VersatileGaussian: Real-time Neural Rendering for Versatile Tasks using Gaussian Splatting

Renjie Li[1⋆], Zhiwen Fan[2,†,⋆], Bohua Wang[3],
Peihao Wang[2], Zhangyang Wang[2], and Xi Wu[4]

[†] Project Lead
[1]THU, [2]UT Austin, [3]Baidu, [4]CUIT
https://VersatileGaussian.github.io

Abstract. The acquisition of multi-task (MT) labels in 3D scenes is crucial for a wide range of real-world applications. Traditional methods generally employ an analysis-by-synthesis approach, generating 2D label maps on novel synthesized views, or utilize Neural Radiance Field (NeRF), which concurrently represents label maps. Yet, these approaches often struggle to balance inference efficiency with MT label quality. Specifically, they face limitations such as (a) constrained rendering speeds due to NeRF pipelines, and (b) the implicit representation of MT fields that can result in continuity artifacts during rendering. Recently, 3D Gaussian Splatting has shown promise in achieving real-time rendering speeds without compromising rendering quality. In our research, we address the challenge of enabling 3D Gaussian Splatting to represent Versatile MT labels. Simply attaching MT attributes to explicit Gaussians compromises rendering quality due to the lack of cross-task information flow during optimization. We introduce architectural and rasterizer design to effectively overcome this issue. Our **VersatileGaussian** model innovatively associates Gaussians with shared MT features and incorporates a feature map rasterizer. The key element of this versatile rasterization is the Task Correlation Attention module, which utilizes cross-task correlations through a soft weighting mechanism that disseminates task-specific knowledge. Across experiments on the ScanNet and Replica datasets shows that VersatileGaussian not only sets a new benchmark in MT accuracy but also maintains real-time rendering speeds (35 FPS). Importantly, this model design facilitates mutual benefits across tasks, leading to improved quality in novel view synthesis even in situations that the ground-truth dense labels are absent, and with the assistant of dense labels from off-the-shelf 2D predictors.

1 Introduction

Efficient and accurate 3D scene modeling and analysis have become pivotal in the realms of computer vision and graphics. Numerous applications necessitate the concurrent rendering of view-consistent multi-task (MT) labels from any

⋆ Equal Contribution.

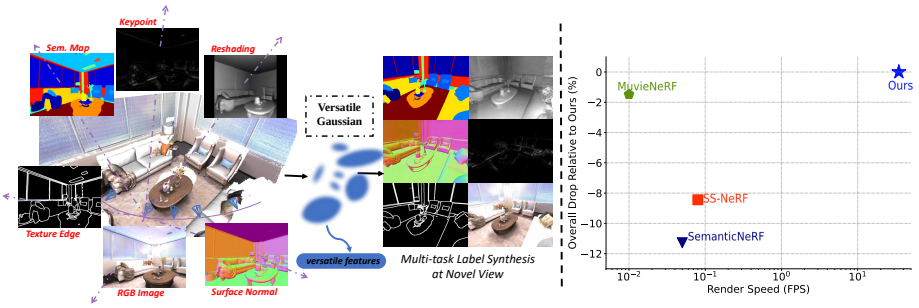

Fig. 1: Our approach **VersatileGaussian** unifies Multi-task rendering by augmenting 3D Gaussian Splatting with integrated feature learning and a meticulously designed multi-task rasterizer. This approach ensures real-time rendering (35 FPS) from any new views within 3D environments (left side). Moreover, it delivers notable enhancements in multi-task performance (right side) across various benchmarks.

viewpoint within a 3D scene, demanding high-resolution rendering in real time. Traditional methods adhere to an analysis-by-synthesis approach, synthesizing

MT labels for novel views using a renderer to generate color image, followed by the application of 2D multi-task discriminative models [44, 50]. Despite their simplicity, these frameworks exhibit view inconsistency when viewpoints change, leading to a degraded user experience. Implicit scene representations, particularly Neural Radiance Fields (NeRFs) [29], have shown superior rendering quality by mapping spatial coordinates to color and density and employing volume rendering [19] techniques at new viewpoints. Nevertheless, MT rendering with implicit representations or their extensions [53, 56] faces significant challenges: NeRF-based methods are inherently slow in training and rendering; moreover, incorporating an MT field can exacerbate this slowness and often result in floating artifacts in the MT field.

An emerging alternative, 3D Gaussian Splatting (3D-GS) [20], offers a point-based representation that achieves faster training and real-time rendering speeds compared to NeRF-based methods, maintaining or surpassing the quality of rendered images. This advancement facilitates real-time rendering applications in VR and AR. However, the 3D-GS framework, primarily designed for image synthesis, does not inherently support the joint learning of dense label maps alongside color images. A naive extension that attaches MT labels to each Gaussian proves to be sensitive to noise and yields only moderate rendering quality (see the experiment section).

In this work, we introduce **VersatileGaussian**, the first approach that enables the real-time rendering of high-quality, versatile label maps from any viewpoint. We propose a method to learn a shared MT feature field for each Gaussian, representing the common visual attributes of the scene. A novel MT rasterizer splats and rasterizes these common features into task-specific features adaptable to any task. The essence of this rasterizer is the facilitation of cross-task information flow for joint optimization, with the lightweight *Task Correlation Attention* enabling rapid yet high-quality feature field decoding, significantly en-

hancing MT accuracy. Experimental validation on benchmark datasets confirms that VersatileGaussian surpasses all prior baselines in both rendering quality and efficiency. Notably, it is observed that even versatile labels inferred from off-the-shelf 2D models can boost the novel view synthesis quality of 3D-GS.

Our key contributions are summarized as follows:

- VersatileGaussian, the first framework that enable real-time, high-quality MT rendering at novel views, drawing inspiration from 3D Gaussian splatting.
- A novel architecture for learning shared feature vectors on each Gaussian and an MT rasterizer that seamlessly decodes into task-specific features, subsequently translated into versatile label maps.
- The Task Correlation Module within the MT rasterizer effectively fosters cross-task correlations, substantially enhancing MT rendering quality.

2 Related Work

2.1 Efficient 3D Representation for NVS.

Novel View Synthesis aims to generate photo-realistic images at novel views using observations at several source views. A category to achieve NVS is to utilize mesh as representation [1, 3, 10] and employ rasterizers [26] to render images and optimize the mesh. The development of Multi-View Stereo [16] (MVS) enables MVS-based NVS methods such as [5, 11, 17, 23], which re-project and blend the images at source views into the target views. Recently, Neural Radiance Fields (NeRF) [29] have achieved great success in NVS by representing scenes as radiance fields parameterized by neural networks, producing photo-realistic images of high quality. Despite the rendering quality benefits of this implicit representation, the rendering speed is reduced due to the high computational cost of querying the radiance fields. Therefore, much effort has been put into accelerating NeRFs by using voxels [6, 12, 18, 32, 39], point clouds [49], decomposition [6, 15, 32], octrees [40, 51], and hash tables [30]. Compared to the neural rendering employed by NeRFs, which queries radiance field values at sampled points, rasterization using splatting is more GPU-friendly and potentially faster. ADOP [33] represents scenes with neural points and leverages differential rasterization to render target images. Recent progress made by Gaussian Splatting (3D-GS) [20] achieves impressive results both in render quality and speed. By representing a scene as a set of attributed Gaussians, its customized rasterizer can achieve very fast, high-quality image rendering using the splatting technique. However, its rasterizer is customized to the explicit representations of RGB SH coefficients attached to each Gaussian, complicating its direct extension to render versatile labels.

2.2 Multi Task Learning for Dense Prediction

Dense prediction tasks, such as semantic segmentation, key point detection, edge detection, and surface normal estimation, play a critical role in computer vision and pattern recognition. Traditionally, researchers have designed separate

neural networks for each task. However, recent studies [2, 14, 25, 45, 48] have shown relations among these tasks. Multi-task learning has become a popular approach to jointly train neural networks for multiple dense prediction tasks. **Encoder-Based Methods** [7,8,35,41] focus on integrating different tasks during the feature extraction stage. By sharing the task features in the encoding stage, these methods ensure that multiple tasks share a common representation, enabling them to propagate consistent information among themselves. This improves model accuracy, reduces the number of parameters, and reduces computational effort [14,25]. **Decoder-Based Methods** [45,48,54,55,58] may also share features for different tasks in the encoding stage but further introduce cross-task feature fusion modules in the decoder. These modules allow task-independent features in the decoding phase to interact and fuse, further improving model accuracy. **Attention** is a widely-used technique to enable multi-task feature interaction and refinement [2, 25, 48, 50]. For example, Ye and Xu proposed InvPT [50], leveraging Transformers to capture global associations in images. This approach achieves simultaneous modeling of global spatial locations and multiple tasks in a unified framework.

2.3 Versatile Labels in 3D representation

Recently, in addition to RGB images, more dense labels have been introduced into 3D representations for scene understanding [13, 42], enabling applications such as semantic segmentation [31,57], panoptic segmentation [24,37], and scene editing [22,46]. Nevertheless, given a set of source views with versatile labels and camera poses, synthesizing the labels of a target view, namely Multi-Task View Synthesis (MTVS), which aims to achieve better render quality for all kinds of labels, is a novel problem recently explored by SS-NeRF [53]. SS-NeRF utilizes NeRF as a 3D representation for multi-task labels. Furthermore, MuvieNeRF [56] proposes a cross-task attention module to facilitate knowledge sharing and information flow among the tasks. However, these two methods suffer from low render speed due to the high computational cost of neural rendering.

3 Preliminaries

3.1 3D Gaussian Splatting

3D Gaussian Splatting (3D-GS) [20] is an explicit 3D scene representation that models the scene using a set of 3D Gaussians. A 3D Gaussian is parameterized by a mean vector $\boldsymbol{x} \in \mathbb{R}^3$ and a covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$:

$$G(\boldsymbol{p}) = \frac{1}{(2\pi)^{3/2}\,|\Sigma|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{p}-\boldsymbol{x})^T \Sigma^{-1}(\boldsymbol{p}-\boldsymbol{x})} \tag{1}$$

3D-GS renders the color $\boldsymbol{c}$ by blending n ordered Gaussians overlapping the pixels using the following render function:

$$\boldsymbol{c} = \sum_{i=1}^{n} \boldsymbol{c}_i \alpha_i \prod_{j=1}^{i-1} (1-\alpha_j) \tag{2}$$

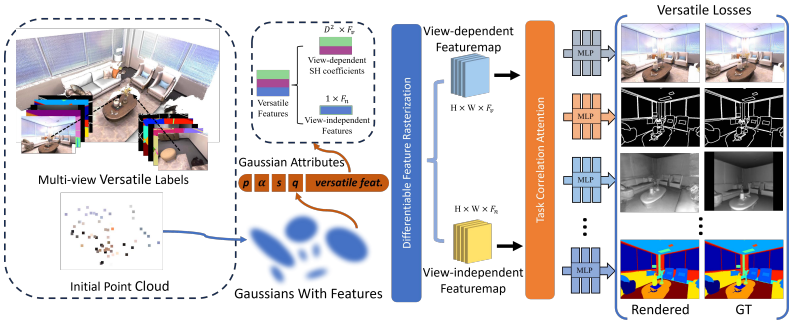

Fig. 2: The pipeline of VersatileGaussian. VersatileGaussian represents versatile labels as view-direction-dependent and view-direction-independent features on the Gaussians. After fast rasterization of the feature maps, a Task Correlation Attention module is used to facilitate task information flow, contributing to better render quality.

where $\boldsymbol{c}_i$ is the color computed from the SH coefficients of the i^{th} Gaussian. α_i is given by evaluating a 2D Gaussian with covariance $\Sigma' \in \mathbb{R}^{2\times2}$ multiplied by a learned per-Gaussian opacity. The 2D covariance matrix Σ' is calculated by projecting the 3D covariance Σ to the camera coordinates:

$$\Sigma' = \boldsymbol{J}\boldsymbol{W}\Sigma\boldsymbol{W}^T\boldsymbol{J}^T \tag{3}$$

where $\boldsymbol{J}$ is the Jacobian of the affine approximation of the projective transformation and $\boldsymbol{W}$ is the view transformation matrix.

In summary, 3D-GS uses a set of 3D Gaussians to represent and render a scene. Each 3D Gaussian is characterized by the following parameters: position $\boldsymbol{x} \in \mathbb{R}^3$, a series of SH coefficients $\left\{\boldsymbol{c}_i \in \mathbb{R}^3 | i = 1, 2, ..., n\right\}$, opacity $\alpha \in \mathbb{R}$, rotation $\boldsymbol{q} \in \mathbb{H}$ and scaling $\boldsymbol{s} \in \mathbb{R}^3$

3.2 Multi-task View Synthesis

Multi-task learning for dense predictions involves learning to generate multiple task labels jointly. The Multi-task View Synthesis (MTVS) problem is slightly different from conventional multi-task dense prediction settings. The goal of MTVS is to jointly synthesize multiple scene properties at novel views using the multi-task labels from given source views [56]. Formally, the goal of MTVS is to learn a model Φ that takes as input a set of V source-view task annotations along with camera poses as references and synthesizes multi-task annotations for a novel view.

$$\mathbf{Y}_T = \Phi\left(\{(\mathbf{Y}_i, \mathbf{P}_i)\}_{i=1}^{V}, \mathbf{P}_T\right), \tag{4}$$

where $\boldsymbol{Y}_i = \left[\boldsymbol{x}_i, \boldsymbol{y}_i^1, \cdots, \boldsymbol{y}_i^K\right]$ denotes RGB images $\boldsymbol{x}_i$ and n other multi-task annotations $\{\boldsymbol{y}_i^j\}_{j=1}^n$ in the i^{th} source view with camera pose $\boldsymbol{P}_i$, and $\boldsymbol{P}_T$ is the camera pose of the target view.

To enable 3D-GS to render multiple labels, either multi-task labels or features have to be explicitly stored on Gaussians.

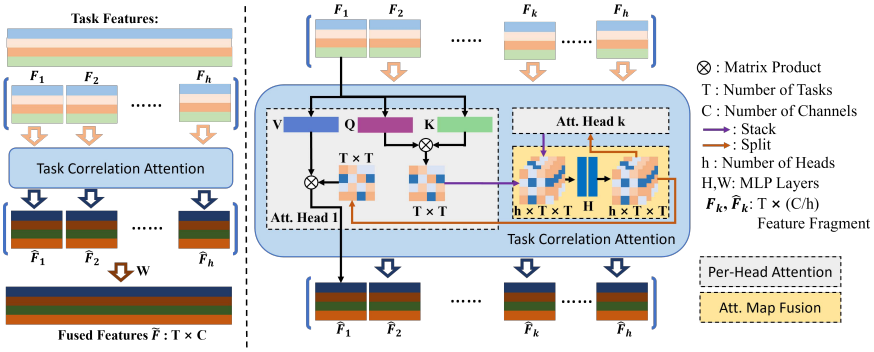

Fig. 3: The Structure of Task Correlation Attention. The per-pixel Task Correlation Attention establishes a soft weighting mechanism to propagate task-wise knowledge. The features are split into several chunks (Left) and fused within and across these feature chunks (Right).

4 Methodology

We introduce **VersatileGaussian**, a real-time approach to render versatile label maps at arbitrary novel view. As illustrated in Fig. 2, an initial point cloud along with a set of source views with multi-task labels and corresponding poses, is provided to optimize the Gaussians. This allows for the synthesis of versatile label maps at any given view. To enable 3D-GS with versatile label representation and rendering, we represent the 3D scene using a feature fields parameterized by Gaussians attached with features. We further extend the differentiable rasterizer to render feature maps at the target view. Then to ensure that versatile labels are robust to noise or incomplete training views, a cross-task attention module is proposed to capture cross-task correlations and facilitate efficient task information fusion. Finally, individual task heads are employed to read out versatile label maps.

4.1 3D Gaussian Feature Fields

3D-GS is an explicit representation where the RGB SH coefficients are stored on each Gaussian to form an appearance field. To model versatile labels and better utilize intrinsic correlations among different labels, feature maps for each task should be rendered at any view. Therefore, we extend 3D-GS to a feature-based version to form a feature field, from which a feature map can be rendered using a differentiable rasterizer. **Representing Tasks as Attributes:** 3D-GS utilize a set of Gaussians to represent a scene, where each Gaussian is parameterized by attributes: position $\boldsymbol{x} \in \mathbb{R}^3$, a serial of SH coefficients $\{\boldsymbol{c}_i \in \mathbb{R}^3 | i = 1, 2, ..., n\}$, opacity $\alpha \in \mathbb{R}$, rotation $\boldsymbol{q} \in \mathbb{H}$ and scaling $\boldsymbol{s} \in \mathbb{R}^3$. To render versatile labels, instead of extending 3D-GS with explicit labels for each task, we represent multiple labels as features on each Gaussian. Specifically, according to the nature

of tasks, the features field is divided into two types, the view-dependent feature field and the view-independent feature field. The view-dependent feature field models view-dependent labels, which vary according to view directions. To make the feature view-dependent, we adopt Spherical Harmonics to approximate the features at different view directions at each Gaussian. Feature field is represented by a serial of SH coefficients $\left\{\boldsymbol{c}_i \in \mathbb{R}^{d_v} | i = 1, 2, ..., n\right\}$ and the view-dependent feature $\boldsymbol{z}^v(\boldsymbol{d})$ of direction $\boldsymbol{d}$ is the sum of all SH basis functions multiplied by the coefficients, formulated as below:

$$\boldsymbol{z}^v(\boldsymbol{d}) = \sum_{i=1}^{n} \boldsymbol{c}_i \mathcal{B}_i(\boldsymbol{d}) \tag{5}$$

where $\mathcal{B}_i : S^3 \rightarrow \mathbb{R}$ is the i^{th} SH basis defined on 3D sphere. The view direction $\boldsymbol{d}$ is implied in the camera pose, thus we omit it in the rest of this section. To model view-independent features, we simply employ a vector $\boldsymbol{z}^n \in \mathbb{R}^{d_n}$ on each Gaussian. **Rasterizer for Versatile Features:** We adopt the fast differentiable rasterizer of 3D-GS [20] to feature maps at a given camera pose. Specifically, the rasterizer takes as input a set of Gaussians where each Gaussian is attributed as: (1) position $\boldsymbol{x} \in \mathbb{R}^3$ (2) opacity $\alpha \in \mathbb{R}$ (3) rotation quaternion $\boldsymbol{q} \in \mathbb{H}$ (4) scaling vector $\boldsymbol{s} \in \mathbb{R}^3$. (5) view-dependent SH coefficients $\left\{\boldsymbol{c}_i \in \mathbb{R}^{d_v} | i = 1, 2, ..., D^2\right\}$ where D is the degree of SH (6) view-independent features $\boldsymbol{z}^n \in \mathbb{R}^{d_n}$. Then the rasterizer renders two feature maps, namely view-dependent feature map $\boldsymbol{Z}^v \in \mathbb{R}^{H \times W \times d_v}$ and view-independent feature map $\boldsymbol{Z}^n \in \mathbb{R}^{H \times W \times d_n}$. Specifically, the view-dependent SH coefficients are firstly transformed to generate the view-dependent features at view direction $\boldsymbol{d}$ correlated to the given camera pose using Eq. (5). Then we use the standard rendering procedure mentioned in Eq. (2) to get the view-dependent and view-independent feature $\boldsymbol{Z}_p^v, \boldsymbol{Z}_p^n$ for each pixel:

$$\left[\boldsymbol{Z}_p^v, \boldsymbol{Z}_p^n\right] = \sum_{k=1}^{n} \left[\boldsymbol{z}_k^v; \boldsymbol{z}_k^n\right] \alpha_k \prod_{j=1}^{k-1} (1 - \alpha_j) \tag{6}$$

4.2 Cross Task Feature Propagation

To enable efficient cross-task information propagation, we design a pixel-wise Task Correlation Attention (TCA) module, establishing a soft weighting mechanism on feature maps for all tasks. **Categorising the Tasks:** We formulate the involved task set as $\mathcal{T} = \{t_1, ..., t_n\}$ where t_k represents the k^{th} task. The tasks are partitioned into two subsets, tailored to their intrinsic nature: (1) The view-dependent tasks $\mathcal{T}_v \subset \mathcal{T}$ such as RGB image, surface norm and shading map, where the labels corresponding to the same 3D point varies when the view direction changes. (2) The view-independent tasks $\mathcal{T}_n \subset \mathcal{T}$ such as semantic map, keypoints and object edge, where the labels corresponding to the same 3D point keep the same despite the change of the view direction. The involved tasks are either view-dependent or view-independent, i.e. $\mathcal{T}_v \cup \mathcal{T}_n = \mathcal{T}$ and $\mathcal{T}_v \cap \mathcal{T}_n = \emptyset$. **Task Correlation Attention:** Considering rendering efficiency, the proposed

Task Correlation Attention is conducted pixel-wisely. Given the view-dependent and view-independent feature $\boldsymbol{Z}_p^v \in \mathbb{R}^{d_v}$ and $\boldsymbol{Z}_p^n \in \mathbb{R}^{d_n}$ of pixel p, we first apply task-specific projections to get the individual features $\boldsymbol{f}_p^t \in \mathbb{R}^d$ of task t at pixel p as follows:

$$\boldsymbol{f}_p^t = \begin{cases} \boldsymbol{W}^t \boldsymbol{Z}_p^v, & t \in \mathcal{T}_v \\ \boldsymbol{W}^t \boldsymbol{Z}_p^n, & t \in \mathcal{T}_n \end{cases}$$

where $\boldsymbol{W}^t \in \mathbb{R}^{d \times d_v}$ if $t \in \mathcal{T}_v$, otherwise $\boldsymbol{W}^t \in \mathbb{R}^{d \times d_n}$. Then the task-correlation attention is applied to each pixel, and thus we can omit the pixel subscript p in $\boldsymbol{f}_p^t$ for shorthand in the rest of this section. Given the individual features for each task $\left\{\boldsymbol{f}^{t_1}, \cdots, \boldsymbol{f}^{t_n}\right\}$, as shown in Fig. 3, we first split the features into fragments along the channel dimension, and then compute the attention score among tasks using multiple separate heads. A linear projection is added to the head dimension to enable each attention function to depend on all of the keys and queries [36], achieving communication across the attention heads. Formally, at each pixel, given individual features $\boldsymbol{f}^{t_k}$ for each task t_k, the fused features $\tilde{\boldsymbol{f}}^{t_k}$ is calculated as:

$$\boldsymbol{A}_h = \sum_{h' \in [H]} \boldsymbol{H}_{h,h'}^{(n)} \boldsymbol{F} \boldsymbol{Q}_{h'} \boldsymbol{K}_{h'}^T \boldsymbol{F}^T, \forall h = 1, \cdots, H. \tag{7}$$

$$\widehat{\boldsymbol{F}}_h = \sum_{h' \in [H]} \boldsymbol{H}_{h,h'}^{(p)} \operatorname{softmax}(\boldsymbol{A}_{h'} / \sqrt{C/H}) \boldsymbol{F} \boldsymbol{V}_h, \forall h = 1, \cdots, H, \tag{8}$$

$$\widetilde{\boldsymbol{F}} = \left[\widehat{\boldsymbol{F}}_1 \cdots \widehat{\boldsymbol{F}}_H\right] \boldsymbol{W}, \tag{9}$$

where $\boldsymbol{F} = \{\boldsymbol{f}^{t_1}, \cdots, \boldsymbol{f}^{t_n}\}_i$ is the feature fragment to be processed by the i^{th} head, $\tilde{\boldsymbol{F}} = \{\tilde{\boldsymbol{f}}^{t_1}, \cdots, \tilde{\boldsymbol{f}}^{t_n}\}$ the fused features. The matrices $\boldsymbol{H}^{(n)}$ and $\boldsymbol{H}^{(p)} \in \mathbb{R}^{H \times H}$ are adopted to correlate the attention weights across the heads. $\boldsymbol{W}$ is a projection matrix to integrate the ray-wise feature calculated by different heads, C and H represent the number of channels and attention heads, respectively. After collecting the fused feature $\tilde{\boldsymbol{f}}^{t_k} \in \mathbb{R}^d$ at each pixel, we use a simple projection matrix $\boldsymbol{W}^{t_k} \in \mathbb{R}^{d_k \times d}$ to readout the explicit labels for each task, specifically, the label $\boldsymbol{y}^{t_k}$ of the k^{th} task is calculated as $\boldsymbol{y}^{t_k} = \boldsymbol{W}^{t_k} \tilde{\boldsymbol{f}}^{t_k}$ for a certain pixel, where d_k is the label dimension of the k^{th} task.

4.3 Optimization

We train the network jointly with all the tasks. The total loss is defined as the weighted average of each task loss as follows:a

$$\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} w^t \mathcal{L}^t \left(\boldsymbol{y}_{\boldsymbol{r}}^t, \hat{\boldsymbol{y}}_{\boldsymbol{r}}^t\right) \tag{10}$$

where $\mathcal{T} = \{t_1, ..., t_n\}$ is the task set, w^t the loss weight of task $t \in \mathcal{T}$, $\mathcal{L}^t(\cdot, \cdot)$ the single task loss for task t, $\boldsymbol{y}^t$ and $\hat{\boldsymbol{y}}^t$the predicted and ground-truth label map of task t.

Table 1: Quantitative Comparison in Replica Datasets. We compare VersatileGaussian with both 2D and 3D baselines, using SemanticNeRF as the reference for overall accuracy. VersatileGaussian significantly outperforms other methods in quality and speed.

Methods	Category.	FPS	Overall	RGB	KP	SH	ED	SN	SL
			$\Delta_m(\%)\uparrow$	PSNR↑	$\mathcal{L}_1\downarrow$	$\mathcal{L}_1\downarrow$	$\mathcal{L}_1\downarrow$	$\mathcal{L}_1\downarrow$	mIoU↑
invPT [50]	2D	5.93	-	-	0.004	0.04	0.03	0.05	0.91
SemanticNeRF [57]	3D	0.05	0.00	26.19	0.495	0.15	0.14	0.13	0.77
SS-NeRF [53]		0.08	+39.11	28.11	0.015	0.09	0.04	0.08	0.62
MuvieNeRF [56]		0.01	-119.28	25.89	0.131	0.53	0.08	0.81	0.35
VersatileGaussian		**34.68**	**+63.96**	**34.57**	**0.003**	0.04	**0.01**	0.05	**0.96**

5 Experiments

In this section, we present the experimental evaluation of our method. We first introduce our implementation details and experimental settings. Then, we compare both the efficiency and effectiveness of our methods to the state-of-the-art multi-task view synthesis methods, both quantitatively and qualitatively. We also ablate the main mechanisms of our method to validate our design.

5.1 Experimental Settings

Tasks: Consistent with [56], we conduct experiments on six tasks, consisting of RGB synthesis (RGB), semantic segmentation (SL), surface normal estimation (SN), texture edge detection (ED), keypoint detection (KP) and reshading (SH). **Datasets:** We evaluate our method on both synthesis and real-world datasets. For synthetic data, we use the Replica [38] dataset provided in [57]. We sample two fragments from each of the eight scenes, with each fragment consisting of 75 frames, covering 300 frames out of 900 in each scene. Four-fifths of these frames are used for training, and the remaining frames are used for evaluation. For real-world data, we adopt ScanNet [9] following NerfingMVS [47]. We sample eight scenes, each consisting of 40 frames, with 35 frames used for training and the rest for testing. The image size for both datasets is 480×640. **Versatile Label Acquisition:** In both datasets, the procedure of versatile label acquisition is the same. For RGB reconstruction and semantic segmentation, the ground-truth labels are provided in the datasets. For surface normal estimation, we calculate the normal map from the ground-truth depth map. For edge detection, the ground-truth labels are generated by the canny detector [4] from the ground-truth instance map, followed by the dilation algorithm and Gaussian filtering. For key point detection, the Difference-of-Gaussians [27] (DoG) detector is applied to the RGB image to generate labels for supervision and evaluation. For reshading, following [56], we adopt the pre-trained model provided in [52] to generate reshading maps. **Metrics for Evaluation:** For RGB Synthesis, we use the Peak Signal-to-Noise Ratio (PSNR), which is commonly used to evaluate images. For semantic segmentation, mean Intersection over Union (mIoU)

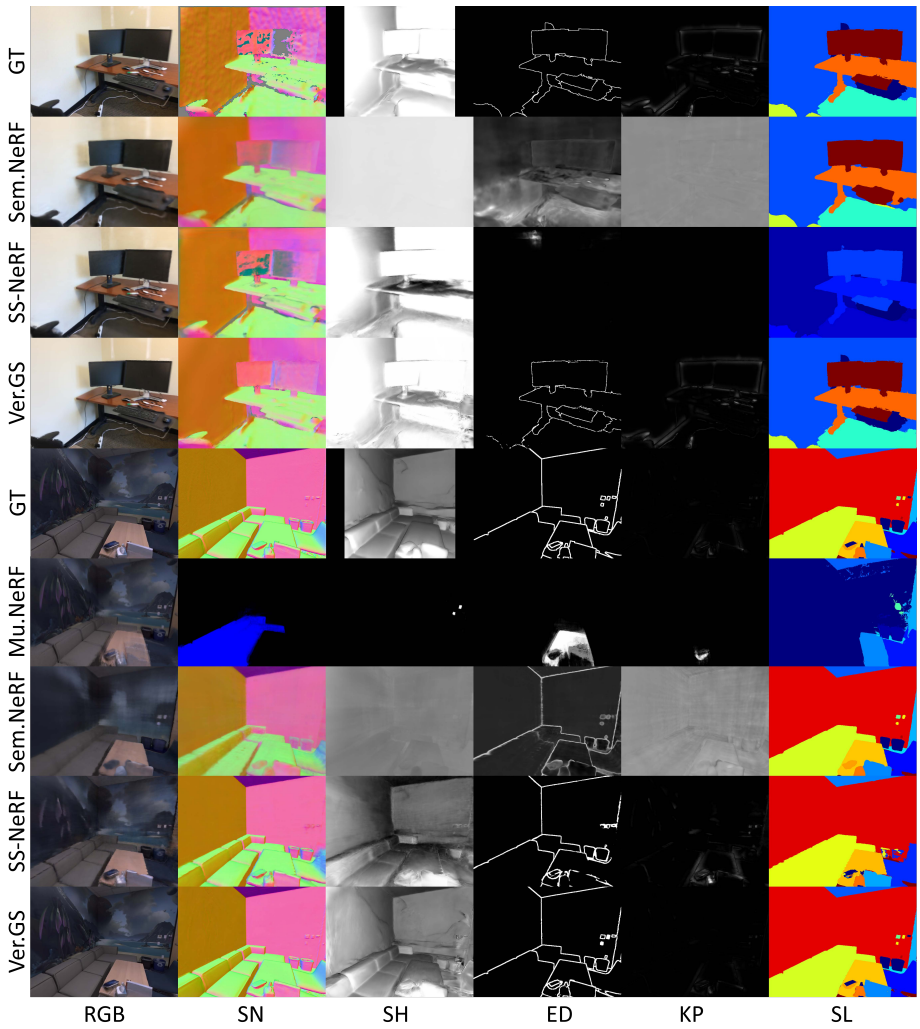

Fig. 4: Visualization of the 3D approaches on Replica and ScanNet dataset. The top four rows show results on ScanNet, and the bottom five rows show results on Replica. GT refers to the ground-truth label, Mu.NeRF to Muvie NeRF [56], Sem.NeRF to Semantic NeRF [57], SS-NeRF to Scene-Property Synthesis with NeRF [53], and Ver.GS to our VersatileGaussian. The columns from left to right display labels for RGB, SN, SH, ED, KP, and SL, respectively.

Table 2: Quantitative Comparison in ScanNet Datasets. The SemanticNeRF is selected as the reference for the overall accuracy. It is clear that VersatileGaussian achieves the best performance on most tasks with significantly faster rendering speed.

Methods	Category.	FPS	Overall	RGB	KP	SH	ED	SN	SL
			$\Delta_m(\%)\uparrow$	PSNR↑	$\mathcal{L}_1\downarrow$	$\mathcal{L}_1\downarrow$	$\mathcal{L}_1\downarrow$	$\mathcal{L}_1\downarrow$	mIoU↑
invPT [50]	2D	6.02	-	-	0.008	0.16	0.06	0.13	**0.91**
SemanticNeRF [57]	3D	0.05	0.00	23.63	0.555	0.14	0.25	0.10	0.74
SS-NeRF [53]		0.08	+22.41	24.64	0.081	0.18	0.03	0.12	0.78
VersatileGaussian		**35.1**	**+54.16**	**26.27**	**0.007**	**0.04**	**0.02**	**0.07**	0.90

is adopted as the evaluation metric. The remaining tasks are evaluated using the L1 error. Further, we employ multi-task learning performance as in [28, 43] to evaluate the overall performance among all tasks. The overall accuracy is measured by the average relative performance gain as:

$$\Delta_m = \frac{1}{T}\sum_{i}^{T}(-1)^{l_i}(M_{m,i} - M_{b,i})/M_{b,i} \tag{11}$$

where $M_{m,i}$ and $M_{b,i}$ are the metrics of task i for the model m and baseline b respectively. l_i is set to 0 if a higher value means better performance or 1 otherwise. Also, to evaluate the efficiency of different methods, the frames-per-second (FPS) metric is employed to measure the rendering speed.

5.2 Implementation Details

We implemented VersatileGaussian using the PyTorch framework, incorporating customized CUDA kernels for both the forward and backward passes of feature rasterization. This extension builds upon the rasterizer provided in [20]. We train and evaluate our method on a single RTX3090 GPU. **Model Details:** To make it efficient, we use a small feature dimension where $d^v = 12, d^n = 32, d = 32$, and $H = 2$. Other hyperparameters are derived from Gaussian Splatting [20]. **Optimization Parameters:** The optimization parameters such as warm-up iterations, intervals to densify Gaussians, learning rates, etc., are derived from 3D-GS [20]. **Losses:** The individual loss for surface normal estimation, reshading, keypoint detection, and object edge detection is $\mathcal{L}_1(\boldsymbol{y}, \hat{\boldsymbol{y}}) = |\boldsymbol{y} - \hat{\boldsymbol{y}}|$, where $\boldsymbol{y}$ is the predicted label and $\hat{\boldsymbol{y}}$ is the ground-truth label. For semantic segmentation, we use the Weighted Balanced Cross-Entropy loss $\mathcal{L}^{\text{SL}} = -\frac{1}{C}\sum_{k=1}^{C}\omega_k\hat{\boldsymbol{y}}_k\log(\boldsymbol{y}_k)$, where $\boldsymbol{y}_k$ is the k^{th} component of vector $\boldsymbol{y}$, and ω_k the weight for the k^{th} class, which is calculated over all training views. The individual loss or RGB synthesis is a linear combination of $L1$ and D-SSIM: $\mathcal{L}^{\text{rgb}} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}$, where λ is set to 0.2 in practice. The weights of tasks in the total loss are set as $\lambda^{\text{RGB}} = 0.6$, $\lambda^{\text{SL}} = 0.5$ and $\lambda^{\text{SN}} = \lambda^{\text{ED}} = \lambda^{\text{KP}} = \lambda^{\text{SH}} = 0.1$.

Table 3: Ablation Studies on Replica and ScanNet. Evaluation of the two key designs of VersatileGaussian on both Replica and ScanNet.

Variant	DataSet	Overall $\Delta_m(\%)\uparrow$	RGB PSNR↑	KP $\mathcal{L}_1\downarrow$	SH $\mathcal{L}_1\downarrow$	ED $\mathcal{L}_1\downarrow$	SN $\mathcal{L}_1\downarrow$	SL mIoU↑
Explicit	Replica	0.00	34.32	0.0041	0.11	0.011	0.058	0.95
+FF		+10.95	32.86	0.0037	0.05	**0.008**	0.065	0.95
+FF+TCA		**+14.41**	**34.57**	**0.0035**	**0.04**	0.009	**0.056**	0.95
Explicit	ScanNet	0.00	24.76	0.0085	0.046	0.0230	0.081	0.89
+FF		+1.35	24.60	0.0084	0.044	0.0223	0.081	0.90
+FF+TCA		**+8.65**	**26.27**	**0.0077**	**0.039**	**0.0201**	**0.075**	**0.91**

5.3 VersatileGaussian is Efficient and Effective

We compare our method with representative baselines to demonstrate its efficiency and effectiveness. Details regarding the datasets, label acquisition procedures, and evaluation metrics are described in Sec. 5.1 **Baselines:** We consider both 2D and 3D baselines. (1) For 3D baselines, we choose recent MuvieNeRF [56]. As it requires extra data at the training stage, we adopt its original settings and training data on Replica and then fine-tune the framework on our sampled fragments. At the testing stage, when evaluating a scene, all the source views with versatile label maps and corresponding camera pose are provided to the framework and we evaluate the outputs at given testing camera poses. We also choose two approaches that are in the per-scene optimization schema, namely Scene-Property Synthesis with NeRF (SS-NeRF) [53] and a simple extension of Semantic NeRF [57]. Similar to the way in which Semantic NeRF generates the semantic map, we extend it with other tasks by simply adding additional shallow MLPs as individual task headers. Then all the task labels are rendered using the standard volumetric rendering equation in NeRF [29], with a shared density field. For the headers corresponding to view-dependent tasks, the standard view direction encoding is provided. For per-scene fine-tuning methods, at the training stage, all source views are provided to optimize the 3D representation, and then they are tested at testing views. (2) For 2D baselines, we choose invPT [50], which takes as input an RGB image and outputs other labels, without the requirement of known camera poses. At the training stage, on Replica and ScanNet, we provide extra 3600 and 4440 frames respectively, including frames at all source views of our testing scenes, as training data. As 2D methods require RGB images as input, at the testing stage, the ground-truth RGB images at testing views are fed into the 2D frameworks, and all tasks except RGB synthesis are evaluated. **Quantitative Results:** As shown in Tab. 1 and Tab. 2, VersatileGaussian significantly outperforms existing approaches in both overall multi-task accuracy and every single task. Notably, VersatileGaussian is at least 400 times faster than existing 3D approaches. **Qualitative Results:** We show some visual samples generated by VersatileGaussian and existing 3D methods.

Table 4: Results on Tanks and Temples Datasets. Incorporating additional tasks from off-the-shelf methods improves RGB rendering quality. The first row is derived from 3D-GS [20].

Models	RGB	keypoint detection	edge detection
3D-GS [20]	23.14	-	-
Ours (2-Tasks)	23.92 (↑ 0.78dB)	0.014	-
Ours (3-Tasks)	25.57 (↑ 2.43dB)	0.012	0.050

As shown in Fig. 4, some approaches do not produce reasonable results in certain cases. Compared to the rest approaches, VersatileGaussian achieves better rendering quality. For example, VersatileGaussian renders clearer RGB images, more accurate SH and SL maps, and more sharp and complete ED maps.

5.4 Boosting 3D-GS with Versatile Labels

We highlight that the RGB synthesis results of 3D-GS can be enhanced using versatile labels, even when these dense labels are inferred from off-the-shelf 2D models. We employ widely-used Tanks and Temples dataset [21] in our experiment. Experimental settings such as the hyper-parameters, the optimization parameters, and the train-test splits are derived from [20]. We train the RGB synthesis task with other tasks (KP from the DoG feature and ED from the canny algorithm), and a performance gain can be observed in Tab. 4.

5.5 Ablation Studies

We conduct ablation studies to justify our designs. To achieve versatile label rendering with 3D-GS, We start from the **Explicit** extension of 3D-GS, where versatile labels are modeled as explicit attributes attached to each Gaussian and rendered following the standard splatting procedure mentioned in [20]. Specifically, view-dependent labels are modeled as SH coefficients, and view-independent labels are modeled as the label value itself. However, for some tasks, data of training views may be noisy (from 2D models or human annotation) or incomplete (from sensors), to which the training of Explicit extension could be sensitive. We further adopt Feature Fields (**FF**) for 3D-GS, where point-wise common features are attached to each Gaussian. Given any target view, a feature map is rasterized from the Gaussians and an individual MLP is employed for each task to read out the explicit labels from the feature map. Each MLP contains two layers and 64 hidden dimensions, which leads to a similar total parameter count compared to the full model. Considering tasks may benefit from each other, we design the **TCA** module to enable cross-task propagation, leading to better details. The quantitative and qualitative results are shown in Tab. 3 and Fig. 5, from which two observations are made: **(1) Learning Versatile Labels is not Trivial:** As shown in Tab. 3 and Fig. 5, The Explicit extension (labeled as Explicit) of 3D-GS yielded moderate performance in tasks such as RGB, Reshading, and

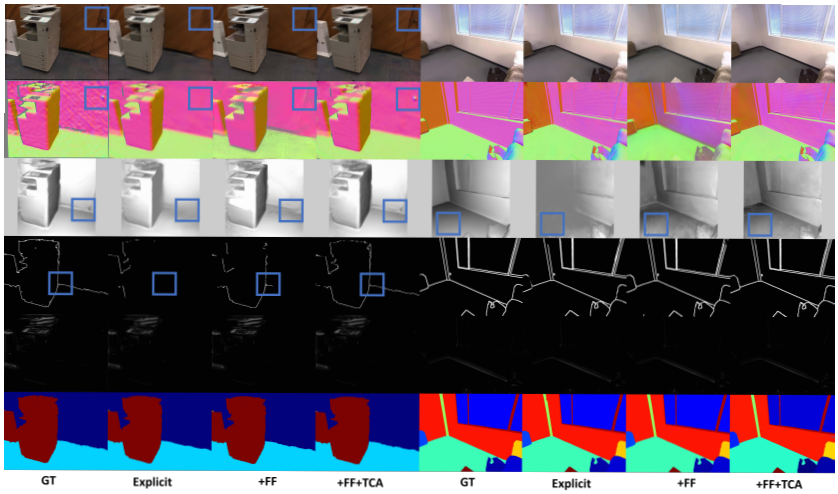

Fig. 5: Visualization of variants in ablation studies. The left 4 columns are from ScanNet and the right 4 columns are from Replica. For each scene, the 4 columns from left to right are the GT label, the Explicit model, the Explicit+FF model, and the Explicit+FF+TCA model (the full model). The rows from top to bottom are RGB, SN, SH, ED, KP, and SL.

Edge. This issue is largely attributable to multi-task labels on training views being noisy (from 2D models) or incomplete (from sensors) for some tasks and the Explicit representation being sensitive to these noises. The model struggles to differentiate between relevant and irrelevant features, as there is no feature learning or cross-task propagation in the explicit baseline method, also a challenge exists in the 2D MTL paper [34]. **(2) Cross Task Feature Propagation Helps with Clearer Details:** As shown in Fig. 5, the columns labeled with +**FF**+**TCA** achieve better rendering quality at details such as corners and door knobs, compared to those without the TCA module (labeled with +**FF**). This is owing to the cross-task feature propagation enabled by the TCA module, which helps each individual task get cues from other related tasks.

6 Conclusion

We present **VersatileGaussian**, the first framework that enables real-time, high-quality multi-task (MT) rendering at novel views. We propose and implement a novel architecture for learning shared feature vectors on each Gaussian and an MT rasterizer that seamlessly decodes into task-specific features, which enables cross-task information flow during optimization. The Task Correlation Attention within the MT rasterizer fosters cross-task correlations, substantially enhancing MT rendering quality. Experiments on Replica and ScanNet datasets show that VersatileGaussian achieves better rendering quality and faster rendering speed than existing methods.

References

1. Brüggemann, D., Kanakis, M., Obukhov, A., Georgoulis, S., Van Gool, L.: Exploring relational context for multi-task dense prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 15869–15878 (2021)
2. Brüggemann, D., Kanakis, M., Obukhov, A., Georgoulis, S., Van Gool, L.: Exploring relational context for multi-task dense prediction. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15849–15858 (2021). https://doi.org/10.1109/ICCV48922.2021.01557
3. Buehler, C., Bosse, M., McMillan, L., Gortler, S., Cohen, M.: Unstructured lumigraph rendering. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 497–504 (2023)
4. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence (6), 679–698 (1986)
5. Chaurasia, G., Duchene, S., Sorkine-Hornung, O., Drettakis, G.: Depth synthesis and local warps for plausible image-based navigation. ACM Transactions on Graphics (TOG) **32**(3), 1–12 (2013)
6. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
7. Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: ICML (2018), https://openreview.net/forum?id=H1bM1fZCW
8. Cipolla, R., Gal, Y., Kendall, A.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7482–7491 (2018). https://doi.org/10.1109/CVPR.2018.00781
9. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
10. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 465–474 (2023)
11. Eisemann, M., De Decker, B., Magnor, M., Bekaert, P., De Aguiar, E., Ahmed, N., Theobalt, C., Sellent, A.: Floating textures. In: Computer graphics forum. vol. 27, pp. 409–418. Wiley Online Library (2008)
12. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5501–5510 (2022)
13. Gafni, G., Thies, J., Zollhofer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8649–8658 (2021)
14. Gao, Y., Ma, J., Zhao, M., Liu, W., Yuille, A.L.: Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3200–3209 (2019). https://doi.org/10.1109/CVPR.2019.00332
15. Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.: Fastnerf: High-fidelity neural rendering at 200fps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14346–14355 (2021)
16. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: 2007 IEEE 11th International Conference on Computer Vision. pp. 1–8. IEEE (2007)

17. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics (ToG) **37**(6), 1–15 (2018)
18. Hedman, P., Srinivasan, P.P., Mildenhall, B., Barron, J.T., Debevec, P.: Baking neural radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5875–5884 (2021)
19. Kajiya, J.T., Von Herzen, B.P.: Ray tracing volume densities. ACM SIGGRAPH computer graphics **18**(3), 165–174 (1984)
20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (ToG) **42**(4), 1–14 (2023)
21. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG) **36**(4), 1–13 (2017)
22. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation. Advances in Neural Information Processing Systems **35**, 23311–23330 (2022)
23. Kopanas, G., Philip, J., Leimkühler, T., Drettakis, G.: Point-based neural rendering with per-view optimization. In: Computer Graphics Forum. vol. 40, pp. 29–43. Wiley Online Library (2021)
24. Kundu, A., Genova, K., Yin, X., Fathi, A., Pantofaru, C., Guibas, L.J., Tagliasacchi, A., Dellaert, F., Funkhouser, T.: Panoptic neural fields: A semantic object-aware neural scene representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12871–12881 (2022)
25. Liu, S., Johns, E., Davison, A.J.: End-to-end multi-task learning with attention. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1871–1880 (2019). https://doi.org/10.1109/CVPR.2019.00197
26. Loper, M.M., Black, M.J.: Opendr: An approximate differentiable renderer. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13. pp. 154–169. Springer (2014)
27. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision. vol. 2, pp. 1150–1157. Ieee (1999)
28. Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1851–1860 (2019)
29. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
30. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
31. Murez, Z., Van As, T., Bartolozzi, J., Sinha, A., Badrinarayanan, V., Rabinovich, A.: Atlas: End-to-end 3d scene reconstruction from posed images. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16. pp. 414–431. Springer (2020)
32. Reiser, C., Peng, S., Liao, Y., Geiger, A.: Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14335–14345 (2021)
33. Rückert, D., Franke, L., Stamminger, M.: Adop: Approximate differentiable one-pixel point rendering. ACM Transactions on Graphics (ToG) **41**(4), 1–14 (2022)

34. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
35. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018), https://proceedings.neurips.cc/paper/2018/file/432aca3a1e345e339f35a30c8f65edce-Paper.pdf
36. Shazeer, N., Lan, Z., Cheng, Y., Ding, N., Hou, L.: Talking-heads attention. CoRR **abs/2003.02436** (2020), https://arxiv.org/abs/2003.02436
37. Siddiqui, Y., Porzi, L., Buló, S.R., Müller, N., Nießner, M., Dai, A., Kontschieder, P.: Panoptic lifting for 3d scene understanding with neural fields. arXiv preprint arXiv:2212.09802 (2022)
38. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
39. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5459–5469 (2022)
40. Takikawa, T., Litalien, J., Yin, K., Kreis, K., Loop, C., Nowrouzezahrai, D., Jacobson, A., McGuire, M., Fidler, S.: Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11358–11367 (2021)
41. Teichmann, M., Weber, M., Zöllner, M., Cipolla, R., Urtasun, R.: Multinet: Real-time joint semantic reasoning for autonomous driving. In: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 1013–1020 (2018). https://doi.org/10.1109/IVS.2018.8500504
42. Tschernezki, V., Laina, I., Larlus, D., Vedaldi, A.: Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In: 2022 International Conference on 3D Vision (3DV). pp. 443–453. IEEE (2022)
43. Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L.: Multi-task learning for dense prediction tasks: A survey. IEEE transactions on pattern analysis and machine intelligence **44**(7), 3614–3633 (2021)
44. Vandenhende, S., Georgoulis, S., Van Gool, L.: Mti-net: Multi-scale task interaction networks for multi-task learning. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16. pp. 527–543. Springer (2020)
45. Vandenhende, S., Georgoulis, S., Van Gool, L.: Mti-net: Multi-scale task interaction networks for multi-task learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 527–543. Springer International Publishing, Cham (2020)
46. Wang, B., Chen, L., Yang, B.: Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. arXiv preprint arXiv:2208.07227 (2022)
47. Wei, Y., Liu, S., Rao, Y., Zhao, W., Lu, J., Zhou, J.: Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5610–5619 (2021)
48. Xu, D., Ouyang, W., Wang, X., Sebe, N.: Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 675–684 (2018). https://doi.org/10.1109/CVPR.2018.00077

49. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5438–5448 (2022)
50. Ye, H., Xu, D.: Inverted pyramid multi-task transformer for dense scene understanding. In: European Conference on Computer Vision. pp. 514–530. Springer (2022)
51. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5752–5761 (2021)
52. Zamir, A., Sax, A., Yeo, T., Kar, O., Cheerla, N., Suri, R., Cao, Z., Malik, J., Guibas, L.: Robust learning through cross-task consistency. arXiv (2020)
53. Zhang, M., Zheng, S., Bao, Z., Hebert, M., Wang, Y.X.: Beyond rgb: Scene-property synthesis with neural radiance fields. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 795–805 (2023)
54. Zhang, Z., Cui, Z., Xu, C., Jie, Z., Li, X., Yang, J.: Joint task-recursive learning for semantic segmentation and depth estimation. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
55. Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., Yang, J.: Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4101–4110 (2019). https://doi.org/10.1109/CVPR.2019.00423
56. Zheng, S., Bao, Z., Hebert, M., Wang, Y.X.: Multi-task view synthesis with neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21538–21549 (2023)
57. Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.J.: In-place scene labelling and understanding with implicit scene representation. In: ICCV (2021)
58. Zhou, L., Cui, Z., Xu, C., Zhang, Z., Wang, C., Zhang, T., Yang, J.: Pattern-structure diffusion for multi-task learning. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4513–4522 (2020). https://doi.org/10.1109/CVPR42600.2020.00457