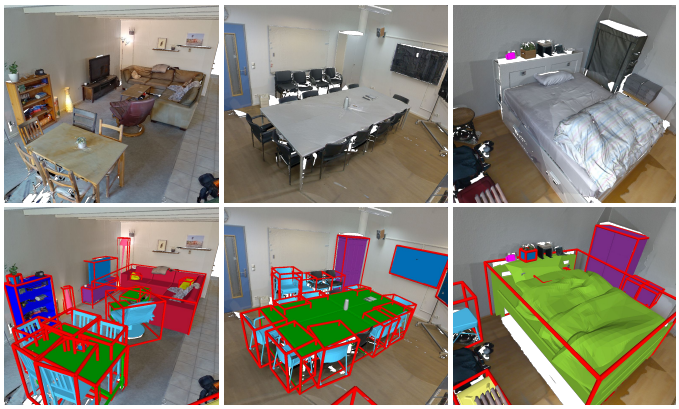
Leveraging Automatic CAD Annotations for Supervised Learning in 3D Scene Understanding

Yuchen Rao[1*] Stefan Ainetter[1*] Sinisa Stekovic[2] Vincent Lepetit[2] Friedrich Fraundorfer[1]

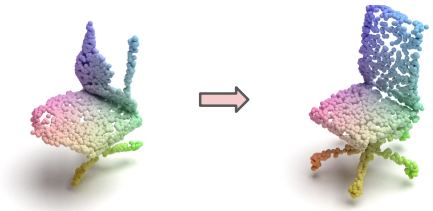
[1] Inst. of Visual Computing, Graz Univ. of Technology, Austria

[2] LIGM, École des Ponts et Chaussees, IP Paris, CNRS, France

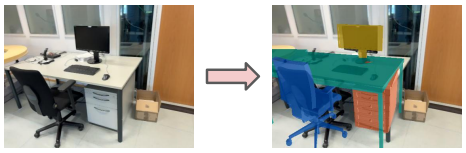
Project page: stefan-ainetter.github.io/SCANnotatepp



Examples of our annotations for ScanNet++



Task 1: Supervised point cloud completion



Task 2: Supervised CAD model retrieval and alignment

Figure 1. We provide high-quality shape and pose annotations for objects in RGB-D scans for the ScanNet++ dataset. Because our pipeline is automatic, we can annotate almost all objects in the scene even when they are only partially visible, and the annotations are of consistently high quality. We demonstrate that we can use these annotations for supervised learning including point cloud completion (Task 1) and CAD model retrieval and alignment (Task 2), and that the model performance can be increased either by using the annotations as additional training data, or by using them to fine tune a pre-trained model for a previously unseen dataset.

Abstract

High-level 3D scene understanding is essential in many applications. However, the challenges of generating accurate 3D annotations make development of deep learning models difficult. We turn to recent advancements in automatic retrieval of synthetic CAD models, and show that data generated by such methods can be used as high-quality ground truth for training supervised deep learning models. More exactly, we employ a pipeline akin to the one previously used to automatically annotate objects in ScanNet scenes with their 9D poses and CAD models. This time, we apply it to the recent ScanNet++ v1 dataset, which previously lacked such annotations. Our findings demonstrate that it is not only possible to train deep learning models on these automatically-obtained annotations but that the resulting models outperform those trained on manually annotated data. We validate this on two distinct tasks: point cloud completion and single-view CAD model retrieval and alignment. Our results underscore the potential of automatic 3D annotations to enhance model performance while significantly reducing annotation costs. To support future research in 3D scene understanding, we release our annotations, which we call SCANnotate++, along with our trained models.

1. Introduction

3D scene understanding requires precise object recognition, reconstruction, and alignment. A critical building block in such technologies is the generation of high-quality 3D data, enabling use-cases of artificial intelligence applications for perceiving, interpreting, and interacting with virtual environments.

[*] The first two authors contributed equally to this work.

A variety of multi-view datasets have been developed for 3D scene understanding, including SceneNN [17], ScanNet [13], ScanNet++ [42], Matterport3D [7], and ARKitScenes [5]. However, most of these datasets lack detailed annotations for object shapes, with the exception of ScanNet, for which the Scan2CAD dataset [3] provides 3D alignments of CAD models to target objects. The lack of available annotations can be explained by the challenge of manually annotating 3D shapes which can be difficult and time consuming for the human annotator, and is therefore prone to human inconsistencies and errors. An alternative approach involves creating synthetic datasets [32, 43], which is computationally expensive and requires significant resources. Additionally, methods trained in synthetic environments are likely to fail when applied on real (physical) world scenes, revealing as the well known issue of synthetic-to-real domain gap in deep learning.

In this work, we examine whether automatic approaches for recovering 3D geometry of objects can be used to train deep learning models. In particular, we examine annotations obtained by HOC-Search [2] that jointly performs discrete and continuous search to automatically retrieve CAD models from a large database of synthetic shapes, *e.g.* ShapeNet [8], and aligns them with objects in 3D scenes. The method assumes that a rough estimate of the object's 3D pose is given for the target object, either through manual annotations or using deep neural networks [39], and then performs optimization in image space of the target scene to create an accurate alignment between a retrieved CAD model and the target object. As [2] further demonstrates for the ScanNet dataset [13], the quality of such annotations is comparable and often better than the manual annotations from the Scan2CAD dataset [3]. This was used by the authors of [3] to create the SCANnotate dataset, which contains CAD model alignments for 18617 objects for 1513 scenes of the ScanNet dataset.

Here, we first apply HOC-Search on the ScanNet++ v1 dataset [42] to automatically obtain additional CAD model alignments for 5290 objects from 280 scans. We call this new dataset SCANnotate++. Then, as evaluation of the annotation quality, we show that deep learning approaches can be trained on such automatically annotated data. We consider two tasks that are critical for 3D scene understanding:

First, we tackle point cloud completion and introduce a pipeline with two stages. We train the ShapeGF point cloud auto-encoder network [6] on complete point clouds from the ShapeNet dataset [8]. Then, we train a second encoder on real world partial point clouds with the objective to output the same embeddings as the first encoder, hence enabling point cloud completion. As our results for the ScanNet dataset [13] demonstrate, our model trained using automatically generated annotations performs better than the model trained on manually annotated training data from Scan2CAD [3].

Dataset	CAD Model Annotations	# 3D Scans	# Annotated Objects	Annotation Method
Pix3D [36]	Pix3D [36]	✗	10k	Manual
ARKit Scenes [5]	LASA [24]	920	11k	Manual
ScanNet [13]	Scan2CAD [3]	1513	14k	Manual
ScanNet [13]	SCANnotate [1, 2]	1513	18k	Automatic
ScanNet++ v1 [42]	SCANnotate++ (ours)	280 (*)	5k	Automatic

Table 1. **Comparison of the most popular real RGB-D datasets for indoor scene understanding where CAD model and object pose annotations are available.** (*) At the time of annotating the dataset, only ScanNet++ v1 was publicly available. Hence, the current version of SCANnotate++ provides annotations for all 280 ScanNet++ v1 scans where ground truth 3D segmentation is available.

Second, we consider CAD model retrieval and alignment from single RGB images. We show that ROCA [16] trained on the automatically generated SCANnotate dataset performs better than the original model trained on manual annotations from the Scan2CAD dataset, which shows the general usefulness of automatically generated CAD model annotations. Furthermore, we show that it is now possible to train and test ROCA on the ScanNet++ dataset by using our proposed SCANnotate++ dataset, and we are able to present compelling results.

In summary, this paper provides insights into how pipelines for automatic annotations can be beneficial when training deep learning models for 3D scene understanding. We adapt HOC-Search to automatically obtain high-quality 3D object annotations for indoor scenes. As the result, we introduce SCANnotate++, a new dataset that provides CAD model alignments for objects in the ScanNet++ dataset. In addition, we showcase two successful use-cases for training deep learning methods on such automatically obtained annotations: point cloud completion, and single-view CAD model retrieval and alignment. We will make our dataset and models available, to foster future developments in 3D scene understanding.

2. Related Work

CAD model annotations for indoor scans. We compare related popular datasets for indoor 3D scene understanding [5, 7, 13, 17, 24, 26, 42] in Table 1. For the ScanNet dataset [13], Scan2CAD [3] introduces annotations by manually retrieving CAD models from ShapeNet [8] and aligning them to target objects. However, such annotations are very difficult to obtain as they require hard labor from human annotators. Frameworks for generating automatic annotations are becoming an important aspect for tasks in 3D scene understanding [9, 11, 18, 19, 40]. In the context of CAD model representations, CAD-Estate [26] introduces a semi-automatic framework for RGB video annotation, that

first automatically retrieves 10 most promising candidates, which is followed by manual verification, and manual annotation of 2D-to-3D correspondences that are used to align CAD model with the scene. SCANnotate [1, 2] and HOC-Search [2] go one step further and perform CAD model retrieval and alignment automatically on scenes of the ScanNet dataset. In this work, we examine the aspect of training supervised learning methods on such automatically generated training data.

Supervised CAD model retrieval. The availability of CAD model annotations enables a range of applications. For example, the manual Scan2CAD annotations [3] were used to train methods for CAD model alignment from point clouds [3, 4, 22]. These methods assume that a pool of objects in the scene is given and the goal is to align objects from this pool to the objects in the scene. In this work, we evaluate the performance of ROCA [16], a supervised method for CAD model retrieval and alignment from single views. ROCA learns to perform CAD model retrieval from a candidate object pool, and estimates dense correspondences to enable alignment based on differentiable Procrustes optimization.

Supervised shape completion methods. CAD model annotations can also be useful for 3D shape completion [12, 14, 23, 30, 37, 38, 41]. Supervised shape completion methods learn more general shape representations and use generative modeling formulations to model ambiguities in reconstruction. Some approaches focus on learning representative shape priors from diverse representations, such as volumetric grids [14, 41], continuous implicit functions [10, 27–29], point clouds [6, 35], and meshes [12, 23, 37, 38], which yield impressive results on the categories used for training. Various learning architectures have been explored. For instance, ShapeGF [6] employs a VQ-VAE backbone to estimate gradient vectors at each point and then integrates these vectors to reconstruct a continuous surface. Luo et al. [25] introduce Diffusion Probabilistic Models for point cloud completion to address the challenges of noise and incompleteness in real-world scans, while Kasten et al. [20] leverage a pretrained text-to-image diffusion model to complete the surface representation based on a test prompt.

3. Automatic Generation of CAD Annotations

Given an RGB-D scan of a static scene, our goal is to generate 3D shape and 9D pose annotations for the objects in the scene. Based on findings from [1, 2], we propose an automatic annotation pipeline that relies on given 3D semantic object instance segmentation to identify the target objects to be annotated, and then utilizes CAD models from a large database for shape retrieval.

To be precise, we use the annotation pipeline proposed in [1], but replace the original shape retrieval algorithm with the more effective HOC-Search [2] algorithm. For HOC-Search, our only modification compared to the original work is the adaptation of the threshold for CAD model selection and pose refinement, which led to more accurate annotations in our experiments.

Figure 2 gives an overview of our annotation process. We achieve high-quality results, in terms of 3D fitness with respect to the target object, and in terms of alignment of re-projections with the corresponding RGB-D frames of the target scan.

In the following sections, we discuss the main parts of our pipeline, including data pre-processing for initial object pose estimation, our shape and pose retrieval algorithm and the refinement of the retrieval results. Then, we describe our manual verification process that ensures the quality of the final annotations. Finally, we provide details and statistics of our resulting SCANnotate++ dataset.

3.1. CAD Model-Retrieval and Pose Reconstruction

Our reconstruction pipeline, based on SCANnotate [1, 2], consists of two main parts: In the first part, we perform CAD model retrieval. Unlike SCANnotate which relies on exhaustive CAD model search, we utilize HOC-Search [2] as our retrieval method. In the second part, we apply CAD model clustering and cloning and pose refinement for cloned CAD models. Afterwards, we perform manual inspection of the annotation results and, if needed, perform manual re-annotation of outliers to ensure high quality results of our annotation.

3.1.1. Preprocessing: Object Pose Initialization

Given the 3D instance segmentation of a target object and the corresponding class label, we first calculate a 3D bounding box as initialization for position and scale of the object. For each object, we estimate an oriented bounding box using Trimesh [15], whereas the bounding box is aligned with the major axes of the target object, and is utilized for pose initialization. The initial pose does not have to be very precise, as the CAD model pose is later optimized during the CAD model retrieval process.

3.1.2. Initial CAD Model and Pose Annotation

After obtaining the initial pose of the target object, we retrieve a CAD model from a shape database and predict its corresponding 3D pose to align well to the observations of the target object. We use HOC-Search [2] as retrieval method, which retrieves good results even in challenging scenarios and in presence of noise, *e.g.* when objects are only partially visible, or when depth sensor values are inaccurate or missing.

HOC-Search performs joint optimization of discrete and continuous parameters, a feature that is crucial to solve our

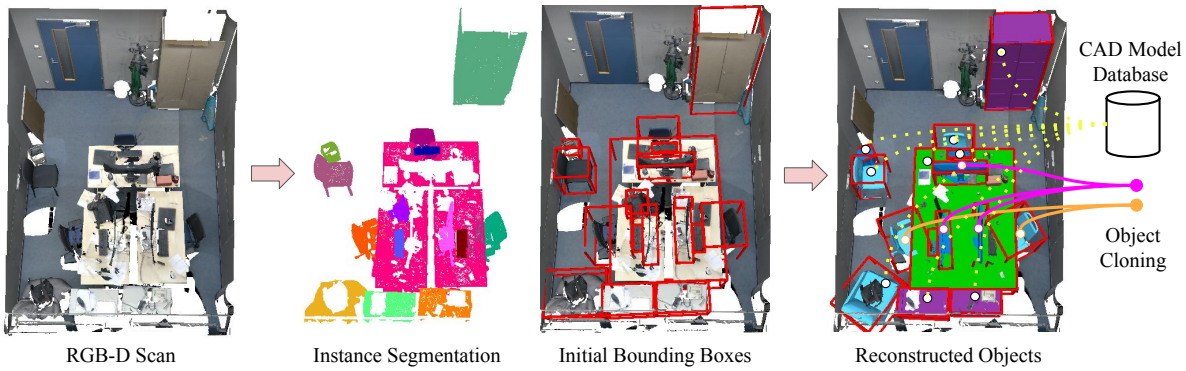


Figure 2. **Our automatic annotation pipeline.** For a given RGB-D scan, we first use the provided 3D object instance segmentation to estimate the pose of each object, visualized as red bounding boxes. The scan, 3D segmentation and initial estimated poses are then used as input for our annotation method. As final result, we provide high quality 3D shape annotations in the form of CAD models retrieved from a large shape database, and corresponding 9D pose annotations for all target objects.

annotation task. Its algorithm can be used in combination with an objective term solely dependent on data observations (for example render-and-compare or chamfer distance), and does not rely on learning. It generalizes very well and is well suited for data annotation, in contrast to 3D deep learning approaches that are prone to errors when performing inference on out-of-distribution data. In this section, we provide an overview, and we refer to the original HOC-Search [2] paper for a more detailed description about the algorithm and the implementation.

HOC-Search is based on two main components, a data structure called HOC-Tree and an efficient algorithm to search this structure. HOC-Tree represents discrete parameters as nodes of a tree. [2] proposes to organize a set of CAD models into a tree structure based on hierarchical clustering of shape similarities. Therefore, the similarity of shapes increases with the depth of the tree, which makes HOC-Tree a very useful data structure in practice. Then, HOC-Search adapts a variant of Monte Carlo Tree Search (MCTS) [33, 34] that efficiently searches tree structures by jointly optimizing discrete and continuous parameters in an iterative search.

Objective term for optimization. One iteration of HOC-Search returns a CAD model and a 3D pose, which is directly inferred from the nodes of the corresponding path through the tree. To evaluate the quality of this solution, we calculate an objective term based on render-and-compare that was originally proposed in [1]. The render-and-compare objective term $\mathcal{L}_{RC}$ can be described as

$$\mathcal{L}_{\text{RC}} = \lambda_{\text{dpt}}\mathcal{L}_{\text{dpt}} + \lambda_{\text{Sil}}\mathcal{L}_{\text{Sil}} + \lambda_{\text{CD}}\mathcal{L}_{\text{CD}} \,, \tag{1}$$

where $\mathcal{L}_{\text{dpt}}$ is the L1-distance between depth maps of the target object ant the CAD model, $\mathcal{L}_{\text{Sil}}$ is the IoU between silhouettes of target object and CAD model, and $\mathcal{L}_{\text{CD}}$ defines the single-direction chamfer distance, with λ_{dpt}, λ_{Sil} and λ_{CD} the corresponding weights. The weights and the hyperparameters for MCTS have been taken from [2]. For each target object we use a maximum number of 1200 MCTS iterations, which empirically showed to be sufficient to converge to a good solution.

Search algorithm: Adapted threshold for CAD model selection and pose refinement. Because the initial pose parameters which are encoded as nodes in the HOC-Tree are discretized, and due to the fact that the shape retrieval relies on a correct initial pose, it would limit the performance of HOC-Search to directly use the object pose based on the discrete parameters. Therefore, HOC-Search implements a pose refinement step, which uses gradient-based optimization to iteratively adapt the initial pose for the selected CAD model after every HOC-Search iteration. As proposed in the HOC-Search paper, we use the render-and-compare objective term as defined in Equation 1 to optimize the 9-DOF pose parameters, using the differentiable rendering pipeline of [31] and the Adam optimizer [21].

Note that HOC-Search performs this pose refinement step after every HOC-Search iteration for which the resulting CAD model provides a better score according to the objective term. We modify this criterion: We empirically found it beneficial to additionally consider CAD models for pose refinement which lead to a slightly worse score (if the new score is smaller than 1.1 times the previous best score). While this modification effectively increases the computation time as more candidates are selected for pose refinement, these additional solutions led in some cases to the most suitable result after pose refinement hence increasing the overall quality of our retrieved annotations.

3.1.3. Final Shape and Pose Refinement

After per-object CAD model and pose annotation for all objects in a scene, we exploit the prior knowledge that multiple objects of the same class in one scene are likely to have

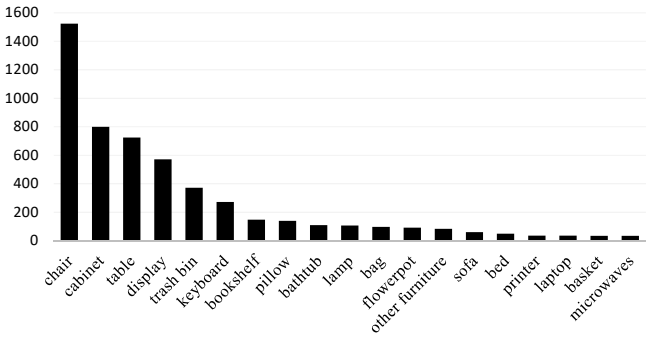


Figure 3. **Class histogram of the SCANnotate++ dataset.** The histogram shows the typical long tail distribution of objects in indoor scenes, whereas common classes like chair, cabinet, and table make up the majority of the annotated objects.

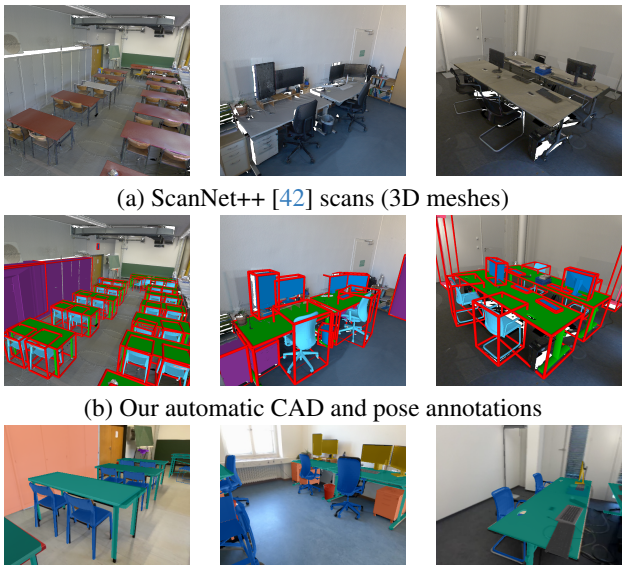
the same shape. We use CAD model clustering and cloning proposed in [1] for the classes chair, cabinet, sofa, bookshelf, display and table. The aim is to cluster all CAD models which belong to the same class according to their shape similarity, using the chamfer distance to quantify shape similarity. Afterwards, all target objects which belong to the same cluster are replaced with the one common CAD model which minimizes the sum of the objective term in Equation 1. As last step, we perform the pose refinement step—the same as HOC-Search—for all CAD models which have been effected by the CAD model cloning, to ensure that the final pose is optimal for the CAD model which represents the shape of the target object.

3.1.4. Manual Quality Verification

We visually check the quality of shape and pose for all objects after automatic annotation, and manually edited them when needed. For chairs, the most complex class in terms of geometry, we had to manually adapt 68 of the 1523 annotated objects, which is roughly 4.5%.

There are indeed some scenarios where our pipeline can produce annotations of insufficient accuracy: If the initial 3D object segmentation or the extracted bounding box proposals are significantly wrong, our pipeline is not able to produce suitable annotations. Furthermore, as our objective term is based on captured sensor data, the quality of the retrieved annotations is directly related to quality of the scan: severe occlusions and partial object captures in input scenes affect the annotation quality.

As part of the manual quality verification, if the rotation around the up-axis of a specific annotation is off by either $[0°, 90°, 180°, 270°]$ we manually adapted it. This failure case can happen for box-like object shapes (for example for classes like cabinet, stove, and washer, because they often look almost similar from each side). For cases where the shape of the retrieved object does not match the shape of the target object well, we manually re-annotated the target object with a suitable CAD model. In either of



(a) ScanNet++ [42] scans (3D meshes)

(b) Our automatic CAD and pose annotations

(c) Our annotations projected into sampled scene images

Figure 4. **Examples of our SCANnotate++ annotations.** Our annotations accurately capture the 3D geometry of target objects, and their re-projections in the images align accurately with the objects.

the aforementioned cases where we manually adapted the pose or shape of the annotation, we afterwards re-run the object pose refinement step to align the annotation to the data observations.

When the initial ground truth instance segmentation from the dataset was not of sufficient quality, we remove our corresponding annotation.

3.2. The SCANnotate++ Dataset

The resulting **SCANnotate++** dataset was generated using our annotation pipeline to annotate 280 scans from the popular ScanNet++ v1 dataset [42] providing 5290 CAD model and pose annotations. Figure 4 highlights the quality of our SCANnotate++ annotations. We rely on ground truth object instance segmentations from ScanNet++, and use ShapeNet [8] as our CAD model database which contains more than 32k objects for the classes in ScanNet++.

Figure 3 shows the distribution of annotated objects over the classes. In total, we annotated objects for 26 different semantic classes where CAD models have been available in the ShapeNet [8] database.

Because our CAD retrieval process is automatic and the objective term is based on observed data, SCANnotate++ offers a high diversity of 3D shapes. For example for the class chair, our pipeline selected 376 chair instances for the 1523 annotated objects, with a total of 6779 chair models available in the ShapeNet database.

Dataset	Number of objects	
	Training	Testing
Scan2CAD [3]	10893	3109
SCANnotate [1, 2]	14424	4193
SCANnotate++ (ours)	4343	947

Table 2. The number of annotated objects used for training and testing for the Scan2CAD [3], SCANnotate [1, 2] and SCANnotate++ (ours) datasets.

4. Experiments

In this section, we evaluate the performance of supervised learning methods trained on automatically generated CAD model annotations for ScanNet [13] and ScanNet++ [42]. More specifically, we focus on tasks of point cloud completion and single-view CAD model retrieval and alignment.

Datasets. We use the automatic annotations from SCANnotate [1, 2] for the ScanNet dataset [13] and our SCANnotate++ annotations for the ScanNet++ v1 dataset [42] as two complementary resources of high quality for training supervised learning methods. To demonstrate the feasibility and effectiveness of automatic annotations, we compare supervised learning methods trained on automatically generated data against manual annotations from Scan2CAD [3], which remains a widely adopted benchmark for annotations on the ScanNet dataset [13].

Table 2 provides a detailed breakdown of the number of training and testing objects across all categories for each dataset. For Scan2CAD and SCANnotate, we utilize the training and testing scene splits defined by the ScanNet framework. Similarly, for SCANnotate++, we adhere to the scene split settings specified in ScanNet++ v1.

4.1. Point Cloud Completion

Given an input partial point cloud of objects in ScanNet and ScanNet++, we want to reconstruct the missing 3D points. We use automatic annotations from SCANnotate and SCANnotate++ to generate complete point clouds as ground truth for supervised learning of point cloud completion and present our architecture in Figure 5.

Our model. We train our model in two phases. In the first phase, we train an auto-encoder following the ShapeGF framework [6] for point cloud completion, using ground truth point clouds sampled from annotated meshes. Here, encoder $z_c = f_{\text{enc}}(x_c)$ encodes a complete point cloud x_c into a latent z_c, and decoder $x_c^{\sim} = f_{dec}(z_c)$ decodes point cloud $x_c^{\sim}$ from z_c. In the second phase, we introduce a second encoder $z_p = g_{\text{enc}}(x_p)$, with the same architecture as $f_{\text{enc}}(\cdot)$, to encode partial input point clouds x_p into latent z_p. We use a weighted combination of Mean Squared Error (MSE) and Kullback-Leibler (KL) divergence:

$$\mathcal{L}_{\text{Latent}} = \lambda_{\text{MSE}}\mathcal{L}_{\text{MSE}}(z_p, z_c) + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}(z_p, z_c)\,, \qquad (2)$$

where λ_{MSE} is set to 1 and λ_{KL} is set to 0.5. Hence, we enforce z_p to be consistent with z_c. At inference, the shared fixed decoder $f_{\text{dec}}(z_p)$ from the pretrained auto-encoder is used to reconstruct the complete point cloud $x_p^{\sim}$.

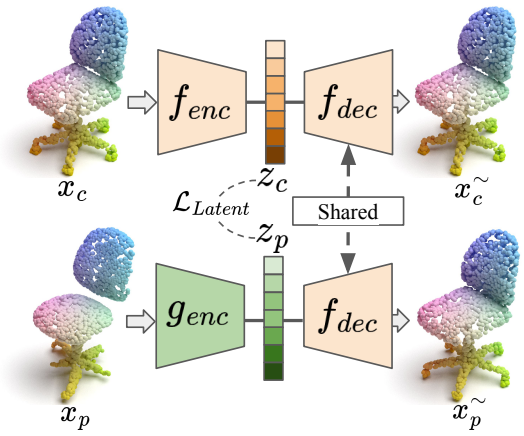


Figure 5. **Overview of our point cloud completion pipeline.** First, we train an auto-encoder based on ShapeGF [6] using complete point clouds x_c. Next, we introduce a separate encoder g_{enc} for partial input point clouds x_p from ScanNet [13] and ScanNet++ [42] and we train it to minimize the difference between latent representations of the ground truth point cloud z_c and partial point cloud z_p. The decoder f_{dec} from the pretrained auto-encoder reconstructs the complete point cloud $x_{\tilde{p}}$ from latent z_p.

Test set	Train set	CD ($\times 10^4$) ↓	EMD ($\times 10^2$) ↓
Scan2CAD [3]	Scan2CAD [3]	54.55	13.94
Scan2CAD [3]	SCANnotate [1, 2]	**38.16**	**9.84**
SCANnotate	SCANnotate	42.66	10.23
SCANnotate	SCANnotate, SCANnotate++	**38.52**	**9.79**
SCANnotate++	SCANnotate,	40.18	9.92
SCANnotate++	SCANnotate & SCANnotate++	**37.43**	**9.65**

Table 3. **Point cloud completion results on scenes of ScanNet [13] and ScanNet++ [42].** Training our network on automatic annotations from SCANnotate [1, 2] results in much better performance compared to training on manually annotated Scan2CAD [3]. Training on both SCANnotate and our SCANnotate++ further improves performance of our network.

Training details. To generate the input point cloud, we first extract partial object meshes from the target objects in the ScanNet [13] or ScanNet++ [42] datasets and uniformly sample 2048 points from these meshes. We obtain the ground truth point clouds by directly sampling 2048 points from the corresponding ground truth meshes. During training, we followed the ShapeGF configuration by using

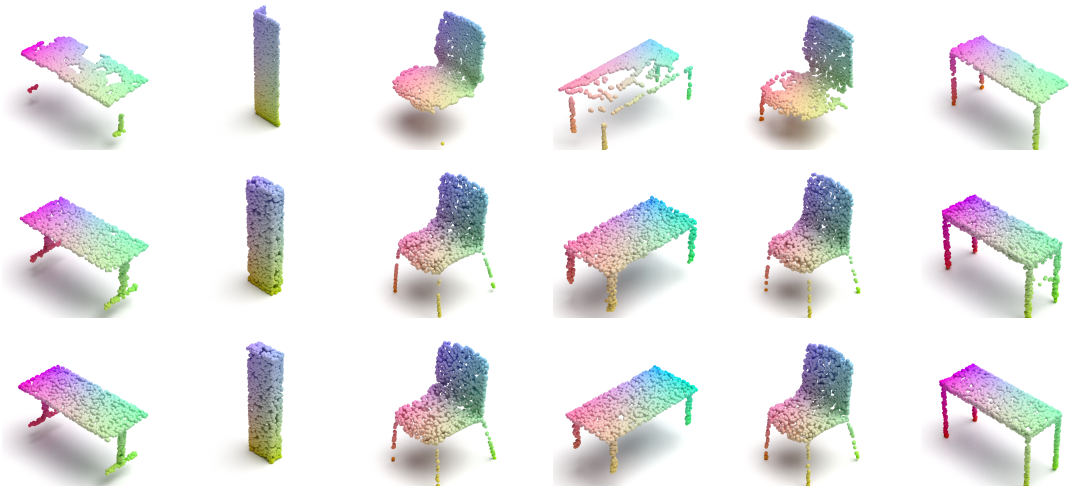


Figure 6. **Qualitative results for point cloud completion on the SCANnotate++ dataset.** The visualization presents results in the following order: Top row shows partial point cloud input, middle row shows the reconstructed point cloud, and bottom row shows the ground truth point cloud. These results demonstrate the usefulness of our annotations for learning point cloud completion for ScanNet++ [42].

Dataset	CAD Annotations	Alignment Accuracy ↑		Retrieval-Aware Alignment Accuracy ↑	
		per class	per instance	per class	per instance
ScanNet	Scan2CAD [3]	19.6	25.4	16.2	19.7
ScanNet	SCANnotate [1, 2]	**27.5**	**33.0**	**20.2**	**22.5**
ScanNet++	SCANnotate++ (ours)	27.3	33.2	17.7	13.8
ScanNet++	SCANnotate++ (ours)(*)	**30.5**	**34.9**	**20.6**	**16.1**

Table 4. **CAD model retrieval and alignment results for ROCA [16] on different datasets.** Training on the automatic annotations from SCANnotate [1, 2] enhances performance on ScanNet compared to using manual annotations from Scan2CAD, proving highly effective for CAD model retrieval and alignment tasks. Similarly, our SCANnotate++ annotations enable generalization to ScanNet++. (*) Pre-training on SCANnotate further boosts performance on ScanNet++. We use the official train and test splits from ScanNet and ScanNet++.

the Adam optimizer with a batch size of 32, a learning rate of 0.001, and the training for 1200 epochs.

Evaluation. We evaluate our completion method on test sets of manual annotations from Scan2CAD [3], and automatic annotations from SCANnotate [1, 2] and SCANnotate++. We measure the performance using the Chamfer Distance (CD) and Earth Mover's Distance (EMD).

As Table 3 shows, our point cloud completion method can be trained on automatically generated training data to produce high-quality results. To further confirm the quality of automatic annotations, we provide results of training our method on manual annotations from Scan2CAD [3]. Even when tested on the test set of Scan2CAD annotations, the method when using our dataset performs much better. We argue that SCANnotate contains more annotations than Scan2CAD, which naturally leads to better performance. Increasing the number of automatic annotations by including our SCANnotate++ annotations improves the quality of reconstructions even further. Nonetheless, due to the limited size of ScanNet++ v1 and SCANnotate++, we were not able to train our model solely on ScanNet++ v1, and hence we exclude these results. We also evaluated the performance of SCANnotate and SCANnotate++ using only an auto-encoder, which yielded a CD of 210.35 and an EMD of 20.20. Figure 6 shows additional qualitative results for point cloud completion on our SCANnotate++ dataset.

4.2. CAD Model Retrieval and Alignment

Given a single image, we want to retrieve and align corresponding 3D CAD models. ROCA [16] showed that supervised learning methods can be applied when trained on manual annotations from Scan2CAD [3]. We show such methods can be effectively trained on automatic annotations from SCANnotate [1, 2] and our SCANnotate++.

ROCA [16]. ROCA is a framework that, from a single image, retrieves and aligns CAD models from a pool of per-scene objects. It identifies a CAD model that closely resembles the target object in both appearance and geometry, while concurrently estimating 9-DoF alignment. This is achieved through a differentiable optimization process that leverages dense 2D-3D correspondences alongside Procrustes alignment.

Figure 7. **Qualitative results for CAD model retrieval and alignment for ROCA [16] trained on different annotations.** Training ROCA on automatic annotations from SCANnotate [1, 2] demonstrates improved alignment of retrieved CAD models over manual annotations from Scan2CAD [3]. Training on SCANnotate++ enables good performance on ScanNet++ [42], and pre-training on SCANnotate (*) helps to further improve these results.

Dataset processing. Alignment of objects can sometimes be ambiguous due to object symmetries. Therefore, similarly to Scan2CAD, we extract symmetries for the target objects. To determine the symmetry level of a CAD model, we rotate the original mesh in increments of 45 degrees. For each rotation, we uniformly sample 5000 points from both the original and rotated meshes, and then compute the Chamfer Distance between them. A threshold of 0.05 is used to determine whether the two models are considered identical or not. Based on the range of valid rotation angles, we categorize the models into four rotation groups, following the settings defined in Scan2CAD [3].

Evaluation. As in our first application, we evaluate our proposed method on the test sets of manual annotations from Scan2CAD [3], and automatic annotations from SCANnotate [1, 2], and our SCANnotate++. In line with the methodology described in ROCA, we evaluate the performance using two metrics: *Alignment Accuracy* for 9-DoF alignment and *Retrieval-Aware Alignment Accuracy* for retrieval. An *Alignment* is 'correct' if the object classification is accurate and the translation error is within $20cm$, the rotation error is within $20°$, and the scale ratio error is within 20%. A *Retrieval-Aware Alignment* is considered correct if the Alignment is correct and the retrieved CAD model class matches the ground truth class.

Table 4 presents class-averaged and instance-averaged performance results for Alignment Accuracy and Retrieval-Aware Alignment Accuracy across the three datasets. In comparison to Scan2CAD [3], automatic SCANnotate [1] annotations yield better CAD model and 9-DoF alignments for the ScanNet dataset [13]. Additionally, utilizing a model pretrained on SCANnotate [1, 2] before training on our SCANnotate++ annotations, improves overall performance on SCANnotate++, demonstrating that increasing the number of automatic annotations further benefits the training. Qualitative results in Figure 7 further reinforce these conclusions: ROCA [16] achieves better CAD model alignment and retrieval results with SCANnotate annotations [1, 2] compared to Scan2CAD [3], while training on both SCANnotate and SCANnotate++ further improves alignments.

5. Conclusion

We integrated recent methods for automatic object shape and pose estimation into an end-to-end pipeline to generate high-quality annotations for previously unannotated datasets, specifically applying it to the ScanNet++ dataset and making the annotations publicly available. We demonstrated the benefits of these annotations for training learning-based models on two downstream tasks. We believe that, similar to advancements in the 2D image domain, leveraging more data can significantly boost the performance of current 3D deep learning methods. Our pipeline is a first step toward generating the necessary annotations to achieve this goal.

References

[1] Stefan Ainetter, Sinisa Stekovic, Friedrich Fraundorfer, and Vincent Lepetit. Automatically Annotating Indoor Images with CAD Models via RGB-D Scans. In *IEEE Winter Conference on Applications of Computer Vision*, 2023. 2, 3, 4, 5, 6, 7, 8

[2] Stefan Ainetter, Sinisa Stekovic, Friedrich Fraundorfer, and Vincent Lepetit. HOC-Search: Efficient CAD Model and Pose Retrieval from RGB-D Scans. In *International Conference on 3D Vision*, 2024. 2, 3, 4, 6, 7, 8

[3] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Nießner. Scan2cad: Learning CAD Model Alignment in RGB-D Scans. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3, 6, 7, 8

[4] Armen Avetisyan, Angela Dai, and Matthias Nießner. End-To-End CAD Model Retrieval and 9dof Alignment in 3D Scans. In *International Conference on Computer Vision*, 2019. 3

[5] 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 *Advances in Neural Information Processing Systems*, 2021. 2

[6] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning Gradient Fields for Shape Generation, 2020. 2, 3, 6

[7] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *International Conference on 3D Vision*, 2017. 2

[8] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, and Others. ShapeNet: An Information-Rich 3D Model Repository. In *arXiv Preprint*, 2015. 2, 5

[9] Yujin Chen, Matthias Nießner, and Angela Dai. 4Dcontrast: Contrastive Learning with Dynamic Correspondences for 3D Scene Understanding. In *European Conference on Computer Vision*, pages 543–560, 2022. 2

[10] Yujin Chen, Yinyu Nie, Benjamin Ummenhofer, Reiner Birkl, Michael Paulitsch, Matthias Müller, and Matthias Nießner. Mesh2NeRF: Direct Mesh Supervision for Neural Radiance Field Representation and Generation. In *European Conference on Computer Vision*, pages 173–191, 2024. 3

[11] Julian Chibane, Francis Engelmann, Tuan Anh Tran, and Gerard Pons-Moll. Box2Mask: Weakly Supervised 3D Semantic Instance Segmentation Using Bounding Boxes. In *arXiv Preprint*, 2022. 2

[12] Angela Dai and Matthias Nießner. Scan2mesh: From Unstructured Range Scans to 3D Meshes. In *Conference on Computer Vision and Pattern Recognition*, pages 5574–5583, 2019. 3

[13] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2, 6, 8

[14] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape Completion Using 3D-Encoder-Predictor CNNs and Shape Synthesis, 2017. 3

[15] Michael Dawson-Haggerty et al. Trimesh. 3

[16] Can Gümeli, Angela Dai, and Matthias Nießner. ROCA: Robust CAD Model Retrieval and Alignment from a Single Image. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3, 7, 8

[17] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. SceneNN: A Scene Meshes Dataset with Annotations. In *International Conference on 3D Vision*, 2016. 2

[18] 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. In *European Conference on Computer Vision*, 2024. 2

[19] Guangda Ji, Silvan Weder, Francis Engelmann, Marc Pollefeys, and Hermann Blum. ARKit LabelMaker: A New Scale for Indoor 3D Scene Understanding. In *arXiv Preprint*, 2024. 2

[20] Yoni Kasten, Ohad Rahamim, and Gal Chechik. Point Cloud Completion with Pretrained Text-To-Image Diffusion Models. In *Advances in Neural Information Processing Systems*, pages 12171–12191, 2023. 3

[21] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference for Learning Representations*, 2015. 4

[22] Florian Langer, Jihong Ju, Georgi Dikov, Gerhard Reitmayr, and Mohsen Ghafoorian. FastCAD: Real-Time CAD Retrieval and Alignment from Scans and Videos. In *European Conference on Computer Vision*, 2024. 3

[23] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-Supervised Single-View 3D Reconstruction via Semantic Consistency. In *European Conference on Computer Vision*, pages 677–693, 2020. 3

[24] Haolin Liu, Chongjie Ye, Yinyu Nie, Yingfan He, and Xiaoguang Han. LASA: Instance Reconstruction from Real Scans Using a Large-Scale Aligned Shape Annotation Dataset. In *Conference on Computer Vision and Pattern Recognition*, 2024. 2

[25] Shitong Luo and Wei Hu. Diffusion Probabilistic Models for 3D Point Cloud Generation. In *Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 3

[26] Kevis-Kokitsi Maninis, Stefan Popov, Matthias Nießner, and Vittorio Ferrari. Cad-Estate: Large-Scale CAD Model Annotation in RGB Videos. In *International Conference on Computer Vision*, 2023. 2

[27] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space, 2019. 3

[28] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[29] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional Occupancy Networks, 2020. 3

[30] Yuchen Rao, Yinyu Nie, and Angela Dai. PatchComplete: Learning Multi-Resolution Patch Priors for 3D Shape Completion on Unseen Categories. In *Advances in Neural Information Processing Systems*, 2022. 3

[31] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D Deep Learning with PyTorch3D. In *arXiv Preprint*, 2020. 4

[32] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In *International Conference on Computer Vision*, 2021. 2

[33] Sinisa Stekovic, Mahdi Rad, Friedrich Fraundorfer, and Vincent Lepetit. MonteFloor: Extending MCTS for Reconstructing Accurate Large-Scale Floor Plans. In *International Conference on Computer Vision*, 2021. 4

[34] Sinisa Stekovic, Mahdi Rad, Alireza Moradi, Friedrich Fraundorfer, and Vincent Lepetit. MCTS With Refinement for Proposals Selection Games in Scene Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 4

[35] David Stutz and Andreas Geiger. Learning 3D Shape Completion Under Weak Supervision. *International Journal of Computer Vision*, 128(5):1162–1181, 2020. 3

[36] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B. Tenenbaum, and William T. Freeman. Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2

[37] Jiapeng Tang, Xiaoguang Han, Junyi Pan, Kui Jia, and Xin Tong. A Skeleton-Bridged Deep Learning Approach for Generating Meshes of Complex Topologies from Single RGB Images. In *Conference on Computer Vision and Pattern Recognition*, pages 4541–4550, 2019. 3

[38] Jiapeng Tang, Xiaoguang Han, Mingkui Tan, Xin Tong, and Kui Jia. Skeletonnet: A Topology-Preserving Solution for Learning Mesh Reconstruction of Object Surfaces from RGB Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[39] Thang Vu, Kookhoi Kim, Tung M. Luu, Thanh Nguyen, and Chang D. Yoo. SoftGroup For 3D Instance Segmentation on Point Clouds. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2

[40] Silvan Weder, Hermann Blum, Francis Engelmann, and Marc Pollefeys. LabelMaker: Automatic Semantic Label Generation from RGB-D Trajectories. In *DV*, 2024. 2

[41] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shapes, 2015. 3

[42] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A High-Fidelity Dataset of 3D Indoor Scenes. In *International Conference on Computer Vision*, 2023. 2, 5, 6, 7, 8

[43] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A Large Photo-Realistic Dataset for Structured 3D Modeling. In *European Conference on Computer Vision*, pages 519–535, 2020. 2