

Cognitive Scan-to-BIM: LLM-Driven Hyperparameter Optimization for Robust MEP Primitive Fitting in Complex Construction Environments

Muhammad Talha¹, Rafiq Ahmad¹, Yitong Li¹, Omair Shafiq², Qipei (Gavin) Mei¹

¹University of Alberta, Edmonton, Canada

Email: {tmuhamm1, rafiq.ahmad, yitong2, qipei.mei}@ualberta.ca

²University Malaysia Pahang Al Sultan Abdullah, Kuantan, Malaysia

Email: pdz24005@adab.umpsa.edu.my

Abstract—Generating accurate Digital Twins of existing buildings remains a persistent bottleneck in Scan-to-BIM workflows, especially for Mechanical, Electrical, and Plumbing (MEP) systems. Although recent deep learning methods can segment MEP components from point clouds with good accuracy, the next step for converting segmented point sets into metrically consistent cylindrical primitives for downstream BIM integration remains brittle and heavily dependent on manually tuned geometric hyperparameters. On active construction sites, especially timber frames, severe clutter, occlusion, and variable point density make such tuning labor-intensive and unstable. This paper presents a novel approach called “Cognitive Fitting,” in which a Large Language Model (LLM) is integrated into a closed-loop optimization process as a proposal engine for hyperparameter search. The LLM does not generate geometry directly. Instead, it receives structured geometric feedback, including coverage and dispersion for candidate ranking together with fragmentation-related diagnostic signals for iterative reasoning, and proposes improved clustering, fitting, and merging parameters. The proposed framework was validated on a challenging residential timber dataset from Canada containing heavily occluded wiring and piping. Results show the strongest gains on the thin and fragmented *Wires* class, where coverage improved from 66.19% to 71.81% relative to a budget-matched non-LLM search condition, while the number of merged output primitives decreased from 341 to 302, indicating improved continuity recovery. For *Pipes*, the gains were smaller but still consistent, with the proposed method achieving the best overall objective score while reducing dispersion relative to the fixed classical baseline. These results show that LLM-guided reasoning can reduce manual hyperparameter tuning in post-segmentation MEP primitive fitting, particularly for thin and fragmented structures in cluttered timber environments.

Index Terms—Scan-to-BIM, Large Language Models (LLM), Hyperparameter Optimization, MEP Reconstruction, Digital Twins, Automation

I. INTRODUCTION

A paradigm shift is underway in the Architecture, Engineering, and Construction (AEC) industry, aligned with Industry 4.0. Central to this shift is the Digital Twin concept. These are not static models. They are dynamic, living systems for full lifecycle asset management [1, 2]. Creating them starts with Scan-to-BIM. This is the process of transforming terrestrial

laser scanning (TLS) data into semantic Building Information Models. TLS hardware today captures geometry rapidly and with millimeter accuracy. The bottleneck has moved downstream. Converting raw point clouds into parametric BIM objects is now the major constraint. Studies indicate this phase can consume 70% of the total modeling effort on a project, all due to manual work [3].

This bottleneck tightens dramatically around MEP systems. Structural elements like walls are large and planar. MEP components are not. They have small cross-sections. Their topologies are complex and curvilinear. They are highly susceptible to occlusion by other building elements [4]. Timber construction compounds these issues. Temporary supports, insulation, and general site clutter further obscure pipes and wires. Standard automated fitting algorithms struggle here. They frequently fail to reconstruct continuous runs. Instead, they output fragmented geometry. Practitioners often call it “spaghetti.” This output requires extensive manual rework, defeating the purpose of automation.

We contend the core problem is not the algorithms themselves. Efficient Random Sample Consensus (RANSAC) for shape detection and density-based clustering like Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) are mathematically sound [5, 6]. The problem is their parameter sensitivity. RANSAC, which finds shapes by randomly sampling points and checking for consensus, needs a distance threshold. A threshold ideal for a large pipe will completely miss a thin wire. HDBSCAN, which groups points based on local density, has settings like minimum cluster size. Clustering settings tuned for a dense mechanical room will catastrophically over-segment sparse attic wiring. Today, finding the right set of parameters for a given scan is manual. It is a trial-and-error process that leans heavily on the modeler’s tacit knowledge. There is no systematic method.

This study proposes an automated solution. We introduce an optimization framework driven by a Large Language Model (LLM). Our hypothesis is straightforward. The heuristic reasoning a BIM engineer applies to diagnose poor fits and adjust

tolerances can be emulated by an LLM operating as a proposal engine inside a closed-loop geometric search process. In our framework, the LLM does not generate geometry, parse BIM entities, or replace the fitting algorithms. Instead, it receives structured geometric feedback from the pipeline, reasons about likely failure modes, and proposes adjusted hyperparameters for the next candidate evaluation. The geometric engine then evaluates those candidates objectively and retains the best-scoring result.

This distinction is important. The contribution of this paper is not “LLM-generated reconstruction.” It is LLM-guided hyperparameter search for post-segmentation primitive fitting in cluttered residential timber scenes. The system is therefore scene- and class-specific: given a semantically segmented point cloud class such as pipes or wires, it searches for better clustering, fitting, and merging settings without requiring manual trial-and-error from an expert modeler.

The diagram illustrates the closed-loop cycle of our framework. At its core is an LLM Agent, instantiated in our experiments with Qwen-2.5-14B under local inference. This agent proposes a set of hyperparameters, θ_t . A Geometric Engine then takes these parameters and the input point cloud. It runs HDBSCAN for clustering and a cylinder fitting stage for primitive estimation, followed by rule-based merging and refinement. Next, an Evaluator module scores the resulting geometry. It computes a scalar objective score J from coverage and dispersion, while also recording additional diagnostic signals such as primitive count and fragmentation. The scalar score is used for candidate ranking, whereas the auxiliary signals are formatted as plain-text feedback and passed to the LLM to support reasoning about the next parameter set, θ_{t+1} . This cycle repeats, pushing the search toward parameter settings that generate cleaner, more continuous, and metrically valid primitives from the segmented point cloud.

To make the feedback loop concrete, the text sent to the LLM may include a short summary such as: “Iteration 7. Parameters used: `min_cluster_size=45`, `ransac_dist=0.008`. Results: Coverage = 58%, Dispersion = 5.2 mm, Merged Primitive Count = 310. The output is still highly fragmented, though fitting error is acceptable.” The LLM then reasons over this summary and proposes a new candidate, for example by relaxing clustering conservatism or widening a merge tolerance. This iterative proposal-and-evaluation dialogue is the core of the Cognitive Fitting approach.

II. RELATED WORK

A. Scan-to-BIM: From Manual to Automated Methods

Initial Scan-to-BIM techniques depended on user guidance. They used primitive fitting or constructive solid geometry (CSG) [1]. These methods worked for basic structures. They cannot deliver the semantic depth modern Digital Twins require. The field changed with deep learning. Current state-of-the-art relies on networks like PointNet++ and Point Transformers [7, 8]. They segment point clouds with high per-point accuracy. But segmentation is just the beginning. The

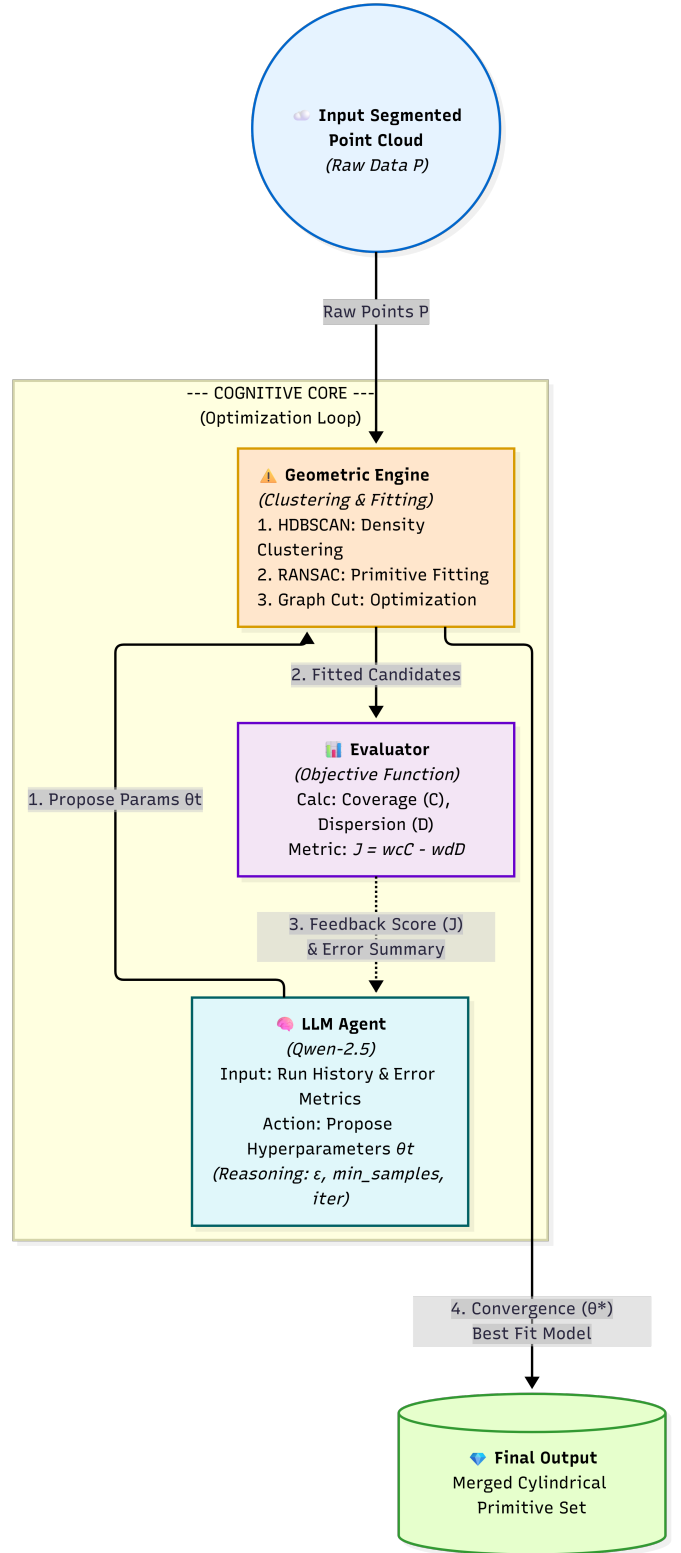


Fig. 1. Architecture of the Cognitive Scan-to-BIM Framework.

crucial next step is reconstruction. Turning labeled points into parametric IFC entities still uses classical geometric fitting algorithms. And these algorithms have a well-known weakness. They are fragile. Their performance degrades badly with real-world noise, occlusion, and uneven point density [9]. You can have perfect segmentation and still get a useless model if the fitting step fails.

B. Why MEP Reconstruction is Especially Hard

MEP systems break the assumptions of most reconstruction pipelines. Their cylindrical, curvilinear geometry is fundamentally different from planar elements. Agapaki and colleagues made this distinction clear, noting industrial pipes demand unique handling [10]. Some recent work tries to solve this with more AI. Yue *et al.* (2025) developed deep learning networks to complete occluded pipes [11]. These show potential for better connectivity. But they come with a heavy cost. They require large, labeled, domain-specific datasets for training. For residential construction, such datasets are rare and expensive to create. The alternative is procedural methods like RANSAC or Hough Transforms [5]. They need no training data. Their flaw is brittleness. Success hinges on a few hyperparameters. A poor choice for a distance threshold or minimum cluster size can ruin the entire output, merging objects or missing them entirely [12]. This creates a clear gap. We need an approach that adapts like deep learning but does not need massive training sets. Our framework targets this gap directly.

C. LLMs as Optimization Engines

Large Language Models are now being applied beyond text. Researchers are testing them as reasoning tools for complex tasks. Romera-Paredes *et al.* (2024) demonstrated LLMs solving combinatorial puzzles [13]. In engineering, Yang *et al.* (2023) used them to tune optimization parameters [14]. More relevant to us, new studies position LLMs as optimization agents. Work by Wang *et al.* (2025) and van Stein *et al.* (2025) shows LLMs can navigate parameter spaces more efficiently than random search [16, 17]. They use semantic cues to guide the process. This suggests hyperparameter tuning is not just a random walk. It can be a reasoning task.

So, when these three areas are taken together, a specific research gap comes into focus. On one side you have powerful but data-hungry deep learning methods for MEP reconstruction. On the other you have lightweight but brittle geometric algorithms that need expert tuning. The tuning process itself is a kind of heuristic reasoning problem. But the application of LLMs as reasoning agents for this specific task in AEC for tuning geometric reconstruction hyperparameters in the face of real-world clutter has not been explored. Previous LLM work in optimization is general. No one has plugged an LLM into the Scan-to-BIM loop to deal with the conditions of an active construction site. That is the gap this paper addresses.

III. METHODOLOGY

The MEP reconstruction task is framed as a black-box optimization problem. The goal is to maximize the geometric

quality of the final fitted primitives. The proposed framework eliminates static configuration. A dynamic LLM Agent iteratively tunes the hyperparameters for standard geometric algorithms.

A. Data and Pre-processing

Validation data comes from a Leica BLK360 G1 scanner. It captured an under-construction residential timber building in Canada. This site represents a worst-case test for automation. The raw scans have high geometric noise from wood grain texture. Stud framing creates severe occlusion. The mix includes rigid pipes and flexible electrical wires, each presenting different challenges. The raw point clouds were pre-processed. This involved statistical outlier removal and voxel down-sampling. The data is then semantically segmented into two target classes: ‘Pipes’ and ‘Wires’. These labeled clouds are the input to the system.

B. Cognitive Fitting Engine Components

The pipeline has three core modules, all parameterized by the LLM.

Adaptive Clustering with HDBSCAN: HDBSCAN is used [6]. It handles variable density better than DBSCAN, which is critical for TLS data. In the final reported search space, the primary HDBSCAN parameter optimized by the LLM is *min_cluster_size*, which controls the smallest group of points that forms a valid object. Additional HDBSCAN controls such as *min_samples* and *cluster_selection_epsilon* were explored in preliminary tuning and then fixed before the main reported experiments.

Shape Fitting with RANSAC: Each cluster is processed by an Efficient RANSAC algorithm [5] to find cylindrical models. The key parameter here is *ransac_dist*. This is the maximum allowed distance from a point to the cylinder surface to be considered an inlier. Set it too high and the cylinder swallows noise. Set it too low and valid surface points get rejected.

Graph-Based Merging: Initial cylinders are often broken. A graph traversal step stitches compatible ones into continuous runs. In the final reported search space, the primary merge parameters optimized by the LLM are *merge_angle* (how much cylinder axes can deviate) and *merge_gap* (maximum spatial gap to connect). Additional compatibility guards, including radius-related constraints, were tuned in preliminary runs and then fixed before the main experiments.

C. The LLM Feedback Loop

This is the novel part of the system. The automated reasoning loop.

For each iteration, t :

- 1) The pipeline runs with the current parameters, θ_t . Performance metrics are computed: **coverage** (C), the percentage of input points assigned to accepted primitives; **dispersion** (D), the average point-to-cylinder fitting error; and structural continuity indicators such as raw and merged primitive counts. These signals are useful because thin MEP structures may exhibit acceptable

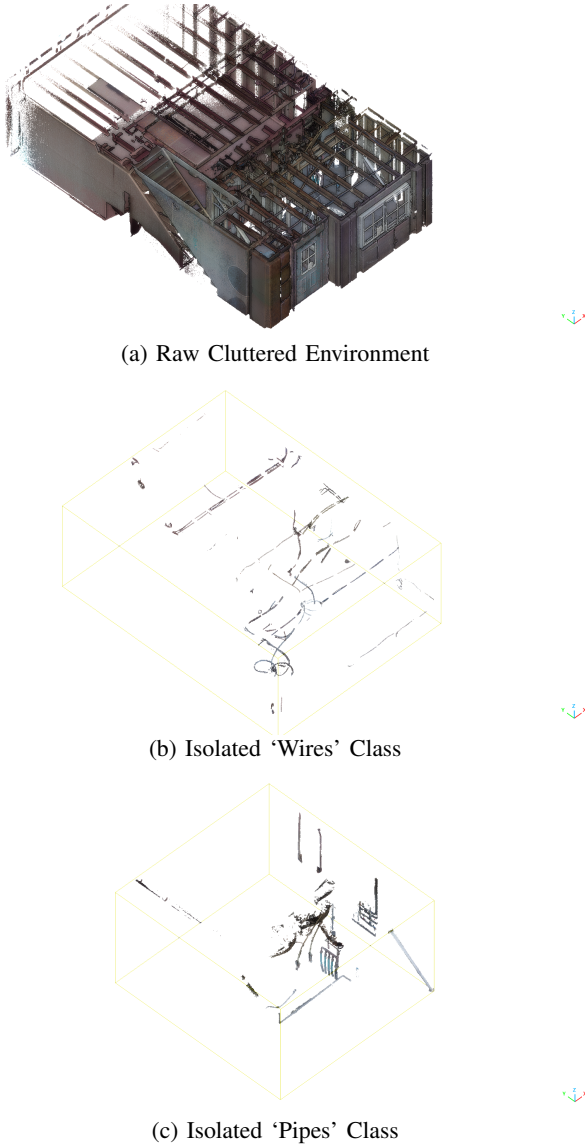


Fig. 2. Input Data Characteristics. Vertical breakdown showing (a) the heavy timber clutter, (b) the disjointed nature of the ‘Wires’ class, and (c) the spatial interference in the ‘Pipes’ class. Note: Views are zoomed to highlight density.

local fitting error while still remaining globally over-fragmented.

- 2) A compact plain-text diagnostic summary is produced and provided to the LLM together with recent search context. The LLM was executed locally using Qwen-2.5-14B-Instruct through LM Studio with fixed low-temperature decoding to reduce stochastic variance. The prompt followed a fixed structured template containing the current parameter set, the latest evaluation metrics, and a short instruction to return only the next bounded candidate configuration — acting as a constrained proposal policy rather than an unconstrained conversational assistant. All candidate parameters were clamped to physically meaningful search ranges before evaluation. A salvage stage using a secondary lightweight DBSCAN

pass was applied to residual noise before final merging; this used fixed settings across all compared methods and was not part of the LLM-tuned search space.

- 3) Candidate solutions are ranked using the scalar objective

$$J(\theta) = w_c \times C - w_d \times \left(\frac{D}{0.01} \right), \quad (1)$$

where the normalization by 0.01 reflects a 10 mm MEP prefabrication tolerance. In all experiments $w_c = 1.0$; $w_d = 0.15$ for wires (favoring continuity) and 0.20 for pipes (favoring precision). These weights were fixed *prior* to running the optimization loop and were not adjusted by observing search outcomes, avoiding circularity. This formulation follows the standard weighted-sum scalarization approach in multi-objective optimization, where weights encode *a priori* preference trade-offs between competing objectives rather than universal constants [15]. In the present study, the selected weights reflect an engineering preference for continuity in wires and geometric precision in pipes. Fragmentation-related signals and primitive counts are treated as auxiliary diagnostic context for the LLM and as secondary evaluation criteria; they are not directly optimized by Eq. (1).

This formulation was chosen deliberately for the present feasibility study. More sophisticated formulations, such as explicit multi-objective optimization or multimodal diagnostic feedback, are promising extensions left outside the scope of this first study.

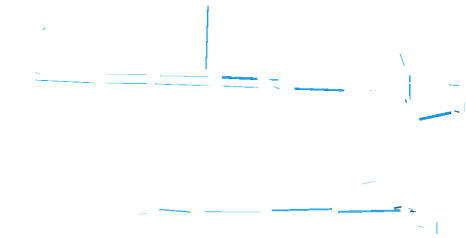
TABLE I
HYPERPARAMETER SEARCH SPACE

Parameter	Unit	Range	Description
hdb_min_cluster	points	[8, 1200]	Minimum points to form a cluster
ransac_dist	meters	[0.001, 0.03]	Max point-to-cylinder distance
merge_angle_deg	degrees	[2, 15]	Max axis deviation for merging
merge_gap	meters	[0.01, 0.12]	Max spatial gap to connect cylinders

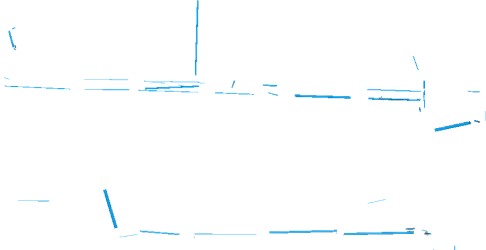
Note: The active LLM-tuned search space reported in this paper consists of the four primary parameters above. Additional clustering and merge-guard parameters, including *min_samples*, *cluster_selection_epsilon*, radius-compatibility, axis-offset, and minimum-overlap constraints, were explored in preliminary runs and then fixed prior to the main reported search.

IV. RESULTS

The proposed framework was compared with two baselines. **Baseline A** uses a standard DBSCAN+RANSAC pipeline with fixed, expert-suggested parameters. **Baseline B** is a budget-matched non-LLM search condition using the same autotuning scaffold but with LLM proposals disabled. The search budget was matched by using the same total round count for each class (24 rounds indexed from 0 to 23 for wires, and 32 rounds indexed from 0 to 31 for pipes), thereby isolating the value of LLM-guided search from generic iterative search. We acknowledge that stronger baselines such as Bayesian optimization or TPE would further strengthen this comparison and are identified as future work. Note that Baseline A uses standard DBSCAN whereas the proposed method uses HDB-SCAN; the comparison therefore reflects practitioner-style



(a) Fragmented Wires Output (Before fix)



(b) Continuous Wires Run (After LLM fix)

LLM Agent Reasoning Step

Observation: Too many pieces (312 fragments)
But: Fitting is accurate (3.1mm error)
Diagnosis: $\text{merge_gap}=0.02\text{m}$ too small
Fix: Increase to $\text{merge_gap}=0.08\text{m}$
Result: $\theta_5: \{\text{merge_gap}: 0.08\text{m}, \dots\}$

The Cognitive Loop:

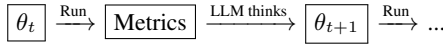


Fig. 3. Example of parameter tuning driven by the LLM. The agent observes a diagnostic summary indicating high fragmentation despite acceptable fitting error. It therefore adjusts merge-related parameters, and the next iteration produces a more continuous wire run. Fragmentation is used here as reasoning context, not as a direct term in the scalar objective.

fixed-parameter fitting against the full proposed framework, while Baseline B provides the controlled ablation of the LLM guidance component specifically.

A. Quantitative Performance

The data shows a consistent benefit from LLM-guided parameter proposals, with the largest gains observed on the more challenging *Wires* class.

TABLE II
PERFORMANCE COMPARISON SUMMARY

Method	Class	Coverage (%)	Dispersion (mm)	J-Score
Baseline A (DBSCAN)	Wires	56.03	1.54	0.5372
Baseline B (Budget-Matched Non-LLM Search)	Wires	66.19	1.33	0.6420
Ours (LLM-Guided)	Wires	71.81	1.40	0.6971
Baseline A (DBSCAN)	Pipes	75.08	4.42	0.6625
Baseline B (Budget-Matched Non-LLM Search)	Pipes	74.15	3.65	0.6685
Ours (LLM-Guided)	Pipes	74.53	3.63	0.6727

For wires, the proposed method achieved the clearest advantage. The best LLM-guided solution reached 71.81% coverage with 1.40 mm dispersion and a best objective score of $J = 0.6971$. Relative to the budget-matched non-LLM search condition, which achieved 66.19% coverage and $J = 0.6420$, the proposed method recovered more wire points while preserving low fitting error. The slight increase in dispersion relative to Baseline B (1.33 to 1.40 mm) reflects the coverage–dispersion trade-off encoded in the objective; the net J-score gain of 0.0551 confirms that the additional coverage outweighs the marginal fitting cost. In addition, the number of merged output primitives decreased from 341 to 302, indicating improved continuity recovery and reduced over-fragmentation in the final reconstructed runs. Notably, fragmentation was not a term in the optimized objective J ; this improvement therefore appears as an emergent consequence of the LLM’s use of auxiliary diagnostic signals rather than direct optimization pressure.

For pipes, the gains were smaller but still consistent. The best LLM-guided solution reached 74.53% coverage with 3.63 mm dispersion and $J = 0.6727$. Compared with the budget-matched non-LLM search condition, the improvement in objective value was modest ($0.6685 \rightarrow 0.6727$), suggesting that the pipe-fitting search landscape is less ambiguous than the wire case. However, compared with the fixed DBSCAN baseline, the proposed method achieved substantially lower dispersion (4.42 mm to 3.63 mm) while maintaining comparable coverage. Notably, Baseline B achieved slightly lower coverage than Baseline A for Pipes (74.15% vs 75.08%), suggesting that undirected search can occasionally degrade fixed-parameter performance on less ambiguous classes within a limited budget.

Taken together, these results support the central claim of the paper. The LLM does not replace the geometry engine; rather, it improves the proposal stage of the search process in those settings where manual tuning would otherwise depend heavily on expert intuition.

B. Convergence Analysis

The search trajectories provide additional insight into how the proposed method behaves. On the *Wires* class, the optimization progressed from an initial score of $J = 0.6215$ in the first evaluated round to a final best score of $J = 0.6971$ by the end of the 24-round search budget. This is not a single-step jump. It is a progressive improvement across the search horizon, indicating that the proposal mechanism moved the search toward increasingly useful regions of the parameter space rather than relying on isolated luck.

By contrast, the budget-matched non-LLM search condition improved more gradually and plateaued earlier, reaching a final best score of $J = 0.6420$. The gap between the two trajectories widened in the later rounds, which is consistent with the quantitative results reported for the *Wires* class. This supports the interpretation that LLM-guided proposals are particularly useful when the reconstruction problem is dominated by fragmentation, continuity recovery, and difficult local trade-offs between clustering and merging.

For *Pipes*, the gains were smaller, which is consistent with the weaker separation observed in the final scores. Taken together, these convergence patterns suggest that the proposed method is most beneficial in the harder thin-class regime, where manual tuning would otherwise rely heavily on qualitative expert judgment.

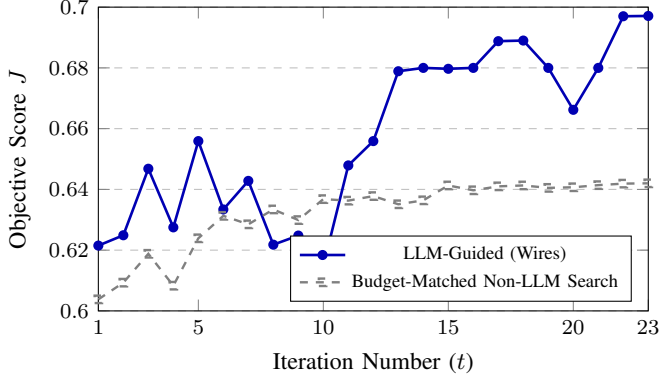


Fig. 4. Measured convergence on the *Wires* class. The LLM-guided search reaches a higher final objective value than the budget-matched non-LLM search condition and exhibits a more directed improvement trajectory over the 23 evaluated rounds.

C. Qualitative Results

Visual inspection supports the numbers. In complex areas like junctions, baseline outputs were often broken or missed curves. The LLM-optimized results show cleaner, more continuous centerlines that navigate occluded spaces more plausibly.

V. DISCUSSION

The presented results support the core hypothesis of the paper. It was posited that a portion of the heuristic reasoning normally applied by a BIM engineer during primitive-fitting calibration could be approximated by an LLM operating inside a closed-loop search process. The results indicate that this is plausible. The LLM-guided method consistently outperformed the fixed classical baseline and also improved upon the budget-matched non-LLM search condition. This suggests that the search is not purely random and not purely geometric. There is value in a proposal mechanism that can interpret compact diagnostic summaries and steer the next candidate toward more promising regions of the parameter space.

At the same time, the experiments clarify where the benefit is strongest. The largest gains occur on wires rather than pipes. The remaining uncovered wire points are likely a mixture of severe occlusion, sparsity in partially observed runs, and upstream segmentation imperfections, rather than solely a failure of the fitting loop itself. This is an important finding. Wires are the more difficult class because they are thinner, more fragmented, less regular, and more easily confused with clutter or partial observations. The stronger improvement on wires therefore indicates that the proposed framework is particularly relevant for the thin-class frontier in Scan-to-BIM automation, where manual tuning remains most labor-intensive. A natural question is whether a Bayesian optimizer

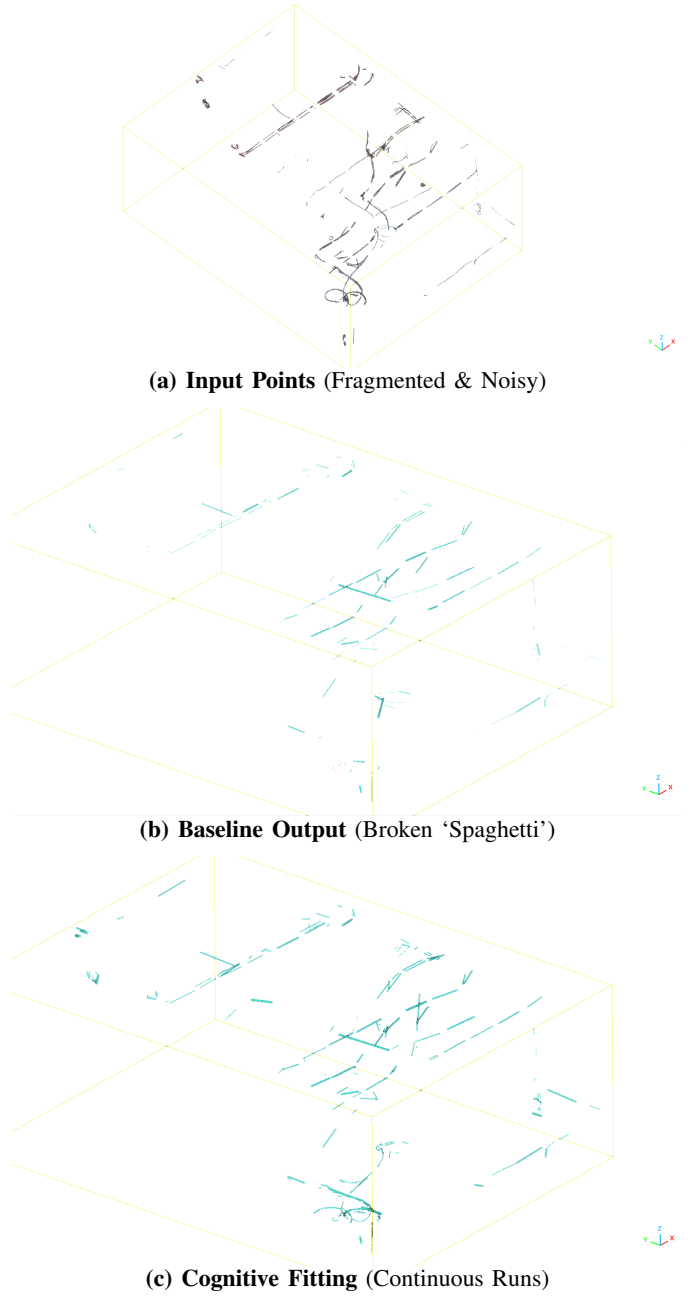


Fig. 5. Qualitative Results: Wires. By stacking the views, the continuity improvements in (c) over the baseline (b) become visible. The LLM successfully bridged gaps caused by occlusion.

would achieve similar gains within the same iteration budget. We cannot rule this out; comparison with Bayesian optimization or TPE under equivalent budgets remains an important next step. However, the budget-matched non-LLM condition serves a distinct purpose: it isolates whether LLM reasoning over geometric diagnostics provides value beyond generic iterative search of the same space, and the positive result for *Wires* suggests that the LLM’s interpretation of structured textual feedback, particularly fragmentation indicators, steers the search in ways that purely numerical methods with no domain-semantic context may not replicate as efficiently under

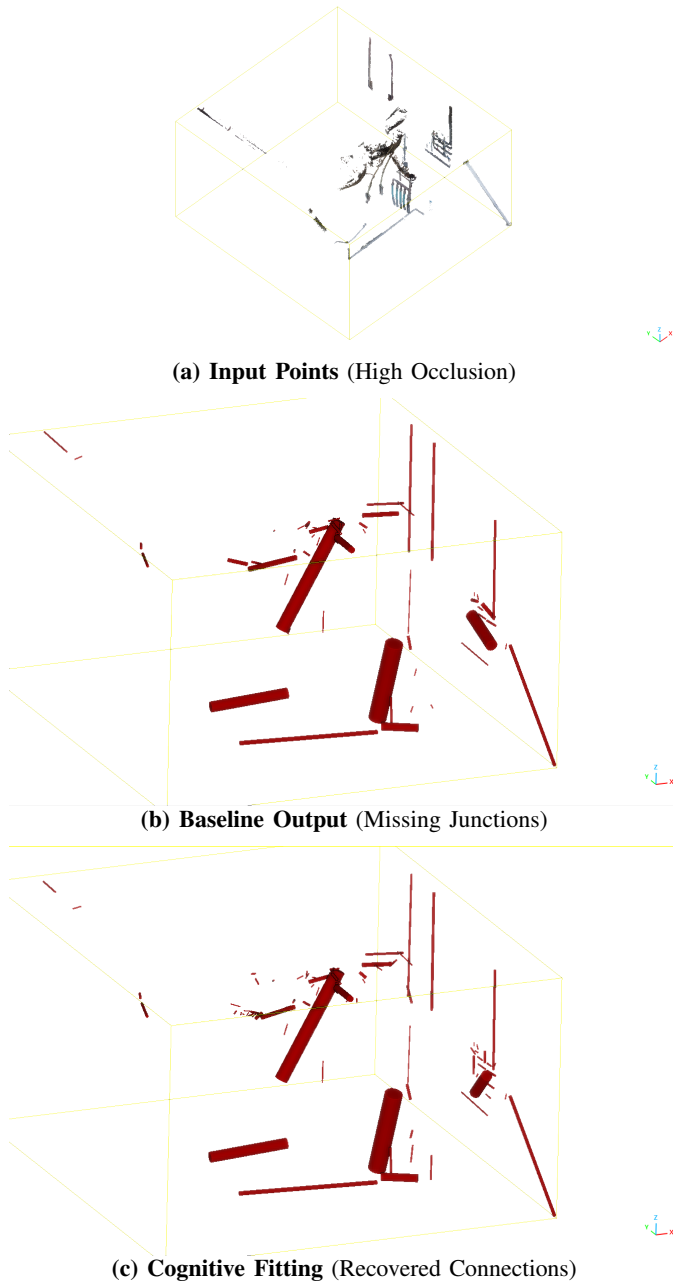


Fig. 6. Qualitative Results: Pipes. The baseline (b) fails to connect segments separated by structural framing. The proposed method (c) successfully merges these into coherent runs.

small iteration budgets.

The study also makes clear what the current method does *not* yet solve. First, the scalar objective remains intentionally simple. It trades off coverage against dispersion, while fragmentation-related measures are used only as auxiliary diagnostics and secondary quality indicators. This simplicity was useful for a first feasibility study, but it also means the objective is not a complete description of reconstruction quality. Second, the present implementation provides the LLM only with structured text feedback. This isolates the reasoning contribution, but it limits the context available to the agent.

Certain failure modes such as false merges in spatially adjacent bundles may be systematically harder to diagnose from scalar summaries alone. Visual residual maps, rendered overlays, or registered RGB cues could plausibly improve diagnosis of failure modes such as under-coverage versus false merging. Third, the current evaluation is based on a single real residential timber project. While this is a practically relevant and difficult testbed, broader cross-site validation is still needed before strong generalization claims can be made.

There are also practical considerations. The weighting in the scalar objective was selected heuristically to reflect class-specific priorities in this dataset. The low-temperature local inference setting improves reproducibility, but prompt sensitivity and repeated-run variance were not exhaustively characterized in the present paper. Finally, the use of an LLM introduces computational overhead that would not exist in a purely geometric optimization loop. A detailed runtime decomposition was not included in the present study and remains an important target for future experimental validation. As a measured reference on the hardware used in this study (RTX 4080, Qwen-2.5-14B-Instruct via LM Studio), each geometric evaluation step required a mean of 30.7 s per candidate across 116 evaluations, while each LLM inference call required a mean of 3.2 s. The full 24-round wire search completed in approximately 61 minutes, this is a one-time autotuning cost per scene class rather than a per-model overhead. For this reason, Cognitive Fitting should be viewed as a targeted autotuning stage for difficult classes and difficult scenes, rather than as a universal replacement for all simpler reconstruction tasks.

Despite these limitations, the practical implications are favorable. The framework automates a stage that is normally dominated by repeated manual trial-and-error. It does so without replacing the underlying geometry engine and without requiring large new training datasets. In that sense, it offers a pragmatic middle ground between brittle classical fitting and fully data-hungry end-to-end learning.

VI. CONCLUSION AND FUTURE WORK

This paper presented a Cognitive Scan-to-BIM framework that uses a Large Language Model as a proposal engine for hyperparameter search in post-segmentation MEP primitive fitting. The central contribution is not direct geometry generation by the LLM. Instead, the contribution lies in coupling structured geometric evaluation with LLM-guided parameter proposals inside a closed-loop optimization process.

On the challenging residential timber dataset, the method produced its strongest gains on the *Wires* class, where the best result reached 71.81% coverage with 1.40 mm dispersion and improved continuity relative to both fixed classical fitting and a budget-matched non-LLM search condition. For *Pipes*, the improvements were smaller but still consistent, with the LLM-guided method achieving the best overall objective score while reducing dispersion relative to the fixed baseline. This class-dependent behavior is itself an important result: it indicates that LLM-guided search is most beneficial in thin, fragmented, and topologically ambiguous regimes.

Several limitations remain. The current scalar objective is a practical surrogate rather than a complete multi-objective quality model. Fragmentation is used as diagnostic context but is not yet directly optimized. The LLM receives only textual feedback, not visual or multimodal evidence. The evaluation is limited to one real project dataset, and prompt sensitivity as well as runtime overhead deserve more systematic analysis.

Immediate future work will focus on four directions. First, the objective function will be extended toward a more explicit multi-objective formulation that better reflects continuity and topology preservation. Second, multimodal diagnostic feedback will be investigated by augmenting scalar summaries with rendered residuals and overlap visualizations. Third, broader validation across additional site conditions will be conducted to assess transferability beyond a single timber project. Fourth, the reconstructed primitives will be integrated into a direct Scan-to-Revit workflow through Dynamo or an equivalent API-based bridge, enabling automatic generation of native BIM elements from the optimized cylinder sets.

Overall, the results show that LLM-guided reasoning can be integrated into a classical geometric reconstruction pipeline in a technically meaningful way. This does not eliminate the need for robust geometry processing, but it does reduce the dependence on manual hyperparameter tuning in one of the most difficult parts of Scan-to-BIM automation.

REFERENCES

- [1] P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques," *Automation in Construction*, vol. 19, no. 7, pp. 829–843, 2010.
- [2] R. Volk, J. Stengel, and F. Schultmann, "Building Information Modeling (BIM) for existing buildings—Literature review and future needs," *Automation in Construction*, vol. 38, pp. 109–127, 2014.
- [3] Q. Wang and M.-K. Kim, "Applications of 3D point cloud data in the construction industry: A fifteen-year review," *Advanced Engineering Informatics*, vol. 39, pp. 306–319, 2019.
- [4] E. Agapaki and I. Brilakis, "CLOI: An Automated Benchmark Framework for Generating Geometric Digital Twins of Industrial Facilities," *Journal of Construction Engineering and Management*, vol. 147, no. 11, p. 04021145, 2021.
- [5] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.
- [6] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.
- [7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017.
- [8] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point Transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 259–16 268.
- [9] F. Bosché and C. T. Haas, "Automated retrieval of 3D CAD model objects in construction range images," *Automation in Construction*, vol. 17, no. 4, pp. 499–512, 2008.
- [10] E. Agapaki, A. Glyn-Davies, S. Mandoki, and I. Brilakis, "CLOI: A Shape Classification Benchmark Dataset for Industrial Facilities," in *Proceedings of the ASCE International Conference on Computing in Civil Engineering 2019*, 2019, pp. 66–73.
- [11] H. Yue, Q. Wang, Y. Yan, and G. Huang, "Deep learning-based point cloud completion for MEP components," *Automation in Construction*, vol. 175, p. 106218, 2025.
- [12] M. Bassier, B. Van Genechten, and M. Vergauwen, "Classification of sensor independent point cloud data of building objects using random forests," *Journal of Building Engineering*, vol. 21, pp. 468–477, 2018.
- [13] B. Romera-Paredes *et al.*, "Mathematical discoveries from program search with large language models," *Nature*, vol. 625, pp. 468–475, 2024.
- [14] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen, "Large language models as optimizers," *arXiv preprint arXiv:2309.03409*, 2023.
- [15] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010.
- [16] W. Wang, J. Peng, M. Hu, W. Zhong, T. Zhang, S. Wang, Y. Zhang, M. Shao, and W. Ni, "LLM Agent for Hyper-Parameter Optimization," *arXiv preprint arXiv:2506.15167*, Jun. 2025.
- [17] N. van Stein, D. Vermetten, and T. Bäck, "In-the-loop Hyper-Parameter Optimization for LLM-Based Automated Design of Heuristics," *ACM Transactions on Evolutionary Learning and Optimization*, 2025.