2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

TUN3D: Towards Real-World Scene Understanding from Unposed Images

Anton Konushin[1, 2], **Nikita Drozdov**[1], **Bulat Gabdullin**[2, 4], **Alexey Zakharov**[1], **Anna Vorontsova**, **Danila Rukhovich**[3], **Maksim Kolodiazhnyi**[1, 2†]
[1]Lomonosov Moscow State University; [2]AXXX, Moscow, Russia; [3]Institute of Mechanics, Armenia; [4]Higher School of Economics

Abstract— **Layout estimation and 3D object detection are two fundamental tasks in indoor scene understanding. When combined, they enable the creation of a compact yet semantically rich spatial representation of a scene. Existing approaches typically rely on point cloud input, which poses a major limitation since most consumer cameras lack depth sensors and visual-only data remains far more common. We address this issue with TUN3D, the first method that tackles joint layout estimation and 3D object detection in real scans, given multi-view images as input, and does not require ground-truth camera poses or depth supervision. Our approach builds on a lightweight sparse-convolutional backbone and employs two dedicated heads: one for 3D object detection and one for layout estimation, leveraging a novel and effective parametric wall representation. Extensive experiments show that TUN3D achieves state-of-the-art performance across three challenging scene understanding benchmarks: (i) using ground-truth point clouds, (ii) using posed images, and (iii) using unposed images. While performing on par with specialized 3D object detection methods, TUN3D significantly advances layout estimation, setting a new benchmark in holistic indoor scene understanding.**

I. INTRODUCTION

Indoor scene understanding is a long-standing computer vision task, with applications in robotics, AR/VR, interior design, real estate and property inspection. Many real-world scenarios require only a compact yet informative structural description of the scene: the room layout (walls, floor, ceiling) and the locations and categories of major objects. Being significantly lighter than dense 3D reconstruction, this representation is especially beneficial for applications running on a device.

Recent approaches address 3D object detection [1], [2], [3], [4], [5], [6], [7], [8], layout estimation [9], [10], or even explore joint prediction of layout and objects from point clouds [11], [12], [13]. Arguably, a joint model could benefit from shared reasoning about the structure and semantics of a scene, but existing methods are either extremely slow [12], [13], or still far behind single-task methods in accuracy [11]. On the contrary, we build our model on top of a real-time 3D object detection model, achieving state-of-the-art performance with impressive latency.

Input data is another critical issue in the 3D scene understanding. Using point clouds at inference time requires either depth sensors and/or accurate multi-view reconstruction. This limits the applicability of point cloud-based methods in scenarios where only video is available, e.g., on customer devices that are equipped with neither depth sensors nor trackers, or for processing prerecorded videos. In this paper, we investigate input data modalities in the 3D scene reconstruction context and move beyond point clouds towards images with camera poses and even unposed images. As a result of this study, we present TUN3D, the first method of layout estimation and 3D object detection from real-world images without access to depth or camera poses.

Our contribution is as follows:

- We show that data requirements in 3D scene understanding can be relaxed from point clouds to multi-view images – with and even without camera poses;
- We propose a compact yet expressive layout parameterization and a new architecture for joint layout estimation and 3D object detection;
- We establish a new state-of-the-art in scene understanding across three challenging scenarios: from ground truth point clouds, posed multi-view images, and unposed multi-view images.

II. RELATED WORK

A. Scene Understanding from Point Clouds

Point cloud-based 3D object detection methods can be categorized into voting-based, transformer-based, and sparse convolutional. Voting-based methods, from the seminal VoteNet [14] to recent SPGroup3D [6], follow a bottom-up paradigm, grouping processed points into object candidates. Transformer-based methods, including the first-in-class GroupFree3D [1] and the most recent UniDet3D [8], use a transformer encoder to predict a set of objects. Methods based on sparse 3D convolutions, such as GSDN [15], FCAF3D [2] and its follow-up TR3D [3], balance speed, accuracy, and scale well to larger scenes, so we develop our TUN3D following this paradigm.

Layout estimation is another long-lasting scene understanding task that has been addressed solely by such methods as Omni-PQ [10], RoomFormer [9], or coupled with 3D object detection in SceneCAD [16] and PQ-Transformer [11].

Recent emergence of large language models boosted the scene understanding research. SceneScript [12] trains a transformer model to generate procedural scene descriptions in

Code is available at `https://github.com/col14m/tun3d`
†Corresponding author: kolodyazhniyma@my.msu.ru

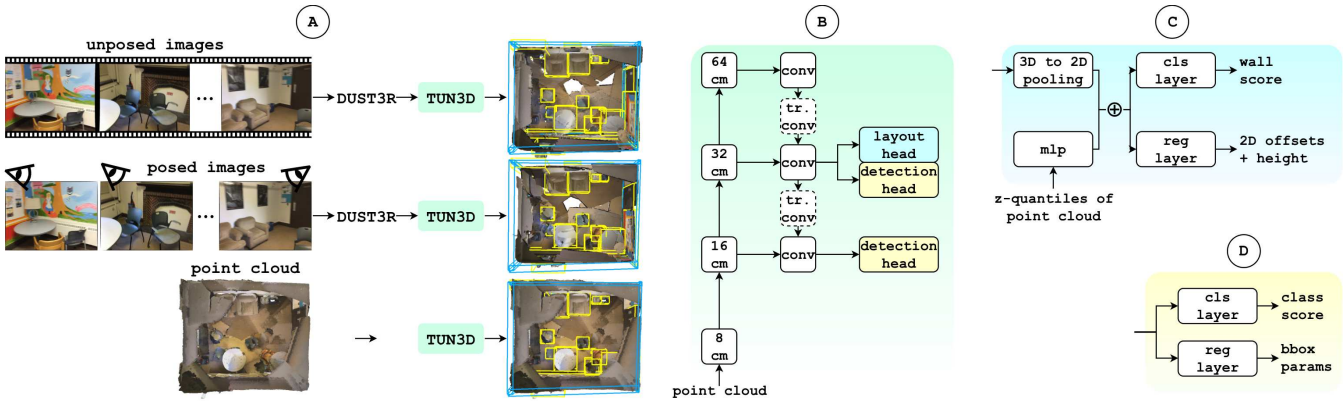


Fig. 1. (A) TUN3D can flexibly process various inputs: unposed images, posed images, and point clouds. (B) TUN3D model is constructed of a 3D sparse-convolutional backbone and neck, followed by two task-specific heads. (C) The novel layout head predicts wall scores and regresses wall parameters for each wall comprising the layout. (D) The detection head outputs object class scores and coordinates of a 3D bounding box of an object.

a language created specifically for this task, while SpatialLM [13] outputs Python code for scene generation. Both SceneScript and SpatialLM are trained with synthetic data solely, and no layout estimation results are shown on real scans.

B. Scene Understanding from Posed Images

Numerous methods can process visual information rather than explicit scene geometry but gain such an ability by utilizing depth for training. ImGeoNet [17] learns geometry from multi-view images with depth supervision. 3DGeoDet [18] uses depth to estimate voxel occupancy, while NeRF-based methods, including NeRF-DetS [5], NeRF-Det++ [7] and GO-N3RDet [19], implement various ways of multi-view feature fusion with depth guidance.

Depth-free 3D object detection from posed images is an emerging research topic, launched with the publication of ImVoxelNet [20] that first tackled 3D object detection from multi-view inputs in an end-to-end manner. Later, NeRF-Det [4] exploite NeRF's ability to infer 3D geometry from sole visual inputs for depth-free 3D object detection. MVSDet [21] utilizes plane sweep for geometry-aware 3D object detection, applying probabilistic sampling and a soft weighting mechanism for feature lifting in the absence of explicit depth. SceneScript [12] also provides a version capable of handling posed images, but neither layout estimation nor 3D object detection is reported on real-world images.

C. Scene Understanding from Unposed Images

Single-view layout estimation [22] and 3D object detection [20] approaches are flexible but limited in scene coverage, making the predictions hardly usable in real-world applications. Panoramic images provide a more complete view of the scene and are widely used for layout estimation [23], [24], [25]. However, they are inherently limited to a single viewpoint, which restricts object coverage and often leads to occlusion issues.

Other scene understanding tasks, such as 3D visual grounding, 3D dense captioning, and 3D question answering, are now being solved with LLMs with visual capabilities that take images as inputs ([26], [27], [28], [29], [30]). Point cloud-based SpatialLM [13] is combined with Mast3R [31] for video processing, and this combination is proved to successfully handle synthetic data. To the best of our knowledge, there are no prior methods of either layout estimation or 3D object detection taking multi-view unposed images as inputs and working with real-world data. In this paper, we aim to close this gap with TUN3D.

III. Scene Understanding from Point Cloud

A. Problem Formulation

In TUN3D, we formulate scene understanding as proposed in PQ-Transformer [11], and predict layouts and detect 3D objects jointly with a single model, given a colored point cloud as an input. More formally, TUN3D model awaits a point cloud $P = \{p_i\}_{i=1}^{N} \subset \mathbb{R}^6$, where each point $p_i = (x_i, y_i, z_i, r_i, g_i, b_i)$ is described with its coordinates in the 3D space and RGB color.

Detected 3D objects are parameterized as $\mathcal{O} = \{(b_k, c_k)\}_{k=1}^{K}$, where $c_k \in \{1, \ldots, C\}$ denotes object categories and b_k stands for the spatial parameters of a 3D bounding box. $b_k = (t_k, s_k)$, where $t_k \in \mathbb{R}^3$ is the center of a 3D bounding box and $s_k \in \mathbb{R}^3_+$ are sizes along the x, y, z-axes.

Layout is defined as a set of walls $\mathcal{W} = \{w_\ell\}_{\ell=1}^{L}$, where each wall $w_\ell = (q_{\ell,1}, q_{\ell,2}, q_{\ell,3}, q_{\ell,4})$ is specified by 3D coordinates of its four corners $q_{\ell,j} \in \mathbb{R}^3$, ordered clockwise.

B. Network Architecture

Our fully-differentiable model is built of a backbone, neck, and two heads addressing 3D object detection and layout estimation, respectively (Fig. 1). Below, we discuss these components architecture-wise and describe techniques used during the training and inference phases.

Backbone is a 3D sparse high-dimensional version of ResNet, first introduced in GSDN [15] and used in FCAF3D [2]. We use an optimized version described in

TR3D [3]. First, an input point cloud is voxelized with a voxel size of 2 cm. Then, four residual blocks of sparse 3D convolutional layers transform the voxel space into 8 cm, 16 cm, 32 cm, and 64 cm-sized spatial grids. The maximum number of channels in all sparse convolutional layers is upper-limited to 128 for the sake of efficiency, since larger values compromise inference speed.

Neck aggregates 3D voxel features from four residual levels of the backbone. Features at each level are processed with one sparse generative transposed 3D convolution and one sparse 3D convolution. The well-known issue of standard convolutions is that they might downscale the visibility field too aggressively. To prevent loss of spatial information, we apply generative convolutional layers: they increase the number of voxels, hence expanding the covered area, so that object candidates beyond the current visibility field can still be processed. We use generative convolutions only at levels of 64 cm and 32 cm.

Detection head in TUN3D is similar to the one used in TR3D [3]. Specifically, it consists of two linear layers stacked sequentially. The weights are shared across 32 cm and 16 cm feature levels. The head returns a set of 3D locations $\hat{\mathcal{V}} = \{\hat{v}_j\}_{j=1}^{J}$, and for each location $\hat{v}_j \in \hat{\mathcal{V}}$, it predicts class logits $\tilde{z}_j \in \mathbb{R}^C$, offset of the center of an object $\Delta t_j \in \mathbb{R}^3$, and log-sizes of its 3D bounding box $\tilde{s}_j \in \mathbb{R}^3$. The canonical representation of the predicted 3D bounding box is derived as $t_j = \hat{v}_j + \Delta t_j$, $s_j = \exp(\tilde{s}_j) \in \mathbb{R}^3_+$, while the resulting class probabilities are calculated as $p_{jc} = \sigma(\tilde{z}_{jc})$.

Layout head takes features from $\hat{\mathcal{V}} = \{\hat{v}_j\}_{j=1}^{J}$ at the 32-cm level as inputs and produces a layout as a set of walls. For each wall, wall logit $\alpha_j \in \mathbb{R}$ is predicted with the classification layer, so that the final wall probability score is $p_j^{\text{wall}} = \sigma(\alpha_j)$. Wall parameters are estimated with a novel regression layer, discussed below.

C. Wall Parameterization

Since we tackle layout estimation in the formulation of PQ-Transformer [11], we use the same wall parameterization. However, in our experiments it appeared to be suboptimal, while other parameterizations delivered better quality. Below, we elaborate on possible alternatives, review existing approaches, and formalize novel ways to define a single wall.

PQ parameterization [11] defines a wall with eight parameters, namely, an offset from the location v_j to the center of the wall $\Delta t_j^{\text{wall}} \in \mathbb{R}^3$, wall length $\ell_j \in \mathbb{R}_+$, wall height $h_j \in \mathbb{R}_+$, and a normal to the wall plane $n_j \in \mathbb{R}^3$, $\|n_j\|_2 = 1$. Accordingly, the wall center in world coordinates is calculated as $c_j = \hat{v}_j + \Delta t_j^{\text{wall}}$, and the four corners $q_{j,1:4}$ are derived from c_j, ℓ_j, h_j, n_j trivially.

4×3D offsets Arguably the most straightforward way to define a wall is to simply specify its four corners. This non-parametric wall representation would contain 3D offsets to all corners, $\Delta q_j^{(k)} \in \mathbb{R}^3$, $k = 1, \dots, 4$, totaling 12 values. Respectively, the wall corners in world coordinates can be restored by summing the location position with offsets, i.e., $q_j^{(k)} = \hat{v}_j + \Delta q_j^{(k)}$.

2×3D offsets + height. Assuming that walls have constant height, we can reduce the number of parameters to just seven: in such a case, we can specify only two 3D offsets to the lower (floor-touching) corners of the wall $\Delta q_j^{(L,1)}, \Delta q_j^{(L,2)} \in \mathbb{R}^3$, and set a relative height $\Delta h_j \in \mathbb{R}_+$. Lower wall corners can then be derived as $q_j^{(L,m)} = \hat{v}_j + \Delta q_j^{(L,m)}$ for $m \in \{1, 2\}$. Upper wall corners are calculated as $q_j^{(U,m)} = q_j^{(L,m)} + \Delta h_j \, e_z$, given that e_z is a global up axis.

2×2D offsets + height (ours) In all three wall parameterizations above, predicted 3D offsets are not mutually constrained, which might result in a malformed scene geometry. But what if we enforce more rigidity by reducing the degrees of freedom even further – will it help to obtain cleaner geometry and make the model more robust and general? Taking insights from outdoor 3D object detection approaches [32], [20], we perform dimensional reduction and define the layout estimation task in the bird's-eye-view (BEV) plane instead of 3D space. The intuition behind this trick is the same as in the outdoor scenario: the cars cannot be stacked on top of each other – so neither can the walls.

The proposed dimensionality reduction is incorporated into the architecture of the layout head (Fig. 1, C). Particularly, 3D features yielded by the neck are first projected onto the floor plane via average pooling. As a result of the pooling procedure, important information about the height of walls is being lost, so later on we enrich those floor-projected features with missing spatial information. To this end, we calculate heights of all points in a scene, which are naturally their z coordinates, since the floor is zero-aligned, and estimate z-quantiles. z-quantiles are then encoded with a small MLP into a single vector encapsulating height distribution in the scene. This scene-level vector is concatenated to all projected features at predicted locations $\hat{u}_j := (\hat{x}_j, \hat{y}_j)$.

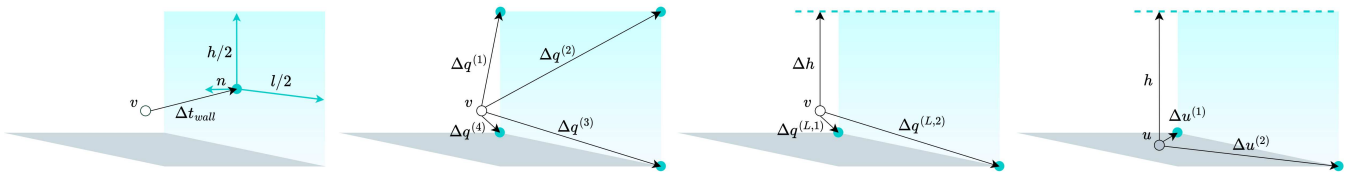
Eventually, the wall is encoded with five parameters: 2D offsets to the two lower wall corners $\Delta u_j^{(1)}, \Delta u_j^{(2)} \in \mathbb{R}^2$ and height $h_j \in \mathbb{R}_+$. The lower wall corners could be computed as $q_j^{(L,m)} = \big(\hat{u}_j + \Delta u_j^{(m)}, 0\big)$, and upper wall corners are $q_j^{(U,m)} = q_j^{(L,m)} + h_j \, e_z$, $m \in \{1, 2\}$.

D. Training

To calculate loss during the training, predicted objects should be assigned to ground truth objects. The matching rules do not need to be the same for both tasks being solved; actually, we apply different matching strategies for objects and walls.

Location-object assignment is performed to couple 3D locations $\{\hat{v}_j\}$ with ground truth objects $\mathcal{O}$. Following the matching strategy in [3], we pre-define the head level for each object category: typically large objects (e.g., bed or sofa) are processed at the third level (32 cm), and smaller ones (e.g., chair or nightstand) are handled at the second (16 cm). Within a feature level specified, each ground truth object is assigned six locations nearest to its center.

Location-wall assignment aims to establish correspondence between 3D locations $\{\hat{v}_j\}$ (or 2D locations $\{\hat{u}_j\}$) and ground truth walls $\mathcal{W}$. For all wall parameterizations, we



PQ [11] | 4×3D offsets | 2×3D offsets + height | **2×2D offsets + height, ours**

Fig. 2. Different wall parameterizations.

adopt the assignment strategy for "large" objects: assigning a wall to the six nearest 3D locations (or their 2D projections onto the floor plane) predicted at the 32-cm feature level. A similar mechanism is integrated for wall parameterizations based on both 3D offsets and 2D floor projections.

Loss is multi-component, where each component contributes to training a specific output layer in heads. Namely, classification in the 3D object detection head is guided with a focal loss, regression of 3D bounding box parameters is being trained with DIoU loss. To penalize erroneous layouts, the focal loss is applied to the outputs of the wall classification layer, and L1 loss is estimated for wall parameters. The overall training loss is thus calculated as follows:

$$\mathcal{L} = \mathcal{L}^{\text{det}}_{\text{focal}} + \mathcal{L}^{\text{det}}_{\text{DIoU}} + \mathcal{L}^{\text{layout}}_{\text{focal}} + \mathcal{L}^{\text{layout}}_{\text{L1}}$$

IV. TOWARDS SCENE UNDERSTANDING FROM UNPOSED IMAGES

In this section, we explore using fewer ground truth modalities for scene understanding during both training and inference. Relaxing the requirements for input data may open new possibilities for running scene understanding applications on customer devices that are not equipped with depth sensors or trackers – or process pure visual data collected with casual cameras. Respectively, besides point cloud being the most informative and resource-intensive modality, we experiment images with camera poses, and even unposed images as inputs.

A. Posed Images

In the pose-aware scenario, our method accepts a set of images $\{I_m\}_{m=1}^{M}$, $I_m \in \mathbb{R}^{H\times W\times 3}$ along with camera intrinsics $K_m \in \mathbb{R}^{3\times 3}$ and camera extrinsics $T_m \in SE(3)$. In real applications, camera poses may be sourced using either specialized hardware (inertial measurement units) or integrated software (visual trackers).

As discussed above, we are already able to process point clouds. The missing part of the puzzle is hence converting images with poses into a point cloud. To this end, we employ dense structure-from-motion methods, namely DUSt3R [33] which has certain advantages over similar approaches. First, it can operate not only in pose-agnostic, but also in pose-aware mode, accepting ground truth poses as auxiliary inputs. Respectively, it is applicable in both posed-image and unposed-image scenarios, making the entire framework flexible and laconic at the same time. Apart from that, using DUSt3R also allows preserving methodological purity of our experimental protocol and avoiding data leakage, since it was not trained on ScanNet, contrary to some competing methods (including its famous follow-up, Mast3R [31]).

Being a dense SfM method, DUSt3R estimates dense depth maps for given frames. Then, the original images and those estimated depths are fused using ground truth camera poses into a TSDF volume. Finally, a point cloud is extracted, – the rest steps of the solution match the steps taken in the point cloud-based scenario.

B. Unposed Images

In the third scenario, our model is challenged to process a sole image collection $\{I_m\}_{m=1}^{M}$ without known camera intrinsics K_m or extrinsics T_m. This formulation is relevant for data captured with most casual customer devices (e.g., smartphones or non-professional cameras), or prerecorded videos with missing capturing information.

DUSt3R reveals its full potential in this most challenging scenario, jointly predicting depth maps and camera parameters. Same as in the pose-aware case, those depth maps are then used for TSDF integration, but here we rely on estimated camera poses instead of ground truth ones.

V. EXPERIMENTS

A. Datasets

ScanNet [34] is a widely used real-world dataset with 1201 scans in the training subset and 312 in the validation part. Following [14], we calculate axis-aligned 3D bounding boxes from semantic per-point labels. SceneCAD [16] further extends ScanNet with 3D layouts that we use in our experiments.

ARKitScenes [35] is an RGB-D dataset containing 4493 training scans and 549 validation scans. The original dataset does not provide ground truth layouts. The validation subset was annotated in Omni-PQ [10], while the training part remains unlabeled; therefore, ARKitScenes is only used as a benchmark in cross-dataset experiments.

S3DIS [36] contains 272 scenes captured in six areas. Following the standard experimental protocol for object detection, we test on Area 5 and train on the rest areas, reporting detection accuracy for five semantic categories. We generate layout annotation by ourselves, calculating bounds of each wall instance.

Structured3D [37] is a large-scale synthetic dataset of 3.5K house designs created by professional designers along with ground truth 3D structure annotations and photo-realistic rendered images. The authors of [9] enriched Structured3D with structural elements, namely walls, windows, and doors,

TABLE I

RESULTS OF LAYOUT ESTIMATION AND OBJECT DETECTION FROM VARIOUS INPUT MODALITIES ON SCANNET AND S3DIS.

Method	Venue	Depth		ScanNet			S3DIS		
		Train	Test	Layout F1	Detection mAP@0.25	Detection mAP@0.5	Layout F1	Detection mAP@0.25	Detection mAP@0.5
GT point clouds									
GSDN [15]	ECCV'20	✓	✓	-	62.8	34.8	-	47.8	25.1
TR3D [3]	ICIP'22	✓	✓	-	72.0	58.1	-	72.5	57.2
SPGroup3D [6]	AAAI'24	✓	✓	-	74.3	59.6	-	69.2	47.2
UniDet3D [8]	AAAI'25	✓	✓	-	**77.0**	**65.0**	-	73.2	57.4
SceneCAD [16]	ECCV'20	✓	✓	37.9	-	-	-	-	-
Omni-PQ [10]	ICRA'23	✓	✓	60.8	-	-	-	-	-
PQ [11]	ICRA'22	✓	✓	54.4	60.9	39.9	29.6	61.1	38.0
TUN3D	-	✓	✓	**66.6**	72.7	60.2	**53.2**	**74.4**	**58.6**
Posed images									
NeRF-Det++ [7]	TIP'25	✓	✗	-	53.3	30.0	-	-	-
ImGeoNet [17]	ICCV'23	✓	✗	-	54.8	28.4	-	-	-
NeRF-DetS [5]	arXiv'24	✓	✗	-	57.6	35.6	-	-	-
GO-N3RDet [19]	CVPR'25	✓	✗	-	58.6	33.7	-	-	-
3DGeoDet [18]	TMM'25	✓	✗	-	59.6	34.3	-	-	-
ImVoxelNet [20]	WACV'22	✗	✗	-	46.7	23.4	-	-	-
NeRF-Det [4]	ICCV'23	✗	✗	-	53.5	27.4	-	-	-
MVSDet [21]	NeurIPS'24	✗	✗	-	56.2	31.3	-	-	-
DUSt3R → PQ	-	✗	✗	44.1	50.3	27.7	10.2	27.0	5.0
DUSt3R → **TUN3D**	-	✗	✗	**55.2**	**57.4**	**35.6**	**37.9**	**34.8**	**13.4**
Unposed images									
DUSt3R → PQ	-	✗	✗	39.7	39.1	16.7	5.0	10.9	1.3
DUSt3R → **TUN3D**	-	✗	✗	**46.5**	**44.0**	**20.7**	**20.8**	**11.0**	**2.2**

annotated in the floorplan. In our experiments, we use the SpatialLM [13] layout annotation created by lifting those floorplan annotations into the 3D space.

B. Implementation Details

Architecture. The backbone and the neck of our model are the same as in TR3D [3]. From the architectural perspective, our main novelty is the layout head. In the case of using our novel wall parameterization, the information about spatial dimensions of a scene is distilled into a single vector. Precisely, we calculate 10 z-quantiles and encode them using a three-layer MLP with a ReLU into a vector of size 40. After that, this vector is concatenated to 128-channel 2D features passed from the neck so that the floor-projected representation contains 128 + 40 = 168 channels.

Training. The training procedure follows the default mmdetection's learning schedule. We employ the Adam optimizer with an initial learning rate of 0.001 and weight decay of 0.0001. To control the size of input scenes, we sample a maximum of 100,000 points per scene. All experiments were conducted using a single Nvidia H100 GPU.

Inference. During the inference, we generate excessive predictions and suppress the redundant ones using NMS, processing objects and walls separately. For objects, a prediction is removed if the 3D IoU between predicted bounding boxes exceeds 0.5. For walls, a prediction is removed if the maximum pairwise distance between all four predicted corners of the walls falls under 75 cm.

C. Metrics

To measure detection quality, we use mean average precision (mAP) under IoU thresholds of 0.25 and 0.5 as a metric.

According to the standard protocol, layout accuracy is assessed with an F1 score for ScanNet [34], S3DIS [36], and ARKitScenes [35] datasets, with walls being matched based on maximum corner-to-corner distance. The Structured3D [37] benchmark uses a variant of the F1 score, where matching is based on projection into the floorplan: specifically, two walls are matched if the IoU of their projections exceeds a given threshold; 0.25 and 0.5 are selected to mimic the evaluation protocol of object detection.

VI. RESULTS

A. Comparison with Prior Approaches

Point clouds. We benchmark the proposed approach in the most common scenario with ground truth point clouds as inputs and report scores on ScanNet and S3DIS (Tab. I), ARKitScenes (Tab. II) and Structured3D (Tab. III). Our method demonstrates superiority in all four benchmarks, with significant performance gains w.r.t. prior state-of-the-art layout estimation methods ($+23.6$ F1 over PQ-Transformer on S3DIS, $+5.8$ and $+4.4$ F1 Omni-PQ on ScanNet and ARKitScenes, $+4.0$ F1@0.25 over SpatialLM on Structured3D). On S3DIS, we report state-of-the-art results among all existing 3D object detection methods. Overall, with consistent improvement over the baseline PQ-Transformer, TUN3D sets a new state-of-the-art in 3D scene understanding.

Posed images. While numerous approaches report 3D object detection performance on ScanNet, we are the first to address

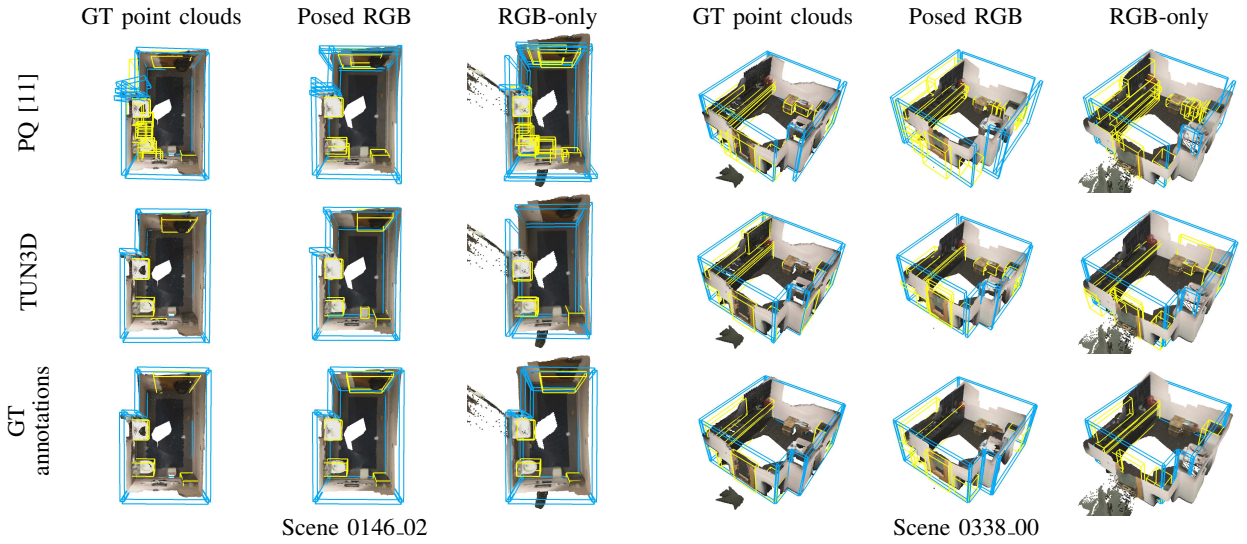


Fig. 3. Ground truth and predicted **layouts** and **objects** on ScanNet dataset.

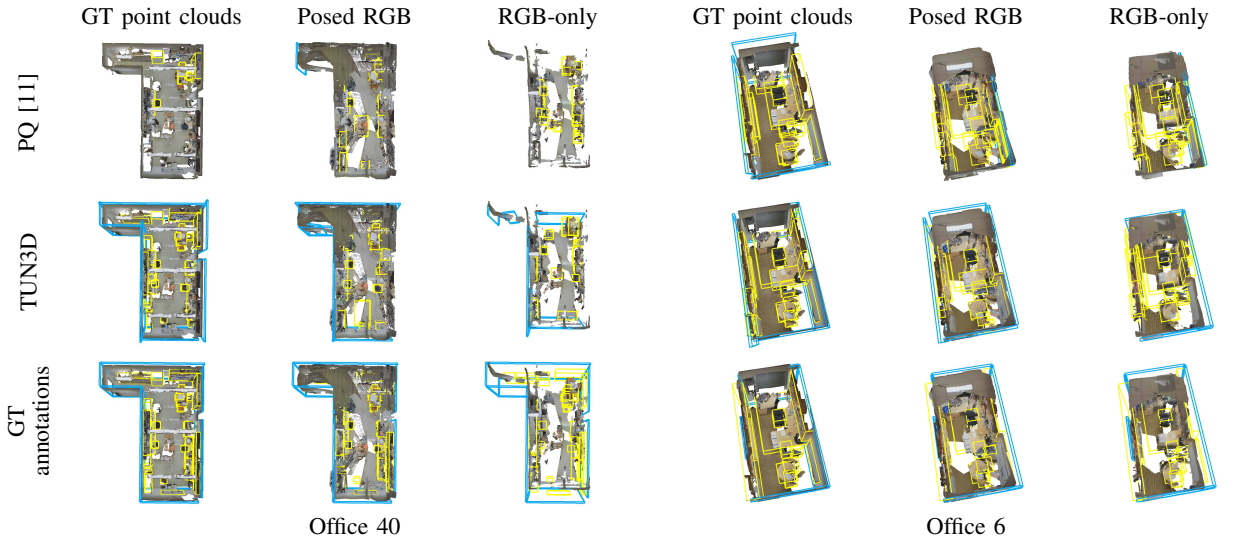


Fig. 4. Ground truth and predicted **layouts** and **objects** on S3DIS dataset.

layout estimation from posed videos in this popular indoor benchmark. So, to establish a baseline, we combine DUSt3R with PQ-Transformer (denoted as DUSt3R → PQ in Tab. I). TUN3D outperforms the baseline in both object detection and layout estimation +11.1 F1 on ScanNet, +27.7 F1 on S3DIS. For apple-to-apple comparison, we separate existing approaches based on the use of depth data for training. TUN3D clearly dominates in the depth-agnostic category and even outperforms recent depth-aware methods in terms of mAP@0.5.

TABLE II

RESULTS OF LAYOUT ESTIMATION FROM GROUND TRUTH POINTS CLOUDS ON ARKITSCENES.

Method	Layout F1
PQ [11]	10.7
Omni-PQ [10]	25.9
TUN3D	**30.3**

TABLE III

RESULTS OF LAYOUT ESTIMATION AND 3D OBJECT DETECTION FROM GROUND TRUTH POINT CLOUDS ON STRUCTURED3D.

Method	Layout		Detection	
	F1@0.25	F1@0.5	F1@0.25	F1@0.5
RoomFormer [9]	70.4	67.2	-	-
SceneScript [12]	83.1	80.8	49.1	36.8
SpatialLM [13]	86.5	84.6	65.6	52.6
TUN3D	**90.5**	**89.6**	**73.9**	**65.4**

Unposed images. In the third track, there are no predecessors reporting scene understanding results on real scans. Respectively, we also obtain reference numbers with a combination of DUSt3R → PQ. As could be expected, DUSt3R → TUN3D outperforms the baseline by a large margin. More surprisingly, predicted layouts appear to be more accurate

than the ones produced by DUSt3R → PQ *with* poses, which actually means that we can drop one input modality (camera poses) yet achieve the same quality of layout estimation.

B. Ablation Experiments

Inference time. In Tab. IV, we compare the efficiency of TUN3D and other methods that report both layout estimation and object detection on ScanNet and S3DIS. As can be seen, our approach is orders of magnitude faster than LLM-based SpatialLM and 4x faster than PQ-Transformer.

TABLE IV
INFERENCE TIME (MS) WITH POSED IMAGES

Method	ScanNet	S3DIS
PQ [11]	217	256
SpatialLM [13]	7935	7976
TUN3D	**49**	**79**

Our main architectural choices are driven by efficiency, including adapting a TR3D-like backbone. To prove them, we experiment with the arguably more accurate recent 3D object detection model, UniDet3D [8], that produces a set of 3D predictions with a transformer decoder. Following the internal logic of the UniDet3D's processing pipeline, we add a layout head that predicts walls as 4×3D offsets. In Tab. VIII, we compare UniDet3D against TUN3D with the same wall parameterization and the final version of TUN3D with the proposed parameterization. Evidently, TUN3D is 1.7x faster while improving the layout F1 score by $+4.4$. 3D object detection results are inferior to the ones of UniDet3D on ScanNet and superior on S3DIS ($+1.2$ mAP@0.5 reported in Tab. I), so an ultimate leader cannot be identified. Overall, we claim that with our architecture, we reach a decent balance of efficiency and accuracy.

Number of images. To identify the sufficient level of coverage, we vary the number of input images. According to the Tab. V, scores increase with the number of frames. We use 45 frames in the final version of TUN3D, which is on par with our competitors: ImVoxelNet [20] uses 50 images, NeRF-Det [4] – 100 images.

Number of z-quantiles. In Tab. VI, we identify the best level of detail when encoding spatial information about the scene. Our model demonstrates reasonable performance even without such information. Yet, 10 z-quantiles bring $+5.2$ F1 on ScanNet and $+4.3$ F1 on S3DIS, which is a noticeable improvement obtained with a negligible computation overhead having no effect on the overall inference time.

TABLE V
RESULTS OF LAYOUT ESTIMATION AND OBJECT DETECTION FROM POSED IMAGES ON SCANNET WITH VARYING NUMBER OF IMAGES.

# Images	Layout F1	Detection mAP@0.25	Detection mAP@0.5
15	43.3	47.1	26.5
25	50.1	52.2	33.1
35	**55.2**	56.1	35.0
45	**55.2**	**57.4**	**35.6**

TABLE VI
RESULTS OF LAYOUT ESTIMATION FROM POSED IMAGES ON SCANNET AND S3DIS WITH VARYING NUMBER OF z-QUANTILES.

# z-quantiles	ScanNet	S3DIS
0	50.0	33.6
1	53.3	33.3
2	53.8	35.3
5	55.1	36.4
10	**55.2**	**37.9**

TABLE VII
RESULTS OF LAYOUT ESTIMATION FROM POSED IMAGES ON SCANNET WITH DIFFERENT WALL PARAMETERIZATIONS.

Wall parameterization	# parameters	Layout F1
PQ [11]	8	51.2
4×3D offsets	12	53.2
2×3D offsets + height	7	53.9
2×2D offsets + height	5	**55.2**

TABLE VIII
RESULTS OF LAYOUT ESTIMATION AND OBJECT DETECTION FROM GROUND TRUTH POINT CLOUDS ON SCANNET WITH DIFFERENT BACKBONE ARCHITECTURES.

Method	Wall param.	Layout F1	Detection mAP@0.25	Time, ms
UniDet3D+layout	4×3D offsets	61.8	**77.0**	213
TUN3D	4×3D offsets	63.2	72.7	**127**
TUN3D	ours	**66.6**	72.7	**127**

Wall parameterization. To prove the efficiency of our proposed wall parameterization, we test it against other options in Tab. VII. Our parameterization is not only expressed with as few as 5 parameters, but also improves layout F1 score by $+1.3$ on ScanNet.

VII. CONCLUSION

We introduced TUN3D, the first method of joint layout prediction and 3D object detection in real scans that takes multi-view images as an input. Besides, TUN3D is trained without ground-truth camera poses or depths, hence relaxing the input data requirements to casually captured images or videos. To achieve this, we developed a lightweight sparse-convolutional model with two single-task heads and proposed a novel and effective layout parameterization. Through experiments across multiple benchmarks and various data modalities, we proved that TUN3D sets a new state-of-the-art in joint layout estimation and 3D object detection from ground-truth point clouds, posed and unposed images, marking a new milestone in holistic indoor scene understanding.

This work was supported by the The Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000C313925P4H0002; grant No 139-15-2025-012).

REFERENCES

[1] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-free 3d object detection via transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2949–2958.

[2] D. Rukhovich, A. Vorontsova, and A. Konushin, "Fcaf3d: Fully convolutional anchor-free 3d object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 477–493.

[3] ——, "Tr3d: Towards real-time indoor 3d object detection," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 281–285.

[4] C. Xu, B. Wu, J. Hou, S. Tsai, R. Li, J. Wang, W. Zhan, Z. He, P. Vajda, K. Keutzer, *et al.*, "Nerf-det: Learning geometry-aware volumetric representation for multi-view 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 23 320–23 330.

[5] C. Huang, X. Li, Y. Qu, C. Wu, X. Li, S. Zhang, and L. Cao, "Nerf-dets: Enhanced adaptive spatial-wise sampling and view-wise fusion strategies for nerf-based indoor multi-view 3d object detection," *arXiv preprint arXiv:2404.13921*, 2024.

[6] Y. Zhu, L. Hui, Y. Shen, and J. Xie, "Spgroup3d: Superpoint grouping network for indoor 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 7811–7819.

[7] C. Huang, Y. Hou, W. Ye, D. Huang, X. Huang, B. Lin, and D. Cai, "Nerf-det++: Incorporating semantic cues and perspective-aware depth supervision for indoor multi-view 3d detection," *IEEE Transactions on Image Processing*, 2025.

[8] M. Kolodiazhnyi, A. Vorontsova, M. Skripkin, D. Rukhovich, and A. Konushin, "Unidet3d: Multi-dataset indoor 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 4, 2025, pp. 4365–4373.

[9] Y. Yue, T. Kontogianni, K. Schindler, and F. Engelmann, "Connecting the dots: Floorplan reconstruction using two-level queries," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 845–854.

[10] H.-a. Gao, B. Tian, P. Li, X. Chen, H. Zhao, G. Zhou, Y. Chen, and H. Zha, "From semi-supervised to omni-supervised room layout estimation using point clouds," *arXiv preprint arXiv:2301.13865*, 2023.

[11] X. Chen, H. Zhao, G. Zhou, and Y.-Q. Zhang, "Pq-transformer: Jointly parsing 3d objects and layouts from point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2519–2526, 2022.

[12] A. Avetisyan, C. Xie, H. Howard-Jenkins, T.-Y. Yang, S. Aroudj, S. Patra, F. Zhang, D. Frost, L. Holland, C. Orme, *et al.*, "Scenescript: Reconstructing scenes with an autoregressive structured language model," in *European Conference on Computer Vision*. Springer, 2024, pp. 247–263.

[13] Y. Mao, J. Zhong, C. Fang, J. Zheng, R. Tang, H. Zhu, P. Tan, and Z. Zhou, "Spatiallm: Training large language models for structured indoor modeling," *arXiv preprint arXiv:2506.07491*, 2025.

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

[15] J. Gwak, C. Choy, and S. Savarese, "Generative sparse detection networks for 3d single-shot object detection," in *European conference on computer vision*. Springer, 2020, pp. 297–313.

[16] A. Avetisyan, T. Khanova, C. Choy, D. Dash, A. Dai, and M. Nießner, "Scenecad: Predicting object alignments and layouts in rgb-d scans," in *European Conference on Computer Vision*. Springer, 2020, pp. 596–612.

[17] T. Tu, S.-P. Chuang, Y.-L. Liu, C. Sun, K. Zhang, D. Roy, C.-H. Kuo, and M. Sun, "Imgeonet: Image-induced geometry-aware voxel representation for multi-view 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6996–7007.

[18] Y. Zhang, Y. Wang, Y. Cui, and L.-P. Chau, "3dgeodet: General-purpose geometry-aware image-based 3d object detection," *arXiv preprint arXiv:2506.09541*, 2025.

[19] Z. Li, H. Yu, Y. Ding, J. Qiao, B. Azam, and N. Akhtar, "Go-n3rdet: Geometry optimized nerf-enhanced 3d object detector," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 27 211–27 221.

[20] D. Rukhovich, A. Vorontsova, and A. Konushin, "Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022, pp. 2397–2406.

[21] Y. Xu, C. Li, and G. H. Lee, "Mvsdet: Multi-view indoor 3d object detection via efficient plane sweeps," *Advances in Neural Information Processing Systems*, vol. 37, pp. 132 824–132 842, 2024.

[22] Y. Nie, X. Han, S. Guo, Y. Zheng, J. Chang, and J. J. Zhang, "Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 55–64.

[23] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "Led2-net: Monocular 360deg layout estimation via differentiable depth rendering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 956–12 965.

[24] Z. Jiang, Z. Xiang, J. Xu, and M. Zhao, "Lgt-net: Indoor panoramic room layout estimation with geometry-aware transformer network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1654–1663.

[25] Y. Dong, C. Fang, L. Bo, Z. Dong, and P. Tan, "Panocontext-former: Panoramic total scene understanding with a transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28 087–28 097.

[26] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 455–14 465.

[27] C. Zhu, T. Wang, W. Zhang, J. Pang, and X. Liu, "Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness," *arXiv preprint arXiv:2409.18125*, 2024.

[28] A.-C. Cheng, H. Yin, Y. Fu, Q. Guo, R. Yang, J. Kautz, X. Wang, and S. Liu, "Spatialrgpt: Grounded spatial reasoning in vision-language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 135 062–135 093, 2024.

[29] D. Zheng, S. Huang, and L. Wang, "Video-3d llm: Learning position-aware video representation for 3d scene understanding," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 8995–9006.

[30] H. Zhi, P. Chen, J. Li, S. Ma, X. Sun, T. Xiang, Y. Lei, M. Tan, and C. Gan, "Lscenellm: Enhancing large 3d scene understanding using adaptive visual preferences," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 3761–3771.

[31] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," in *European Conference on Computer Vision*. Springer, 2024, pp. 71–91.

[32] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[33] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.

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

[35] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. Schwartz, *et al.*, "Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data," *arXiv preprint arXiv:2111.08897*, 2021.

[36] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1534–1543.

[37] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou, "Structured3d: A large photo-realistic dataset for structured 3d modeling," in *European Conference on Computer Vision*. Springer, 2020, pp. 519–535.