URBLEND: URBAN BUILDING MODEL SYNTHESIS FROM INCOMPLETE OBSERVATIONS WITH GENERATIVE PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Building reconstruction often relies on point clouds from LiDAR or visual reconstruction frontends. Noise, occlusion, and uneven sampling make these observations incomplete and uncertain. Recovering complete geometry from such sparse evidence is underconstrained, motivating pretrained shape priors. To support learning and evaluation in this setting, we introduce CityEXP, a scalable data-construction pipeline and benchmark protocol. Agent-coordinated procedural city generation preserves building identities and detailed geometry; high-fidelity scene rendering and visual reconstruction produce incomplete point observations paired with these targets. Our baseline, UrBlend, adapts Hunyuan3D-Omni's pretrained shape prior to building observations. Prepared surfaces supply latent training targets across compatible geometry sources. A pretrained surface decoder produces meshes; separately trained structural decoders produce wireframes after task adaptation. On CityEXP, the baseline achieves canonical-shape $F_{.02}$ of 54.38% and Chamfer distance of 26.34×10^{-3}. CityEXP and UrBlend establish a benchmark and unified baseline for prior-guided building reconstruction.

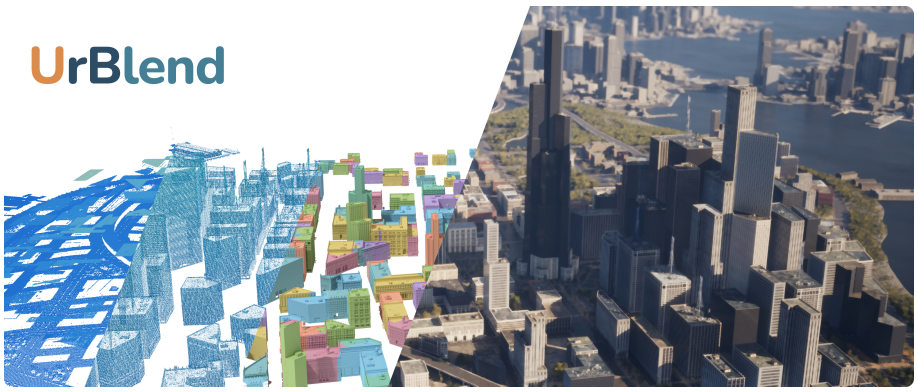

Figure 1: **Data construction for UrBlend.** CityEXP pairs reconstructed point observations (left) with retained building geometry (center) using rendered urban scenes (right).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Urban building reconstruction converts spatial observations into models for visualization, planning, and simulation (Liu et al., 2024; Hua et al., 2026). We study *urban building model synthesis*: inferring an individual building's complete exterior geometry from sparse, incomplete point observations (Figure 1). Such point clouds can come from LiDAR acquisition, including airborne surveys such as AHN (Actueel Hoogtebestand Nederland, n.d.), or from visual reconstruction frontends. Measurement and reconstruction errors perturb the observed points, while occlusion, limited viewpoints, and uneven sampling leave parts of roofs and facades sparsely observed or missing (Liu et al., 2024; Huang et al., 2022). Recovering complete geometry therefore requires reasoning beyond the available points.

Learning this mapping depends on the available observation–model pairs. Large resources such as 3DBAG and BuildingWorld provide building models at specified levels of detail (LoD) (Peters et al., 2022; Huang et al., 2026). BuildingWorld also supports scalable synthetic LiDAR acquisition. Our focus is high-fidelity visual acquisition: detailed building geometry, facade materials, vegetation, and lighting shape the image evidence from which observations are reconstructed. Simplified targets omit fine geometry, while references reconstructed from the same views can inherit observation gaps. Learning detailed reconstruction requires these visual observations paired with independently retained building geometry.

We introduce CityEXP to address this data-construction problem. Agents coordinate layout plans, reusable assets, and native Unreal procedural content generation (PCG). The pipeline preserves each building's identity and component geometry through generation, scene assembly, and export. Rendered views include architectural and environmental detail; visual reconstruction supplies incomplete point observations paired with the retained reference meshes. Buildings therefore retain complete targets even where the rendered views provide little or no evidence (Figure 2).

Sparse observations also motivate a generative prior: several complete buildings may agree with the same visible evidence. Pretrained shape generators learn geometric regularities from diverse assets (Li et al., 2025b; Xiang et al., 2025; Tencent Hunyuan3D Team, 2025a). We use Hunyuan3D-Omni, which already supports incomplete and noisy point controls (Tencent Hunyuan3D Team, 2025b), to provide this prior and a common latent training framework. Our baseline, UrBlend, adapts its point-conditioned generator using native and multiscale Utonia features (Zhang et al., 2026). Prepared surfaces provide latent supervision across source meshes with different connectivity. A surface decoder produces meshes; separately trained structural decoders map the latent format to architectural wireframes after task adaptation.

We evaluate the baseline on CityEXP, with complementary conditioning studies on BuildingWorld3D and structural evaluation on MunichWF. Regional analysis examines the tension between generating missing surfaces and preserving observed geometry.

We make three contributions:

1. CityEXP, a scalable pipeline that pairs high-fidelity urban visual acquisitions with detailed building geometry through agent-coordinated PCG generation and persistent instance identity.
2. UrBlend, a Hunyuan-based generative reconstruction baseline with a common geometry-supervision interface and task-adapted decoders that connect point-conditioned latent generation to surfaces and structure.
3. A benchmark protocol and diagnostic evaluation on CityEXP, characterizing missing-surface recovery and observation fidelity, with complementary conditioning and structural studies.

2 RELATED WORK

Urban data and paired supervision. Learning to reconstruct buildings requires observations paired with geometry that defines the desired output detail. BuildingWorld and 3DBAG provide large-scale building models at specified levels of detail (Huang et al., 2026; Peters et al., 2022). BuildingWorld's Cyber City emphasizes structural geometry and synthetic LiDAR acquisition with

scene occlusion. Compared with its illustrated LoD-based scenes, CityEXP provides richer visual appearance through detailed facade components, materials, vegetation, and lighting for image-derived observations paired with instance geometry. MatrixCity and UrbanScene3D supply scene-level visual observations; SUM supplies reconstructed urban meshes, and CARLA supports controllable simulation (Li et al., 2023; Lin et al., 2022; Gao et al., 2021; Dosovitskiy et al., 2017). CityEXP retains per-building source geometry and identity throughout procedural construction, then pairs these targets with points reconstructed from scene-level images.

Building reconstruction and geometric outputs. Sparse observations leave building surfaces underdetermined, motivating both geometric constraints and learned shape regularities. Planar methods combine primitive fitting with structural constraints to recover compact envelopes (Nan & Wonka, 2017; Huang et al., 2022); Point2Mesh exploits repetition within the observed shape (Hanocka et al., 2020). Learned completion predicts missing points (Yuan et al., 2018; Yu et al., 2021), while signed-distance and occupancy fields represent continuous surfaces (Park et al., 2019; Mescheder et al., 2019). Building-specific methods predict roofs, wireframes, or polygon meshes (Li et al., 2022; Huang et al., 2024; Liu et al., 2025b; 2024; 2026). Wireframes require corner and edge labels; mesh-sequence generators require valid connectivity and tractable target sequences (Chen et al., 2024a;b). BuildAnyPoint completes points and normals before autoregressive meshing; OCCDiff generates implicit occupancy geometry through latent diffusion (Hua et al., 2026; Sui et al., 2025). Our baseline samples prepared surfaces to obtain latent targets, avoiding the need for a common mesh connectivity across sources. Structural decoder training still requires graph labels.

Pretrained generative priors. Latent shape models separate surface representation from generative learning (Zhang et al., 2023; Li et al., 2025b; Xiang et al., 2025). Pretrained asset generators learn geometric regularities that can inform completion where observations provide little evidence (Tencent Hunyuan3D Team, 2025a; Li et al., 2025a). GenPC transfers generative priors through depth prompting and geometric fusion; Hunyuan3D-Omni provides explicit control from incomplete and noisy point clouds (Li et al., 2025a; Tencent Hunyuan3D Team, 2025b). Our baseline retains Hunyuan's native point controls and adds multiscale Utonia conditioning (Zhang et al., 2026). We study how this adaptation affects building reconstruction, including completion and observation fidelity on CityEXP.

Table 1: **Selected urban scene resources and reconstruction supervision.** The two panels are independent comparisons; rows do not denote dataset–method pairs.

Urban scene resources				**Reconstruction supervision**			
Source	Scalable generation	Native GT	Scene acq.	Method	Mesh	Dense	Wire
UrbanScene3D	✗	✗	✓	BuildingGPT	✗	✗	✓
MatrixCity	✗	✗	✓	Point2Building	✓	✗	—
SUM / Helsinki	✗	✗	✗	BuildAnyPoint	✓	✗	—
CARLA	✓[e]	✗	✓	OCCDiff	✓[g]	✓[g]	—
CityEXP (Ours)	✓	✓	✓	**Ours**	✓[g]	✓[g]	✓[c]

✓: available or compatible after preparation; ✗: unavailable; —: conversion unverified.

Data: Scalability means procedural city generation; native GT requires paired per-building source meshes, excluding reconstructions. Scene acquisition preserves shared context and occlusion. [e]Experimental generation (CARLA Team, n.d.). CityEXP reports its implemented pipeline (Appendix A.3).

Supervision: Mesh denotes surface meshes without wireframe labels; dense denotes detailed surface targets without a compact mesh representation; wire denotes vertices and edges without faces. All assume paired observations. Autoregressive targets must meet topology and size limits. [g]Requires validity checks and surface/occupancy sampling. [c]Requires recoverable faces, otherwise repair or rejection. Structural heads still require graph labels.

3 TASK FORMULATION

We consider an individual building with a known instance association. The input is a sparse, incomplete point cloud $P_b = \{x_i\}_{i=1}^{n_b} \subset \mathbb{R}^3$, obtained from scanning or an upstream image-reconstruction system. The primary output is a complete exterior surface mesh:

$$\widehat{\mathcal{M}}_b = F(P_b; \xi), \tag{1}$$

where ξ controls stochastic generation. For image-derived inputs, images and camera poses serve upstream reconstruction and localization; the generator receives the building's point condition.

Training associates P_b with a reference surface $\mathcal{M}_b^\star$. Multiple acquisitions provide different observations of one target,

$$\mathcal{D} = \{(\{P_b^{(k)}\}_{k=1}^{K_b}, \mathcal{M}_b^\star)\}_{b=1}^{B}, \tag{2}$$

whereas a detailed mesh and a simplified LoD envelope define different targets. Unobserved regions admit several plausible completions; evaluation measures agreement with the designated reference surface. Architectural wireframes form a separate structural readout.

Coordinates. Observations and targets are paired in scene coordinates. A target-derived crop defines a known-box setting; independent normalization of the cloud and target defines the canonical-shape comparison used in the surface experiments. Appendix A gives the coordinate transform. Section 6 specifies each setting; canonical-shape comparison and world-coordinate reconstruction measure different properties.

4 CITYEXP: PROCEDURAL DATA CONSTRUCTION AND BENCHMARK

CityEXP addresses the limited availability of paired incomplete observations and detailed building geometry. Its procedural pipeline constructs urban scenes while retaining the geometry of each building, then acquires observations through high-fidelity scene rendering (Figure 2). This separates the complete supervision target from the sparse, occluded evidence available to a reconstruction model and provides data for adapting and evaluating generative priors.

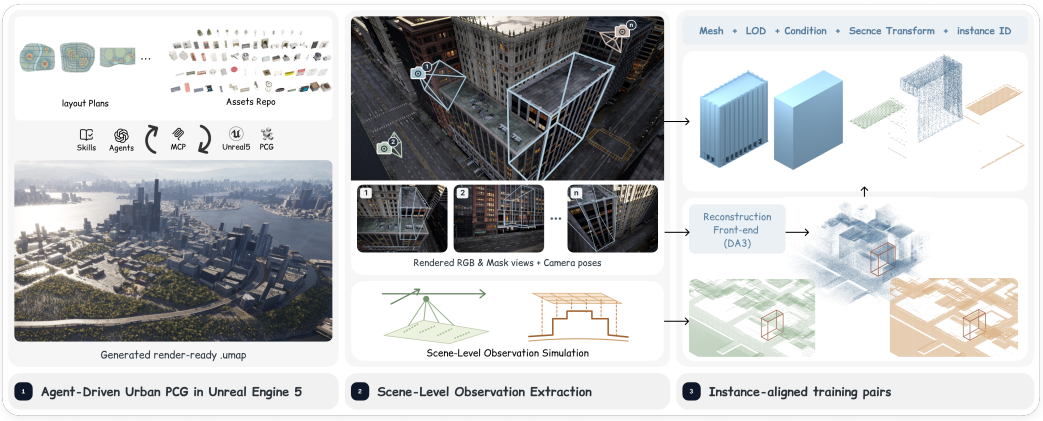

Figure 2: **Data pipeline.** Agents coordinate layout plans, reusable assets, and native PCG generation. The evaluated baseline uses DA3 point conditions reconstructed from rendered multi-view images with known camera poses. Geometric scene sampling is a separate observation route. Instance identities and coordinate transforms associate these conditions with reference meshes and structural LoD proxies.

Agent-assisted scene construction. Agents coordinate layout and height plans, reusable asset selection, native generation, and verification. Unreal PCG (Epic Games, n.d.b) generates buildings from fixed footprints and height settings using City Sample assets (Epic Games, n.d.a). Separate

context branches add parking, vegetation, plazas, and waterways; roads and static furnishings complete the scene. Building generation runs in independent batches, which are assembled and checked after saving and reopening. This organization supports extending the collection through additional layouts and generation batches while preserving instance identity. Appendix A.1 provides the skill index and execution protocol, including saved artifacts and verification steps.

Geometry retained at generation time. Each retained building has a record of its building and generation identities, actual component geometry, transforms, and generated roof and bottom-cap geometry. These records directly connect the scene instance to its exported reference mesh. Rendering and geometry export consume the same frozen generation. Detailed meshes and structural LoD proxies remain distinct targets. We focus on exterior surface geometry and use ManifoldPlus (Huang et al., 2020) to prepare detailed component meshes as watertight surfaces before sampling and latent encoding (Section 5.1). Geometric changes introduced by preparation are recorded.

Scene-level observations. Rendering retains detailed facade geometry and materials, neighboring buildings, vegetation, scene lighting, and atmospheric effects. These visual factors shape the image evidence used by reconstruction systems such as VGGT and DA3 (Wang et al., 2025; Lin et al., 2025). The baseline uses DA3 to reconstruct rendered RGB images with known Unreal camera poses. Its point conditions contain incomplete coverage and reconstruction errors, while target geometry remains independent of visibility. Appendix A.3 distinguishes this route from geometric surface sampling.

Pairing and benchmark protocol. Instance identities and transforms associate observations with reference geometry in a common scene coordinate frame. Quality assessment checks instance association, target completeness, coordinate consistency, and usage rights. The evaluated cohort is frozen before training and divided by building identity, stratified by scene, into training, validation, and test sets. Building, generation, and exact mesh identities are disjoint; acquisition pools and source asset families are shared across splits. Methods within an input setting receive the same observations and localization information. Target-derived localization defines a known-box setting. The generator uses points; image reconstruction belongs to observation preparation. Experiment-specific normalization is described in Section 6.

Collection scope. The enhanced collection contains four generated cities with 9,269 retained building instances. This count measures the scene inventory. We establish a reconstruction baseline on a fixed cohort drawn from the baseline City 01 and enhanced Cities 02–04. Cohort admission requires a target encode–decode F-score of at least 0.9 against the prepared surface. Scene inventory and admitted reconstruction pairs are separate counts. Appendix A specifies pairing metadata, and Appendix B.1 gives the cohort and split details.

5 URBLEND: A GENERATIVE RECONSTRUCTION BASELINE

UrBlend adapts Hunyuan3D-Omni to predict complete-building latents from observed points (Figure 3). A frozen surface encoder supplies the latent training targets. Given a point cloud P^o in its local frame, the model samples a latent and decodes its surface:

$$\widehat{z} \sim p_\phi(z \mid C(P^o)), \qquad \widehat{\mathcal{M}} = D_{\mathrm{surf}}(\widehat{z}). \tag{3}$$

Here ϕ denotes the adapted parameters, and D_{surf} is the pretrained ShapeVAE surface decoder with isosurface extraction.

5.1 SHARED SUPERVISION FROM HETEROGENEOUS GEOMETRY

We use exterior surfaces to supervise geometry from different source representations. Let G_b denote a LoD model, triangle mesh, recoverable wireframe, or CAD-derived boundary surface. Shell construction and surface preparation produce a watertight mesh for each accepted input:

$$\bar{\mathcal{M}}_b = \mathcal{W}(\mathcal{H}(G_b)), \qquad z_b^\star = E_{\mathrm{shape}}\big(\mathcal{S}_{50/50}(\bar{\mathcal{M}}_b)\big)\,. \tag{4}$$

$\mathcal{H}$ constructs the exterior shell, and $\mathcal{W}$ validates the surface and applies ManifoldPlus repair when necessary (Huang et al., 2020). Admissible surfaces bypass repair; failures are recorded. Wireframes

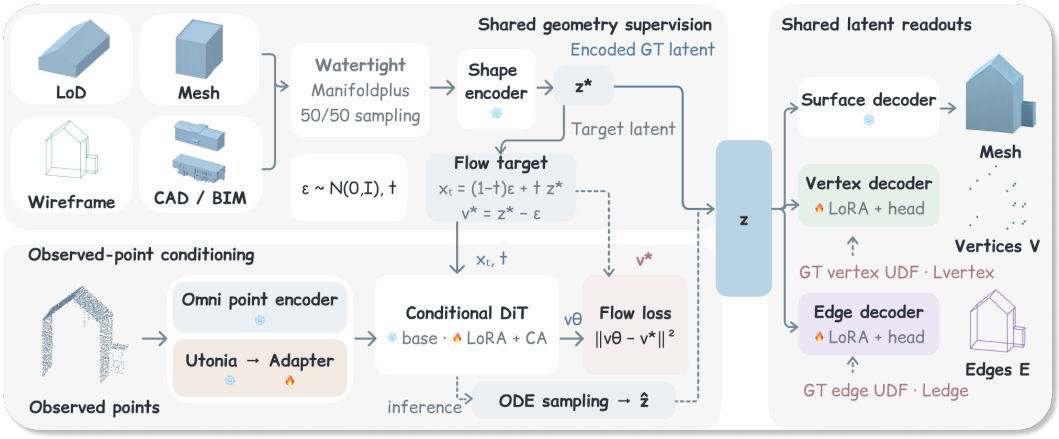

Figure 3: **Point-conditioned flow-matching training.** Target branch: watertight surfaces are sampled and encoded by a frozen shape encoder; wireframes require recoverable faces. Observation branch: native and Utonia pathways condition the DiT, supervised by $z^\star - \epsilon$. Snowflakes/flames mark frozen/trainable parameters. The right panel shows generated-latent readouts for a MunichWF building from the separate task-adapted Utonia-replacement model; its vertex and edge decoders train separately on target latents and structural labels.

require recoverable faces. The prepared mesh $\bar{\mathcal{M}}_b$ defines the encoder target; evaluation specifies whether $\mathcal{M}_b^\star$ is the prepared or original surface. $\mathcal{S}_{50/50}$ combines 50% area-uniform samples and 50% sharp-edge samples, with normals, to represent surfaces and architectural boundaries.

Sampling avoids matching vertex ordering or connectivity, while keeping LoD envelopes and detailed meshes as distinct targets. Conditional training requires paired observations; structural readouts require graph labels. Appendices D.1 and G detail encoding and geometric losses.

5.2 Native and Multiscale Point Conditioning

We retain Hunyuan3D-Omni's native point pathway (Tencent Hunyuan3D Team, 2025b) and augment it with a frozen Utonia encoder (Zhang et al., 2026); both process the same observed cloud (Figure 4). The adapter combines projected Utonia features H_ℓ from five hierarchy levels with Fourier encodings of positions X_ℓ and learned scale embeddings. A two-layer Transformer maps these features to additional condition tokens:

$$C_{\text{native}} = E_{\text{native}}(P^o), \qquad C_{\text{ut}} = A_\phi(\{H_\ell, X_\ell\}_{\ell=0}^4). \tag{5}$$

Within the diffusion Transformer (DiT) (Peebles & Xie, 2023), for hidden state h and $C = (C_{\text{native}}, C_{\text{ut}})$, each cross-attention block adds a gated Utonia response to the native response:

$$\mathcal{A}(h, C) = \mathcal{A}_{\text{native}}(h, C_{\text{native}}) + \gamma\, \mathcal{A}_{\text{ut}}(h, C_{\text{ut}}). \tag{6}$$

The branches share query/output projections and use separate key/value projections; each zero-initialized gate γ preserves the initial native response.

Training updates the point adapter, the added attention branch and its gates, together with LoRA parameters for native key/value and shared query/output projections (Hu et al., 2022). The original backbone weights, native point encoder, Utonia encoder, and ShapeVAE remain frozen. Appendix D.1 gives dimensions and training settings.

5.3 Conditional Flow Matching

We train the adapted generator with flow matching between Gaussian noise and the encoded complete-building target (Lipman et al., 2023; Liu et al., 2023). With $t = 0$ at noise and $t = 1$

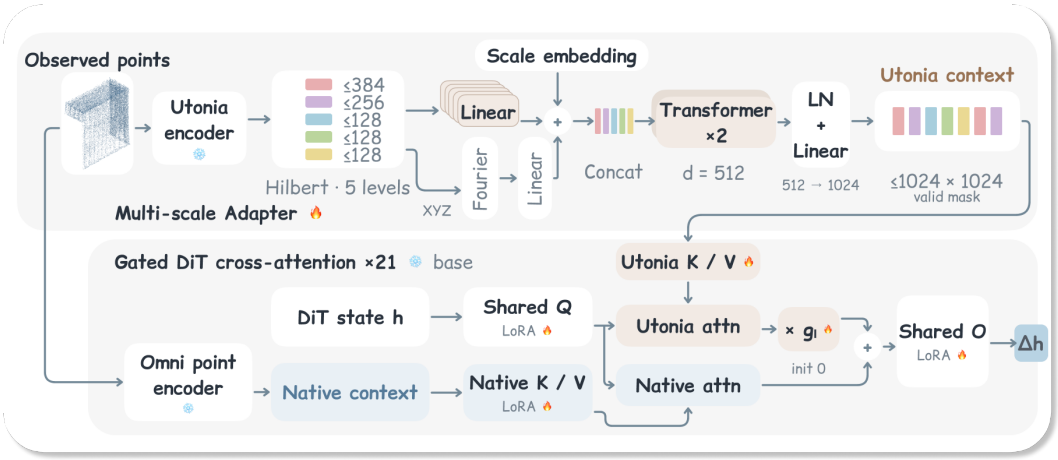

Figure 4: **Multiscale point adapter and gated DiT conditioning.** Multiscale Utonia features condition the pretrained DiT through gated cross-attention. Snowflakes/flames mark frozen/trainable parameters.

at the clean target,

$$x_t = (1-t)\epsilon + tz^\star, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{7}$$

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\big[\|v_\phi(x_t, t, C) - (z^\star - \epsilon)\|_2^2\big]. \tag{8}$$

Training samples a random time for velocity supervision. Inference integrates from noise under the observed-point condition and decodes $\widehat{z}$ directly into geometry.

5.4 Surface and Structural Decoding

The surface decoder extracts the zero-level set of a signed-distance field using marching cubes (Lorensen & Cline, 1987). Separate vertex and edge decoders train on encoded target latents and structural labels to predict bounded unsigned-distance fields (Figure 5). At inference they receive generated latents from the task-adapted Utonia-replacement generator on MunichWF (Section 6.3). We use an octree to accelerate vertex-candidate queries. Appendix E reports representation-only controls.

6 Experiments

We establish a reconstruction baseline on CityEXP and examine its recovery of missing surfaces and preservation of observed geometry. BuildingWorld3D conditioning comparisons and MunichWF structural decoding provide complementary studies of the shared training framework.

6.1 Reconstruction Baseline on CityEXP

Protocol. We adapt Ours from the BuildingWorld3D checkpoint at step 54k for 2,000 updates; validation $F_{.02}$ selects update 1,610. The quality-filtered cohort holds out test buildings from training and selection within shared scenes and acquisition pools. DA3 conditions use known Unreal camera poses and GT-box localization; clouds and prepared targets are normalized independently. Appendix B.1 details the protocol.

Reconstruction results. The baseline achieves $F_{.02}$ 54.38% with 100% valid outputs (Table 2); per-scene $F_{.02}$ ranges from 53.22% to 56.26%. These scores measure canonical-shape agreement within the sampled scenes.

Completion and observation fidelity. On target regions unsupported by the input, the baseline recovers 36.20% of the surface within tolerance 0.02. Mean distance on these regions decreases

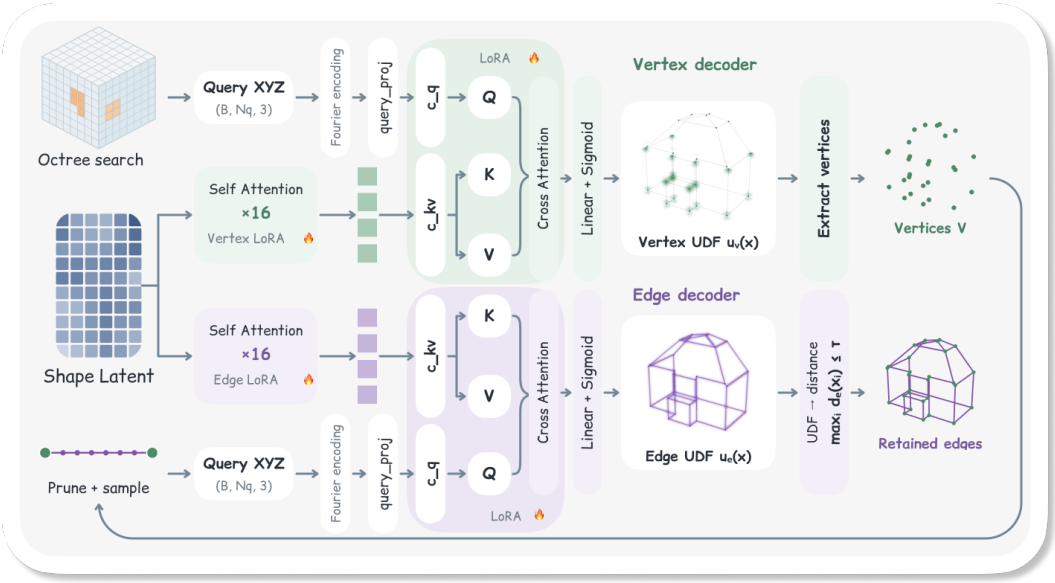

Figure 5: **Vertex and edge decoding.** Separate vertex and edge decoders predict bounded unsigned-distance fields from shape latents and query points. The fields support vertex and edge extraction for wireframes in the task-adapted structural study.

Table 2: **CityEXP reconstruction baseline.** The checkpoint is selected on validation buildings. DA3 point conditions use GT-box localization and independent cloud/target normalization. CD is multiplied by 10^3; F-scores and valid-output rates are percentages.

Method	CD↓	$F_{.01}$ ↑	$F_{.02}$ ↑	Valid↑
Ours	26.34	21.97	54.38	100.00

from 45.98×10^{-3} for the DA3 input to 42.35×10^{-3} for the prediction; on supported regions, it increases from 5.31×10^{-3} to 23.93×10^{-3}. These results show partial completion with a loss of observation fidelity. The regions are defined by point proximity; Appendix B.2 gives the protocol and uncertainty.

6.2 Conditioning ablations on BuildingWorld3D

Setup. Four configurations share the Omni EMA DiT and ShapeVAE and use 1,319,954 LoD2 training pairs, 128 city-balanced validation buildings, and 1,280 test buildings. Conditions are simulated point clouds from BuildingWorld3D. Exact raw mesh/cloud duplicates are grouped before splitting; cities and similar geometries can occur across partitions. Observations and targets are normalized independently, so these scores measure canonical-shape similarity.

Native baselines use LoRA ranks 16 and 64. Utonia replacement substitutes Utonia features for native conditioning; Ours retains native conditioning and adds the Utonia branch in Section 5. Ours trains 3.19% of its DiT and condition-module parameters. All runs complete the same number of updates; validation selects checkpoints for Table 3. Appendix D compares the common final update. Scores average all three generations per building using fixed checkpoints, 25 Euler steps, disabled classifier-free guidance, and marching cubes at resolution 128. Repeats vary the generation seed.

Conditioning comparison. At validation-selected checkpoints, Ours has the lowest mean CD and highest mean $F_{.02}$; the rank-64 native model has the highest $F_{.01}$. At the common final update, Ours has the best means among the four configurations: CD 10.40×10^{-3}, $F_{.01}$ 62.09%, and $F_{.02}$ 92.41%. For Ours minus the rank-64 native baseline, the validation-selected $F_{.02}$ difference is $+0.09$ percentage points (95% interval $[-0.17, 0.34]$), compared with $+0.29$ points ($[0.03, 0.55]$) at the final update. Ours is competitive with the stronger native baseline, with small gains dependent on

Table 3: Point-conditioning ablation on BuildingWorld3D with Omni DiT and ShapeVAE. Each run completes the same number of updates; its checkpoint is selected by validation $F_{.02}$. These are canonical-shape scores: observations and targets use independently cached normalizations. CD is multiplied by 10^3; F-scores are percentages. Train. gives the trainable fraction of DiT and condition parameters. Selected step denotes the checkpoint chosen by validation $F_{.02}$. Bold values denote best performance, and underlined values are the second-best.

Method	r	Selected step (k)	Train. (%)	CD ↓	$F_{.01}$ ↑	$F_{.02}$ ↑
Native	16	60.00	0.16	11.11	58.95	90.69
Native	64	59.00	0.63	<u>10.64</u>	**61.82**	<u>92.02</u>
Utonia replacement	16	57.00	3.12	10.75	59.85	91.46
Ours	16	54.00	3.19	**10.54**	<u>61.39</u>	**92.11**

checkpoint selection. The configurations share an update budget and differ in capacity and compute; Utonia conditioning adds both pretrained features and adapter parameters.

Dependence on observations. For the fixed Utonia-replacement model, $F_{.02}$ falls from 91.46% with matched conditions to 79.42% with permuted features, 68.14% with zero features, 6.05% with another building's condition, and 2.26% without conditioning (Table 9). The six empty no-condition outputs remain in the F-score denominator. These inference-time interventions use fixed Utonia-replacement weights and show dependence on observations.

Appendix F shows selected surfaces and reverse views from the Utonia-replacement model C49k on BuildingWorld3D.

6.3 Building wireframe decoding

MunichWF evaluates vertex and edge decoding from generated shape latents. A Utonia-replacement generator adapted on MunichWF supplies latents to separately trained vertex and edge decoders. On the matched 4,887-building test subset, corner/edge F1 is 94.40%/91.00%, compared with 93.06%/89.62% for BuildingGPT under the local protocol. Higher precision accompanies lower recall, and the distance metrics favor BuildingGPT. Appendix E reports the full comparison, target-latent diagnostics, and a qualitative junction-regularization example (Figure 8). Additional surface and wireframe examples appear in Appendix H.

7 Discussion and conclusion

CityEXP pairs detailed building geometry with incomplete observations through scalable procedural construction and high-fidelity urban rendering. UrBlend supplies a reference baseline that adapts Hunyuan's shape prior to incomplete building observations. Task-adapted geometric decoders recover wireframes from generated latents.

The baseline recovers some unsupported geometry, but preserving observed surfaces remains challenging. The surface experiments evaluate canonical-shape similarity within the declared cohorts; the CityEXP test holds out buildings within shared scenes after target-quality filtering. Structural decoding uses a separately task-adapted generator. Surface preparation and latent encoding can remove detail. The current benchmark therefore measures reconstruction of prepared targets; fidelity to the original detailed geometry remains a separate requirement.

AI USE STATEMENT

We used generative AI tools to assist with language polishing, manuscript organization, literature review, and interpretation of experimental results. The authors take full responsibility for the final manuscript.

ETHICS STATEMENT

Third-party assets retain their source licenses and attribution requirements. These licenses govern redistribution of source and derived assets.

REPRODUCIBILITY STATEMENT

Sections 3 and 5 specify the task and model; Section 4 defines the data-construction and benchmark protocol. Appendices B, D, and E detail the metrics, model configurations, checkpoint selection, and evaluation protocols. Appendix B.1 specifies the frozen CityEXP cohort, quality criterion, split, training configuration, and scoring procedure. Dataset versions distinguish the evaluated four-scene cohort from the enhanced-city inventory in Appendix A.3.

REFERENCES

Actueel Hoogtebestand Nederland. Over het programma AHN, n.d. URL https://www.ahn.nl/over-het-programma-ahn. Official programme documentation, accessed September 26, 2026.

CARLA Team. Digital twin tool, n.d. URL https://github.com/carla-simulator/carla/blob/master/Docs/adv_digital_twin.md. Official documentation, accessed September 26, 2026.

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, and Chi Zhang. MeshAnything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024a. URL https://arxiv.org/abs/2406.10163.

Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. MeshAnything V2: Artist-created mesh generation with adjacent mesh tokenization. *arXiv preprint arXiv:2408.02555*, 2024b. URL https://arxiv.org/abs/2408.02555.

Zhaiyu Chen, Yilei Shi, Liangliang Nan, Zhitong Xiong, and Xiao Xiang Zhu. PolyGNN: Polyhedron-based graph neural network for 3D building reconstruction from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 218:693–706, 2024c. doi: 10.1016/j.isprsjprs.2024.09.031. URL https://arxiv.org/abs/2307.08636.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16. PMLR, 2017. URL https://proceedings.mlr.press/v78/dosovitskiy17a.html.

Epic Games. City Sample, n.d.a. URL https://dev.epicgames.com/documentation/en-us/unreal-engine/city-sample-project-unreal-engine-demonstration. Official Unreal Engine documentation, accessed September 26, 2026.

Epic Games. Procedural Content Generation Overview, n.d.b. URL https://dev.epicgames.com/documentation/en-us/unreal-engine/procedural-content-generation-overview. Official Unreal Engine documentation, accessed September 26, 2026.

Weixiao Gao, Liangliang Nan, Bas Boom, and Hugo Ledoux. SUM: A benchmark dataset of semantic urban meshes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 179:108–120, 2021. doi: 10.1016/j.isprsjprs.2021.07.008. URL https://3d.bk.tudelft.nl/projects/meshannotation/.

Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2Mesh: A self-prior for deformable meshes. *ACM Transactions on Graphics*, 39(4):126:1–126:12, 2020. doi: 10.1145/3386569.3392415. URL https://arxiv.org/abs/2005.11084.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Tongyan Hua, Haoran Gong, Yuan Liu, Di Wang, Ying-Cong Chen, and Wufan Zhao. BuildAnyPoint: 3D building structured abstraction from diverse point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17099–17109, 2026. URL https://arxiv.org/abs/2602.23645.

Jin Huang, Jantien Stoter, Ravi Peters, and Liangliang Nan. City3D: Large-scale building reconstruction from airborne LiDAR point clouds. *Remote Sensing*, 14(9):2254, 2022. doi: 10.3390/rs14092254. URL https://3d.bk.tudelft.nl/liangliang/publications/2022/city3d/city3d.pdf.

Jingwei Huang, Yichao Zhou, and Leonidas Guibas. ManifoldPlus: A robust and scalable watertight manifold surface generation method for triangle soups. *arXiv preprint arXiv:2005.11621*, 2020. URL https://arxiv.org/abs/2005.11621.

Shangfeng Huang, Ruisheng Wang, Bo Guo, and Hongxin Yang. PBWR: Parametric-building-wireframe reconstruction from aerial LiDAR point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27778–27787, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_PBWR_Parametric-Building-Wireframe_Reconstruction_from_Aerial_LiDAR_Point_Clouds_CVPR_2024_paper.html.

Shangfeng Huang, Ruisheng Wang, and Xin Wang. BuildingWorld: A structured 3D building dataset for urban foundation models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(7):5085–5094, 2026. doi: 10.1609/aaai.v40i7.37422. URL https://arxiv.org/abs/2511.06337.

Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology. KIT IFC examples, 2021. URL https://www.ifcwiki.org/index.php?title=KIT_IFC_Examples. Official example models, accessed September 26, 2026.

An Li, Zhe Zhu, and Mingqiang Wei. GenPC: Zero-shot point cloud completion via 3D generative priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1308–1318, 2025a. URL https://arxiv.org/abs/2502.19896.

Li Li, Nan Song, Fei Sun, Xinyi Liu, Ruisheng Wang, Jian Yao, and Shaosheng Cao. Point2Roof: End-to-end 3D building roof modeling from airborne LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 193:17–28, 2022. doi: 10.1016/j.isprsjprs.2022.08.027. URL https://doi.org/10.1016/j.isprsjprs.2022.08.027.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, and Yan-Pei Cao. TripoSG: High-fidelity 3D shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025b. URL https://arxiv.org/abs/2502.06608.

Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. MatrixCity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3205–3215, 2023. URL https://arxiv.org/abs/2309.16553.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. URL https://arxiv.org/abs/2511.10647.

Liqiang Lin, Yilin Liu, Yue Hu, Xingguang Yan, Ke Xie, and Hui Huang. Capturing, reconstructing, and simulating: the UrbanScene3D dataset. In *European Conference on Computer Vision*, pp. 93–109, 2022. doi: 10.1007/978-3-031-20074-8_6. URL https://vcc.tech/UrbanScene3D.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.03003.

Yujia Liu, Stefano D'Aronco, Konrad Schindler, and Jan Dirk Wegner. PC2WF: 3D wireframe reconstruction from raw point clouds. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=8X2eaSZxTP.

Yujia Liu, Anton Obukhov, Jan Dirk Wegner, and Konrad Schindler. Point2Building: Reconstructing buildings from airborne LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 215:351–368, 2024. doi: 10.1016/j.isprsjprs.2024.07.012. URL https://arxiv.org/abs/2403.02136.

Yujun Liu, Ruisheng Wang, Shangfeng Huang, and Guorong Cai. EdgeDiff: Edge-aware diffusion network for building reconstruction from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17008–17018, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liu_EdgeDiff_Edge-aware_Diffusion_Network_for_Building_Reconstruction_from_Point_Clouds_CVPR_2025_paper.html.

Yuzhou Liu, Lingjie Zhu, Hanqiao Ye, Shangfeng Huang, Xiang Gao, Xianwei Zheng, and Shuhan Shen. BWFormer: Building wireframe reconstruction from airborne LiDAR point cloud with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22215–22224, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liu_BWFormer_Building_Wireframe_Reconstruction_from_Airborne_LiDAR_Point_Cloud_with_CVPR_2025_paper.html.

Yuzhou Liu, Lingjie Zhu, Hanqiao Ye, Yujun Liu, Shangfeng Huang, Xiang Gao, Ruisheng Wang, and Shuhan Shen. BuildingGPT: Auto-regressive building wireframe reconstruction model with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36400–36410, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liu_BuildingGPT_Auto-Regressive_Building_Wireframe_Reconstruction_Model_with_Reinforcement_Learning_CVPR_2026_paper.html.

William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. doi: 10.1145/37402.37422. URL https://doi.org/10.1145/37402.37422.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Mescheder_Occupancy_Networks_Learning_3D_Reconstruction_in_Function_Space_CVPR_2019_paper.html.

Liangliang Nan and Peter Wonka. PolyFit: Polygonal surface reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2353–2361, 2017.

URL https://openaccess.thecvf.com/content_iccv_2017/html/Nan_PolyFit_Polygonal_Surface_ICCV_2017_paper.html.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Park_DeepSDF_Learning_Continuous_Signed_Distance_Functions_for_Shape_Representation_CVPR_2019_paper.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Ravi Peters, Balázs Dukai, Stelios Vitalis, Jordi van Liempt, and Jantien Stoter. Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the netherlands. *Photogrammetric Engineering & Remote Sensing*, 88(3):165–170, 2022. doi: 10.14358/PERS.21-00032R2. URL https://arxiv.org/abs/2201.01191.

Jialu Sui, Rui Liu, and Hongsheng Zhang. OCCDiff: Occupancy diffusion model for high-fidelity 3D building reconstruction from noisy point clouds. *arXiv preprint arXiv:2512.08506*, 2025. URL https://arxiv.org/abs/2512.08506.

Tencent Hunyuan3D Team. Hunyuan3D 2.1: From images to high-fidelity 3D assets with production-ready PBR material. *arXiv preprint arXiv:2506.15442*, 2025a. URL https://arxiv.org/abs/2506.15442.

Tencent Hunyuan3D Team. Hunyuan3D-Omni: A unified framework for controllable generation of 3D assets. *arXiv preprint arXiv:2509.21245*, 2025b. URL https://arxiv.org/abs/2509.21245.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025. URL https://arxiv.org/abs/2503.11651.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3D latents for scalable and versatile 3D generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21469–21480, 2025. URL https://arxiv.org/abs/2412.01506.

Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PoinTr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12498–12507, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Yu_PoinTr_Diverse_Point_Cloud_Completion_With_Geometry-Aware_Transformers_ICCV_2021_paper.html.

Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In *International Conference on 3D Vision*, 2018. URL https://arxiv.org/abs/1808.00671.

Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics*, 42(4):1–16, 2023. doi: 10.1145/3592442. URL https://arxiv.org/abs/2301.11445.

Yujia Zhang, Xiaoyang Wu, Yunhan Yang, Xianzhe Fan, Han Li, Yuechen Zhang, Zehao Huang, Naiyan Wang, and Hengshuang Zhao. Utonia: Toward one encoder for all point clouds. *arXiv preprint arXiv:2603.03283*, 2026. URL https://arxiv.org/abs/2603.03283.

A CITYEXP CONSTRUCTION, ORGANIZATION, AND QUALITY ASSESSMENT

CityEXP links procedural city construction to building reconstruction through geometry recorded during generation. Scene rendering and target export consume the same retained instances, providing paired observations and geometric supervision with explicit identities and coordinate frames.

A.1 AGENT SKILL INDEX AND EXECUTION PROTOCOL

We document agent-assisted construction as a reusable workflow skill with six indexed operations (Table 4). Each operation specifies inputs, tool actions, saved artifacts, and acceptance checks. The agent coordinates procedural generators and scene-editing tools; native Unreal PCG generates the building geometry. Graph bindings, asset versions, and run configurations are recorded for each construction run.

Table 4: Skill index for agent-assisted CityEXP construction. Each stage passes its saved outputs and verification records to the next.

Operation	Inputs	Tool actions	Artifacts and checks
S1. Asset and graph inspection	Installed project, engine, and asset library	Inspect graph topology, exposed parameters, modules, materials, and mesh bounds.	Source versions and available graph/asset inventory; verify required assets and graph bindings.
S2. Layout and input freezing	Footprints, roads, context regions, building IDs, and seeds	Freeze height and context plans; use native Perlin noise for enhanced-city height variation.	Hashed layout and building plans; check identities and layout constraints.
S3. Native generation and capture	Frozen plans, procedural graphs, and style settings	Assign an execution ID, run PCG, and capture actual generated components.	Per-building recipes and execution events; verify plot ownership, transforms, and roof/cap geometry.
S4. Retention and assembly	Accepted building batches and context outputs	Retain actors, preserve world transforms, and assemble the existing batches.	Entry map and assembly manifest; compare inventories before and after retention.
S5. Save and reopen verification	Assembled maps, recipes, and expected inventories	Save, switch to an empty map, reload, and compare the restored scene.	Saved maps and verification records; check identities, instances, transforms, materials, and dynamic geometry.
S6. Rendering and handoff	Reopened scene and camera plan	Read actual camera parameters, warm up the scene, render, and inspect images.	Images, camera records, and source manifest; check framing, appearance, and correspondence to the frozen scene.

Execution order and identity. S1–S2 fix the source assets, layout, building identities, generation settings, and input hashes. Before each native `ExecuteGraphInstance` call in S3, the workflow allocates a unique execution identifier. A style-selection retry receives a new identifier and retains the previous attempt record. Completion is determined from native generation events and state readback. The capture record binds the requested building to the selected plot and actual component inventory, including static-mesh instances, component and instance transforms, material assignments, and dynamic roof and bottom-cap geometry. This records the building produced by that execution.

Retention and scene assembly. In S4, clearing the PCG link retains generated actors and their components. Inventory comparisons before and after this operation check that geometry and transforms are preserved. Setting a translated building root restores the component world transforms.

Assembly then combines accepted building batches with separately generated parking, plaza, vegetation, and river context. Reused roads and environment assets and explicitly placed static furnishings are recorded separately from native PCG outputs. Accepted buildings are assembled from their retained generation outputs.

Persistence and acquisition. S5 checks persistence through a save–empty-map–reopen sequence, comparing restored building identities, component inventories, transforms, materials, and dynamic geometry against the retained records. In S6, camera extrinsics and intrinsics are read from the actual loaded camera configuration. Scene warmup precedes rendering, and image inspection checks framing, grounding, material appearance, and contextual occlusion. The workflow retains the inspected outputs with their camera and source records.

Common source for downstream products. Rendering and geometry export consume the same frozen generation: the assembled scene supplies images, while the per-building records supply detailed meshes and structural-LoD proxies. Image-derived point observations are paired with the exported geometry through their shared identities and coordinate records. Observation variants share the same target building geometry. Scene persistence checks verify the retained scene; separate geometry and pairing checks determine which downstream products enter the reconstruction benchmark.

A.2 Building records and pairing

Table 5 specifies the building-level schema. Accounting separates the scene inventory, exported geometry, and quality-assessed reconstruction pairs; model inputs remain distinct from supervision and evaluation metadata.

Table 5: Building-level metadata and information boundaries.

Group	Required content	Role
Identity	Dataset revision, scene, building, asset family	Index / split
Observation	Points, units, source, available attributes, preprocessing	Input
Localization	Crop, axis conversion, center, scale, orientation, provenance	Track-specific input
Surface target	Complete geometry, target detail convention, repairs, transforms	Supervision / evaluation
Structural target	Junctions, edges, and matching conventions, when available	Auxiliary analysis
Visibility	Acquisition or scene visibility, supported-region definition	Evaluation only
Latent target	Shape encoder version, samples, posterior and selection rule	Training / diagnostic
Provenance	Licenses, split hashes, checksums, exclusions, seeds	Reproducibility

Instance association and coordinates. Scene, generation, and building identities associate observations with their geometric targets. The scene record retains the source components, per-instance transforms, and generated roof and boundary geometry used to compose each reference mesh. The scene-to-export transform is applied once. Coordinate metadata describes units, axis conventions, instance pose, normalization, and bounding-box provenance, together specifying the complete coordinate mapping. Point and mesh frames are checked through coordinate round trips. Target-derived localization is used only in explicitly declared known-box settings. After axis and unit conversion, the local frame is

$$x^o = Q^{\mathsf{T}}(x^w - c)/s, \qquad x^w = c + sQx^o, \quad s > 0, \quad Q \in \mathrm{SO}(3), \tag{9}$$

where c, s, and Q are the center, scale, and rotation. A common transform preserves the spatial relationship of observations and targets. Independent cloud/target normalization instead defines a canonical-shape comparison, as used in the surface experiments.

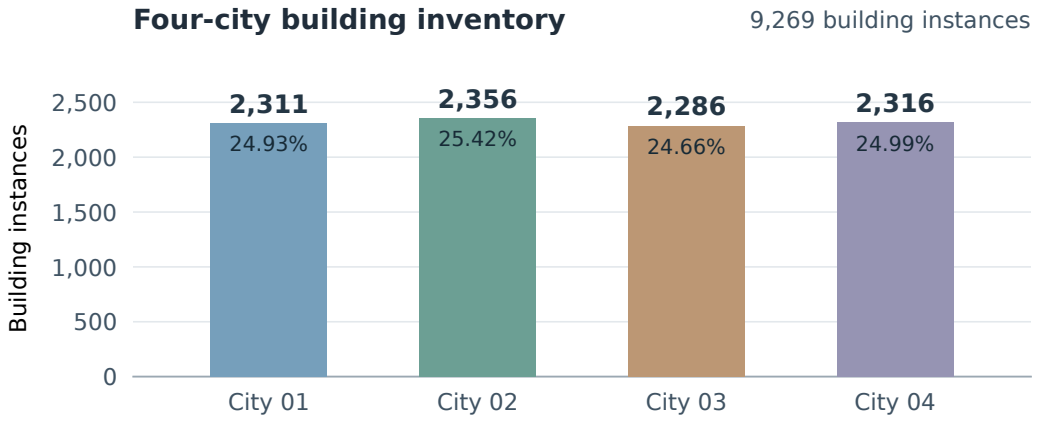

Figure 6: **Building distribution across the four CityEXP cities.** The four generated scenes contain 9,269 retained building instances. Numbers above bars give per-city counts; percentages give each city's share of the total. Each city contributes approximately one quarter of the inventory. Counts refer to generated building instances.

Geometric quality and inclusion criteria. Quality assessment covers instance association, coordinate consistency, exterior geometry, source-family provenance, and license records. Geometric assessment distinguishes successful mesh conversion from fidelity to the source and rendered instance. It checks exterior coverage and disconnected or missing parts. A quality-assessed pair must satisfy all of these criteria. The evaluated CityEXP cohort and its encode–decode quality criterion are described in Appendix B.1. Failure records retain invalid inputs and unsuccessful target encodings within the declared evaluation population. Each building is counted once across observation variants and latent replicates.

A.3 Dataset scope

The enhanced collection comprises four completed city scenes with 9,269 retained building instances. The per-city counts are 2,311, 2,356, 2,286, and 2,316 (Figure 6). Six ungenerated plots are excluded. The maps and source assets have been archived and checked by reopening and readback. These checks establish a recoverable scene inventory. The reported inventory counts scenes and retained building instances; quality-assessed reconstruction pairs form a separate accounting unit. Figure 7 shows the four enhanced scenes.

Geometry exports and source versions. The baseline City 01 assembly contains 2,311 buildings. Batch composition exported 2,311 detailed and 2,311 structural-LoD building files. These counts describe the geometry export inventory. This baseline is distinct from enhanced City 01 despite the matching building count. Enhanced Cities 02–04 have completed export and archival of 6,958 detailed meshes and 6,958 structural proxies, with full archive readback verification.

Structural proxies use supported roof boundaries and bottom caps to form closed envelopes. Detailed component meshes retain a different level of geometry, including openings, thin structures, and potentially open boundaries. The two exports therefore remain separate target types.

Observation inventory and rendering. The baseline City 01 collection contains 31 batches and 7,530 rendered images, with RGB, decoded building-ID masks, and cameras. Enhanced Cities 02–04 have completed 603 render batches with 144,512 frames: 48,061, 48,236, and 48,215 frames, respectively. These archives contain RGB, three raw ActorID passes, cameras, and metadata. The archived products precede ActorID decoding and instance-mask postprocessing. The reported frame count covers enhanced Cities 02–04. Rendering uses $1{,}280\times960$ images, an 84° horizontal field of view, and a five-camera rig, with scene lighting, fog, clouds, and postprocessing retained.

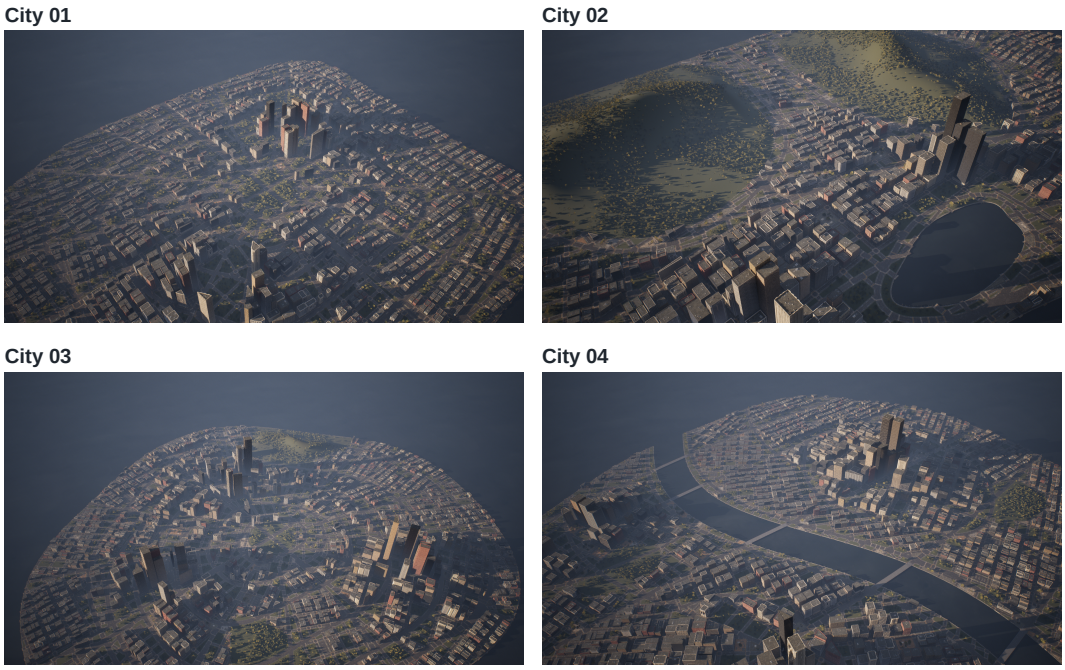

Figure 7: **City 01–04 scene previews.** Archived Unreal Engine views of the four enhanced CityEXP scenes, showing their building layouts, height variation, vegetation, and water bodies. Original views are shown.

The separate AHN-style (Actueel Hoogtebestand Nederland, n.d.) vertical-grid route samples collision or fallback triangles; its sampled geometry follows the collision or fallback representation. Its component test on the official Unreal PCG demo is separate from the enhanced-city acquisitions and the DA3 reconstruction baseline. The evaluated DA3 observations and their source collections are described in Appendix B.1.

Evaluation scope. The reconstruction baseline uses quality-filtered pairs assembled from baseline City 01 and enhanced Cities 02–04. This cohort is distinct from the 9,269-instance enhanced-city inventory: the first city comes from a different assembly revision, and the cohort includes only targets with accepted encode–decode scores at the fixed cutoff. The partition protocol and measured baseline are reported in Appendix B.1.

B Evaluation metrics and comparison protocols

Let P and Q here denote area-uniform samples from predicted and target surfaces. In the declared evaluation frame,

$$\mathrm{CD}(P,Q) = \frac{1}{2|P|}\sum_{p\in P}\min_{q\in Q}\|p-q\|_2 + \frac{1}{2|Q|}\sum_{q\in Q}\min_{p\in P}\|q-p\|_2. \tag{10}$$

Sample counts, random seeds, target-surface conventions, and units are specified separately for each evaluation. For tolerance τ,

$$\mathrm{Prec}_\tau = |P|^{-1}\sum_{p\in P}\mathbf{1}[\min_{q\in Q}\|p-q\|_2 < \tau], \tag{11}$$

$$\mathrm{Rec}_\tau = |Q|^{-1}\sum_{q\in Q}\mathbf{1}[\min_{p\in P}\|q-p\|_2 < \tau], \tag{12}$$

$$F_\tau = \frac{2\,\mathrm{Prec}_\tau\,\mathrm{Rec}_\tau}{\mathrm{Prec}_\tau + \mathrm{Rec}_\tau}, \tag{13}$$

with $F_\tau = 0$ when the denominator is zero. Tolerances refer to the coordinate system of the corresponding evaluation. Reported surface scores use normalized coordinates; world-coordinate evaluation requires a separate coordinate protocol.

Observed and unobserved geometry. Regional protocols extend whole-surface evaluation. Appendix B.2 reports input-support-based scores on CityEXP. Visibility-based evaluation uses sensor or scene provenance to partition Q into Q_{obs} and Q_{unobs}. Regional recall and GT-to-prediction distance measure completeness in each partition; whole-mesh precision penalizes extraneous geometry. A proximity-based partition instead measures support by input points and uses an explicit distance threshold. Visibility and point support therefore define separate regional metrics. Scores for empty partitions are undefined. Cross-condition comparisons use a fixed reference partition and hence an identical scoring region.

Structural matching. Junction and edge precision/recall depend on the matching rule, coordinate frame, tolerances, duplicate-edge treatment, and handling of unmatched endpoints. Comparisons with graph methods retain the benchmark's metric definitions. Scores computed under different definitions are reported with their respective protocols. Appendix E specifies the MunichWF protocol; face reconstruction is a distinct task.

Failures, alignment, and stochastic outputs. Missing or invalid outputs receive zero F-score and count against the valid-output rate. Chamfer distance is conditional on a valid surface and is reported alongside that rate; an overall ranking requires a predeclared failure penalty. GT alignment is excluded from the main reconstruction score. Optional aligned shape diagnostics disclose which errors they remove. Seeds are fixed before testing, and averages and dispersion are reported; best-of-many selection using test GT is a separately labeled oracle diagnostic.

B.1 CityEXP reconstruction baseline

Cohort and source versions. The evaluated cohort contains buildings from the earlier baseline City 01 and enhanced Cities 02–04. City 01 therefore differs from the enhanced City 01 shown in Figure 7. The cohort uses detailed meshes, prepared as watertight encoder targets, and paired DA3 observations. Admission requires a cached target encode–decode F-score of at least 0.9 against the prepared target. This criterion assesses representation fidelity against the prepared surface.

Membership was frozen on September 26, 2026 before training and splitting. It contains the qualifying targets from the preceding training pools; preceding validation targets, targets awaiting scores at the cutoff, and later scores are excluded. The resulting cohort is a fixed quality-filtered sample of the scene inventory.

Partition and information boundaries. A deterministic hash of scene and building identity, with seed 20260910, allocates buildings to training, validation, and test partitions using proportional scene quotas. Building identities, generation identities, and exact source-mesh hashes are disjoint. The partitions share reusable asset families, city scenes, and acquisition pools; overlapping observations and neighboring-building points are retained. This evaluates held-out targets within shared scenes.

DA3 uses rendered images with known Unreal camera poses. A reference-mesh bounding box defines the observation crop. All cropped points, including noise and neighboring geometry, pass to the point-conditioning pipeline. The cloud uses its own bounding-box center and longest extent for normalization; the target uses the prepared-mesh frame. Training varies the crop yaw within $\pm 5^\circ$ and the point-encoding seed. Evaluation uses zero yaw and fixed per-building seeds, with the observed cloud as the generator condition.

Training and model selection. Training initializes Ours from the BuildingWorld3D checkpoint at step 54k, with fresh optimizer and scheduler states. The backbone, native point encoder, Utonia encoder, and ShapeVAE remain frozen; the condition modules and rank-16 LoRA parameters are adapted. We run 2,000 updates on eight GPUs with effective batch size 64. AdamW uses learning rate 10^{-4}, weight decay 0.01, betas $(0.9, 0.95)$, 100 warm-up updates, and a minimum learning-rate ratio of 0.1.

Table 6: CityEXP per-scene results for the same selected model. CD is multiplied by 10^3; F-scores are percentages. City 01 uses the earlier baseline assembly.

Scene	CD↓	$F_{.01}$ ↑	$F_{.02}$ ↑
City 01 (baseline)	26.13	21.02	53.22
City 02 (enhanced)	25.23	22.96	56.26
City 03 (enhanced)	26.91	21.33	53.59
City 04 (enhanced)	27.02	22.89	54.82
All buildings	26.34	21.97	54.38

Validation computes the flow objective on the validation partition and generated surface metrics on a fixed subset. The best generated-surface $F_{.02}$ selects update 1,610. On this same subset, $F_{.02}$ rises from 32.05% before adaptation to 53.34%, while CD decreases from 55.40×10^{-3} to 26.92×10^{-3}. These validation measurements serve model selection and are separate from the test results below. The completed 2,000-update run supplies the reported model.

Generation and scoring. Each test building receives one generation with seed 7, 25 Euler steps, and marching cubes at resolution 128 over $[-1.01, 1.01]^3$. Evaluation compares 100,000 area-uniform predicted-surface samples with the 40,960 uniform target points in the encoder cache, excluding sharp-edge samples. CD is the mean of the two directed unsquared Euclidean distances; F-scores use strict thresholds 0.01 and 0.02 in the normalized frame. Scores average all test buildings, each represented by one valid output.

The separate complete-target encode–decode diagnostic yields $F_{.02}$ 97.12% on these test targets. It measures representation fidelity and is excluded from the point-conditioned baseline table.

B.2 Recovery of DA3-unsupported surfaces

Protocol. We evaluate the same CityEXP predictions and validation-selected checkpoint as Appendix B.1. This analysis uses the full cropped DA3 cloud X and the repaired target surface S. The saved scene-to-target transform places X in the target frame; the predicted mesh $\widehat{S}$ retains its decoder coordinates. Generation uses the independently normalized condition described in the main protocol. We sample 100,000 points uniformly by surface area from each target and prediction, using seeds 0 and 1, respectively. Target samples in the lowest 5% of target height with outward normal component $n_y < -0.7$ form a bottom proxy and are excluded. For the remaining samples Q_n, define

$$M = \{q \in Q_n : d(q, X) \geq 0.02\}, \qquad R_\epsilon = \frac{1}{|M|} \sum_{q \in M} \mathbf{1}[d(q, \widehat{S}) < \epsilon]. \tag{14}$$

Here d is Euclidean distance to the nearest input point or mesh triangle, as appropriate. M is a fixed input-support partition defined independently of the prediction. We also measure the reduction in mean distance on M, from $d(q, X)$ to $d(q, \widehat{S})$. To assess erroneous additions, novel-surface precision is the fraction of predicted samples with $d(p, X) \geq 0.02$ that satisfy $d(p, S) < \epsilon$. These samples exclude the same bottom proxy, using target height and predicted normals. All thresholds use normalized target units.

Results and uncertainty. All predictions are valid. Two targets have empty M and are excluded only from missing-region metrics. The large-gap subset contains the 102 buildings with $|M|/|Q_n| \geq 20\%$, selected from input support before examining predictions. Table 7 averages buildings equally. At $\epsilon = 0.02$, recovery is 36.20% (95% bootstrap interval 34.59–37.80%) overall and 29.51% (25.87–33.29%) on the large-gap subset. Paired mean-distance reductions are 3.64×10^{-3} ($[1.31, 5.92] \times 10^{-3}$) and 15.64×10^{-3} ($[7.01, 24.52] \times 10^{-3}$), respectively. Intervals use 2,000 building resamples with seed 20260926 and describe this shared-scene test cohort. Distances improve for 557 of 1,022 nonempty-gap targets and 61 of 102 large-gap targets.

Interpretation. The results support partial surface completion, with limited precision of the added geometry and lower fidelity in already supported regions. Input recovery at $\epsilon \leq 0.02$ is zero by

Table 7: **DA3-unsupported surface recovery.** The large-gap subset is selected by input support alone. Recovery and precision use tolerance 0.02 and are percentages. Distances are multiplied by 10^3; paired cells show DA3 / prediction. Relative reductions are calculated from the group mean distances.

Metric	Full test set	Large-gap subset
Missing-surface recovery ↑	36.20	29.51
Missing-region distance ↓	45.98 / 42.35	71.88 / 56.24
Relative distance reduction (%) ↑	7.91	21.76
Novel-surface precision ↑	52.69	48.19
Supported-region distance ↓	5.31 / 23.93	6.16 / 29.00

construction. With M fixed and $\epsilon = 0.04$, prediction/input recovery is 63.82/68.45% overall and 52.70/41.75% on the large-gap subset. Gains therefore depend on the region and tolerance. Table 7 also records the associated increase in supported-region distance.

Complete-target reconstruction control. We additionally score the saved complete-target ShapeVAE reconstructions on the same buildings, target samples, and fixed regions. At tolerance 0.02, their supported-region coverage is 97.24%, compared with 57.75% for conditional generation; missing-region coverage is 93.11%, compared with 36.20%. Mean supported-region distance is 3.56×10^{-3} versus 23.93×10^{-3}. On this cohort, which was filtered by target encode–decode quality, the representation can therefore retain much of the geometry lost during conditional generation. The remaining gap may reflect observation ambiguity, conditioning, normalization, and latent generation. Complete-target reconstruction provides a representation diagnostic using encoded target geometry. These findings motivate treating UrBlend as a reference baseline for reconstruction from incomplete observations, with observation fidelity an important direction for improvement.

C Auxiliary architectural structure readout

Let $\Phi_{\theta,\eta}(z)$ denote spatial features of the geometric decoder, with pretrained base parameters θ and optional task-specific updates η. A trainable head H_ω predicts architectural junctions and edges:

$$\widehat{\mathcal{G}} = H_\omega(\Phi_{\theta,\eta}(\widehat{z})). \tag{15}$$

This auxiliary interface tests architectural information in the shape representation. GT-latent readout controls use complete-target-encoded latents $z^\star$ and measure information retained by the representation. The point-to-graph evaluation in Section 6.3 instead uses latents predicted from the permitted point cloud and includes upstream failures and coordinate errors.

GT encode–decode and GT-latent-to-graph scores quantify complete-target representation fidelity. Point-conditioned reconstruction is evaluated separately using generated latents. Mesh evaluation of graph outputs requires an explicit face-construction stage. Comparisons with BuildingGPT use the same point input, graph target, and matching protocol.

D DiT configuration study and condition dependence

D.1 Main additive configuration

Geometry targets. The target encoder receives samples from an exterior building shell. Admissibility checks cover watertightness, winding, degeneracy, self-intersections, and coordinates. Already admissible shells bypass repair; conversion failures and geometric changes are recorded. Wireframes require face recovery, and CAD inputs refer to converted boundary geometry. The input contains 40,960 area-uniform samples and 40,960 sharp-edge samples, with normals and a sharp-sample flag. A uniform-only fallback is recorded separately when eligible edges are absent. The frozen Hunyuan3D-Omni ShapeVAE encoder produces one fixed posterior sample per building, $z^\star \in \mathbb{R}^{4096\times 64}$. The checkpoint's latent scale is applied once for generation and removed once before decoding. The common encoder and latent convention support source geometries with dif-

ferent vertex ordering and connectivity. Conditional training uses paired observations and reference geometry.

Conditioning implementation. The main model uses the Hunyuan3D-Omni EMA DiT. Its native point encoder receives 2,048 observed XYZ points, together with the official zero image embedding and type tokens; image conditioning is disabled. A frozen Utonia encoder contributes features at five hierarchy levels, with token caps $384, 256, 128, 128, 128$ and valid-length masks. The point-memory adapter contains two Transformer layers of width 512 with eight attention heads, followed by a projection to width 1,024. It supplies at most 1,024 additional condition tokens from the same observed cloud. Each of the 21 DiT cross-attention blocks has an independent Utonia key/value branch and key normalization, initialized from the native attention, with a zero-initialized scalar gate. Native and Utonia branches share query and output projections (Figure 4).

Adapted parameters. Rank-16, $\alpha = 32$ LoRA updates native key/value and shared query/output projections. The trainable set also includes point memory, Utonia-branch key/value projections, key normalization, and gates. The original DiT parameters, native point encoder, Utonia encoder, and ShapeVAE remain frozen. Adaptation jointly trains the point branch and LoRA parameters.

Time sampling. The clean target is denoted by $z^\star$, with $t = 0$ at noise and $t = 1$ at the clean endpoint. Training samples $u = \mathrm{sigmoid}(\mathcal{N}(0, 1))$ and sets $t = au/[1 + (a - 1)u]$, where $a = \sqrt{0.125}$. The velocity target remains $z^\star - \epsilon$, with generation running from noise to the clean endpoint.

D.2 Configurations and evaluation

Completed training and checkpoint selection. The four configurations in Table 3 share Omni EMA DiT initialization, Omni ShapeVAE, 1,319,954 admitted BuildingWorld3D training pairs, 128 tuning buildings, and the same 1,280 frozen test buildings. Each run completes the same scheduled update budget. The primary comparison selects one checkpoint per completed run by tuning-set $F_{.02}$: 60k/59k for native rank 16/64, 57k for Utonia replacement, and 54k for Ours. Each checkpoint generates three samples per test building, with seeds 7, 17, and 27; scores average all samples within each building before averaging buildings. All generation seeds share each model's fixed trained checkpoint.

Configuration identity. The two native baselines use Omni's native encoder with 2,048 observed XYZ points, the official zero image embedding, and type tokens; image conditioning is disabled. They differ in Q/K/V/O LoRA rank, 16 versus 64. Utonia replacement uses frozen multiscale features, a trainable point-memory encoder, independent native-initialized point K/V projections, and gated point attention in place of native conditioning. It adapts Q/O with rank-16 LoRA. Ours retains native conditioning and adds the zero-initialized gated Utonia attention residual, with rank-16 Q/K/V/O LoRA. Both Utonia configurations admit at most 1,024 Utonia tokens; Ours additionally retains the native pathway. Their trainable capacities differ because of the additional conditioning modules. Tables report the trainable fraction, $100 \times N_{\mathrm{trainable}}/N_{\mathrm{total}}$, as a percentage. Both counts cover the DiT and condition modules, excluding the frozen VAE and offline Utonia encoder. For Ours, this fraction is 3.19%; the pretrained weights remain frozen.

Fixed-final-update comparison. Table 8 evaluates all four runs after the same number of updates, independently of validation-best selection. Ours achieves the best means in all three metrics: CD 10.40×10^{-3}, $F_{.01}$ 62.09%, and $F_{.02}$ 92.41%. Each row uses one checkpoint under its stated selection rule. All 30,720 generated outputs across the two four-model groups pass the evaluator's geometry checks, with a valid-output rate of 100%.

Paired confidence intervals. We use 10,000 paired bootstrap resamples of buildings, after averaging the three generation seeds within each building, with bootstrap seed 20260919. For validation-selected checkpoints, Ours minus the rank-64 native baseline has a CD difference of -0.10×10^{-3} (95% interval $[-0.22, 0.01] \times 10^{-3}$) and an $F_{.02}$ difference of $+0.09$ percentage points (interval $[-0.17, 0.34]$). Both intervals include zero. The native baseline's $F_{.01}$ exceeds ours by 0.43 points (interval $[0.09, 0.78]$). At the common final update, the same difference is -0.19×10^{-3} in CD (interval $[-0.31, -0.07] \times 10^{-3}$), $+0.29$ points in $F_{.02}$ (interval $[0.03, 0.55]$), and $+0.26$ points in

Table 8: Fixed-final-update comparison. All checkpoints are taken after the same number of updates, independently of validation-best selection. All variants use Omni DiT and ShapeVAE, with the same canonical-shape protocol as Table 3. CD is multiplied by 10^3; F-scores are percentages. Train. gives the trainable fraction of DiT and condition parameters. Bold values denote best performance, and underlined values are the second-best.

Method	r	Step (k)	Train. (%)	CD ↓	$F_{.01}$ ↑	$F_{.02}$ ↑
Native	16	61.88	0.16	10.98	59.56	91.07
Native	64	61.88	0.63	<u>10.58</u>	<u>61.83</u>	<u>92.13</u>
Utonia replacement	16	61.88	3.12	10.90	59.48	91.22
Ours	16	61.88	3.19	**10.40**	**62.09**	**92.41**

Table 9: Inference-time condition interventions with the fixed Utonia replacement checkpoint C57k. CD is multiplied by 10^3; F-scores are percentages. Empty outputs receive zero F-score; CD uses valid outputs. These interventions use fixed model weights. Bold values denote best performance, and underlined values are the second-best.

Condition	Step (k)	CD ↓	$F_{.01}$ ↑	$F_{.02}$ ↑	Empty
Matched	57.00	**10.75**	**59.85**	**91.46**	0
Mismatched building	57.00	217.95	2.26	6.05	0
No condition	57.00	343.96	0.99	2.26	6
Zero features	57.00	25.86	35.98	68.14	0
Permuted features	57.00	<u>16.79</u>	<u>45.03</u>	<u>79.42</u>	0

$F_{.01}$ (interval $[-0.05, 0.58]$). At the common final update, the intervals favor Ours over the native baseline for CD and $F_{.02}$. Under validation-based selection, the corresponding intervals include zero, while the $F_{.01}$ interval favors the native baseline. These unadjusted intervals describe variation across test buildings for fixed trained models.

Inference-time condition interventions. Table 9 fixes the replacement checkpoint C57k, generation noise, decoder, and geometry protocol. Matched results reuse exactly the corresponding primary evaluation. A seeded cross-building permutation supplies mismatched observations, with no self-pairings. Zero-feature and permuted-feature controls preserve observed coordinates and valid lengths; the latter applies a seeded nonzero cyclic shift to valid features within each hierarchy level while keeping coordinates fixed. All 6,400 building-level feature permutations change the feature assignment. The no-condition control skips the Utonia memory encoder and all point-attention residuals, including their gates and LoRA contribution. The resulting condition path retains the building-independent pretrained output bias, with native and image conditioning disabled.

Correct conditions yield $F_{.02}$ 91.46%, compared with 79.42% after feature permutation and 68.14% after feature removal. The corresponding paired gains, computed before rounding, are 12.03 points (95% interval $[11.44, 12.65]$) and 23.32 points (interval $[22.43, 24.20]$). Cross-building mismatch and removal of the entire condition pathway reduce $F_{.02}$ to 6.05% and 2.26%, respectively. These interventions demonstrate that the Utonia-replacement checkpoint uses observation identity and feature–position correspondence. They characterize inference behavior with fixed model weights; zero-feature and no-condition inputs lie outside the training distribution.

Each condition is evaluated on 3,840 outputs. The no-condition run has six empty outputs on six different buildings; all other outputs are valid. Empty outputs receive zero F-score and undefined CD. CD first averages the valid seeds within each building, then averages buildings; every building retains at least one valid seed, so all 1,280 buildings remain in the CD comparison. F-scores include every requested output. Apart from these six empty outputs, there are no invalid meshes, missing outputs, or execution failures in any control.

Geometry protocol. Inference starts from noise and runs 25 Euler steps with classifier-free guidance disabled before surface decoding. Marching cubes uses resolution 128, bounds $[-1.01, 1.01]^3$, and level zero. Each prediction is sampled with 100,000 area-uniform surface points and compared with the 40,960 uniform GT samples stored in the verified geometry cache. CD is half the sum of the

Table 10: Matched-subset wireframe evaluation on MunichWF: the same 4,887 buildings and local metric protocol for all rows. Bold values denote best performance, and underlined values are the second-best. Distances are in $\times 10^{-2}$; corner/edge scores are percentages.

Method	WED↓	ACO↓	CP↑	CR↑	CF1↑	EP↑	ER↑	EF1↑
BuildingGPT	**1.34**	**1.19**	<u>92.15</u>	**93.99**	<u>93.06</u>	<u>89.66</u>	**89.58**	<u>89.62</u>
Ours	<u>1.45</u>	<u>1.31</u>	**96.89**	<u>92.02</u>	**94.40**	**94.72**	<u>87.56</u>	**91.00**

Public-code WED. BuildingGPT uses seed 42; our C-derived generator with vertex and edge decoders averages seeds 7/17/27. Protocol and separate literature results: Appendix E.

two mean unsquared Euclidean nearest-neighbor distances. F-scores use strict distance thresholds of 0.01 and 0.02 in cached canonical coordinates. Observations and target geometry retain their independently cached normalizations, defining a canonical-shape evaluation. GT-latent encode–decode measurements quantify representation fidelity separately.

Runtime scope. Evaluation uses H20D GPUs, mixed bfloat16 precision, generation batch size 8, and decoder batch size 2. The recorded timings include generation, decoding, metrics, diagnostics, and test I/O, but exclude previously cached condition extraction. Whole-device peak memory includes the independent decoder. The reported runtime and parameter counts describe each configuration's compute requirements.

Qualitative configuration identity. The BuildingWorld3D qualitative examples in Appendix F use C49k, the Utonia-replacement configuration. The MunichWF experiment adapts this configuration to wireframe decoding. Each set of outputs is identified by its generating checkpoint and task.

Interpretation. The configuration comparisons measure reconstruction performance under a shared data and evaluation protocol. The condition interventions establish the sensitivity of the Utonia-replacement model to observation identity and feature–position correspondence. Differences in encoder features, adapter capacity, and computation are attributes of the compared configurations.

E MunichWF wireframe decoding

This study evaluates architectural structure readout on MunichWF, introduced by BuildingGPT from Munich LoD2 meshes used by PolyGNN (Liu et al., 2026; Chen et al., 2024c), using a separately task-adapted generator. The dataset, adaptation, and evaluation protocol are independent of the CityEXP surface-reconstruction study.

E.1 Adaptation and matched evaluation

Task adaptation. This experiment starts from the BuildingWorld-trained Omni C49k model. It is adapted for 5,000 steps on 77,288 MunichWF training buildings, with 1,579 separate validation buildings used for surface F-score selection. The vertex and edge decoders are trained separately from the generator and convert its latent predictions into wireframes. The matched evaluation uses this task-specific adaptation.

Evaluation. Table 10 compares BuildingGPT and Ours on the same 4,887 buildings under the local metric protocol. Inference uses observed points for conditioning and normalization. Our model uses generation seeds 7, 17, and 27 and 25 Euler steps; BuildingGPT uses seed 42. Corner and edge scores pool matched counts within each seed, then average seeds, using threshold 0.1 in source coordinates and 20 samples per edge. Corner matching is independent of edge matching. Table 11 includes five additional methods with literature results.

Results. Ours obtains CF1 94.40% and EF1 91.00%, compared with 93.06% and 89.62% for BuildingGPT under the matched protocol. Its corner and edge precision are higher, while recall

is lower. WED and ACO are also higher, showing the precision–recall and distance tradeoffs. Complete-target latent controls are reported below.

E.2 Scoring details and additional controls

Evaluation protocol and baseline sources. The literature comparison includes PC2WF, which predicts and prunes vertex and edge candidates (Liu et al., 2021), and Point2Roof, which reconstructs roofs from airborne points (Li et al., 2022). PBWR directly regresses parameterized edges (Huang et al., 2024); EdgeDiff generates edge sets through conditional diffusion (Liu et al., 2025a). BWFormer detects height-map corners before lifting and connecting them in 3D (Liu et al., 2025b). Their MunichWF scores below are transcribed from BuildingGPT Table 1.

Measured WED is public-code, bug-compatible WED, averaged over buildings and then seeds. WED was recovered from the model's archived graphs; all seven original corner/edge statistics were reproduced from those same predictions. Tables use two decimal places. Graph-corrected WED is 0.123464 for this model. It uses a different metric definition from the public-compatible WED reported in the table.

Table 11: MunichWF wireframe results. BuildingGPT and Ours are evaluated on the same 4,887-building subset; other rows are reported results on 5,000 buildings (Liu et al., 2026). Distances are in $\times 10^{-2}$; corner/edge scores are percentages. Results from different populations are shown for context and are not jointly ranked.

Method	WED↓	ACO↓	CP↑	CR↑	CF1↑	EP↑	ER↑	EF1↑
PC2WF	38.87	32.66	10.20	13.80	10.00	0.60	2.20	0.80
Point2Roof	4.10	3.75	87.30	49.20	61.50	83.80	32.10	43.90
PBWR	1.54	1.48	95.60	90.30	92.60	89.90	88.30	88.90
EdgeDiff	1.39	1.32	97.00	91.60	93.90	92.40	91.40	91.60
BWFormer	3.56	2.74	93.30	89.90	91.40	89.80	86.40	87.90
BuildingGPT	1.34	1.19	92.15	93.99	93.06	89.66	89.58	89.62
Ours	1.45	1.31	96.89	92.02	94.40	94.72	87.56	91.00

Complete-target latent controls. These controls decode Omni posterior-sampled complete-target latents. The vertex and edge decoders use the same checkpoints as in the end-to-end experiment.

Table 12: Our vertex/edge decoder diagnostics using complete-GT latents. ACO is in $\times 10^{-2}$; corner/edge scores are percentages. Full-test rows count the 113 target-preprocessing rejections as empty predictions.

Decoder	Test N	ACO↓	CP↑	CR↑	CF1↑	EP↑	ER↑	EF1↑
Ours	4,887	0.30	99.06	95.70	97.35	97.75	93.75	95.71
Ours	5,000	0.30	99.06	92.35	95.59	97.75	90.46	93.96

On 4,887 buildings, GT-latent CF1/EF1 are 97.35%/95.71%, versus 94.40%/91.00% with point-generated latents: gaps of 2.96/4.71 points before rounding. These empirical differences measure the effect of latent source. Full-population controls count 113 unavailable GT encodings as empty predictions. Seed averages quantify generation variability for fixed trained checkpoints.

E.3 Qualitative junction regularization

Figure 8 shows case DEBY_LOD2_4957372 from a separate native-conditioned configuration: an Omni DiT at step 20,500, a Hunyuan3D-2.1 latent representation, and MunichWF vertex/edge decoders at steps 3,002/4,503. This configuration differs from the Utonia-replacement model in Table 10. Generation uses 2,048 observed points, seed 7, 25 Euler steps, and no classifier-free guidance. Neither target latents nor target vertex proposals enter this prediction.

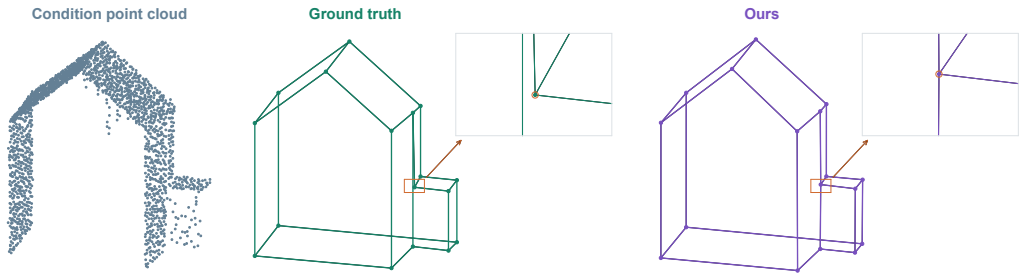

Figure 8: **Local junction regularization during reconstruction.** Input point cloud, reference wireframe, and the native-conditioned Omni/H21 reconstruction with deterministic intermediate-vertex edge blocking. The insets show a displaced reference junction and a more closely aligned predicted connection. The displayed junction reflects both the reconstructed wireframe and this post-processing step.

The readout blocks a candidate edge when another candidate vertex lies within 0.02 source-coordinate units of its interior. At the highlighted junction, the offset from the line through the corresponding upper and lower endpoints decreases from 0.01078 in the reference to 0.00364 in the prediction. This is a local regularization example from the full inference pipeline, including the deterministic edge rule; it does not isolate an effect of pretraining or establish the physical ground truth.

The case was selected near the per-case edge-F1 75th percentile. Reference and prediction share the same camera and scale, with matched inset crops and magnification. No manual geometry edits were applied.

F BUILDINGWORLD3D QUALITATIVE EXAMPLES

The six examples in Figure 9 were manually selected from a 48-building preview drawn from a frozen 1,280-building test evaluation. Selection emphasizes successful examples with nontrivial layouts, roof geometry, or height variation. All six use the same Utonia-replacement variant at step 49,000 and generation seed 7. The surfaces come from point-conditioned DiT sampling on BuildingWorld3D, before MunichWF-specific adaptation.

The input clouds use observed-point normalization, while the processed GT surfaces use the cached target normalization. The figure compares these normalized shapes. Each triplet uses the same camera and display scale, and both surface columns use the same material. We reuse the preview's original reverse-view renderings, preserving the generated geometry and the shared generation seed. Only empty image margins are trimmed for layout, identically across each triplet. Residual rounding, surface ripples, and lost fine roof detail remain visible.

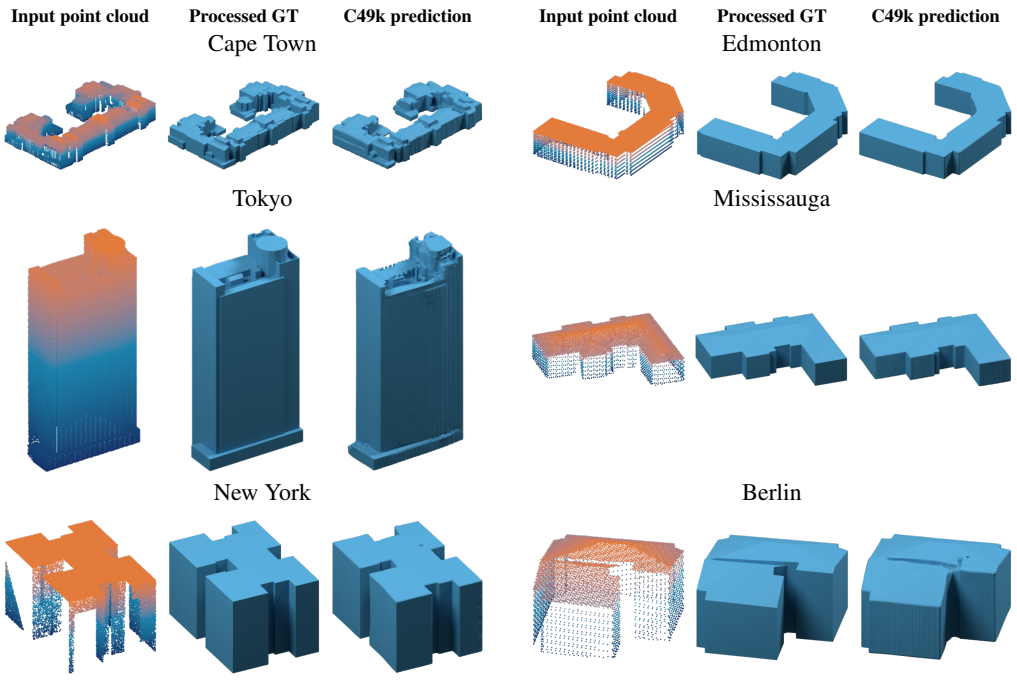

Figure 9: **Reverse views of all six selected BuildingWorld3D cases.** Each triplet shows the input, processed GT, and C49k Utonia-replacement prediction from the preview's fixed reverse camera.

Table 13: **CAD/IFC geometry-codec diagnostics.** F-scores (%) compare the same decoded mesh to the original triangulated source and the prepared encoder target, respectively. CC counts decoded connected components. Surface scores are computed for cases that pass preparation in the two selected collections.

Source model	Status	Original $F\uparrow$	Prepared $F\uparrow$	CC
Casa Martirana	Decoded	94.26	95.01	35
FZK House	Decoded	54.89	96.68	1
Gothic Vault	Decoded	99.97	99.96	1
Great Court Roof	Decoded	92.85	93.09	2,540
Klein Building	Rejected	—	—	—
Smiley West	Decoded	71.74	99.69	10
Haus G–H	Decoded	36.65	93.40	113
IfcOpenHouse	Decoded	69.84	97.42	2
Jasmin-Sun 105	Decoded	46.14	92.49	1
KIT Institute	Decoded	66.92	99.38	13
Niedriha	Decoded	47.21	93.61	1
Revit Basic	Rejected	—	—	—

G GEOMETRY-INTERFACE COMPATIBILITY AND PREPARATION LOSSES

Scope and protocol. We evaluate surface preparation and encoding–decoding on twelve CAD/IFC models from two six-model collections. Ten models yield decoded surfaces, and two fail the preparation checks. The completed runs encode prepared surfaces with the same frozen Hunyuan3D-Omni ShapeVAE. We quantify fidelity separately against the original geometry and the prepared encoder target.

IFC models are evaluated with their component placements and converted to triangle surfaces. The native CAD example uses a saved aggregate BRep from FreeCAD, using the saved boundary geometry. Preparation uses ManifoldPlus (Huang et al., 2020) at depth 8 and checks the resulting surface for connectivity and manifoldness. A geometric drift threshold of 2% is recorded as a diagnostic alongside the surface-validity checks. The encoder receives 40,960 area-uniform and 40,960 sharp-edge samples, drawn from 124,928 candidates per group. We use a posterior sample with fixed seeds and decode at marching-cubes resolution 256 using fixed pretrained weights.

Original and prepared references. Table 13 evaluates each decoded surface against both the original triangulated source and the prepared encoder target. Both comparisons use the same coordinate transform, with 32,768 surface samples per mesh and a distance threshold of 1% of the respective reference's longest bounding-box side. These relative-scale scores are reported relative to each reference's extent. The original reference includes the source's internal geometry. The Gothic Vault example uses finer triangulation to retain curvature at its declared source scale and remains one case in the collection.

The two CAD/IFC collections demonstrate both usable examples and substantial losses. Casa Martirana and Gothic Vault retain high original-reference F-scores. In contrast, KIT Institute scores 99.38% against the prepared target but 66.92% against the original source. All five completed cases in the second collection exceed the recorded preparation-drift threshold. These gaps expose source surfaces lost before encoding; prepared-target scores alone cannot establish full BIM fidelity.

We also assess connectivity. Great Court Roof has 2,540 decoded connected components, with 97.69% of surface area outside the largest component, despite an original-reference F-score of 92.85%. KIT Institute and IfcOpenHouse contain four and two nonmanifold edges, respectively. Klein Building and Revit Basic are rejected because their prepared outputs fail the combined connectivity/manifoldness check. The interface converts source geometry into a surface representation.

Selected cross-representation examples. Figure 10 shows a BuildingWorld3D LoD2 example, paired low- and high-detail geometry of one earlier CityEXP building, KIT Institute, and Casa Martirana. All five use the same frozen Omni VAE. The figure illustrates representation compatibility by

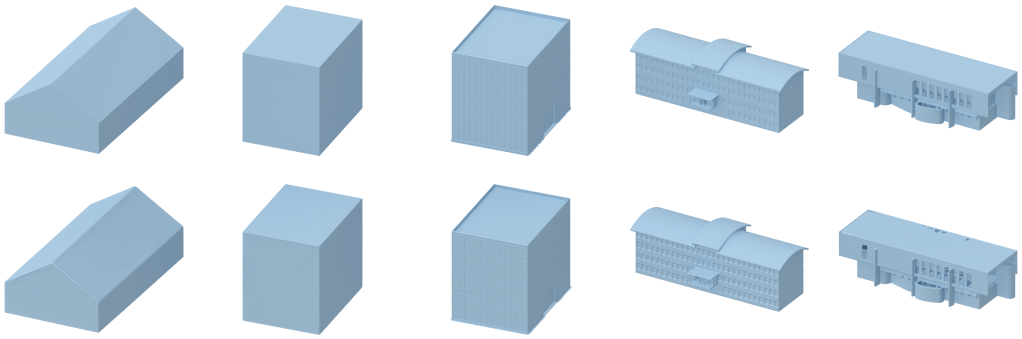

Figure 10: **Selected surface encoding–decoding examples.** Columns from left to right: BuildingWorld3D LoD2, CityEXP structural LoD, CityEXP detailed mesh of the same building, IFC/BIM (KIT Institute), and native CAD (Casa Martirana). Top: prepared encoder targets. Bottom: surfaces decoded by the same frozen Omni VAE. Each pair shares its camera and display scale; scales differ across columns. The inputs are prepared surfaces, and the outputs are geometry-codec reconstructions. KIT Institute: KIT/IAI. Casa Martirana: Yorik van Havre, CC BY 4.0.

comparing prepared targets with decoded surfaces. Different columns use different display scales; each input–output pair retains its view and scale.

Sources. Casa Martirana is the FreeCAD BIM example by Yorik van Havre (CC BY 4.0). FZK House, Smiley West, and KIT Institute are credited to KIT/IAI (Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, 2021). Gothic Vault, Klein Building, and the Great Court Roof approximation are Geometry Gym examples, with the latter obtained from the IfcOpenShell example mirror. IfcOpenHouse is an author-published example; Jasmin-Sun 105, Niedriha, Haus G–H, and Revit Basic come from the open BIM test-file collection. Redistribution follows each source's license and permission requirements.

H ADDITIONAL SURFACE AND WIREFRAME RECONSTRUCTIONS

Figures 11, 12, and 13 show selected reconstruction examples. Each row corresponds to one building. Columns show the input condition, ground-truth target, and prediction, respectively.

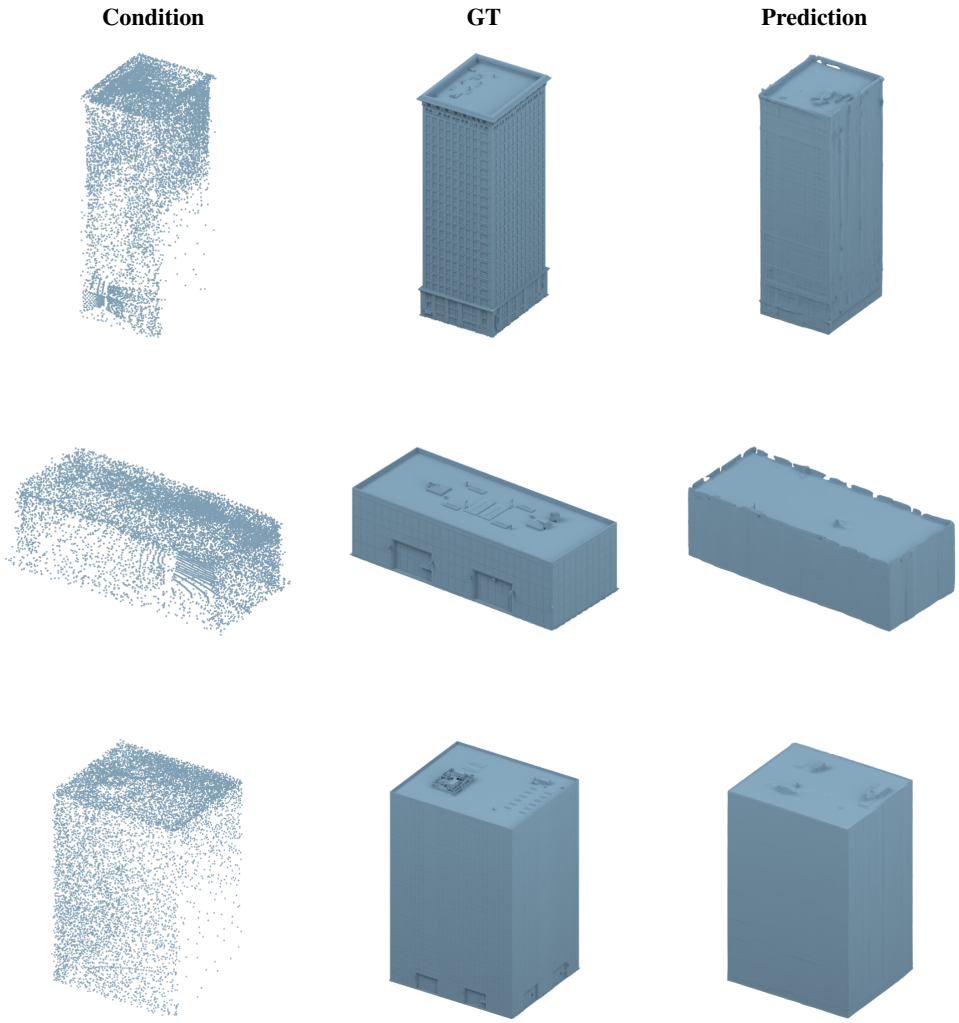

Figure 11: **Detailed-building surface reconstruction.** Predictions recover the overall building form, while facade and rooftop details remain smoother than the ground-truth targets.

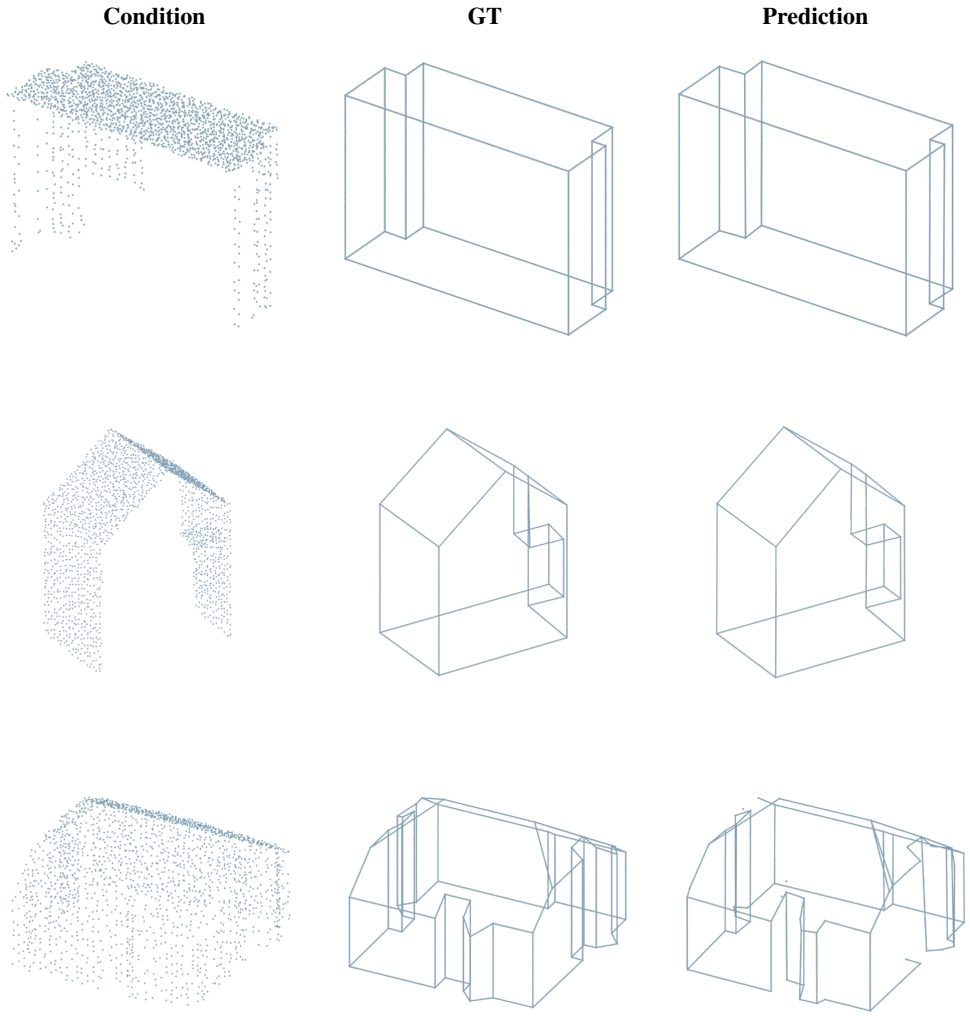

Figure 12: **Wireframe reconstruction examples.** The panels compare point conditions, reference graphs, and predicted vertices and edges. The bottom example retains visible missing and displaced connections.

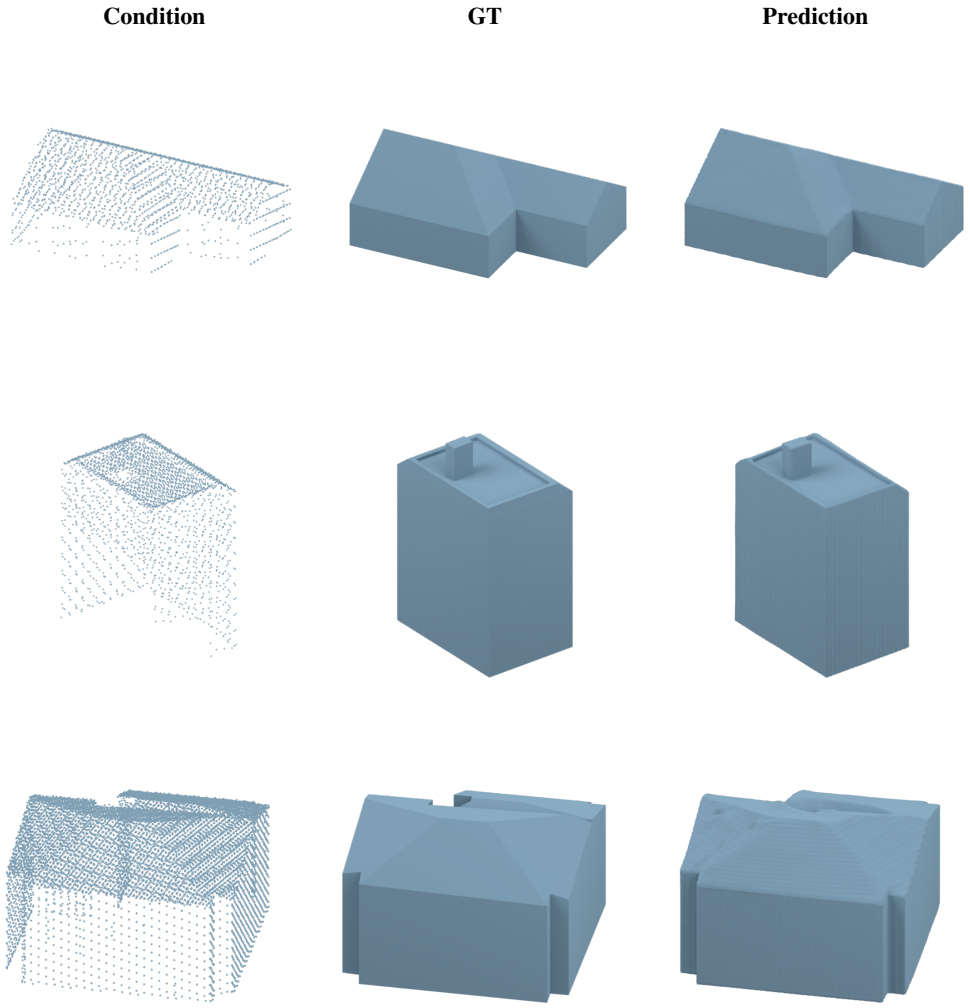

Figure 13: **Building-envelope reconstruction examples.** Point conditions, ground-truth surfaces, and predictions for buildings with different roof structures. Surface ripples and deviations from the reference roof geometry remain visible.