Efficient Decoupled Feature 3D Gaussian Splatting via Hierarchical Compression

Zhenqi Dai[1] Ting Liu[1,2*] Yanning Zhang[1]
[1]ASGO, School of Computer Science, Northwestern Polytechnical University
[2]Shenzhen Research Institute of Northwestern Polytechnical University
daizq@mail.nwpu.edu.cn, liuting@nwpu.edu.cn, ynzhang@nwpu.edu.cn

Abstract

Efficient 3D scene representation has become a key challenge with the rise of 3D Gaussian Splatting (3DGS), particularly when incorporating semantic information into the scene representation. Existing 3DGS-based methods embed both color and high-dimensional semantic features into a single field, leading to significant storage and computational overhead. To mitigate this, we propose Decoupled Feature 3D Gaussian Splatting (DF-3DGS), a novel method that decouples the color and semantic fields, thereby reducing the number of 3D Gaussians required for semantic representation. We then introduce a hierarchical compression strategy that first employs our novel quantization approach with dynamic codebook evolution to reduce data size, followed by a scene-specific autoencoder for further compression of the semantic feature dimensions. This multi-stage approach results in a compact representation that enhances both storage efficiency and reconstruction speed. Experimental results demonstrate that DF-3DGS outperforms previous 3DGS-based methods, achieving faster training and rendering times while requiring less storage, without sacrificing performance—in fact, it improves performance in the novel view semantic segmentation task. Specifically, DF-3DGS achieves remarkable improvements over Feature 3DGS, reducing training time by 10× and storage by 20×, while improving the mIoU of novel view semantic segmentation by 4%. Code is available at https://github.com/dai647/DF_3DGS.

1. Introduction

With the rapid advancement of 3D vision technology, efficiently and accurately representing and processing 3D scenes has become a critical research topic [1, 2, 5, 29, 41, 46]. Neural Radiance Fields (NeRF) [24] and 3D Gaussian Splatting (3DGS) [16] have emerged as promising techniques for achieving high-quality 3D scene novel view synthesis from multi-view images. However, both methods encapsulate only appearance and geometric details, lacking additional semantic information for the 3D scene, which is essential for downstream tasks such as semantic segmentation and language-guided editing.

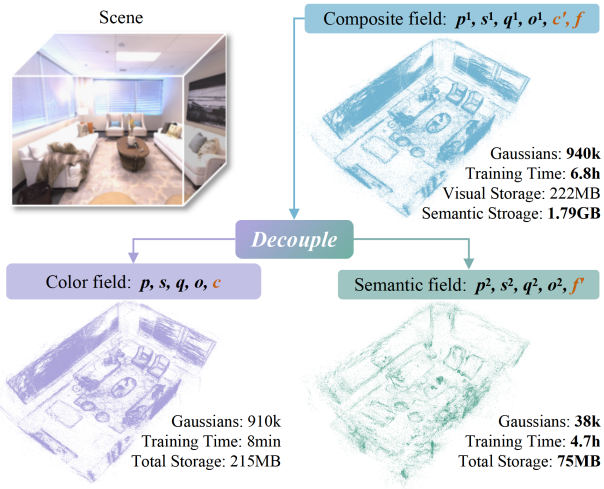

Figure 1. **Illustation of 3D Gauissian point cloud.** By decoupling the semantic field from the color field, the number of Gaussians is significantly reduced, leading to a reduction in the storage of the semantic field from 1.79 GB (semantic features only) to just 75 MB. Each 3D Gaussian is parameterized by its position p, scale s, rotation quaternion q, opacity o, color c, and semantic feature f (dimension=512).

In recent years, NeRF-based 3D semantic understanding methods have attempted to use the neural radiance fields to store additional semantic features for the 3D scene [17, 19, 38]. However, NeRF-based methods are inherently slow in both training and rendering due to the computational complexity of NeRF's ray-marching volume rendering technique. 3DGS as an alternative to implicit radiance field representations, explicitly represents the field

*Ting Liu is the corresponding author.

using 3D Gaussians and has demonstrated superior training and rendering speeds compared to NeRF-based methods. Recent 3DGS-based methods have attempted to distill multi-view semantic features from the 2D pre-trained model into 3DGS for 3D semantic understanding. For instance, Feature 3DGS [47] constructs the semantic field by associating a trainable semantic feature with each Gaussian, which is then optimized alongside the color field. While 3DGS is efficient and fast for high-quality 3D scene representation, embedding high-dimensional semantic features into a massive number of 3D Gaussians can lead to prohibitive storage requirements and significantly reduce the efficiency of both training and rendering.

Some recent works [22, 31, 36] have focused on reducing the dimensions of semantic features for each 3D Gaussian to improve efficiency. However, the large number of 3D Gaussians still limits overall efficiency. As shown in Fig. 1, semantic information in scenes generally contains fewer high-frequency components compared to color information, resulting in a significantly reduced number of 3D Gaussians in the 3DGS semantic field when color information is excluded. Although the 3D Gaussians in the color field cannot share parameters such as p, s, q and o with those in the independent semantic field, the reduction in the number of Gaussians in the semantic field still leads to a substantial decrease in overall storage requirements. Furthermore, the 3D Gaussians in the color and semantic fields exhibit strong positional correlation, facilitating their integration for downstream tasks such as language-guided editing.

Based on this observation, we present Decoupled Feature 3DGS (DF-3DGS): a semantic field distillation technique decoupled from color space based on the 3D Gaussian Splatting framework. Specifically, We remove all color-related parameters, such as SH, from the 3D Gaussians and introduce a low-dimensional, semantic-related latent vector for each Gaussian. All 3D Gaussian parameters are updated during training. This decoupled approach differs significantly from existing 3DGS-based methods [31, 36, 47]. To further improve efficiency by reducing the dimensionality of semantic features, we first propose an adaptive data compression method that employs a novel quantization approach to enhance the compactness of semantic features, with an adaptive codebook generated from the current scene. The clustering effect of the quantization process makes the resulting features more robust, thereby enhancing the overall performance of our model. We then train a scene-specific autoencoder to map these features into a low-dimensional latent space for semantic compression. With fewer 3D Gaussians and low-dimensional semantic features, DF-3DGS significantly reduces storage requirements, accelerates training and rendering, and outperforms previous 3DGS-based methods on the novel view semantic segmentation task.

In summary, our main contributions are as follows:

- We introduce an innovative and efficient feature 3D Gaussian Splatting method for 3D semantic field reconstruction, decoupled from the color space. This approach eliminates the need for color information during training and significantly reduces the number of 3D Gaussians required.
- We introduce a hierarchical compression strategy that first employs a novel quantization approach to adaptively extract core semantic features from the current scene and enhance the compactness of semantic representations. Then, we train a scene-specific autoencoder based on the extracted core semantic features to further compress the semantic feature dimensions.
- Experimental results highlight the remarkable effectiveness of DF-3DGS, achieving substantial improvements over prior 3DGS-based methods. DF-3DGS not only dramatically reduces storage requirements and accelerates both training and rendering, but also consistently outperforms these methods in the novel view semantic segmentation task.

2. Related work

2.1. 3D Gaussian Splatting

Fast reconstruction and real-time rendering have consistently been important goals in 3D representation techniques. NeRF [24] has demonstrated superior novel view synthesis quality compared to traditional methods [4, 13]. Despite significant efforts to improve optimization and rendering efficiency [6, 25], NeRF-based methods still face challenges related to slow training and rendering speeds.

Recently, 3D Gaussian Splatting has been proposed to model points as 3D Gaussians for 3D scene representation [16], which offers faster reconstruction and rendering speeds while achieving high-quality visual results compare to NeRF-based methods. However, the large number of 3D Gaussians and their associated attributes necessitate effective compression techniques. Some methods consider compressing 3DGS using pruning [10, 11], codebooks [26, 27], and entropy constraints [12], other methods achieve compression by exploring the relations of Gaussians [8, 23].

Inspired by the success of 3D Gaussian Splatting, many researchers works have extended it to other tasks [7, 14, 15, 21, 30, 40, 42–45, 48, 49]. For instance, Hu et al. [14] proposed the GaussianAvatar, an innovative method that efficiently generates realistic human avatars with dynamic 3D appearances from a single video source. Zhou et al. [48] presented a novel text-to-360° 3D scene generation technology. Unlike the aforementioned tasks, this paper primarily focuses on utilizing 3DGS for efficient and accurate scene semantic understanding.

2.2. 3D semantic Field

Early efforts to develop 3D semantic fields included Distilled Feature Fields [17, 19] and Neural Feature Fusion Fields [38]. These approaches are designed to achieve 3D scene semantic consistency by integrating 2D foundation model [3, 20, 28, 32, 33] features from multi-view images into a NeRF. For example, DFF [19] distill LSeg [20] or DINO [3, 28] features across multiple views into a NeRF, whcih enable us to semantically select and edit regions in the radiance field.

Due to the inherent rendering efficiency bottleneck of NeRF, researchers have explored some 3D semantic field reconstruction methods based on 3DGS. Feature 3DGS [47] introduces a general semantic field distillation technique based on 3DGS, which integrats high-dimensional semantic features of SAM [18] or LSeg [20] with 3D Gaussians. However, high-dimensional feature parameters from millions of 3D Gaussians in a scene significantly reduce training and rendering efficiency and require substantial storage resources. Recent methods [22, 31, 36, 47] focus on reducing the semantic feature dimensions to alleviated this issue, but a large number of 3D Gaussians still poses an efficiency limitation.

Additionally, previous 3DGS-based semantic field reconstruction methods are highly coupled with the color field. For example, LangSplat [31] and FMGS [50] perform semantic distillation on the reconstructed color field, Feature 3DGS [47], LEGaussians [36], and CLIP-GS [22] enable each 3D Gaussian to simultaneously learn color and semantic features. In contrast, our method attempts to decouple semantic field reconstruction from color information, meaning that no additional color information is needed for semantic field reconstruction beyond the semantic features of multi-view images. This approach significantly reduces the number of 3D Gaussians, enabling more efficient semantic field reconstruction and rendering. Furthermore, despite the independent semantic field lacking color information, it shows a strong positional correlation with 3D Gaussians in the visual field, making our method suitable for downstream tasks such as language-guided editing.

3. Method

3.1. Overview

Different from existing methods that directly incorporate semantic features into the color field, we propose Decoupled Feature 3D Gaussian Splatting (DF-3DGS), an efficient method for 3D scene semantic representation that separates the color and semantic fields to enhance semantic reconstruction and improve the efficiency. The method can leverage any 2D pre-trained model to extract semantic features from multi-view images, which are then used to reconstruct a 3D semantic field. This field is reconstructed and rendered independently from the color field, reducing both storage and computational complexity. In this work, we specifically utilize LSeg [20], a language-driven zero-shot semantic segmentation model, to extract these semantic features.

To further optimize efficiency, we propose a hierarchical compression approach, which first discretizes the semantic features through quantization and then further compresses them using a scene-specific autoencoder. The quantization process with our proposed adaptive codebook discretizes the feature space by grouping similar semantic features into distinct codewords, thereby compactly representing the semantic information. Subsequently, the scene-specific autoencoder compresses the quantized features into a lower-dimensional latent space, enhancing storage efficiency while preserving key semantic information, ultimately improving overall performance.

3.2. Decoupled Feature 3D Gaussians

3D Gaussian Splatting (3DGS) is a technique primarily used for visual scene reconstruction. In this method, each 3D Gaussian is characterized by several learnable attributes: position $p \in \mathbb{R}^3$, scale $s \in \mathbb{R}^3$, rotation quaternion $q \in \mathbb{R}^4$, opacity $o \in \mathbb{R}$ and color $c \in \mathbb{R}^3$. These parameters represent the spatial and visual properties of the Gaussians in the 3D space, and they can be learned from data to achieve realistic 3D scene representations. Using differentiable rasterization R to render a image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ from a specific camera pose p_{cam}, represented as:

$$\mathbf{I} = R(p, s, q, o, c; p_{cam}). \tag{1}$$

To incorporate semantic information within 3D Gaussian Splatting (3DGS), some methods like Feature 3DGS [47] distill semantic information by embedding a semantic feature attribute f within the 3DGS, enabling it simultaneously render an image $\mathbf{I}$ and a semantic feature map $\mathbf{F}_r$ by volumetric rendering, represented as:

$$\mathbf{I}, \mathbf{F}_r = R(p^1, s^1, q^1, o^1, c', f; p_{cam}). \tag{2}$$

Due to the high-frequency components in the visual information and the sparsity of the semantic information, the number of 3D Gaussians required for the semantic field is much smaller than that for the color field, as shown in Fig. 1. Therefore, we propose decoupling the semantic field from the color field, allowing it to be reconstructed independently, as represented by:

$$\mathbf{F}_r = R(p^2, s^2, q^2, o^2, f'; p_{cam}). \tag{3}$$

Note that, in the Decoupled 3DGS semantic field, the rendered $\mathbf{F}_r$ maintains the same size $H \times W$ as the image. This helps establish positional correspondence of 3D Gaussians between the semantic field and the color field,

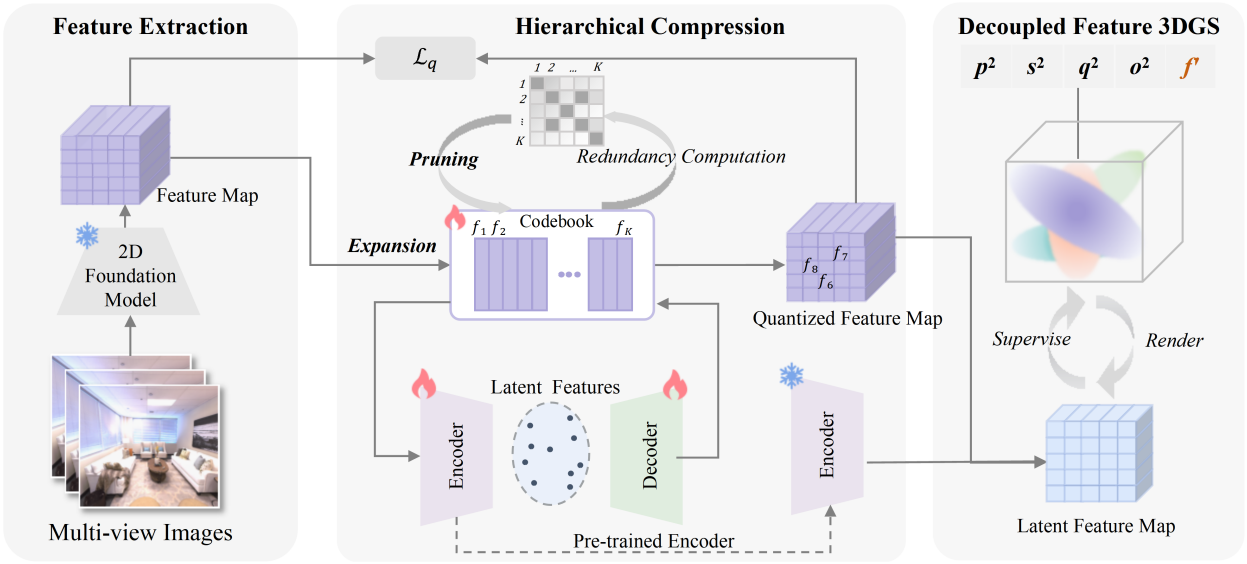

Figure 2. **Overview of DF-3DGS:** We decouple the color and semantic fields to reduce 3D Gaussian usage, followed by hierarchical compression with a novel quantization approach and a scene-specific autoencoder for efficient semantic field reduction.

which is crucial for tasks such as language-guided editing. During training, $\mathbf{F}_r$ is downsampled to the feature map size $H' \times W'$ by bilinear interpolation before computing the L1 loss with the semantic feature map $\mathbf{F}_l$:

$$\mathcal{L}_f = ||\mathbf{F}_l - \mathbf{F}_r||_1 \tag{4}$$

3.3. Hierarchical Compression

Embedding the raw high-dimensional features into a large number of 3D Gaussians significantly reduces 3D Gaussian efficiency. Therefore, we consider compressing these high-dimensional semantic features into a lower dimension before embedding them in 3D Gaussians.

3.3.1. Adaptive Data Compression

Directly training an autoencoder on all multi-view image features faces the challenge of redundant and imbalanced training data, as each semantic contains many highly similar features, and the number of features per semantic in a scene varies greatly. To address this, we propose an adaptive data compression method based on quantization, which can dynamically generate the codebook that adapts to different scenes. Additionally, the clustering effect during the quantization helps alleviate inter-class semantic feature interference, thereby enhancing the robustness and accuracy of the semantic features, and facilitating more accurate scene understanding. Unlike the previous quantization approaches, which require specifying the number of potential semantic categories, our method dynamically adjusts the size of the semantic feature codebook based on the complexity of the scene.

Specially, we construct an adaptively evolving codebook $S = \{f_e \in \mathbb{R}^D \mid e = 1, 2, \ldots, K\}$, where D denotes the dimension of raw features, f_e represents the e-th codeword, and K is the size of codebook S. Similar to VQ-VAE [39], the optimization of S and the quantization of all semantic features extracted from multi-view images are performed simultaneously. In the following, we will detail each of these steps.

Quantization. Given a raw semantic feature $f_{raw} \in \mathbb{R}^D$, to transform it into a quantized version f_t, we perform a max-similarity search within codebook S and use cosine similarity $\cos(\cdot)$ as the similarity measure:

$$t = argmax_{e \leq K} \cos(f_{raw}, f_e). \tag{5}$$

For all features of each image after this quantization procedure, we can obtain a feature index map $M \in \mathbb{R}^{H' \times W'}$.

Optimization. For simplicity, we illustrate with a single image. Given a image feature map $\mathbf{F} \in \mathbb{R}^{H' \times W' \times D}$, we obtain its feature index map M and quantized version $\hat{\mathbf{F}} = S[M] \in \mathbb{R}^{H' \times W' \times D}$ by quantization. Then we optimize S by minimizing the cosine distance loss between $\mathbf{F}$ and $\hat{\mathbf{F}}$. Considering the class imbalance in feature index map M, we compute the cosine distance loss for each class and average them to obtain the final loss $\mathcal{L}_q$.

$$\mathcal{L}_q = \frac{1}{T} \sum_{\tau=1}^{T} \frac{1}{N_\tau} \sum_{i=1}^{H'} \sum_{j=1}^{W'} (1 - \cos(\mathbf{F}_{i,j}, \hat{\mathbf{F}}_{i,j})) \times \delta(M_{i,j}, \tau) \tag{6}$$

Here, T is the class number in M, N_τ is the pixel number of class τ in M, δ is the Kronecker delta function:

$$\delta(a, b) = \begin{cases} 1 & \text{if } a = b, \\ 0 & \text{if } a \neq b. \end{cases} \tag{7}$$

Adaptively Evolving S. For a given scene, the optimal size K of codebook S is not fixed, as the complexity and diversity of semantic features vary across different scenes. Manually estimating K for each scene is inefficient due to the high variability in feature distributions and the computational cost associated with determining an appropriate codebook size for each scenario. To address this issue, we propose a method that can adaptively and dynamically adjust S to different scenes. It consists of the following three parts:

- ***Initialization***: We initialize S by sampling ρ feature points from the feature map of the first image using Farthest Point Sampling [9], where cosine similarity is used as the distance metric to select the most diverse points.
- ***Expansion***: We set a threshold α, when the cosine similarity between $\mathbf{F}_{i,j}$ and its quantized version $\hat{\mathbf{F}}_{i,j}$ falls below α, we expand S by adding $\mathbf{F}_{i,j}$ as a new semantic feature to it:

$$S = S \cup \{\mathbf{F}_{i,j}\}, \quad \text{if} \quad \cos(\mathbf{F}_{i,j}, \hat{\mathbf{F}}_{i,j}) < \alpha. \tag{8}$$

  To maintain optimization efficiency and prevent excessive feature expansion, we impose a maximum limit K_{max}, ensuring that new features are only added when the size K of S is below K_{max} this threshold.
- ***Pruning***: To avoid redundancy in S during updates, we set a threshold β. When the cosine similarity between f_e and f_g ($\forall f_e, f_g \in S, e < g$) exceeds β, f_g is deemed redundant and pruned from S:

$$S = S \setminus \{f_g\}, \quad \text{if} \quad \cos(f_e, f_g) > \beta. \tag{9}$$

Both the *Expansion* and *Pruning* are processed in batches and performed simultaneously with quantization and optimization.

3.3.2. Semantic Compression

After constructing the codebook S, we use a scene-specific autoencoder to further compress the dimension of the high-dimensional semantic features $f_e \in S$.

Specifically, an encoder E is used to map the D-dimensional semantic features $f_e \in \mathbb{R}^D$ to d-dimensional latent features $f_e^l = E(f_e) \in \mathbb{R}^d$, where the small size of codebook S ensures that $d \ll D$ is feasible. We then use a decoder Ψ to reconstruct the f_e from the compressed features f_e^l. The training loss $\mathcal{L}_{ae}$ of the autoencoder is as follows:

$$\mathcal{L}_{ae} = ||\Psi(E(f_e)) - f_e||_2. \tag{10}$$

After compressing the semantic features in S using the autoencoder described above, we obtain a low-dimensional semantic feature codebook $S' = \{f_e^l \in \mathbb{R}^d \mid e = 1, 2, \ldots, K\}$. Based on S', we can generate the latent feature map $\mathbf{F}_l = S'[M] \in \mathbb{R}^{H' \times W' \times d}$ of each image by their feature index map M. This latent feature map $\mathbf{F}_l$ is then used to train the Decoupled Feature 3DGS, as shown in Eq. (4).

3.4. Applications

Novel view semantic segmentation. In the novel view semantic segmentation task, we first assign labels to each codeword in S by performing a max-similarity search based on the cosine similarity between each semantic feature and the text feature corresponding to the given labels. Subsequently, leveraging the trained decoupled feature 3D Gaussian, we render the latent feature map $\mathbf{F}_r$ from a novel viewpoint. The resulting low-dimensional feature map is then projected back to the original feature dimension ($D = 512$) via the pre-trained decoder Ψ, which forms part of the auto-decoder employed in the Semantic Compression (Sec. 3.3.2). Finally, we quantize $\mathbf{F}_r$ using S, generating a feature index map M. This quantization ensures that the decoded $\mathbf{F}_r$ aligns with the discrete codebook representation learned during training, improving the model's ability to map pixels to the correct semantic labels. The labels for each pixel in $\mathbf{F}_r$ are then retrieved by querying M, completing the semantic segmentation process.

Language-guided Editing. Unlike novel view semantic segmentation, the essence of language-guided editing is to perform semantic classification on each 3D Gaussian, and then perform editing operations such as object removal and color modification for the 3D Gaussians belonging to specified categories. The process of semantic classification is similar to the approach used in novel view semantic segmentation. When a set of categories is provided, 3D Gaussians are classified using a max-similarity approach, similar to novel view semantic segmentation. If only a single target category is specified, each Gaussian is queried using a similarity threshold γ to determine whether it belongs to the target category.

Although the aforementioned method can accomplish text label assignment for 3D Gaussians in the semantic field, the primary objective of language-guided editing requires modifications to the color field. To bridge this gap, we establish correspondence between the independent color and semantic fields by exploiting their spatial correlations. Specifically, if the Euclidean distance in the position space between a 3D Gaussian in the color field and a 3D Gaussian labeled τ in the semantic field is less than a hyperparameter λ, then the 3D Gaussian in the color field is also assigned the label τ. In this manner, we indirectly transfer semantic labels to the color field, enabling the editing of 3D Gaussians based on their semantic associations.

4. Experiments

4.1. Experiment Settings

Dataset. To evaluate the effectiveness of our method, we conduct experiments on the Replica dataset [37], a widely-used multi-view indoor scene dataset that consists of high-quality indoor scenes with photo-realistic textures and per-

Method	Feature dim	mIoU↑	Training time↓	Gaussian Number	Storage↓	FPS↑
LEGaussians [36][CVPR2024]	8	32.0	28min	571.0k	18.1MB	77.66[1]
LangSplat [31][CVPR2024]	3×3	58.5	10+5×3min	572.2k	20.7MB	61.45/3
Feature 3DGS (w/ speed-up) [47]	128	73.4	99min	607.2k	296.5MB	19.60
Feature 3DGS [47][CVPR2024]	512	73.9	351min	602.2k	1175.2MB	5.66
Ours (just decouple)	512	73.5	263min	29.6k	59.4MB	6.26
Ours (decouple+ADC)	512	77.9	220min	28.7k	57.6MB	9.37
Ours (decouple+ADC+SC)	9	76.6	8min	110.5k	13.0MB	36.08

Table 1. **Quantitative comparison results.** All metric data are obtained in the same environment by calculating the average across all scenes, where "Feature dim" represents the dimension of the semantic features stored in the 3D Gaussians, "ADC" represents Adaptive Data Compression, "SC" represents Semantic Compression. In the results of LangSplat, "×3 & /3" represents 3 different semantic levels.

primitive semantic classes. We experiment on five scenes from the Replica: room0, room1, office0, office2 and office3. For each scene, 900 images with a size of 640×480 are captured along a randomly chosen trajectory, and then COLMAP [34, 35] is used to estimate the camera pose for each image while performing undistortion, the sparse point cloud output by COLMAP will be used for the initialization of 3D Gaussians. To ensure the difference between the new viewpoint images and the training dataset, we select every 10th image starting from the 1st for the test dataset and every 10th image starting from the 6th for the training dataset, obtaining approximately 90 images for each.

Evaluation Metrics. For novel view semantic segmentation, following Feature 3DGS [47], we manually re-label some pixels with semantically close labels such as "rugs" and "floor". We measure the mean intersection over union (mIoU) based on our annotations and use class = 10 for the mIoU metric. The rendering speed (FPS) is measured by rendering feature maps. In addition, the efficiency of the model is evaluated based on the storage requirements of semantic information and training time. Note that when comparing storage, our method incorporates all attribute data of 3D Gaussians in the semantic field into the calculation. In contrast, for other comparison methods that utilize composite fields (color and semantic), only the semantic features of the 3D Gaussians are included in the computation. Additionally, for any method requiring extra data (e.g., decoder) to infer semantic feature maps, these supplementary data will also be accounted for in the storage calculation. When calculating training time, the time required for the compression and quantization of semantic features is also included.

Implementation Details. All experiments were conducted on a system equipped with an Intel Xeon Gold 6326 CPU and an NVIDIA GeForce RTX3090 GPU. Following Feature 3DGS [47], we utilize the pre-trained LSeg [20] to extract the semantic features of each image, and the feature map from the LSeg image encoder has feature size 360×480 with dimension 512. Adam optimizer is used to train our DF-3DGS for 30k iterations, the learning rate of the semantic feature is 0.005. For Adaptive Data Compression, we set $\rho = 8$, $K_{max} = 100$, $\alpha = 0.9$, $\beta = 0.99$, and train it for 28 epochs with a learning rate of 0.0008. Additionally, we use grid sampling to downsample the original feature map, thereby accelerating the optimization of S. For the autoencoder, we set $d = 9$ for the latent feature's dimension and train it for 5k iterations with a learning rate of 0.0001.

4.2. Novel View Semantic Segmentation

We compare our method both qualitatively and quantitatively with Feature 3DGS [47], LangSplat [31], and LEGaussians [36] for novel view semantic segmentation, following their default parameters. Note that, since LangSplat renders three levels of feature maps, we select the level with the highest IoU (for each category in every image) for comparison.

Our method surpasses other comparative methods in both segmentation quality and efficiency, as shown in Tab. 1. When focusing solely on segmentation quality, our method, without using an autoencoder for dimensionality reduction, achieves a 4% (73.9% *vs.* 77.9%) improvement in mIoU compared to the second-best method, Feature 3DGS, and reduces training time by approximately 2.2 hours (351 min *vs.* 220 min). Due to the decoupling of the semantic field from the color field, the number of 3D Gaussians required for the semantic field is less than 5% (602.2k *vs.* 28.7k) of the Feature 3DGS. This results in a 95.1% (1175.2MB *vs.* 57.6MB) reduction in storage compared to Feature 3DGS, and the rendering FPS of the feature map also increases by 3.71 (5.66 *vs.* 9.37).

If model efficiency is considered, Feature 3DGS can use speed-up operations to accelerate training and rendering. However, its training time and storage requirements remain unacceptable, and it still cannot achieve real-time rendering. In contrast, our method, after using an autoencoder for dimensionality reduction, can render the feature map in real-time (FPS>30) and still leads Feature 3DGS by 3.2% (73.4% *vs.* 76.6%) in mIoU. Moreover, the training time

[1]This FPS is obtained by removing *torch.cuda.empty_cache()* from the source code provided by [36]; otherwise, it drops to 29.67.

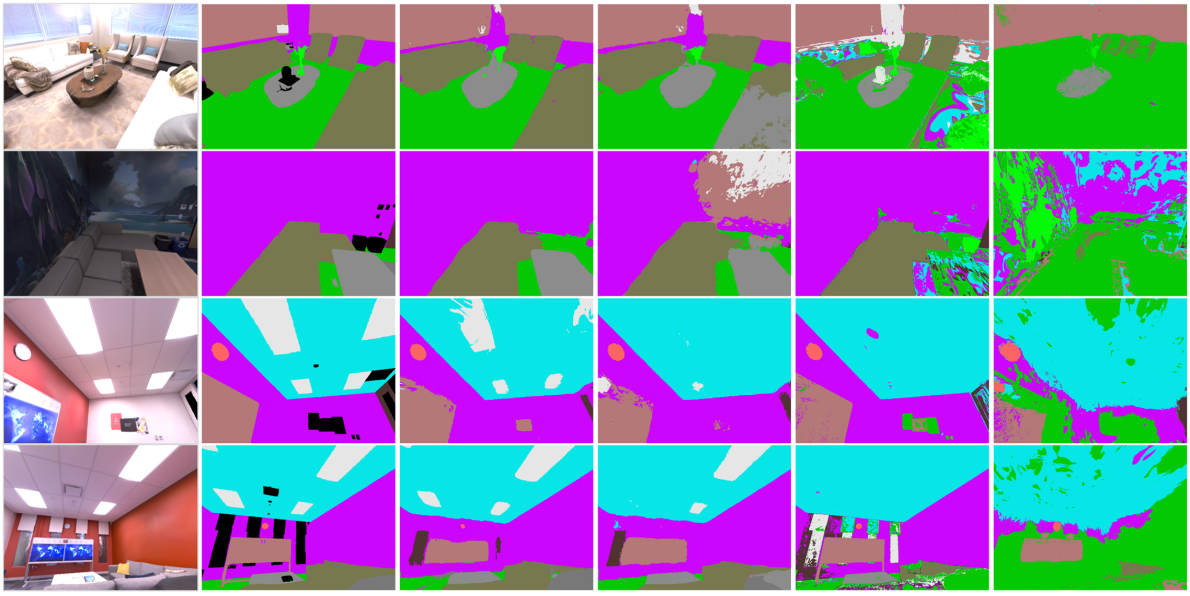

Figure 3. **Novel view semantic segmentation results on scenes from the Replica dataset.** Our method demonstrates more accurate segmentation results.

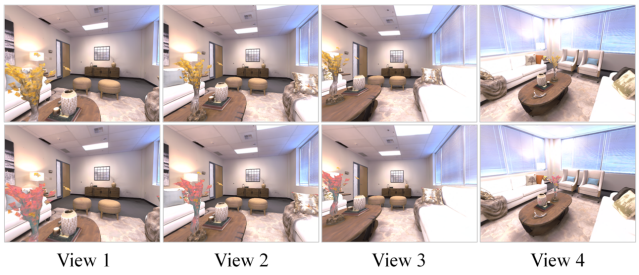

Figure 4. **Color Modification.** The first and second rows represent the synthesized results from different viewpoints before and after modifying the color of the **plant**, respectively.

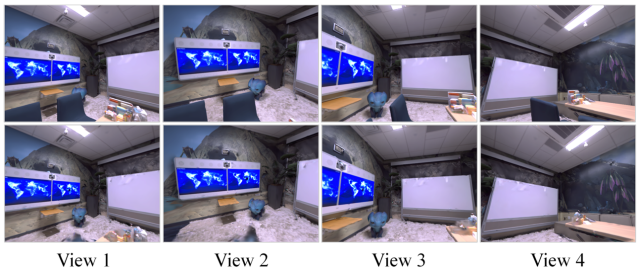

Figure 5. **Object Removal.** The first and second rows represent the synthesized results from different viewpoints before and after the removal of the **chair**, respectively.

is reduced to just 8 minutes (99min *vs.* 8min), and the storage requirement is only 13MB (296.5MB *vs.* 13MB). Fig. 3 displays a qualitative comparison of the novel view semantic segmentation results, demonstrating the efficacy of our method in segmenting challenging objects in indoor scenes.

4.3. Language-guided Editing

In this section, we demonstrate the capability of DF-3DGS combined with traditional visual 3DGS in performing editable novel view synthesis. This process typically involves an editing operation and a target object, such as "delete the chair", which is achieved by deleting or setting the transparency to zero for the 3D Gaussians related to the chair, where the classification of 3D Gaussians is obtained through text queries (Sec. 3.4). As shown in Fig. 5, our method is able to remove the chair while preserving the original state of other objects, demonstrating its 3D scene awareness and its significant value in **handling occlusions** during novel view synthesis. Furthermore, we also showcase the method's ability to modify the color of specific objects, which is very useful for **artistic design** in 3D scenes. As shown in Fig. 4, we modified the color of the plant while keeping the appearance of other objects unchanged.

4.4. Ablation Study

In this section, we will analyze the effectiveness of various components of our method, including the decoupling of the semantic field and the color field and hierarchical compression.

4.4.1. Discussion of decoupled semantic field

Comparing our method (just decouple) with Feature 3DGS, under the same condition of using the original LSeg features (dimension=512), as shown in Tab. 1, our method uses approximately 572.6k fewer (602.2k *vs.* 29.6k) 3D Gaussians

than Feature 3DGS, this clearly demonstrates the sparsity of the semantic field. Although our method stores additional parameters such as p^2, s^2, q^2 and o^2 for each 3D Gaussian, the overall storage is still significantly reduced by approximately 1GB (1175.2MB *vs.* 59.4MB). Additionally, the training time is reduced by about 1.5 hours (351min *vs.* 263min), while the mIoU only decreases by 0.4% (73.9% *vs.* 73.5%). These results clearly demonstrate the necessity of decoupling the semantic field from the color field.

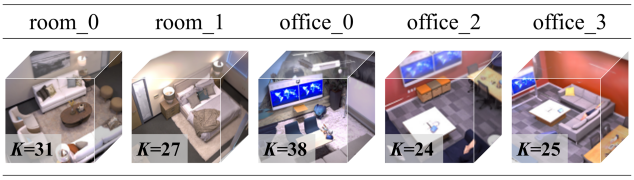

Figure 6. **Adaptive codebook size across different scenes.**

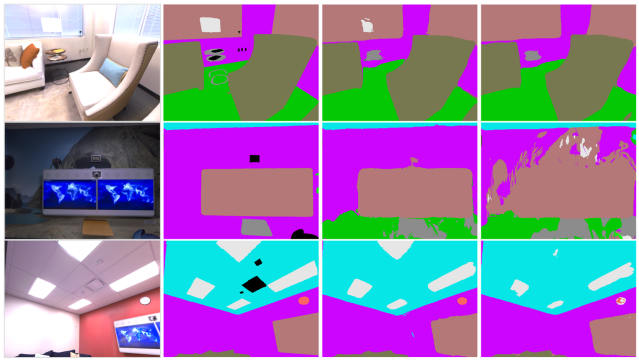

Figure 7. **Qualitative comparison of Adaptive Data Compression (ADC).**

4.4.2. Discussion of hierarchical compression

Adaptive Data Compression. Adaptive Data Compression serves two primary purposes: First, it adaptively extracts the core semantic features from the scene; Second, it quantizes the original feature map, and the quantized features eliminate noise among the same semantic features, thereby making the features more robust and discriminative, this enhances the ability to query image features using text features. As shown in Fig. 6, our method can adaptively extract varying numbers of core semantic features depending on the scene, thus flexibly constructing the codebook S. As shown in Tab. 1, it is evident that Adaptive Data Compression significantly improves the mIoU by 4.4% (73.5% *vs.* 77.9%). Additionally, Fig. 7 shows that it also yields better qualitative segmentation results. These results demonstrate the accuracy of the core semantic features extracted by Adaptive Data Compression and the effectiveness of the quantization.

Semantic Compression. Tab. 1 shows that although using Semantic Compression to reduce the semantic feature dimensions results in a 1.3% decrease (77.9% *vs.* 76.6%) in mIoU, it still outperforms Feature 3DGS (73.9% *vs.* 76.6%). Additionally, the training time is reduced by approximately 3.5 hours (220min *vs.* 8min), the storage is decreased by 44.6 MB (57.6MB *vs.* 13MB), and the FPS is significantly improved by 26.71 (9.37 *vs.* 36.08). These results demonstrate the effectiveness of compressing the semantic feature dimensions. We further analyzed the impact of different semantic feature dimensions on our method. As shown in Tab. 2, as the dimension of semantic features increases, the mIoU generally shows an upward trend, but the training time also increases. We find that the number of 3D Gaussians exhibits a negative correlation with the dimension of the semantic features. This may be because low-dimensional semantic features have difficulty capturing multi-view semantic information, thus requiring more 3D Gaussians for modeling. Despite the increase in the number of Gaussians as the dimension of semantic features decreases, the overall storage requirement still decreases. Additionally, the FPS decreases as the dimension of the semantic features increases. Considering both quality and efficiency, we choose 9 as our compression dimension in the experiments.

Metric	3	6	9	12	15	512
mIoU	73.6	75.7	76.6	76.0	76.3	77.9
Training	6min	7min	8min	9min	10min	220min
Gaussians	131.5k	123.5k	110.5k	90.1k	106.7k	28.7k
Storage	12.0MB	12.8MB	13.0MB	12.3MB	15.3MB	57.6MB
FPS	39.58	36.91	36.08	38.53	33.26	9.37

Table 2. **Ablation study of latent feature's dimensions.**

5. Conclusion

In this paper, we proposed DF-3DGS, a novel method for constructing a 3D semantic field that is decoupled from the color field, significantly reducing the number of 3D Gaussians required for explicit reconstruction of the semantic field. Additionally, we introduce an adaptive data compression technique and a scene-specific autoencoder to optimize the compact representation of semantic features, resulting in significant gains in storage efficiency and reconstruction speed. The experimental results clearly demonstrate that our method outperforms other 3DGS-based approaches, such as Feature 3DGS, in both the quality and the reconstruction efficiency of the semantic field.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (No.62106201, No.62376217), and in part by the Guangdong Basic and Applied Basic Research Foundation (No.2025A1515011501). The authors thank the anonymous reviewers and the Area Chair for their helpful feedback.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021. 1

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19697–19705, 2023. 1

[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 (ICCV)*, pages 9650–9660, 2021. 3

[4] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. Graph.*, 32(3), 2013. 2

[5] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14124–14133, 2021. 1

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision – ECCV 2022*, pages 333–350, Cham, 2022. Springer Nature Switzerland. 2

[7] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21476–21485, 2024. 2

[8] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *Computer Vision – ECCV 2024*, pages 422–438, Cham, 2025. Springer Nature Switzerland. 2

[9] Y. Eldar, M. Lindenbaum, M. Porat, and Y.Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997. 5

[10] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps, 2024. 2

[11] Ben Fei, Jingyi Xu, Rui Zhang, Qingyuan Zhou, Weidong Yang, and Ying He. 3d gaussian splatting as new era: A survey. *IEEE Transactions on Visualization and Computer Graphics*, page 1–20, 2024. 2

[12] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings, 2024. 2

[13] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.*, 37(6), 2018. 2

[14] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 634–644, 2024. 2

[15] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4220–4230, 2024. 2

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 1, 2

[17] 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 (ICCV)*, pages 19729–19739, 2023. 1, 3

[18] 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 Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 3

[19] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems*, pages 23311–23330. Curran Associates, Inc., 2022. 1, 3

[20] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation, 2022. 3, 6

[21] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8508–8520, 2024. 2

[22] Guibiao Liao, Jiankun Li, Zhenyu Bao, Xiaoqing Ye, Jingdong Wang, Qing Li, and Kanglin Liu. Clip-gs: Clip-informed gaussian splatting for real-time and view-consistent 3d semantic understanding. *arXiv preprint arXiv:2404.14249*, 2024. 2, 3

[23] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20654–20664, 2024. 2

[24] 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. *Commun. ACM*, 65(1):99–106, 2021. 1, 2

[25] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), 2022. 2

[26] KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Compgs: Smaller and faster gaussian splatting with vector quantization, 2024. 2

[27] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10349–10358, 2024. 2

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 3

[29] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[30] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5020–5030, 2024. 2

[31] 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 (CVPR)*, pages 20051–20060, 2024. 2, 3, 6

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 3

[34] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[35] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision – ECCV 2016*, pages 501–518, Cham, 2016. Springer International Publishing. 6

[36] 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 (CVPR)*, pages 5333–5343, 2024. 2, 3, 6

[37] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The replica dataset: A digital replica of indoor spaces, 2019. 5

[38] 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, 2022. 1, 3

[39] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 4

[40] Linhan Wang, Kai Cheng, Shuo Lei, Shengkun Wang, Wei Yin, Chenyang Lei, Xiaoxiao Long, and Chang-Tien Lu. Dc-gaussian: Improving 3d gaussian splatting for reflective dash cam videos, 2024. 2

[41] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2021. 1

[42] 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 (CVPR)*, pages 20310–20320, 2024. 2

[43] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth, 2024.

[44] 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 (CVPR)*, pages 20331–20341, 2024.

[45] 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 (CVPR)*, pages 6796–6807, 2024. 2

[46] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2021. 1

[47] 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 (CVPR), pages 21676–21685, 2024. 2, 3, 6

[48] Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In *Computer Vision – ECCV 2024*, pages 324–342, Cham, 2025. Springer Nature Switzerland. 2

[49] Lingting Zhu, Zhao Wang, Jiahao Cui, Zhenchao Jin, Guying Lin, and Lequan Yu. Endogs: Deformable endoscopic tissues reconstruction with gaussian splatting, 2024. 2

[50] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *International Journal of Computer Vision*, 2024. 3