FUSION 360 GALLERY: A DATASET AND ENVIRONMENT FOR PROGRAMMATIC CAD RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Parametric computer-aided design (CAD) is a standard paradigm used for the design of manufactured objects. CAD designers perform modeling operations, such as *sketch* and *extrude*, to form a construction sequence that makes up a final design. Despite the pervasiveness of parametric CAD and growing interest from the research community, a dataset of human designed 3D CAD construction sequences has not been available to-date. In this paper we present the *Fusion 360 Gallery* reconstruction dataset and environment for learning CAD reconstruction. We provide a dataset of 8,625 designs, comprising sequential *sketch* and *extrude* modeling operations, together with a complementary environment called the *Fusion 360 Gym*, to assist with performing CAD reconstruction. We outline a standard CAD reconstruction task, together with evaluation metrics, and present results from a novel method using neurally guided search to recover a construction sequence from raw geometry.

1 INTRODUCTION

The manufactured objects that surround us in everyday life are created in computer-aided design (CAD) software using common modeling operations such as *sketch* and *extrude*. With just these two modeling operations, a highly expressive range of 3D designs can be created (Figure 1). Parametric CAD files contain construction sequence information that is critical for documenting design intent, maintaining editablity, and downstream simulation and manufacturing. Despite the value of this information, it is often lost due to data translation or error and must be reverse engineered from raw geometry or even 3D scan data. Although the task of reconstructing CAD operations from raw geometry has been pursued for over 40 years (Shah et al., 2001), recent advances in neural networks for 3D shape generation has spurred new interest in CAD reconstruction. However, learning-based approaches to CAD reconstruction have not yet had access to a human-designed dataset of 3D CAD construction sequences, instead relying on synthetic data for both training and testing purposes, e.g.

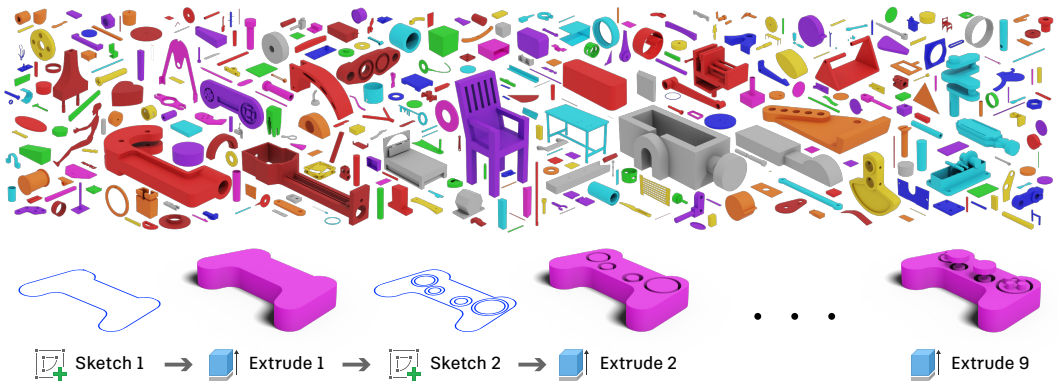

Figure 1: Top: A subset of designs containing 3D CAD construction sequences from the *Fusion 360 Gallery* reconstruction dataset. Bottom: An example construction sequence using *sketch* and *extrude* modeling operations.

Li et al. (2020). The absence of real world data has limited work on CAD reconstruction using common sketch and extrude modeling operations. Instead a focus has been on reconstruction from simple geometric primitives (Sharma et al., 2017; Tian et al., 2019; Ellis et al., 2019) that lack the rich parametric sketches commonly used in mechanical CAD (e.g. Figure 2). As there is no existing learning-based approach to reconstruct 'sketch and extrude' sequences, we take a first step towards this goal by introducing data, a supporting software environment, and a novel CAD reconstruction method for 'sketch and extrude' designs.

In this paper we present the *Fusion 360 Gallery* reconstruction dataset and environment for learning CAD reconstruction. The dataset contains 8,625 designs created by users of Autodesk Fusion 360 using a simple subset of CAD modeling operations: *sketch* and *extrude*. To the best of our knowledge this dataset is the first to provide human designed 3D CAD construction sequence data for use with machine learning. To support research with the dataset we provide an environment called the *Fusion 360 Gym* for working with CAD reconstruction. A key motivation of this work is to provide insights into the process of *how* people design objects. Furthermore, our goal is to provide a universal benchmark for research and evaluation of learning-based CAD reconstruction algorithms, bridging the gap between the computer graphics and machine learning community. To this end we describe a standard CAD reconstruction task and associated evaluation metrics with respect to the ground truth construction sequence. We also introduce a novel method for CAD reconstruction of 'sketch and extrude' designs using neurally guided search. This search employs a policy, trained using imitation learning, consisting of a graph neural network encoding of CAD geometry.

This paper makes the following contributions:

- We present the *Fusion 360 Gallery* reconstruction dataset containing construction sequence information for 8,625 human-designed 'sketch and extrude' CAD models.
- We introduce an environment called the *Fusion 360 Gym*, standardizing the CAD reconstruction task in a Markov Decision Process formulation.
- We introduce a novel method and results using neurally guided search on the CAD reconstruction task, contextualizing future research.

2 Related Work

CAD Datasets Existing 3D CAD datasets have largely focused on providing mesh geometry (Chang et al., 2015; Wu et al., 2015; Zhou & Jacobson, 2016; Mo et al., 2019b; Kim et al., 2020). However, the de facto standard for Parametric CAD is the boundary representation (B-Rep) format, containing valuable analytic representations of surfaces and curves suitable for high level control of 3D shapes. B-Reps are collections of trimmed parametric surfaces along with topological information which describes adjacency relationships and the ordering of elements such as faces, loops, edges, and vertices (Weiler, 1986). B-Rep datasets have recently been made available with both human-designed (Koch et al., 2019) and synthetic data (Zhang et al., 2018; Jayaraman et al., 2020; Starly, 2020). Missing from these datasets is construction sequence information. We believe it is critical to understand not only *what* is designed, but *how* that design came about.

Parametric CAD files contain valuable information on the history of a design, revealing how the design was constructed. Schulz et al. (2014) provide a standard collection of human designs with full parametric history, albeit a limited set of 67 designs in a proprietary format. A large challenge is translating the daunting number of modeling operations contained within proprietary parametric CAD files into data suitable for machine learning. SketchGraphs (Seff et al., 2020) constrains the broad area of parametric CAD by focusing on the underlying 2D engineering sketches, including sketch construction sequences. In the absence of 3D human design data, learning-based approaches have instead leveraged synthetic CAD construction sequences (Sharma et al., 2017; Li et al., 2020). The dataset presented in this paper is, to the best of our knowledge, the first to provide human-designed 3D CAD construction sequence information suitable for use with machine learning.

CAD Reconstruction The task of CAD reconstruction involves recovering the sequence of modeling operations used to construct a CAD model from raw geometry input, such as triangle meshes, point clouds, or B-Reps. Despite extensive prior work (Shah et al., 2001), CAD reconstruction remains a challenging problem as it requires deductions on both continuous parameters (e.g., ex-

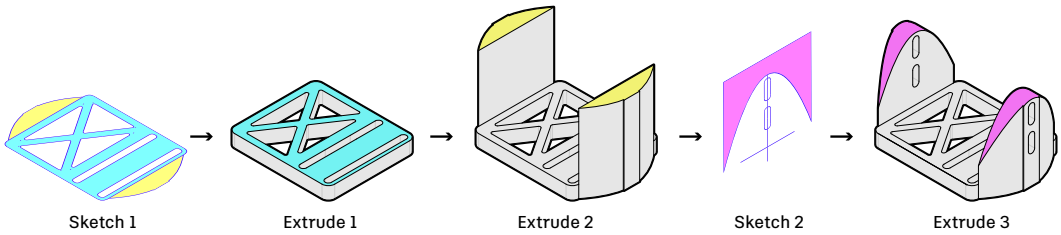

Figure 2: An example design sequence from the *Fusion 360 Gallery* reconstruction dataset. Sketch profiles are sequentially extruded to *join* (Extrude 1, Extrude 2) or *cut* (Extrude 3) geometry using built in boolean operations. The color coded areas show the sketch profiles that partake in each extrude operation.

tracting the dimensions of primitives) and discrete operations (e.g., choosing a proper operation for the next step), leading to a mixed combinatorial search space. To recover the sequence of operations, traditional methods typically run global search methods (e.g., evolutionary algorithms as in Hamza & Saitou (2004), Weiss (2009), Friedrich et al. (2019), and Fayolle & Pasko (2016)) with heuristic rules to prune the search space (Shapiro & Vossler, 1993; Buchele, 2000; Buchele & Roles, 2001; Buchele & Crawford, 2003). Heuristic approaches are also available in a number of commercial software tools, often as a user-guided semi-automatic system (Autodesk, 2012). Another approach is using program synthesis (Du et al., 2018; Nandi et al., 2017; 2018; 2020) to infer CAD programs written in domain specific languages from given shapes. CAD reconstruction is also related to the inverse procedural modeling problem (Talton et al., 2011; Stava et al., 2014; Vanegas et al., 2012), which attempts to reverse-engineer procedures that can faithfully match a given target. Inverse procedural modeling methods typically work with synthetic data, while our paper tackles tasks on real CAD models and operations.

Compared to the rule-based or grammar-based methods above, learning-based approaches can potentially learn the types of rules that are typically hard-coded, automate scenarios that require user-input, and generalize when confronted with unfamiliar geometry. One of the earliest such works is CSGNet (Sharma et al., 2017), which trains a neural network to infer the sequence of Constructive Solid Geometry (CSG) operations based on visual inputs. More recent works along this line of research include Ellis et al. (2019), Tian et al. (2019), and Kania et al. (2020). Typically associated with these methods are a customized, domain specific language (e.g., CSG) that parameterizes the space of geometry, some heuristic rules that limit the search space, and a neural network generative model (Zou et al., 2017; Mo et al., 2019a; Chen et al., 2020; Jones et al., 2020). Our approach is, to the best of our knowledge, the first to apply a learning-based method to reconstruction using common *sketch* and *extrude* CAD modeling operations from real human designs.

3 Fusion 360 Gallery Reconstruction Dataset

The *Fusion 360 Gallery* reconstruction dataset[1] is produced from designs submitted by users of the CAD software Autodesk Fusion 360. The dataset contains CAD construction sequence information from a subset of 'sketch and extrude' designs. We intentionally limit the data to the *sketch* and *extrude* modeling operations to reduce the complexity of the CAD reconstruction task. Figure 1 shows a random sampling of the designs in the dataset. Each design is provided in three different representations: B-Rep, mesh, and construction sequence JSON text format. An official 80:20 train-test split is provided with 6,900 and 1,725 designs respectively. We now briefly outline the *sketch* and *extrude* modeling operations and provide additional details in Section A.1 of the appendix.

3.1 Sketch

Unlike free-form sketches, sketches in CAD are composed of 2D geometric primitives (lines, circles, splines etc.), associated dimensions (distance, diameter, angle etc.) and constraints (symmetry, tangent, parallel etc.). Sketch geometry is represented by points, that create curves, that in turn form loops within profiles. The intersection of curves, as the user draws, automatically creates closed

[1] https://github.com/xxxx/Fusion360GalleryDataset

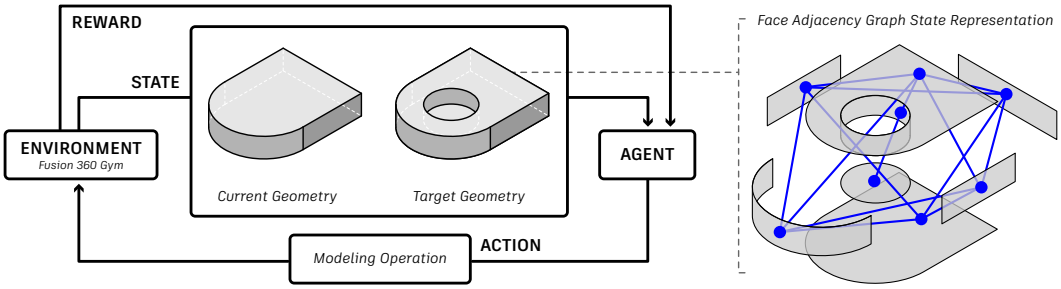

Figure 3: The *Fusion 360 Gym* interacts with an *agent* in a sequential decision making scenario (left) with the *state* containing geometries represented as face adjacency graphs (right).

loops and profiles that are serialized as both raw curves and trimmed profiles. Sketch profiles form the basis for 3D extrusion as shown in (Figure 2). The as-designed ordering of sketch operations is not stored in the native design files, however a consistent ordering can be derived by traversing the sketch profiles in sequence.

3.2 Extrude

An extrude operation takes one or more sketch profiles and extrudes them from 2D into a 3D B-Rep body. A distance parameter defines how far the profile is extruded. A notable feature of extrude operations in Fusion 360 is the ability to perform Boolean operations in the same step. As a user extrudes a sketch profile, they choose to create a *new body*, *join*, *cut*, or *intersect* with other bodies in the design (Figure 2). Additional extrude options are available such as two-sided extrude, symmetrical extrude, and tapered extrude. The combination of expressive sketches and extrude operations with built in Boolean capability enables a wide variety of designs to be constructed.

4 Fusion 360 Gym

Together with the dataset we provide an open source environment, called the *Fusion 360 Gym*, for standardizing the CAD reconstruction task. The *Fusion 360 Gym* wraps the underlying Fusion 360 Python API (Autodesk, 2014) and serves as the environment that interacts with an intelligent agent for the task of CAD reconstruction (Figure 3). Specifically, the *Fusion 360 Gym* formalizes the following Markov Decision Process:

- **state**: Contains the current geometry, and optionally, the target geometry to be reconstructed. We use a B-Rep face-adjacency graph as our state representation.
- **action**: A modeling operation that allows the agent to modify the current geometry. We consider two action representations: sketch extrusion and face extrusion.
- **transition**: *Fusion 360 Gym* implements the transition function that applies the modeling operation to the current geometry.
- **reward**: The user can define custom reward functions depending on the task. We provide an implementation of intersection over union (IoU) as one measure to compare the current and target geometry.

4.1 State Representation

In order for an agent to successfully reconstruct the target geometry, it is important that we have a suitable state representation. In *Fusion 360 Gym*, we use a similar encoding scheme to Jayaraman et al. (2020) and represent the current and target geometry with a face-adjacency graph (Ansaldi et al., 1985), illustrated in Figure 3 (right). Crucial to this encoding are the *geometric* features of the elements, such as point-locations, and *topological* features specifying how these elements are connected to each other. Specifically, the vertices of the face-adjacency graph represent B-Rep faces (trimmed parametric surfaces) in the design, with graph vertex features representing the size,

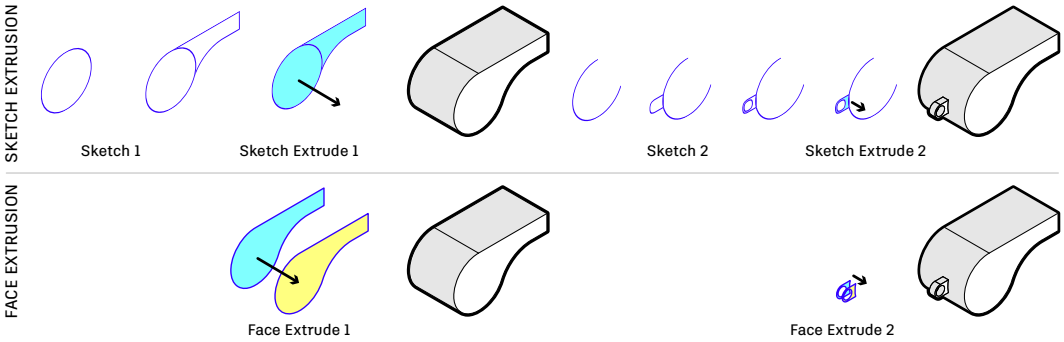

Figure 4: Action representations supported by the *Fusion 360 Gym* include low-level sketch extrusion (top) and simplified face extrusion (bottom).

orientation, and curvature of the faces. The edges of the face-adjacency graph represents B-Rep edges in the design, that connect the adjacent B-Rep faces to each other.

4.2 Action Representation

In *Fusion 360 Gym* we support action representations for two common patterns of modeling operations: *sketch extrusion* and *face extrusion*.

Sketch Extrusion In sketch extrusion, the agent must first select a sketch plane, draw on this plane using a sequence of curve primitives, such as lines and arcs, to form closed loop profiles. The agent then selects a profile to extrude a given distance and direction (Figure 4, top). Using this representation it is possible to construct novel geometries by generating the underlying sketch primitives and extruding them by an arbitrary amount. Although all designs in the *Fusion 360 Gallery* reconstruction dataset can be constructed using sketch extrusion, in practice this is challenging. Benko et al. (2002) show that to generate sketches suitable for mechanical engineering parts, the curve primitives often need to be constructed alongside a set of constraints which enforce regularities and symmetries in the design. Although the construction of the constraint graphs is feasible using techniques like the one shown by Liao et al. (2019), enforcing the constraints requires a complex interaction between the machine learning algorithm and a suitable geometric constraint solver, greatly increasing the algorithm complexity. We alleviate this problem by supporting another pattern of modeling operations called *face extrusion*.

Face Extrusion In face extrusion, a face from the target design is used as the extrusion profile rather than a sketch profile (Figure 4, bottom). This is possible because the target design is known in advance during reconstruction. An action a in this scheme is a triple $\{\text{face}_{start}, \text{face}_{end}, \text{op}\}$ where the start and end faces are parallel faces referenced from the target geometry, and the operation type is one of the following: *new body, join, cut, intersect*. Target constrained reconstruction using face extrusion has the benefit of narrowly scoping the prediction problem with shorter action sequences and simpler actions. Conversely, not all geometries can be reconstructed with this simplified strategy due to insufficient information in the target, e.g., Extrude 3 in Figure 2 cuts across the entire design without leaving a start or end face. Of the design sequences in the reconstruction dataset, 59.2% can be directly converted to a face extrusion sequence. We estimate that approximately 80% of designs in our dataset can be reconstructed by finding alternative construction sequences.

4.3 Synthetic Data Generation

The *Fusion 360 Gym* supports generation of semi-synthetic data by taking existing designs and modifying or recombining them. For instance, we can randomly perturb the sketches and the extrusion distances, and even 'graft' sketches from one design onto another. We also support distribution matching of parameters, such as the number of faces, to ensure that synthetic designs match a human-designed dataset distribution. Learning-based systems can leverage semi-synthetic data to expand the number of samples in the *Fusion 360 Gallery* reconstruction dataset. We provide examples of synthetic data and commands for the *Fusion 360 Gym* in Section A.2 of the appendix.

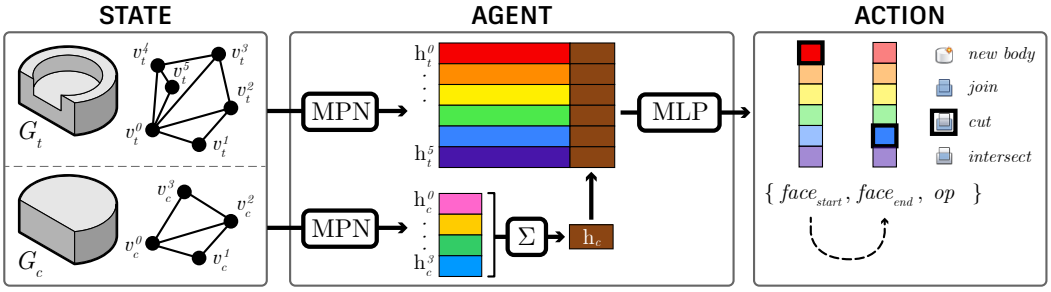

Figure 5: Given a *state* comprising the target geometry G_t and current geometry G_c, the *mpn agent* uses a message passing network (MPN) to predict an *action* as a face extrusion modeling operation.

5 CAD Reconstruction

5.1 Task

The goal of CAD reconstruction is to recover the sequence of modeling operations used to construct a CAD model with only the raw geometry as input. This task can be specified using different input geometry representations, including B-Rep, mesh, or point cloud, with progressively lower fidelity. Each representation presents a realistic scenario where parametric CAD information is absent and needs to be recovered. Given a target geometry G_t, we wish to find a sequence of CAD modeling operations (actions) $\mathcal{A} = \{a_0, a_1, \cdots\}$ that generates an output geometry G, such that every point in space is in its interior, if and only if, it is also in the interior of G_t.

Evaluation Metrics We prescribe three evaluation metrics, IoU, exact reconstruction, and conciseness. IoU measures the intersection over union of G and G_t: $\texttt{iou}(G, G_t) = |G \cap G_t|/|G \cup G_t|$. Exact reconstruction measures whether $\texttt{iou}(G, G_t) = 1$. As multiple correct sequences of CAD modeling operations exist, a proposed reconstruction sequence $\mathcal{A}$ need not match the ground truth sequence $\hat{\mathcal{A}}_t$ provided an exact reconstruction is found. Let $\texttt{conciseness}(\mathcal{A}, \hat{\mathcal{A}}_t) = |\mathcal{A}|/|\hat{\mathcal{A}}_t|$, where a score ≤ 1 indicates the agent found a reconstruction with equal or fewer steps, and a score > 1 indicates more inefficient reconstructions.

5.2 Method

We now present a method for CAD reconstruction using neurally-guided search (Ellis et al., 2019; Kalyan et al., 2018; Tang et al., 2019; Devlin et al., 2017) from *B-Rep input* using *face extrusion* modeling operations. Rather than discovering a sequence of construction by exploration, the agent is trained to match known reconstruction sequences present in the training set using imitation learning. We leverage search at inference time to recover the given target geometry.

Imitation Learning To perform imitation learning, we leverage the fact that we have the ground truth sequence of modeling operations (actions) $\hat{\mathcal{A}}_t = \{\hat{a}_{t,0} \cdots \hat{a}_{t,n-1}\}$ for each design G_t in the reconstruction dataset. We feed the ground truth action sequence $\hat{\mathcal{A}}_t$ into the *Fusion 360 Gym*, starting from the empty geometry G_0, and output a sequence of partial constructions $G_{t,1} \cdots G_{t,n}$ where $G_{t,n} = G_t$. We then collect the supervised dataset $\mathcal{D} = \{(G_0, G_t) \rightarrow \hat{a}_{t,0}, (G_{t,1}, G_t) \rightarrow \hat{a}_{t,1} \cdots\}$ and train a supervised agent π_θ that takes the pair of current-target constructions (G_c, G_t) to a modeling operation action a_c, which would transform the current geometry closer to the target. Formally, we optimize the expected log-likelihood of correct actions under the data distribution:

$$E_{(G_c,G_t)\sim\mathcal{D}}\left[\log \pi_\theta\Big(\hat{a}_c|\big(G_c, G_t\big)\Big)\right] \tag{1}$$

Agent The agent takes a pair of geometries (G_c, G_t) as state, and outputs the corresponding face-extrusion action $a = \{\text{face}_{start}, \text{face}_{end}, \text{op}\}$ (Figure 5). The two geometries G_c, G_t are given using a face-adjacency graph similar to Jayaraman et al. (2020), where the graph vertexes represent the faces of the geometry, with vertex features calculated from each face: 10×10 grid of 3D points,

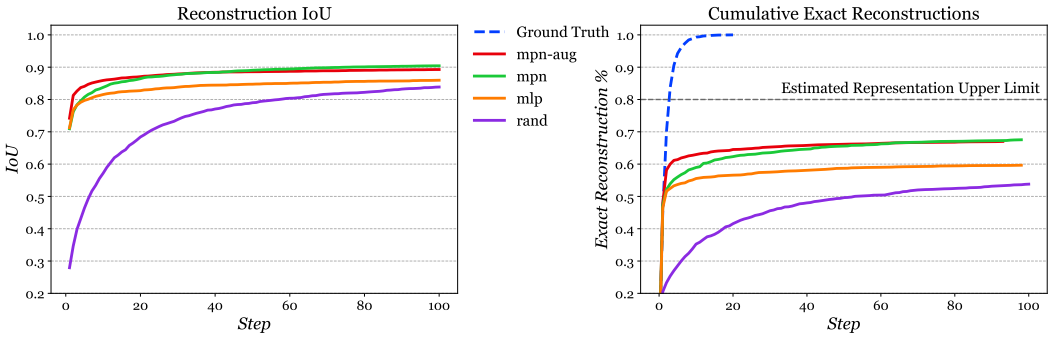

Figure 6: Reconstruction results over 100 search steps using random rollouts. For exact reconstructions, 0.8 is the estimated representation upper limit of the face extrusion action representation.

normals, and trimming mask, in addition to the face surface type. The edges simply signify connectivity of adjacent faces. The inputs are encoded using two *separate* message passing networks (MPN) (Kipf & Welling, 2016) aggregating messages along the edges of the graph. The encoded vectors representing the *current* geometry are summed together (h_c in Figure 5), and concatenated with the encoded vertexes of the target geometry ($h_t^0 \cdots h_t^5$). The concatenated vectors are used to output the action using a multi-layer perceptron (MLP), with an auto-regressive connection between the two target faces.

Search Given a neural agent $\pi_\theta(a|(G_c, G_t))$ capable of furthering a current geometry toward the target geometry, we can amplify its performance at test time using search. Here, we report the result of the most basic of search strategies: random rollout. We let the agent interact with the environment by sampling a sequence of actions according to π_θ up to a fixed rollout length of $\max(\frac{f_p}{2}, 2)$, where f_p is the number of planar faces in G_t. If the agent is successful at reconstructing the target, we stop. Otherwise, we repeat the process until we exhaust a global budget of 100 steps.

Data Augmentation We also study the effects of data augmentation by supplementing the designs in the training set with 10,063 synthetic designs. Each design uses existing sketches in the training set at a random orientation and applies two or more extrude operations to match the distribution of the number of faces in the training set as described in Section 4.3.

5.3 RESULTS

Using the test set of the *Fusion 360 Gallery* reconstruction dataset, we evaluate four different agents. The **rand** agent uniformly samples from the available actions to serve as a baseline without any learning. **mlp** is a simple MLP agent that does not take advantage of shape topology. **mpn** is the MPN agent. **mpn-aug** is the MPN agent trained on the standard training set augmented with additional synthetic designs. This set of agents forms an ablation study revealing the effects of different components of our approach.

For a target design G_t, the agent uses the random rollout search algorithm and attempts reconstruction over multiple rollouts. Each time the agent takes an action (a step) during search, we track the

Agent	**IoU**		**Exact Reconstruction %**		**Conciseness**
	20 Steps	100 Steps	20 Steps	100 Steps	
mpn-aug	**0.8707**	0.8928	**0.6452**	0.6701	**0.9706**
mpn	0.8644	**0.9042**	0.6232	**0.6754**	1.0168
mlp	0.8274	0.8596	0.5658	0.5965	0.9763
rand	0.6840	0.8386	0.4157	0.5380	1.2824

Table 1: Reconstruction results for IoU and exact reconstruction at 20 and 100 search steps using random rollouts. Conciseness is measured over exact reconstructions; lower values are better.

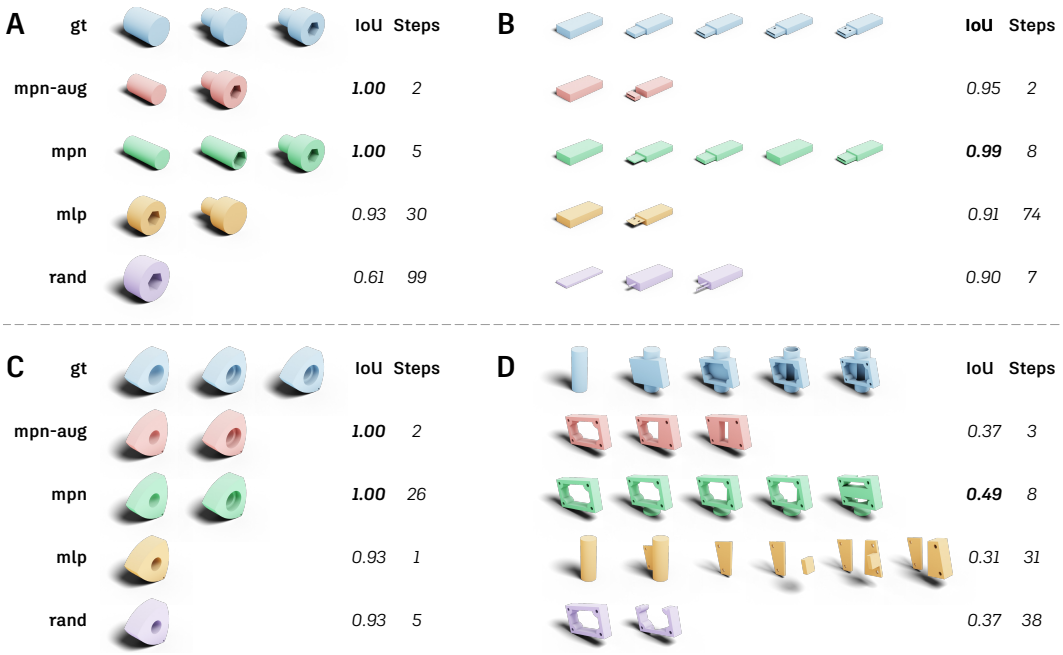

Figure 7: Qualitative construction sequence results comparing the ground truth (gt) to reconstructions using different agents with random search.

best IoU the agent has discovered so far, and whether exact reconstruction is achieved. We cap the total search budget to 100 steps. We report the evaluation metric of our agent as a function of the number of steps in Figure 6. We detail the exact results at step 20 and 100 in Table 1. Step 20 represents the point where it is possible to perform exact reconstructions for all designs in the test set. We also detail the conciseness of the recovered sequence for exact reconstructions. Figure 7 shows ground truth (gt) construction sequences compared with other agents using random search. The rollout with the highest IoU is shown with the IoU score and total search steps taken. Steps that don't change the geometry or occur after the highest IoU are omitted from the visualization.

Discussion We note that all neurally guided agents outperform the random agent baseline. The topology information available with a MPN is found to improve reconstruction performance, while data augmentation improves conciseness and reconstruction performance early in the search. We find that random rollout scales poorly with search budget and presents an opportunity for future research on alternate search procedures (additional search experiments are provided in Section A.3 of the appendix). For practical application of CAD reconstruction it is necessary to have an exact reconstruction where all details of a design are reconstructed in a concise way. It is notable that incorrect reconstructions can score well with the IoU metric, but omit important design details. We therefore suggest IoU should be a secondary metric, with future work focusing primarily on improving exact reconstruction performance with concise construction sequences.

6 Conclusion and Future Directions

In this paper we presented the *Fusion 360 Gallery* reconstruction dataset and environment for learning CAD reconstruction from sequential 3D CAD data. We outlined a standard CAD reconstruction task, together with evaluation metrics, and presented results from a neurally guided search approach. We envision a number of future directions that could leverage the reconstruction dataset: new representations for sequential geometry capable of performing CAD reconstruction and generation from B-Rep, mesh, point cloud, or image data; reinforcement learning approaches that mimic and improvise sequential modeling operations; and sketch and constraint synthesis from 3D geometry or images. Finally, beyond the simplified design space of *sketch* and *extrude* lies the full breadth of rich sequential CAD modeling operations.

REFERENCES

Silvia Ansaldi, Leila De Floriani, and Bianca Falcidieno. Geometric modeling of solid objects by using a face adjacency graph representation. *ACM SIGGRAPH Computer Graphics*, 19(3):131–139, 1985.

Autodesk. *Inventor Feature Recognition*, 2012. URL https://apps.autodesk.com/INVNTOR/en/Detail/Index?id=9172877436288348979.

Autodesk. *Fusion 360 API*, 2014. URL http://help.autodesk.com/view/fusion360/ENU/?guid=GUID-7B5A90C8-E94C-48DA-B16B-430729B734DC.

Pal Benko, Geza Kos, Pál Benkő, Laszlo Andor, Géza Kós, Tamas Varady, László Andor, and Ralph Martin. Constrained fitting in reverse engineering, 2002.

Suzanne F Buchele and Richard H Crawford. Three-dimensional halfspace constructive solid geometry tree construction from implicit boundary representations. In *Proceedings of the eighth ACM symposium on Solid modeling and applications*, pp. 135–144, 2003.

Suzanne F Buchele and Angela C Roles. Binary space partitioning tree and constructive solid geometry representations for objects bounded by curved surfaces. In *CCCG*, pp. 49–52. Citeseer, 2001.

Suzanne Fox Buchele. Three-dimensional binary space partitioning tree and constructive solid geometry tree construction from algebraic boundary representations. 2000.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 45–54, 2020.

Jacob Devlin, Jonathan Uesato, Surya Bhupatiraju, Rishabh Singh, Abdel-rahman Mohamed, and Pushmeet Kohli. Robustfill: Neural program learning under noisy i/o. *arXiv preprint arXiv:1703.07469*, 2017.

Tao Du, Jeevana Priya Inala, Yewen Pu, Andrew Spielberg, Adriana Schulz, Daniela Rus, Armando Solar-Lezama, and Wojciech Matusik. Inversecsg: Automatic conversion of 3d models to csg trees. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018.

Kevin Ellis, Maxwell Nye, Yewen Pu, Felix Sosa, Josh Tenenbaum, and Armando Solar-Lezama. Write, execute, assess: Program synthesis with a repl. In *Advances in Neural Information Processing Systems*, pp. 9169–9178, 2019.

Pierre-Alain Fayolle and Alexander Pasko. An evolutionary approach to the extraction of object construction trees from 3d point clouds. *Computer-Aided Design*, 74:1–17, 2016.

Markus Friedrich, Pierre-Alain Fayolle, Thomas Gabor, and Claudia Linnhoff-Popien. Optimizing evolutionary csg tree extraction. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1183–1191, 2019.

Karim Hamza and Kazuhiro Saitou. Optimization of constructive solid geometry via a tree-based multi-objective genetic algorithm. In *Genetic and Evolutionary Computation Conference*, pp. 981–992. Springer, 2004.

Pradeep Kumar Jayaraman, Aditya Sanghi, Joseph Lambourne, Thomas Davies, Hooman Shayani, and Nigel Morris. Uv-net: Learning from curve-networks and solids. *arXiv preprint arXiv:2006.10211*, 2020.

R. Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy Mitra, and Daniel Ritchie. Shapeassembly: Learning to generate programs for 3d shape structure synthesis. *ACM Transactions on Graphics (TOG), Siggraph Asia 2020*, 39(6):Article 234, 2020.

Ashwin Kalyan, Abhishek Mohta, Oleksandr Polozov, Dhruv Batra, Prateek Jain, and Sumit Gulwani. Neural-guided deductive search for real-time program synthesis from examples. *arXiv preprint arXiv:1804.01186*, 2018.

Kacper Kania, Maciej Zieba, and Tomasz Kajdanowicz. Ucsg-net–unsupervised discovering of constructive solid geometry tree. *arXiv preprint arXiv:2006.09102*, 2020.

Sangpil Kim, Hyung-gun Chi, Xiao Hu, Qixing Huang, and Karthik Ramani. A large-scale annotated mechanical components benchmark for classification and retrieval tasks with deep neural networks. In *Proceedings of 16th European Conference on Computer Vision (ECCV)*, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

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 Conference on Computer Vision and Pattern Recognition*, pp. 9601–9611, 2019.

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)*, 39(6): 164:1–164:14, 2020. doi: https://doi.org/10.1145/3414685.3417807.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In *Advances in Neural Information Processing Systems*, pp. 4255–4265, 2019.

Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J. Mitra, and Leonidas J. Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Trans. Graph.*, 38(6), November 2019a. ISSN 0730-0301. doi: 10.1145/3355089.3356527.

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 909–918, 2019b.

Chandrakana Nandi, Anat Caspi, Dan Grossman, and Zachary Tatlock. Programming language tools and techniques for 3d printing. In *2nd Summit on Advances in Programming Languages (SNAPL 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

Chandrakana Nandi, James R Wilcox, Pavel Panchekha, Taylor Blau, Dan Grossman, and Zachary Tatlock. Functional programming for compiling and decompiling computer-aided design. *Proceedings of the ACM on Programming Languages*, 2(ICFP):1–31, 2018.

Chandrakana Nandi, Max Willsey, Adam Anderson, James R. Wilcox, Eva Darulova, Dan Grossman, and Zachary Tatlock. Synthesizing structured cad models with equality saturation and inverse transformations. In *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp. 31–44, 2020.

Adriana Schulz, Ariel Shamir, David I. W. Levin, Pitchaya Sitthi-Amorn, and Wojciech Matusik. Design and fabrication by example. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2014)*, 33(4), 2014.

Ari Seff, Yaniv Ovadia, Wenda Zhou, and Ryan P. Adams. Sketchgraphs: A large-scale dataset for modeling relational geometry in computer-aided design. In *ICML 2020 Workshop on Object-Oriented Learning*. 2020.

Jami J Shah, David Anderson, Yong Se Kim, and Sanjay Joshi. A discourse on geometric feature recognition from cad models. *J. Comput. Inf. Sci. Eng.*, 1(1):41–51, 2001.

Vadim Shapiro and Donald L Vossler. Separation for boundary to csg conversion. *ACM Transactions on Graphics (TOG)*, 12(1):35–55, 1993.

Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. Csgnet: Neural shape parser for constructive solid geometry. corr abs/1712.08290 (2017). *arXiv preprint arXiv:1712.08290*, 2017.

Binil Starly. *FabWave - 3D Part Repository*, 2020. URL https://www.dimelab.org/fabwave.

O. Stava, S. Pirk, J. Kratt, B. Chen, R. Mundefinedch, O. Deussen, and B. Benes. Inverse procedural modelling of trees. *Comput. Graph. Forum*, 33(6):118–131, September 2014. ISSN 0167-7055. doi: 10.1111/cgf.12282. URL https://doi.org/10.1111/cgf.12282.

Jerry O. Talton, Yu Lou, Steve Lesser, Jared Duke, Radomír Měch, and Vladlen Koltun. Metropolis procedural modeling. *ACM Trans. Graph.*, 30(2), April 2011. ISSN 0730-0301. doi: 10.1145/1944846.1944851. URL https://doi.org/10.1145/1944846.1944851.

Yunhao Tang, Shipra Agrawal, and Yuri Faenza. Reinforcement learning for integer programming: Learning to cut. *arXiv preprint arXiv:1906.04859*, 2019.

Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. Learning to infer and execute 3d shape programs. In *International Conference on Learning Representations*, 2019.

Carlos A. Vanegas, Ignacio Garcia-Dorado, Daniel G. Aliaga, Bedrich Benes, and Paul Waddell. Inverse design of urban procedural models. *ACM Trans. Graph.*, 31(6), November 2012. ISSN 0730-0301. doi: 10.1145/2366145.2366187. URL https://doi.org/10.1145/2366145.2366187.

K.J. Weiler. *Topological structures for geometric modeling*. Technical report RPI, Center for Interactive Computer Graphics. University Microfilms, 1986.

Daniel Weiss. *Geometry-based structural optimization on CAD specification trees*. PhD thesis, ETH Zurich, 2009.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Zhibo Zhang, Prakhar Jaiswal, and Rahul Rai. Featurenet: machining feature recognition based on 3d convolution neural network. *Computer-Aided Design*, 101:12–22, 2018.

Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 900–909, 2017.