UniDet3D: Multi-dataset Indoor 3D Object Detection

Maksim Kolodiazhnyi[1], **Anna Vorontsova**[2], **Matvey Skripkin**[1],
Danila Rukhovich[3], **Anton Konushin**[1]

[1] Artificial Intelligence Research Institute, Moscow, Russia
[2] NEURA Robotics GmbH, Metzingen, Germany
[3] University of Luxembourg, Luxembourg
kolodiazhnyi@airi.net, anna.vorontsova@neura-robotics.com, skripkin@airi.net, danila.rukhovich@uni.lu, konushin@airi.net

Abstract

Growing customer demand for smart solutions in robotics and augmented reality has attracted considerable attention to 3D object detection from point clouds. Yet, existing indoor datasets taken individually are too small and insufficiently diverse to train a powerful and general 3D object detection model. In the meantime, more general approaches utilizing foundation models are still inferior in quality to those based on supervised training for a specific task. In this work, we propose UniDet3D, a simple yet effective 3D object detection model, which is trained on a mixture of indoor datasets and is capable of working in various indoor environments. By unifying different label spaces, UniDet3D enables learning a strong representation across multiple datasets through a supervised joint training scheme. The proposed network architecture is built upon a vanilla transformer encoder, making it easy to run, customize and extend the prediction pipeline for practical use. Extensive experiments demonstrate that UniDet3D obtains significant gains over existing 3D object detection methods in 6 indoor benchmarks: ScanNet (+1.1 mAP_{50}), S3DIS (+9.1 mAP_{50}), ARKitScenes (+19.4 mAP_{25}), MultiScan (+9.3 mAP_{50}), 3RScan (+3.2 mAP_{50}), and ScanNet++ (+2.7 mAP_{50}).

1 Introduction

3D object detection from point clouds aims at simultaneous localization and recognition of 3D objects given a 3D point set. As a core technique for 3D scene understanding, it is widely applied in robotics, AR, and 3D scanning.

Due to major variations in scale and visual appearance of indoor scenes, complemented with different selections and placement of objects, indoor 3D data tends to be complex and diverse. Besides, captured by various sensors ranging from Kinect to generic smartphone cameras, indoor data is inconsistent regarding point cloud density and scene coverage. This leads to a domain gap between different datasets.

Indoor benchmarks contain at most thousands of scenes, e.g., the popular ScanNet (Dai et al. 2017) has 1513 scenes, a more recent ARKitScenes (Baruch et al. 2021) has 5042 scenes, while S3DIS (Armeni et al. 2016), ScanNet++ (Yeshwanth et al. 2023) are the order of magnitude smaller. None of the datasets contains data of sufficient diversity and volume to train a general model which can be transferred between datasets without severe loss of quality.

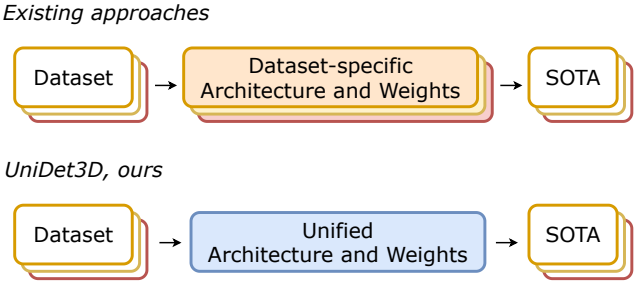

Figure 1: Existing 3D object detection methods use different architectures and weights to achieve state-of-the-art metrics on different datasets. We propose UniDet3D trained single time on a mixture of datasets and achieving even better results.

Applying a 3D scene understanding model outside the single training domain is possible to a certain extent with visual-language models, that encapsulate the fundamental knowledge about the world via establishing relations between imagery and textual data.

However, visual-language models imply open-vocabulary problem formulation rather than limiting label spaces to predefined categories. In 3D scene understanding, visual-language models are used to precompute 2D image features, which are lifted to 3D space. Still, these 2D-to-3D approaches in 3D instance segmentation (Nguyen et al. 2024; Takmaz et al. 2024) and 3D object detection (Lu et al. 2023; Zhang et al. 2024) are inferior to supervised baselines (Schult et al. 2023; Misra, Girdhar, and Joulin 2021). In the meantime, the size of existing real-world indoor datasets is currently insufficient for training visual-language models that can provide high-quality 3D features directly (Jia et al. 2024).

State-of-the-art 3D object detection accuracy in indoor benchmarks is achieved via classic supervised training on categorical labels. Not only do they struggle to generalize to new visually distinct scenes and unseen objects – but handling novel categories remains an unresolved issue.

In 2D object detection, training using data from other

sources rather than the target domain is a fruitful direction being actively investigated. The most common and basic way is pretraining with diverse and voluminous out-of-domain data, followed by fine-tuning using in-domain data. Another paradigm implies training jointly on a mixture of in-domain and out-of-domain data. Similarly, we address the generalization of 3D object detection across domains represented in indoor 3D datasets.

Creating a multi-dataset 3D object detection method can be decomposed into four sub-tasks.

First, the **network architecture** should be carefully designed so that handling data from different sources should not impose a major computational overhead. We claim a novel detection architecture as one of our major contributions. Currently, the best scores on each indoor benchmark are achieved by dataset-specific methods: sparse convolutional TR3D (on S3DIS) (Rukhovich, Vorontsova, and Konushin 2023), transformer-based V-DETR (on ScanNet) (Shen et al. 2023). On the contrary, we design a unified approach based on a pure self-attention encoder architecture without positional encoding and cross-attention.

Second, **training datasets** representing different domains should be properly chosen and mixed. We mix up long-lasting and well-known ScanNet (Dai et al. 2017), S3DIS (Armeni et al. 2016) and ARKitScenes (Baruch et al. 2021), and enrich them with smaller-scale MultiScan (Mao et al. 2022), 3RScan (Wald et al. 2019), and ScanNet++ (Yeshwanth et al. 2023).

Third, output data should be transformed into a label space shared across multiple datasets. To this end, we explore different ways of **merging category labels** across datasets with inconsistent annotations.

Finally, the **multi-dataset training procedure** should be set up for robust performance in all domains – rather than compromising the quality in most cases in favor of certain specific scenarios. To identify the best design choices, we experiment with several training strategies and show that joint training ensures higher scores on test splits of all datasets in the mixture.

2 Related Work

2.1 3D Detection Architectures

Existing 3D object detection methods can be categorized into voting-based, expansion-based, and transformer-based. The proposed UniDet3D falls into the latter group, still we briefly overview both voting-based and expansion-based methods, since we include them in our quantitative comparison.

Voting-based methods (Qi et al. 2019; Chen et al. 2020; Engelmann et al. 2020; Xie et al. 2020; Cheng et al. 2021; Zhang et al. 2020; Xie et al. 2021; Wang et al. 2022b; Zheng et al. 2022; Zhu et al. 2024) pioneered the field, with VoteNet (Qi et al. 2019) being the first method that introduced point voting for 3D object detection. Subsequent methods mainly follow the line of extending VoteNet with additional modules and tricks to improve detection quality. The latest work in this row, SPGroup3D (Zhu et al. 2024), exploits superpoint clustering, which has already proved itself to be beneficial for 3D instance segmentation (Kolodiazhnyi et al. 2024). Yet, using superpoints is not limited to voting-based approaches, and we also cluster an input point cloud into superpoints in our transformer-based UniDet3D.

Expansion-based methods (Gwak, Choy, and Savarese 2020; Rukhovich, Vorontsova, and Konushin 2022, 2023; Wang et al. 2022a) generate virtual center features from surface features using a generative sparse decoder, and predict high-quality 3D region proposals. GSDN (Gwak, Choy, and Savarese 2020) adapts fully convolutional architecture for 3D object detection. FCAF3D (Rukhovich, Vorontsova, and Konushin 2022) proposes anchor-free proposal generation, while TR3D (Rukhovich, Vorontsova, and Konushin 2023) achieves real-time inference with a lightweight generative decoder. CAGroup3D (Wang et al. 2022a) improves the results of FCAF3D by running the second refinement stage.

Transformer-based methods (Misra, Girdhar, and Joulin 2021; Liu et al. 2021; Wang et al. 2024; Shen et al. 2023) dominate 3D object detection. Following the seminal Group-Free (Liu et al. 2021) work, they first extract point cloud features with a sparse-convolutional backbone and then predict objects from input queries with a transformer decoder through cross-attending to the backbone features. V-DETR (Shen et al. 2023) upgrades Group-Free with a vertex relative positional encoding. Uni3DETR (Wang et al. 2024) extends over indoor and outdoor datasets. 3DETR (Misra, Girdhar, and Joulin 2021) replaces the backbone with a transformer encoder, making the entire network transformer-based. Overall, a decent part of progress in transformer-based 3D object detection is attributed to sophisticated architectures, elaborated positional encoding, and non-trivial interaction between modules. Besides, existing methods use computationally extensive Hungarian matching to assign predicted bounding boxes to ground truth ones during the training.

On the contrary, we use a simple self-attention encoder architecture without positional encoding and cross-attention that are typically needed in the decoder part. We also replace Hungarian matching with a lightweight effective alternative. By designing UniDet3D model, we follow a plug-and-play paradigm, so that each component can be easily replaced and tailored to user limitations and requirements.

2.2 Multi-dataset Object Detection

Most existing object detection methods are trained on a single dataset, so that both the volume of data and semantics diversity are limited. Recently, training object detection on multiple datasets (Cai et al. 2022; Shi et al. 2021; Zhao et al. 2020; Zhou, Koltun, and Krähenbühl 2022) has proved to boost the model quality, generalization ability, and robustness in the 2D domain. Different strategies of joining input sources and heterogeneous label spaces have been proposed so far, e.g., recent works (Meng et al. 2023; Wang et al. 2023) leverage large language models to handle an open set of categories via representing them using natural language. Several attempts have been made to address multi-dataset training of 3D object detection in out-

door scenarios (Zhang et al. 2023; Soum-Fontez, Deschaud, and Goulette 2023), either in LIDAR point clouds or monocular images (Brazil et al. 2023; Li et al. 2024). We argue that outdoor-targeted approaches taking benefits from large-scale annotated datasets cannot be straightforwardly adapted to handle orders of magnitude smaller collections of indoor data. In this paper, we investigate multi-dataset training of 3D object detection in the indoor domain.

3 Multi-dataset 3D Detection Training

3D object detection aims to predict a location $b_i \in \mathbb{R}^7$ and a class-wise detection score $p_i \in \mathbb{R}^{|L|}$ for each object i in a point cloud P. The detection score denotes confidence for a bounding box to belong to an object with label $c \in L$, where L is the set of all classes (label space) of a dataset $\mathcal{D}$.

Many 3D object detection methods are trained and tested using the ScanNet dataset (Dai et al. 2017), which contains balanced annotations for 18 common object classes; making training relatively simple. Training on ScanNet usually implies straightforwardly minimizing a loss ℓ, e.g. a bounding box-level log-likelihood, over a sampled point cloud $\hat{P}$ and annotated 3D bounding boxes $\hat{B}$ from the dataset $\mathcal{D}$:

$$\min_{\Theta} \mathbb{E}_{(\hat{P},\hat{B})\sim\mathcal{D}} \left[\ell(\mathcal{M}(\hat{P};\Theta), \hat{B}) \right]. \quad (1)$$

Here, $\hat{B}$ contains class-specific box annotations. The loss ℓ is defined on two sets of bounding boxes, predicted and ground truth ones, being matched based on the overlap criterion.

Let us now consider training a 3D object detection model on K datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots$, having label spaces $L_1, L_2, \ldots$, respectively. The naive solution is to train a *separate* model M_k on a dataset D_k using a dataset-specific loss ℓ_k:

$$\min_{\Theta} \mathbb{E}_{(\hat{P},\hat{B})\sim\mathcal{D}_k} \left[\ell_k(\mathcal{M}_k(\hat{P};\Theta), \hat{B}) \right]. \quad (2)$$

However, training a single common model instead of several dataset-specific models can boost the performance for all datasets. In 2D object detection, this is achieved using a *partitioned* label space (Zhou, Koltun, and Krähenbühl 2022),

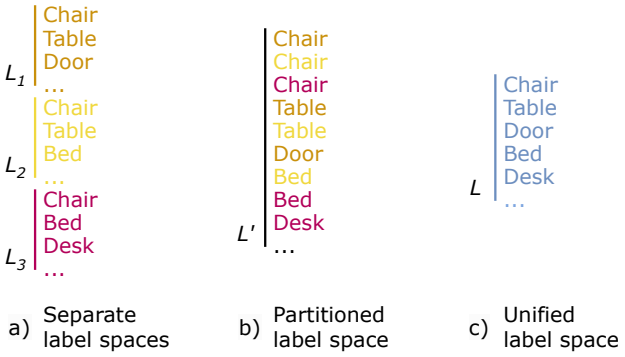

Figure 2: Three common ways of handling heterogeneous label spaces for training. The partitioned scheme implies using a separate classification head for each dataset. UniDet3D follows the unified scheme, using the same de-duplicated set of labels during both the training and inference.

which is equivalent to training K models $\mathcal{M}_1, \ldots, \mathcal{M}_K$ with the same architecture $\mathcal{M}$ but the last dataset-specific classification layer. The common model is trained by minimizing the K dataset-specific losses:

$$\min_{\Theta} \mathbb{E}_{\mathcal{D}_k} \left[\mathbb{E}_{(\hat{P},\hat{B})\sim\mathcal{D}_k} \left[\ell_k(\mathcal{M}_k(\hat{P};\Theta), \hat{B}) \right] \right]. \quad (3)$$

Still, when using the partitioned label space, probabilities of classes from different datasets are estimated regardless of the similarity of these classes, e.g. probabilities for a chair category in the ScanNet and ARKitScenes datasets are predicted separately and independently (see Fig). While this may allow for better per-dataset scores, it complicates the interpretation of results for the end user, since per-dataset predictions should be somehow aggregated. This observation naturally leads to the *unified* scheme, which is combining all labels of all datasets into a dataset $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \ldots$, and uniting the label spaces $L = L_1 \cup L_2 \cup \ldots$. Similar labels get merged, making the common label space unambiguous. The optimization procedure remains the same as for default training on a single dataset:

$$\min_{\Theta} \mathbb{E}_{(\hat{P},\hat{B})\sim\mathcal{D}_1\cup\mathcal{D}_2\cup\ldots} \left[\ell(\mathcal{M}(\hat{P};\Theta), \hat{B}) \right]. \quad (4)$$

In an empirical study below, we show that the unified scheme supersedes the partitioned one not only regarding simplicity, interpretability, and fewer training parameters but also delivers higher accuracy.

4 3D Detection Network

The overall scheme of UniDet3D is depicted in Fig. Given a point cloud, a sparse 3D U-Net network extracts point-wise features. In parallel, superpoints are obtained through unsupervised clustering. Then, point features are aggregated within each superpoint by simple averaging (or superpoint pooling), giving superpoint features. Superpoint features are passed as queries to a vanilla transformer encoder. The encoder outputs are processed with two separate MLPs, one estimating regression parameters of objects' bounding boxes, and another predicting class probabilities in multi-dataset shared label space.

4.1 Backbone and Pooling

3D U-Net. Assuming that an input point cloud contains N points, the input can be formulated as $P \in \mathbb{R}^{N\times 6}$. Each 3D point is parameterized with three colors *r*, *g*, *b*, and three coordinates *x*, *y*, *z*. Following (Choy, Gwak, and Savarese 2019), we voxelize the point cloud and use a U-Net-like backbone composed of sparse 3D convolutions to extract point-wise features $P' \in \mathbb{R}^{N\times C}$.

Superpoint pooling. To build an end-to-end framework, we directly feed point-wise features $P' \in \mathbb{R}^{N\times C}$ into superpoint pooling layer based on pre-computed superpoints (Landrieu and Simonovsky 2018). The superpoint pooling layer obtains superpoint features $S \in \mathbb{R}^{M\times C}$ via average pooling over those point-wise ones inside each superpoint. Without loss of generality, we suppose that there are M superpoints computed from the input point cloud. Notably, the superpoint pooling layer downsamples the input

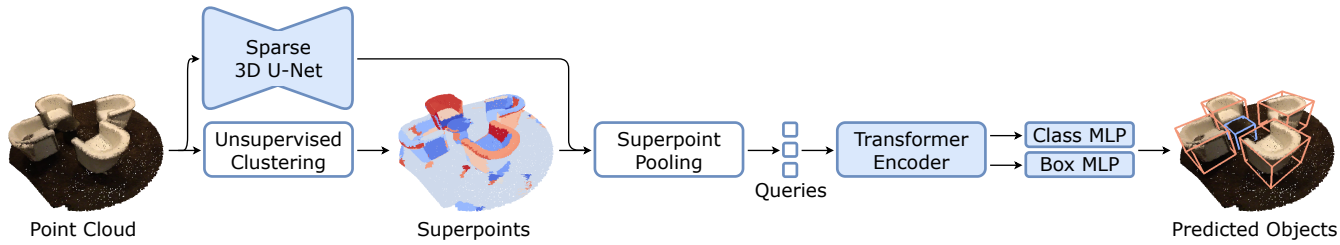

Figure 3: Overview of the proposed method. UniDet3D takes the point cloud as an input, and extracts point features using a sparse 3D U-Net network. Point features are averaged across superpoints in the superpoint pooling. Aggregated features serve as input queries to a vanilla transformer encoder. Finally, 3D bounding boxes are derived from encoder outputs with a box MLP and class MLP, where box MLP estimates the location of a 3D bounding box w.r.t. the mass center of the superpoint, and class MLP outputs probabilities of object classes in the unified label space.

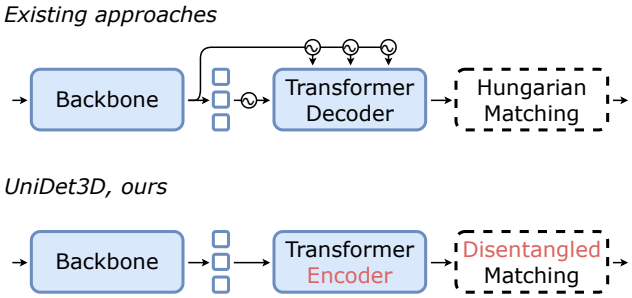

Figure 4: Comparison with existing transformed-based 3D object detection methods. We introduce encoder-only transformer architecture w/o positional encoding for queries or attention layers. This allows us to change unstable Hungarian matching for a simpler disentangled scheme.

point cloud to hundreds of superpoints, which significantly reduces the computational overhead of subsequent processing and optimizes the representation capability of the entire network.

4.2 Transformer Encoder

Backbone features after superpoint pooling are processed with a transformer encoder network. This network takes M queries as an input and outputs M object proposal features.

Existing transformer-based 3D object detection methods (Misra, Girdhar, and Joulin 2021; Liu et al. 2021; Wang et al. 2024; Shen et al. 2023) exploit different techniques to establish connections between queries and points in an input point cloud. In particular, queries are initialized via the farthest point sampling. Then, positional encoding is added to preserve spatial information, and queries are cross-attended to points with positional guidance in a *decoder* part of the network. On the contrary, we use a vanilla transformer *encoder* without bells and whistles (Fig. 4).

Overall, we employ a simple, elegant transformer architecture based on self-attention between input queries solely. Our experiments show that positional encoding is redundant, bringing negligible to no accuracy improvement at the cost of extra computations and more complex design.

4.3 Head

The head takes M object proposals as inputs and produces a single 3D bounding box and a class label for each proposal, estimated via linear layers. The classification layer (*class MLP* in the Fig. 3) outputs probabilities of $|L|$ object classes. The parameters of 3D bounding boxes are regressed with a *box MLP*. Each 3D bounding box is represented in the form of an 8-value vector (Rukhovich, Vorontsova, and Konushin 2022), where the first six values denote positive distances from the proposal coordinate to the faces of a predicted 3D bounding box, and the last two values define a rotation angle. The distances to the faces are given w.r.t. the mass center of the superpoint, making it the only positional information used to predict 3D bounding boxes.

4.4 Training

To train a transformer-based method end-to-end, we need to define a cost function between queries and ground truth objects, develop a matching strategy that minimizes this cost function, and formulate a loss function being applied to the matched pairs.

Cost function. We use a pairwise matching cost $\mathcal{C}_{ik}$ to measure the similarity of the i-th proposal and the k-th ground truth. $\mathcal{C}_{ik}$ is derived from a classification probability and predicted bounding box:

$$\mathcal{C}_{ik} = -\lambda \cdot p_{i,c_k} + DIoU(b_i, \hat{b}_k), \tag{5}$$

where p_{i,c_k} indicates the probability of i-th proposal belonging to the c_k semantic category. b_i and $\hat{b}_k$ stands for predicted and ground truth bounding boxes. The distance between ground truth and predicted boxes is measured with the DIoU function as in TR3D (Rukhovich, Vorontsova, and Konushin 2023). In our experiments, we use $\lambda = 0.25$.

Matching. Previous transformer-based 3D object detection methods (Misra, Girdhar, and Joulin 2021; Liu et al. 2021; Wang et al. 2024; Shen et al. 2023) use bipartite matching based on a Hungarian algorithm (Kuhn 1955). This common approach has a major drawback: an excessive number of meaningful matches between proposals and ground truth objects makes the training process long-lasting

and unstable. To overcome this issue, we adapt the disentangled matching scheme introduced in recent 3D instance segmentation work (Kolodiazhnyi et al. 2024).

Since an object query is initialized with superpoint features, this object query can be unambiguously matched with this superpoint. We assume that a superpoint can belong only to one object, which gives a correspondence between a superpoint, an object query, an object proposal derived from this object query, and a ground truth object.

We assign each object with three nearest superpoints, so, to get a bipartite matching, we only need to filter out extra superpoints matched to the same object. This task reformulation simplifies cost function optimization, as we can set the most weights in a cost matrix to infinity:

$$\hat{\mathcal{C}}_{ik} = \begin{cases} \mathcal{C}_{ik} & \text{if } i\text{-th superpoint} \in k\text{-th object} \\ +\infty & \text{otherwise} \end{cases} \tag{6}$$

Loss. After matching proposals with ground truth instances, instance losses can finally be calculated. Classification errors are penalized with a cross-entropy loss $\mathcal{L}_{cls}$. Besides, for each match between a proposal and a ground truth object, we compute the regression loss $\mathcal{L}_{reg}$ as a DIoU function between predicted and ground truth boxes.

The total loss $\mathcal{L}$ is formulated as:

$$\mathcal{L} = \beta \cdot \mathcal{L}_{cls} + \mathcal{L}_{reg} \tag{7}$$

where $\beta = 0.5$.

5 Experiments

5.1 Datasets

We evaluate our method on six real-world indoor benchmarks: ScanNet (Dai et al. 2017), ARKitScenes (Baruch et al. 2021), S3DIS (Armeni et al. 2016), MultiScan (Mao et al. 2022), 3RScan (Wald et al. 2019), ScanNet++ (Yeshwanth et al. 2023). For methodological purity, we do not add single-view RGB-D datasets such as SUN RGB-D (Song, Lichtenberg, and Xiao 2015) to the mixture but only use datasets containing multi-view 3D reconstructions. In the absence of ground truth 3D bounding boxes, we calculate axis-aligned bounding boxes from 3D instance labels through a standard approach (Qi et al. 2019).

Dataset	# Train. Scenes	# Val. Scenes	# Classes
ScanNet	1201	312	18
ARKitScenes	4493	549	17
S3DIS	204	68	5
MultiScan	174	42	17
3RScan	385	47	18
ScanNet++	230	50	84
Overall	6687	1068	99

Table 1: Quantitative statistics of indoor datasets in our mixture. ScanNet and ARKitScenes are relatively large-scale, while S3DIS, MultiScan, 3RScan, and ScanNet++ are times smaller.

ScanNet (Dai et al. 2017) contains 1513 reconstructed 3D indoor scans with per-point instance and semantic labels of 18 categories. The training subset consists of 1201 scans, while 312 scans are used for validation.

ARKitScenes (Baruch et al. 2021) consists of 5042 scans of 1661 venues captured using a tablet with an online ARKit tracking system. This is the only dataset in the list labeled with oriented bounding boxes. We use the official training and validation splits of 4493 and 549 scans, respectively.

Stanford Large-Scale 3D Indoor Spaces (S3DIS) (Armeni et al. 2016) contains scans of 272 rooms, annotated with instance and semantic labels of five furniture categories. We use the official *Area 5* split, where 68 rooms serve for validation, and 204 rooms comprise the training subset.

MultiScan (Mao et al. 2022) is a small yet extensively labeled RGB-D dataset of 273 scans of 117 indoor scenes with 11K objects, primarily intended for part mobility estimation. It contains per-frame camera poses, textured 3D surface meshes, and fine object-level semantic labels.

3RScan (Wald et al. 2019) is designed as a benchmark for temporal visual analysis, e.g., change detection or visual localization. It features 1482 3D reconstructions of 478 scenes alongside calibrated RGB-D sequences, textured 3D meshes and instance and semantic annotations.

ScanNet++ (Yeshwanth et al. 2023) is an instance segmentation and novel-view synthesis benchmark. It contains 450 RGB-D sequences recorded with an iPhone and 3D scans captured using a laser scanner with sub-millimeter resolution and annotated with long-tail semantics.

5.2 Evaluation

For all datasets, we use mean average precision (mAP) under IoU thresholds of 0.25 and 0.5 as a metric.

We upper-limit the number of points in an input point cloud by $N = 100000$ points, as proposed in (Rukhovich, Vorontsova, and Konushin 2022, 2023). Since these points are sampled randomly, both training and evaluation procedures are randomized. To obtain statistically significant results, we run training 5 times and test each trained model 5 times independently. We report the best and average metrics across 5×5 trials: this allows comparing UniDet3D to the 3D object detection methods that report either a single best or an average value.

5.3 Implementation Details

We implement UniDet3D in the mmdetection3d (Contributors 2020) framework. All training details are the same as in OneFormer3D (Kolodiazhnyi et al. 2024), particularly, we use AdamW optimizer with an initial learning rate of 0.0001, weight decay of 0.05, batch size of 8, and polynomial scheduler with a base of 0.9 for 1024 epochs. We apply the standard augmentations: horizontal flipping, random rotations around the z-axis, and random scaling. During the training, we assign a ground truth object to the three nearest superpoints. Since during the inference we seek for one-to-one matching, we suppress redundant superpoints using

Method	Venue	ScanNet		ARKitScenes		S3DIS		MultiScan		3RScan		ScanNet++	
		mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}
Best result													
VoteNet	ICCV'19	58.6	33.5	35.8									
HGNet	CVPR'20	61.3	34.4										
GSDN	ECCV'20	62.8	34.8			47.8	25.1						
3D-MPA	CVPR'20	64.2	49.2										
MLCVNet	CVPR'20	64.5	41.4	41.9									
3DETR	ICCV'21	65.0	47.0										
BRNet	CVPR'21	66.1	50.9										
H3DNet	ECCV'20	67.2	48.1	38.3									
VENet	ICCV'21	67.7											
Group-Free	ICCV'21	69.1	52.8										
RBGNet	CVPR'22	70.6	55.2										
HyperDet3D	CVPR'22	70.9	57.2										
FCAF3D	ECCV'22	71.5	57.3			66.7	45.9	53.8	40.7	60.1	42.6	22.3	11.4
Uni3DETR	NIPS'23	71.7	58.3			70.1	48.0						
TR3D	ICIP'23	72.9	59.3			74.5	51.7	56.7	42.3	62.3	45.4	26.2	14.5
SPGroup3D	AAAI'24	74.3	59.6			69.2	47.2						
CAGroup3D	NIPS'22	75.1	61.3										
V-DETR	ICLR'24	77.4	65.0										
UniDet3D		**77.5**	**66.1**	**61.3**	**47.1**	**75.2**	**60.8**	**64.2**	**51.6**	**64.7**	**48.6**	**26.4**	**17.2**
Average across 25 trials													
Group-Free	ICCV'21	68.6	51.8										
RBGNet	CVPR'22	69.9	54.7										
FCAF3D	ECCV'22	70.7	56.0			64.9	43.8	52.5	39.2	59.6	40.4	21.4	11.0
TR3D	ICIP'23	72.0	57.4			72.1	47.6	55.0	41.2	61.5	44.2	24.3	13.9
SPGroup3D	AAAI'24	73.5	58.3			67.7	43.6						
CAGroup3D	NIPS'22	74.5	60.3										
V-DETR	ICLR'24	76.8	64.5										
UniDet3D		**77.1**	**65.2**	**60.2**	**46.0**	**73.3**	**57.9**	**62.4**	**50.8**	**62.1**	**45.6**	**24.4**	**16.3**

Table 2: Comparison of the detection methods on 6 datasets: ScanNet, S3DIS, ARKitScenes, MultiScan, 3RScan, and ScanNet++. Our UniDet3D trained jointly on 6 datasets sets the new state-of-the-art in all benchmarks. Results obtained by running existing methods on the novel datasets are marked gray.

Label Space	ScanNet		ARKitScenes		S3DIS		MultiScan		3RScan		ScanNet++	
	mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}	mAP_{25}	mAP_{50}
from scratch												
separate	77.0	65.0	59.6	45.7	57.2	39.7	46.1	33.1	45.1	31.4	21.6	12.2
joint training												
partitioned	77.0	65.1	59.8	45.8	71.2	56.2	62.0	50.5	**62.6**	45.4	24.1	16.0
unified	**77.1**	**65.2**	**60.2**	**46.0**	**73.3**	**57.9**	**62.4**	**50.8**	62.1	**45.6**	**24.4**	**16.3**

Table 3: Scores (average across 25 trials) obtained using different label spaces. Expectedly, joint training is especially beneficial for small datasets. Switching from the partitioned (159 classes) to unified (99 classes) label space not only increases interpretability for an end user but also has a positive effect on overall accuracy, which is a valuable practical outcome.

NMS. No test-time augmentations are applied during the inference time. All experiments are conducted using a single NVidia V100.

5.4 Comparison to Prior Work

We compare UniDet3D against various 3D object detection methods. According to the Tab. 2, UniDet3D consistently outperforms the competitors not only in the *best* but also in the *average* scores, which indicates the statistical significance of results. In the well-known ScanNet and S3DIS benchmarks, UniDet3D sets state-of-art results, superseding second-best scores by at least +1 mAP_{50} on ScanNet and impressive +9.1 mAP_{50} on S3DIS. To obtain reference values for smaller datasets MultiScan, 3RScan, and ScanNet++, we train and evaluate FCAF3D (Rukhovich, Vorontsova, and Konushin 2022) and TR3D (Rukhovich, Vorontsova, and

Scan-Net	ARKit-Scenes	S3DIS mAP_{25}	S3DIS mAP_{50}	MultiScan mAP_{25}	MultiScan mAP_{50}	3RScan mAP_{25}	3RScan mAP_{50}
from scratch							
		57.2	39.7	46.1	33.1	45.1	31.4
fine-tuning							
✓	✓	71.3	54.3	60.2	49.1	60.8	45.6
joint training							
✓		72.0	55.3	59.0	46.2	59.8	42.0
	✓	65.5	48.3	46.5	34.7	55.4	40.5
✓	✓	**73.3**	**57.9**	**62.4**	**50.8**	**62.1**	**45.6**

Table 4: Scores (average across 25 trials) obtained on the S3DIS, MultiScan, and 3RScan test splits after either training from scratch on the train splits on the respective datasets, using pre-training, or joint training. The joint training on the mixture of larger ScanNet and ARKitScenes datasets is the most beneficial.

Konushin 2023), two strong baselines with publicly available code. The observed improvement over these methods is especially tangible for MultiScan, where the gain is +7.5 mAP_{25} and +9.3 mAP_{50}.

Method	PE	HM	mAP_{25}	mAP_{50}	Inference time, ms
3DETR	✓	✓	65.0	47.0	170
Group-Free	✓	✓	69.1	52.8	157
Uni3DETR	✓	✓	71.7	58.3	283
V-DETR	✓	✓	77.4	65.0	240
	✓		77.4	66.0	233
UniDet3D		✓	75.2	64.5	224
			77.5	**66.1**	224

Table 5: Comparison of transformed-based methods on the ScanNet validation split, all trained on the ScanNet training split solely. PE is positional encoding, HM is Hungarian matching (applied only during the training). UniDet3D without PE and HM hits the highest scores.

5.5 Ablation Studies

Training schemes. To emulate real usage, we consider small S3DIS, MultiScan, and 3RScan as target datasets, and leverage large ScanNet and ARKitScenes as sources of extra training data. In Tab. 4, we compare three training schemes:

- training *from scratch* on target dataset;
- *fine-tuning* after pre-training on a mixture of ScanNet and ARKitScenes;
- *joint training* on a mixture of ScanNet and/or ARKitScenes and the target dataset.

While transformer-based approaches dominate on large-scale datasets, they cannot train sufficiently on limited data – which is the case when using extra data and more elaborate training schemes appears to be the most profitable. Respectively, UniDet3D easily outperforms both transformer and non-transformer methods on the large ScanNet, but is notably inferior to convolutional baselines TR3D and FCAF3D, when trained *from scratch* on S3DIS, MultiScan, or 3RScan.

After simple *fine-tuning*, our model surpasses baseline approaches, which evidences our unified architecture to effectively adapt to target domains after learning general concepts from the voluminous mixture of training datasets. This result is valuable for practitioners seeking a customizable approach that can be trained quickly under limited computational powers. *Joint training* adds extra +3.6 and +1.7 mAP_{50} on S3DIS and MultiScan, respectively. Expectedly, the amount and variety of training data also matter: using both ScanNet and ARKitScenes ensures higher accuracy than using them solely.

Merging different label spaces. The benefits of joint training are fully revealed for small datasets, and so is the difference between partitioned and unified label spaces. According to Tab 3, unifying label space improves the overall quality over the partitioned label space and brings +1.7 mAP_{50} on S3DIS. Taking the better interpretability of unified classes and the smaller size of the classification layer (99 unified classes against 159 in the partitioned label space), we can claim the unified label space as a preferred option. Not only is this an interesting experimental finding, but a useful feature for real-world applications.

Positional encoding and Hungarian matching. In this study, we measure the effect of positional encoding and matching strategy on the model's performance.

To match randomly initialized queries and point cloud features, transformer-based methods use positional encoding and cross-attention. Since our superpoint-induced query initialization strategy preserves spatial information, the need for adding positional encoding is questionable. Apart from that, UniDet3D's query initialization allows employing *disentangled matching* instead of costly *Hungarian matching*.

To ensure competitive comparison, we implement vertex relative positional encoding proposed in the previous state-of-the-art V-DETR (Shen et al. 2023). As seen in Tab. 5, UniDet3D trained without positional encoding and Hungarian matching achieves the highest detection accuracy on ScanNet among transformer-based methods. In the meantime, eliminating positional encoding reduces time- and memory- footprint, so overall we can claim both transformer-specific parts to be redundant.

6 Conclusion

In this work, we proposed UniDet3D, a 3D object detection model trained on a mixture of indoor datasets. By unifying label spaces across datasets in the supervised joint training scheme, UniDet3D generalizes to various indoor environments. The network architecture of the proposed method is built upon a vanilla transformer encoder, making the entire pipeline easy to use and adapt to user requirements. Extensive experiments prove UniDet3D to deliver state-of-the-art results in 6 indoor benchmarks: ScanNet, S3DIS, ARKitScenes, MultiScan, 3RScan, and ScanNet++.

References

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1534–1543.

Baruch, G.; Chen, Z.; Dehghan, A.; Dimry, T.; Feigin, Y.; Fu, P.; Gebauer, T.; Joffe, B.; Kurz, D.; Schwartz, A.; et al. 2021. ARKitScenes–A Diverse Real-World Dataset For 3D Indoor Scene Understanding Using Mobile RGB-D Data. *arXiv preprint arXiv:2111.08897*.

Brazil, G.; Kumar, A.; Straub, J.; Ravi, N.; Johnson, J.; and Gkioxari, G. 2023. Omni3d: A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13154–13164.

Cai, L.; Zhang, Z.; Zhu, Y.; Zhang, L.; Li, M.; and Xue, X. 2022. Bigdetection: A large-scale benchmark for improved object detector pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4777–4787.

Chen, J.; Lei, B.; Song, Q.; Ying, H.; Chen, D. Z.; and Wu, J. 2020. A hierarchical graph network for 3D object detection on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 392–401.

Cheng, B.; Sheng, L.; Shi, S.; Yang, M.; and Xu, D. 2021. Back-tracing Representative Points for Voting-based 3D Object Detection in Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8963–8972.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3075–3084.

Contributors, M. 2020. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5828–5839.

Engelmann, F.; Bokeloh, M.; Fathi, A.; Leibe, B.; and Nießner, M. 2020. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9031–9040.

Gwak, J.; Choy, C.; and Savarese, S. 2020. Generative sparse detection networks for 3d single-shot object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, 297–313. Springer.

Jia, B.; Chen, Y.; Yu, H.; Wang, Y.; Niu, X.; Liu, T.; Li, Q.; and Huang, S. 2024. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. *arXiv preprint arXiv:2401.09340*.

Kolodiazhnyi, M.; Vorontsova, A.; Konushin, A.; and Rukhovich, D. 2024. Oneformer3d: One transformer for unified point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20943–20953.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.

Landrieu, L.; and Simonovsky, M. 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4558–4567.

Li, Z.; Xu, X.; Lim, S.; and Zhao, H. 2024. UniMODE: Unified Monocular 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16561–16570.

Liu, Z.; Zhang, Z.; Cao, Y.; Hu, H.; and Tong, X. 2021. Group-Free 3D Object Detection via Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2949–2958.

Lu, Y.; Xu, C.; Wei, X.; Xie, X.; Tomizuka, M.; Keutzer, K.; and Zhang, S. 2023. Open-vocabulary point-cloud object detection without 3d annotation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1190–1199.

Mao, Y.; Zhang, Y.; Jiang, H.; Chang, A. X.; and Savva, M. 2022. MultiScan: Scalable RGBD scanning for 3D environments with articulated objects. In *Advances in Neural Information Processing Systems*.

Meng, L.; Dai, X.; Chen, Y.; Zhang, P.; Chen, D.; Liu, M.; Wang, J.; Wu, Z.; Yuan, L.; and Jiang, Y.-G. 2023. Detection hub: Unifying object detection datasets via query adaptation on language embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11402–11411.

Misra, I.; Girdhar, R.; and Joulin, A. 2021. An End-to-End Transformer Model for 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2906–2917.

Nguyen, P.; Ngo, T. D.; Kalogerakis, E.; Gan, C.; Tran, A.; Pham, C.; and Nguyen, K. 2024. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4018–4028.

Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, 9277–9286.

Rukhovich, D.; Vorontsova, A.; and Konushin, A. 2022. FCAF3D: fully convolutional anchor-free 3D object detection. In *IEEE/CVF European Conference on Computer Vision (ECCV)*, 477–493. Springer.

Rukhovich, D.; Vorontsova, A.; and Konushin, A. 2023. Tr3d: Towards real-time indoor 3d object detection. In *2023 IEEE International Conference on Image Processing (ICIP)*, 281–285. IEEE.

Schult, J.; Engelmann, F.; Hermans, A.; Litany, O.; Tang, S.; and Leibe, B. 2023. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 8216–8223. IEEE.

Shen, Y.; Geng, Z.; Yuan, Y.; Lin, Y.; Liu, Z.; Wang, C.; Hu, H.; Zheng, N.; and Guo, B. 2023. V-detr: Detr with vertex relative position encoding for 3d object detection. *arXiv preprint arXiv:2308.04409*.

Shi, B.; Zhang, X.; Xu, H.; Dai, W.; Zou, J.; Xiong, H.; and Tian, Q. 2021. Multi-dataset pretraining: A unified model for semantic segmentation. *arXiv preprint arXiv:2106.04121*.

Song, S.; Lichtenberg, S. P.; and Xiao, J. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 567–576.

Soum-Fontez, L.; Deschaud, J.-E.; and Goulette, F. 2023. Mdt3d: Multi-dataset training for lidar 3d object detection generalization. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5765–5772. IEEE.

Takmaz, A.; Fedele, E.; Sumner, R.; Pollefeys, M.; Tombari, F.; and Engelmann, F. 2024. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. *Advances in Neural Information Processing Systems*, 36.

Wald, J.; Avatisyan, A.; Navab, N.; Tombari, F.; and Niessner, M. 2019. RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

Wang, H.; Ding, L.; Dong, S.; Shi, S.; Li, A.; Li, J.; Li, Z.; and Wang, L. 2022a. Cagroup3d: Class-aware grouping for 3d object detection on point clouds. *Advances in Neural Information Processing Systems*, 35: 29975–29988.

Wang, H.; Shi, S.; Yang, Z.; Fang, R.; Qian, Q.; Li, H.; Schiele, B.; and Wang, L. 2022b. RBGNet: Ray-Based Grouping for 3D Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1110–1119.

Wang, Z.; Li, Y.; Chen, X.; Lim, S.-N.; Torralba, A.; Zhao, H.; and Wang, S. 2023. Detecting everything in the open world: Towards universal object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11433–11443.

Wang, Z.; Li, Y.-L.; Chen, X.; Zhao, H.; and Wang, S. 2024. Uni3detr: Unified 3d detection transformer. *Advances in Neural Information Processing Systems*, 36.

Xie, Q.; Lai, Y.-K.; Wu, J.; Wang, Z.; Lu, D.; Wei, M.; and Wang, J. 2021. VENet: Voting Enhancement Network for 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3712–3721.

Xie, Q.; Lai, Y.-K.; Wu, J.; Wang, Z.; Zhang, Y.; Xu, K.; and Wang, J. 2020. Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10447–10456.

Yeshwanth, C.; Liu, Y.-C.; Nießner, M.; and Dai, A. 2023. ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Zhang, B.; Yuan, J.; Shi, B.; Chen, T.; Li, Y.; and Qiao, Y. 2023. Uni3d: A unified baseline for multi-dataset 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9253–9262.

Zhang, D.; Li, C.; Zhang, R.; Xie, S.; Xue, W.; Xie, X.; and Zhang, S. 2024. FM-OV3D: Foundation Model-Based Cross-Modal Knowledge Blending for Open-Vocabulary 3D Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16723–16731.

Zhang, Z.; Sun, B.; Yang, H.; and Huang, Q. 2020. H3dnet: 3d object detection using hybrid geometric primitives. In *European Conference on Computer Vision*, 311–329. Springer.

Zhao, X.; Schulter, S.; Sharma, G.; Tsai, Y.-H.; Chandraker, M.; and Wu, Y. 2020. Object detection with a unified label space from multiple datasets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, 178–193. Springer.

Zheng, Y.; Duan, Y.; Lu, J.; Zhou, J.; and Tian, Q. 2022. HyperDet3D: Learning a Scene-conditioned 3D Object Detector. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5575–5584.

Zhou, X.; Koltun, V.; and Krähenbühl, P. 2022. Simple multi-dataset detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7571–7580.

Zhu, Y.; Hui, L.; Shen, Y.; and Xie, J. 2024. SPGroup3D: Superpoint Grouping Network for Indoor 3D Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7811–7819.