Text-to-Code Generation for Modular Building Layouts in Building Information Modeling

Yinyi Wei, Xiao Li
The University of Hong Kong

Content

01
Background

Background

Text-based generative design

Leveraging NLP and generative AI to create building layouts based on user-provided voice- or text-based descriptions

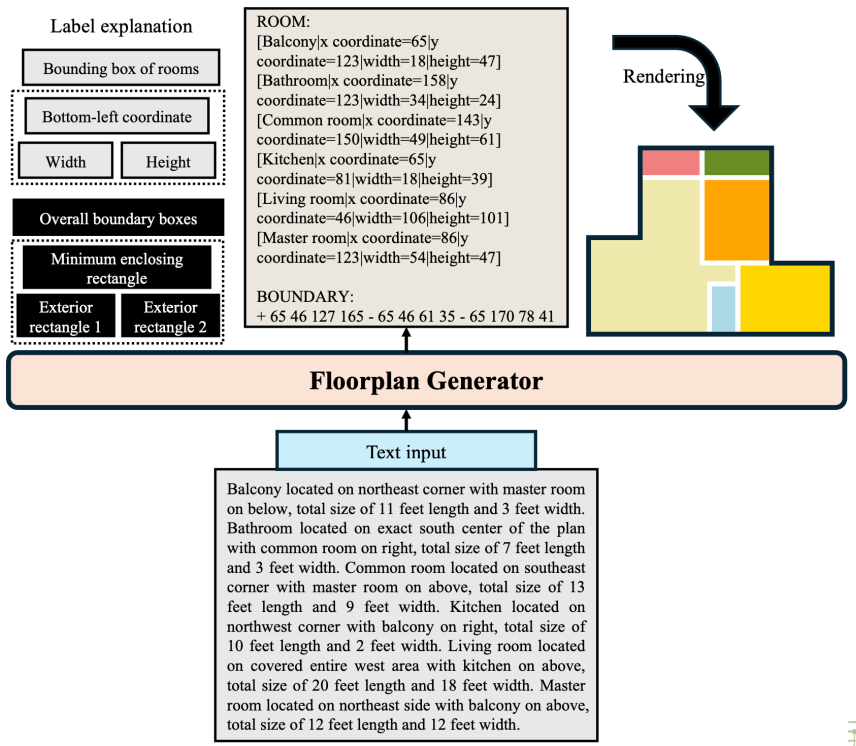
Merits

- Accessibility
- Speed
- Customization
- Innovation

Challenges

- Modality gap
- Data deficiency
- Ambiguity
- Diversity

Seq2Seq pipeline for text-based generation (Wei et al., 2024)



Wei Y, Li X. Graph-Augmented Text-Based Floorplan Generation[C]//2024 International Conference on Automation in Manufacturing, Transportation and Logistics (ICaMaL). IEEE, 2024: 1-9.

Background

Modular construction

Building components are manufactured offsite and transported to the construction site for assembly

Features

- Factory fabrication
- On-site assembly
- Design flexibility
- Time and cost efficiency
- Quality control

Conventional construction

Modular construction

Background

Modular building layout (MBL)

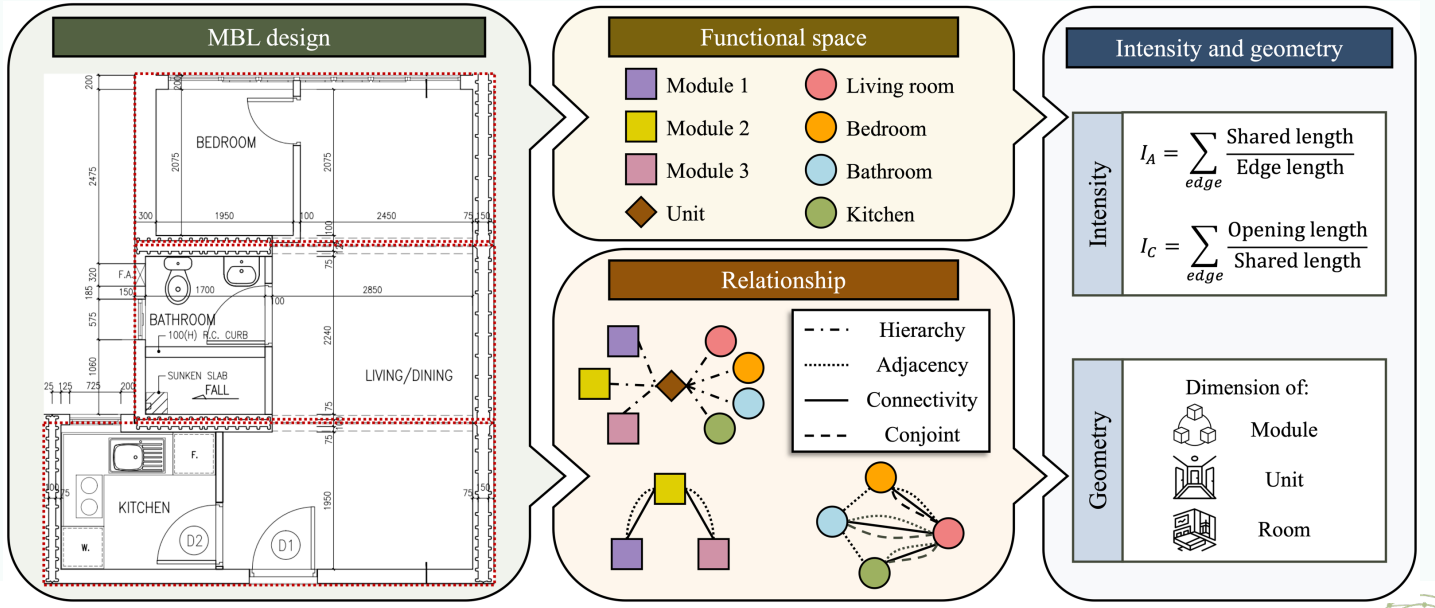
Design and spatial arrangement of 3D volumetric units, which may include several modules and units within a layout

Challenges

- Design constraints
- Limited flexibility
- Complexity
- Need for new design philosophies

Key elements of MBL

- Functional space
- Relationship
  - Adjacency
  - Connectivity
  - Conjoint
- Intensity



CI³ Lab Collaborative Intelligence

Background

Building information modeling (BIM)

Digital, object-oriented representation capturing physical and functional aspects (e.g., materials) across building lifecycle

Features

- 3D parametric modeling
- Lifecycle integration
- Multi-disciplinary collaboration
- Data-rich components
- Visualization & simulation & analysis

BIM vs CAD

Aspect	BIM	CAD
Model type	3D object-based	2D drawing-based
Data content	Geometry + semantic + performance+ metadata	Geometry only
Lifecyle coverage	Full lifecycle (design → build → operate)	Design & documentation
Changes	Parametric and automatic updates	Manual revisions
Coordination	Multi-disciplinary, collaborative	Isolated files per discipline

02
Research Gaps and Objectives

Research Gaps

Text to BIM-based MBL

Translating user natural language descriptions into MBL in Building Information Modeling

Image-based generation: Generates 2D pixel-level floorplans or layouts	1. Lacks semantic structure and parametric information; 2. Not directly compatible with BIM workflows, requiring error-prone post-processing; 3. Difficult to extract precise geometry or relationships for downstream tasks (e.g., quantity takeoff, simulation)
Coordinate-based generation: Outputs bounding boxes of BIM components	1. Relies on absolute coordinates, making it sensitive to layout scale and ordering; 2. Struggles with complex spatial/topological constraints (e.g., adjacency, connectivity, hierarchy); 3. Requires careful conflict resolution (e.g., overlaps, gaps)
IFC-based generation: Directly synthesizes IFC files—the standard BIM data schema	1. IFC schema is highly verbose, redundant, and low-level; 2. Encodes not just geometry but also lifecycle metadata, relationships, and properties—making generation extremely complex; 3. Hard for generative models (e.g., LLMs) to learn due to structural rigidity and lack of high-level abstractions; 4. Minor syntax or semantic errors often render the IFC file invalid or unusable in BIM software

Our solution

Code-based generation: Generates executable, high-level BIM code that can be directly executed in a BIM environment to produce a valid, parametric model	1. Produces models that are immediately compatible with industry-standard BIM platforms; 2. Uses relative positioning and high-level abstractions instead of error-prone absolute coordinates; 3. Built on reusable classes and functions, making it easy to extend with new components or design rules; 4. Frames layout generation as a sequence-to-sequence code generation task, leveraging the strong syntactic and structural understanding of LLMs; 5. Code output is human-readable, enabling designers to inspect, edit, or correct the generated logic

Research Objectives and Contributions

香 港 大 學
THE UNIVERSITY OF HONG KONG

Research Question

Research question

How can natural language instructions be effectively translated into MBLs within a BIM environment?

Research Aim

Research Aim

To develop an approach that generates BIM-based MBLs from natural language instructions.

Research Objectives

Objective 1

To formulate the concepts of BIM-based MBLs following MBL principles.

Objective 2

To curate data pairing natural language instructions with action-based instructions.

Objective 3

To learn the mapping relationships between texts and BIM-based MBLs.

Contributions

Text2MBL

A text-to-code generation framework to translate texts into executable, semantically rich BIM code for MBLs.

Object-oriented BIM code architecture

An object-oriented BIM code framework (Module/Unit/Room/Utils classes) with high-level APIs for layout construction via sequential actions.

Data curation

198 real MBL designs were collected and annotated; synthetic data (partial & full) was generated to augment training.

Methodology

Seq2Seq problem formulation and open-source LLMs fine-tuning.

Proof-of-concept

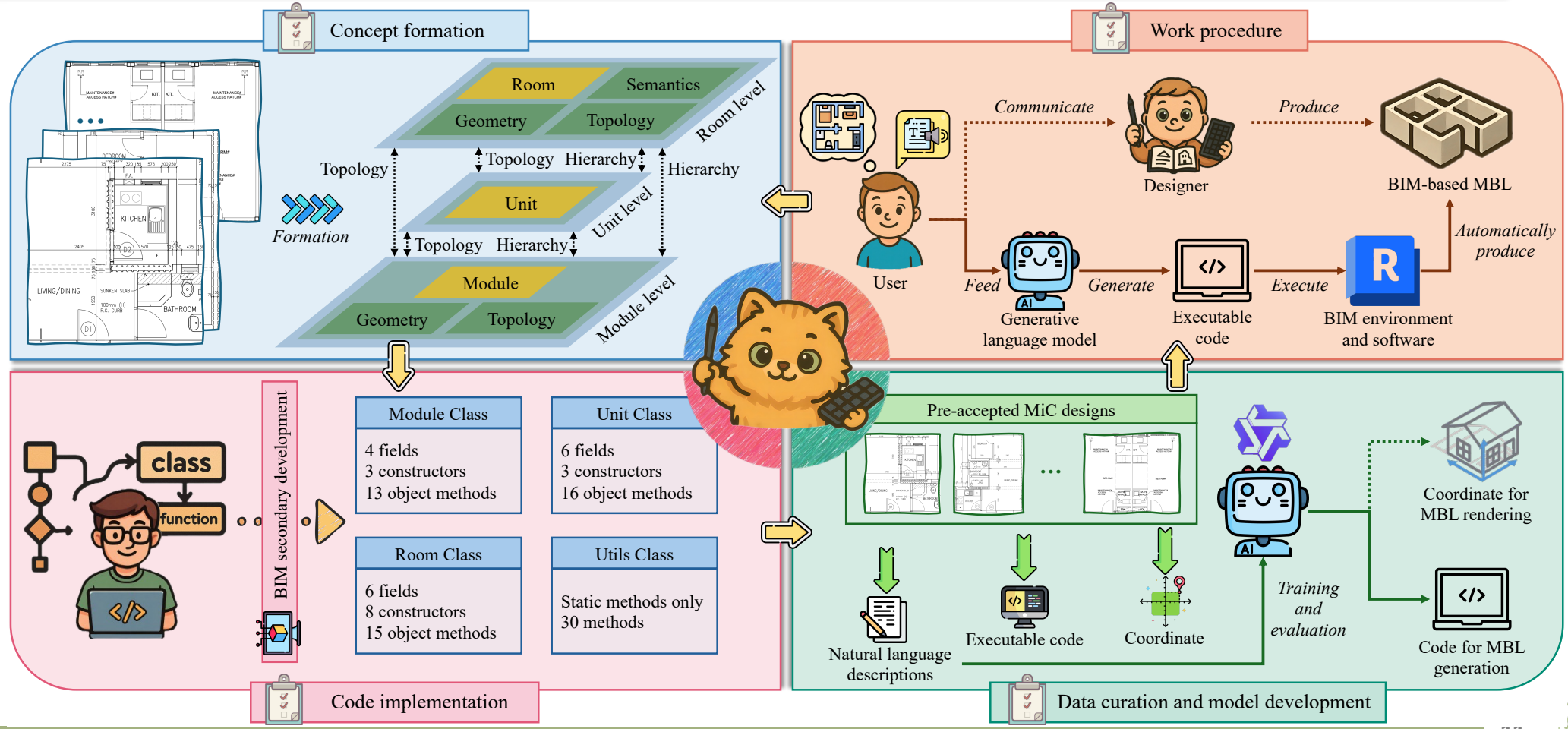
A proof-of-concept system is developed as a plug-in for Autodesk Revit using its API.

CI³Lab Collaborative Intelligence

Research Framework

Text to BIM-based MBL

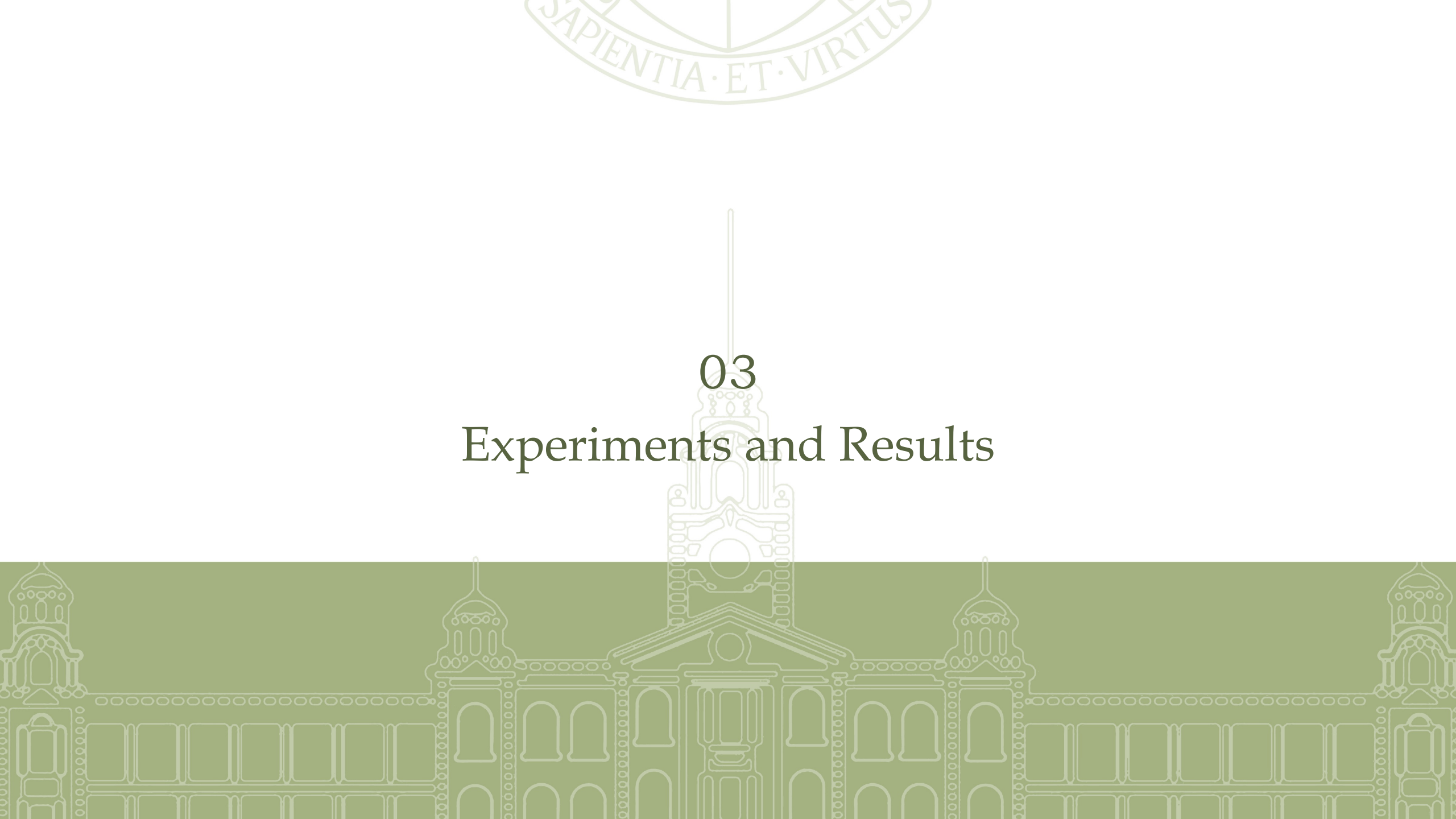
Translating user natural language descriptions into MBL in Building Information Modeling



Research Framework

Code-based actions in Text2MBL, exemplified via relevant MBLs, code snippets, and BIM formats

Action	Method	MBL (E.g.)	Code (E.g.)	BIM (E.g.)
Module creation	Absolute coordinate		*Module module = new Module(name: "Module", point: initial_point, length: 2800, width: 6880);*	
	Relative position		*Module module_3 = new Module(name: "Module 3", module: module_1, direction: "south", length: 2240, width: 1620, alignment: "east", offset_direction: "west", offset: 2000);*	
Module operation	Module split		*List<Module> new_modules = Utils.SplitModule(module: module_2, direction: "west-east", ratio: 0.5);\nModule module_2_north = new_modules[0];\nModule module_2_south = new_modules[1];*	
	Module merging		*Utils.MergeModules(modules: new List<Module> module_1, module_2);*	
Unit initialization	Combination of modules		*Unit unit_1 = new Unit(name: "Unit 1", modules: new List<Module> module_1, module_2_north);*	
	Combination of directional modules		*Unit unit_1 = new Unit(name: "Unit 1", modules: new List<Module> module_1, module_2, module_3 , direction: "north", dimensions: new List<double> 5800, 5800, 5800);*	
Room assignment	Module/Unit-based		*Room living_room = new Room(name: "Living Room", module: module, unit: unit, regular: false);*	
	Diretion-oriented		*Room kitchen = new Room(name: "Kitchen", module: module, unit: unit, direction: "south", dimension: 1800, open: true);*	
	Corner-oriented		*Room kitchen = new Room(name: "Kitchen", module: module, unit: unit, corner: "southwest", length: 1600, width: 1200, offset_direction: "none", offset: 0, open: true);*	
	Relative		*Room kitchen = new Room(name: "Kitchen", unit: unit, room: bathroom, direction: "east", length: 1640, width: 1220, alignment: "north", offset_direction: "none", offset: 0, open: false);*	
Element placement	Door		*Utils.CreateDoorForRoom(room: bedroom_3, direction: "west", alignment: "south", offset: 0, set: "in", set_dimension: 600);*	
	Hole		*Utils.CreateHole(module: module_2, direction: "north", alignment: "none", offset: 0, dimension: 2000);*	



03
Experiments and Results

Experiments and Results

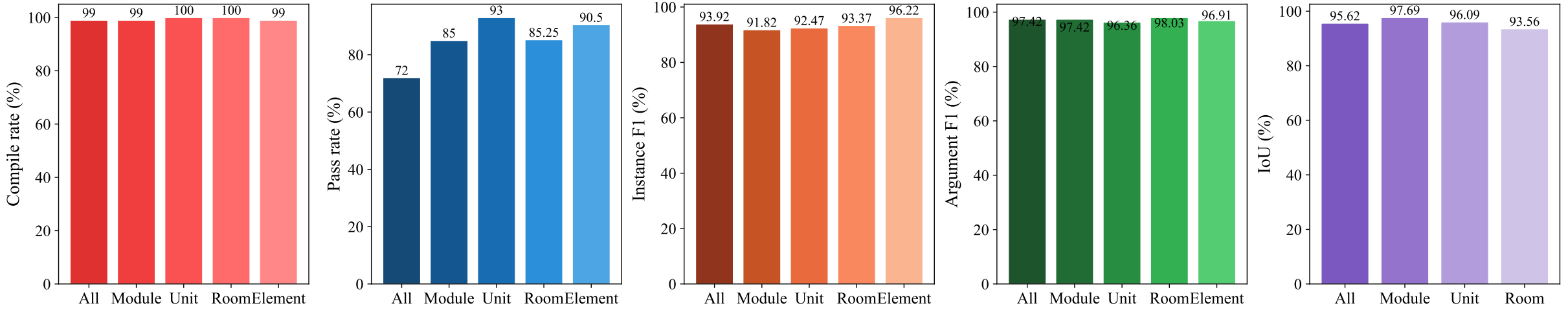
Effect of output formats (code or coordinate) on on geometric consistency metrics

Model	IoU (%)	Module IoU (%)	Unit IoU (%)	Room IoU (%)
Qwen2.5-Instruct				
1.5B - Coordinate	78.13	86.98	80.27	69.85
1.5B - Code	91.65	95.47	93.79	86.79
7B - Coordinate	85.38	90.35	90.39	77.33
7B - Code	95.83	98.51	98.11	91.64
Qwen2.5-Coder-Instruct				
1.5B - Coordinate	78.33	87.71	80.20	69.85
1.5B - Code	94.19	97.01	95.59	90.45
7B - Coordinate	84.64	90.12	88.18	77.53
7B - Code	98.43	98.78	99.19	97.37
Qwen2.5-Math-Instruct				
1.5B - Coordinate	78.61	87.03	85.09	67.57
1.5B - Code	94.18	97.49	97.17	89.06
7B - Coordinate	85.34	90.32	89.51	78.02
7B - Code	94.86	98.07	97.46	89.86

Compared with coordinate-driven approach, code-driven approach demonstrated superior geometric consistency (IoUs) in generating MBLs.

Experiments and Results

Component-level performance across (1) compile rate, (2) pass rate, (3) instance F1, (4) argument F1 and (5) IoU



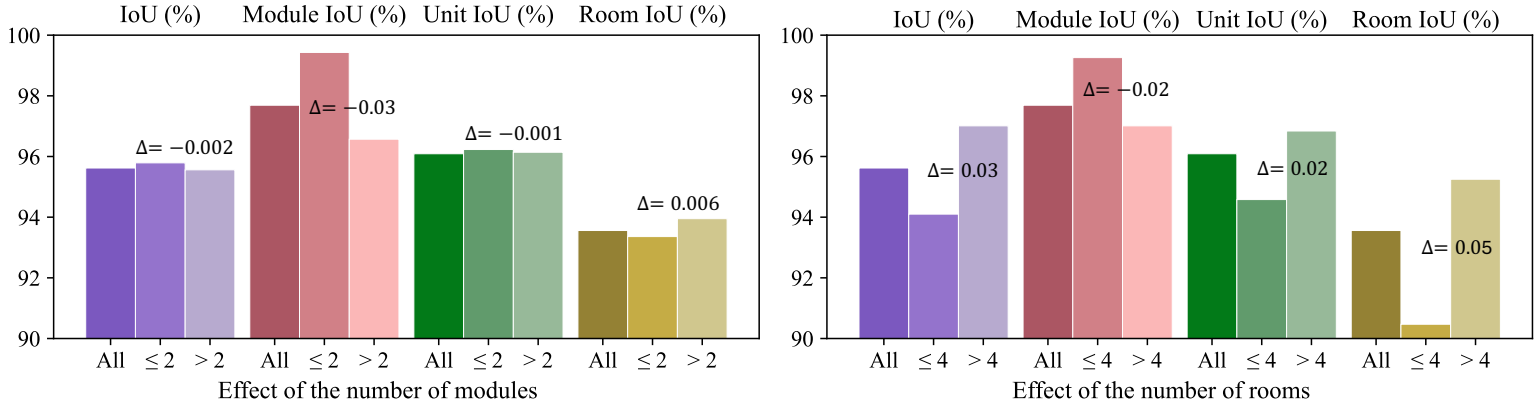
The semantic fidelity metrics (i.e., instance F1 and argument F1 scores) achieved better results than the pass rate metric, suggesting that models were more effective at extracting argument information than at producing perfectly executable code without deviations.

Experiments and Results

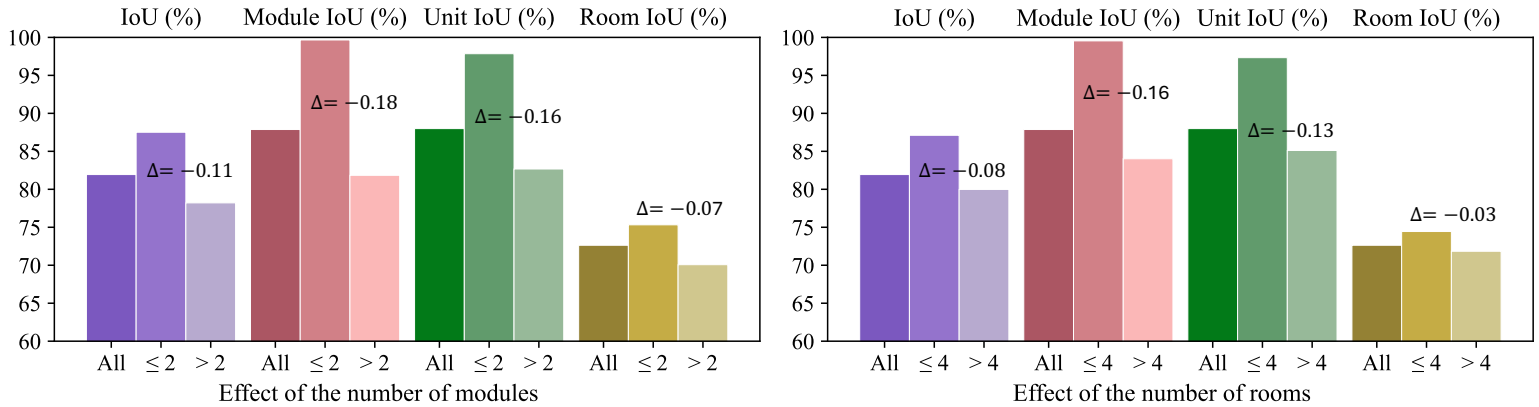
香 港 大 學
THE UNIVERSITY OF HONG KONG

Effect of the number of modules and rooms on geometric consistency metrics

Code output



Coordinate output



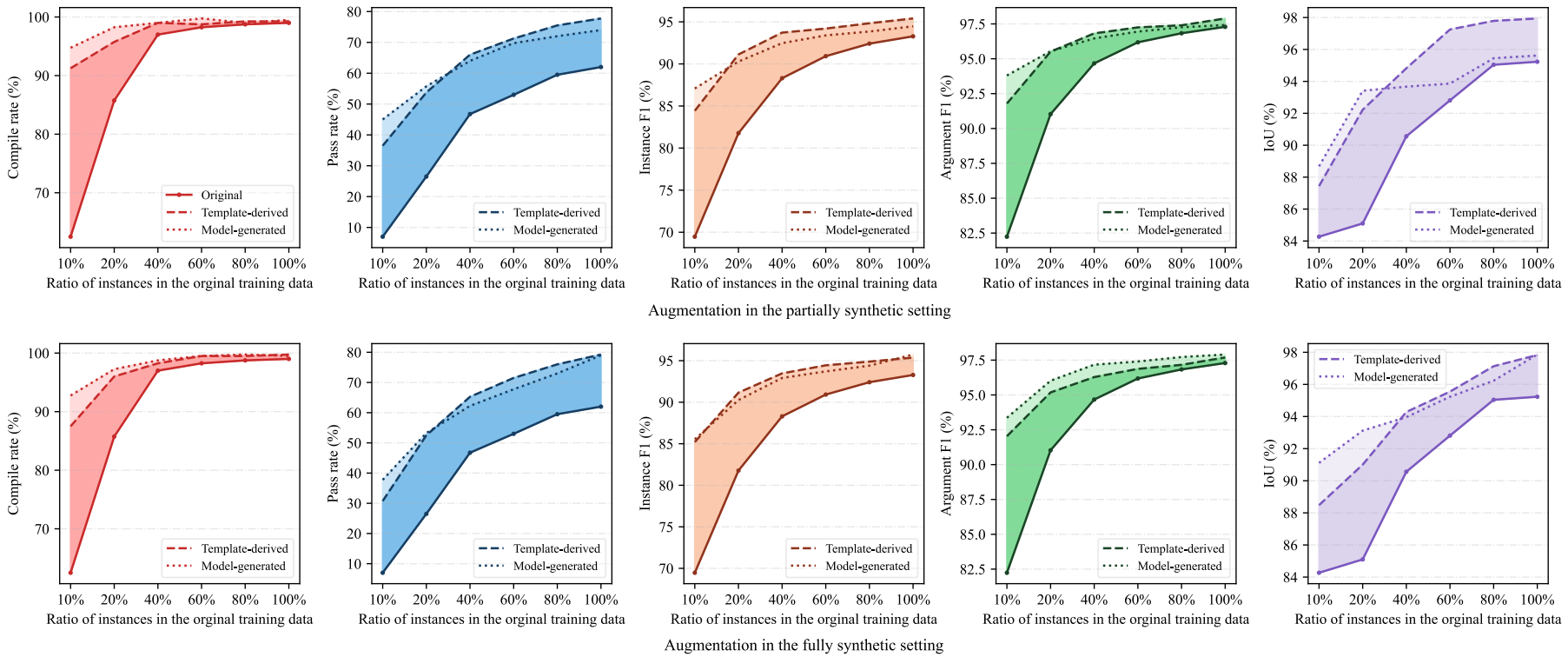
Increasing the number of components led to a sharper performance decline in the coordinate sequence output compared to the code output. This difference can be attribute to the hierarchically designed code architecture, which replaces spatial reasoning with semantic understanding relying on relative position, delivering more robust results.

CI³ Lab Collaborative Intelligence

Experiments and Results

香 港 大 學
THE UNIVERSITY OF HONG KONG

Ratios of original training data augmented with synthetic data
(the corresponding performance improvements for metrics are highlighted)



Although the synthetic data exhibited suboptimal performance deal to limitations such as overly rigid description logic, it may still be beneficial as an augmentation when combined with the original training data.

CI³ Lab Collaborative Intelligence

香 港 大 學

THE UNIVERSITY OF HONG KONG

Thank You