Improving 2D Feature Representations by 3D-Aware Fine-Tuning

Yuanwen Yue[1], Anurag Das[2], Francis Engelmann[1,3],
Siyu Tang[1], and Jan Eric Lenssen[2]

[1] ETH Zurich [3] Google
[2] Max Planck Institute for Informatics, Saarland Informatics Campus

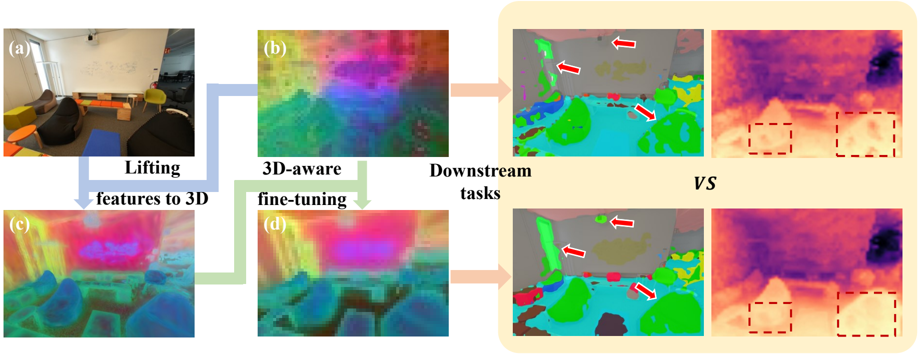

Fig. 1: We propose 3D-aware fine-tuning to improve 2D foundation features. Our method starts with lifting *2D image features* (*e.g.* DINOv2 [42]) **(b)** to a 3D representation. Then we finetune the 2D foundation model using the *3D-aware features* **(c)**. We demonstrate that incorporating the *fine-tuned features* **(d)** results in improved performance on downstream tasks such as semantic segmentation and depth estimation on a variety of datasets with simple linear probing **(right)**. Feature maps are visualized using principal component analysis (PCA).

Abstract. Current visual foundation models are trained purely on unstructured 2D data, limiting their understanding of 3D structure of objects and scenes. In this work, we show that fine-tuning on 3D-aware data improves the quality of emerging semantic features. We design a method to lift semantic 2D features into an efficient 3D Gaussian representation, which allows us to re-render them for arbitrary views. Using the rendered 3D-aware features, we design a fine-tuning strategy to transfer such 3D awareness into a 2D foundation model. We demonstrate that models fine-tuned in that way produce features that readily improve downstream task performance in semantic segmentation and depth estimation through simple linear probing. Notably, though fined-tuned on a single indoor dataset, the improvement is transferable to a variety of indoor datasets and out-of-domain datasets. We hope our study encourages the community to consider injecting 3D awareness when training 2D foundation models. Project page: https://ywyue.github.io/FiT3D.

Keywords: Representation learning · Foundation models · Gaussian splatting · Scene understanding

1 Introduction

Ever since the emergence of deep neural networks, vision systems are largely trained on 2D datasets. With the scalability of recent architectures, like vision transformers (ViT) [13], several large vision models [7, 24, 33, 42, 47] have been trained from a rich set of 2D images by either supervised or self-supervised learning. Visual foundation models have shown impressive utility as general feature extractors that can be applied to improve results on downstream tasks, such as segmentation [36, 53], depth estimation [30, 48, 59], or correspondence estimation [1, 61]. They are trained on a large amount of readily available 2D images and, thus, learn statistics about object and scene structure in 2D-pixel space.

Images, as a simple projection of our 3D world, are easy to obtain and provide an efficient way to depict the visual world while at the same time discarding explicit 3D geometry information. It is expected that vision systems purely trained on 2D images cannot fully understand the underlying 3D structure of our world [15]. There are several promising properties of our 3D world, for example, multi-view consistency, and multi-view fusion for solving single-view ambiguities. A crucial limitation of the training setups of these models is that they don't fully reason about the 3D structure of seen objects. Training images are presented to the network in an unstructured way, without any multi-view or video correspondences that would allow matching observations of the same object from multiple views. As a consequence, these models have limited 3D understanding of objects observed from, *e.g.*, different views are not producing view-consistent features.

In contrast, when we humans observe images, we effortlessly achieve a holistic understanding by not only perceiving the 2D visual content but also exploiting the inferred underlying 3D structure, which we have learned through lifelong observation of stereo and temporal information. In this work, we investigate if large scale 2D vision models can also profit from equipping them with such 3D-aware understanding abilities induced by showing the right type of data.

To this end, we design a novel two-stage approach to improve the 3D-aware understanding ability of 2D foundation models. In the first stage, we aim to obtain 3D-aware features as training data. Motivated by recent advancements in neural scene representation, we design an approach to lift multi-view 2D foundation features into an efficient 3D Gaussian representation [31]. The lifting process exploits multi-view consistency and allows 2D features from different views to complement each other. Moreover, the fused features (Fig. 1 (c)) exhibit high resolution with fine details thanks to the learned 3D structure, emerging from multi-view RGB guidance. Once trained, the 3D Gaussians can render features for arbitrary views. In the following, we refer to features obtained in this way as 3D-aware. In the second stage, we utilize the rendered 3D-aware features to finetune the 2D foundation models (Fig. 2). To this end, we design an efficient fine-tuning strategy to transfer such 3D awareness into 2D foundation models. After fine-tuning, we evaluate the feature quality on downstream tasks that might profit from a better 3D understanding, namely semantic segmentation and depth estimation. Extensive experiments demonstrate that incorporating the

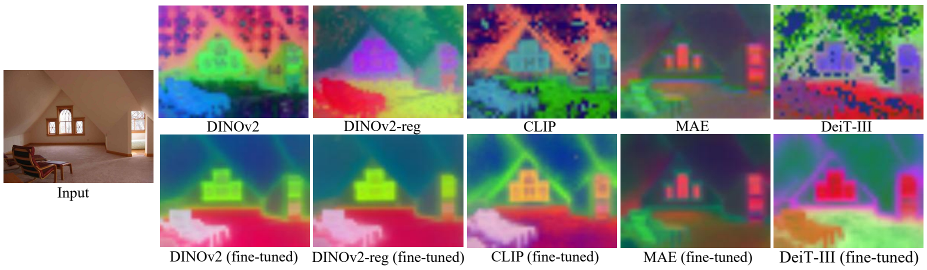

Fig. 2: Our 3D-aware fine-tuning is universal and applicable to a variety of 2D vision models, *e.g.* DINOv2 [42], DINOv2-reg [10], CLIP [46], MAE [24], and DeiT-III [54] (*c.f.* Sec. 4.5).

3D-aware features improves downstream tasks with simple linear probing and exhibits generalization ability on out-of-domain datasets.

2 Related Work

We give an overview about recent self-supervised 2D representation learning techniques in Sec. 2.1, and how emerging features have been distilled into 3D representations in Sec. 2.2. Then, we discuss previous work that utilizes 3D information to improve 2D representation methods in Sec. 2.3.

2.1 2D Representation Learning

Representation learning [4] has achieved remarkable progress in the image domain. It aims to learn generalizable visual features from a rich set of data. Self-supervised representation learning has gained particular interest since it does not require labeled data. Early works employ pretext tasks for pre-training, which aim to exploit inherent data attributes to automatically generate surrogate labels [12,14,20,41,43,56]. Later, contrastive learning [23] has been popularly used for representation learning by leveraging discriminative signals between images or groups of images [6, 7, 21, 25, 42]. More recently, motivated by BERT [11], a new paradigm of masked image modeling [3,8,24] has been proposed for scalable visual learning. Nevertheless, all those methods are only trained on 2D image data, without accessing the underlying 3D structure. Our work aims to supplement the features purely learned from 2D observations with 3D awareness.

2.2 Distilled Feature Fusion Fields

Neural radiance fields (NeRF) [40] emerge as a promising scene representation for high-quality 3D reconstruction and novel view synthesis. Recently, some works [16,32,34,55] explore distilling pre-trained image features (*e.g.* DINO [7], CLIP [46], LSeg [35], or OpenSeg [19]) into NeRF via neural rendering. Without requiring any labels, such distilled feature fusion fields enable several zero-shot 3D scene understanding tasks, *e.g.* segmentation, scene editing, and open-vocabulary queries. We share similar inspiration from these works by distilling

2D features into a 3D representation. However, instead of focusing on perception tasks with feature fields, we are interested in leveraging the rendered 3D-aware features to in turn improve the 2D feature extractor. We demonstrate that the transferred 3D awareness can readily improve the 2D features on both semantic and geometric tasks. Moreover, we extend the recent Gaussian-based representation [31] by designing a method to distill 2D features into 3D Gaussians while keeping high efficiency and memory under bound. There are several concurrent works introducing 3D Gaussians with semantic features [45, 50, 63]. However, none of these works distill features back into 2D models. Our work shows, for the first time, that semantic features fused into 3D representations can effectively improve 2D foundation models via fine-tuning.

2.3 Injecting 3D Priors to 2D

Existing works mainly focus on fusing multi-view 2D features into the 3D representation [22, 28, 29, 39, 44, 49, 52, 57]. Little attention has been paid to the other direction of incorporating 3D awareness into 2D representation learning. Pri3D [27] uses geometric constraints (multi-view consistency and 2D-3D correspondence) from RGB-D reconstructions to learn 3D priors for image-based representations with contrastive learning. Recently, inspired by the masked autoencoder (MAE) [24], several works adopt the masked image modeling strategy to learn 3D priors [2, 26, 58]. However, all these methods require pre-training the 2D feature extractor, typically a Vision Transformer (ViT) backbone [13], using their hand-crafted pretext tasks. The pre-trained models are then employed to downstream tasks via fine-tuning. By contrast, we aim to transfer the 3D awareness embedded in multi-view fused features to the 2D feature extractor through fine-tuning with little computational resources. Our 3D-aware features readily improve downstream task performance with simple linear probing.

3 Method

In this section, we introduce our method for fine-tuning 2D foundation models with 3D-aware features. We present a two-stage pipeline (*c.f.* Fig 3). In the first stage, we lift multi-view inconsistent 2D features into a multi-view consistent and 3D-aware representation. The representation and setup are described in Sec. 3.1. In the second stage, we use the obtained 3D-aware feature representations as training dataset to fine-tune the 2D feature extractor, which is detailed in Sec. 3.2. Last, we describe the linear probing methodology for feature evaluation in Sec. 3.3.

3.1 Lifting Features to 3D

Lifting semantic 2D features into 3D has been a trend recently and several different options exist (*c.f.* Sec. 2). For our purposes of using larger amounts of scenes as training data for 2D models, the most important aspect is efficiency.

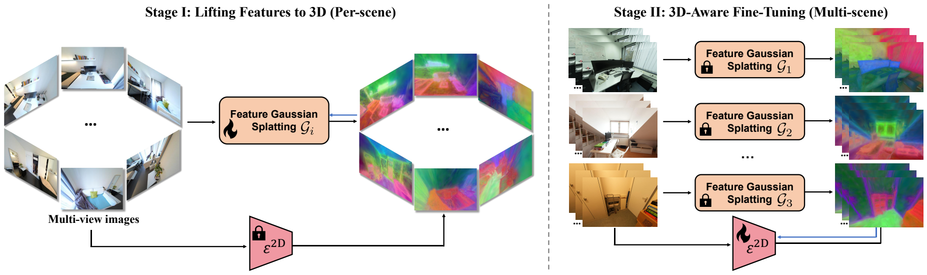
Stage I: Lifting Features to 3D (Per-scene)
...
Feature Gaussian Splatting $\mathcal{G}_i$
...
Multi-view images
ε^{2D}
Stage II: 3D-Aware Fine-Tuning (Multi-scene)
Feature Gaussian Splatting $\mathcal{G}_1$
Feature Gaussian Splatting $\mathcal{G}_2$
...
Feature Gaussian Splatting $\mathcal{G}_3$
ε^{2D}

Fig. 3: Overall pipeline. We present a two-stage pipeline. In the first state, we lift 2D foundation features (*e.g.* DINOv2 [42]) into 3D-aware features by training 3D Gaussian representation $\mathcal{G}_i$. In the second stage, we use the rendered features to finetune the 2D foundation model ε^{2D}. With $\rightarrow$ we denote gradient flow.

The representation needs to (1) be able to efficiently fit a large number of scenes into 3D representations and (2) have a fast rendering mechanism for efficient integration into a fine-tuning loop of a 2D foundation model. Thus, we utilize the recent advances in 3D Gaussian splatting [31], which enable fast optimization and real-time rendering. Fig. 4 illustrates how we extend Gaussian splatting to lift 2D foundation features and we detail the method below.

3D feature Gaussians. Adapting the formulation of 3D Gaussian splatting [31], we define a set of 3D Gaussians as

$$\mathcal{G} = \{(\boldsymbol{\mu}, \mathbf{s}, \mathbf{R}, \alpha, \mathbf{SH}, \mathbf{f})_j)\}_{1 \leq j \leq M}, \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^3$ is the 3D mean of the Gaussian, $\mathbf{S} = \operatorname{diag}(\mathbf{s}) \in \mathbb{R}^{3\times 3}$ is the Gaussian scale, $\mathbf{R} \in \mathbb{R}^{3\times 3}$ its orientation, $\alpha \in \mathbb{R}$ is a per-Gaussian opacity, and $\mathbf{SH}$ a vector of spherical harmonic coefficients, encoding view-dependent color. The Gaussian covariance matrix is obtained by combining scale and orientation as $\mathbf{\Sigma} = \mathbf{RSS}^\top\mathbf{R}^\top$. In addition to the original parameters, we introduce a per-Gaussian feature vector $\mathbf{f} \in \mathbb{R}^D$ to store distilled 2D features in 3D space. Those feature vectors are rasterized into a 2D feature image with our designed feature rasterizer. Inspired by the differentiable color rasterizer of Gaussian splatting, we rasterize the features using point-based α-blending as follows:

$$\mathbf{F}^{\text{low}} = \sum_{i \in \mathcal{N}} \mathbf{f}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_i) \tag{2}$$

where $\mathcal{N}$ is a set of ordered Gaussians overlapping the pixel, $\mathbf{f}_i$ is the feature of each Gaussian and α_i is given by evaluating a 2D Gaussian with covariance $\mathbf{\Sigma}$ multiplied with a learned per-point opacity.

Up-projecting features. A strong limitation of 3D Gaussians as representation is their memory consumption. Since there can be millions of Gaussians per scene, it is impossible to store, *e.g.*, the 384-dimensional DINO features directly on each of the 3D Gaussians. Therefore, to stay memory efficient and keep the fast

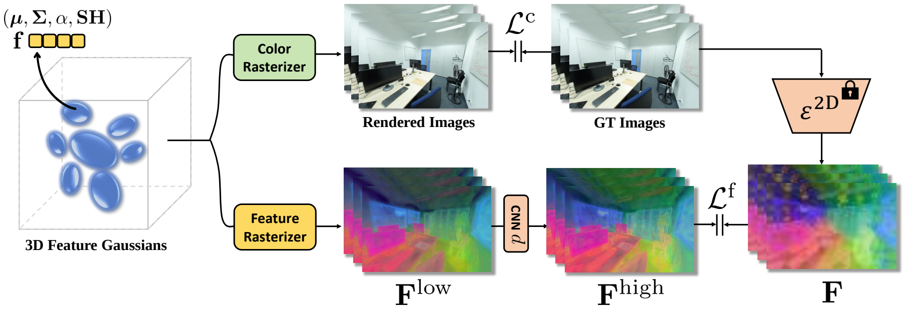
$(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \alpha, \mathbf{SH})$
f
3D Feature Gaussians
Color Rasterizer
Feature Rasterizer
Rendered Images
GT Images
$\mathcal{L}^{\mathrm{c}}$
$\varepsilon^{2\mathrm{D}}$
CNN d
$\mathcal{L}^{\mathrm{f}}$
$\mathbf{F}^{\mathrm{low}}$
$\mathbf{F}^{\mathrm{high}}$
$\mathbf{F}$

Fig. 4: Lifting 2D features into 3D Gaussian representation. We equip each Gaussian with a low-dimensional feature vector $\mathbf{f}$. We render colors using the same color rasterizer as Gaussian splatting [31]. We design a feature rasterizer to render a low-dimensional feature image $\mathbf{F}^{\mathrm{low}}$, which is subsequently projected to a high-dimensional feature image $\mathbf{F}^{\mathrm{high}}$ using a simple CNN. We use 2D foundation features $\mathbf{F}$ from model $\varepsilon^{2\mathrm{D}}$ to supervise the feature learning.

rendering process, we opt for storing lower dimensional features $\mathbf{f} \in \mathbb{R}^D$ with $D \ll 384$ and train a scene-specific pixel-space CNN decoder $d : \mathbf{F}^{\mathrm{low}} \mapsto \mathbf{F}^{\mathrm{high}}$ to up-project feature images into high-dimensional feature space after rendering. We analyze the trade-off introduced by this approach in Sec. 4.6.

Optimization. For a given scene, the full 3D Gaussian representation, including our distilled features, is obtained using optimization. Let $\{\mathbf{I}_i\}_{1\leq i\leq N}$ be a set of multi-view images of a scene with corresponding camera parameters, $\{\mathbf{F}_i\}_{1\leq i\leq N}$ a corresponding set of feature maps from a 2D feature extractor (*e.g.* DINOv2 [42]), and r^{rgb}, r^{feat} rasterization functions that render a set of Gaussians into an RGB or feature image, respectively, using the camera pose $\mathbf{P}_i$ of image i. Then, we optimize the Gaussian parameters, to optimally represent images $\mathbf{I}_i$ and feature images $\mathbf{F}_i$:

$$\hat{\mathcal{G}} = \underset{\{(\boldsymbol{\mu},\mathbf{s},\mathbf{R},\alpha,\mathbf{SH},\mathbf{f})_i\}}{\arg\min} \sum_{i=1}^{N} \mathcal{L}^c(r^{\mathrm{rgb}}(\mathcal{G}, \mathbf{P}_i), \mathbf{I}_i) + \mathcal{L}^f(d(r^{\mathrm{feat}}(\mathcal{G}, \mathbf{P}_i), \mathbf{F}_i), \tag{3}$$

where $\mathcal{L}^c$ is a pixel-wise l_1 loss combined with a D-SSIM term on RGB images, and $\mathcal{L}^f$ is a pixel-wise l_1 loss on feature images. Notably, we only optimize $\mathbf{f}$ with gradients coming from $\mathcal{L}^f$ (feature images) and the rest of the parameters only on $\mathcal{L}^c$ (RGB loss). This has proven to be essential to obtain a consistent 3D feature representation, as a loss from feature space does not lead to correct Gaussian mean, covariance and opacity. We speculate that the reason for this is the missing 3D consistency of the 2D feature extractor. Only through forcing them into a 3D consistent representation, we make them consistent in return.

Algorithm 1 3D-aware fine-tuning algorithm

```
Input: Pre-trained Feature Gaussian representations ${\mathcal{G}_1, ..., \mathcal{G}_K}$, pre-trained 2D feature extractor $\varepsilon_{\theta}^{2D}$, a set of images $\{\mathbf{I}_i\}_{i=1}^{N}$ and associated camera poses $\{\mathbf{P}_i\}_{i=1}^{N}$.
Output: Fine-tuned 2D feature extractor $\varepsilon_{\hat{\theta}}^{2D}$.
Load $\mathcal{G} \sim \{\mathcal{G}_1, ..., \mathcal{G}_K\}$
while fine-tuning do
    Sample an image $\mathbf{I}_i$ and camera pose $\mathbf{P}_i$,    $i \sim \mathcal{U}\{1, N\}$
    Retrieve associated feature Gaussian $\mathcal{G}$ and CNN decoder $d$
    Render $\mathbf{F}^{\text{high}} \leftarrow d(r^{\text{feat}}(\mathcal{G}, \mathbf{P}_i))$
    Step $\theta$ by minimizing $\mathcal{L}(\varepsilon_{\theta}^{2D}(\mathbf{I}_i), \mathbf{F}^{\text{high}})$
end while
return $\varepsilon_{\hat{\theta}}^{2D}$
```

3.2 3D-Aware Fine-Tuning

The procedure described in the last section is used to fit 3D feature Gaussian representations of K scenes. The algorithm of 3D-aware fine-tuning is outlined in Algorithm 1. The fine-tuning process requires training pairs of original 2D feature maps and 3D-aware feature maps. Since it is memory-intensive to save the feature maps, we generate the training pairs on the fly. Considering it is time-consuming to load each pre-trained Gaussian when rendering features, we pre-load all the Gaussians into CPU memory. In each step of the training loop, we randomly sample a view from all the training images, then retrieve its associated feature Gaussian and scene-specific CNN decoder and finally render features $\mathbf{F}^{\text{high}}$ as the ground truth features for fine-tuning. The fine-tuning loss is a l_1 loss between $\mathbf{F}^{\text{high}}$ (resized) and the output features of the fine-tuned 2D feature extractor.

The above design makes the fine-tuning process efficient and keeps memory consumption under control. Notably, we only need to fine-tune the 2D feature extractor with a small number of epochs (*e.g.* 1 epoch for DINOv2 [42]) with a small learning rate without additionally introducing any network component. The fine-tuning process is fast and computation-friendly. An analysis of fine-tuning time is in Sec. 4.6.

3.3 Linear Probing for Downstream Tasks

After fine-tuning on 3D-aware features, we evaluate the emerging features on a set of standard benchmark downstream tasks. To this end, we train a linear head on top of the features to solve tasks of semantic segmentation and depth estimation on several datasets.

Semantic segmentation. A linear layer is trained to predict class logits from patch tokens. The linear layer produces a low-resolution logit map, which is then upsampled to full resolution to obtain a segmentation map.

Depth estimation. We concatenate the `[CLS]` token of the ViT to each patch token. We divide the depth prediction range into 256 uniformly distributed bins [5] and use a linear normalization. Then a simple linear layer is trained using a classification loss.

Feature assembly. We concatenate original 2D features with our fine-tuned features. We observe this is key to preserving the generalization ability of the original 2D feature extractor while incorporating the 3D awareness in our fine-tuned features. Different strategies for feature assembly are evaluated in Sec. 4.6.

4 Experiments

4.1 Datasets

Training. We train the feature Gaussians on ScanNet++ [60], which is a large-scale dataset of 3D indoor scenes containing sub-millimeter resolution laser scans, registered DSLR images, and commodity RGB-D streams from iPhone. We train on the official training split of 230 scenes, which contain 140451 views.

Evaluation. To examine the effectiveness of the fine-tuned features, we conduct extensive experiments on downstream 2D tasks including semantic segmentation and depth estimation. There is no direct competitor in our study and we instead focus on whether our 3D-aware fine-tuning can bring performance gains compared with the standard 2D feature extractor. We conduct most of the experiments with DINOv2 [42] while also demonstrating the generality of our approach with other vision models in Sec. 4.5. We first evaluate on ScanNet++ [60] validation set, which contains 50 scenes with 30638 images. Then we move on to other indoor datasets ScanNet [9] and NYUd [51]. which have a similar data distribution with ScanNet++ but were captured with different sensors. To investigate the generalization ability of the fine-tuned features, we also perform out-of-domain evaluation on generally distributed datasets including ADE20k [62], Pascal VOC [17] and the outdoor dataset KITTI [18].

4.2 Implementation Details

Feature Gaussians. We wrote custom CUDA kernels for feature rasterization. Each Gaussian is initialized with a random feature vector of dimension of 64. We implement the up-projecting CNN with a single convolutional layer with a kernel size of 3×3. We train the feature Gaussians of each scene for novel view synthesis and feature rendering jointly for 30000 iterations.

Fine-tuning. We finetune DINOv2 small with a feature dimension of 384 with a batch size of 2 with a learning rate of 1e-5 for 1 epoch. We use horizontal flip as data augmentation. We use the AdamW [38] optimizer with a weight decay factor 1e-4. The fine-tuning on a single Nvidia Tesla A100 takes 8.5 hours.

Linear probing. We follow the linear probing protocol with DINOv2 [42] to ensure a fair comparison. For semantic segmentation, we train the linear layer for 40K iterations with 8 GPUs. For depth estimation, we train the linear layer for 38400 iterations with 8 GPUs. In addition, we use the same data augmentation and learning rate schedule with DINOv2.

4.3 Within-domain Evaluation

Quantitative comparison. We demonstrate the effectiveness of incorporating our 3D-aware features on downstream semantic segmentation (see Tab. 1) and depth estimation task (see Tab. 2) for indoor scenes. For semantic segmentation task, our 3D aware features consistently improve DINOv2 features, achieving a significant performance gain of 2.6%, 2.0% mIoU, and 1.2% on ScanNet++ [60], NYUv2 [51] and ScanNet [9] datasets, respectively. Our 3D-aware DINOv2 features also improve performance on the depth estimation task. In particular, our enhanced features consistently reduce the RMSE across datasets by achieving 0.34 *vs.* 0.37 (DINOv2) for ScanNet++ [60], 0.42 *vs.* 0.44 (DINOv2) for NYUv2 [51] and 0.29 *vs.* 0.31 (DINOv2) for ScanNet [9] datasets.

Qualitative comparison. We qualitatively show the benefits of 3D-aware features in Fig. 5 and Fig. 6. We observe the improvements are mainly reflected in two aspects: (1) cleaner segmentation/depth estimation in homogeneous or textureless regions, *e.g.* on walls and boards, and (2) better prediction with fine-grained details, *e.g.* on legs of chairs or tables. For (1), during the lifting of 2D features to 3D, features from multiple views are aggregated into a holistic representation, thus information from one view implicitly complements other views. We hypothesize that such multi-view awareness is transferred to DINOv2 through fine-tuning. By contrast, standard DINOv2 struggles to infer accurate segmentation or depth from a single image when with ambiguity, thus leading to noisy prediction. For (2), in our feature lifting process, we train the geometry properties (*e.g.* position and opacity) of Gaussians with RGB color as supervision. The RGB guidance helps feature Gaussians learn detailed 3D structure and render high-resolution feature maps (*c.f.* Fig. 1 (c)). During the fine-tuning process, the model learns to estimate fine-grained features of objects (*c.f.* Fig. 1 (d) *vs.* (b)), which is helpful for capturing detailed structure in downstream tasks.

Table 1: Semantic segmentation scores on indoor datasets. 3D-aware fine-tuning consistently leads to improved performance on semantic segmentation in comparison to standard DINOv2 across different indoor datasets.

	ScanNet++ [60]			NYUv2 [51]			ScanNet [9]		
Method	mAcc (↑)	mIoU (↑)	aAcc (↑)	mAcc (↑)	mIoU (↑)	aAcc (↑)	mAcc (↑)	mIoU (↑)	aAcc (↑)
DINOv2 [42]	40.84	30.19	80.25	76.88	65.55	82.43	55.86	43.6	73.54
+ Ours	**43.4**	**32.76**	**83.54**	**80.52**	**67.5**	**83.37**	**58.32**	**44.84**	**74.37**

Table 2: Depth estimation scores on indoor datasets. 3D-aware fine-tuning consistently leads to improved performance on depth estimation in comparison to standard DINOv2 across different indoor datasets.

	ScanNet++ [60]		NYUv2 [51]		ScanNet [9]	
Method	RMSE (↓)	Rel (↓)	RMSE (↓)	Rel (↓)	RMSE (↓)	Rel (↓)
DINOv2 [42]	0.3742	0.2836	0.4423	0.1392	0.3089	0.1557
+ Ours	**0.3361**	**0.2401**	**0.4198**	**0.1300**	**0.2921**	**0.1459**

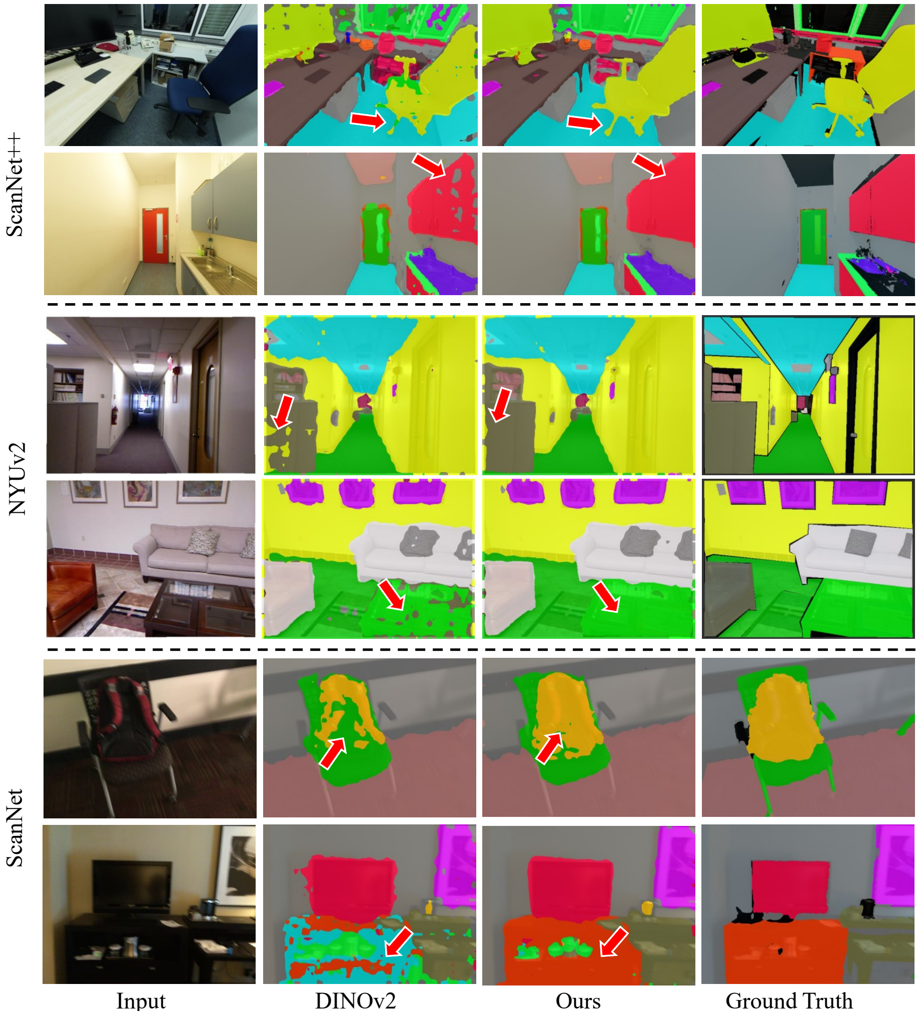

Fig. 5: Qualitative results on semantic segmentation on indoor datasets. After fine-tuning DINOv2 with 3D-aware features, we obtain less noisy and more compact segmentation results, especially for detailed structures and in homogeneous regions.

4.4 Out-of-domain Evaluation

We train feature Gaussians and fine-tune DINOv2 on ScanNet++, a dataset that contains only indoor scenes with the usual content, *e.g.* tables, chairs and other indoor furniture. We want to analyze how the gains obtained in this setting generalize to other domains, *e.g.* outdoor scenes. For semantic segmentation, we conduct linear probing on ADE20k [62] and Pascal VOC [17]. For depth estimation, we conduct linear probing on KITTI [18].

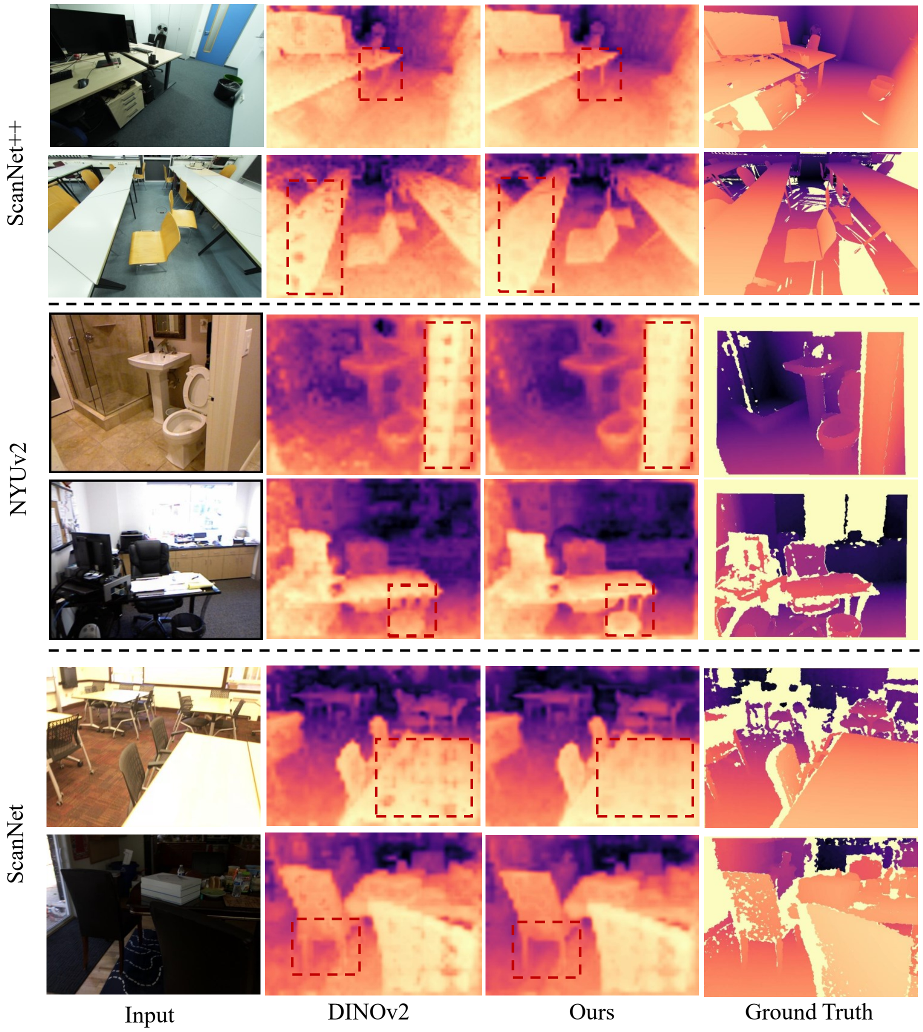

Fig. 6: Qualitative results on depth estimation on indoor datasets. After incorporating 3D-aware features, we obtain cleaner depth in textureless regions and more detailed depth on fine-grained structures, *e.g.* legs of tables or chairs.

Quantitative comparison. We observe that the improvement brought by 3D-aware features is generalizable to out-of-domain challenging datasets and also outdoor driving scenes, although to a smaller degree. As shown in Tab. 3, for semantic segmentation task, incorporating our 3D-aware features brings a gain of 1.6% mIoU on the ADE20k [62] and a gain of 1.2% mIoU on Pascal VOC [17] over standard DINOv2 features. Furthermore, we also compare our performance on urban scene dataset KITTI [18] for depth estimation and observe our 3D-aware features help to reduce RMSE from 3.03 (DINOv2) to 2.91.

Table 3: Quantitative performance on out-of-domain datasets. 3D-aware fine-tuning noticeably improves semantic segmentation on ADE20k and Pascal VOC and depth estimation on KITTI, demonstrating the transferability of the fine-tuned features, even under a significant domain gap.

	ADE20k [62]			Pascal VOC [17]			KITTI [18]	
Method	mAcc (↑)	mIoU (↑)	aAcc (↑)	mAcc (↑)	mIoU (↑)	aAcc (↑)	RMSE (↓)	Rel (↓)
DINOv2 [42]	56.74	44.28	79.73	90.61	81.14	95.72	3.03	0.10
+ Ours	**58.71**	**45.93**	**81.05**	**91.04**	**82.35**	**96.14**	**2.91**	**0.09**

Table 4: Generalization on other 2D vision models. Our 3D-aware fine-tuning applies to other 2D vision models and readily improves their performance.

	DINOv2-reg		**CLIP**		**MAE**		**DeiT-III**	
	mIoU (↑)	RMSE (↓)	mIoU (↑)	RMSE (↓)	mIoU (↑)	RMSE (↓)	mIoU (↑)	RMSE (↓)
Original	30.92	0.4190	25.61	0.4324	17.19	0.4855	18.62	0.4350
+ Ours	**33.39**	**0.3824**	**28.82**	**0.3960**	**20.27**	**0.4795**	**22.98**	**0.3820**

Qualitative comparison. We show qualitative comparison on out-of-domain datasets in Fig. 7. We observe similar improvements as in the within-domain datasets. Even though the 3D-aware fine-tuning is only conducted on indoor dataset ScanNet++, the fine-tuned features exhibit transferability to improve segmentation results for the detailed structures of common objects like bicycle and animal and help denoise segmentation of objects like tree, building and pillar. On depth estimation, the incorporated 3D-aware features are helpful in capturing the detailed structure of trees.

4.5 Generalization to Other Vision Models

We conduct experiments on more vision models (DINOv2-reg [10], CLIP [46], MAE [24], DeiT-III [54]) to prove the universality of our method. We show the linear probing results of semantic segmentation and depth estimation on ScanNet++ validation set in Tab. 4. Our method consistently improves all the models. We also visualize the features in Fig. 2. Note that the original MAE features are already clean but our method still improves them.

4.6 Ablation Studies and Analysis

We conduct ablation studies on semantic segmentation on NYUv2 dataset with DINOv2.

Feature dimension of each Gaussian. We attach a low-dimensional feature vector with each Gaussian and then up-project it to the same space with DINOv2. Tab. 5 indicates that increasing the feature dimension from 32 to 64 will improve the performance of fine-tuned DINOv2 with an acceptable higher memory and longer training time. Increasing the feature dimension further to 128 is

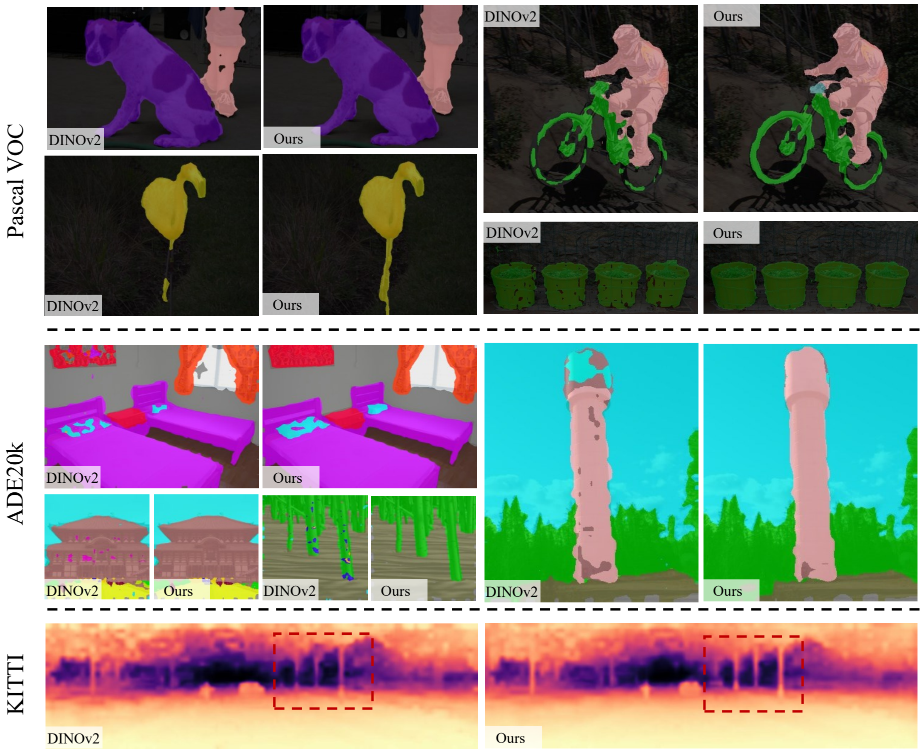

Fig. 7: Qualitative results on out-of-domain datasets. Our 3D-aware features help DINOv2 achieve less noisy segmentation and capture detailed structure.

not feasible in our hardware due to the large memory consumption. We chose a feature dimension of 64 as a good compromise between model performance, memory consumption, as well as training time.

Feature assembly strategy. We study different strategies to assemble the fine-tuned features with the original DINOv2 features in Tab. 7. We explore simple channel-wise adding and concatenation. Alternatively, we first concatenate the fine-tuned features with the original DINOv2 features then use a liner layer to fuse them to the same feature space of DINOv2. We observe simple concatenation works well in incorporating learned 3D-aware features while preserving original generalization ability.

Fine-tuning epochs. We finetune DINOv2 using the features rendered by the pre-trained feature Gaussians. Tab. 8 suggests that a single epoch is sufficient to transfer the 3D awareness to DINOv2 and more epochs may harm the model's generalization ability.

Fine-tuning *vs.* adapter. Besides directly fine-tuning DINOv2, we explore an alternative strategy where we keep DINOv2 frozen and introduce an adapter on top of that. The adapter is a single Swin Transformer block [37]. We observe the

Table 5: Ablation study on feature dimension of 3D Gaussian. Increasing feature dimensions improves performance at the cost of larger memory consumption and longer training time.

Feature dimension	Average memory (MB)	Average Training time (h)	mAcc (↑)	mIoU (↑)	aAcc (↑)
32	370	1.3	78.77	67.15	**83.44**
64	495	1.6	**80.52**	**67.5**	83.37
128	750	2.5	-	-	-

Table 6: Ablation study on fine-tuning *vs.* adapter. An adapter is a tiny network plugged into the frozen DINOv2 features.

Strategy	mAcc (↑)	mIoU (↑)	aAcc (↑)
Fine-tuning	**91.04**	**82.35**	**96.14**
Adapter	90.97	82.02	95.96

Table 7: Ablation study on feature assembly. We study different strategies to assemble fine-tuned features with the original DINOv2.

Strategy	mAcc (↑)	mIoU (↑)	aAcc (↑)
Adding	77.97	66.0	82.85
Linear fusion	78.22	66.39	82.89
Concatenation	**80.52**	**67.5**	**83.37**

Table 8: Ablation study on fine-tuning epochs. We find fine-tuning with a single epoch with 8.5 hours is sufficient to achieve good performance.

Epochs	Fine-tun. time (h)	mAcc (↑)	mIoU (↑)	aAcc (↑)
1	8.5	**80.52**	**67.5**	83.37
2	17	78.72	67.25	**83.54**
3	25.5	79.5	67.18	83.24

adapter can achieve comparable performance with fine-tuning (Tab. 6), however, with longer training time. We stick with the fine-tuning strategy for simplicity without introducing any additional network component.

Limitations and discussion. Our work makes an initial step to transfer multi-view consistent and 3D-aware features encoded by a 3D Gaussian representation to 2D foundation model via fine-tuning. We demonstrate the 3D-aware features are helpful for downstream tasks. However, we still need the original features to keep the generalization ability. We attribute this to the limited diversity of our 3D training data (only a single indoor dataset) and hypothesize that this can be remedied by fine-tuning on larger-scale data.

5 Conclusion

In this work, we present a method to inject 3D awareness into 2D foundation models. We first lift 2D foundation features into a 3D Gaussian representation and then use the rendered multi-view consistent and 3D-aware features to in turn fine-tune the 2D foundation model. Our experiments show that incorporating the fine-tuned features readily leads to improved performance on both semantic and geometric tasks through simple linear probing. Although we only conduct the 3D-aware fine-tuning on a single dataset ScanNet++, we demonstrate the learned 3D awareness is transferable across a variety of datasets in different domains. We hope our work inspires future research to consider equipping 2D foundation models with 3D-aware understanding.

Acknowledgements Francis Engelmann is partially supported by an ETH AI Center postdoctoral research fellowship and an ETH Zurich Career Seed Award.

References

1. Amir, S., Gandelsman, Y., Bagon, S., Dekel, T.: Deep ViT Features as Dense Visual Descriptors. In: European Conference on Computer Vision (ECCV) Workshops (2022) 2
2. Bachmann, R., Mizrahi, D., Atanov, A., Zamir, A.: MultiMAE: Multi-modal Multi-task Masked Autoencoders. In: European Conference on Computer Vision (ECCV) (2022) 4
3. Bao, H., Dong, L., Piao, S., Wei, F.: BEiT: BERT Pre-training of Image Transformers. In: International Conference on Learning Representations (ICLR) (2022) 3
4. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2013) 3
5. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth Estimation Using Adaptive Bins. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 8
6. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In: International Conference on Neural Information Processing Systems (NeurIPS) (2020) 3
7. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-Supervised Vision Transformers. In: International Conference on Computer Vision (ICCV) (2021) 2, 3
8. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative Pretraining From Pixels. In: International Conference on Machine Learning (ICML) (2020) 3
9. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 8, 9
10. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision Transformers Need Registers. In: International Conference on Learning Representations (ICLR) (2024) 3, 12
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL (2018) 3
12. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised Visual Representation Learning by Context Prediction. In: International Conference on Computer Vision (ICCV) (2015) 3
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (ICLR) (2020) 2, 4
14. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative Unsupervised Feature Learning With Convolutional Neural Networks. In: International Conference on Neural Information Processing Systems (NeurIPS) (2014) 3
15. El Banani, M., Raj, A., Maninis, K.K., Kar, A., Li, Y., Rubinstein, M., Sun, D., Guibas, L., Johnson, J., Jampani, V.: Probing the 3D Awareness of Visual Foundation Models. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 2
16. Engelmann, F., Manhardt, F., Niemeyer, M., Tateno, K., Tombari, F.: OpenNeRF: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered

Novel Views. In: International Conference on Learning Representations (ICLR) (2024) 3

17. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision (2015) 8, 10, 11, 12
18. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision Meets Robotics: The Kitti Dataset. The International Journal of Robotics Research (2013) 8, 10, 11, 12
19. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling Open-Vocabulary Image Segmentation With Image-Level Labels. In: European Conference on Computer Vision (ECCV) (2022) 3
20. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised Representation Learning by Predicting Image Rotations. In: International Conference on Learning Representations (ICLR) (2018) 3
21. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap Your Own Latent-a New Approach to Self-Supervised Learning. In: International Conference on Neural Information Processing Systems (NeurIPS) (2020) 3
22. Ha, H., Song, S.: Semantic Abstraction: Open-world 3D Scene Understanding From 2D Vision-Language Models. In: Conference on Robot Learning (CoRL) (2022) 4
23. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality Reduction by Learning an Invariant Mapping. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2006) 3
24. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked Autoencoders Are Scalable Vision Learners. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2, 3, 4, 12
25. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum Contrast for Unsupervised Visual Representation Learning. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 3
26. Hou, J., Dai, X., He, Z., Dai, A., Nießner, M.: Mask3D: Pre-training 2D Vision Transformers by Learning Masked 3D Priors. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 4
27. Hou, J., Xie, S., Graham, B., Dai, A., Nießner, M.: Pri3D: Can 3D Priors Help 2D Representation Learning? In: International Conference on Computer Vision (ICCV) (2021) 4
28. Huang, R., Peng, S., Takmaz, A., Tombari, F., Pollefeys, M., Song, S., Huang, G., Engelmann, F.: Segment3D: Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels. European Conference on Computer Vision (ECCV) (2024) 4
29. Jatavallabhula, K.M., Kuwajerwala, A., Gu, Q., Omama, M., Chen, T., Li, S., Iyer, G., Saryazdi, S., Keetha, N., Tewari, A., et al.: ConceptFusion: Open-set Multimodal 3D Mapping. Robotics: Science and Systems (RSS) (2023) 4
30. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 2
31. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Transactions on Graphics (2023) 2, 4, 5, 6
32. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: LERF: Language Embedded Radiance Fields. In: International Conference on Computer Vision (ICCV) (2023) 3

33. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment Anything. In: International Conference on Computer Vision (ICCV) (2023) 2
34. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing Nerf for Editing via Feature Field Distillation. In: International Conference on Neural Information Processing Systems (NeurIPS) (2022) 3
35. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven Semantic Segmentation. In: International Conference on Learning Representations (ICLR) (2022) 3
36. Li, F., Zhang, H., Xu, H., Liu, S., Zhang, L., Ni, L.M., Shum, H.Y.: Mask DINO: Towards a Unified Transformer-Based Framework for Object Detection and Segmentation. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 2
37. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In: International Conference on Computer Vision (ICCV) (2021) 13
38. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: International Conference on Learning Representations (ICLR) (2019) 8
39. Mazur, K., Sucar, E., Davison, A.J.: Feature-realistic Neural Fusion for Real-time, Open Set Scene Understanding. In: International Conference on Robotics and Automation (ICRA) (2023) 4
40. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: European Conference on Computer Vision (ECCV) (2020) 3
41. Noroozi, M., Favaro, P.: Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In: European Conference on Computer Vision (ECCV) (2016) 3
42. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: DINOv2: Learning Robust Visual Features Without Supervision. Transactions on Machine Learning Research (2023) 1, 2, 3, 5, 6, 7, 8, 9, 12
43. Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B.: Learning Features by Watching Objects Move. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 3
44. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T.: OpenScene: 3D Scene Understanding with Open Vocabularies. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 4
45. Qin, M., Li, W., Zhou, J., Wang, H., Pfister, H.: LangSplat: 3D Language Gaussian Splatting. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 4
46. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning Transferable Visual Models From Natural Language Supervision. In: International Conference on Machine Learning (ICML) (2021) 3, 12
47. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis With Latent Diffusion Models. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2
48. Saxena, S., Herrmann, C., Hur, J., Kar, A., Norouzi, M., Sun, D., Fleet, D.J.: The Surprising Effectiveness of Diffusion Models for Optical Flow and Monocular Depth Estimation. In: International Conference on Neural Information Processing Systems (NeurIPS) (2024) 2

49. Shen, W., Yang, G., Yu, A., Wong, J., Kaelbling, L.P., Isola, P.: Distilled Feature Fields Enable Few-Shot Language-Guided Manipulation. In: Conference on Robot Learning (CoRL) (2023) 4
50. Shi, J.C., Wang, M., Duan, H.B., Guan, S.H.: Language Embedded 3D Gaussians for Open-Vocabulary Scene Understanding. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 4
51. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor Segmentation and Support Inference From RGBD Images. In: European Conference on Computer Vision (ECCV) (2012) 8, 9
52. Takmaz, A., Fedele, E., Sumner, R.W., Pollefeys, M., Tombari, F., Engelmann, F.: OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In: International Conference on Neural Information Processing Systems (NeurIPS) (2023) 4
53. Tan, H., Wu, S., Pi, J.: Semantic Diffusion Network for Semantic Segmentation. In: International Conference on Neural Information Processing Systems (NeurIPS) (2022) 2
54. Touvron, H., Cord, M., Jégou, H.: Deit III: Revenge of the ViT. In: European Conference on Computer Vision (ECCV) (2022) 3, 12
55. Tschernezki, V., Laina, I., Larlus, D., Vedaldi, A.: Neural Feature Fusion Fields: 3D Distillation of Self-Supervised 2D Image Representations. In: International Conference on 3D Vision (3DV) (2022) 3
56. Wang, X., Gupta, A.: Unsupervised Learning of Visual Representations Using Videos. In: International Conference on Computer Vision (ICCV) (2015) 3
57. Weder, S., Blum, H., Engelmann, F., Pollefeys, M.: LabelMaker: Automatic Semantic Label Generation from RGB-D Trajectories. In: International Conference on 3D Vision (3DV) (2024) 4
58. Weinzaepfel, P., Leroy, V., Lucas, T., Brégier, R., Cabon, Y., Arora, V., Antsfeld, L., Chidlovskii, B., Csurka, G., Revaud, J.: CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion. In: International Conference on Neural Information Processing Systems (NeurIPS) (2022) 4
59. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 2
60. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes. In: International Conference on Computer Vision (ICCV) (2023) 8, 9
61. Zhang, J., Herrmann, C., Hur, J., Polania Cabrera, L., Jampani, V., Sun, D., Yang, M.H.: A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence. In: International Conference on Neural Information Processing Systems (NeurIPS) (2023) 2
62. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene Parsing Through ade20K Dataset. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 8, 10, 11, 12
63. Zhou, S., Chang, H., Jiang, S., Fan, Z., Zhu, Z., Xu, D., Chari, P., You, S., Wang, Z., Kadambi, A.: Feature 3DGS: Supercharging 3D Gaussian Splatting to Enable Distilled Feature Fields. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 4