SHARP Challenge 2023: Solving CAD History and pArameters Recovery from Point clouds and 3D scans. Overview, Datasets, Metrics, and Baselines.

Dimitrios Mallis[⋆] dimitrios.mallis@uni.lu
Sk Aziz Ali[⋆] skaziz.ali@uni.lu
Elona Dupont[⋆] elona.dupont@uni.lu
Kseniya Cherenkova[⋆†] kseniya.cherenkova@uni.lu
Ahmet Serdar Karadeniz[⋆] ahmet.karadeniz@uni.lu
Mohammad Sadil Khan[⋆] mohammadsadil.khan@uni.lu
Anis Kacem[⋆] anis.kacem@uni.lu
Gleb Gusev[†] gleb@artec3d.com
Djamila Aouada[⋆] djamila.aouada@uni.lu

[⋆]SnT, University of Luxembourg [†] Artec 3D

Abstract

Recent breakthroughs in geometric deep learning (DL) and the availability of large computer-aided design (CAD) datasets have advanced the research on learning CAD modeling processes and relating them to real objects. In this context, 3D reverse engineering of CAD models from 3D scans is considered to be one of the most sought-after goals for the CAD industry. However, recent efforts continue to make multiple simplifying assumptions and applications in real-world settings remain limited. The SHARP Challenge 2023 aims at pushing the research a step closer to the real-world scenario of CAD reverse engineering through dedicated datasets and tracks. In this paper, we define the proposed SHARP 2023 tracks, describe the provided datasets, and propose a set of baseline methods along with suitable evaluation metrics to assess the performance track solutions. All proposed datasets[1] *along with useful routines and the evaluation metrics*[2] *are publicly available.*

1. Introduction

3D reverse engineering is defined as the deduction of intermediate design steps, complete history, and final intent in a reasonable fashion from a given 3D scan of its corresponding computer-aided design (CAD) model.

In today's digital era, using CAD software is the standard approach for designing objects ahead of manufacturing. However, CAD modeling cannot be seen as straightforward and simple procedural design, as it requires the

[1]https://cvi2.uni.lu/cc3d-data
[2]https://gitlab.uni.lu/cvi2/iccv2023-sharp-challenge

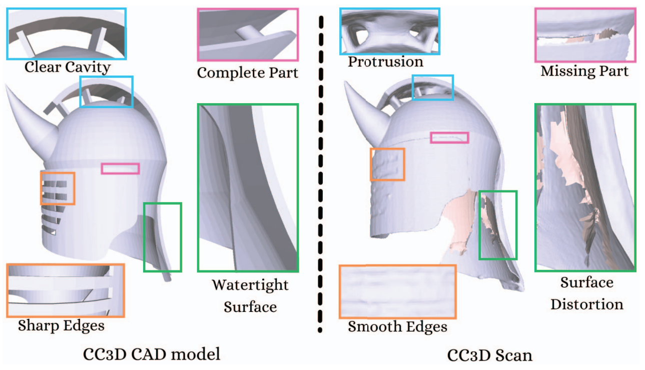

Figure 1: 3D scans in CC3D dataset [5] contain various artefacts (unwanted protrusions, smoothness over surfaces/edges, missing regions). Tackling these artefacts is essential for robust Scan-to-CAD algorithms.

skills of highly qualified engineers. Consequently, 3D reverse engineering has been a long-sought-after goal in CAD industry due to the huge resources and time that it could save [28, 9]. Such technique, also referred to as *Scan-to-CAD*, consists of scanning objects and automatically concluding their corresponding CAD models. Recently, solving this problem has attracted a large interest from the Computer Vision and Graphics research communities [31, 13, 18, 23, 16, 9, 28, 30, 12, 17], thanks to the huge advances in 3D geometric deep learning and the availability of open repositories for CAD models. The idea is to learn a mapping from 3D scans to CAD models using the available models submitted by the designers in open repositories such as OnShape [24] and 3D Content Central [3]. While the representation of 3D scans is well established and often consists of meshes or point clouds, CAD model representations may vary depending on the use case.

One approach for representing CAD models is through its final shape as a collection of geometric primitives, e.g., cylinders, cubes. Such representation is expressed by Constructive Solid Geometry (CSG) modeling, however, modern CAD workflows use Feature-based modeling as a superior alternative. Feature-based modeling is widely used as it allows to create solids by iteratively adding features such as holes, slots, or bosses, thus giving more expressiveness to designers [33]. In this setting, the final object's geometry and topology are stored as a *Boundary Representation* (B-Rep) which is a graph structure encoding parametric faces and edges, loops, and vertices [16]. Accordingly, recent works have tried to infer some of the attributes of B-Reps from point clouds to enable their editability. Some of them focused on inferring the edges [6, 35, 23], other attempts considered the prediction of the faces [27, 19], and few of them aimed at predicting both faces and edges [21, 10].

CAD modeling can also be seen as the process that allows the creation of the final model referred to as *design history*. Design history consists of the set of ordered steps that were followed by the designer using a CAD software. In feature-based modeling, these ordered steps involve the drawing of CAD sketches [17, 25, 29] followed by CAD operations such as extrusion, revolution, etc [31, 30]. Thanks to the availability of dedicated datasets [13, 30], multiple works in literature focused on learning this design history in order to automatically generate plausible CAD models [31, 34], complete partial designs according to the intent of designers [32], or predict it from point clouds [28, 31, 15, 20].

From the two aforementioned representations for CAD models, the problem of Scan-to-CAD can be seen as either related to recovering some attributes of the B-Rep from the corresponding 3D scan, or inferring the design history that allowed its creation. Despite recent findings, this problem is far from being solved. In particular, the current efforts remain very limited in the context of real-world scenarios due to the strong assumptions that are made to over-simplify the problem. For instance, it is very common to consider simple objects (e.g., cubes and cylinders) and restrict the study to the basic extrusion operation [31, 34, 28]. Furthermore, most of the works in literature assimilate 3D scans to sampled point clouds on CAD models [31, 28, 15] which is not the case in real world scenarios. Indeed, as mentioned in [6, 5], 3D scans are often subject to scanning artifacts resulting in smoothed high-level geometrical details and missing parts. Compared to uniformly sampled point clouds on CAD models, these artifacts make the problem of Scan-to-CAD more challenging.

The aim of the SHARP challenge 2023 is to encourage and help the research community to get a step closer to the real-world setting of inferring CAD history and parameters of objects from their 3D scans. In particular, different variants of the CC3D dataset [5] are proposed along with three different tracks. It is important to highlight that the CC3D dataset has the advantage of bringing pairs of realistic 3D scans with their corresponding CAD models, thus enabling a more realistic scenario of Scan-to-CAD as compared to using sampled point clouds on CAD models. Furthermore, as stated in [9], the CAD models in CC3D dataset are more complex in nature than the ones used in literature [31, 30]. The tracks proposed in SHARP challenge span over the design history and the B-Rep of the CAD models, with one of them tackling the inference of B-Rep parametric edges from 3D scans, the second focusing on the per-point segmentation of B-Rep faces from scans, and the third aiming at segmenting scans into ordered CAD operation steps and types of the corresponding design history. Simple baseline solutions to the aforementioned tracks are also proposed along with a set of dedicated metrics to assess their performance.

The rest of the paper is organised as follows. Section 2 defines the different tracks introduced in the SHARP challenge and describes the datasets. In Section 3, the baseline methods for the proposed tracks are described. The evaluation metrics used to assess the performance of the methods are described in Section 4. Section 5 reports the results of the baseline methods. Finally, conclusions and perspectives of the proposed challenge are drawn in Section 6.

2. Challenge and Dataset Description

The SHARP 2023 challenge focuses on three different tasks to bridge the gap between 3D scans and their corresponding CAD models. Three versions of the CC3D dataset [5] are used in these tracks. The CC3D dataset is derived from open CAD repositories such as 3D Content Central [3]. Unlike other alternatives such as ABC dataset [13], where the noise is usually synthetically added to the sampled point cloud, the 3D scans were obtained by virtually scanning the corresponding CAD models, using a proprietary 3D scanning pipeline developed by Artec3D [2]. As shown in Figure 1, the 3D shapes in CC3D dataset may have artifacts in the form of missing parts or protrusions due to specifics of a scanning system. The total number of samples of the CC3D dataset used in SHARP challenge is 31185, split into 25612 training samples and 5573 test samples. Note that the same sets are used for all three tracks and the corresponding B-Rep models of the training set are also provided. The overall objective is to infer different information for the CAD model given a 3D scan. While *Track 1* and *Track 2* focus on inferring geometrical and topological properties of the Boundary Representation of the CAD model, *Track 3* is centered around predicting attributes of the design history of the CAD model. More formally, let us consider a 3D scan represented by a point cloud $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}^{N\times 3}$, where $\mathbf{x}_i$ denotes the 3D coordinates of a point i and N the number of points.

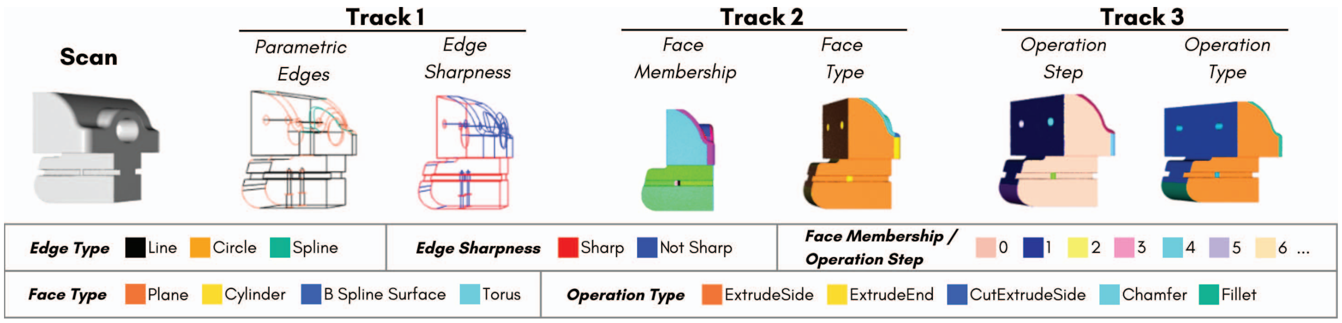

Figure 2: Predictions targets for the SHARP Challenge 2023. Proposed tracks relate to recovering geometrical and topological properties of the B-rep *(track 1 and track 2)*, as well as attributes of the design history of the CAD model *(track 3)*.

The objective of *Track 1* and *Track 2* is to predict different attributes from the B-Rep $\mathcal{B}$ of the corresponding CAD model, while *Track 3* aims at recovering other attributes from the design history $\mathcal{H}$ of the CAD model. The datasets and the attributes of each track are described next.

Track 1: Parametric Sharp Edge Inference. Given a 3D scan $\mathbf{X}$, the goal of *Track 1* (see Figure 2) is to recover: (1) the set of parametric edges $\{\mathbf{e}_j\}_{j=1}^{N_e} \in \mathcal{E}$ that are present in the the B-Rep $\mathcal{B}$, where $\mathcal{E}$ denotes the set of 3D parametric curves among circles, lines, and splines; (2) and a sharpness label $s_j \in \{0, 1\}$ indicating whether a recovered edge $\mathbf{e}_j$ is sharp ($s_j = 1$) or not ($s_j = 0$). Note that recovering these parametric B-Rep edges and their sharpness from 3D scans is critical for CAD reverse engineering. Indeed, these edges encode the topology and the geometry of the boundary of the B-Rep and some of them can be part of the sketches drawn by the designer.

CC3D-PSE dataset: This dataset consists of a set of 3D scans, annotated with a set of parametric edges and corresponding sharpness values. Both the ground truth edges and the sharpness are extracted from the B-Rep of the corresponding CAD model. The parametric edges are directly extracted from the B-Reps using OpenCascade API [1], and their sharpness value is computed as the angle between the normals of the two neighboring surface patches to the edge. The distribution of the sharpness values (*left* of Figure 3) reveal that about 30% of the edges have a sharp value lower than a corresponding angle of 10 degrees. Also, about 50% of the edges have a sharpness of 1.57 corresponding to an angle of 90 degrees, which is expected in the context of CAD models. Three types of parametric edges are considered (lines, circles and splines). From the distribution of the different edge types per model (*right* of Figure 3), we observe that the line type is the most common type. Additionally, about 15% of the CAD models in the CC3D-PSE have more than 500 edges, demonstrating the complexity of the dataset. The annotations of different edges types are parametrized as follows: (1) A line is parameterized by a start and an end point $\mathbf{p}_s = (p_s^x, p_s^y, p_s^z) \in \mathbb{R}^3$ and $\mathbf{p}_e = (p_e^x, p_e^y, p_e^z) \in \mathbb{R}^3$.

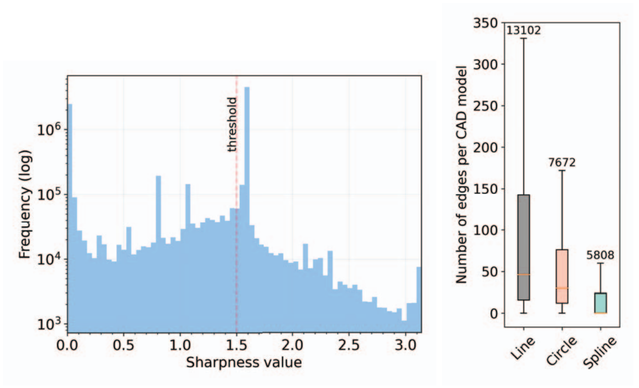

Figure 3: *(left)* Histogram of sharpness values for the edges in the CC3D-PSE dataset. *(right)* Boxplot of the number of edges per CAD models for different edge types.

(2) A circle (or circular arc) is defined by a start point $\mathbf{p}_s = (p_s^x, p_s^y, p_s^z) \in \mathbb{R}^3$, and an end point $\mathbf{p}_e = (p_e^x, p_e^y, p_e^z) \in \mathbb{R}^3$, a center point $\mathbf{p}_c = (p_c^x, p_c^y, p_c^z) \in \mathbb{R}^3$, a normal vector $\vec{\mathbf{n}} = (n_x, n_y, n_z) \in \mathbb{R}^3$, and a radius $r \in \mathbb{R}$. (3) A spline is parameterized by a degree $K \in \mathbb{N}$ and a set of keypoints $\mathbf{P}_k = [\mathbf{p}_k^1, \mathbf{p}_k^2, \dots, \mathbf{p}_k^K]$, where each $\mathbf{p}_k = (p_k^x, p_k^y, p_k^z) \in \mathbb{R}^3$. Note that the original CC3D-PSE dataset has been used in the SHARP 2022 challenge [8] and in [6] for parametric sharp edge inference. This updated version includes all B-Rep edges and their sharpness.

Track 2: Boundary-Representation (B-Rep) Face Segmentation. Given a 3D scan $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, the goal of *Track 2* (see Figure 2) is to simultaneously segment it into: **(1)** B-Rep face memberships, which consists of predicting for each point $\mathbf{x}_i$ a face that it should belong to in the B-Rep $\mathcal{B}$; **(2)** the type of that B-Rep face. In other words, the objective is to predict a point-to-face membership matrix $\mathbf{M} \in \{0, 1\}^{N \times N_f}$, where N_f denotes the number of the faces in the B-Rep $\mathcal{B}$, and a per-point face type matrix $\mathbf{T} \in \{0, 1\}^{N \times N_{f_t}}$, where N_{f_t} is the number of B-Rep face types considered. Note that each point $\mathbf{x}_i$ can only belong to a single face and should have a unique type, which suggests that each row of $\mathbf{M}$ (and $\mathbf{T}$) should have a single entry

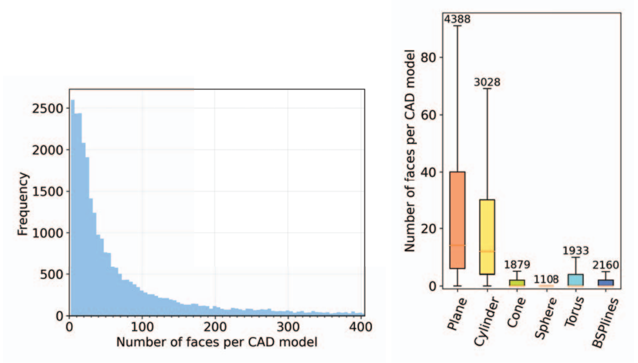

Figure 4: *(left)*: Histogram of the number of faces per CAD model for 95% the CC3D-BRepFace dataset. *(right)* Box-plot of the number of faces per CAD models for the different types of faces per CAD model.

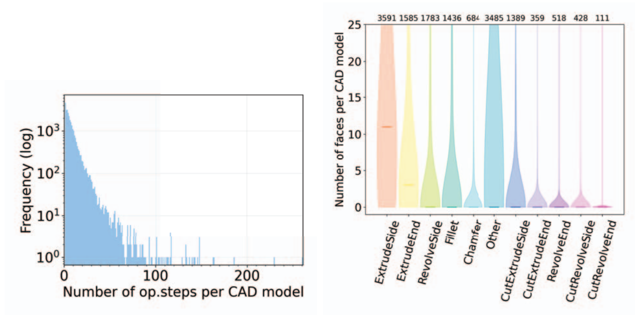

Figure 5: *(left)*: Histogram of the number of operation steps per CAD model in the CC3D-Ops dataset. *(right)*: violin plot of the number of faces per CAD models for the different operation types per CAD model. The horizontal lines represent the median values.

with 1 and 0 anywhere else. The objective of this track is to infer the B-Rep face structure from raw 3D scans which is extremely important for CAD reverse engineering.

CC3D-BRepFace dataset: Track 2 uses C3D-BRepFace, a newly introduced version of the CC3D dataset [5] that brings the annotations of the B-Rep face structure to 3D scans. In particular, the 3D scans were annotated with the face membership and type labels according to the corresponding B-Rep. This is done by processing the B-Reps with OpenCascadeAPI [1] and transferring the labels to the points of the corresponding 3D scans through nearest neighbor assignment. This results in two annotations per point for each 3D scan: **(1)** an ID of the B-Rep face that it belongs to and; **(2)** the type of the face among six possible surface types (Plane, Cylinder, Cone, Sphere, Torus, or B-Spline). Figure 4 shows statistics of the CC3D-BRepFace dataset. The CC3D-BRepFace dataset contains a wide range of model complexities with 50% of the models containing more that 37 faces and the maximum number of faces per model being 4388. We see that Plane surface is the most common face type in the dataset followed by Cylinder. Moreover, we found out that $\approx 60\%$ of the models contain at least 3 different types of face types.

Track 3: Operation Type and Step Segmentation. Given a 3D scan $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, the goal of *Track 3* is to simultaneously segment into: (1) the ordered CAD operation steps that allowed its creation; (2) the corresponding CAD operation types that were used by the designer. In other words, the objective is to predict a point-to-step membership matrix $\mathbf{S} \in \{0,1\}^{N \times N_s}$, where N_s denotes the number of steps used in the CAD history $\mathcal{H}$, and a per-point CAD operation type matrix $\mathbf{O} \in \{0,1\}^{N \times N_{O_t}}$, where N_{O_t} is the number of CAD operation types considered. Similarly to *Track 2*, each point $\mathbf{x}_i$ can only belong to a single CAD operation step and should have a unique CAD type, which suggests that each row of $\mathbf{S}$ (and $\mathbf{O}$) should have a single entry with 1 and 0 anywhere else. Although *Track 3* and *Track 2* seem to be conceptually similar, there are two main differences. Firstly, the membership task in *Track 3* aims at inferring ordered CAD steps in contrast to *Track 2* where the order of face membership does not matter. Secondly and most importantly, the nature of the labels that are targeted in the two tracks is totally different. While *Track 2* focuses on the B-Rep face structure, *Track 3* goes beyond B-Reps and aims at recovering the CAD operation history out of 3D scans. As highlighted in [9, 16, 31], recovering these operation types and steps is crucial for CAD reverse engineering as it not only provides information about how the model was constructed but also at which stage of the design the geometry was created.

CC3D-Ops dataset: The dataset used in *Track 3* extends the CC3D-Ops dataset introduced in [9]. While the original CC3D-Ops dataset contains CAD operation type and step annotations on B-Rep faces, the extended version offers the same annotations on the corresponding 3D scans at the point level. In particular, each point of a 3D scan is labelled with: **(1)** a CAD operation step identifier and; **(2)** a CAD operation type that can be one of the following types: ExtrudeSide, ExtrudeEnd, RevolveSide, Fillet, Chamfer, CutExtrudeSide, CutExtrudeEnd, RevolveEnd, CutRevolveSide, CutRevolveEnd, and Other. Figure 5 shows statistics about these annotations. In the left part of this Figure, it can be observed that most of the models of the CC3D-Ops dataset were constructed in 50 operation steps or less. In particular, 90% of the models were created in 16 or less operation steps, while the maximum number of steps per model is 261. In the right part of Figure 5, the operation type distribution is shown in violin plot. It can be seen that the most common operation type is the extrusion type but the dataset also contains models with a large number of faces created from less common operations such as revolution.

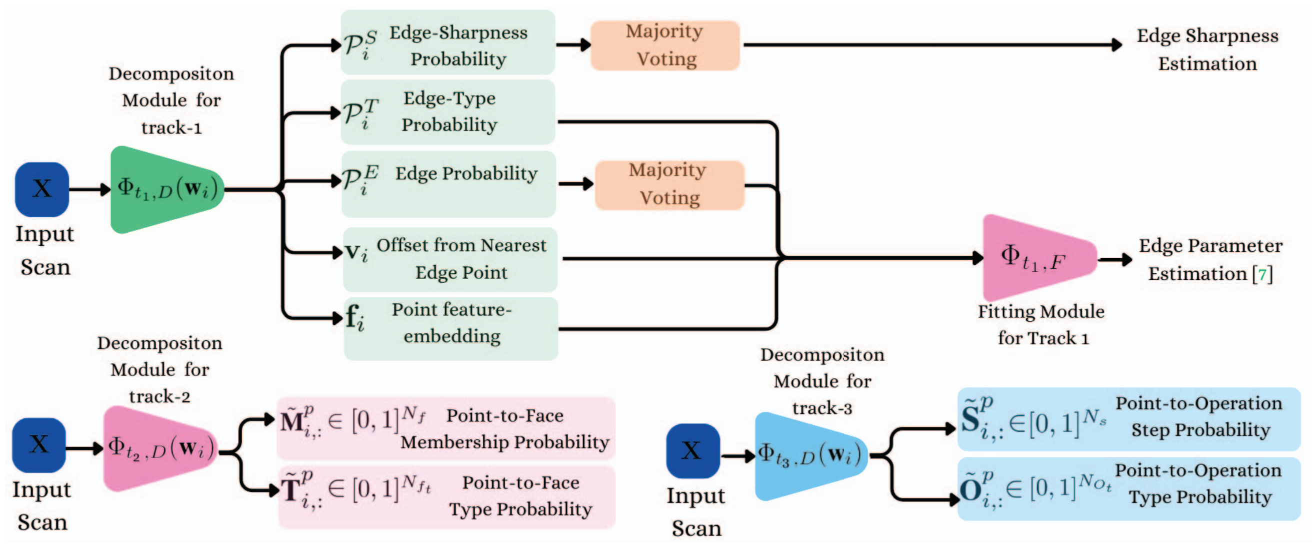
Decompositon Module for track-1
$\Phi_{t_1,D}(\mathbf{w}_i)$
X
Input Scan
$\mathcal{P}_i^S$ Edge-Sharpness Probability
$\mathcal{P}_i^T$ Edge-Type Probability
$\mathcal{P}_i^E$ Edge Probability
$\mathbf{v}_i$ Offset from Nearest Edge Point
$\mathbf{f}_i$ Point feature-embedding
Majority Voting
Majority Voting
Edge Sharpness Estimation
$\Phi_{t_1,F}$
Fitting Module for Track 1
Edge Parameter Estimation [7]
Decompositon Module for track-2
$\Phi_{t_2,D}(\mathbf{w}_i)$
$\tilde{\mathbf{M}}^p_{i,:} \in [0,1]^{N_f}$ Point-to-Face Membership Probability
$\tilde{\mathbf{T}}^p_{i,:} \in [0,1]^{N_{f_t}}$ Point-to-Face Type Probability
Decompositon Module for track-3
$\Phi_{t_3,D}(\mathbf{w}_i)$
$\tilde{\mathbf{S}}^p_{i,:} \in [0,1]^{N_s}$ Point-to-Operation Step Probability
$\tilde{\mathbf{O}}^p_{i,:} \in [0,1]^{N_{O_t}}$ Point-to-Operation Type Probability

Figure 6: Baseline model architecture for the three tracks. The input point cloud is first encoded through a PCVNN encoder. For *Track 1*, the fitting module of [7] is used to produce the set of parametric edges. For *Track 2* and *Track 3*, the mapping from the vertex embeddings to the target labels is learned directly through separate MLPs.

3. Baseline Methods

We provide simple baselines for all three tracks introduced for the SHARP challenge. Across tracks, the innput 3D scan is represented as a point cloud $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ that is uniformly downsampled to fixed number of points $N = 10k$. For all three tracks, the input $\mathbf{X}$ is mapped to a learned per-point representation $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N] \in \mathbb{R}^{N\times 960}$ through a point cloud backbone Φ_b. In this work, we model Φ_b as the Point-Voxel CNN introduced in [22]. Then, $\mathbf{W}$ is processed by output heads Φ_{t_1}, Φ_{t_2} and Φ_{t_3}, developed to address tracks one to three as discussed next.

Track 1 Baseline. The goal of this track is to learn the mapping Φ_{t_1} from the learned representation $\mathbf{W}$ to a set of parametric edges $\{\mathbf{e}_j\}_{j=1}^{N_e} \in \mathcal{E}$ and their corresponding sharpness labels $\{s_j\}_{j=1}^{N_e} \in \{0,1\}$. Our solution to this problem is based on the recently proposed SepicNet [7]. Φ_{t_1} comprises two modules, the *decomposition module* $\Phi_{t_1,D}$ followed by the *fitting module* $\Phi_{t_1,F}$ that are trained in an end-to-end manner (see Figure 6 (*left*)).

Decomposition module $\Phi_{t_1,D}$: The decomposition module detects edge points, consolidates them along the edge and groups them into different segments with primitive types identified. More specifically, for each point representation $\mathbf{w}_i$, we have $\Phi_{t_1,D}(\mathbf{w}_i) = \{\mathcal{P}_i^E, \mathbf{v}_i, \mathcal{P}_i^T, \mathbf{f}_i, \mathcal{P}_i^S\}$, where $\mathcal{P}_i^E \in [0,1]$ is the probability that the i'th point is an edge point, $\mathbf{v}_i \in \mathbb{R}^3$ is a displacement vector to the closest edge, $\mathcal{P}_i^T \in [0,1]^3$ denote the probabilities of the primitive type of the closest edge among three possible types, namely, line, circle, and spline, and $\mathbf{f}_i \in \mathbb{R}^{D_{emb}}$ is a per-point embedding that is used to cluster edge points into distinct segments. Finally, $\mathcal{P}_i^S \in [0,1]$ predicts the probability that the closest edge is sharp. To group points corresponding to the same segment, a differentiable version of the mean-shift clustering algorithm is used [26]. Curve type and sharpness for each segment are determined through majority voting. Training labels for edge points, types, sharpness and displacement vectors are derived from the ground truth whereas the per-point embeddings are trained through a triplet loss that contrasts points from the same segment to points from different segments.

Fitting module $\Phi_{t_1,F}$: The fitting module performs parameter estimation for each detected curve/segment. Curve parameters are estimated through least-squares fitting using differentiable SVD [11] on the set of segmented edge points. Different formulations for differentiable curve parameter estimation are provided for lines, arcs, and splines. Thus, a fitting loss can be formed as a sum of distances between the points sampled on the predicted parameterized segments and points sampled uniformly on the ground truth segments. During training, the decomposition module is first pretrained for 100 epochs and then jointly finetuned with the fitting module using a combination of decomposition and fitting losses. For a more thorough description of the fitting module, readers are referred to [7].

Track 2 Baseline. The goal of this track is to predict a point-to-face membership matrix $\mathbf{M} \in \{0,1\}^{N\times N_f}$ and a per-point face type matrix $\mathbf{T} \in \{0,1\}^{N\times N_{f_t}}$, ($N_f$ denotes the number of faces and N_{f_t} is the number of B-Rep face types). The above can be equivalently formulated as learning, for each $\mathbf{w}_i$, the mapping $\Phi_{t_2}(\mathbf{w}_i) = \{\tilde{\mathbf{M}}^p_{i,:}, \tilde{\mathbf{T}}^p_{i,:}\}$, where $\tilde{\mathbf{M}}^p_{i,:} \in [0,1]^{N_f}$ encodes the estimated probabilities of the i'th point belonging to one of the N_f faces, and $\tilde{\mathbf{T}}^p_{i,:} \in [0,1]^{N_{f_t}}$ denotes the face type probabilities of the i'th point. This results in two probability matrices $\tilde{\mathbf{M}}^p \in [0,1]^{N \times N_f}$ and $\tilde{\mathbf{T}}^p \in [0,1]^{N \times N_{f_t}}$ for a scan $\mathbf{X}$.

To learn $\tilde{\mathbf{T}}^p$, a standard categorical cross-entropy loss is employed *w.r.t* the ground truth per-point face types. Learning $\tilde{\mathbf{M}}^p$ for face membership prediction additionally requires addressing the inherent ambiguity in face membership labelling, as face labels are arbitrary and do not necessarily have a semantic meaning. Inspired by [19], a Hungarian matching [14] step is adopted to match the predicted face grouping to the ground truth by computing a Relaxed Intersection over Union (RIoU) [19] cost between the predicted membership probabilities and the ground truth face labels. Following class assignment recovery, the RIoU is further employed in a loss function to learn the grouping as in [19]. Finally, the estimated face memberships $\tilde{\mathbf{M}} \in \{0,1\}^{N \times N_f}$ and face types $\tilde{\mathbf{T}} \in \{0,1\}^{N \times N_{f_t}}$ are obtained through majority voting from $\tilde{\mathbf{M}}^p$ and $\tilde{\mathbf{T}}^p$, respectively. Note that since $\mathbf{X}$ is downsampled with $N = 10k$ during training, an upsampling of the predictions to the original scan resolution is used at inference.

Track 3 Baseline. The goal of this track is to predict a point-to-step membership matrix $\mathbf{S} \in \{0,1\}^{N \times N_s}$ and a per-point CAD operation matrix $\mathbf{O} \in \{0,1\}^{N \times N_{O_t}}$, where N_s denotes the number of the steps used in the CAD history and N_{O_t} is the number of CAD operation types considered. Similarly to *Track 2*, the above can be equivalently formulated as learning, for each per-point representation $\mathbf{w}_i$, the mapping $\Phi_{t_3}(\mathbf{w}_i) = \{\tilde{\mathbf{S}}^p_{i,:}, \tilde{\mathbf{O}}^p_{i,:}\}$ where $\tilde{\mathbf{S}}^p_{i,:} \in [0,1]^{N_s}$ encode the estimated probabilities of the i'th point belonging to one of the N_s CAD operation steps, and $\tilde{\mathbf{O}}^p_{i,:} \in [0,1]^{N_{O_t}}$ denotes the CAD operation type probabilities of the i'th point. This yields two probability matrices $\tilde{\mathbf{S}}^p \in [0,1]^{N \times N_s}$ and $\tilde{\mathbf{O}}^p \in [0,1]^{N \times N_{O_t}}$ for a scan $\mathbf{X}$. Compared to the proposed solution for Section 3, the operation steps are ordered and thus, there is no need to address any labelling ambiguity. Hence, both $\tilde{\mathbf{S}}^p$ and $\tilde{\mathbf{O}}^p$ can be learned through a standard categorical cross-entropy loss. The final predicted operation steps $\tilde{\mathbf{S}} \in \{0,1\}^{N \times N_s}$ and types $\tilde{\mathbf{O}} \in \{0,1\}^{N \times N_{O_t}}$ are obtained by majority voting from the learned probability matrices $\tilde{\mathbf{S}}^p$ and $\tilde{\mathbf{O}}^p$, respectively. Note that the upsampling of the predictions is also performed here as described in Section 3 for *Track 2*.

4. Evaluation Metrics

To evaluate produced solutions, a set of dedicated metrics are considered to form a final score between 0 and 1. Evaluation is conducted in th Codalab platform[4,5,6].

[4]https://codalab.lisn.upsaclay.fr/competitions/13629
[5]https://codalab.lisn.upsaclay.fr/competitions/13956
[6]https://codalab.lisn.upsaclay.fr/competitions/13676

Evaluation Metrics for Track 1. The evaluation of *Track 1* consists of assessing the quality of the estimated parametric edges $\{\tilde{\mathbf{e}}_j\}_{j=1}^{\tilde{N}_e}$ and their predicted sharpness with respect to the ground truth $\{\mathbf{e}_j\}_{j=1}^{N_e}$. Note that the number of predicted edges $\tilde{N}_e$ can be different from the ground truth number N_e. This is achieved following three criteria that are described in the following. For notation simplicity, we will denote the predicted set of edges by $\tilde{\mathbf{e}}$ and the ground truth ones by $\mathbf{e}$.

Edge Recovery Score: This score measures the similarity between the predicted and the ground truth sets of parametric edges denoted by $\tilde{\mathbf{e}}$ and $\mathbf{e}$, respectively. Given the parametric formulation of the predicted and ground truth edges described in Section 2, the first step is to uniformly sample a set of 3D points on these edges proportionally to their length. This results in two 3D point sets $\tilde{\mathrm{Z}} = \{\tilde{\zeta}_i\}_{i=1}^{\tilde{N}_\zeta}$ and $\mathrm{Z} = \{\zeta_i\}_{i=1}^{N_\zeta}$ for the predicted and ground truth edges, respectively. Then, two unidirectional Chamfer distances [4] are separately computed between the sampled point set on the predicted edges $\tilde{\mathrm{Z}}$ and the sampled one on the ground truth Z as follows,

$$d_{CD}\left(\tilde{\mathrm{Z}}, \mathrm{Z}\right) = \frac{1}{\tilde{N}_\zeta D_{\tilde{\mathrm{Z}}}} \sum_{i=1}^{\tilde{N}_\zeta} \min_{\zeta_j \in \mathrm{Z}} \|\tilde{\zeta}_i - \zeta_j\|_2^2 \,, \tag{1}$$

$$d_{CD}\left(\mathrm{Z}, \tilde{\mathrm{Z}}\right) = \frac{1}{N_\zeta D_{\mathrm{Z}}} \sum_{i=1}^{N_\zeta} \min_{\tilde{\zeta}_j \in \tilde{\mathrm{Z}}} \|\zeta_i - \tilde{\zeta}_j\|_2^2 \,, \tag{2}$$

where $D_{\tilde{\mathrm{Z}}}$ and D_{Z} denote the length of the diagonal of the bounding box of $\tilde{\mathrm{Z}}$ and Z, respectively. The obtained Chamfer distances further are normalized through a function $\Phi_k(d_{CD}) = e^{-k d_{CD}}$, that maps a distance d_{CD} to a score in $[0,1]$, where the parameter k is chosen according to conducted baselines. The two unidirectional scores are finally averaged to obtain a single edge recovery score

$$S_e(\tilde{\mathbf{e}}, \mathbf{e}) = \frac{1}{2}(\Phi_k(d_{CD}(\tilde{\mathrm{Z}}, \mathrm{Z})) + \Phi_k(d_{CD}(\mathrm{Z}, \tilde{\mathrm{Z}}))) \,. \tag{3}$$

Note that accurately predicted edges close to the ground truth are expected to have a high edge recovery score S_e.

Edge length Score: This score quantifies the similarity between the total edge length of the prediction $\tilde{\mathbf{e}}$ and that of the ground truth $\mathbf{e}$. The length of the predicted and ground truth edges denoted as $\tilde{L}$ and L, respectively, are computed by summing over the lengths of all edges in $\tilde{\mathbf{e}}$ and $\mathbf{e}$. Note that for splines, the length is estimated by densely sampling points on the spline and then accumulating the consecutive point distances. The normalized length score is given by

$$S_l(\tilde{\mathbf{e}}, \mathbf{e}) = 1 - |\frac{1 - (\tilde{L}/L)}{1 + (\tilde{L}/L)}| \,, \tag{4}$$

Note that this score has a range of $[0,1]$. Predicted edges with accurate length estimations are expected to have high length scores S_l.

Sharpness Score: The sharpness estimation task is formulated as a binary classification problem where each predicted edge $\tilde{\mathbf{e}}_j \in \tilde{e}$ is classified as either sharp ($\tilde{s}_j = 1$) or not-sharp ($\tilde{s}_j = 0$). Since ground truth sharpness is given as a continuous value (ranging in $[0, 2\pi]$), we use a threshold of 1.5, above which an edge is considered sharp. The sharpness score is defined as a weighted accuracy of the predicted sharpness scores $\tilde{\mathbf{s}}$ with respect to the ground truth $\mathbf{s}$. In practice, similarly to the edge recovery score, we sample two 3D point sets $\tilde{\mathrm{Z}} = \{\tilde{\boldsymbol{\zeta}}_i\}_{i=1}^{\tilde{N}_\zeta}$ and $\mathrm{Z} = \{\boldsymbol{\zeta}_i\}_{i=1}^{N_\zeta}$ for the predicted and ground truth edges, respectively. Since the sharpness label is defined per edge, this label is transferred to the 3D points that form that edge. This results in a set of ground truth sharpness labels $\{\sigma_i\}_{i=1}^{i=N_\zeta} \in \{0, 1\}$, where each σ_i denotes the sharpness label of the point $\boldsymbol{\zeta}_i$. Similarly to the ground truth, another set of predicted sharpness labels $\{\tilde{\sigma}_i\}_{i=1}^{i=\tilde{N}_\zeta} \in \{0, 1\}$ is produced from the predicted edge sharpness labels $\tilde{\mathbf{s}}$. The sharpness score is then given by the following weighted accuracy,

$$S_s = \frac{1}{\tilde{N}_\zeta} \sum_{i=1}^{\tilde{N}_\zeta} \Phi_k(\|\tilde{\boldsymbol{\zeta}}_i - \boldsymbol{\zeta}_{\Gamma(i)}\|_2^2) \cdot \mathbb{1}(\tilde{\sigma}_i = \sigma_{\Gamma(i)}) \;, \quad (5)$$

where $\Gamma(i)$ matches an index i of a point $\tilde{\boldsymbol{\zeta}}_i$ in the predicted edges to the index of the closest point in the ground truth edges in the sense of Euclidean distance. Φ_k is the same mapping function used in the edge recovery score and $\mathbb{1}(.)$ is an indicator function.

Final Score: The final score of *Track 1* is a combination of the three scores mentioned above. For each sample, it is computed as the average of the edge recovery score S_e, the length score S_l, and the sharpness Score S_s, or $S_{track1} = \frac{1}{3}(S_e + S_l + S_s)$.

Evaluation Metrics for Track 2. The goal of Tracks 2 is to predict the B-Rep face membership and the types for each point of the scan. Accordingly, we evaluate the predictions following two criteria that are described below.

Face Membership Score: Given a scan $\mathbf{X}$, the predicted face membership $\tilde{\mathbf{M}} \in \{0, 1\}^{N \times \tilde{N}_f}$ is evaluated with respect to the ground truth face membership $\mathbf{M} \in \{0, 1\}^{N \times N_f}$ using Intersection Over Union (IoU). Note that the number of predicted face memberships $\tilde{N}_f$ can be different from the one of the ground truth N_f. Moreover, the evaluation of the face membership segmentation task requires addressing the inherent ambiguity in face membership labelling, as predicted face labels do not necessarily have a predefined match with ground truth class labels. To handle this issue, we use the Hungarian matching algorithm [14] to perform optimal matching between predicted face memberships and ground truth face memberships. Hungarian matching is able to find the best one-to-one correspondence that maximizes the total IoU across all matched pairs. This results in the following face membership score,

$$S_m = \frac{1}{N_f} \sum_{i=1}^{N_f} IoU(\Lambda(\tilde{\mathbf{M}})_{:,i}, \mathbf{M}_{:,i}) \;, \quad (6)$$

where $\Lambda()$ is the reordering from Hungarian matching.

Face Type Score: Similarly to the face membership score, the predicted face types $\tilde{\mathbf{T}} \in \{0, 1\}^{N \times N_{f_t}}$ is evaluated with respect to the ground truth face types $\mathbf{T} \in \{0, 1\}^{N \times N_{f_t}}$ using IOU. However, in contrast to the face membership scenario where a Hungarian matching was necessary, the IoU is directly computed for the face types to yield the following face type score,

$$S_t = \frac{1}{N_{f_t}} \sum_{i=1}^{N_{f_t}} IoU(\tilde{\mathbf{T}}_{:,i}, \mathbf{T}_{:,i}) \;. \quad (7)$$

Final Score: The final score for each sample is the average of the face membership score S_m and the face type score S_t and is given by $S_{track2} = \frac{1}{2}(S_m + S_t)$.

Evaluation Metrics for Track 3. As in *Track 2*, the predicted CAD operation steps and types are evaluated following two criteria.

CAD Step Score: Given a scan $\mathbf{X}$, the predicted face membership $\tilde{\mathbf{S}} \in \{0, 1\}^{N \times \tilde{N}_s}$ is evaluated with respect to the ground truth face membership $\mathbf{S} \in \{0, 1\}^{N \times N_s}$ using IoU. As the objective is to predict ordered steps, Hungarian matching is not used and the CAD step score is given by

$$S_{st} = \frac{1}{N_s} \sum_{i=1}^{N_s} IoU(\tilde{\mathbf{S}}_{:,i}, \mathbf{S}_{:,i}) \;. \quad (8)$$

CAD Type Score: As done for the face types of Track 2, the predicted CAD operation types $\tilde{\mathbf{O}} \in \{0, 1\}^{N \times N_{O_t}}$ is evaluated with respect to the ground truth face types $\mathbf{O} \in \{0, 1\}^{N \times N_{O_t}}$ using the following IoU based score

$$S_{ot} = \frac{1}{N_t} \sum_{i=1}^{N_t} IoU(\tilde{\mathbf{O}}_{:,i}, \mathbf{O}_{:,i}) \;. \quad (9)$$

Final Score: The final score for each sample is the average of the CAD step score S_{st} and the CAD type score S_{ot} and is given by $S_{track3} = \frac{1}{2}(S_{st} + S_{ot})$.

5. Results

The proposed baseline methods for all three tracks are evaluated on the dedicated test partition. Results are shown in Table 1 (*left*) and are also given on the online challenge leaderboard (hosted on the Codalab platform[4,5,6]). It is essential to note that the performance of the evaluated baselines is not particularly robust, with a reported final score of 0.34 for *Track 1*, and *Track 3* and 0.32 for *Track 2*; We highlight that the primary motivation of this experimental analysis is to establish a reference point on baseline performance

	Track 1				Track 2			Track 3		
	S_e	S_l	S_s	S_{track1}	S_m	S_t	S_{track2}	S_{st}	S_{ot}	S_{track3}
Baseline	0.38	0.18	0.46	**0.34**	0.28	0.35	**0.32**	0.29	0.39	**0.34**

	Consistency
Face Type (*Track 2*)	0.79
Operation Type (*Track 3*)	0.90

Table 1: *(left)* Performance of our proposed baselines on the dedicated test partition for all three tracks of the challenge. *(right)* Consistency as the percentage of vertices sharing the same face membership / operation step also having the same type. As in [9] sub-operation types are grouped into a single type, for example, *extrude.end* and *extrude.side*, into a single *extrude*.

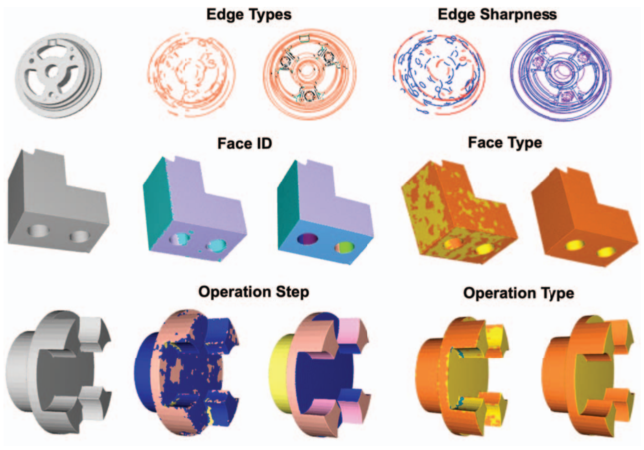

Figure 7: Qualitative results for proposed baselines for the three tracks of the challenge (one row per track). Model prediction (*left*) is contrasted to the ground truth labels (*right*). Colour labelling as in Figure 2.

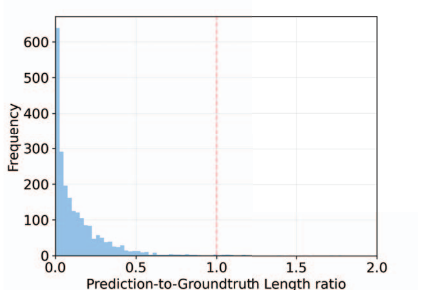

Figure 8: Histogram of *Prediction-to-Groundtruth* length ratios $\tilde{L}/L$ across test samples.

for all tracks, rather than striving to set new track records. Consequently, the baselines were not subject to optimization. For instance, we did not specifically address issues such as class imbalance, undertake extensive hyperparameter tuning or utilise the adaptive sampling scheme of [7] for enhanced edge detection. For qualitative results on Figure 7, we observe that our baseline model tends to segment larger circles into fragmented sets of shorter edges, cluster face IDs together and often conflates distinct operation steps. These findings underscore the extent of the difficulty of the challenge and highlight potential areas for improvements in future iterations.

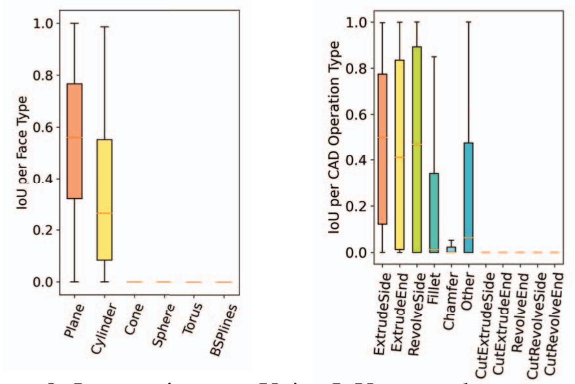

Figure 9: Intersection over Union IoU reported per type, for face types (*Track 2*) and operation types (*Track 3*).

To provide additional insights on model performance, we present a histogram of *prediction-to-groundtruth* length ratios in Figure 8. We find that our baseline consistently underestimates edge lengths thus limiting performance. In Figure 9, we report per-type IoU for face types (*Track 2*) and operation types (*Track 3*). Our baseline struggles to capture less common types in both cases due to a significant imbalance in class frequency (as also shown in Figure 4 and Figure 5). Finally, we follow [9] and report face and step prediction consistency in Table 1 *(right)* as the percentage of vertices sharing the same face membership / operation step also having the same type. We identify that future improvements in terms of consistency (we report 0.79 for *Track 2* and 0.90 for *Track 3*) can positively affect performance.

6. Conclusion

In this paper, we introduce the SHARP challenge 2023, aiming to address the nuances of the Scan-to-CAD problem through three distinct tracks. For every track, a new version of the challenging CC3D dataset is presented, along with an exhaustive description of the evaluation metrics and proposed baseline methodologies. This challenge is designed to encourage forthcoming advancements in reverse engineering from 3D scans in a real-world setting.

Acknowledgement: Present work is supported by the National Research Fund, Luxembourg under the BRIDGES2021/IS/16849599/FREE-3D and IF/17052459/CASCADES projects, and by Artec 3D.

References

[1] Open CASCADE Technology Documentation — dev.opencascade.org. https://dev.opencascade.org/doc/overview/html/. [Accessed 26-07-2023]. 3, 4

[2] Professional 3D Scanners — Artec 3D — Scanning Solutions — artec3d.com. https://www.artec3d.com. [Accessed 31-07-2023]. 2

[3] 3d Content Central. https://www.3dcontentcentral.com/. 1, 2

[4] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*. PMLR, 2018. 6

[5] Kseniya Cherenkova, Djamila Aouada, and Gleb Gusev. Pvdeconv: Point-voxel deconvolution for autoencoding cad construction in 3d. In *2020 IEEE International Conference on Image Processing (ICIP)*, 2020. 1, 2, 4

[6] Kseniya Cherenkova, Elona Dupont, Anis Kacem, Ilya Arzhannikov, Gleb Gusev, and Djamila Aouada. Sepicnet: Sharp edges recovery by parametric inference of curves in 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2, 3

[7] Kseniya Cherenkova, Elona Dupont, Anis Kacem, Ilya Arzhannikov, Gleb Gusev, and Djamila Aouada. Sepicnet: Sharp edges recovery by parametric inference of curves in 3d shapes. *ArXiv*, 2023. 5, 8

[8] cvi2. https://cvi2.uni.lu/sharp2022/challenge2/. 3

[9] Elona Dupont, Kseniya Cherenkova, Anis Kacem, Sk Aziz Ali, Ilya Arzhannikov, Gleb Gusev, and Djamila Aouada. Cadops-net: Jointly learning cad operation types and steps from boundary-representations. *arXiv preprint arXiv:2208.10555*, 2022. 1, 2, 4, 8

[10] Haoxiang Guo, Shilin Liu, Hao Pan, Yang Liu, Xin Tong, and Baining Guo. Complexgen: Cad reconstruction by b-rep chain complex generation. *ACM Transactions on Graphics (TOG)*, 2022. 2

[11] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Training deep networks with structured layers by matrix backpropagation. *ArXiv*, 2015. 5

[12] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *CVPR*, 2017. 1

[13] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 1, 2

[14] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. 6, 7

[15] Joseph George Lambourne, Karl Willis, Pradeep Kumar Jayaraman, Longfei Zhang, Aditya Sanghi, and Kamal Rahimi Malekshan. Reconstructing editable prismatic cad from rounded voxel models. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. 2

[16] Joseph G Lambourne, Karl DD Willis, Pradeep Kumar Jayaraman, Aditya Sanghi, Peter Meltzer, and Hooman Shayani. Brepnet: A topological message passing system for solid models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 4

[17] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J. Mitra. Sketch2cad: Sequential cad modeling by sketching in context. *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2020)*, 2020. 1, 2

[18] Changjian Li, Hao Pan, Adrien Bousseau, and Niloy J Mitra. Free2cad: parsing freehand drawings into cad commands. *ACM Transactions on Graphics (TOG)*, 2022. 1

[19] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas J Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2, 6

[20] Pu Li, Jianwei Guo, Xiaopeng Zhang, and Dong-Ming Yan. Secad-net: Self-supervised cad reconstruction by learning sketch-extrude operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[21] Yuanqi Li, Shun Liu, Xinran Yang, Jianwei Guo, Jie Guo, and Yanwen Guo. Surface and edge detection for primitive fitting of point clouds. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 2

[22] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, 2019. 5

[23] Albert Matveev, Ruslan Rakhimov, Alexey Artemov, Gleb Bobrovskikh, Vage Egiazarian, Emil Bogomolov, Daniele Panozzo, Denis Zorin, and Evgeny Burnaev. Def: Deep estimation of sharp geometric features in 3d shapes. *ACM Transactions on Graphics (TOG)*, 2022. 1, 2

[24] Onshape. https://www.onshape.com/. 1

[25] Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P Adams. Vitruvion: A generative model of parametric cad sketches. *arXiv preprint arXiv:2109.14124*, 2021. 2

[26] Gopal Sharma, Difan Liu, Evangelos Kalogerakis, Subhransu Maji, Siddhartha Chaudhuri, and Radom'ir Mvech. Parsenet: A parametric surface fitting network for 3d point clouds. *ArXiv*, 2020. 5

[27] Gopal Sharma, Difan Liu, Subhransu Maji, Evangelos Kalogerakis, Siddhartha Chaudhuri, and Radomír Měch. Parsenet: A parametric surface fitting network for 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*. Springer, 2020. 2

[28] Mikaela Angelina Uy, Yen-Yu Chang, Minhyuk Sung, Purvi Goel, Joseph G Lambourne, Tolga Birdal, and Leonidas J Guibas. Point2cyl: Reverse engineering 3d objects from point clouds to extrusion cylinders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2

[29] Karl DD Willis, Pradeep Kumar Jayaraman, Joseph G Lambourne, Hang Chu, and Yewen Pu. Engineering sketch generation for computer-aided design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[30] Karl DD Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. *ACM Transactions on Graphics (TOG)*, 2021. 1, 2

[31] Rundi Wu, Chang Xiao, and Changxi Zheng. Deepcad: A deep generative network for computer-aided design models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 2, 4

[32] Xiang Xu, Pradeep Kumar Jayaraman, Joseph G Lambourne, Karl DD Willis, and Yasutaka Furukawa. Hierarchical neural coding for controllable cad model generation. 2023. 2

[33] Xianghao Xu, Wenzhe Peng, Chin-Yi Cheng, Karl DD Willis, and Daniel Ritchie. Inferring cad modeling sequences using zone graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021. 2

[34] Xiang Xu, Karl DD Willis, Joseph G Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. Skexgen: Autoregressive generation of cad construction sequences with disentangled codebooks. 2022. 2

[35] Xiangyu Zhu, Dong Du, Weikai Chen, Zhiyou Zhao, Yinyu Nie, and Xiaoguang Han. Nerve: Neural volumetric edges for parametric curve extraction from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2