: Qualitative comparison of novel view synthesis.

We keep the reconstructed geometry and opacity fixed during this stage. Thus, semantic learning does not modify the completed scene structure, but only learns a semantic field over the fixed 4D Gaussian representation.

For completed object Gaussians, we further impose object-level semantic sharing. Let $\mathcal{G}_m^{\text{obj}}$ be the set of Gaussians belonging to the m-th completed object. Instead of assigning independent embeddings to these Gaussians, we use a shared object embedding:

$$\mathbf{f}_i = \mathbf{f}_m^{\text{obj}}, \qquad \forall G_i \in \mathcal{G}_m^{\text{obj}}. \tag{12}$$

This sharing transfers semantic supervision from visible object regions to their completed, originally unobserved parts, ensuring that the object maintains a consistent semantic identity under novel viewpoints.

Overall, this optimization constructs a unified semantic field over the completed 4D Gaussian scene. The reconstructed and generated Gaussians provide geometry and appearance, while multi-level mask supervision and object-level embedding sharing provide semantic consistency, enabling open-vocabulary querying, semantic segmentation, and object-level interaction.

4 Experiments

Datasets. We evaluate our method on two synthetic datasets. We generated 25 multi-object dynamic scenes using the Kubric simulator (Greff et al., 2022) , referred to as Kubric4D-25. Each scene is rendered from five synchronized camera trajectories, providing controlled yet challenging scenarios with diverse object scales, motions, viewpoints, and occlusions. We additionally construct Mesh4D-6, a more realistic dataset consisting of six multi-object scenes composed from publicly available high-quality 3D mesh assets. For each scene, one monocular video is used for training, while four held-out views for Kubric4D-25 and one held-out view for Mesh4D-6 are used for evaluation.

Implementation Details. All experiments are conducted on a single NVIDIA Tesla V100 GPU. For semantic feature extraction, we use OpenCLIP ViT-B/16 and compress the extracted CLIP features into a three-dimensional latent space for Gaussian semantic field optimization. For object appearance adaptation, we adopt the Partial Image + Geometry to Multiview model of MV-Adapter (Huang et al., 2025) to generate geometry-aware multi-view appearance supervision. More implementation details are provided in the Appendix.

Method	Kubric4D-25		Mesh4D-6	
	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑
4DGaussians (Wu et al., 2024a)	0.192	16.83	0.541	13.51
Shape-of-Motion (Wang et al., 2025)	0.086	20.96	0.561	9.71
Genmojo (Chu et al., 2025)	0.099	18.80	0.566	11.61
Ours	**0.072**	**20.97**	**0.400**	**19.64**
w/o MV-Adapter (Huang et al., 2025)	0.074	20.72	0.548	8.467

Table 1: Novel view generation quantitative comparison on Kubric4D-25 and Mesh4D-6.

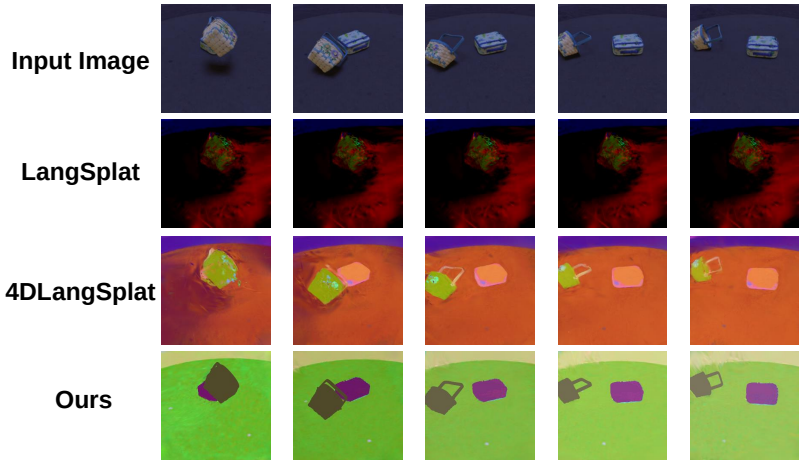


Figure 4: Qualitative comparison of semantic fields

Baselines. For 4D reconstruction, we compare our method with both non-generative and generative approaches. The non-generative baselines include 4D Gaussian Splatting (Wu et al., 2024a) and Shape of Motion (Wang et al., 2025), while GenMoJo (Chu et al., 2025) serves as the generative baseline. For semantic field reconstruction, we compare against queryable semantic-field methods, including LangSplat (Qin et al., 2024), DGD (Labe et al., 2024), and 4D LangSplat (Li et al., 2025).



(a) Guidance Image **(b) w/o MV-Adapter** **(c) w/ MV-Adapter**

Figure 5: (a) Input guidance, including object crops extracted from the training image. (b) Complete object appearance produced by SAM3D. (c) Object appearance directly adapted by MV-Adapter to match the guidance image.

Novel-View Generation. For the Kubric4D-25 dataset, the novel views are offset from the training views by approximately $45°$. Furthermore, for the Mesh4D-6 dataset, we select novel views with the largest possible viewpoint offsets, reaching up to $180°$ relative to the input view, to evaluate the model's ability to complete previously unobserved regions. Table 1 presents the novel-view generation results. Our method consistently outperforms both non-generative reconstruction approaches and the generative baseline. We further conduct an ablation study by removing MV-Adapter from our framework. The resulting performance exhibits a slight degradation, indicating that the mul-

tiview generative prior provided by MV-Adapter contributes to more accurate completion of unobserved regions. A visualized example on whether using MV-Adapter is attached in Figure 5. We find the appearance optimized using MV-Adapter is more harmonious. Moreover, our method achieves a $1.74\times$ training speedup over GenMOJO (Chu et al., 2025) on the Kubric4D-25 dataset, demonstrating its high computational efficiency.

Method	Kubric4D-25	
	mIoU↑	mAcc↑
LangSplat (Qin et al., 2024)	28.91	91.67
DGD (Labe et al., 2024)	15.19	95.37
4DLangSplat (Li et al., 2025)	36.26	95.23
Ours	**63.98**	**98.08**

Table 2: Text-based query quantitative comparison on Kubric4D-25.

Text-Based Querying. For the Kubric4D-25 dataset, we use the category and description associated with each object during scene generation as the ground-truth text queries. The corresponding 2D segmentation masks are obtained through feature matching, and the results are evaluated using mean Intersection over Union (mIoU) and mean accuracy (mAcc). As shown in Table 2, our method consistently outperforms the competing approaches on both metrics, demonstrating more accurate and reliable language-guided object localization and segmentation.

Scene Editing. Beyond text-based object querying, our framework also supports text-guided object removal and appearance editing. For object removal, we first identify the corresponding 2D mask from the input text and selectively remove the Gaussians that contribute to the masked region. For appearance editing, we employ the Text-Geometry-to-Multiview model of MV-Adapter(Huang et al., 2025) to generate six-view images conditioned on the text prompt and object geometry. These generated views are then used to optimize the Gaussians associated with the target object, enabling text-guided modification of its appearance while preserving its underlying geometry and motion.

5 Conclusion

We presented LEG4D, a language-embedded generative 4D Gaussian framework for constructing complete and queryable dynamic scene representations. LEG4D generates complete object-level Gaussians for partially observed objects, refines their spatial alignment, adapts their appearance, and fuses them into the reconstructed 4D scene. It further learns a unified language-embedded Gaussian field, where shared object-level semantic features enable both observed and generated regions to respond consistently to open-vocabulary queries. Experiments demonstrate improved novel-view rendering and text-based object localization over representative reconstruction and semantic-field baselines. These results highlight the potential of combining generative object completion with language-aware 4D Gaussian reconstruction for more complete and interactive dynamic scene modeling.

6 Limitations

While LEG4D demonstrates promising performance, several limitations remain. Its completion quality depends on the underlying 3D generative prior and the accuracy of object masks, depth estimates, and correspondences. The current framework also does not explicitly model physical interactions, illumination changes, or complex non-rigid deformations. Moreover, its semantic field primarily captures time-agnostic object-level identity rather than fine-grained part semantics, actions, or temporally varying object states. Finally, the multi-stage generation, alignment, appearance adaptation, and semantic optimization pipeline introduces additional computational cost. Future work will explore stronger feed-forward completion and alignment models, joint scene-object optimization, and richer part-aware and spatio-temporal semantic representations.

AI USE STATEMENT

We used generative AI tools to assist with language polishing and drafting parts of the manuscript. We also used these tools to assist with literature retrieval, research ideation, and research execution. All AI-assisted content was reviewed and verified by the authors. The research methodology, experimental results, and final conclusions were determined by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses synthetically generated datasets and does not involve human-subject studies or the collection of personal information. We are not aware of specific ethical concerns associated with this work.

REPRODUCIBILITY STATEMENT

We provide details of the proposed method, implementation, and experimental settings in the main paper and appendix. The datasets and evaluation metrics used in our experiments are also specified. We will release the code and relevant configurations to facilitate reproduction of our results.

REFERENCES

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026.

Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21401–21412, 2024.

Wen-Hsuan Chu, Lei Ke, and Katerina Fragkiadaki. Dreamscene4d: Dynamic multi-object scene generation from monocular videos. *Advances in Neural Information Processing Systems*, 37: 96181–96206, 2024.

Wen-Hsuan Chu, Lei Ke, Jianmeng Liu, Mingxiao Huo, Pavel Tokmakov, and Katerina Fragkiadaki. Generative 4d scene gaussian splatting with object view-synthesis priors. *arXiv preprint arXiv:2506.12716*, 2025.

Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, Joao Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Proceedings of the Asian Conference on Computer Vision*, pp. 3257–3274, 2024.

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022.

Johan Edstedt, David Nordström, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. Roma v2: Harder better faster denser feature matching. *arXiv preprint arXiv:2511.15706*, 2025.

Gal Fiebelman, Tamir Cohen, Ayellet Morgenstern, Peter Hedman, and Hadar Averbuch-Elor. 4-legs: 4d language embedded gaussian splatting. In *Computer Graphics Forum*, volume 44, pp. e70085. Wiley Online Library, 2025.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12479–12488, 2023.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3749–3761, 2022.

Zehuan Huang, Yuan-Chen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16377–16387, 2025.

Shengxiang Ji, Guanjun Wu, Jiemin Fang, Jiazhong Cen, Taoran Yi, Wenyu Liu, Qi Tian, and Xinggang Wang. Segment any 4d gaussians. *arXiv preprint arXiv:2407.04504*, 2024.

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

Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 19729–19739, 2023.

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.

Isaac Labe, Noam Issachar, Itai Lang, and Sagie Benaim. Dgd: Dynamic 3d gaussians distillation. In *European Conference on Computer Vision*, pp. 361–378. Springer, 2024.

Puhao Li, Tengyu Liu, Yuyang Li, Muzhi Han, Haoran Geng, Shu Wang, Yixin Zhu, Song-Chun Zhu, and Siyuan Huang. Ag2manip: Learning novel manipulation skills with agent-agnostic visual and action representations. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 573–580. IEEE, 2024.

Wanhua Li, Renping Zhou, Jiawei Zhou, Yingwei Song, Johannes Herter, Minghan Qin, Gao Huang, and Hanspeter Pfister. 4d langsplat: 4d language gaussian splatting via multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22001–22011, 2025.

Yun-Jin Li, Mariia Gladkova, Yan Xia, and Daniel Cremers. Trase: Tracking-free 4d segmentation and editing. In *2026 International Conference on 3D Vision (3DV)*, pp. 1670–1679. IEEE, 2026.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9298–9309, 2023.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pp. 800–809. IEEE, 2024.

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.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5865–5874, 2021a.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021b.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10318–10327, 2021.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20051–20060, 2024.

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.

Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023.

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, 2024a.

Jin-Chuan Shi, Chengye Su, Jiajun Wang, Ariel Shamir, and Miao Wang. Mono4deditor: Text-driven 4d scene editing from monocular video via point-level localization of language-embedded gaussians. *arXiv preprint arXiv:2510.09438*, 2025.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International conference on learning representations*, volume 2024, pp. 39838–39859, 2024b.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations*, volume 2024, pp. 33879–33896, 2024.

Diwen Wan, Ruijie Lu, and Gang Zeng. Superpoint gaussian splatting for real-time high-fidelity dynamic scene reconstruction. *arXiv preprint arXiv:2406.03697*, 2024.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Chang Zeng, Yanxi Zhou, Xiao-Xiao Long, et al. Spatialvid: A large-scale video dataset with spatial annotations. 2026.

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9660–9672, 2025.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20310–20320, 2024a.

Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024b.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20331–20341, 2024.

Jin Yao, Radowan Mahmud Redoy, Sebastian Elbaum, Matthew Dwyer, and Zezhou Cheng. Labelany3d: Label any object 3d in the wild. *Advances in neural information processing systems*, 38:133188–133216, 2026.

Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European conference on computer vision*, pp. 162–179. Springer, 2024.

Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6796–6807, 2024.

A Appendix

A.1 Data Preprocessing

Given an input video $\{I_t\}_{t=1}^{T}$, we preprocess camera parameters, depth maps, object masks, point tracks, and language features before Gaussian optimization.

Geometry preprocessing. We first estimate metric depth $\mathbf{D}_t^{\text{metric}}$ and monocular depth $\mathbf{D}_t^{\text{mono}}$ for each frame. The monocular predictions are aligned to the metric scale as

$$\bar{\mathbf{D}}_t = \mathcal{A}\left(\mathbf{D}_t^{\text{mono}}, \mathbf{D}_t^{\text{metric}}\right), \tag{13}$$

where $\mathcal{A}$ denotes the depth-alignment procedure.

The RGB frames, aligned depths, and estimated intrinsics $\mathbf{K}$ are then provided to a depth-assisted DROID reconstruction pipeline to recover camera poses:

$$\{\mathbf{T}_t\}_{t=1}^{T} = \mathcal{R}_{\text{cam}}\left(\{I_t\}_{t=1}^{T}, \{\bar{\mathbf{D}}_t\}_{t=1}^{T}, \mathbf{K}\right). \tag{14}$$

Foreground and object masks $\{\mathbf{S}_t\}_{t=1}^{T}$ are used to identify dynamic regions. Using the foreground and object masks as guidance, we further employ a video inpainting model to recover background regions occluded by dynamic objects, producing temporally consistent background frames for subsequent scene reconstruction. We further apply BootsTAPIR within these regions to obtain long-range point trajectories

$$\mathcal{P}^n = \{\mathbf{p}_t^n, v_t^n\}_{t=1}^{T}, \tag{15}$$

where $\mathbf{p}_t^n$ is the image coordinate and v_t^n denotes its visibility. These tracks are lifted into 3D using the aligned depths and camera parameters.

Language feature preprocessing. We use Tracking Anything with DEVA to extract temporally consistent multi-region masks:

$$\mathcal{M}_t = \left\{\mathbf{M}_t^k\right\}_{k=1}^{K_t}. \tag{16}$$

For each region, an OpenCLIP feature is extracted as

$$\mathbf{c}_t^k = E_{\text{CLIP}}\left(I_t \odot \mathbf{M}_t^k\right). \tag{17}$$

To reduce storage and optimization costs, a scene-specific encoder compresses each CLIP feature into a three-dimensional latent vector:

$$\mathbf{z}_t^k = E_{\phi_{\mathcal{S}}}\left(\mathbf{c}_t^k\right), \qquad \mathbf{z}_t^k \in \mathbf{R}^3. \tag{18}$$

The corresponding decoder is trained to reconstruct the original feature:

$$\mathcal{L}_{\text{AE}} = \sum_{t,k} \left\| D_{\psi_{\mathcal{S}}}\left(\mathbf{z}_t^k\right) - \mathbf{c}_t^k \right\|_2^2. \tag{19}$$

The compressed features are assigned to their corresponding mask regions to construct the pixel-aligned language feature map $\mathbf{F}_t^{\text{lang}}$. Its validity mask is defined as

$$\mathbf{V}_t^{\text{lang}}(\mathbf{p}) = \mathbf{1}\left[\sum_{k=1}^{K_t} \mathbf{M}_t^k(\mathbf{p}) > 0\right]. \tag{20}$$

These maps provide time-agnostic semantic supervision during the language feature optimization stage.

A.2 TRAINING PROCEDURE.

The optimization is divided into two stages: scene reconstruction and language feature learning. During the reconstruction stage, we jointly optimize the foreground, background, and inserted object representations, together with their motion parameters. Gaussian positions, scales, rotations, and colors are optimized with learning rates of 1.6×10^{-4}, 5×10^{-3}, 1×10^{-3}, and 1×10^{-2}, respectively. The opacity learning rate is set to 1×10^{-2} for the foreground and 5×10^{-2} for the background and inserted objects. The shared motion bases use a learning rate of 1.6×10^{-4}, while the motion coefficients of the foreground and inserted objects use 1×10^{-2} and 1×10^{-3}, respectively. The reconstruction stage lasts for 500 epochs, after which the resulting model is saved as the initialization for language feature learning.

At epoch 500, we switch to the language feature learning stage. All parameters except the per-Gaussian language features are frozen, including Gaussian positions, scales, rotations, opacities, colors, motion parameters, object deformation parameters, and camera parameters. We optimize the foreground language features with a learning rate of 2.5×10^{-3} and use 1×10^{-1} for the background and inserted-object language features. Using the precomputed pixel-aligned semantic feature maps as supervision, this stage embeds semantic information into the fixed 4D Gaussian representation without altering its reconstructed geometry, appearance, object placement, or temporal motion.

A.3 VISUALIZATION OF KUBRIC4D-25

Figure 6 presents representative examples from our Kubric4D-25 dataset. The dataset contains multi-object dynamic scenes with diverse object categories, scales, motions, spatial arrangements, and occlusion patterns. Each scene is rendered from multiple temporally synchronized camera trajectories, providing observations of the same dynamic content from different viewpoints. These examples illustrate the visual and motion diversity of the dataset used in our experiments.



Figure 6: **Representative scenes from the Kubric4D-25 dataset.** Each row presents synchronized observations of the same dynamic scene from different camera viewpoints.