Learnable Geometry and Connectivity Modelling of BIM Objects

Haritha Jayasinghe
jmhaj2@cam.ac.uk
Ioannis Brilakis
ib340@cam.ac.uk

Department of Engineering
University of Cambridge
Cambridge, UK

Abstract

Accurate modelling of object geometries and their connectivity is a critical yet often overlooked aspect of the 3D scan to Building Information Model (BIM) pipeline. It is essential for the extraction of high-level structural information of infrastructure. In this paper, we first propose a novel method for parametric modelling of both primitive and non-primitive geometries. Element models are generated from predictions using a differentiable method, enabling both the integration of fitting error into the loss function, as well as further optimisation of predictions using gradient descent. This eliminates the need for custom distance heuristics, allowing for scalability to any object with parametric geometry. We evaluate our method on a novel benchmark and demonstrate that it accurately predicts model parameters despite the presence of occlusions. Moreover, we validate the utility of the extracted parameters by adopting them to infer connectivity between objects in a scan. This is achieved by framing connectivity inference as a link prediction task on a Graph Neural Network (GNN). This method learns the underlying nature of connectivity relationships within a BIM model using model parameters, and significantly outperforms rule-based methods for connectivity inference. Furthermore, we release a new synthetic dataset of industrial facility BIM element scans [1].

1 Introduction

Digital models provide sorely needed documentation to ensure optimal maintenance and breakdown prevention of ageing infrastructure in a wide variety of domains such as industrial facilities, roads, railways and buildings. The high time and labour cost of manual modelling has led to significant interest in automation of infrastructure modelling. The current approach for this, known as "Scan-to-BIM" [30] consists of the following steps: (1) raw data collection, (2) data preparation, (3) geometric modelling, and (4) semantic enrichment of the model. This paper focuses on the latter two steps.

Geometric modelling involves the generation of geometrically accurate models of objects of interest from 3D scans. Traditionally, this was achieved through geometric primitive fitting techniques such as RANSAC [25]. However, these methods are limited to a limited number of geometric primitives such as planes, cylinders, spheres and cones. Thus, they

[1] Code and data available at https://github.com/haritha-j/industrial-facility-relationships

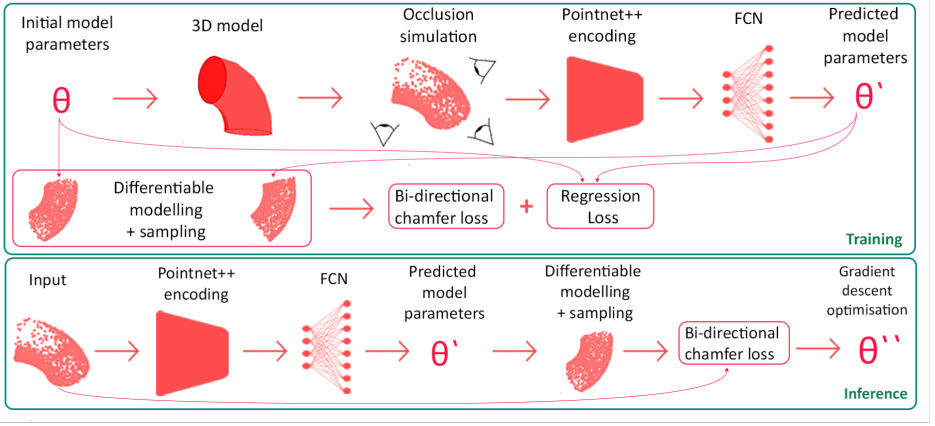

Figure 1: Our parametric modelling method consists of parameter regression, followed by fine-tuning through gradient descent over chamfer loss

fail to represent complex geometries that do not consist solely of such primitives. Recently, these methods were replaced by instance segmentation models, which output point clusters for each object [10, 21, 29, 31]. Although segmentation methods have been heavily researched, extracting high-level geometric representations for objects has received little attention. Instead, the preferred approach is to retrieve the best fitting template from a library of object templates for each object. This leads to considerable inaccuracies in geometric representation [19]. Furthermore, the structural information encoded within high-level representations is essential for a variety of downstream use cases of digital models such as pipe network modelling and infrastructure monitoring.

The term 'semantic enrichment' encompasses a broad range of information that could be added to a model. Crucial amongst these are connectivity relationships. Infrastructure is a network of interconnected elements. Thus, connectivity is paramount to an infrastructure model. Yet current automation approaches merely segment and model individual, disjoint elements. Many types of infrastructure typically contain hundreds of thousands of elements of various categories [1], making relationship detection a complex task for both man and machine. This is compounded by occlusions and noise within scans. Consequently, automated connectivity modelling has thus far relied solely on simple, hand-crafted rules.

This paper introduces a new method for geometric model fitting on object point clusters. We propose a supervised framework based on PointNet++ [22] to regress model parameters. We ensure stable training by introducing an additional loss term based on chamfer loss, which is measured by sampling points on the predicted object surface in a differentiable manner. This also enables further fine-tuning obtained predictions using gradient descent. Crucially, this method is extendable to arbitrary geometric shapes. We next utilise these parameters to introduce a learned approach for object connectivity inference. This serves as an example for the utility of high-level parametric representations of geometry for downstream tasks. In summary, our contributions are: (1) A novel method for modelling both primitive and non-primitive geometries of BIM objects. (2) A new method for exploiting model geometry for connectivity modelling with zero hand coded rules. (3) The first synthetic dataset of industrial facility BIM models with connectivity labels.

2 Related Work

We focus primarily on work in the fields of geometric modelling and connectivity inference, as well as relevant work on primitive fitting and inductive link prediction.

Geometric Modelling. Current literature on infrastructure digitisation focuses predominantly on point cloud instance segmentation. These are based on 3D deep learning architectures such as PointNet++ [23], DGCNN [31], and Point Cloud Transformer (PCT) [10]. They encompass multiple domains such as industrial facilities [4], buildings [6] and bridges [33]. However, rule-based extractors are also used for continuous assets such as roads and railways [2, 15]. The segmented clusters are modelled geometrically using shape retrieval from a pre-defined template library [2, 6, 16]. Beyond the BIM domain, approaches to extract high-level geometrical information broadly focus on either surface modelling [27], geometric primitive fitting or Constructive Solid Geometry (CSG) representations.

Primitive Fitting. Supervised learning methods have been used in recent approaches for primitive fitting (SPFN) [19] and shape template learning [9]. Typically, direct regression of parameters from 3D points leads to low accuracy owing to the disparity between regression loss and fitting error. To alleviate this, SPFN introduces a "Fitting residual loss" which measures the square distance between the cloud and the predicted primitive surface [19]. As with RANSAC, minimising the L2 loss between the primitive surface and a set of points reduces to a least-squares optimisation problem. This requires non-linear iterative solvers for any but the most basic primitive shapes such as planes. Thus, the authors are forced to design a new differentiable heuristic distance function for each primitive shape they consider, severely limiting the method's adaptability to any shape that cannot be described by existing primitives. Alternatively, CSGNet [26] models elements through a set of boolean operations on shape primitives. While this approach succeeds in representing a far wider range of geometries, it does not extract high-level structural information of the object, and suffers from low accuracy due to reliance on classification over a discrete parameter space instead of regression [19].

Connectivity Inference. Object connectivity inference has been scarcely explored in most domains and relies heavily on hand-coded rules. For instance, in industrial facilities, rules regarding center-lines of piping elements are used to calculate their connectivity [20, 23, 34]. Such methods are restrictive, require parameter tuning and are prone to failure in complex scenarios. They also exhibit poor scalability across different infrastructure domains, requiring domain expertise to craft new rules for each domain.

Inductive Link Prediction. Inductive link prediction involves the prediction of links between nodes that were unobserved during training. Various contemporary GNN models such as Graph Convolution Networks (GCN) [18] are capable of inductive learning. However, in the case of a completely disconnected new node that solely contains feature information, feature aggregation fails due to absence of edges. This issue is addressed by DEAL [12] and Edgeless-GNN [28]. In particular, Edgeless-GNN constructs a proxy graph based on similarity of node attributes whose edges are used for message-passing [28]. However, these approaches still rely on the existence of edges on the rest of the graph, and do not function on completely edgeless graphs.

BIM graph representations. BIM models are naturally suited for graph representation, as a model is essentially a collection of interconnected elements [13]. The most common open-source BIM format, Industry Foundation Classes (IFC) supports graph representations. However, beyond their use for querying building information [35], applications of graph learning on BIM models are scarce and focus solely on node classification. Specifically, a

few works utilise graph representations of IFC models for room-type classification. These utilise GCN or a modified GraphSAGE architecture and focus on indoor spaces [5, 32].

3 Parametric Modelling

This section proposes a method for geometric modelling of BIM elements. We focus on a subset of element classes found in industrial facilities, namely cylinders, elbows, tees and flanges. These elements account for the majority of the modelling workload [3]. Segmented element point clusters from existing industrial facilities are the input data source. A parameter inference model is trained for each class of interest. For training, we generate a synthetic dataset of 3D object models for each class from randomly generated model parameters. For each object, points are sampled from 3 random camera poses to simulate occlusion. Thus, occlusions are modelled on a per-element basis.

Model parameter prediction is performed in two steps. (1). Parameter regression and (2). Parameter fine-tuning (Figure 1). For parameter regression, PointNet++ is used as an encoder, and is followed by a fully connected network for regression. We address the low accuracy that results from training solely on regression loss by incorporating model fitting error through chamfer loss L_c. During training, for each set of predicted parameters θ_{pred}, we sample points from the predicted model p_{pred}. We achieve this by directly calculating points on the surface of the model denoted by the parameters, ensuring differentiability. The same process is repeated to obtain points p_{gt} for the ground truth parameters θ_{gt}. Sampling is performed at predefined intervals to ensure a 1:1 mapping between points from the two models. Next, we compute Bi-directional chamfer loss between these two models. Bi-directional loss restricts the network from predicting extremely large or small models to minimise loss in a single direction. During the initial phases of training, the model parameters may not represent a realistic geometric model, making chamfer loss ill-suited for training. Thus, we cap chamfer loss to a fixed ceiling (L_{max}), and combine it with mean squared regression loss L_r.

$$L_c = \min(L_{max}, \sum_j \min_i ||p_{gt,i} - p_{pred,j}||_2 + \sum_i \min_j ||p_{gt,i} - p_{pred,j}||_2) \tag{1}$$

$$L_r = \sum_{i=1}^{D} (\theta_{gt,i} - \theta_{pred,i})^2 \tag{2}$$

Computing chamfer loss between directly sampled points from the surface of the predicted model and ground truth model is a key component of our approach. This allows us to bypass the aforementioned least squares optimisation problem. This in turn negates the need to devise a unique distance heuristic to compute distance between the predicted surface and ground truth points for each object shape. This facilitates the scalability of our approach to any complex non-primitive shape with minimal additional work, provided that its surface can be modelled parametrically.

During inference, coarse parameter predictions are obtained using the above regression model. Next, they are fine-tuned by iteratively optimising parameters to minimise fitting error, represented by weighted chamfer loss between points on the surface of the predicted model and input point cloud p_{in}. A larger bias (β) towards forward loss (loss from input cloud to predicted model) ensures a degree of tolerance towards occlusions. The differentiable nature of the chamfer loss calculation enables the gradients of the fitting error to be

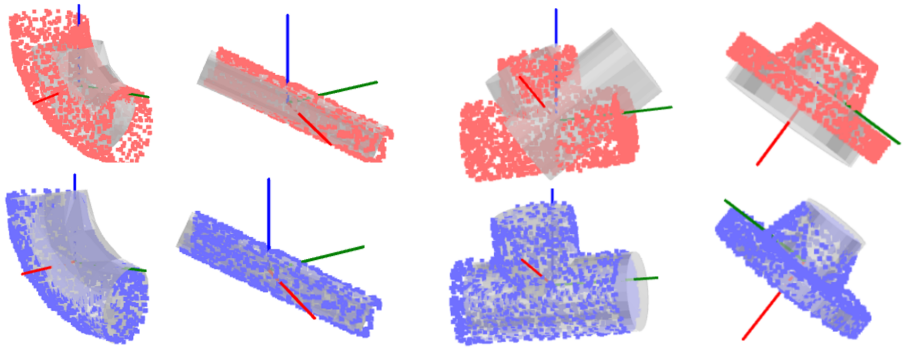

Figure 2: Predicted model (grey) for sample point clusters from each element class, before fine-tuning (top), and after fine-tuning (bottom)

computed. Thus, gradient descent is employed for optimisation. As the initial parameters are the coarse predictions from the first step, we posit that the optimisation process is unlikely to suffer from local minima. Figure 2 depicts model fit before and after fine-tuning.

$$Error = \sum_j \min_i ||p_{in,i} - p_{pred,j}||_2 + \beta \times \sum_i \min_j ||p_{in,i} - p_{pred,j}||_2 \quad (3)$$

4 Connectivity Inference

This section proposes a method for automated connectivity inference between elements of a BIM model, given a set of segmented point clusters of elements. It builds upon previous work on a BIM graph connectivity inference approach [14] which demonstrates the feasibility of graph inference on BIM elements. This approach achieves low precision and recall due to the use of naive bounding box based node features. Elements and their relationships are represented in the form of a graph. Each element is modelled as a graph node, and its geometric features are encoded into a graph node feature vector. Relationships are denoted by graph edges. Relationship detection is modelled as an edge prediction task and a GNN is trained for this purpose (Figure 3).

The choice of node features regulates the amount of information the GNN can exploit during training. Potential features include: (a) minimum-oriented bounding boxes (MOBB) of elements (*e.g.,* principle axis, dimensions and center of bounding box), (b) geometric element parameters, or (c) a sampled subset of points. MOBBs can easily be derived but are limited to crude information regarding element geometry and position. These are heavily affected by outliers and errors in instance segmentation. In contrast, element specific features are more difficult to extract from point clouds, but are more descriptive, especially for shapes such as elbows. They can represent geometries accurately with few parameters. Examples include element position (*e.g.,* centre point), orientation (*e.g.,* cylinder axis) and element geometry (*e.g.,* radius, length, elbow angle). Thus, we utilise the work proposed in Section 3 to extract model parameters from each point cluster, and use these to compute node features.

In order to represent the BIM model with homogeneous nodes, we seek a node feature set that is universally applicable to all element types in our industrial facility dataset. To this

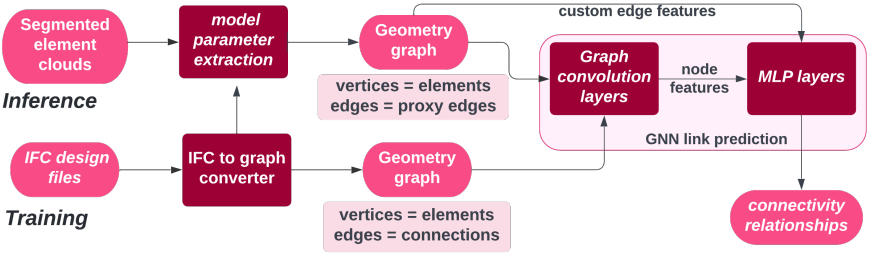

Figure 3: Our connectivity inference approach learns the underlying nature of connectivity relationships in a given BIM domain using a GNN

end, we compute features that capture the axis direction, radius, and position of each opening at the end of an element. Tees contain three such open ends while pipes, flanges and elbows contain two. Therefore, the feature vector supports up to three sets of features describing an open end. In addition, we concatenate the element class label to the node feature vector.

Our GNN architecture is primarily based on the GraphSAGE [11], and consists of a series of GraphSAGE layers followed by Multilayer Perceptron (MLP) layers, similar to [14]. The GraphSAGE layers aggregate information between nodes to compute node features while the MLP layers compute edge probability between each pair of nodes. However, we make key modifications to the standard GraphSAGE architecture. Crucially, we directly compute two edge features relevant to connectivity between each pair of nodes and concatenate them to the MLP layer input. Specifically, we compute the minimum distance d and angle α between the axes $((r_x, e_x), (r_y, e_y))$, of each pair of open ends.

$$d = \frac{|(e_x \times e_y) \cdot (r_x - r_y)|}{||(e_x \times e_y)||} \qquad \alpha = \tan^{-1}\left(\frac{(e_x \cdot e_y)}{(||e_x|| \times ||e_y||)}\right) \tag{4}$$

Our training dataset consists of object and connectivity information from a large industrial facility. We dynamically sample negative edges during training as the negative graph contains n^2 potential edges for n elements. For inference, we reduce the search space by omitting edge candidates between elements spaced beyond a distance threshold. During inference, we are initially presented with a completely edgeless graph as input. To counter this, we extend the concept of generating proxy edges for edgeless node inference [28], by generating proxy edges for the entire graph based on element proximity. These edges ensure feature aggregation between all neighbours within a distance threshold.

5 Experiments

We evaluate our method qualitatively and quantitatively using our industrial facility dataset. For parametric modelling, we measure fitting error and model parameter error, and provide a comparison with RANSAC. For connectivity inference, we compare our approach with the existing rule-based inference method, and provide an ablation study to validate our approach. Further ablation studies for the parametric modelling approach are provided in the supplementary material.

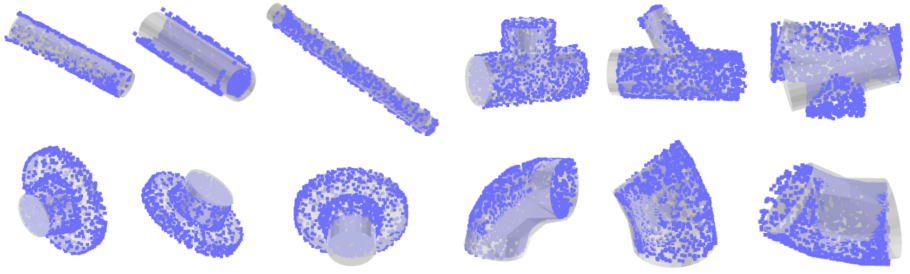

Figure 4: Sample predicted models for each class in industrial facility dataset, chamfer loss for each model lies within the 2.5m-25m range.

Dataset. We extract object and connectivity information from a design file of an offshore LNG hub to generate a dataset, obviating the need for manual connectivity annotation. This ensures that the dataset reflects the geometry and topology found in real data. The resultant dataset is used for both qualitative evaluation of the parameter modelling approach as well as for training and quantitative evaluation of the connectivity inference approach. The dataset consists of around 37,000 pipe, flange, elbow and tee elements with around 31,000 connectivity relationships (Figure 6, right). The dataset contains two independent sections forming disjoint graphs of approximately similar size, one of which is used for testing.

5.1 Parametric Modelling

Implementation details. The synthetically generated object dataset used for training the parameter regression network consists of 2048 points each for 4,096 objects per class of interest. 10% of this dataset serves as the test set. The differentiable sampling of points from predicted geometries is parallelised to integrate into model training, and to enable batch optimisation during the fine-tuning step. The fine-tuning step is carried out for 100 iterations during inference, and ADAM is used for optimisation [17]. We provide a comparison of ADAM with optimisation methods such as BFGS and Newton-CG, with and without robustness. Robustness is implemented via a Huber kernel over chamfer loss terms, as suggested by Fitzgibbon [8]. A comparison for elbow elements is given in Table 1. These methods show no distinct improvement in parameter errors, and are still sensitive to local minima. We believe ADAM performs well because it provides individual learning rates for each parameter. Notably, this also enables us to optimise a large batch of input clouds in parallel by simply optimising the sum of losses over the input dataset.

We experimentally determine that a 3x weight on forward chamfer loss ensures occlusion resilience. We utilise Point Cloud Library's RANSAC cylinder segmentation implementation for benchmarking [24]. This includes modifications over vanilla RANSAC such as local optimisation. Additional variations of RANSAC such as PROSAC and Randomised MSAC did not provide notable improvements.

Performance evaluation. To the best of our knowledge, there is no existing baseline for parametric model fitting of industrial facility BIM elements aside from RANSAC based approaches on pipes. Previous work on other element types focus exclusively on shape retrieval from CAD model libraries and thus are not directly comparable [2]. For evaluation, we measure chamfer loss (Figure 5) and parameter error on a selected set of geometric parameter

Metric	GD(Adam)-H	GD(Adam)	Newton-CG-H	Newton-CG
Axis dev. (deg)	19.1	18.7	19.8	19.4
Radius error (%)	6.8	6.9	7.9	7.7
Position error (mm)	16.3	16.3	15.7	15.9

Table 1: Optimisation method comparison, H=Huber kernel (k=0.001)

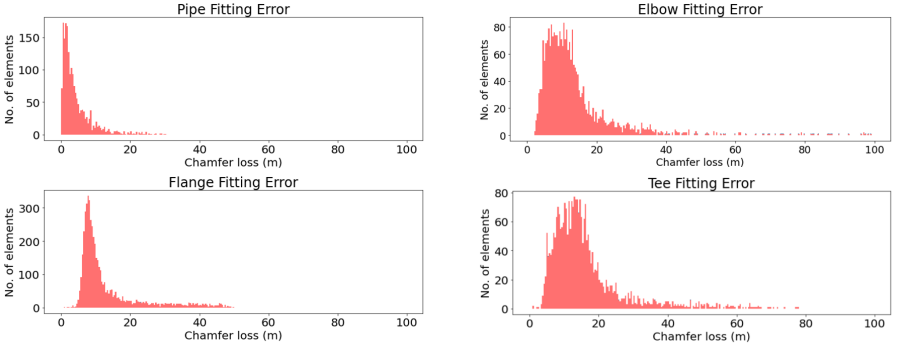

Figure 5: Chamfer loss distribution across each class (no. of points = 2048).

predictions for each class in Table 2. We note that performance diminishes with complexity of the model. For instance, less visually obvious features such as the axis / center of an elbow carries larger error. However, visual analysis on the industrial facility dataset illustrates that the predicted models exhibit acceptable fit in the majority of cases (Figure 4).

Next, we compare our approach with the RANSAC baseline for pipe and tee objects in Table 3. RANSAC cannot be directly applied for segmenting elbows and flanges as they cannot be modelled with available primitives. Thus, we omit these shapes. We observe that our model exhibits significant improvement over RANSAC, particularly in the presence of considerable occlusions and noise.

5.2 Connectivity Inference

Implementation details. Our GNN architecture comprises 4 SageGraph layers and 5 MLP layers with ReLU activations, and is trained with Binary Cross-Entropy loss. An ablation study on these parameters and alternative edge feature computation methods is provided in supplementary material. We adopt F1 score, recall, and precision as evaluation metrics.

Performance evaluation. Overall, our approach achieves a precision of 83.5, recall of 98.9, and F1 score of 90.5 on the test-set at 0.5 classification threshold. Visual analysis (Figure 6, left) shows that it struggles with small parallel pipes in close proximity and tees with small

Metric	Pipe	Flange	Elbow	Tee
Chamfer loss (m)	4.32	13.13	17.94	16.16
Primary Axis deviation (deg)	10.5	12.7	18.7	9.9
Radius error (%)	4.4	4.7	7.0	16.7
Element position error (mm)	1.11	1.04	16.30	1.13

Table 2: Averaged per class error metrics on test-set.

	Low occlusion/ Noise				High occlusion/ Noise			
	Pipe		Tee		Pipe		Tee	
	R	O	R	O	R	O	R	O
Chamfer loss (m)	6.6	4.2	23.3	14.7	10.1	6.4	31.6	19.4
Primary Axis deviation (deg)	7.6	8.9	7.9	8.8	16.1	11.3	29.6	12.9
Radius error (%)	3.1	11.7	8.0	14.3	13.4	8.1	21.8	16.2
Element position error (mm)	1.2	1.0	1.3	1.2	1.9	1.5	3.8	1.4

Table 3: Comparison with RANSAC baseline on tee and pipe elements, with high and low occlusion and noise. High and low occlusions are achieved by sampling points through two and four camera poses respectively (R = RANSAC, O = Ours).

Feature representation	Precision	Recall	F1
Geometric model node features & custom edge features	83.5	98.9	90.5
Geometric model node features only	83.5	97.6	90.0
MOBB node features only [14]	74.3	94.9	83.3

Table 4: Ablation study on node and edge features.

radii. Element-wise analysis further demonstrates markedly poorer performance on links between connecting elements (Table 5). We posit that this is primarily due to the scarcity of such element relationships in training data (Figure 6, right).

For comparison, we implement the method proposed by Oh and Kwang [20]. Despite additional optimisations and parameter modifications to suit our dataset, the approach fails to provide significant results, achieving a F1 score below 0.5. This attests to the inadequacy of rule-based approaches in complex datasets. Indeed, previous work provides only visual analysis of connectivity inference and operates on much simpler datasets [20, 23]. However, it should be noted that these rule-based approaches rely solely on pipe information, and derive existence of elbow and tee joints through connectivity, as geometries for these elements were not extracted. Our approach is able to take advantage of these element classes due to our geometric modelling step.

We provide a brief ablation study to validate the utility of our node and edge features (Table 4). Our results show that custom edge features representing open ends of piping elements contribute to higher model performance, affirming the relevance of task-dependent feature design. Crucially, we observe that the model performs substantially better when element geometries are denoted by geometric model parameters as opposed to MOBB parameters. Thus, we validate the relevance of high-level structural information in BIM modelling.

Precision	Flange	Elbow	Tee	Pipe
Flange	75.9	97.3	28.5	97.6
Elbow		98.3	61.0	98.0
Tee			94.1	71.6
Pipe				69.4

Recall	Flange	Elbow	Tee	Pipe
Flange	97.6	100.	58.0	99.9
Elbow		100.	100.	99.8
Tee			94.1	99.4
Pipe				97.4

Table 5: Precision and recall by element type on test set.

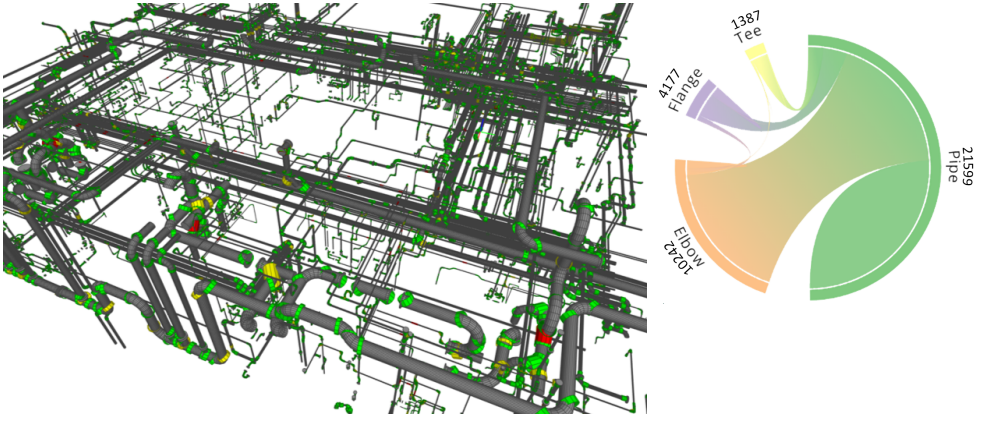

Figure 6: Results of connectivity inference on industrial facility test set *(green = true positives, yellow = false positive, red = false negative)* (left) and chord diagram of relationship distribution by class in dataset (right)

6 Conclusion

We introduce a novel method for modelling both geometry and connectivity of elements within a BIM model. Our parametric modelling approach provides a more accurate geometric representation compared to existing template matching or primitive fitting methods while ensuring occlusion resilience. Our approach also bypasses the necessity of custom heuristic distance functions, ensuring compatibility with arbitrary parametric models. Moreover, our GNN connectivity inference approach learns the underlying nature of connectivity relationships within a BIM model from a design dataset. Crucially, this method ensures easy adaptability of our method to other BIM domains as well as the ability to handle complex connectivity scenarios. Aside from outperforming existing approaches, the method validates the significance of extracting high-level structural information from BIM object point clusters.

Current limitations of our approach include; the susceptibility to local minima due to erroneous initial predictions, failure cases due to severe occlusions, and reliance on instance segmentation as a pre-processing step. Potential future work includes pre-training the regression model with self-supervised learning via occlusion completion and additional experiments with robustness to mitigate non-corresponding points. Furthermore, the derived connectivity information can be fed back to the initial steps of the Scan-to-BIM pipeline, enhancing the performance of upstream tasks such as instance segmentation by providing additional context.

7 Acknowledgements

This research has received funding from the Engineering and Physical Sciences Research Council (EPSRC), AVEVA Group plc and British Petroleum plc.

References

[1] Eva Agapaki and Ioannis Brilakis. CLOI-NET: Class segmentation of industrial facilities' point cloud datasets. *Advanced Engineering Informatics*, 45, 8 2020. ISSN 14740346. doi: 10.1016/j.aei.2020.101121.

[2] Eva Agapaki and Ioannis Brilakis. Geometric Digital Twinning of Industrial Facilities: Retrieval of Industrial Shapes. 2 2022. doi: 10.48550/arxiv.2202.04834. URL https://arxiv.org/abs/2202.04834v1.

[3] Eva Agapaki, Graham Miatt, and Ioannis Brilakis. Prioritizing object types for modelling existing industrial facilities. *Automation in Construction*, 96:211–223, 12 2018. ISSN 09265805. doi: 10.1016/j.autcon.2018.09.011.

[4] Eva Agapaki, S M Asce, Alex Glyn-Davies, Sara Mandoki, Ioannis Brilakis, M Asce, Ph D Candidate, Laing O', and Rourke Reader. CLOI: A Shape Classification Benchmark Dataset for Industrial Facilities. Technical report, 2019.

[5] Adam Buruzs, Miloš Šipetić, Brigitte Blank-Landeshammer, and Gerhard Zucker. IFC BIM Model Enrichment with Space Function Information Using Graph Neural Networks. *Energies*, 15(8), 4 2022. ISSN 19961073. doi: 10.3390/en15082937.

[6] Jingdao Chen, Zsolt Kira, and Yong K. Cho. Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction. *Journal of Computing in Civil Engineering*, 33(4):04019027, 7 2019. ISSN 0887-3801. doi: 10.1061/(asce)cp.1943-5487.0000842.

[7] Yun Jian Cheng, Wen Ge Qiu, and Dong Ya Duan. Automatic creation of as-is building information model from single-track railway tunnel point clouds. *Automation in Construction*, 106, 10 2019. ISSN 09265805. doi: 10.1016/j.autcon.2019.102911.

[8] Andrew Fitzgibbon. Robust Registration of 2D and 3D Point Sets. *Image and Vision Computing*, 2002. doi: 10.1016/j.imavis.2003.09.004.

[9] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning Shape Templates with Structured Implicit Functions. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. doi: 10.1109/ICCV.2019.00725.

[10] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: Point cloud transformer. 12 2020. doi: 10.1007/s41095-021-0229-5. URL http://arxiv.org/abs/2012.09688http://dx.doi.org/10.1007/s41095-021-0229-5.

[11] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. 6 2017. URL http://arxiv.org/abs/1706.02216.

[12] Yu Hao, Xin Cao, Yixiang Fang, Xike Xie, and Sibo Wang. Inductive Link Prediction for Nodes Having Only Attribute Information. 7 2020. doi: 10.24963/ijcai.2020/168. URL http://arxiv.org/abs/2007.08053http://dx.doi.org/10.24963/ijcai.2020/168.

[13] Ali Ismail, Barbara Strug, and Grażyna Ślusarczyk. Building Knowledge Extraction from BIM/IFC Data for Analysis in Graph Databases. pages 652–664. 2018. doi: 10.1007/978-3-319-91262-2{_}57.

[14] Haritha Jayasinghe and Ioannis Brilakis. Element Relationship Modelling for Industrial Facility Digitisation Using Graph Neural Networks. *Manuscript submitted for publication*, 2023.

[15] Andrés Justo, Mario Soilán, Ana Sánchez-Rodríguez, and Belén Riveiro. Scan-to-BIM for the infrastructure domain: Generation of IFC-complaint models of road infrastructure assets and semantics using 3D point cloud data. *Automation in Construction*, 127, 7 2021. ISSN 09265805. doi: 10.1016/j.autcon.2021.103703.

[16] Hyungki Kim, Changmo Yeo, Inhwan Dennis Lee, and Duhwan Mun. Deep-learning-based retrieval of piping component catalogs for plant 3D CAD model reconstruction. *Computers in Industry*, 123, 12 2020. ISSN 01663615. doi: 10.1016/J.COMPIND.2020.103320.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 12 2014. URL http://arxiv.org/abs/1412.6980.

[18] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 9 2016. doi: 10.48550/arxiv.1609.02907. URL https://arxiv.org/abs/1609.02907v4.

[19] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas Guibas. Supervised Fitting of Geometric Primitives to 3D Point Clouds. Technical report, 2018.

[20] Inyoung Oh and Hee Kwang. Automated recognition of 3D pipelines from point clouds. *The Visual Computer*, 37:1385–1400, 2021. doi: 10.1007/s00371-020-01872-y. URL https://doi.org/10.1007/s00371-020-01872-y.

[21] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. 12 2016. URL http://arxiv.org/abs/1612.00593.

[22] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, 6 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.

[23] Rongqi Qiu, Qian-Yi Zhou, and Ulrich Neumann. LNCS 8691 - Pipe-Run Extraction and Reconstruction from Point Clouds. Technical report, 2014.

[24] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *Proceedings - IEEE International Conference on Robotics and Automation*, 2011. ISBN 9781612843865. doi: 10.1109/ICRA.2011.5980567.

[25] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 0:1–12, 2007. doi: 10.1111/j.1467-8659.2007.01016.x.

[26] Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. CSGNet: Neural Shape Parser for Constructive Solid Geometry. *Conference on Computer Vision and Pattern Recognition, {CVPR} 2018*, 12 2017. doi: 10.1109/CVPR.2018.00578.

[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 *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12352 LNCS, pages 261–276. Springer Science and Business Media Deutschland GmbH, 2020. ISBN 9783030585709. doi: 10.1007/978-3-030-58571-6{_}16.

[28] Yong-Min Shin, Cong Tran, Won-Yong Shin, and Xin Cao. Edgeless-GNN: Unsupervised Representation Learning for Edgeless Nodes. 4 2021. URL http://arxiv.org/abs/2104.05225.

[29] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view Convolutional Neural Networks for 3D Shape Recognition. *ICCV*, 2015. URL http://vis-www.cs.umass.edu/mvcnn.

[30] Pingbo Tang, Daniel Huber, Burcu Akinci, Robert Lipman, and Alan Lytle. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques, 2010. ISSN 09265805.

[31] Yue Wang, Michael M Bronstein, Justin M Solomon, Yongbin Sun, Ziwei Liu, and Sanjay E Sarma. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph. 1, 1, Article*, 1(1):13, 2019. doi: 10.1145/3326362. URL https://doi.org/10.1145/3326362.

[32] Zijian Wang, Rafael Sacks, and Timson Yeung. Exploring graph neural networks for semantic enrichment: Room type classification. *Automation in Construction*, 134, 2 2022. ISSN 09265805. doi: 10.1016/j.autcon.2021.104039.

[33] Tian Xia, Jian Yang, and Long Chen. Automated semantic segmentation of bridge point cloud based on local descriptor and machine learning. *Automation in Construction*, 133, 1 2022. ISSN 09265805. doi: 10.1016/j.autcon.2021.103992.

[34] Yuan Xie, Siyi Li, Tianrui Liu, and Yiyu Cai. As-built BIM reconstruction of piping systems using PipeNet. *Automation in Construction*, 147, 3 2023. ISSN 09265805. doi: 10.1016/J.AUTCON.2022.104735.

[35] Junxiang Zhu, Peng Wu, and Xiang Lei. IFC-graph for facilitating building information access and query. *Automation in Construction*, 148, 4 2023. ISSN 09265805. doi: 10.1016/J.AUTCON.2023.104778.