TRASE: Tracking-free 4D Segmentation and Editing

Yun-Jin Li[⋆1] Mariia Gladkova[⋆1,2†] Yan Xia[1,2]
Daniel Cremers[1,2]
[1] TU Munich [2] Munich Center for Machine Learning
{yunjin.li, mariia.gladkova, yan.xia, cremers}.@tum.de

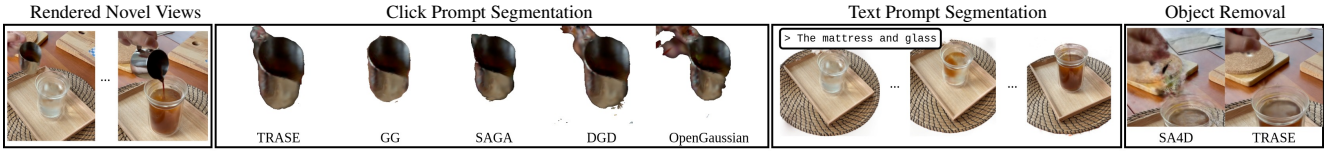

Figure 1. We propose TRASE, a novel tracking-free 4D segmentation approach. TRASE achieves superior object segmentation from click prompts and further supports interactive editing tasks such as object removal and text-prompt-based segmentation.

Abstract

Understanding dynamic 3D scenes is crucial for extended reality (XR) and autonomous driving. Incorporating semantic information into 3D reconstruction enables holistic scene representations, unlocking immersive and interactive applications. To this end, we introduce TRASE, a novel tracking-free 4D segmentation method for dynamic scene understanding. TRASE learns a 4D segmentation feature field in a weakly-supervised manner, leveraging a soft-mined contrastive learning objective guided by SAM masks. The resulting feature space is semantically coherent and well-separated, and final object-level segmentation is obtained via unsupervised clustering. This enables fast editing, such as object removal, composition, and style transfer, by directly manipulating the scene's Gaussians. We evaluate TRASE on five dynamic benchmarks, demonstrating state-of-the-art segmentation performance from unseen viewpoints and its effectiveness across various interactive editing tasks. Our project page is available at: https://yunjinli.github.io/project-sadg/

1. Introduction

Humans experience the world in motion, where objects persist over time despite changes in pose, shape, and appearance. Our ability to recognize objects remains stable, even as we move and observe them from different vantage points. This phenomenon suggests the existence of neural 3D representations, which preserve semantic consistency across both space and time. Such representations are central for applications in augmented reality, gaming, and autonomous systems, where interactive manipulation of dynamic scenes requires object-level segmentation that is spatio-temporally consistent and efficient.

Neural Radiance Fields (NeRFs) [28] have significantly advanced 3D reconstruction by enabling photorealistic novel view synthesis, but their implicit nature makes them computationally too expensive for real-time interaction. Gaussian Splatting (3DGS) [18] has recently emerged as a powerful alternative, achieving real-time rendering with explicit and compact scene representations. Dynamic extensions of 3DGS [25, 45] allow faithful reconstruction of moving scenes, but they lack built-in object segmentation and rely purely on geometry and color. While several works have integrated semantic information into static 3D Gaussian representations [4, 46], they fail to handle motion, deformation, and viewpoint variation in dynamic scenes.

Only a handful of works have explored the unification of semantics and dynamics in 3DGS [17, 22]. SA4D [17] incorporates segmentation into dynamic Gaussian splatting, using video object trackers to match masks across views similar to Gaussian Grouping [46]. However, approaches using these trackers often suffer from identity switches, causing inconsistent segmentation in multi-view settings - particularly in the presence of occlusions, non-rigid deformations, or rapid motion, as shown in Fig. 3. DGD [22] addresses temporal consistency by distilling semantically-aware features from foundation models like CLIP [38] and DINOv2 [29], enabling tracking. Yet, these features are high-dimensional, leading to significant computational

†Corresponding author. ⋆ Equal contribution.

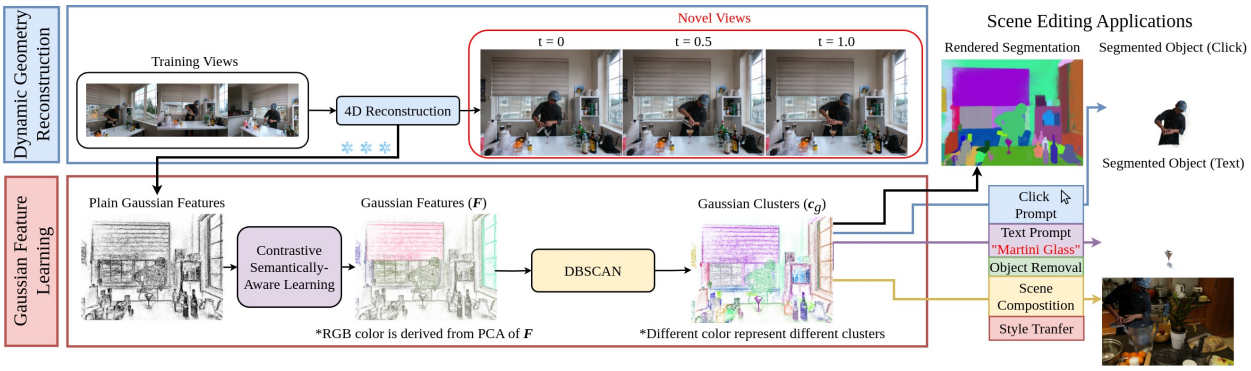

Figure 2. **Our pipeline.** TRASE consists of two main components: dynamic geometry reconstruction (Sec. 3.1) and Gaussian feature learning (Sec. 3.2). We adopt the approach from [45] to effectively learn dynamic 3D reconstruction. Given a 4D reconstruction, we learn Gaussian features $\boldsymbol{F} \in \mathbb{R}^{N\times 32}$ using a novel contrastive learning objective guided by SAM [20] masks. Once trained, we apply clustering [9] directly to the learned features, enabling segmentation field rendering. Our representation supports various scene-editing applications, including object segmentation via click/text prompts in our GUI, object removal, and scene composition.

overhead from both the large memory requirements and slow training.

In this work, we present TRASE, a tracking-free segmentation framework for dynamic scene understanding. TRASE learns temporally consistent segmentation features in a contrastive setting using only 2D binary masks. Unlike prior contrastive segmentation methods [10, 11, 47], which explicitly enforce cluster formation during training and promote statistical consistency of features, TRASE applies contrastive learning directly at the pixel-pair level in a fine-grained rendered feature space, deferring object clustering to a final post-training step. This design enables TRASE to learn a segmentation field with both high temporal consistency and sharp, well-defined object boundaries. It also effectively mitigates common artifacts, such as floaters and out-of-FoV errors, that frequently degrade prior contrastive learning approaches.

To the best of our knowledge, we are the first to introduce a benchmark for segmentation in novel views of dynamic scenes. While established datasets like DAVIS [1, 2, 30, 31] and VOS [40, 43, 44] address dynamic video segmentation, none evaluate segmentation quality in unseen viewpoints. Moreover, our benchmark is significantly larger than the evaluation protocol suggested in [22], as our test sequences comprise single- and multi-view scenes and captures real-world scenarios. Beyond achieving state-of-the-art segmentation accuracy, TRASE is tailored for real-time interactive scene editing. Our compact 32-dimensional feature space requires minimal post-processing and integrates seamlessly into object manipulation tasks such as style transfer, recoloring, composition, and removal.

Our contributions can be summarized as follows:

- We propose TRASE, a new approach for multi-view consistent segmentation of dynamic scenes without tracking labels. To achieve this, we design a novel contrastive learning objective based on differentiable feature rendering and contrastive learning, ensuring segmentation stability across time and views.
- We offer a broad benchmark for segmentation in novel views of dynamic scenes, which encompass a broad spectrum of camera setups and captured scenarios.
- We extensively evaluate our method on five dynamic datasets and achieve state-of-the-art segmentation performance in both single- and multi-view settings.
- We demonstrate the applicability of our feature space to several scene editing tasks, including object removal, style transfer, and scene composition.

2. Related Works

2.1. 3D Segmentation

In static scenes. With the emergence of various vision foundation models, such as Segment Anything Model (SAM) [20], DINO [3], and DINOv2 [29], researchers start to integrate the semantics from these foundation models into their reconstruction. SA3D [5] is the pioneering method that extends 2D SAM masks across images into 3D consistent object masks. GARField [19] introduces hierarchical grouping, leveraging the physical scales of 2D masks to group objects of different sizes and represent scenes at different granularity scales. Gaussian Grouping [46] first proposes to utilize consistent object IDs generated by video tracker DEVA [6] to render object IDs into camera views. Concurrent work such as SAGA [4] utilizes the masks generated from SAM, combined with scales obtained from 3D data inspired by GARField, to segment objects of various sizes. GAGA [26] proposes a mask group identity assignment technique using a 3D-aware memory bank to compensate for inaccurate semantic labels and use group IDs as pseudo-labels for identity encoding. Our model implicitly

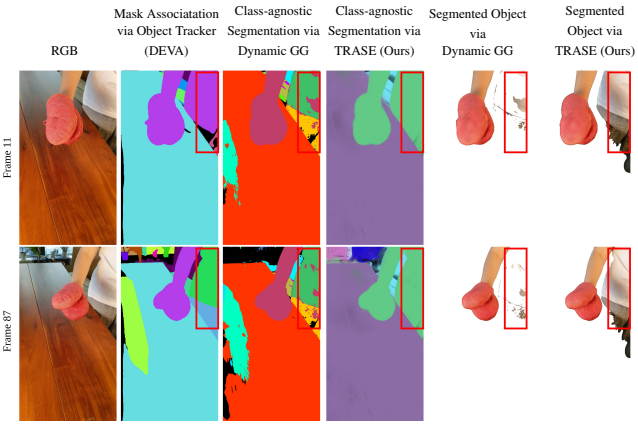

Figure 3. Example failure case of the DEVA video tracker [6], where colors indicate different object IDs. Due to the inconsistent visibility of the human torso, DEVA produces unreliable object masks, leading to noisy class-agnostic segmentation and poor supervision for dynamic Gaussian Grouping [46] (GG).

learns multi-view consistent semantic Gaussian features using our designed contrastive loss in a single training stage without identity assignments.

In dynamic scenes. To the best of our knowledge, only two prior works, SA4D [17], and DGD [22], address the segmentation task in dynamic 3DGS. DGD extends the rasterization pipeline of 3DGS to be able to render the semantic features of each Gaussian. The rendered Gaussian feature map is then trained with the supervision of DINOv2 [29] or CLIP [32] to mimic the output from their feature extractors. The drawbacks of such supervision are longer training times and bigger 3DGS models, as the dimensions of the features generated by DINOv2 and CLIP are quite large (384 and 512), and the Gaussian features are also stored. Following the identity encoding concept from Gaussian Grouping and its derivatives [8, 26], SA4D introduces a temporal identity feature field. As the method relies on consistent object IDs for supervision, it is sensitive to temporal tracking inaccuracies due to occlusion or fast motion. A concurrent work, Split4D [15], follows several proposed design choices, including single-frame supervision and a contrastive learning objective, and utilizes a recent 4D reconstruction model to achieve 4D segmentation. Our TRASE does not rely on tracking labels and effectively handles single- and multi-view data with fast rendering times, minimal post-processing, and low memory footprint.

2.2. Contrastive Learning for 3D Segmentation

A number of recent works define multi-view 3D segmentation as a contrastive learning objective. Contrastive Gaussian Clustering (CGC) [10] training aims to maximize intra-cluster similarity in the 2D rendered feature space by enforcing the feature's proximity with its corresponding cluster's mean feature and ensuring its spatial consistency. OmniSeg3D [47] follows a similar CGC objective and supervises 3D point clustering using SAM-based image patches. ContrastiveLift [11] proposes a slow-fast contrastive fusion strategy, where a slow network aggregates global object-level information while a fast network refines per-point features. OpenGaussian [39] enforces intra-cluster similarity and inter-cluster dissimilarity of rendered features and further trains their discretized version. Split4D [15] utilizes the same objective as [10] and additionally enforces the features to align with DINOv2 [29] in the early training stage. Our TRASE proposes a contrastive objective based on soft sample mining of pixel pairs, which focuses on only imperfect features and allows us to learn the segmentation field effectively in 4D. No costly distillation is required as we supervise the training with binary 2D masks.

2.3. Scene Editing Applications

Learning a semantically aware latent space enables interactive scene editing at the object level. Existing radiance-field editing methods are either restricted to single objects [14, 38] or rely on per-object scene decompositions [21, 42], which limit editing to a fixed set of instances. Other approaches, such as DINO [29], distill features into volumetric fields [37], but their implicit representations require expensive re-rendering for consistency, hindering real-time interaction. Gaussian-based methods [18] alleviate NeRF runtime issues via explicit point-based primitives: SAGA [46] enables local Gaussian editing, and Feature3DGS [49] leverages 2D foundation models like SAM [33] and LSeg [23] for promptable, language-guided 3D edits, yet all focus on static scenes. In contrast, our method edits dynamic objects with temporally consistent results and supports diverse tasks, including style transfer, object removal, and composition [22]. While SA4D [17] offers semantic scene editing, it requires prior object labels and limited interaction. In contrast, our framework provides a user-friendly graphical interface that supports simple text prompts and mouse-based selection.

3. Method

In this section, we introduce our novel TRASE. As illustrated in Fig. 2, the pipeline comprises two main components: dynamic geometry reconstruction (Sec. 3.1) and Gaussian feature learning (Sec. 3.2). We adopt the Deformable-3DGS [45] pipeline to reconstruct the 4D scene. Once the 4D reconstruction is learned, we freeze this representation and proceed with Gaussian feature learning using SAM [20] masks and our novel contrastive learning objective in the rendered feature space.

3.1. Dynamic Geometry Reconstruction

We follow the approach proposed in [45] and describe it briefly for completeness. Unlike Dynamic3DGS [25]

model, where the reconstruction is stored per frame, resulting in high memory consumption, an MLP learns per-Gaussian deformation $(\delta \boldsymbol{x}_i, \delta \boldsymbol{r}_i, \delta \boldsymbol{s}_i)$ with respect to the static canonical space $\mathcal{G}_c$, which is defined as $\mathcal{G}_c = \{\boldsymbol{x}_i \in \mathbb{R}^3, \boldsymbol{r}_i \in \mathbb{R}^4, \boldsymbol{s}_i \in \mathbb{R}^3, \alpha_i \in \mathbb{R}, \boldsymbol{sh}_i \in \mathbb{R}^{3\times(D_{max}+1)^2}\}$ for $i = 1, ..., N$. $\boldsymbol{x}_i$ is the center position of the i-th Gaussian, $\boldsymbol{r}_i$ is the quaternion representing the rotation, $\boldsymbol{s}_i$ is the scaling vector, α_i refers to the opacity, and $\boldsymbol{sh}_i$ is the Spherical Harmonic Coefficients encoding the color information ($D_{max} = 3$) as per [45].

The resulting Gaussians $\mathcal{G}_t$ at timestamp t are defined as

$$\mathcal{G}_t = \{\boldsymbol{x}_i + \delta \boldsymbol{x}_i, \boldsymbol{r}_i + \delta \boldsymbol{r}_i, \boldsymbol{s}_i + \delta \boldsymbol{s}_i, \alpha_i, \boldsymbol{sh}_i\}_{i=1,...,N}. \quad (1)$$

Once $\mathcal{G}_t$ is obtained, we proceed with the standard rasterization pipeline [18] to render the image $\boldsymbol{I}_r$. The color loss $\mathcal{L}_{\text{color}}$ is computed with the ground truth image $\boldsymbol{I}_{\text{GT}}$ as

$$\mathcal{L}_{\text{color}} = (1-\lambda)\mathcal{L}_1(I_{\text{GT}}, I_r) + \lambda \mathcal{L}_{\text{D-SSIM}}(I_{\text{GT}}, I_r), \quad (2)$$

where λ refers to the weighting parameter for the structural similarity loss term.

3.2. Gaussian Feature Learning

Summary. Our feature learning augments each 3D Gaussian with a 32-dimensional feature vector, enabling segmentation without explicit tracking labels. It is worth noting that, in contrast to view-dependent spherical encoding, our semantic feature vector is consistent across different views and time. Thanks to differentiable feature rendering [49] and SAM [20] masks, we construct and explicitly enforce pixel-mask associations in the Gaussian feature field. Our novel contrastive learning objective is designed to be efficient by sub-sampling and effective by integrating SAM [20] masks. Using only 2D guidance and without any object labels, Gaussians are trained to separate per-pixel rendered features from different segments and align those within the same segment. To achieve this, we design a soft-mining objective. Unlike hard mining, which selects pairs strictly by similarity, soft mining uses a relaxed, existential rule that increases pixel coverage and makes training less sensitive to noisy feature similarities. Finally, the learned features are globally clustered, producing multi-view consistent segmentation.

Our notation. A generated set of SAM masks $\boldsymbol{\mathcal{M}}_{\text{SAM}}$ comprise M' binary masks of a given image $\boldsymbol{I}_{\text{GT}}$, where each mask $\boldsymbol{M}_i$ corresponds to a different segment. With this, we define the pixel-mask correspondence vector $\boldsymbol{y}_i$, which captures information across all binary masks in $\boldsymbol{\mathcal{M}}_{\text{SAM}}$ for a certain pixel coordinate (u_i, v_i). This vector is similar to a one-hot encoding technique; however, there is no guarantee that a pixel belongs to a single mask due to 2D segmentation inaccuracies.

The similarities of pixel-mask vectors for a pair of pixels (i, j) can be retrieved by computing the Gram matrix of their stacked matrix $\boldsymbol{Y} \in \{0,1\}^{N_p \times M'}$, where N_p refers to the number of sampled pixels. All entries of the resulting Gram matrix are then bounded to either 0 (negative pair) or 1 (positive pair) to form a mask-based correspondence matrix $\boldsymbol{C} \in \{0,1\}^{N_p \times N_p}$ among sampled pixels, formally defined as

$$\boldsymbol{C}(i,j) = \begin{cases} 1, & \text{if} \quad \boldsymbol{y}_i^T \cdot \boldsymbol{y}_j > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Analogous to the binary mask-based Gram matrix $\boldsymbol{C}$, we compute a feature-based matrix $\boldsymbol{C}_F$, capturing the pairwise similarities between the rendered features $\{\boldsymbol{I}_F(u_i, v_i)\}_{i=1,...,N_p}$. Following common practices [4], we consider a smooth version of Gaussian features, as it is more robust than considering individual features, which might have high similarity scores corresponding to unrelated objects. The smooth Gaussian feature is computed as the mean feature of its K nearest neighbors [7] based on the Gaussian's 3D position.

Our learning objective. To robustly learn multi-view coherent feature embeddings, we adopt a soft-mined contrastive learning objective. Unlike standard hard mining, which selects positive or negative pairs strictly based on their individual similarity values, soft mining employs an existential criterion: a positive pixel pair (i,j) is considered into final positive loss computation $\mathcal{L}_{\text{pos}}$ if there exists any positive pair involving i whose similarity falls below the positive threshold τ_p. Similarly, a pair is included in the negative loss if there exists any negative pair involving i whose similarity exceeds the negative threshold τ_n. By design, the soft-mined strategy broadens pixel coverage during training, in contrast to naive hard sampling that typically concentrates on a sparse set of imperfect pixel pairs and leaves many pairs untrained. We further restrict the selection to non-redundant pairs by considering only the upper-triangular portion of the matrices ($i < j$). Formally, the soft-mined masks are defined as

$$\boldsymbol{M}^+_{ij} = \left(\exists k \ : \ \boldsymbol{C}_{ik} = 1 \wedge \boldsymbol{C}_{\boldsymbol{F},ik} < \tau_p\right) \ \wedge \ (i < j), \quad (4)$$

$$\boldsymbol{M}^-_{ij} = \left(\exists k \ : \ \boldsymbol{C}_{ik} = 0 \wedge \boldsymbol{C}_{\boldsymbol{F},ik} > \tau_n\right) \ \wedge \ (i < j). \quad (5)$$

The positive and negative losses are then computed as weighted sums over these masks:

$$\mathcal{L}_{\text{pos}} = -\frac{1}{Z} \sum_{i,j} \boldsymbol{M}^+_{ij} \boldsymbol{W}_{ij} \boldsymbol{C}_{\boldsymbol{F},ij}, \quad (6)$$

$$\mathcal{L}_{\text{neg}} = \frac{1}{Z} \sum_{i,j} \boldsymbol{M}^-_{ij} \boldsymbol{W}_{ij} \, \text{ReLU}(\boldsymbol{C}_{\boldsymbol{F},ij}), \quad (7)$$

where Z is a normalization factor (e.g., the total number of pixel pairs), $\boldsymbol{W} \in \mathbb{R}^{N_p \times N_p}$ is a pixel-weight matrix similar to [4], and $\text{ReLU}(\cdot)$ ensures that only positive contributions

Method	NeRF-DS-Mask		HyperNeRF-Mask		Neu3D-Mask		Immersive-Mask		Technicolor-Mask	
	mIoU↑	mAcc↑	mIoU↑	mAcc↑	mIoU↑	mAcc↑	mIoU↑	mAcc↑	mIoU↑	mAcc↑
GG [46]	0.8351	0.9729	0.8341	0.9771	0.8864*	0.9932*	0.7673*	0.9776*	0.7979*	0.9800*
SAGA [4]	0.8072	0.9644	0.7660	0.9579	0.6941	0.9516	0.7395	0.9747	0.7913	0.9792
DGD [22]	0.7125	0.9551	0.8297	0.9777	0.7721	0.9851	0.7981	0.9850	0.7791	0.9728
SA4D [17]	0.6740	0.9360	0.7371	0.9616	0.8832*	0.9391*	0.7987*	0.9835*	0.8271*	0.9734*
CGC [35]	0.8100	0.9718	0.8157	0.9768	0.8754	0.9927	0.8856	0.9925	0.8806	0.9812
OpenGaussian [39]	0.6939	0.9564	0.6311	0.9411	0.8178	0.9899	0.7120	0.9793	0.8390	0.9694
TRASE (Ours)	**0.8768**	**0.9831**	**0.8663**	**0.9845**	**0.9022**	**0.9945**	**0.9234**	**0.9945**	**0.9308**	**0.9917**

Table 1. Segmentation accuracy of 4D segmentation methods on our benchmark. TRASE outperforms the baselines on average mIoU and mAcc. The quantitative results for each sequence can be found in the supplementary. *: GG and SA4D need consistent object IDs from a video tracker for supervision [6]. To evaluate on the multi-view camera dataset, we train models on a camera closest to the test view.

are considered for negative pairs. The final loss is then defined in Eq. (8) with an extra regularization term to stabilize the feature learning.

$$\mathcal{L} = \mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{neg}} + \mathcal{L}_{\text{reg}}, \tag{8}$$

$$\mathcal{L}_{\text{reg}} = \left(1 - \frac{1}{HW}\sum_{i=1}^{HW} \|\boldsymbol{I}_F(u_i, v_i)\|_2\right)^2 \tag{9}$$

Inference. To enable interactive applications with the learned 4D representation, we deploy DBSCAN algorithm [9] to cluster the Gaussian features into several groups and generate the so-called Gaussian clusters. In contrast to the K-Means algorithm [24, 27], it does not require any prior knowledge of the number of clusters in a scene. As can be seen in Fig. 2, clusters represent distinct objects, which verifies accurate alignment of our learned feature space with the object-level semantic field.

We further observe that the object boundaries can be effectively refined by applying a simple similarity threshold, which prunes Gaussians with features that are less similar to the corresponding cluster's mean. This simple technique enhances the method's performance in challenging segmentation scenarios, such as those involving occlusions and affected by sparse observations.

4. Experimental Results

We first provide details of our proposed 4D segmentation benchmark in Sec. 4.1 and outline the evaluation protocol in Sec. 4.2 . We present segmentation results of our model, along with both quantitative and qualitative comparisons to other related works [4, 22, 46] in Sec. 4.3. Last but not least, we demonstrate the capabilities of our learned semantic representation on a number of downstream tasks, including scene editing and selection via our designed graphical user interface (Sec. 6). Please refer to the supplementary material for implementation details.

4.1. 4D Segmentation Benchmark

We manually annotate dynamic sequences of existing novel-view synthesis datasets and present their segmentation-aware versions: *NeRF-DS-Mask*, *HyperNeRF-Mask*, *Neu3D-Mask*, *Immersive-Mask*, and *Technicolor-Mask*. Utilizing the powerful SAM2 model [33], which provides consistent object masks across video sequences, we define the objects of interest for each sequence to evaluate their segmentation performance in our model and manually refine them. Qualitative examples of the augmented sequences, together with a description of the raw datasets, can be found in the supplementary material.

4.2. Evaluation Protocol and Baselines

We compare our model with SAGA [4], Gaussian Grouping (GG) [46], SA4D [17], and DGD [22]. Additionally, we add evaluation with the contrastive learning methods such as Contrastive Gaussian Clustering (CGC) [10] and OpenGaussian [39]. Note that SAGA, Gaussian Grouping, CGC, and OpenGaussian are designed for static scenes and do not include a module to encode temporal information. To evaluate them on the 4D segmentation task, we integrate our 4D reconstruction backbone into their pipeline, extending their approach to effectively handle dynamic scenes. GG and SA4D require video object tracking labels for their supervision. To enable evaluation of the multi-view sequences from our benchmarks, we train the models on the camera closest to the test view (cam5 for Neu3D, cam2 for Immersive, and cam1 for Technicolor). After training, we manually perform click prompts in the rendered novel views of each scene trained by different approaches to select the objects of interest defined in the benchmark. Apart from the quantitative results, we also present qualitative results from various benchmarks, as certain insights cannot be fully captured through numerical evaluations alone. Following common practices, we use Mean Intersection over Union (mIoU) and Mean Pixel Accuracy (mAcc) to quantitatively evaluate the performance of the models.

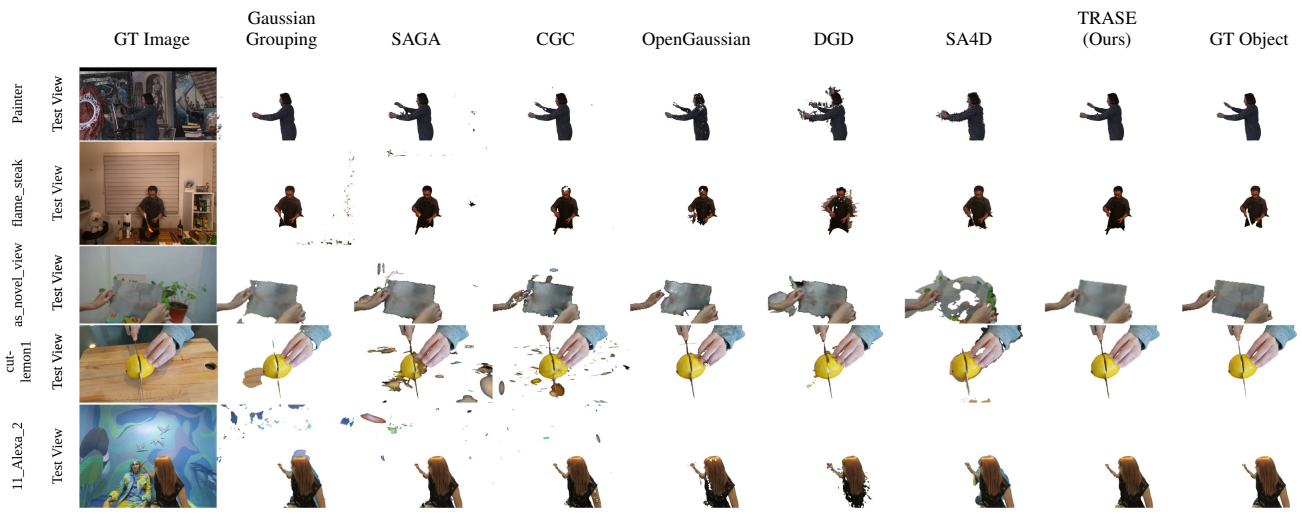

Figure 4. Segmentation qualitative results. While Gaussian Grouping [46] and SAGA [4] suffer from spurious Gaussians, our method demonstrates crisp and floater-free segmentation. Object boundaries in SA4D [17] and DGD [22] segmentations are not tight and capture part of the background while rendering. OpenGaussian [39] is not able to fully capture the whole objects in its segmentation. Our model consistently demonstrates superior segmentation quality and crisp object masks.

4.3. Segmentation Results

	Preproc	Training	Total (min)	Storage (MB)
DGD [22]	0.08	103	103	285
TRASE (Ours)	16	16	**32**	**60**

Table 2. Storage and time comparison between DGD [22] and TRASE on NeRF-DS [41]. TRASE achieves over 3× faster training time compared to DGD while only requiring 4.5× less memory in storage due to its compact Gaussian features.

As shown in Tab. 1, TRASE achieves state-of-the-art performance across all benchmarks, surpassing other segmentation methods. Please refer to the supplementary material for details on each sequence of the benchmark. Further, we select some sequences to illustrate the quality of the segmentation of each model in Fig. 4. These comparisons reveal that TRASE qualitatively surpasses existing models and exhibits multi-view consistent segmentation performance. Notably, TRASE maintains strong segmentation consistency to test camera views in multi-view sequences such as *flame_steak*, *Painter*, and *11_Alexa_2*. This robust performance can be attributed to the compact Gaussian features learned through the proposed Gaussian feature training. In contrast, DGD segments objects by comparing cosine similarity scores between features retrieved via click prompts. However, determining a suitable threshold that generalizes across various scenes is neither straightforward nor intuitive. TRASE employs clustering based on the Gaussian features to group the Gaussians into several segments. As this operation is done in 3D space, TRASE achieves cross-view consistency and doesn't need to associate objects from different views. This advantage is evident in *cut-lemon1* example in Fig. 4, where wrongly associated object IDs can cause segmentation failure for [46].

5. Ablation Studies

5.1. Time and Storage Analysis

We perform a comprehensive time and storage analysis of TRASE and DGD [22] using an RTX 3080 and an Intel i7-12700 on the NeRF-DS [41] dataset. The results are shown in Tab. 2, where "Pre-Proc" refers to the generation of DINOv2 384-dim features for DGD and SAM masks for TRASE. Rendering 384-dim features into 2D is computationally inefficient, significantly slowing down DGD's training process. TRASE uses 32-dim Gaussian features, which can be rendered much more efficiently. As a result, TRASE achieves over **3**× faster total time than DGD while only requiring **4.5**× less space in storage.

5.2. Soft Sample Mining

We visually demonstrate the effectiveness of our soft-mined pixel-pair sampling strategy in comparison to other contrastive objectives: sampling all positive–negative pairs (all) and sampling only pairs that individually satisfy the thresholds (hard). We include contrastive-based baselines such as CGC [10] and OpenGaussian [39] for completeness. By investigating the out-of-FoV regions, i.e., 3D regions outside the observable space of train and test views, we observed floaters in the segmentation by our method without soft selection and other contrastive baselines, as shown in Fig. 5. Our mining approach effectively eliminates these floaters, leading to more stable and reliable segmentation. It is worth noting that no post-processing filtering is applied to ensure

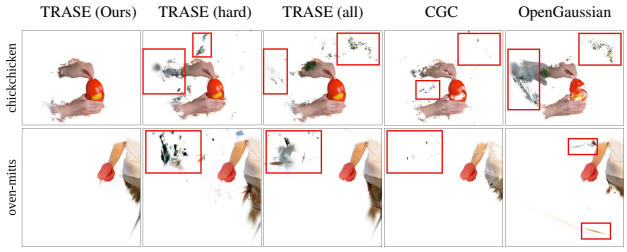

Figure 5. Qualitative results of segmented objects in *chickchicken* and *oven-mitts* sequence in out-of-FoV viewpoint. TRASE with soft sample mining (Ours) exhibits superior performance over TRASE with hard / all mining, CGC [10], and OpenGaussian [39].

Method	HyperNeRF-Mask		Immersive-Mask		Technicolor-Mask	
	mIoU↑	mAcc↑	mIoU↑	mAcc↑	mIoU↑	mAcc↑
TRASE (multi-frame)	0.8636	0.9833	0.8968	0.9933	**0.9309**	0.9913
TRASE (Ours)	**0.8663**	**0.9845**	**0.9234**	**0.9945**	0.9308	**0.9917**

Table 3. Ablation study on single-frame mask supervision. The multi-frame baseline propagates masks between consecutive frames using Gaussian motion flow to impose temporal constraints. While this increases training time by a factor of **2**, it does not provide significant additional supervision, confirming the effectiveness of single-frame mask supervision.

a fair comparison.

5.3. Single-frame Mask Supervision

Inspired by classical tracking methods [13] and recent work linking 3DGS dynamics to pixel motion [12], we investigate the impact of incorporating temporal constraints on our single-frame mask supervision. Specifically, we compare our standard single-frame approach with a temporal extension, where masks are propagated across consecutive frames to encourage segmentation consistency over time. To implement this, we estimate optical flow between consecutive training timestamps t_s and t_{s+1} using differentiable rendering with accumulated 2D Gaussian displacements [12], and warp mask pixels from t_s to t_{s+1}. Quantitative results are presented in Tab. 3. As observed, the multi-frame baseline does not provide significant improvements over the single-frame supervision, despite the doubled training cost, as an additional 2D flow and all associated parameters are rendered. This validates our design choice to rely on single-frame mask supervision, which achieves strong segmentation performance efficiently.

5.4. Post-processing Filtering

We provide an ablation study of our method with respect to the filtering step, which removes Gaussians whose features are too dissimilar to the object's mean. As shown in Tab. 4, the proposed filtering step enhances the segmentation quality of our method. Particularly, artifacts on the object's boundaries are effectively removed, mitigating the inaccuracies in input masks or the effects of occlusions.

Method	Filtering	NeRF-DS-Mask		HyperNeRF-Mask		Neu3D-Mask	
		mIoU↑	mAcc↑	mIoU↑	mAcc↑	mIoU↑	mAcc↑
SAGA [4]	✘	0.8072	0.9644	0.7660	0.9579	0.6941	0.9516
	✓	0.8421	0.9778	0.8223	0.9779	0.8026	0.9853
TRASE	✘	0.8760	0.9829	0.8368	0.9790	0.8738	0.9923
TRASE (Ours)	✓	**0.8768**	**0.9831**	**0.8663**	**0.9845**	**0.9022**	**0.9945**

Table 4. Effect of filtering spurious Gaussians for SAGA [4] and TRASE. Despite the improved baseline's performance, TRASE maintains the superior segmentation quality due to its spatially consistent feature space.

Figure 6. Qualitative results for SAGA [4] with (w/) and without (w/o) our proposed filtering. Even though our filtering step can filter most of the artifacts from SAGA. Their resulting segmented objects are still not optimal compared to TRASE.

We further demonstrate the advantage of our learned feature field by showing that while the post-processing step enhances segmentation accuracy, it is not the sole factor driving our superior performance. We also note that this technique is applicable only to certain baselines such as SAGA [4]. For example, methods like CGC [10] and DGD [22] already incorporate a similarity threshold during inference. OpenGaussian [39] gets the assigned class ID for each Gaussian from their codebook. By contrast, SA4D [17] and Gaussian Grouping [46] learn an identity field, making our post-processing inapplicable to them. In Tab. 4 and Fig. 6, we quantify and illustrate the effectiveness of applying our proposed filtering step to SAGA [4]. However, because the features learned under the baseline's hard-contrastive objective remain suboptimal, the segmentation quality still exhibits more artifacts compared to TRASE.

6. Editing Applications

In addition to accurate segmentation capabilities and multi-view consistent rendered masks, we demonstrate the effectiveness of our learned semantic feature field in a range of downstream tasks. Specifically, we focus on interactive editing applications and develop a graphical user interface (GUI) to grant full user control over the representation. Thanks to its explicit nature and our semantic abstraction to Gaussian clusters, we achieve real-time operation capability in handling many editing tasks. We provide snapshots of the GUI in the supplementary.

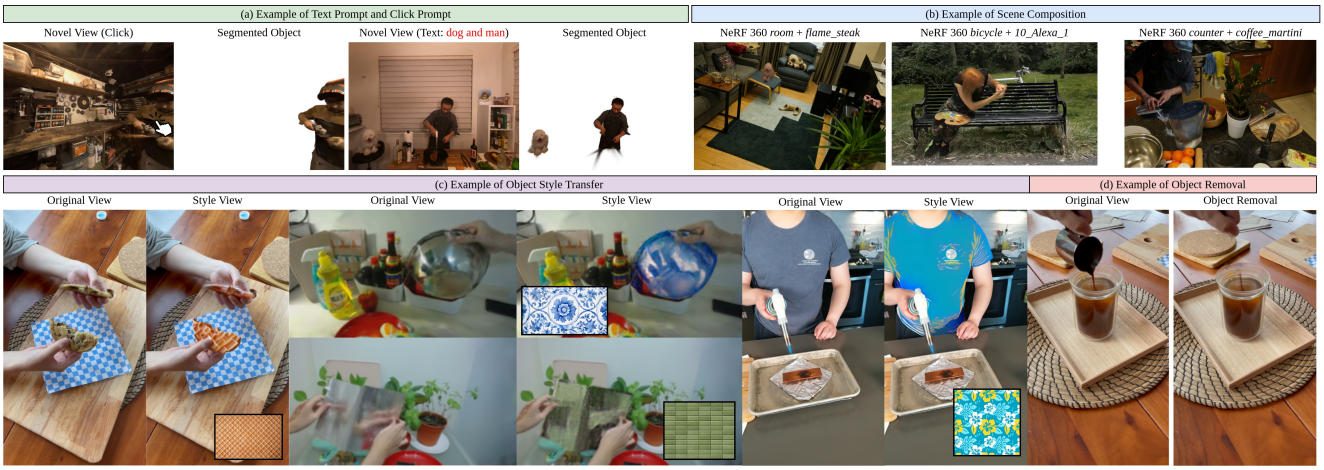

Figure 7. Versatile scene editing applications. (a) The object of interest can be selected by click prompts or text prompts. (b) Scene composition can be done by manipulating the selected Gaussians in another scene. (c) Style transfer of the segmented objects to obtain different textures. (d) Object removal for the selected object.

In addition to simple click prompts, we also support text-based prompts to make object selection even more intuitive. To enhance 3D content editing in dynamic scenes, we integrate style transfer capabilities for the selected objects. By segmenting objects directly in 3D, we facilitate scene composition and object removal through straightforward manipulation of the selected Gaussians. Due to the space limit, we shall provide more qualitative results in the supplementary. In the following, we describe the implementation details of editing tasks based on our semantically-aware feature field.

Text Prompt. Inspired by Gaussian Grouping [46], we employ Grounded SAM [34] to generate a 2D mask based on the text prompt in the novel view. The 2D mask is reprojected into the 3D space with the rendered depth information. Finally, we associate each reprojected point with its nearest Gaussian in the 3D scene and retrieve its corresponding cluster. If the number of associated Gaussians of a given cluster is more than a threshold, the cluster would be selected. The example of text prompts on different sequences is shown in Fig. 7 (a).

Scene Composition. We also perform qualitative results on scene composition. As the segmentation is performed in 3D, we can simply put the selected Gaussians from the dynamic scene into another static or dynamic scene. The scales and coordinate transformation need to be adapted. We show some examples of scene composition in Fig. 7.

Object Style Transfer. We adopt the static Gaussian style transfer from StyleSplat [16] for dynamic scenes. Given a rendered image $\boldsymbol{I}_r$ and a style image $\boldsymbol{I}_s$, we extract their feature maps ($\boldsymbol{F}_r$, $\boldsymbol{F}_s$ respectively) from VGG16 [36] and optimize only the SHs of the Gaussians of the selected cluster with the nearest-neighbor feature matching (NNFM) loss [48]. Fig. 7 (c) illustrates an example of style transfer on the segmented object.

Object Removal. We can also remove objects by simply deleting Gaussians belonging to the specific cluster as illustrated in Fig. 7 (d). It is worth noting that an additional inpainting step may be required, where the newly exposed parts have not been learned from multi-view observations.

7. Conclusion

We introduced a novel framework for dynamic scene understanding, which enables multi-view consistent segmentation without any object tracking supervision. Our TRASE effectively combines dynamic 3D Gaussian Splatting 3DGS [45] and 2D SAM [20] masks in a contrastive learning objective, which lifts semantic information into 3D space and learns expressive Gaussian features based on soft sample mining. This results in cross-view consistency when rendering segmented objects, enhancing the quality and coherence of object segmentation across different views. Evaluated on various novel-view datasets, TRASE shows superior performance both quantitatively and qualitatively. We further demonstrate the effectiveness of the learned feature field on downstream editing tasks such as point and text prompts, style transfer, object removal, and scene composition. Our approach sets a strong foundation for further research into dynamic scene understanding and scene editing, especially in complex multiview scenarios. We intend to release our code and benchmarks for future developments.

Acknowledgements. This work was supported by the ERC Advanced Grant "SIMULACRON" (agreement #884679), the GNI Project "AI4Twinning", and the DFG project CR 250/26-1 "4DYoutube". Yan was supported by the Anhui Provincial Natural Science Foundation (Grant No. 2508085MF142).

References

[1] Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv:1803.00557*, 2018. 2

[2] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv:1905.00737*, 2019. 2

[3] 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*, pages 9650–9660, 2021. 2

[4] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. *arXiv preprint arXiv:2312.00860*, 2023. 1, 2, 4, 5, 6, 7

[5] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with nerfs. *Advances in Neural Information Processing Systems*, 36:25971–25990, 2023. 2

[6] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1316–1326, 2023. 2, 3, 5

[7] Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. 4

[8] Bin Dou, Tianyu Zhang, Yongjia Ma, Zhaohui Wang, and Zejian Yuan. Cosseggaussians: Compact and swift scene segmenting 3d gaussians. *arXiv preprint arXiv:2401.05925*, 2024. 3

[9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996. 2, 5

[10] Myrna C Silva et al. Contrastive gaussian clustering: Weakly supervised 3d scene segmentation. *arXiv preprint arXiv:2404.12784*, 2024. 2, 3, 5, 6, 7

[11] Yash Bhalgat et al. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. In *NeurIPS*, 2023. 2, 3

[12] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. 2024. 7

[13] Mariia Gladkova, Nikita Korobov, Nikolaus Demmel, Aljoša Ošep, Laura Leal-Taixé, and Daniel Cremers. Directtracker: 3d multi-object tracking using direct image alignment and photometric bundle adjustment. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3777–3784. IEEE, 2022. 7

[14] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020. 3

[15] Yongzhen Hu, Yihui Yang, Haotong Lin, Yifan Wang, Junting Dong, Yifu Deng, Xinyu Zhu, Fan Jia, Hujun Bao, Xiaowei Zhou, et al. Split4d: Decomposed 4d scene reconstruction without video segmentation. *ACM Transactions on Graphics (TOG)*, 44(6):1–15, 2025. 3

[16] Sahil Jain, Avik Kuthiala, Prabhdeep Singh Sethi, and Prakanshul Saxena. Stylesplat: 3d object style transfer with gaussian splatting, 2024. 8

[17] 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. 1, 3, 5, 6, 7

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 3, 4

[19] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21530–21539, 2024. 2

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, Piotr Dollar, and Ross B Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 3, 4, 8

[21] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 3

[22] Isaac Labe, Noam Issachar, Itai Lang, and Sagie Benaim. Dgd: Dynamic 3d gaussians distillation, 2024. 1, 2, 3, 5, 6, 7

[23] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 3

[24] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 5

[25] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 1, 3

[26] Weijie Lyu, Xueting Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group any gaussians via 3d-aware memory bank. *arXiv preprint arXiv:2404.07977*, 2024. 2, 3

[27] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967. 5

[28] 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. 1

[29] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 1, 2, 3

[30] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 2

[31] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 2

[32] 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*, pages 8748–8763. PMLR, 2021. 3

[33] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 5

[34] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 8

[35] Myrna C. Silva, Mahtab Dahaghin, Matteo Toso, and Alessio Del Bue. Contrastive gaussian clustering: Weakly supervised 3d scene segmentation. *https://arxiv.org/abs/2404.12784*, 2024. 5

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 8

[37] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *2022 International Conference on 3D Vision (3DV)*, pages 443–453. IEEE, 2022. 3

[38] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022. 1, 3

[39] 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, 2025. 3, 5, 6, 7

[40] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas S. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *CoRR*, abs/1809.03327, 2018. 2

[41] Zhiwen Yan, Chen Li, and Gim Hee Lee. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8295, 2023. 6

[42] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021. 3

[43] Linjie Yang, Yuchen Fan, and Ning Xu. The 2nd large-scale video object segmentation challenge - video object segmentation track, 2019. 2

[44] Linjie Yang, Yuchen Fan, and Ning Xu. The 4th large-scale video object segmentation challenge - video object segmentation track, 2022. 2

[45] 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*, pages 20331–20341, 2024. 1, 2, 3, 4, 8

[46] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. 2024. 1, 2, 3, 5, 6, 7, 8

[47] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024. 2, 3

[48] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, pages 717–733. Springer, 2022. 8

[49] 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*, pages 21676–21685, 2024. 3, 4