DINO in the Room: Leveraging 2D Foundation Models for 3D Segmentation

Karim Knaebel[1,⋆] Kadir Yilmaz[1,⋆] Daan de Geus[1,2] Alexander Hermans[1]
David Adrian[3] Timm Linder[3] Bastian Leibe[1]
[1]RWTH Aachen University [2]Eindhoven University of Technology [3]Bosch Center for AI
vision.rwth-aachen.de/ditr

Abstract

Vision foundation models (VFMs) trained on large-scale image datasets provide high-quality features that have significantly advanced 2D visual recognition. However, their potential in 3D scene segmentation remains largely untapped, despite the common availability of 2D images alongside 3D point cloud datasets. While significant research has been dedicated to 2D–3D fusion, recent state-of-the-art 3D methods predominantly focus on 3D data, leaving the integration of VFMs into 3D models underexplored. In this work, we challenge this trend by introducing DITR, a generally applicable approach that extracts 2D foundation model features, projects them to 3D, and finally injects them into a 3D point cloud segmentation model. DITR achieves state-of-the-art results on both indoor and outdoor 3D semantic segmentation benchmarks. To enable the use of VFMs even when images are unavailable during inference, we additionally propose to pretrain 3D models by distilling 2D foundation models. By initializing the 3D backbone with knowledge distilled from 2D VFMs, we create a strong basis for downstream 3D segmentation tasks, ultimately boosting performance across various datasets.

1. Introduction

The idiom "elephant in the room" refers to a situation where something important is being ignored, while it should be discussed. Currently, 2D foundation models, such as DINOv2 [35], remain largely ignored for 3D segmentation, despite their ability to provide strong semantic priors. Therefore, we believe that we need to talk about the ~~elephant~~ DINO in the room.

The current paradigm for state-of-the-art 3D segmentation, both in literature and on benchmarks, is to use models with specialized 3D backbones that are trained from scratch [9, 59, 61]. In contrast, for the highly related task of image segmentation, the current paradigm is to use 2D backbones initialized with pretrained weights of strong vision foundation models (VFMs) [7, 13, 24–26, 35, 43, 46, 69]. These VFMs are predominantly trained in a self-supervised manner on large-scale image datasets, enabling strong generalization capabilities. However, current 3D datasets [1, 5, 6, 10, 47, 53, 72] are orders of magnitude smaller than their 2D counterparts [49, 50]. As a result, while progress has been made [63], generalist 3D foundation models have yet to emerge. Interestingly, though, we observe that 3D point cloud data is often accompanied by corresponding 2D images [6, 10, 47, 53]. This raises the question: how can we leverage the power of 2D foundation models for 3D segmentation?

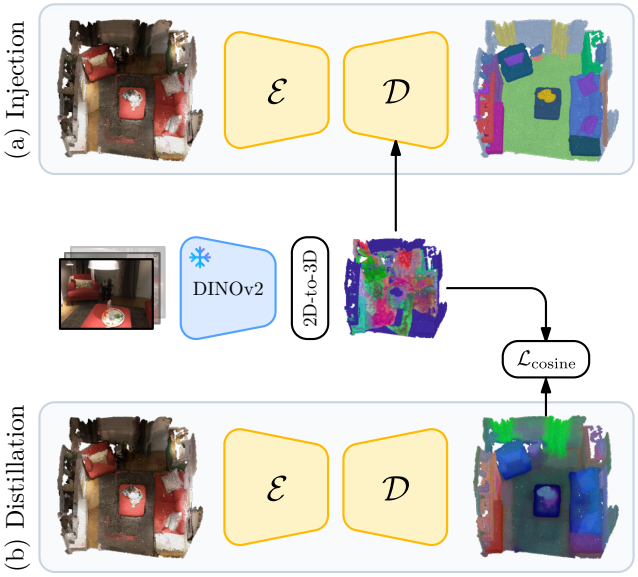

Figure 1. **DINO in the room (DITR)**. We present an approach to (a) inject or (b) distill DINOv2 features into 3D semantic segmentation models that yields state-of-the-art results across indoor and outdoor 3D benchmarks.

For most 3D point clouds, corresponding images are typically available due to the process with which this data is captured. In indoor scenarios, a scene is captured with RGB or RGB-D cameras as a video sequence and a col-

⋆Equal contribution. The order is determined by a last-minute coin flip.

ored point cloud is formed via 3D reconstruction [10, 47]. As a result, the data inherently includes 2D images, and correspondences between images and point clouds are obtained as part of data preprocessing. In outdoor scenarios, 3D street scenes are typically captured with LiDAR scanners mounted on vehicles that are often also equipped with cameras, providing corresponding 2D images [5, 6, 53]. Despite the availability of these images, current state-of-the-art methods for both indoor and outdoor 3D semantic segmentation typically only use 3D data [27, 61, 62, 68]. While 2D–3D fusion has been investigated with moderate success [2, 22, 45, 54, 70], existing works have thus far not capitalized on the existence of semantically rich VFMs.

In this paper, we challenge the current paradigm and claim that VFMs can, as they did in 2D, enrich 3D models with generalist features that are not easily learned from the 3D data. To verify this, we take a state-of-the-art 3D segmentation model [61] and augment it with 2D VFM features. Concretely, we take a frozen DINOv2 [35] and extract 2D image features that correspond to the points in the associated 3D point cloud. Next, for the 3D points that have been matched to 2D features, we inject these 2D features into the 3D model at different decoder stages. Then, we train this model for the regular segmentation objective, resulting in a 2D-to-3D injection approach (see Fig. 1 (a)). With this injection approach, which only uses unlabeled images, the semantic segmentation performance improves significantly, achieving new state-of-the-art results. Especially for indoor datasets with many semantic classes and for outdoor datasets that provide 360° camera coverage, this setup outperforms the 3D-only baseline on public leaderboards by large margins, *i.e.*, +7.1 mIoU on ScanNet200 [47] and +2.4 mIoU on nuScenes [6]. Furthermore, it achieves the top score on the recent ScanNet++ benchmark [67], outperforming the next best approach by +3.0 mIoU. We call this approach *DINO In The Room* (DITR).

Moreover, we show that even if images are not available during inference, it is still possible to utilize features from the same frozen DINOv2 model to enhance segmentation performance by pretraining 3D models through *distillation* [17]. Specifically, we take 3D point cloud datasets that have corresponding images and teach a 3D student model to output features aligned with those extracted from a DINOv2 teacher model, using a distillation objective (see Fig. 1 (b)). Subsequently, the pretrained, distilled student model can simply be fine-tuned for 3D segmentation in a regular fashion, without requiring corresponding images, and thus without causing any overhead during inference. This distillation pretraining enables the 3D model to capture the semantic richness of 2D foundation models without requiring any labeled data, effectively using them as distillation targets instead of semantic labels. Furthermore, it allows pretraining across multiple datasets without adjusting for dataset-specific semantic label sets, as the target feature space remains consistent. With this setup, we observe consistent improvements on all datasets compared to random initialization. On SemanticKITTI [5], where there is only a single image per 3D LiDAR scan, and on S3DIS [1], where camera views are sparser, this distillation setup even outperforms DITR. We call this approach *Distill DITR* (D-DITR).

Altogether, these results highlight the significant yet underexplored potential of VFMs for 3D segmentation tasks. Therefore, we recommend that whenever corresponding images are available, they should be used to complement 3D segmentation models, preferably using injection and otherwise using distillation. Additionally, we note that while we focus on DINOv2 in this paper, as it is one of the strongest existing VFMs, we show that the general setup can also leverage even stronger models, like the very recently introduced DINOv3 [51].

In summary, our contributions are as follows:

- We show that injection of DINOv2 features from 2D images into a 3D model can significantly improve the segmentation performance, achieving new state-of-the-art results on multiple indoor and outdoor datasets.
- We demonstrate that DINOv2 can also act as a teacher to pretrain 3D models through distillation, improving performance without requiring 2D data during inference.

2. Related Work

3D Semantic Segmentation. Despite architectural differences, most 3D backbones for dense tasks such as segmentation follow a U-Net-like hierarchical encoder–decoder design with skip connections. Early methods apply point-wise fully-connected layers [40, 41], or continuous 3D convolutions [37, 56] to capture local geometric patterns. Later, MinkUNet [9] significantly improves efficiency and accuracy by voxelizing point clouds and utilizing sparse 3D convolutions [15]. Following the success of Vision Transformers [12], attention mechanisms have been increasingly adopted for 3D segmentation, although their quadratic computational complexity poses challenges for large-scale scenes. To address this, several methods [29, 58, 59, 66, 71] restrict attention to local neighborhoods, where points are subsampled and aggregated hierarchically across multiple layers. Following this, Point Transformer V3 (PTv3) [61] employs space-filling curves to map 3D point clouds into 1D sequences, effectively preserving the local neighborhood structure while significantly simplifying and accelerating local attention computations. Most recently, Sonata [63] scales up PTv3, tripling the number of parameters, and adopts DINOv2-style self-supervised learning on point clouds, enabling the use of a substantially larger 3D training corpus [1, 3, 4, 10, 44, 67, 72], albeit still orders of magnitude smaller than its 2D counterparts. In contrast, our approach enhances PTv3 by directly incorpo-

rating features from powerful 2D VFMs such as DINOv2, thereby leveraging the vast amount of 2D training data on which these models are pretrained.

2D–3D Fusion. Significant research has been dedicated to fusing 2D and 3D information for 3D segmentation. Existing fusion methods are often designed with a domain-specific focus, addressing either indoor or outdoor scenes.

Indoor fusion methods exploit multi-view information from RGB-D videos, as points are visible from multiple angles. Kundu *et al*. [28] render 2D images and their corresponding ground-truth labels from 3D mesh reconstructions to train a 2D segmentation model. During inference, resulting multi-view 2D predictions are aggregated in 3D space. Other approaches leverage 2D segmentation annotations provided by the 3D datasets [10], either by pretraining 2D networks [23] or by jointly training 2D and 3D networks with interactions between them [19]. Recent methods use dedicated architectures for 2D–3D fusion, introducing either an aggregation module that selectively fuses 2D features into a 3D backbone [45], or a unified 2D–3D model with alternating 2D–3D layers enabling multi-stage fusion of multi-modal information [22].

Outdoor fusion methods combine appearance-based 2D features with geometric LiDAR features for segmentation. Some methods [54, 74] project LiDAR points to the image plane and jointly train a camera and a LiDAR network, while simultaneously aligning their features. Other approaches introduce more sophisticated fusion mechanisms, such as selecting semantically relevant 2D regions for each point via local attention modules [70], or establishing 2D–3D correspondences through joint spatial and semantic reasoning [31]. Finally, 4D-Former [2] explores 2D–3D fusion for LiDAR segmentation and tracking, integrating 2D features into a 3D backbone at multiple stages.

While these works effectively use 2D information for 3D segmentation and achieve modest improvements, we observe that the community has not leveraged 2D VFMs for 3D segmentation, leaving a powerful information source untapped. In this work, to the best of our knowledge, we are the first to explore how the power of VFMs can be leveraged for *state-of-the-art* 3D segmentation, and demonstrate how 2D-to-3D injection with DITR yields new state-of-the-art results. Moreover, DITR is the first fusion approach shown to be effective for both indoor and outdoor scenes, highlighting its general applicability.

2D–3D Distillation. Recent methods explore various strategies to distill 2D representations into 3D backbones for enhanced representation learning in LiDAR point clouds. Yan *et al*. [64] propose a joint training framework, where a 3D model is trained for both 3D segmentation and alignment with a 2D model. Several approaches apply contrastive learning either on super-points and super-pixels generated by pretrained 2D backbones [32, 48], or on prototypes created in both the 2D and 3D domains [8]. Others [33, 34] propose new contrastive loss functions, and conduct experiments with linear probing or limited-data setups. Recently, ScaLR [39] studies the effects of the size of the backbones and datasets in an image-to-LiDAR distillation setting. These methods focus on—and succeed in—image-to-LiDAR distillation for zero-shot, limited-data, or linear probing 3D segmentation settings. However, they often yield little or no improvement when used as pretrained models that are subsequently fine-tuned on 100 % of the data, and fall short of state-of-the-art methods in that setting. In contrast, we demonstrate that our D-DITR distillation *is* an effective pretraining step that significantly improves the fine-tuning performance of a state-of-the-art model (PTv3). Moreover, unlike previous methods, it also works especially well for dense indoor scenes.

Orthogonal to other distillation methods, there is also a line of work that distills specific capabilities from VFMs. Peng *et al*. [38] distill language-aligned CLIP [43] features from multiple views for 3D open-vocabulary segmentation. Other approaches [20, 36] use projected SAM [26] segmentation masks as pseudo ground-truth labels to train class-agnostic 3D segmentation models. Although these approaches also distill from 2D foundation models, they focus on obtaining task-specific capabilities in a zero-shot setting to avoid the reliance on 3D segmentation annotations. In contrast, we use distillation to obtain a pretrained 3D model that extracts semantically rich point features, which can then be fine-tuned for 3D segmentation and, in principle, other 3D tasks as well.

3. Method

We propose two variants of DITR for 3D semantic segmentation: an *injection* approach that uses 2D features from DINOv2 [35] during training and inference, and D-DITR, a *distillation* approach that aligns 3D features with DINOv2 features during a pretraining phase, which can be followed by image-free fine-tuning on a specific dataset. In the following, we describe both variants in detail.

3.1. Injection

In many modern 3D datasets, images from calibrated cameras are provided alongside the point clouds [1, 5, 6, 10, 53]. We leverage these images to inject semantically rich 2D features into the 3D backbone. An overview of this process is shown in Fig. 2: we first map 3D points to their corresponding pixels to extract associated DINOv2 features (2D-to-3D Mapping), and then inject these features into the skip connections of the 3D backbone's decoder (3D Feature Fusion).

2D-to-3D Mapping. Let $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}_{i=1}^{N}$ be a 3D point cloud of N points, and assume a collection of K calibrated

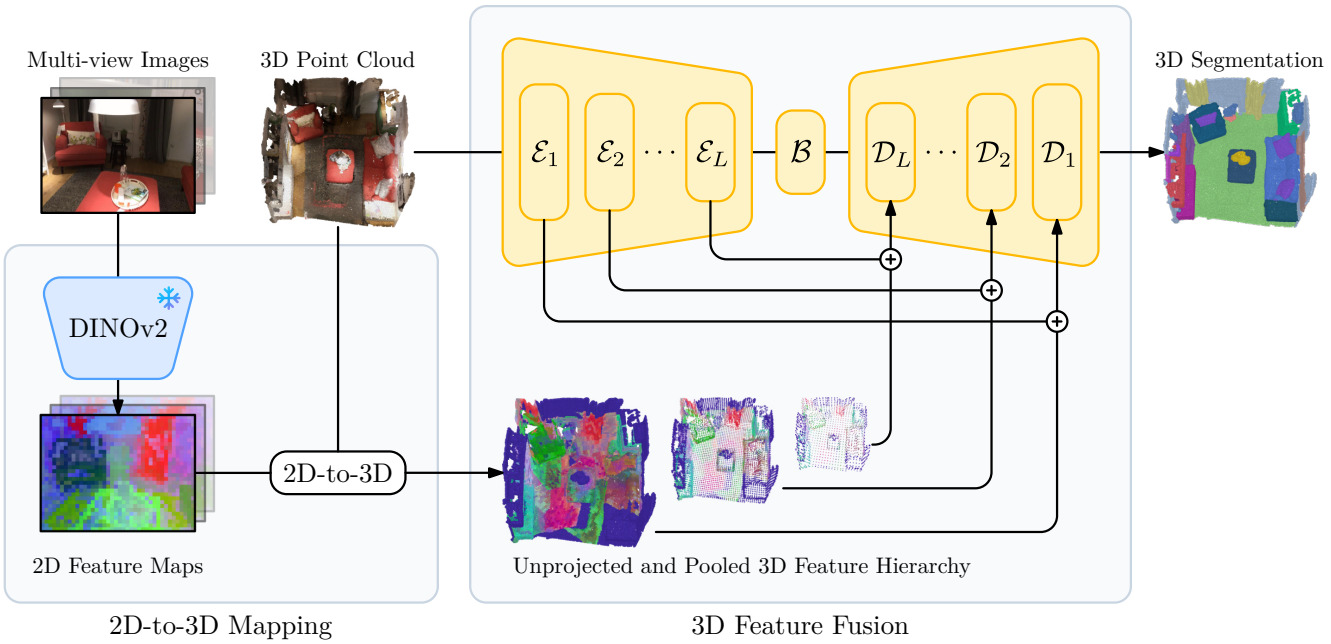

Figure 2. **DITR architecture overview.** We extract 2D image features from a frozen DINOv2 [35] model ❄ and unproject them (2D-to-3D) onto the 3D point cloud. The unprojected features are subsequently max-pooled to create a multi-scale feature hierarchy. The raw point cloud is fed through a 3D backbone 🔥 and the unprojected image features are added to the skip connection between the encoder $\mathcal{E}_l$ and decoder $\mathcal{D}_l$ blocks on each level $l \in \{1, 2, \dots, L\}$. The model is then trained with the regular segmentation loss.

cameras, $\{(\mathbf{I}_k, \mathbf{K}_k, \mathbf{T}_k)\}_{k=1}^{K}$. Each camera has intrinsics $\mathbf{K}_k$ and world-to-camera extrinsics $\mathbf{T}_k$, with $\mathbf{I}_k$ denoting its captured image of resolution $H \times W$. We feed each 2D image $\mathbf{I}_k$ into a frozen DINOv2 ViT [12], obtaining patch-level embeddings $\mathbf{F}_k \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times D_{2\text{D}}}$, where P is the patch size and $D_{2\text{D}}$ is the feature dimension. For each point $\mathbf{p}_i$, we transform it into the k-th camera space as $\mathbf{q}_i^k = \mathbf{T}_k\,(\mathbf{p}_i, 1)$ and multiply by the intrinsics $\mathbf{K}_k$ to obtain homogeneous pixel coordinates $\left(x_i^k,\, y_i^k,\, z_i^k\right) = \mathbf{K}_k \mathbf{q}_i^k$. The pixel coordinates are then given by $\left(u_i^k,\, v_i^k\right) = \left(x_i^k / z_i^k,\, y_i^k / z_i^k\right)$. We consider point $\mathbf{p}_i$ visible in the k-th image if $u_i^k \in [0, W)$, $v_i^k \in [0, H)$ and the depth z_i^k is positive (view-frustum culling). If $\mathbf{p}_i$ is visible in the k-th image, we determine the corresponding patch index as

$$\left(\hat{u}_i^k,\, \hat{v}_i^k\right) = \left(\left\lfloor \frac{u_i^k}{P} \right\rfloor, \left\lfloor \frac{v_i^k}{P} \right\rfloor\right),$$

and assign the corresponding feature from the 2D feature map $\mathbf{F}_k$ to $\mathbf{p}_i$. If $\mathbf{p}_i$ is visible in multiple images, we randomly select one of them to provide the feature. Otherwise, if $\mathbf{p}_i$ is not visible in any image, we assign an all-zero feature vector. Empirically, we find that directly assigning the feature from the frozen 2D feature map $\mathbf{F}_k$ at patch index $(\hat{u}_i^k, \hat{v}_i^k)$ yields better results than bilinearly interpolating features from $\mathbf{F}_k$ at the pixel coordinates (u_i^k, v_i^k). Also, aggregating from multiple 2D feature maps resulted in inferior performance compared to random selection.

3D Feature Fusion. Because point clouds are unstructured, common 3D backbones [9, 61, 73] voxelize the input, keeping only one point per voxel, thereby reducing $\mathcal{P}$ to $\mathcal{P}'$ with $M \leq N$ points. These backbones typically follow a U-Net-like encoder–decoder architecture, processing $\mathcal{P}'$ at multiple spatial resolution levels $l \in \{1, 2, \dots, L\}$. At each encoder level, a pooling layer downsamples the point features to a coarser spatial resolution. Formally, let $\mathbf{X}_l^{\mathcal{E}} \in \mathbb{R}^{M_l \times D_l}$ denote the output of the l-th level encoder block $\mathcal{E}_l$, where $l = 1$ is the finest (original) resolution with $M_1 = M$. After the encoder and the bottleneck $\mathcal{B}$, the decoder upsamples and refines these features back toward the original resolution, yielding $\mathbf{X}_l^{\mathcal{D}}$ as the output of each decoder block $\mathcal{D}_l$. To simplify notation, we define $\mathbf{X}_{L+1}^{\mathcal{D}}$ to be the output of the bottleneck $\mathcal{B}$.

To incorporate DINOv2 features, we first use the previously described assignment to gather point-wise features $\mathbf{X}^{2\text{D}} \in \mathbb{R}^{M \times D_{2\text{D}}}$ for the points in $\mathcal{P}'$ from the 2D feature maps $\mathbf{F}_k$. Then, to mirror the spatial structure of the decoder features $\mathbf{X}_l^{\mathcal{D}}$ across different levels, we repeatedly apply max pooling to $\mathbf{X}^{2\text{D}}$, obtaining features $\mathbf{X}_l^{2\text{D}}$ for each level l. In the decoder, features from the encoder ($\mathbf{X}_l^{\mathcal{E}}$) and unpooled features from the previous decoder level ($\uparrow\mathbf{X}_{l+1}^{\mathcal{D}}$) are combined via skip connections. In DITR, we additionally inject $\mathbf{X}_l^{2\text{D}}$ into these skip connections. Specifically, the fused input to the l-th decoder block $\mathcal{D}_l$ becomes

$$\underbrace{f_l^{\mathcal{D}}(\uparrow\mathbf{X}_{l+1}^{\mathcal{D}})}_{\text{prev. dec. block}} + \underbrace{f_l^{\mathcal{E}}(\mathbf{X}_l^{\mathcal{E}})}_{\text{skip connection}} + \underbrace{f_l^{2\text{D}}(\mathbf{X}_l^{2\text{D}})}_{\text{DINOv2}},$$

where $+$ denotes element-wise addition, and each f_l is a

linear projection into the decoder's feature dimension followed by batch normalization [21] and GELU [16]. Finally, the output $\mathbf{X}_1^{\mathcal{D}}$ of the last decoder block yields per-point features for $\mathcal{P}'$, which are passed through a linear segmentation head to produce class logits. With our injection approach, we ensure that rich 2D VFM features are available at all 3D decoder feature resolutions, enabling access to 2D information where needed and at different levels of granularity.

3.2. Distillation

While injecting DINOv2 features directly into the 3D backbone can significantly boost segmentation performance, relying on calibrated images at inference time may be restrictive in certain real-world scenarios. To address situations where only 3D data can be used at test time, we propose a distillation scheme (D-DITR) to transfer 2D knowledge into a pure 3D model as a pretraining step.

Pretraining via 2D-to-3D Alignment. During pretraining, we first match 3D points to 2D patches and assign per-point DINOv2 features $\mathbf{X}^{\text{2D}}$ as described in Sec. 3.1. We then feed only the point cloud into the 3D backbone, but instead of predicting segmentation logits, the network's final linear layer regresses the DINOv2 features. For each point $\mathbf{p}_i$, we denote the predicted feature by $\mathbf{x}_i^{\text{pred}}$ and the corresponding DINOv2 target feature by $\mathbf{x}_i^{\text{2D}}$. We minimize the following cosine similarity loss [38], averaged over all visible points:

$$\mathcal{L}_{\text{cosine}} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left[1 - \frac{\mathbf{x}_i^{\text{pred}} \cdot \mathbf{x}_i^{\text{2D}}}{\|\mathbf{x}_i^{\text{pred}}\| \, \|\mathbf{x}_i^{\text{2D}}\|} \right], \qquad (1)$$

where $\mathcal{V}$ is the set of indices of points visible in at least one camera. This distillation pretraining encourages the 3D backbone to replicate the semantically rich representations of DINOv2, which capture fine-grained details without being constrained by the coarse granularity of traditional semantic segmentation annotations. An overview of the distillation setup is shown in Fig. 1 (b).

Image-Free Inference. After pretraining, we discard the linear regression head and replace it with the standard segmentation head. The pretrained model can then be finetuned on any 3D segmentation dataset without requiring image features. The final model thus makes predictions based on 3D data only, yet benefits from the semantic knowledge transferred from strong 2D models.

Multi-Dataset Training. A key advantage of our 2D-to-3D distillation approach is that it does not require any annotated data. It only requires aligned 2D and 3D inputs. This enables us to combine multiple unannotated point cloud datasets under a single distillation objective, where each point simply regresses to its corresponding DINOv2 feature. Following prior multi-dataset work [62], we maintain separate batch-normalization layers per dataset, which empirically leads to more stable training.

4. Experiments

4.1. Datasets

We evaluate our method on well-established 3D indoor and outdoor segmentation benchmarks. For indoor segmentation, we select ScanNet [10], ScanNet200 [47], ScanNet++ v2 (SN++) [67], and S3DIS [1]. All four datasets consist of colored point clouds and corresponding RGB-D frames. For outdoor segmentation, we select nuScenes [6], SemanticKITTI (Sem.KITTI) [5], and Waymo [53].

4.2. Implementation Details

We use PTv3 [61] as our 3D backbone, as it is the state-of-the-art model for 3D semantic segmentation across both indoor and outdoor datasets. Unless stated otherwise, we retain the default architecture and hyperparameters of PTv3 to ensure fair comparison. Additional details are provided in the supplementary material. For all datasets, we adhere to standard evaluation protocols and metrics. Reproducing PTv3 results on the S3DIS[1] and SemanticKITTI[2] datasets has been notoriously hard for the community. Therefore, we also report our reproduced results as a reference.

For outdoor scenarios, we observe large performance improvements with larger DINOv2 variants and thus we opt for the largest ViT-g model. For indoor datasets, the ViT-L and ViT-g variants yield comparable results (as shown in Tab. 6). Consequently, we choose the ViT-L model to optimize resource efficiency, enabling the use of more images while remaining within GPU memory constraints.

During both training and inference on outdoor scenes, we use all available camera views: one for SemanticKITTI, five for Waymo, and six for nuScenes. These camera views appear in a consistent orientation with respect to the LiDAR sensor and can be assumed always to be present. For the indoor datasets, the number of RGB-D frames per scene can vary tremendously and always surpasses the number of frames we can effectively process with the available GPU memory. Therefore, unless specified otherwise, we select 10 uniformly sampled, temporally equidistant frames from each RGB-D video during inference, while randomly sampling 10 views per scene during training to enhance data diversity. Since DINOv2 is pretrained at a maximum resolution of 518×518, we resize input images to maintain a similar number of patches while preserving aspect ratio [57].

4.3. Main Results

Injection. Tabs. 1 and 2 show that DITR significantly outperforms the reproduced PTv3 baseline [61] by injecting DINOv2 image features into the PTv3 backbone. These improvements are consistent across both indoor and outdoor benchmarks. We find that the performance improve-

[1] https://github.com/Pointcept/Pointcept/issues/154
[2] https://github.com/Pointcept/Pointcept/issues/186

Method	ScanNet		ScanNet200		S3DIS	SN++
	Val	Test	Val	Test	Area5	Test
ST [29]	74.3	73.7	—	—	72.0	—
PTv1 [71]	70.6	—	27.8	—	70.4	—
PointNeXt [42]	71.5	71.2	—	—	70.5	—
MinkUNet [9]	72.2	73.6	25.0	25.3	65.4	45.6
OctFormer [58]	75.7	76.6	32.6	32.6	—	46.0
Swin3D [66]	76.4	—	—	—	72.5	—
PTv2 [59]	75.4	74.2	30.2	—	71.6	44.5
PTv3 [61]	77.5	77.9	35.2	37.8	73.4†	48.8
↳ reproduced	76.8	—	35.4	—	72.1	—
Sonata [63]	79.4	—	36.8	—	**76.0**	49.5‡
DVA [45]	71.0	—	—	—	67.2	—
BPNet‡ [19]	73.9	74.9	—	—	—	—
DMF-Net [65]	75.6	75.2	—	—	—	—
VMVF* [28]	76.4	74.6	—	—	—	—
ODIN* [22]	77.8	74.4	40.5	36.8	68.6	—
DITR	**80.5**	**79.7**	**41.2**	**44.9**	74.1	**52.5**

Table 1. **Indoor semantic segmentation results (mIoU).** DITR significantly outperforms PTv3 on all datasets and obtains state-of-the-art results on three of them. We compare against both 3D-only (top) and 2D–3D fusion methods (bottom). † Uses a smaller point patch size and relative positional encoding. ‡ Latest results from the challenge in combination with multi-dataset fine-tuning. * Requires 2D segmentation labels.

Method	nuScenes		Sem.KITTI		Waymo
	Val	Test	Val	Test	Val
MinkUNet [9]	73.3	—	63.8	—	65.9
SPVNAS [55]	77.4	—	64.7	66.4	—
Cylinder3D [73]	76.1	77.2	64.3	67.8	—
SphereFormer [30]	78.4	81.9	67.8	**74.8**	69.9
PTv2 [59]	80.2	82.6	70.3	72.6	70.6
PTv3 [61]	80.4	82.7	70.8	74.2	71.3
↳ reproduced	79.9	—	68.3	—	71.5
Sonata [63]	81.7	—	**72.6**	—	72.9
4D-Former [2]	78.9	80.4	66.3	—	—
2DPASS [64]	79.4	80.8	69.3	72.9	—
MSeg3D [31]	80.0	81.1	66.7	—	69.6
LCPS [70]	80.5	78.9	67.5	62.8	—
DITR	**84.2**	**85.1**	69.0	74.4	**73.3**

Table 2. **Outdoor semantic segmentation results (mIoU).** DITR obtains state-of-the-art results on datasets with full camera coverage: nuScenes and Waymo. We compare against both 3D-only (top) and 2D–3D fusion methods (bottom).

ment is particularly large on the hidden test set of ScanNet200, where we observe a gain of +7.1 mIoU. This shows that image feature injection is especially beneficial for more complex segmentation tasks with many categories. The only dataset where we observe only moderate gains is the Sem.KITTI dataset. However, this is to be expected because

Method	ScanNet	ScanNet200	S3DIS
PTv3 [61]	77.5	35.2	73.4†
↳ reproduced	76.8	35.4	72.1
D-DITR	78.6	37.2	—
D-DITR (multi-dataset)	**79.2**	**37.7**	**75.0**

Table 3. **Indoor distillation results (mIoU).** We evaluate models pretrained and fine-tuned on the same datasets, as well as one pretrained on ScanNet and Structured3D jointly. All models outperform the randomly initialized PTv3 baseline. † Uses a smaller point patch size and relative positional encoding.

Method	nuScenes	Sem.KITTI	Waymo
PTv3 [61]	80.4	70.8	71.3
↳ reproduced	79.9	68.3	71.5
D-DITR	**80.9**	—	71.6
D-DITR (multi-dataset)	80.7	**69.8**	**72.1**

Table 4. **Outdoor distillation results (mIoU).** We evaluate models pretrained and fine-tuned on the same datasets, as well as one pretrained on all three. In all cases, our models outperform the randomly initialized PTv3 baseline. Sem.KITTI is excluded from single-dataset pretraining due to its single-camera setup.

this dataset provides images from only a single front-facing camera, covering only a fraction of the 3D points, limiting the number of points for which 2D injection can be applied.

Compared to existing state-of-the-art methods, we note that DITR not only outperforms all previous 2D–3D fusion methods, but also the very recent Sonata [63] on most datasets. The improved performance compared to existing fusion methods is especially noteworthy because these methods only focus on either indoor or outdoor segmentation, while DITR is effective in both settings, and because some of them require 2D segmentation labels, which DITR does not. The fact that DITR surpasses Sonata, on the other hand, is striking because Sonata uses a PTv3 backbone that is three times larger than DITR's, and because it uses an expanded 3D training corpus. Interestingly, we observe in Tab. 6 that DITR still outperforms Sonata even when using the small DINOv2-S backbone, with DITR's model size—including a frozen DINOv2—amounting to only half of that of Sonata. This reinforces our main message: given the relatively limited availability of 3D data, 2D VFMs are crucial for advancing 3D segmentation, and even the inexpensive small variants can be highly effective.

Distillation. For distillation pretraining, we explore two settings: (1) pretraining on individual datasets and fine-tuning on the same dataset to show the effectiveness of VFM features as distillation targets, and (2) joint pretraining on multiple datasets. In the indoor multi-dataset distillation case, we jointly pretrain on ScanNet and Structured3D [72] and for the outdoor case we use nuScenes,

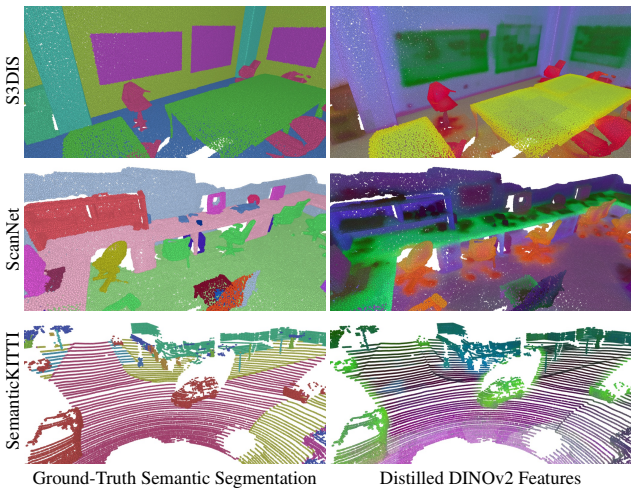

Figure 3. **D-DITR distillation.** We show PCA projected features from D-DITR after distillation. Many objects in the ground-truth segmentation are clearly separable in the predicted features, indicating the value of DINOv2 features for 3D segmentation. The colors are not expected to match, but color "clusters" should align.

SemanticKITTI, and Waymo. We exclude S3DIS from pretraining due to its smaller size and instead use it to assess generalization on an unseen dataset. The results are presented in Tab. 3 and Tab. 4 as D-DITR. We find that single-dataset distillation yields consistent improvements over the PTv3 baseline, demonstrating the effectiveness of DINOv2 distillation. The use of multi-dataset distillation further improves the results. In indoor datasets, distillation pretraining significantly improves the segmentation performance, with gains of +2.4 mIoU on ScanNet, +2.3 mIoU on ScanNet200 and +2.9 mIoU on S3DIS over the state-of-the-art 3D baseline. The improvement on S3DIS is particularly interesting, indicating that the distilled features can generalize to unseen datasets. For outdoor datasets, D-DITR also produces consistent improvements over the baseline. In particular, the distillation approach yields notable gains on SemanticKITTI, and even surpasses the *injected* DITR model (see Tab. 2). This indicates that even when images are not available, or compute constraints do not allow for the computation of DINOv2 features during inference, D-DITR can utilize the strong 2D features that it sees during pretraining for enhanced segmentation during inference.

Fig. 3 visualizes feature predictions by D-DITR prior to fine-tuning. When compared to the ground-truth segmentation labels, objects are clearly separable by color, even when visualizing only the first three principal components of the high-dimensional feature space. This further indicates the value of 2D VFM features for 3D semantic segmentation.

4.4. Ablation Study

2D–3D Injection Method. As described in Sec. 3.1, DITR injects 2D image features into all blocks of the 3D decoder. In Tab. 5, we compare this approach to several alternatives, including the 2D–3D injection methods that are employed by existing state-of-the-art 2D–3D fusion methods for indoor 3D segmentation that do not require 2D labels [23, 45, 65]. The results show that *early* injection—*i.e.*, before the 3D encoder—as employed by MVPNet [23] and DVA [45], is better than *no* injection, but that it underperforms DITR's *intermediate* injection in all decoder blocks. The same applies to DMF-Net's approach [65] of having separate 3D encoder–decoders for feature extraction and 3D segmentation, and injecting features between these two encoder–decoders. Finally, we also evaluate *late* injection methods, which either inject 2D features only in the last decoder block or after the last decoder block, but we find that these also perform worse than DITR's default injection. These results highlight the importance of informing the 3D model about 2D features at multiple stages of the network, as done by DITR's 2D–3D injection approach.

Injection method	ScanNet200	nuScenes
None (*i.e.*, PTv3 [61])	35.2	79.9
2D image feature injection		
Before 3D encoder [23, 45]	40.1	82.3
Between two 3D encoder–decoders [65]	40.4	82.2
In 3D decoder (all blocks) (*i.e.*, **DITR**)	**41.2**	**83.1**
In 3D decoder (last block only)	40.1	82.5
After 3D decoder	37.6	82.5
Other		
`[cls]` tokens in 3D decoder (all blocks)	37.4	79.4

Table 5. **Comparison of different 2D–3D injection methods.** We compare different methods to inject 2D features, ordered from *early* to *late* injection. All experiments use DINOv2-L.

In addition to the injection of 2D image features, we also experiment with injecting the `[cls]` tokens, generated by the 2D VFM, as additional tokens into the self-attention operations of the 3D decoder. This is inspired by the observation in Tab. 8 that DITR even achieves significant improvements on invisible points, suggesting that it might leverage additional features as a kind of global context, which could be contained by the `[cls]` token as well. However, we find that injecting the `[cls]` tokens only yields a small boost on indoor scenarios and does not impact outdoor performance. Therefore, we did not further pursue this direction.

Image Backbone Ablation. To assess the impact of the image backbone used to extract 2D features, we experiment with various commonly used image backbones with frozen weights as shown in Tab. 6. When comparing different DINOv2-pretrained ViT models, there is a consistent trend that larger models perform better. Comparing DINOv2 with other pretraining approach, we find that a standard ViT-L model pretrained on IN21k [11, 52] shows a modest performance improvement in indoor scenarios compared to the

Image backbone		ScanNet200	nuScenes
Pretraining	Model		
—	—	35.2	80.4
IN21k [52]	ViT-L	38.2	80.2
AIMv2 [14]	ViT-L	39.1	82.8
SigLIP 2 [57]	ViT-g	38.1	83.6
DINOv2 [35]	ViT-S	38.2	82.8
DINOv2 [35]	ViT-B	40.7	83.0
DINOv2 [35]	ViT-L	41.2	83.1
DINOv2 [35]	ViT-g	40.8	**84.2**
DINOv3 [51]	ViT-L	**42.3**	83.9

Table 6. **Injection with different image backbones.** We compare different image backbones and pretraining schemes for DITR.

% Data	Scratch	CSC [18]	MSC [60]	PPT [62]	D-DITR
1 %	26.0	28.9$^{\uparrow 2.9}$	29.2$^{\uparrow 3.2}$	31.3$^{\uparrow 5.3}$	**34.1**$^{\uparrow 8.1}$
5 %	47.8	49.8$^{\uparrow 2.0}$	50.7$^{\uparrow 2.9}$	52.2$^{\uparrow 4.4}$	**56.6**$^{\uparrow 8.8}$
10 %	56.7	59.4$^{\uparrow 2.7}$	61.0$^{\uparrow 4.3}$	62.8$^{\uparrow 6.1}$	**65.2**$^{\uparrow 8.5}$
20 %	62.9	64.6$^{\uparrow 1.7}$	64.9$^{\uparrow 2.0}$	66.4$^{\uparrow 3.5}$	**68.3**$^{\uparrow 5.4}$
100 %	72.2	73.8$^{\uparrow 1.6}$	75.3$^{\uparrow 3.1}$	75.8$^{\uparrow 3.6}$	**76.2**$^{\uparrow 4.0}$

Table 7. **ScanNet "limited reconstructions" benchmark with the MinkUNet [9] 3D backbone.** All methods perform pretraining on unannotated raw data, followed by segmentation fine-tuning on fixed labeled subsets of ScanNet. For fair comparison, all methods use a MinkUNet backbone (not PTv3).

no-injection baseline, but reduces performance on the outdoor nuScenes dataset. SigLIP 2 [57] and AIMv2 [14], on the other hand, consistently outperform the baseline in both indoor and outdoor settings, but perform worse than DINOv2. Finally, DINOv3 [51], which was released only days ago, performs even better than DINOv2 with ViT-L, and achieves the absolute best ScanNet200 performance. These results show that strong pretraining, as used in foundation models, is key to achieving consistent gains across diverse environments. They also suggest that our conclusions are not specific to DINOv2 but hold with very recent, more powerful VFMs like DINOv3 as well, and that these models are simply drop-in replacements in our framework.

4.5. Additional Analyses

ScanNet Data Efficient Benchmark. To demonstrate that D-DITR is also effective in label-scarce settings, we use the ScanNet "limited reconstructions" benchmark and show results for different percentages of available data during fine-tuning in Tab. 7. For this setting, we use MinkUNet [9] as the student model to compare with previous unsupervised approaches in a consistent setting. Similar to CSC [18], we use additional unlabeled images. D-DITR outperforms all previous unsupervised pretraining approaches in all settings. The gap is even larger when a small percentage of data is available during fine-tuning. Overall, this experiment shows that our distillation approach can be applied to other commonly used 3D backbones and that it provides a strong supervision signal compared to raw 3D data.

Method	ScanNet200		SemanticKITTI	
	Visible	Invisible	Visible	Invisible
PTv3 [61]	35.2	34.5	68.5	68.1
DITR	**41.5**	**39.7**	**71.6**	**68.3**

Table 8. **Performance on visible and invisible points.** We compare the segmentation performance of points visible in at least one image to those that are never visible. For SemanticKITTI, only 18.9 % of labeled points are visible.

Visibility vs. Performance. To better understand the source of the performance improvement, in Tab. 8, we compare the performance of DITR and the baseline on visible and invisible points separately. DITR shows a notable improvement on visible points for both the ScanNet200 and SemanticKITTI datasets. Moreover, it still outperforms the PTv3 baseline even on invisible points. The large improvement on ScanNet200 suggests that the model captures global context from the 2D features, aiding the segmentation of invisible regions.

Resource Usage. We assess the runtime impact of DITR, compared to the PTv3 baseline without additional injection of 2D features. On ScanNet200, using a ViT-L DINOv2 backbone and 10 camera views, training time increases from 12 to 15 hours using two H100 GPUs. The average per-scene inference latency increases from 41 to 76 ms, while the required GPU memory increases from 1.4 to 5.6 GiB (using a single H100 GPU for inference).

5. Conclusion

We demonstrate that the rich semantic features of 2D VFMs, like DINOv2, can be leveraged to advance 3D segmentation performance by large margins. First, with DITR, we demonstrate that injecting frozen 2D VFM features into a 3D model's decoder yields significant performance gains, achieving new state-of-the-art results. Second, we show that 2D VFMs can enable substantial improvements even when images are unavailable during inference, by using our D-DITR distillation strategy to pretrain a 3D backbone. Notably, both DITR and D-DITR require only unlabeled images and are not bound by the choice of VFM, making them readily and generally applicable. In conclusion, given our promising results, we strongly advocate the use of 2D VFMs for 3D scene understanding whenever possible.

Acknowledgements. K. Knaebel, K. Yilmaz and A. Hermans are funded by the project "Context Understanding for Autonomous Systems" by Robert Bosch GmbH. Compute resources were granted by RWTH under projects `rwth1604` and `rwth1730`.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *CVPR*, 2016. 1, 2, 3, 5

[2] Ali Athar, Enxu Li, Sergio Casas, and Raquel Urtasun. 4D-Former: Multimodal 4D Panoptic Segmentation. In *CoRL*, 2023. 2, 3, 6

[3] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. SceneScript: Reconstructing Scenes With An Autoregressive Structured Language Model. In *ECCV*, 2024. 2

[4] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes - a diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *NeurIPS*, 2021. 2

[5] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, 2019. 1, 2, 3, 5

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *CVPR*, 2020. 1, 2, 3, 5

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, 2021. 1

[8] Haoming Chen, Zhizhong Zhang, Yanyun Qu, Ruixin Zhang, Xin Tan, and Yuan Xie. Building a Strong Pre-Training Baseline for Universal 3D Large-Scale Perception. In *CVPR*, 2024. 3

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR*, 2019. 1, 2, 4, 6, 8

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017. 1, 2, 3, 5

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 7

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 2, 4

[13] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A Visual Representation for Neon Genesis. *Image and Vision Computing*, 149: 105171, 2024. 1

[14] Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor Guilherme Turrisi da Costa, Louis Béthune, Zhe Gan, et al. Multimodal Autoregressive Pre-training of Large Vision Encoders. In *CVPR*, 2025. 8

[15] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *CVPR*, 2018. 2

[16] Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. 5

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[18] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring Data-Efficient 3D Scene Understanding with Contrastive Scene Contexts. In *CVPR*, 2021. 8

[19] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional Projection Network for Cross Dimension Scene Understanding. In *CVPR*, 2021. 3, 6

[20] Rui Huang, Songyou Peng, Ayca Takmaz, Federico Tombari, Marc Pollefeys, Shiji Song, Gao Huang, and Francis Engelmann. Segment3D: Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels. *ECCV*, 2024. 3

[21] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 2015. 5

[22] Ayush Jain, Pushkal Katara, Nikolaos Gkanatsios, Adam W Harley, Gabriel Sarch, Kriti Aggarwal, Vishrav Chaudhary, and Katerina Fragkiadaki. ODIN: A Single Model for 2D and 3D Segmentation. In *CVPR*, 2024. 2, 3, 6

[23] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view PointNet for 3D Scene Understanding. In *CVPR Workshops*, 2019. 3, 7

[24] Tommie Kerssies, Daan de Geus, and Gijs Dubbelman. How to Benchmark Vision Foundation Models for Semantic Segmentation? In *CVPR Workshops*, 2024. 1

[25] Tommie Kerssies, Niccolò Cavagnero, Alexander Hermans, Narges Norouzi, Giuseppe Averta, Bastian Leibe, Gijs Dubbelman, and Daan de Geus. Your ViT is Secretly an Image Segmentation Model. In *CVPR*, 2025.

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *ICCV*, 2023. 1, 3

[27] Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. OneFormer3D: One Transformer for Unified Point Cloud Segmentation. In *CVPR*, 2024. 2

[28] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual Multi-view Fusion for 3D Semantic Segmentation. In *ECCV*, 2020. 3, 6

[29] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified Transformer for 3D Point Cloud Segmentation. In *CVPR*, 2022. 2, 6

[30] Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical Transformer for LiDAR-Based 3D Recognition. In *CVPR*, 2023. 6

[31] Jiale Li, Hang Dai, Hao Han, and Yong Ding. MSeg3D: Multi-modal 3D Semantic Segmentation for Autonomous Driving. In *CVPR*, 2023. 3, 6

[32] Youquan Liu, Lingdong Kong, Jun CEN, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment Any Point Cloud Sequences by Distilling Vision Foundation Models. In *NeurIPS*, 2023. 3

[33] Anas Mahmoud, Jordan S. K. Hu, Tianshu Kuai, Ali Harakeh, Liam Paull, and Steven L. Waslander. Self-Supervised Image-to-Point Distillation via Semantically Tolerant Contrastive Loss. In *CVPR*, 2023. 3

[34] Anas Mahmoud, Ali Harakeh, and Steven Waslander. Image-to-Lidar Relational Distillation for Autonomous Driving Data. In *ECCV*, 2024. 3

[35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, 2024. 1, 2, 3, 4, 8

[36] Aljosa Osep, Tim Meinhardt, Francesco Ferroni, Neehar Peri, Deva Ramanan, and Laura Leal-Taixé. Better Call SAL: Towards Learning to Segment Anything in Lidar. In *ECCV*, 2024. 3

[37] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Pointconv: Deep Convolutional Networks on 3D Point Clouds. In *CVPR*, 2019. 2

[38] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D Scene Understanding with Open Vocabularies. In *CVPR*, 2023. 3, 5

[39] Gilles Puy, Spyros Gidaris, Alexandre Boulch, Oriane Siméoni, Corentin Sautier, Patrick Pérez, Andrei Bursuc, and Renaud Marlet. Three Pillars Improving Vision Foundation Model Distillation for Lidar. In *CVPR*, 2024. 3

[40] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, 2017. 2

[41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 2017. 2

[42] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In *NeurIPS*, 2022. 6

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 1, 3

[44] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *NeurIPS*, 2021. 2

[45] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning Multi-View Aggregation In the Wild for Large-Scale 3D Semantic Segmentation. In *CVPR*, 2022. 2, 3, 6, 7

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *CVPR*, 2022. 1

[47] David Rozenberszki, Or Litany, and Angela Dai. Language-Grounded Indoor 3D Semantic Segmentation in the Wild. In *ECCV*, 2022. 1, 2, 5

[48] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-Lidar Self-Supervised Distillation for Autonomous Driving Data. In *CVPR*, 2022. 3

[49] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. *arXiv preprint arXiv:2111.02114*, 2021. 1

[50] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An Open Large-Scale Dataset for Training Next Generation Image-Text Models. *NeurIPS*, 2022. 1

[51] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. 2, 8

[52] Andreas Peter Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers. *TMLR*, 2022. 7, 8

[53] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *CVPR*, 2020. 1, 2, 3, 5

[54] Mingkui Tan, Zhuangwei Zhuang, Sitao Chen, Rong Li, Kui Jia, Qicheng Wang, and Yuanqing Li. EPMF: Efficient perception-aware multi-sensor fusion for 3D semantic segmentation. *IEEE TPAMI*, 2024. 2, 3

[55] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *ECCV*, 2020. 6

[56] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *CVPR*, 2019. 2

[57] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil

Mustafa, et al. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*, 2025. 5, 8

[58] Peng-Shuai Wang. OctFormer: Octree-based Transformers for 3D Point Clouds. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023. 2, 6

[59] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point Transformer V2: Grouped Vector Attention and Partition-based Pooling. In *NeurIPS*, 2022. 1, 2, 6

[60] Xiaoyang Wu, Xin Wen, Xihui Liu, and Hengshuang Zhao. Masked Scene Contrast: A Scalable Framework for Unsupervised 3D Representation Learning. In *CVPR*, 2023. 8

[61] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler, Faster, Stronger. In *CVPR*, 2024. 1, 2, 4, 5, 6, 7, 8

[62] Xiaoyang Wu, Zhuotao Tian, Xin Wen, Bohao Peng, Xihui Liu, Kaicheng Yu, and Hengshuang Zhao. Towards Large-scale 3D Representation Learning with Multi-dataset Point Prompt Training. In *CVPR*, 2024. 2, 5, 8

[63] Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-Supervised Learning of Reliable Point Representations. In *CVPR*, 2025. 1, 2, 6

[64] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2DPASS: 2D Priors Assisted Semantic Segmentation on LiDAR Point Clouds. In *ECCV*, 2022. 3, 6

[65] Chaolong Yang, Yuyao Yan, Weiguang Zhao, Jianan Ye, Xi Yang, Amir Hussain, Bin Dong, and Kaizhu Huang. Towards Deeper and Better Multi-view Feature Fusion for 3D Semantic Segmentation. In *International Conference on Neural Information Processing*, 2023. 6, 7

[66] Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3D: A Pretrained Transformer Backbone for 3D Indoor Scene Understanding. *arXiv preprint arXiv:2304.06906*, 2023. 2, 6

[67] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 2, 5

[68] Kadir Yilmaz, Jonas Schult, Alexey Nekrasov, and Bastian Leibe. Mask4Former: Mask Transformer for 4D Panoptic Segmentation. In *ICRA*, 2024. 2

[69] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *ICCV*, 2023. 1

[70] Zhiwei Zhang, Zhizhong Zhang, Qian Yu, Ran Yi, Yuan Xie, and Lizhuang Ma. LiDAR-Camera Panoptic Segmentation via Geometry-Consistent and Semantic-Aware Alignment. In *ICCV*, 2023. 2, 3, 6

[71] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point Transformer. In *ICCV*, 2021. 2, 6

[72] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A Large Photo-realistic Dataset for Structured 3D Modeling. In *ECCV*, 2020. 1, 2, 6

[73] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation. In *CVPR*, 2021. 4, 6

[74] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-Aware Multi-Sensor Fusion for 3D LiDAR Semantic Segmentation. In *ICCV*, 2021. 3