Structurally Disentangled Feature Fields Distillation for 3D Understanding and Editing

Yoel Levy
The Hebrew University of Jerusalem
yoel.levy@mail.huji.ac.il

David Shavin
The Hebrew University of Jerusalem
david.shavin@mail.huji.ac.il

Itai Lang
University of Chicago
itai.lang83@gmail.com

Sagie Benaim
The Hebrew University of Jerusalem
sagiebenaim@gmail.com

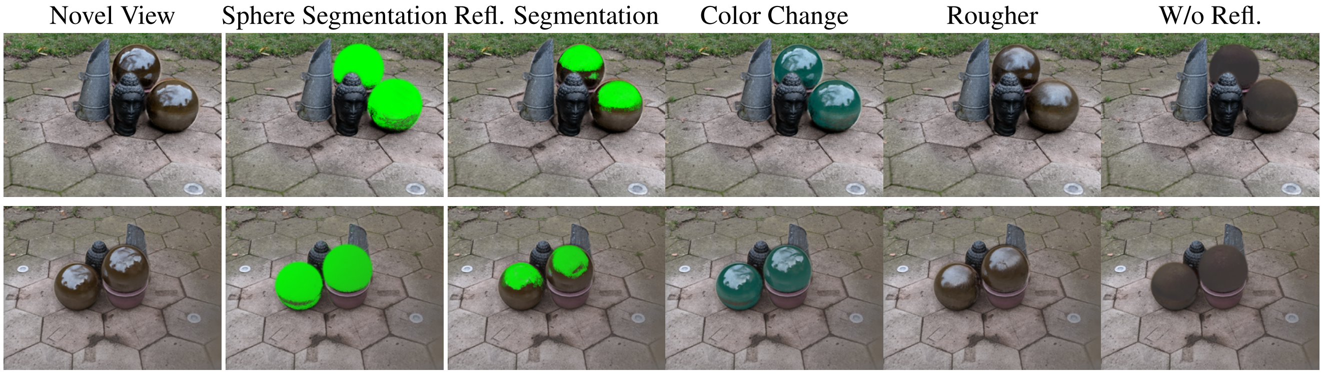

Figure 1. An illustration of our method for the Garden-spheres scene from the real-world RefNeRF dataset [29]. By distilling 2D DINOv2 [22] features into our disentangled feature field representation, our method enables: (i). High-fidelity novel view synthesis, (ii). 3D segmentation of the spheres, (iii). 3D segmentation of the reflective region of the spheres, (iv). Color editing while adhering to reflections, (v). Changing the roughness of the spheres, (vi). Removing the reflection from the spheres. Refl. = Reflection.

Abstract

Recent work demonstrated the ability to leverage or distill pre-trained 2D features obtained using large pre-trained 2D foundation models into 3D features, enabling impressive 3D editing and understanding capabilities with only 2D supervision. While powerful, such features contain significant view-dependent components, especially in scenes with complex materials and reflections. When distilled into a single 3D feature field, these inconsistencies are averaged, degrading feature quality and harming downstream tasks like segmentation. We hypothesize that explicitly modeling the physical causes of view-dependence is key to "cleaning" these features during distillation. To this end, we propose to decompose the 3D feature field into view-independent and view-dependent components, guided by a physically-based reflection model. Our core contribution is demonstrating that this structural disentanglement improves the quality and view-invariance of the distilled semantic features. This leads to improved 3D segmentation, particularly in challenging reflective regions, and enables higher-fidelity physically-grounded editing applications. Our project page is available at https://structurallydisentangled.github.io/.

1. Introduction

2D feature distillation has become widely used in computer graphics and computer vision, whereby 2D image features obtained using large-scale self-supervised methods are "distilled" to the parameters of an underlying 3D model. Doing so enables 3D semantic understanding and editing, given 2D RGB and feature supervision only. Several works [10, 12, 37] considered the setting of novel view synthesis with an underlying NeRF [18] or a Gaussian Splatting [9] model, achieving impressive 3D understanding and editing capabilities. Each 3D point (in the former) or Gaussian (in the latter) stores a feature value, which can be rendered to different views using volumetric rendering. The rendered views correspond to 2D feature maps obtained using self-supervised 2D methods such as DINOv2 [22]. This paradigm was later extended to text [10, 23], enabling 3D text-based understanding and editing Although impressive, current models assume that 3D features are captured using a single view-independent feature field (or a single 3D feature value per point or Gaussian). A critical issue is that the source 2D features are not view-consistent (Sec. 4.1 and Fig. 3). This is especially true in reflective areas. Current models average such view-dependent variations, resulting in inferior features for downstream tasks such as 3D segmentation.

In this work, we argue that instead of ignoring view-dependence, we should model it explicitly to better isolate the stable, view-independent semantic features. Our key insight is that by using a physically-inspired decomposition of the scene's appearance, we can provide a powerful structural prior to also disentangle the feature field. To this end, we propose to capture 3D features using *multiple disentangled feature fields* that capture different structural components of 3D features. While these structural components could involve a variety of physical properties such as lighting and deformations, we focus on the disentanglement of view-dependent and view-independent features.

Specifically, the view-independent field encodes stable object semantics, geometry, identity, and diffuse/material attributes, while the view-dependent field isolates appearance residuals caused by specular reflections and other camera-dependent effects. Our disentangled feature fields can be learned using 2D supervision only, in an unsupervised manner, thus enabling the disentanglement of 2D (and 3D) features into components that are view-independent and view-dependent. Subsequently, each component can be controlled separately, enabling semantic and structural understanding and editing capabilities. For instance, 3D segmentation can be achieved by considering only view-independent features, and discarding the view-dependent ones, resulting in significant improvement compared to baselines, which average the view-dependent component of features. Using a user click, one can segment an entire object in 3D or only its reflective component, edit view-independent object properties, such as its color, while adhering to reflections, or remove the object's reflections. An illustration is provided in Fig. 1.

To achieve our desired disentanglement, we propose computing the feature value of a 3D point along a viewing direction as the combination of the outputs of two disentangled feature fields: (i). A reflected view feature field capturing view-dependent features that arise from specular object reflections, and (ii). A view-independent feature field capturing diffuse features of the object which depend only on the location of a 3D point and not the viewing direction.

We evaluate our approach on a variety of objects from a diverse set of scenes from the Shiny Blender dataset [29] as well as real-world scenes from the RefNeRF real-world dataset [29] and Mip-NeRF-360 dataset [1]. In terms of 3D understanding, we consider the tasks of: (i). 3D segmentation, for which our structurally disentangled representation achieves superior results compared to a single holistic feature field [12], and (ii). Segmentation of view-dependent components in a scene. For example, one can segment only the reflective component of an object selected using a user click. In terms of 3D editing, we demonstrate the applicability of our approach on (i). The ability to remove the reflective component of an object, and lastly (ii). The ability to edit individual 3D component. For example, changing the object's color while correctly adhering to reflections or manipulating its roughness.

In summary, we offer the following contributions: (1). We are the first to show that a physical decomposition of appearance can be leveraged to disentangle and improve distilled semantic feature fields. (2). We validate our method on 3D semantic segmentation, demonstrating that our cleaner features lead to improved performance. (3). We further show that our disentangled representation allows for more robust and physically-consistent results in downstream editing tasks, such as reflection-aware color changes and roughness manipulation.

2. Related Work

2.1. Structured Novel View Synthesis

Neural Radiance Field (NeRF) [18] reconstructs a 3D scene from 2D images by mapping 3D spatial locations and viewing directions to color and density values. Different works introduced physical modeling within a NeRF-like formulation to model geometry, lighting, materials, and reflections. For instance, to model geometry, works integrate a signed distance function (SDF) formulation [15, 17, 21, 25, 32, 35, 39]. Other works extended NeRF to enable relighting by disentangling appearance into scene lighting and materials [2–4, 27, 40, 41]. PaletteNeRF [13] decomposes appearance into a linear combination of palette-based bases. In the context of reflections modeling, approaches such as [24] represented appearance using disentangled view-dependent specular and reflective appearance. Ref-NeRF [29] modeled appearance through two disentangled radiance fields, one for view-independent diffuse properties and another for reflective properties. Ref-NeuS [7] addresses reflective surface reconstruction through a reflection-aware photometric loss. BakedSDF [36] and ENVIDR [16] extended this for glossy surfaces with material decomposition. UniSDF [31] improved upon these by including two separate radiance fields for reflections. IntrisicNeRF [38] factors a NeRF scene into multi-view consistent components, including reflectance and shading. Unlike the above, our focus is on structural disentanglement for feature fields, modeling 3D features together with appearance. As far as we can ascertain, our work is the first to enable such disentanglement.

2.2. 2D Feature Distillation

Due to the scarcity of 3D data, recent work attempted to distill or lift 2D features trained using pre-trained self-supervised 2D models (such as DINO [5], DINOv2 [22] or CLIP-LSeg [14]). In particular, [12, 28] extended NeRF to model not only appearance, but also semantics, through an additional view-independent feature field that maps a 3D point into a 3D feature. Other works embedded semantic information into NeRFs [26, 34], while further works focused on integrating open-vocabulary text-based features, by embedding CLIP features into NeRF [10] or Gaussian splatting [23]. Feature3DGS [42] extends feature field distillation to 3D Gaussian Splatting, achieving faster training and rendering than NeRF-based methods by using an explicit point-based representation with low-dimensional features per Gaussian and a convolutional decoder for upsampling. However, these models assume that 3D features are captured using a single view-independent feature field, which averages view-dependent variations present in 2D foundation model outputs. Our key insight is that these view-dependent inconsistencies—particularly in reflective regions—should be explicitly modeled rather than averaged. By decomposing features into physically-motivated view-independent and view-dependent components, we obtain cleaner semantic features that improve downstream tasks such as 3D segmentation.

3. Method

Fig. 2 illustrates an overview of our method. Our method builds upon prior NeRF representation [31] and view-dependent appearance model [29]. For completeness, we describe the NeRF backbone [31] and appearance decomposition model in Secs. 3.1 and 3.2. Within these sections, we present our novel feature field disentanglement. Then, we explain how elements in the scene are segmented and edited using our proposed feature decomposition (Sec. 3.3), and discuss the intuition behind our decomposition (Sec. 3.4).

3.1. Structurally Disentangled Feature Fields

Preliminaries. NeRF [18] represents a 3D scene as a radiance field parametrized by an MLP that predicts color $\mathbf{c}(\mathbf{x}) \in \mathbb{R}^3$ and volume density $\sigma \in \mathbb{R}$ given a 3D point $\mathbf{x} \in \mathbb{R}^3$. Volume rendering then generates views from viewing direction $\mathbf{d} \in \mathbb{R}^3$. Similarly, feature fields [10, 12, 37] use an additional MLP to map 3D coordinates to feature vectors $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^n$. Previous work learns a single feature field, omitting view-dependent information.

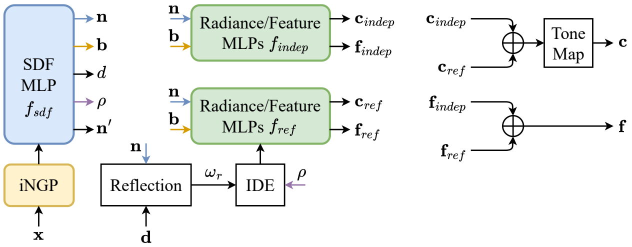

Figure 2. **The method's pipeline.** We decompose the appearance color of the scene $\mathbf{c}$ into physical components $\mathbf{c}_{indep}$ and $\mathbf{c}_{ref}$ and sum them to compute the color of the scene at location $\mathbf{x}$ and viewing direction $\mathbf{d}$. We also learn a decomposed feature field for the scene, $\mathbf{f}_{indep}$ and $\mathbf{f}_{ref}$, which enables physically-oriented semantic understanding and editing applications. Please see Sec. 3 for more details.

Signed distance representation. Similar to previous work [31], our model's backbone is an MLP computing the signed distance function $d(\mathbf{x})$ (SDF) at each location $\mathbf{x}$. The scene's surface is defined at the zero level set of the function. $\mathbf{x}$ undergoes a contraction transformation [1, 31], limiting its values to $[0, 2)$.

The density $\sigma(\mathbf{x})$ for volume rendering is computed by $\sigma(\mathbf{x}) = \alpha\Psi_\beta(d(\mathbf{x}))$, where Ψ_β is the cumulative distribution of a Laplace distribution with zero mean and a scale parameter β that is learned during optimization. The benefit of the SDF representation is that the normal of the scene surface can be easily derived as the gradient of the distance function:

$$\mathbf{n}(\mathbf{x}) = \nabla d(\mathbf{x}) / ||\nabla d(\mathbf{x})||_2. \tag{1}$$

The normal is used as input to the components in our model for computing the disentangled feature representation.

To better reconstruct high-frequency details and to speed up training, we use the iNGP hash encoding [20]. Each location $\mathbf{x}$ is mapped to a high-dimensional feature vector, which is the concatenation of the iNGP's pyramid-level features. This feature vector is the input to our SDF MLP. To ease notations, we omit in this section the notation of $\mathbf{x}$ for location-dependent quantities.

Radiance and features components. We decompose the appearance of the scene into two elements: a view-independent color $\mathbf{c}_{indep}$ and a view-dependent reflected color $\mathbf{c}_{ref}$. The view-independent color is calculated as:

$$\mathbf{c}_{indep} = f_{indep}(\mathbf{n}, \mathbf{b}), \tag{2}$$

where f_{indep} is a learned MLP, $\mathbf{n}$ is the normal defined in Eq. 1, and $\mathbf{b}$ is a bottleneck vector output from the SDF MLP. This vector enables additional degrees of freedom to accommodate second-order effects, such as varying illumination over the scene [29, 31]. $\mathbf{c}_{indep}$ represents the view-independent color and accordingly is independent of the viewing direction $\mathbf{d}$.

The other color component, $\mathbf{c}_{ref}$, is view-dependent. It is responsible for reflective scene regions and models the reflected radiance in the scene. Following previous work [29], we calculate the reflection for the viewing direction about the normal:

$$\omega_r = 2(\omega_o \cdot \mathbf{n})\mathbf{n} - \omega_o, \tag{3}$$

where $\omega_o = -\hat{\mathbf{d}}$, is a unit vector pointing to the camera from a point in space, and $\hat{\mathbf{d}}$ is a viewing direction. Then, we use Integrated Directional Encoding (IDE) [29] in the computation of $\mathbf{c}_{ref}$:

$$\mathbf{c}_{ref} = f_{ref}(\mathbf{n}, \mathbf{b}, \text{IDE}(\omega_r, \kappa)), \tag{4}$$

where $\kappa = 1/\rho$, and ρ is the surface roughness predicted by the SDF MLP. Finally, the rendered color is given by:

$$\mathbf{c} = \text{tonemap}(\mathbf{c}_{indep} + \mathbf{c}_{ref}), \tag{5}$$

where $\text{tonemap}(\cdot)$ is a tone mapping function converting linear color to the sRGB format and clipping the output to the range $[0, 1]$.

In this work, we are the first to propose to leverage the physically-based appearance model to decompose the *feature field* of the scene $\mathbf{f}$ as follows:

$$\mathbf{f} = \mathbf{f}_{indep} + \mathbf{f}_{ref}, \tag{6}$$

where $\mathbf{f}_{indep}$ and $\mathbf{f}_{ref}$ are computed in the same manner as $\mathbf{c}_{indep}$, Eq. (2), and $\mathbf{c}_{ref}$, Eqs. (3) and (4). The feature components are used as a *semantic* representation for their color counterparts. We note that the semantic feature from the independent feature field only, $\mathbf{f}_{indep}$, can be used for understanding tasks such as 3D segmentation, as 3D segmentation is inherently view-independent. This is an example where a physical disentanglement to view-dependent and independent components can result in "improving" or "cleaning" undesirable feature components required for 3D segmentation. By jointly modeling features and colors, one can also enable a diversity of editing applications as illustrated in Sec. 4.

3.2. Training Objective

The training of our structural decomposition of the scene is supervised by posed images and their corresponding DINOv2 [22] feature maps. Given a set of posed images $\{C_1^{gt}, \ldots, C_m^{gt}\}$, where $C_i^{gt} \in \mathbb{R}^{H \times W \times 3}$, we obtain associated 2D feature maps $\{F_1^{gt}, \ldots, F_m^{gt}\}$ using the pretrained feature extractor DINOv2, where $F_i^{gt} \in \mathbb{R}^{H \times W \times n}$, and n is the feature dimension. The rendered color images are compared to the given images with an l_2 loss:

$$\mathcal{L}_c = \frac{1}{m} \sum_i ||C_i - C_i^{gt}||_2^2, \tag{7}$$

where C_i is the scene appearance rendered from our model for the view direction of image C_i^{gt}. Additionally, we regularize the SDF MLP learning with the eikonal loss [8, 31] to promote the approximation of a valid SDF:

$$\mathcal{L}_e = \frac{1}{|\mathbf{x}|} \sum_{\mathbf{x}} (||\nabla \mathbf{d}(\mathbf{x})||_2 - 1)^2. \tag{8}$$

We also encourage normal smoothness by comparing the computed normal $\mathbf{n}(\mathbf{x})$ from the SDF to the normal $\mathbf{n}'(\mathbf{x})$ predicted by the SDF MLP:

$$\mathcal{L}_p = \sum_{\mathbf{x}} w_{\mathbf{x}} \, ||\mathbf{n}(\mathbf{x}) - \mathbf{n}'(\mathbf{x})||^2. \tag{9}$$

Further, we penalize back-facing normals using the orientation loss [29]:

$$\mathcal{L}_o = \sum_{\mathbf{x}} w_{\mathbf{x}} \max(0, \mathbf{n}(\mathbf{x}) \cdot \mathbf{d}(\mathbf{x}))^2. \tag{10}$$

For feature learning, we have found that optimizing the radiance and feature fields concurrently compromises the learning process of both. Thus, we first learn the decomposed appearance of the scene using the total loss:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_e \mathcal{L}_e + \lambda_p \mathcal{L}_p + \lambda_o \mathcal{L}_o. \tag{11}$$

Then, we freeze the appearance model and train MLPs for optimizing our proposed feature field decomposition using:

$$\mathcal{L}_f = \frac{1}{m} \sum_i ||F_i - F_i^{gt}||_2^2. \tag{12}$$

3.3. Segmentation and Editing

Our structurally decomposed feature field enables physically oriented segmentation and editing of the scene. We segment the scene as follows. First, we select a rendered feature component $F_{comp}(x, y)$, where (x, y) is the pixel location, and F_{comp} can be F_{indep} or F_{ref}. Then, we correlate the selected feature with the corresponding decomposed feature field in 3D. The location of features with a correlation factor above a threshold belongs to the segmented region of interest in the scene. Once we obtain the region of interest, we can control and modify the physical properties *locally*, such as the view-independent color, roughness, level of reflection, and more, as demonstrated in Sec. 4.

3.4. How is Disentanglement Achieved?

The MLPs $\mathbf{f}_{indep}$ and $\mathbf{f}_{ref}$ predict the view-independent and view-dependent 3D feature components (see Fig. 2). Disentanglement is achieved via three constraints: (i). By design, $\mathbf{f}_{indep}$ cannot model view-dependent effects, as it does not receive the view direction as input, unlike $\mathbf{f}_{ref}$, which models view-dependent components. (ii). The total feature

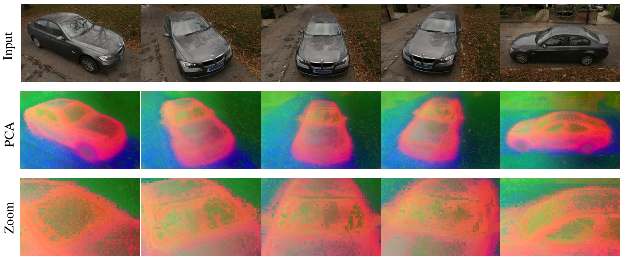

Figure 3. PCA of DINOv2 features for ground-truth input views of the Sedan scene from real-world dataset of [29]. We zoom in on the windshield, illustrating differences in corresponding locations between views.

value is modeled by summing the outputs of $\mathbf{f}_{indep}$ and $\mathbf{f}_{ref}$ (Eq. (6)), with volumetric rendering enforcing consistency with ground-truth 2D features. (iii). Although not explicitly enforced, $\mathbf{f}_{indep}$ is biased to capture view-independent information: for a given 3D position it produces a single feature shared across all views, while $\mathbf{f}_{ref}$ represents higher-cost view-dependent residuals only when required, encouraging a compact decomposition.

4. Experiments

We evaluate our representation on multiple vectors. First, we demonstrate that input features from DINOv2 contain both view-dependent and view-independent information. Second, we consider the ability to segment objects in 3D space. Unlike previous methods, our disentangled feature fields capture both diffuse and reflective properties and allow for a better reconstruction of the semantic components in the scene. We also enable the segmentation of the reflective component of objects in isolation from different novel views. Third, we consider the ability to remove view-dependent (reflective) components in the scene for specific semantic objects segmented using our approach. Fourth, we consider the ability to edit the scene, where one can manipulate or change only specific objects and their dependent (reflective) properties in isolation.

Datasets. We evaluate our method on synthetic scenes from the Shiny Blender [29] dataset and real-world scenes from the RefNeRF real-world [29] dataset and Mip-NeRF-360 dataset [1]. We consider a diverse set of scenes and objects from both real world and synthetic scenes, featuring multiple objects and variable lighting conditions, highlighting our generality. In particular, we consider 11 scenes and 25 objects from 3 real-world and synthetic datasets. This is comparable to recent work such as DFF or RefNeRF. We consider the standard train-view/novel-view splits provided by the respective datasets and evaluate our model on such novel views. Additionally, our method supports incorporating arbitrary 2D semantic features extracted from models like CLIP-Lseg and DINOv2. The webpage provides associated videos depicting novel views and corresponding segmentation, removal, and editing results. We also provide implementation details in the supplementary.

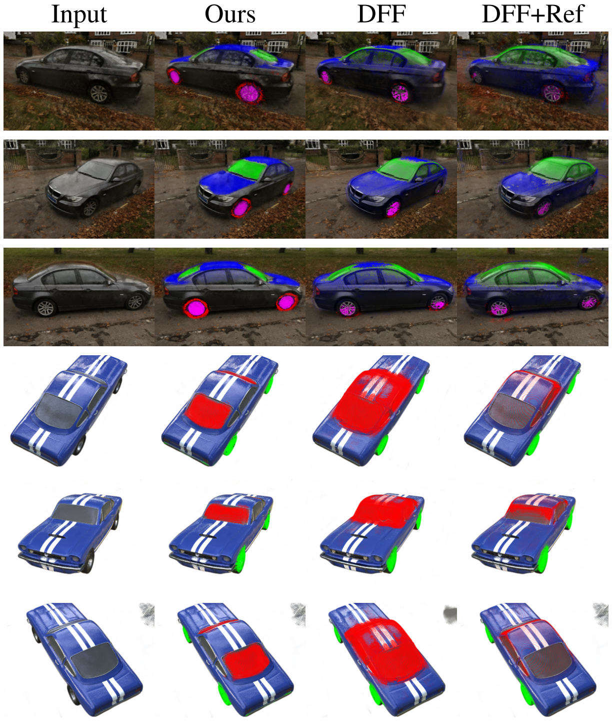

Figure 4. 3D objects segmentation from three novel views, for the Sedan scene from real-world RefNeRF [29] dataset for the objects of Bonnet-top, Windshield, Hubcaps and Wheels, and for the Car scene from synthetic Shiny Blender [29] dataset for the objects of Windshield and Wheels. We compare our result to DFF [12] and to a baseline where DFF is optimized for features while RefNeRF is optimized for appearance (see Sec. 4.2).

4.1. Feature Analysis

We first consider whether the distilled DINOv2 features capture both view-dependent and view-independent components. To this end, we visualize the PCA of the features for five ground-truth views of the Sedan scene from the real-world RefNeRF dataset. As seen in Fig. 3, while the features appear similar, there are notable differences, particularly in reflective regions such as the windshield (zoomed in). As further evidence, we note the recent work of [6], which demonstrates that features obtained from large foundation models, DINOv2 in particular, are not 3D view consistent. As such, applying our model has the advantage of disentangling view-dependent and view-independent components of a given feature view and can enhance downstream applications that require view-independent feature representations. This is illustrated in Sec. 4.2, where we show that our view-independent feature field yields better 3D segmentation results than using the full (both dependent and independent) feature field or baselines. Beyond 3D segmentation, one can render ground truth views only using

Scene (Objects)	Ours	Ours implicit	Ours total	Ours optt	DFF	DFF+ Ref	Feature3DGS
Bicycle (Bench, Wheels)	**0.583**	0.381	0.407	0.444	0.518	0.504	0.278
Counter (Mitten, Plant, Tray)	**0.824**	0.705	0.641	0.664	0.713	0.629	0.383
Garden (Ball, Plant, Tabletop)	**0.840**	0.813	0.786	0.389	0.824	0.700	0.506
Kitchen (Lego)	**0.871**	0.761	0.631	0.778	0.856	0.850	0.582
Gardenspheres (Cone, Head, Spheres)	**0.825**	0.663	0.647	0.619	0.776	0.770	0.748
Sedan (Bonnet-top, Windshield, Hubcaps, Wheel)	**0.643**	0.337	0.377	0.485	0.625	0.512	0.371
Toycar (Body, Wheel)	**0.709**	0.499	0.176	0.283	0.689	0.654	0.383
Teapot (Cover)	**0.762**	0.125	0.654	0.768	0.529	0.399	0.669
Mean (Real World Scenes)	**0.757**	0.628	0.594	0.523	0.691	0.582	0.490
Car (Windshield, Wheels)	**0.799**	0.437	0.402	0.088	0.523	0.445	0.199
Toaster (Body, Toasts)	0.906	0.844	0.833	0.839	**0.940**	0.787	0.647
Helmet (Body, Windshield)	**0.863**	0.795	0.705	0.894	0.731	0.472	0.589
Mean (Shiny Blender)	**0.856**	0.568	0.550	0.648	0.731	0.527	0.478

Table 1. Mean IoU for segmentation of objects from the Shiny Blender synthetic dataset [29] (bottom) and real-world scenes (top). First four scenes are taken from the real world RefNeRF [29] dataset while the other four real world scenes are from the Mip-NeRF-360 dataset. On the RHS, we compare our approach to DFF [12], a baseline where DFF is optimized for features while RefNeRF is optimized for appearance (see Sec. 4.2), and Feature 3DGS [42]. On the LHS, we also consider variants of our approach: (1). Ours-implicit: Our approach, but with implicit, instead of explicit, representation of physical properties [s.a. roughness], (2). Ours-total: Our approach, but using the total features [dependent and independent] for segmentation, (3). Our-optt: Our approach, where we optimized the color and features together. For each scene, we show, in brackets, the segmented objects.

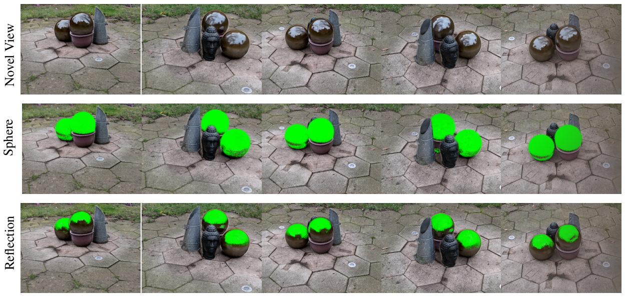

Figure 5. Segmentation of the spheres for novel views of the Garden-spheres real-world scene using either the full segmentation of the sphere (second row) or only the reflective component of the spheres (third row).

the view-independent component, discarding their view-dependent component.

4.2. Improved Feature Distillation

We consider our method's capability in providing improved 3D features. To this end, we demonstrate that our distilled features can be used to improve downstream semantic segmentation as well segment view-dependent components. We focus on distilling general-purpose DINOv2 features to obtain a versatile 3D representation, rather than task-specific segmentation features (e.g., from LSeg), as our goal is to improve the quality of the core 3D feature field for multiple downstream applications.

4.2.1 Full Object Segmentation.

In Fig.4, we visualize our segmentation of three novel views for a real-world scene and a synthetic scene. The segmentation is obtained by clicking on each object and using our disentangled view-independent feature component, described in Sec. 3.3. We compare our method to Distilled Feature Field (DFF) [12], the closest method to ours, which uses a single view-independent feature field. As RefNeRF is an improved appearance model, we also consider another baseline whereby appearance is obtained as in RefNeRF, using both view-dependent and view-independent feature fields, while semantics is obtained as in DFF, using a view-independent feature field. We also compare to Feature3DGS [42], which extends feature distillation to 3D Gaussian Splatting. As can be seen, our method results in a superior segmentation and can successfully segment both reflective and non-reflective regions well, while the baseline is worse, particularly in reflective regions. Additional results are provided in the webpage.

For numerical evaluation, we compare the segmentation of objects for both synthetic and real-world scenes. We manually obtain ground-truth segmentations by first applying the Segment Anything Model [11] and subsequently manually refining masks (see examples in the webpage). As seen in Tab. 1, our method results in better mean IoU scores.

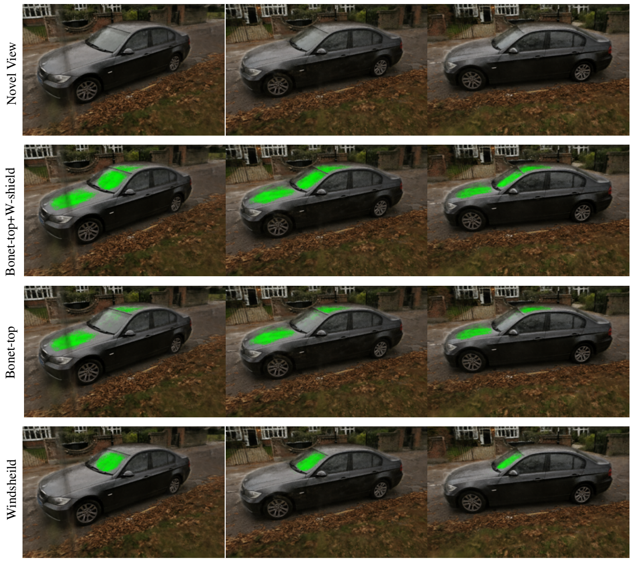

Figure 6. Segmentation of the reflective component of different semantic components of the real-world car scene. The first row displays three novel views. We then demonstrate the segmentation of the reflective component of (1). Both the bonnet-top and the windshield (second row), (2). The bonnet-top (third row), and (3). The windshield (fourth row).

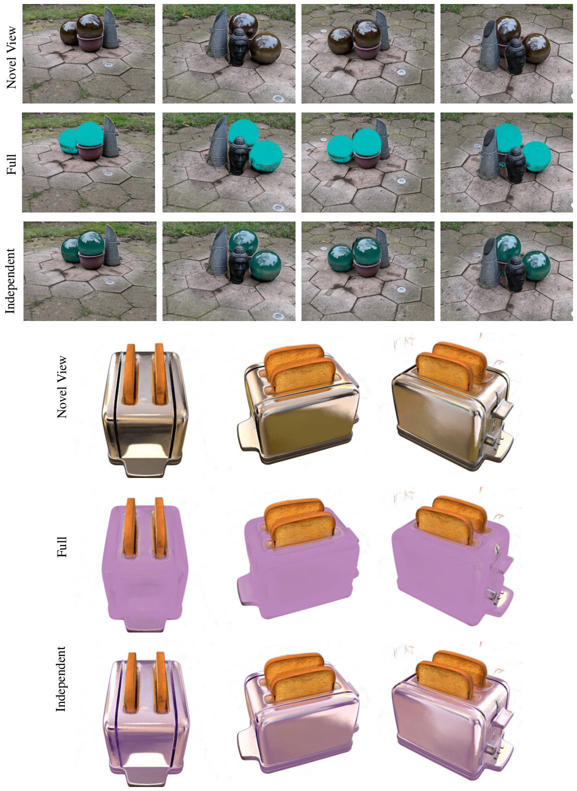

Figure 7. Editing the color of i). The spheres in the real-world gardenspheresand (ii). The toaster in the synthetic toaster, either (1). using all radiance fields, (*full*) or (2). using the independent component only (*indepdent*).

As can be seen, our method significantly outperforms DFF, Feature3DGS, and a direct DFF+RefNeRF baseline, confirming that our structural disentanglement is critical for success.

4.2.2 Explicit modeling

Our approach leverages explicit components to model the view-dependent and view-independent feature fields. Specifically, we model roughness and reflections explicitly, as detailed in Eq. 3 and Eq. 4 in Sec. 3. This enables novel applications that incorporate physical properties. However, it is unclear whether explicit molding is preferable to implicit modeling of view-dependent and view-independent feature fields when one is interested in segmentation only. To evaluate the impact of explicit modeling, we considered a baseline that implicitly models view-dependent and independent feature fields by excluding the Reflection, IDE and roughness components (Eq. 3 and Eq. 4). As seen in Tab. 1, this results in inferior performance. Our explicit model also provides superior appearance results (SSIM/PSNR/LPIPS), also in comparison to DFF, as shown in the supplementary.

4.2.3 View Dependent Segmentation.

We consider the ability to segment the view-dependent reflective surfaces of objects. We begin by segmenting semantic objects using the view-independent features according to Sec. 3.3 and subsequently select only a subset of points corresponding to features from the view-dependent (reflectance) feature field. Fig. 5 illustrates our success in segmentation for the Garden-spheres scene, for both the entire sphere as well as only the view-dependent reflection. Fig. 6 illustrates the ability to segment the reflective region of the (1). bonnet-top and the windshield, (2). bonnet-top, and (3). windshield. As can be seen, only the reflective region of the desired semantic entity is depicted. Additional examples are provided in the webpage.

4.3. Downstream Editing Applications

Our decomposed feature representation not only improves segmentation but also leads to higher-fidelity and more robust outcomes in downstream editing tasks, which are challenging for methods that do not separate view-dependent components.

4.3.1 Removal

The ability to edit roughness is inherited from the physical reflection model. However, our contribution is the ability to obtain a clean segmentation of a specific object using our improved f_{indep} features, and then apply this physical edit only to that object, a task that is difficult for baselines, which struggle to segment these objects correctly.

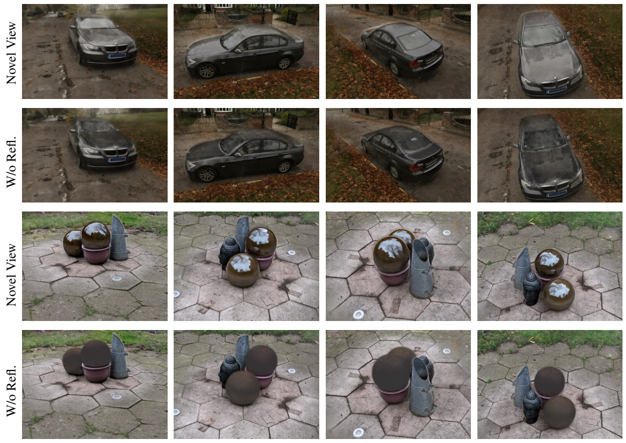

Figure 8. Removing the reflective component of (i). Bonnet-top and windshield in sedan scene (ii). spheres in the gardenspheres.

While prior work has explored 3D object removal [19, 30, 33], our structurally disentangled representation enables a novel capability of removing only the reflective component of objects. This can be achieved by selecting the 3D points corresponding to a given object (using a click) and then rendering for those points only the color component from the view-independent radiance field. Fig. 8 shows the removal of the reflective component of the bonnet-top and windshield for the car scene and the Garden-spheres scene. Our method enables the removal of the reflected radiance from the object and the retention of its diffuse color.

4.3.2 Editing

Color Editing. We consider the ability to edit an object's color while adhering to its reflections. In Fig. 7, we change the color of the segmented 3D points of (i). the spheres, for the real-world Garden-spheres scene, and (ii). the toaster body for the synthetic toaster scene. This color change occurs either (1). using both radiance fields (independent and reflection) or (2). using the independent component only, leaving the view-dependent reflective component unchanged. Using the latter results in a more natural manipulation that correctly adheres to reflections. Additional examples are shown on the webpage. To assess our color editing abilities numerically, we conducted a user study on colored objects. We consider 5 colors and 5 scenes (1 object each) and asked users to assess: 1. Color faithfulness ("how well does the desired color match the object?"), 2. Realism ("how realistic does the object look?"), 3. Reflections match unedited scene ("how well do the reflections match the unedited scene?"). We considered 25 users and obtained a MOS score (1-worse, 5-best) of **4.6**/**3.1**/**4.6** vs. 2.0/2.8/**4.6** in comparison to DFF for questions 1/2/3, respectively.

Roughness Editing. Next, we consider the ability to manipulate physical components. In Eq. 4, we utilize the roughness parameter κ that controls the roughness. To this end, we consider the ability to manipulate the roughness of individual objects in the scene. We do so by segmenting the 3D points of an object and varying the roughness parameter κ for those points. Fig. 9 illustrates two examples: (i). the spheres in the real-world scene of the Garden-spheres scene, and (ii). the helmet case for the synthetic helmet scene. Additional examples are provided in the webpage.

Ablation Study. In the webpage we consider an ablation in which our segmentation is performed using not only the independent feature component $\mathbf{f}_{indep}$, but also the independent and dependent components together, $\mathbf{f}_{indep} + \mathbf{f}_{ref}$. As can be seen, this results in a worse segmentation.

We also consider two additional ablations: (1) We optimized the appearance and feature model together, as opposed to first training the appearance model only and then the feature model (see Sec. 3.2). For 3D segmentation (as in Tab 1), on average, it is worse, e.g., we get mIOU 0.648 (vs our 0.856) for Shiny Blender. (2). We also conducted an ablation where we removed the tone mapping function (Eq. 5). We observe that appearance is slightly worse. Specifically, for the Gardenphere scene we obtain PSNR/LPIPS/SSIM of 28.8/0.180/0.809 vs our 28.9/0.180/0812. This results in a similar minor performance drop for 3D segmentation.

Limitations. While our method shows a significant improvement over baselines , we acknowledge that lifting 2D features to pristine, perfectly complete 3D masks remains a challenging open problem. We also note that while our method is designed for segmenting and editing reflective regions of semantic objects in a scene, it cannot do so at the instance-based level. For example, for the spheres in Fig. 1, selecting one of the spheres will result in capturing and editing both spheres. As in standard multiview reconstruction, errors can occur when the number of multiview ground truth features is insufficient. Also, our task is highly unsupervised, as we aim to disentangle semantics and appearance in 3D space, given only 2D entangled appearance and semantics supervision. Improved physical modeling of, e.g., reflections and enhanced and generalizable feed-forward models may result in improved performance.

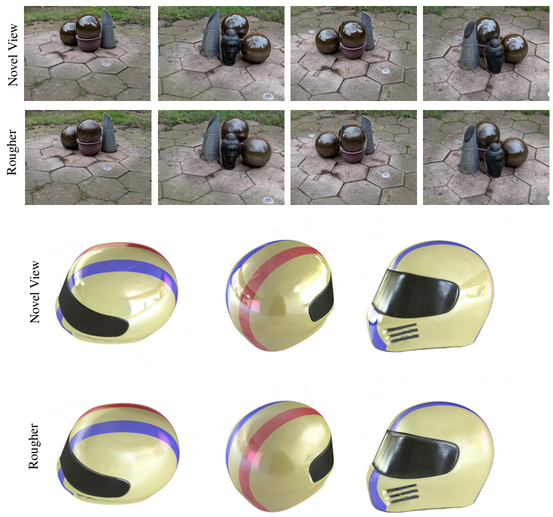

Figure 9. Controlling the roughness of (i). The spheres in Garden-spheres, and (ii). The helmet (only the case) for the helmet scene.

5. Conclusion

We presented a method for 3D scene understanding and editing by distilling pretrained 2D features into a structurally disentangled feature field representation. Our approach captures view-dependent and view-independent features, enabling superior 3D segmentation over prior work. Our improved features allow for robust segmentation and high-fidelity manipulation of reflective components, and editing colors and roughness while preserving reflections. In future work, we aim to improve 3D consistency of foundation models by rerendering views using only the view-independent component and finetuning on such features. We also aim to extend our method to additional physical properties, such as lighting.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 2, 3, 5

[2] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020. 2

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021.

[4] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 34: 10691–10704, 2021. 2

[5] 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. 3

[6] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21795–21806, 2024. 5

[7] Wenhang Ge, Tao Hu, Haoyu Zhao, Shu Liu, and Ying-Cong Chen. Ref-neus: Ambiguity-reduced neural implicit surface learning for multi-view reconstruction with reflection. *arXiv preprint arXiv:2303.10840*, 2023. 2

[8] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit Geometric Regularization for Learning Shapes. In *Proceedings of Machine Learning and Systems 2020*, pages 3569–3579. MLSys 2020 Organizing Committee, 2020. 4

[9] 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

[10] 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, 2, 3

[11] 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. 6

[12] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing NeRF for Editing via Feature Field Distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022. 1, 2, 3, 5, 6

[13] Zhengfei Kuang, Fujun Luan, Sai Bi, Zhixin Shu, Gordon Wetzstein, and Kalyan Sunkavalli. Palettenerf: Palette-based appearance editing of neural radiance fields. *arXiv preprint arXiv:2212.10699*, 2022. 2

[14] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 3

[15] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 2

[16] Ruofan Liang, Huiting Chen, Chunlin Li, Fan Chen, Selvakumar Panneer, and Nandita Vijaykumar. Envidr: Implicit differentiable renderer with neural environment lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 79–89, 2023. 2

[17] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *European Conference on Computer Vision*, pages 210–227. Springer, 2022. 2

[18] 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, 2, 3

[19] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 7

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 3

[21] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021. 2

[22] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3, 4

[23] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3D Language Gaussian Splatting. *arXiv preprint arXiv:2312.16084*, 2023. 2, 3

[24] Ravi Ramamoorthi et al. Precomputation-based rendering.

Foundations and Trends® in Computer Graphics and Vision, 3(4):281–369, 2009. 2

[25] Radu Alexandru Rosu and Sven Behnke. Permutosdf: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8466–8475, 2023. 2

[26] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023. 3

[27] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2

[28] 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

[29] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5491–5500, 2022. 1, 2, 3, 4, 5, 6

[30] Dongqing Wang, Tong Zhang, Alaa Abboud, and Sabine Süsstrunk. Inpaintnerf360: Text-guided 3d inpainting on unbounded neural radiance fields. *arXiv preprint arXiv:2305.15094*, 2023. 7

[31] Fangjinhua Wang, Marie-Julie Rakotosaona, Michael Niemeyer, Richard Szeliski, Marc Pollefeys, and Federico Tombari. UniSDF: Unifying Neural Representations for High-Fidelity 3D Reconstruction of Complex Scenes with Reflections. *arXiv preprint arXiv:2312.13285*, 2023. 2, 3, 4

[32] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2

[33] Silvan Weder, Guillermo Garcia-Hernando, Aron Monszpart, Marc Pollefeys, Gabriel J Brostow, Michael Firman, and Sara Vicente. Removing objects from neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16528–16538, 2023. 7

[34] Jingfeng Yao, Xinggang Wang, Lang Ye, and Wenyu Liu. Matte anything: Interactive natural image matting with segment anything model. *Image and Vision Computing*, page 105067, 2024. 3

[35] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2

[36] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P Srinivasan, Richard Szeliski, Jonathan T Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023. 2

[37] Jianglong Ye, Naiyan Wang, and Xiaolong Wang. FeatureNeRF: Learning Generalizable NeRFs by Distilling Foundation Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8962–8973, 2023. 1, 3

[38] Weicai Ye, Shuo Chen, Chong Bao, Hujun Bao, Marc Pollefeys, Zhaopeng Cui, and Guofeng Zhang. Intrinsicnerf: Learning intrinsic neural radiance fields for editable novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 339–351, 2023. 2

[39] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022. 2

[40] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 2

[41] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)*, 40(6):1–18, 2021. 2

[42] 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, 6